The Architecture of an RBridge Solution to TRILL
draft-ietf-trill-rbridge-arch-05.txt

Abstract

RBridges are link layer (L2) devices that use a routing protocol as a control plane. This combines several of the benefits of the link layer with those of the network layer. For example RBridges use existing link state routing, without necessarily requiring configuration, to improve aggregate throughput, for RBridge to
RBridge traffic. RBridges also may support IP multicast and IP address resolution optimizations. They are intended to be applicable to L2 network sizes similar to those of conventional bridges and are intended to be backward compatible with those bridges as both ingress/egress and transit. They also support VLANs (although this generally requires configuration) while otherwise attempting to retain as much ‘plug and play’ as is already available in existing bridges. This document proposes an architecture for RBridge systems as a solution to the TRILL problem, defines terminology, and describes basic components and desired behavior. One (or more) separate documents will specify protocols and mechanisms that satisfy the architecture presented herein.
# Table of Contents

1. Introduction ................................................ 4
2. Background .................................................. 7
   2.1. Existing Terminology .................................. 7
   2.2. RBridge Terminology ................................... 10
3. Components ................................................... 12
   3.1. RBridge Device ........................................ 13
   3.2. RBridge Data Model .................................... 14
      3.2.1. Unicast TRILL Forwarding Database .............. 14
      3.2.2. Multi-destination TRILL Forwarding Database .... 14
      3.2.3. Ingress TRILL Forwarding Database ............... 16
4. Functional Description ....................................... 17
   4.1. TRILL Campus Auto-configuration ....................... 17
   4.2. RBridge Peer Discovery ................................ 18
   4.3. Topology Discovery .................................... 18
   4.4. Determination of End Station Points of Attachment .. 18
   4.5. Learning .............................................. 19
   4.6. Tunneling ............................................. 19
5. RBridge Operation ........................................... 20
   5.1. RBridge General Operation ............................. 20
   5.2. Ingress/Egress Operations ............................. 21
   5.3. Transit Forwarding Operations ......................... 24
      5.3.1. Unicast .......................................... 25
      5.3.2. Broadcast, Multicast and Flooding .............. 25
         5.3.2.1. Broadcast .................................. 25
         5.3.2.2. Multicast .................................. 27
         5.3.2.3. Flooding .................................... 28
   5.4. Routing Protocol Operation ............................ 30
   5.5. Other Bridging and Ethernet Protocol Operations ..... 30
      5.5.1. Wiring Closet Problem ............................ 31
6. How RBridges Address the TRILL Problem Space ............... 32
7. Conclusions ................................................ 32
8. Security Considerations ................................... 33
9. IANA Considerations ....................................... 34
10. Acknowledgments .......................................... 34
11. References ................................................ 34
    11.1. Normative References ............................... 34
    11.2. Informative References ............................. 34
12. Author’s Addresses ........................................ 35

Intellectual Property Statement...............................
Disclaimer of Validity........................................
Copyright Statement..........................................
1. Introduction

This document describes an architecture that addresses the TRILL problem and applicability statement [2]. This architecture describes a solution that is composed of a set of devices called RBridges. RBridges cooperate together in an Ethernet network to provide a layer two delivery service that makes efficient use of available links using a link state routing protocol. The service provided is analogous to creation of a single, virtual device composed of an overlay of tunnels, constructed between RBridge devices, using paths determined by link state routing. RBridges thus support increased aggregate RBridge to RBridge bandwidth, and fault tolerance, when compared to conventional Ethernet bridges (which forward frames via a spanning tree, in a non-VLAN or single VLAN context, or multiple spanning trees), while still being compatible with bridges and hubs.

The principal objectives of this architecture is to provide an overview of the use of these RBridges in meeting the following goals:

1) Provide a form of optimized layer two delivery service.

2) Use existing technology as much as possible.

3) Allow for configuration free (or minimal configuration) deployment.

In providing a (optimized) layer two (L2) service, key factors we want to maintain are: transparency to higher layer (layer 3 and above) delivery services and mechanisms, and use of location independent addressing. Optimization of the L2 delivery service consists of: use of an optimized subset of all available paths and support for optimization of ARP/ND and pruning of multicast traffic delivery paths.

Not all optimizations are necessarily expected to be supported in initial specification and some subset of these optimizations may be specified at a later time. This architecture should allow some level of optimization support to be provided in compliant implementations, in as many case as possible.

To accomplish the goal of using existing technologies as much as possible, we intend to specify minimal extensions to an existing link-state routing protocol, as well as defining specific subsets of existing bridging technologies that this architecture is intended to makes use of.
The extent to which routing protocol extensions may be required depends on the closeness of the "fit" of the chosen routing protocol (in this case, IS-IS) to RBridge protocol requirements. The specific of routing protocol use - along with appropriate extensions and enhancements - will be defined in corresponding RBridge protocol specifications (see [3] for example).

Specific protocol specifications will also describe the details of interactions between the RBridge protocol and specific L2 technologies - i.e. - Virtual Local Area Networking (VLAN), L2 Multicast, etc. This document describes the general nature of the RBridge solution without restricting related specifications.

As an overview, however, the intention is to use a link-state routing protocol to accomplish the following:

1) Discover RBridge peers.

2) Determine RBridge link topology.

3) Potentially advertise L2 reachability information; note that - at this time - the default method for acquiring L2 reachability information specified in [3] depends on use of data-plane learning (see Bridge Learning in the terminology section below).

4) Establish L2 delivery using shortest path (versus STP, RSTP or MSTP).

There are additional RBridge protocol requirements - above and beyond those addressed by any existing routing protocol - that are identified in this document and need to be addressed in corresponding RBridge protocol specifications.

To allow for configuration free deployment, specific protocol specifications should explicitly define the conditions under which RBridges may - and may not - be deployed as-is (plug and play), and the mechanisms that are required to allow this. For example, the first requirement any RBridge protocol must meet is to derive information required by link-state routing protocol(s) for protocol start-up and communications between peers - such as higher-layer addressing and/or identifiers, encapsulation header information, etc.

At the abstract level, RBridges need to maintain the following information:
1) Peer information,

2) Topology information,

3) Forwarding information —
   a. unicast,
   b. flooded, and
   c. multicast.

In addition, RBridge specifications may suggest (or require) the maintenance of other information as needed to support ARP/ND and multicast optimizations.

Peer information may be acquired via the routing protocol, or may be discovered as a result of RBridge-specific peer discovery mechanisms. Details of specific peer information requirements — as well as how this information will be acquired is specified in protocol specifications (e.g. - [3]).

Topology information is expected to be acquired via the link-state routing protocol.

In addition to the requirement that the routing protocol should be an existing link-state routing protocol, which may provide a mechanism for (or re-use of an existing) neighbor/peer discovery, the routing protocol should be able to work with minimal (or no) configuration — using algorithmically derived addressing, for example, assuming the use of addresses is required.

Given the potential choices, IS-IS routing has been chosen at this time, and is assumed in this architecture. The fact that this is assumed in this architecture is — in no way — intended to preclude use of another link-state routing protocol, or any other routing protocol, in any solution not described in this architecture.

Forwarding information is derived from the combination of attached MAC address learning, snooping of multicast-related protocols (e.g. - IGMP), and routing advertisements and path computations using the link-state routing protocol.

Other information — such as the mapping of MAC and IP addresses, or multicast pruning information — may be learned using snooping
of ARP/ND or IGMP (for example) and it is possible that RBridges
may need to participate actively in these protocols.

The remainder of this document outlines the TRILL architecture
of an RBridge-based solution and describes RBridge components,
interactions and functions. Note that this document is not
intended to represent the only solution to the TRILL problem
statement, nor does it specify the protocols that instantiate
this architecture - or that only one such set of protocols is
prescribed. The former may be contained in other architecture
documents and the latter would be contained in separate
specification documents (see - e.g. - [3]).

2. Background

This architecture is based on the RBridge system described in an
Infocom paper [1]. That paper describes the RBridge system as a
specific instance; this document abstracts architectural
features only. The remainder of this section describes the
terminology of this document, which may differ from that of the
original paper.

2.1. Existing Terminology

The following terminology is defined in other documents. A brief
definition is included in this section for convenience and - in
some cases - to remove any ambiguity in how the term may be used
in this document, as well as in derivative documents intended to
specify components, protocol, behavior and encapsulation
relative to the architecture described in this document.

- IEEE 802.1D and IEEE 802.1Q: IEEE documents which include
  specification for bridged Ethernet, including Media Access
  Control (MAC) bridges and the BPDUs used in spanning tree
  protocol (STP) [5], [8].

- ARP: Address Resolution Protocol - a protocol used to find an
  address of form X, given a corresponding address of form Y.
  In this document, ARP refers to the well-known protocol used
  to find L2 (MAC) addresses, using a given L3 (IP) address.
  See [7] for further information on IP ARP.

- Bridge: an Ethernet (L2, 802.1D) device with multiple ports
  that receives incoming frames on a port and transmits them on
  zero or more of the other ports; bridges support both bridge
  learning and STP. Transparent bridges do not modify the L2
  PDU being forwarded.
o Bridge Learning: process by which a bridge determines on which (if any) single outgoing port to transmit (forward or copy) an incoming unicast frame. This process depends on consistent forwarding as "learning" uses the source MAC address of frames received on each interface. Layer 2 (L2) forwarding devices "learn" the location of L2 destinations by peeking at layer 2 source addresses during frame forwarding, and store the association of source address and receiving interface. L2 forwarding devices use this information to create "filtering database" entries and - gradually - eliminate the need for flooding.

o Bridge Protocol Data Unit (BPDU): the frame type associated with bridge control functions (for example: STP/RSTP).

o Bridged LAN: see IEEE 802.1Q-2005, Section 3.3 [8].

o Broadcast Domain: the set of (layer 2) devices that must be reached (or reachable) by (layer 2) broadcast traffic injected into the domain.

o Broadcast Traffic: traffic intended for receipt by all devices in a broadcast domain.

o Ethernet: a common layer 2 networking technology that includes, and is often equated with, 802.3.

o Filtering Database: database containing association information of (source layer 2 address, arrival interface). The interface that is associated with a specific layer 2 source address, is the same interface which is used to forward frames having that address as a destination. When a layer 2 forwarding device has no entry for the destination layer 2 address of any frame it receives, the frame is "flooded".

o Flooded Traffic: traffic that is subject to flooding - i.e. - being forwarded on all interfaces, except the one on which it was received, within a LAN or VLAN.

o Flooding: the process of forwarding traffic to ensure that frames reach all possible destinations when the destination location is not known. In "flooding", an 802.1D forwarding device forwards a frame for any destination not "known" (i.e. - not in the filtering or forwarding database) on every active interface except that one on which it was received. See also VLAN flooding and flooded traffic.
- Frame: in this document, frame refers to an Ethernet (L2) unit of transmission (PDU), including header, data, and trailer (or payload and envelope).

- Hub: Ethernet device with multiple ports that transparently transmits frames arriving on any port to all other ports. This is a functional definition, as there are devices that combine this function with certain bridge-like functions that may - under certain conditions - be referred to as "hubs".


- LAN: Local Area Network, is a computer network covering a small geographic area, like a home, office, or group of buildings, e.g., as based on IEEE 802.3 technology, see also IEEE 802.1Q-2005, Section 3.11 [8].

- MAC: Media Access Control - mechanisms and addressing for L2 frame forwarding.

- Multicast Forwarding: forwarding methods that apply to frames with broadcast or multicast destination MAC addresses.

- Node: a device with an L2 (MAC) address that sources and/or sinks L2 frames.

- Packet: in this document, packet refers to L3 (or above) data transmission units (PDU - e.g. - an IP Packet (RFC791 [4]), including header and data.

- PDU: Protocol Data Unit - unit of data to be transmitted by a protocol. To distinguish L2 and L3 PDUs, we refer to L2 PDUs as "frames" and L3 PDUs as "packets" in this (and related) document(s).

- Router: a device that performs forwarding of IP (L3) packets, based on L3 addressing and forwarding information. Routers forward packets from one L2 broadcast domain to another (one, or more in the IP multicast case) - distinct - L2 broadcast domain(s). A router terminates an L2 broadcast domain.

- Spanning Tree Protocol (STP): an Ethernet (802.1D) protocol for establishing and maintaining a single spanning tree among all the bridges on a local Ethernet segment. Also, Rapid Spanning Tree Protocol (RSTP). In this document, STP and RSTP are considered to be the same.
SPF: Shortest Path First - an algorithm name associated with routing, used to determine a shortest path graph traversal.

TRILL: Transparent Interconnect over Lots of Links - the working group and working name for the problem domain to be addressed in this document.

Unicast Forwarding: forwarding methods that apply to frames with unicast destination MAC addresses.

Unknown Destination - a destination for which a receiving device has no filtering database entry. Destination (layer 2) addresses are typically "learned" by (layer 2) forwarding devices via a process commonly referred to as "bridge learning" (see definition above).

VLAN: Virtual Local Area Network, see IEEE 802.1Q-2005 [8].

VLAN Flooding: flooding as described previously, except that frames are only forwarded on those interfaces configured for participation in the applicable VLAN.

### 2.2. RBridge Terminology

The following terms are defined in this document and intended for use in derivative documents intended to specify components, protocol, behavior and encapsulation relative to the architecture specified in this document.

Adjacent RBridges: RBridges that communicate directly with each other without relay through other RBridges.

Cooperating RBridges: a set of communicating RBridges that will share a consistent set of forwarding information.

Designated RBridge (DRB): one approach to resolving several issues associated with having multiple RBridges as candidate ingress and egress points for a single LAN (or VLAN) is to designate an RBridge to act as the ingress and egress in that case. The RBridge designated to handle ingress and egress traffic to a specific Ethernet link (or VLAN associated with that link) having shared access and multiple RBridges is such a link’s "Designated RBridge", or DRB. The Designated RBridge may (for example) be determined by an election process among those RBridges having shared access via a single LAN, or may be selected by some other means.
- Edge RBridge (edge of a TRILL Campus): describes RBridges that may serve to ingress frames into the TRILL Campus and egress frames from the TRILL Campus. L2 frames transiting an TRILL Campus enter, and leave, it via an edge RBridge.

- Egress RBridge: for any specific frame, the RBridge through which that frame leaves the TRILL Campus. For frames transiting a TRILL Campus, the egress RBridge is an edge RBridge where RBridge encapsulation is removed from the transit frames prior to exiting the TRILL Campus.

- Encapsulation database: in the TRILL context, the database that the Ingress RBridge uses to map the layer 2 destination address in the received frame to the egress RBridge.

- Forwarding Tunnels: in this document, Campus Forwarding Tunnels (or Forwarding Tunnels) is used to refer to the paths for forwarding transit frames, encapsulated at an RBridge ingress and decapsulated at an RBridge egress.

- Ingress RBridge: for any specific frame, the RBridge through which that frame enters the TRILL Campus. For frames transiting a TRILL Campus, the ingress RBridge is the edge RBridge where RBridge encapsulation is added to the transit traffic entering the TRILL Campus.

- Multi-Destination Frames: Broadcast or Multicast frames, or Unicast frames destined to a MAC DA that is unknown i.e. - flooded frames (see flooded traffic). Frames that need to be delivered to multiple egress RBridges, via the RBridge Distribution Tree.

- Peer RBridge: The term "Peer RBridge", or (where usage is not ambiguous) the term "Peer", are used in the RBridge context to refer to any of the RBridges that make up a TRILL campus.

- RBridge: a logical device as described in this document, which incorporate both routing and bridging features, thus allowing for the achievement of TRILL Architecture goals. A single RBridge device which can cooperate with other RBridge devices to create a TRILL Campus.

- RBridge Distribution Tree: This term or (where usage is not ambiguous) the term "distribution tree", refers to a tree used by RBridges to deliver multi-destination frames. An RDT, or distribution tree, is computed using a specific RBridge as the root. May also be referred to as an R-tree.
o TRILL Campus: this term, or the term "Campus" (where usage is not ambiguous) is used in the RBridge context to refer to the set of cooperating RBridges and TRILL Links that connect them to each other.

o TRILL Forwarding Database: this term, or the term "forwarding database" (where not ambiguous) is used in an RBridge context to refer to the database that maps the egress TRILL address to the next hop TRILL link.

o TRILL Header: a ‘shim’ header that encapsulates the ingress L2 frame and persists throughout the transit of a TRILL Campus, which may be further encapsulated within a hop-by-hop L2 header (and trailer). The hop-by-hop L2 encapsulation in this case includes the source MAC address of the immediate upstream RBridge transmitting the frame and destination MAC address of the receiving RBridge – at least in the unicast forwarding case.

o TRILL Link: this term, or the term "Link" (where its usage is not ambiguous) is used in the RBridge context to refer to the Layer 2 connection that exists either between RBridges, or between an RBridge and Ethernet end stations.

3. Components

A TRILL Campus is composed of RBridge devices and the forwarding tunnels that connect them; all other Ethernet devices, such as bridges, hubs, and nodes, operate conventionally in the presence of an RBridge.

```
+-------------------------------+                   +-------------------------------+
|               Higher Layer Entities               |
+-------------------------------+                   +-------------------------------+
| \ TRILL Layer | RBridge Relay Entity | TRILL Layer / |
+-------------------------------+                   +-------------------------------+
| Data Link Layer |                               | Data Link Layer |
+-------------------------------+                   +-------------------------------+
| Physical Layer |                               | Physical Layer |
+-------------------------------+                   +-------------------------------+
          | P 1                   |                               | P 2

Figure 1: Simplified Architecture of an RBridge

Figure 1 shows an RBridge that contains:
An RBridge Relay Entity connecting two RBridge ports

At least one physical port (two in this example)

Higher layer Entities, including at least the IS-IS protocol

At the TRILL Layer, an RBridge encapsulates incoming Ethernet frames with a TRILL header to forward them to other RBridges.

3.1. RBridge Device

An RBridge is a device - having some of the characteristics of both bridges and routers - that forwards frames on an Ethernet link segment. It has one or more Ethernet ports which may be wired or wireless; the particular physical layer is not relevant. An RBridge is defined more by its behavior than its structure, although it logically contains three tables, which may be used to describe the externally visible behavior of an RBridge relative to its peers and may also distinguish RBridges from conventional bridges.

Conventional bridges contain a learned filtering (or forwarding) database, and spanning tree port state information. The bridge learns which nodes are accessible from a particular port by assuming bi-directional consistency: the source addresses of incoming frames indicate that the incoming port is to be used as output for frames destined to that address. Incoming frames are checked against the learned filtering (forwarding) database and forwarded to the particular port if a match occurs, otherwise they are flooded out all active ports (except the incoming port).

Spanning tree port state information indicates the ports that are active in the spanning tree. Details of STP operation are out of scope for this document, however the result of STP is to disable ports which would otherwise result in more than one path traversal of the spanning tree.

RBridges, by comparison, have a TRILL forwarding database, used for forwarding of RBridge encapsulated frames across the TRILL Campus and by the ingress RBridge to determine the encapsulation to use for frames received as un-encapsulated from non-RBridge devices. The TRILL forwarding database is described in the following sections.
3.2. RBridge Data Model

The following tables represent the logical model of the data required by RBridges in forwarding unicast and multicast data across a TRILL Campus.

3.2.1. Unicast TRILL Forwarding Database

The Unicast TRILL Forwarding Database is a forwarding table for unicast traffic within the TRILL Campus, allowing tunneled traffic to transit the TRILL Campus from ingress to egress. The size of a fully populated Unicast TRILL Forwarding Database at each RBridge is maximally bounded by the product of the number of Adjacent RBridge peers and VLANs.

RBridges may have separate Unicast TRILL Forwarding Databases for each VLAN, if this is supported by configuration. Note that scaling concerns may dictate otherwise, either in specific of RBridge protocol specification, or in deployment. The Unicast TRILL Forwarding Database is continually maintained by RBridge routing protocols and/or MAC learning. (see Section 5.4).

The Unicast TRILL Forwarding Database contains data specific to RBridge forwarding for unicast traffic. The specific fields contained in this table are to be defined in RBridge protocol specifications. In the abstract, however, the table should contain forwarding direction and encapsulation associated with an RBridge encapsulated frame received - determined by the TRILL "shim" header destination and VLAN (if applicable).

3.2.2. Multi-destination TRILL Forwarding Database

The Multi-destination TRILL Forwarding Database consists of a set of forwarding entries used for support of RBridge Distribution Trees (RDT). Multi-destination TRILL Forwarding Database entries are distinct from typical Unicast TRILL Forwarding Database entries because there may be zero or more of them that match for any incoming frame.

The Multi-destination TRILL Forwarding Database may overlap the Unicast TRILL Forwarding Database, or be instantiated as a separate table, in specific compliant implementations.

In discussing entries to be included in the Multi-destination TRILL Forwarding Database, the following entities are temporarily defined, or further qualified:
o Root RBridge - the RBridge that is the head end of an RDT. All RBridges within a TRILL Campus are potential Root RBridges.

o Egress RBridge - an RBridge that is the tail end of a path corresponding to a specific Multi-destination TRILL Forwarding Database entry. All RBridges within a TRILL Campus are potential egress RBridges. Not all RBridges within a TRILL Campus will be on the shortest path between any ingress RBridge and any other egress RBridge.

o Local RBridge - the RBridge that forms and maintains the Multi-destination TRILL Forwarding Database entry (or entries) under discussion. The local RBridge may be a root RBridge, or an egress RBridge with respect to any set of entries in the Multi-destination TRILL Forwarding Database.

o RBridge TRILL Campus Egress Interface - an interface on any RBridge where a transit RBridge encapsulated frame would be decapsulated prior to forwarding. With respect to such an interface, the local RBridge is the egress RBridge.

Each local RBridge will maintain - as a logical representation - a set of entries for at least the following, corresponding to a subset of all possible forwarding paths:

o Zero or more entries grouped for each root RBridge - keyed by some root RBridge identifier - used to determine forwarding of broadcast, multicast, and flooded frames originally RBridge encapsulated by that ingress within the TRILL Campus.

o Corresponding to each of these entry groups, one entry for each of zero or more egress RBridge - where the local RBridge is on the shortest path toward that egress RBridge.

o Corresponding to each of these entry groups, one entry for each of zero or more TRILL Campus egress interfaces.

Each entry would contain an indication of which single interface a broadcast, multicast or flooded frame would be forwarded for each (root RBridge, egress RBridge) pair. Entries would also contain any required encapsulation information, etc. required for forwarding on a given interface, and toward a corresponding specific egress RBridge.
Note that the above information is one logical representation of the information required to perform a reverse path forwarding check (or RPFC) as is discussed in [3].

A local RBridge could maintain a full set of entries from every RBridge to every other RBridge, however - depending on topology - only a subset of these entries would ever be used. In addition, a topology change that changed selection of shortest paths would also very likely change other elements of the entries, negating possible benefits from having pre-computed Multi-destination TRILL Forwarding Database entries.

Multi-destination TRILL Forwarding Database entries should also include VLAN identification information relative to each set of Root RBridges, to allow scoping of broadcast, multicast and flooding forwarding by configured VLANs.

Multi-destination TRILL Forwarding Database entries may also include Multicast-Group Address specific information relative to each egress RBridge that is a member of a given well-known multicast group, to allow scoping of multicast forwarding by multicast group.

Implicit in this data model is the assumption that the TRILL "shim" header encapsulation will contain information that explicitly identifies the TRILL Campus ingress RBridge for any broadcast, multicast or flooded frame.

Maintenance of this Multi-destination TRILL Forwarding Database will be defined in appropriate protocol specifications used to instantiate this architecture. Note that doing this does not strictly require those specification to adopt this data model. The protocol specification needs to include mechanisms and procedures required to establish and maintain the Multi-destination TRILL Forwarding Database in consideration of potential SPF recomputations resulting from network topology changes.

### 3.2.3. Ingress TRILL Forwarding Database

The Ingress TRILL Forwarding Database determines how arriving traffic will be encapsulated, for forwarding toward the egress RBridge, via the TRILL Campus. It becomes configured in much the same way that bridge learning occurs: by snooping incoming traffic, and assuming bi-directional consistency.
This learned information at an egress RBridge may be propagated to all other RBridges in the TRILL Campus via the RBridge routing protocol, as an alternative to direct MAC learning from data frames. However, the information propagated in this fashion may be quite large and filtering to prevent overwhelming edge RBridges would require extensive per-VLAN state information in core RBridges. Hence the current model is that the default mode for learning L2 reachability information is via learning from the data plane directly in a manner very analogous to bridge learning.

Using this approach, the ingress TRILL Forwarding Database may be as large as the number of nodes on the Ethernet LAN, for all VLANs in which a specific ingress RBridge is a participant.

The Ingress TRILL Forwarding Database essentially determines the tunnel encapsulation used to transport each specific frame across the TRILL Campus, for frames entering at this ingress.

4. Functional Description

The RBridge Architecture is largely defined by RBridge behavior; the logical components are minimal, as outlined in Section 3.

4.1. TRILL Campus Auto-configuration

Cooperating RBridges self-organize to compose a single TRILL Campus system. The details for how this occurs are given in protocol specification(s).

At an architectural level, it is sufficient to note that every end station attached to a TRILL Campus should have a primary point of attachment to the TRILL Campus, as might be defined (for example) by a Designated RBridge. Furthermore, if it is possible that there are 802.1Q (or 802.1D) bridges in a local LAN, the association of specific RBridges and the end stations for which they act as primary point of attachment must be determined in a way that is consistent with forwarding in an 802.1Q (or 802.1D) network. If a DRB election process were used, each TRILL Link (or VLAN set within a TRILL Link) attached to a TRILL Campus would have a single Designated RBridge (either for the link, or for a subset of the VLANs on the link); using DRBs as an example, the DRB would be where all traffic intended to transit a TRILL Campus enters and exits.

Other approaches might be used as well. For example, the primary point of attachment for each end station (or set of end

Gray Expires August, 2008 [Page 17]
stations), might be configured, or determined in some other way, on a per end station (or set of end stations) basis. Any such approach must either allow for the possibility of LAN bridges or not be used when such a possibility exists.

This rule applies strictly on a per-VLAN basis.

High-level steps that may be included in auto-configuration are: RBridge peer discovery, topology discovery, determination of end station points of attachment (DRB election, for example), learning and forwarding (tunneling) TRILL encapsulated frames.

4.2. RBridge Peer Discovery

Proper operation of the TRILL solution using RBridges depends on the existence of a mechanism for discovering peer RBridges. Failure to discover all peer RBridges leads inevitably to an incomplete discovery of the RBridge topology.

RBridge peer discovery can be accomplished in a relatively easy re-use of well-known techniques based on broadcast - such as the use of IS-IS "hello" messages.

4.3. Topology Discovery

Proper operation of RBridges also depends on the existence of a mechanism for determining the RBridge topology. An accurate determination of RBridge topology is required in order to determine how traffic frames will flow in the topology and thus avoid the establishment of persistent loops in frame forwarding, or construction of a partitioned local LAN.

Fortunately, accurate topology determination is a fundamental requirement of a functioning link-state routing protocol. The complexity that applies in this architecture directly relates to the existence of multiple VLANs on a TRILL Link.

For this reason, RBridges (in terms of protocol definition, implementation and deployment) should avoid unnecessary use of multiple VLANs - in particular on links that will be, or may be, used for transit of TRILL encapsulated frames.

4.4. Determination of End Station Points of Attachment

The mechanisms and details of how end stations are associated with a specific RBridge - when multiple RBridges are available (connected to a local LAN or VLAN) - for the purpose of
providing ingress to and egress from the Rbridge Campus, will be provided by protocol specification(s). A potential approach to be considered is the use of a "Designated RBridge (DRB) Election", on either a per link, or per VLAN (set) basis.

Architecturally, it is important to note that this determination must be based on an accurate view of the topology, including availability of certain links in a given topology for traffic associated with any given VLAN. Otherwise, it is possible to partition a TRILL Link-local LAN (assuming that RBridges may be deployed and configured to replace existing 802.1Q bridges) as a result of a failure - under circumstances in which such a partition would not have occurred with previously deployed 802.1Q bridges.

Protocol specification(s) need to define how accurate VLAN topology is to be determined and applied in determination of end station primary points of attachment. Protocol specification(s) also need to state the limitations that any chosen mechanisms may impose on the solution (in terms of scalability and ease of deployment, for example). Finally, protocol specification(s) need to describe how determination is made with respect to which RBridge(s) are responsible for learning new end station information, and for flooded and broadcast frames for reaching known and unknown end stations on any link or VLAN.

4.5. Learning

The protocol specifications need to define how learning of MAC-layer reachability information is expected to occur - at least in the default case.

As described previously, a major consideration is the complexity associated with receiving reachability information for a lot of end-stations for which an ingress RBridge has no interest. This is the case, for example, where a large number of VLANs are in use (see [8]). This issue does not arise if learning is based on the data plane (similar to bridge learning) - as is currently described as a default learning mode in [3].

4.6. Tunneling

RBridges pass encapsulated frame traffic to each other effectively using tunnels. These tunnels use an Ethernet link layer header, together with a TRILL header.
Specifics of encapsulation are to be defined in appropriate protocol/encapsulation specifications.

It is the combination of the local MAC destination (which is for a locally attached RBridge) and the TRILL encapsulation that distinguishes RBridge to RBridge traffic from other traffic. The link-layer header includes source and destination addresses, which typically identify the local RBridges (the sending and receiving RBridges relative to the local TRILL Link).

The TRILL header is required to support loop mitigation for (at least) unicast traffic within the TRILL Campus; traffic loops in forwarding between RBridges and non-RBridge devices, as well as across non-RBridge devices between RBridges, is beyond the scope of this document.

The TRILL header and encapsulation:

- must clearly identify the traffic as RBridge traffic - the outer Ethernet header may, for instance, use an Ethertype number unique to RBridges;
- should also identify a specific (egress) RBridge - the TRILL header may, for example, include an identifier unique to the egress RBridge, in the unicast case;
- should include the RBridge transit route, a hopcount, or a timestamp to prevent indefinite looping of a frame.

5. RBridge Operation

This section is intended primarily to serve as a tutorial for RBridge operations. As such in any case where this section says anything in disagreement with specific protocol specifications, the protocol specification over-rides.

5.1. RBridge General Operation

As described in sections above, operations that apply to all RBridges include peer and topology discovery (including hello messaging, negotiation of RBridge identifiers and link-state routing), determination of RBridge primary points of attachment (for local end stations - possibly via DRB election), shortest path (SPF) computation and either learning or advertising reachability for specific L2 (MAC Ethernet destination) addresses within a broadcast domain.
In addition, all RBridges will compute RBridge Distribution Trees for delivery of (potentially VLAN scoped) broadcast, multicast and flooded frames to each peer RBridge. Setting up these trees early is important as there is otherwise no means for frame delivery across the TRILL Campus during the learning phase. Because it is very likely to be impossible (at an early stage) for RBridges to determine which RBridges are edge RBridges, it is preferable that each RBridge compute these trees for all RBridges as early as possible – even if some entries will not be used.

The specifics of each of these operational steps will be defined in protocol specifications (such as [3]).

5.2. Ingress/Egress Operations

Operation specific to edge RBridges involves RBridge learning, advertisement, encapsulation and decapsulation (at ingress and egress RBridges, respectively).

As described previously, RBridge learning is similar to typical bridge learning – i.e. - all RBridges listen promiscuously to L2 Frames on each local LAN and acquire end station location information associated with source MAC addresses in L2 frames they observe.

If multiple RBridges are available (i.e. - connected to a local LAN or VLAN), a determination must be made as to which RBridge is the primary point of attachment for each end station (or end stations by groups – using, for example, the VLAN associations of VLAN aware end stations) requiring ingress and egress services from an RBridge Campus. This is a minimal requirement, and there is no reason why this same determination might not be made in all cases, even the degenerate case in which only one RBridge may be used. This could be the case, for instance, if a DRB election is done in all cases, including when the DRB can only be the one and only RBridge to which local end stations might be attached.

In the degenerate case – where only one RBridge is connected to a specific Ethernet LAN – obviously that RBridge will be (or become) the primary point of attachment for local end stations requiring ingress/egress services from the RBridge Campus.

Minimally, only the RBridge determined as the primary point of attachment for a given (set of) end station(s) is required to perform RBridge learning for that (set of) end station(s) while
they remain associated with interface(s) connected to the local LAN or VLAN.

Note that - in some cases - the determination of primary points of attachment and learning may be tightly bound operations. This might be the case if - for example - the determination is based on literal configuration of end station and RBridge attachments. In other cases, learning would occur in a more conventional - and flexible - way, if (for example) an automated process of selecting a DRB (such as DRB election) is used on a per-link or per-VLAN (set) basis.

Assuming learning is required by a specific solution described in any protocol specification(s), as RBriges learn segment-local MAC source addresses, it creates an entry in its learned filtering/forwarding database that associates that MAC source address with the interface on which it was learned.

Similarly - to support ARP/ND optimization - IP-to-MAC mappings may also be learned by snooping corresponding protocol messages. Protocol specifications may include either optional or required behaviors to support ARP/ND, or multicast, learning and distribution methods.

Periodically, as determined by RBridge protocol specification, each RBridge may advertise this learned information to its RBridge peers. These advertisements would propagate to all edge RBridges (as potentially scoped by associated VLAN information for each advertisement). Each edge RBridge would incorporate this information in the form of a Unicast TRILL Forwarding Database entry.

Note that currently, [3] specifies that this is not the default mode, and that learning primarily occurs via the data plane at egress, as well as at ingress.

The trade-off is between the complexity associated with flooding data verses the complexity associated with flooding reachability information.

For applications in which it is likely that most edge RBridges will not want to receive most of the reachability information, flooding avoidance requires either that the method is not used, or that intermediate (core, in at least some cases) RBridges need to keep VLAN specific state information to limit the scope of advertisement flooding.
If it is desired - in any specific solution - to support discovery of new end station attachments, RBridges may also discover that a new end station has become locally attached (for which they may be, or become, an edge RBridge) as a result of receiving un-encapsulated frames that require forwarding. Any such solution must specify how a primary point of attachment is determined when this occurs. Several possible approaches exist.

If an automated DRB selection process (such as DRB election) is the approach in use for a specific solution (on a per-link, or per-VLAN, basis), this determination is automatic for any link (or VLAN on a local link). If an RBridge is the DRB for a local link or VLAN, and has not previously learned that the MAC destination for a frame is local (this could be the case - for instance - for the very first frame it observes), then the RBridge could be required to forward (or flood) the frame via the TRILL Campus to all other RBridges (potentially within a VLAN scope).

When a frame is received, which must be forwarded across the RBridge Campus, the responsible RBridge would flood the frame unless it has already created a Unicast TRILL Forwarding Database entry for the frame’s MAC destination address. If it has a corresponding Unicast TRILL Forwarding Database entry, then it would use that. The RBridge in this instance would be an ingress RBridge for the frame being forwarded across the RBridge Campus.

The encapsulation used by this ingress RBridge would be determined by the Unicast TRILL Forwarding Database entry - if one exists - or the Unicast TRILL Forwarding Database-equivalent entry for the RBridge Distribution Tree.

When the encapsulated frame arrives at egress RBridge(s), it is decapsulated and forwarded via the egress interface(s) onto the local link (or VLAN on the local link).

In using the approach of learning from the data plane, the egress RBridge stores information related to content of the frame’s TRILL encapsulation for use in subsequent reverse traffic in a manner directly analogous to bridge learning.

Note that an egress RBridge will - in most case - be the RBridge determined to be the primary point of attachment for a destination end station on the local link or VLAN accessed via its egress interface(s). Exceptions to this might exist under circumstances in which use of distinct RBridges for ingress and
egress for a common (set of) end station(s) does not produce local forwarding ambiguity. Any protocol specification that allows this must also unambiguously describe the precise circumstances under which it is allowed - as well as the limitations and issues this introduces in the solution specified.

Also note that specific solution(s) in protocol specification(s) will need to describe how determination of an end station’s primary point of attachment (RBridge) occurs for the case where a specific end station has not yet been discovered at any ingress or egress interface - except under circumstances where discovery of new end stations is not supported, or explicitly disallowed. In the example in which a DRB election is used, this determination is both trivial and automatic. In an approach where end station and RBridge attachment/association is configured, this should be relatively obvious - if inflexible.

In the DRB example, if the destination MAC address of a received frame does not correspond to a learned MAC destination address at an egress interface, it will forward the frame on all interfaces for which it is either the designated RBridge. If a received frame does correspond to a learned MAC destination address at an egress interface, the RBridge will forward the frame via that interface only.

Any specific solution’s protocol specification(s) should also allow for the fact that flooded frames may arrive at a single local LAN (or VLAN) - where local end stations may be attached - via multiple R Bridges. Protocol specification(s) need to account for how determination of which RBridge is exclusively responsible for forwarding such frames is to be made. This is essential to avoid having the same frame arrive multiple times and potentially from widely disparate directions (i.e. - on different interfaces of individual bridges).

5.3. Transit Forwarding Operations

There are two models for transit forwarding within a TRILL Campus: unicast frame forwarding for known destinations, and everything else. The difference between the two is in how the encapsulation is determined. Exactly one of these models will be selected - in any instantiation of this architecture- for each of the following forwarding modes:

- Unicast frame forwarding
- Forwarding of non-unicast frames
5.3.1. Unicast

In unicast forwarding, the TRILL header is specific to the egress RBridge and MAC destination in the outer Ethernet encapsulation is specific to the next hop RBridge.

As the frame is prepared for transmission at each RBridge, the next hop MAC destination information is determined at that local RBridge using a corresponding Unicast TRILL Forwarding Database entry based on the TRILL "shim" header.

5.3.2. Broadcast, Multicast and Flooding

RBridge Distribution Trees are used for forwarding of broadcast, multicast and unknown destination frames across the TRILL Campus. In a simple implementation, it is possible to use the Multi-destination TRILL Forwarding Database entries for all frames of these types.

However, this approach results in possibly severe inefficiencies in at least the multicast case.

As a consequence, instantiations of this architecture should allow for local optimizations on a hop by hop basis.

Examples of such optimizations are included in the sections below.

5.3.2-1. Broadcast

The path followed in transit forwarding of broadcast frames will have been established through actions initiated by each RBridge (as any RBridge is eligible to subsequently become an ingress RBridge) in the process of computing Multi-destination TRILL Forwarding Database entries.

The protocol specification will most likely require each RBridge to assume that it may be a transit as well as an ingress and egress RBridge and establish forwarding information relative to itself and each of its peer R Bridges, and stored in the Multi-destination TRILL Forwarding Database. At least one exception case exists and that is when RBridges are configured to treat a given link as a point to point link between two RBridges.
Forwarding information should logically exist in two forms:
transit encapsulation information for interfaces over which the 
RBridge will forward a multipoint frame to one or more adjacent 
RBridges and a decapsulation indication for each interface over 
which the RBridge may egress frames from the TRILL Campus. In 
each case, the Multi-destination TRILL Forwarding Database 
includes some identification of the interface on which a frame 
is forwarded toward any specific egress RBridge for frames 
received from any specific ingress RBridge.

Note that an interface over which an RBridge may egress frames 
is any interface for which the RBridge is the primary point of 
end station attachment for one or more end stations, or the 
RBridge determined to be responsible for broadcast delivery.

RBridges must not wait to determine that one (or more) Ethernet 
end stations are present on an interface before deciding to 
forward decapsulated broadcast frames on that interface. Again, 
an exception case would exist if RBridges have been configured 
to treat a local link as a point to point connection between two 
RBridges, or otherwise configured to ignore possible presence of 
end stations in this case.

Forwarding information is selected for each broadcast frame 
received by any RBridge (based - for example - on identifying 
the ingress RBridge, or distribution tree root, for the frame) 
for all corresponding Multi-destination TRILL Forwarding 
Database entries. Each RBridge may thus be required to replicate 
one RBridge encapsulated broadcast frame for each interface that 
is determined from Multi-destination TRILL Forwarding Database 
entries corresponding to the frame’s ingress RBridge (or 
distribution tree root). This includes decapsulated broadcast 
frames for each interface for which the RBridge is responsible 
for providing egress for broadcast frames (as might have been 
determined previously by DRB election, for example).

Note that frame replication and forwarding should be scoped by 
VLAN, if VLAN support is provided. Also note that an egress 
RBridge may be required to transmit a decapsulated frame on the 
same interface on which it previously received the corresponding 
RBridge encapsulated frame.

This approach for broadcast forwarding might be considered to 
add complexity because replication occurs at all RBridges along 
the ingress RBridge tree, potentially for both RBridge 
encapsulated and decapsulated broadcast frames. However, the 
replication process is similar to replication of broadcast
traffic in 802.1D bridges with the exception that additional replication may be required at each interface for egress from the TRILL Campus.

Note that the additional replication associated with TRILL Campus egress may be made to exactly conform to 802.1D bridge broadcast replication in implementations that model a TRILL Campus egress as a separate logical interface.

Using this approach results in one and only one copy of the broadcast frame being delivered to each egress RBridge.

5.3.2-2. Multicast

Multicast forwarding is reducible to broadcast forwarding in the simplest (default) case. However, protocol specifications may require, or recommend and implementations may choose - using mechanisms that are out of scope for this document - to optimize multicast forwarding. In order for this to work effectively, however, support for awareness of multicast "interest" is required for all R Bridges.

Without optimization, multicast frames are injected by the ingress RBridge onto an RDT by - for instance - encapsulating the frame with a MAC destination multicast address, and forwarding it according to its local Multi-destination TRILL Forwarding Database. Again, without optimization, each RBridge along the path toward all egress R Bridges will similarly forward the frame according to their local Multi-destination TRILL Forwarding Database.

Using this approach results in one and only one copy of the multicast frame being delivered to appropriate egress R Bridges. However, using this approach, multicast delivery is identical to broadcast delivery - hence very inefficient.

In any optimization approach, RBridge encapsulated multicast frames will use either a broadcast or a group MAC destination address. In either case, the recognizably distinct destination addressing allows a frame forwarding decision to be made at each RBridge hop. RBridges may thus be able to take advantage of local knowledge of multicast distribution requirements to eliminate the forwarding requirement on interfaces for which there is no recipient interested in receiving frames associated with any specific group address.
As stated earlier, in order for RBridges to be able to implement multicast optimization, distribution of learned multicast group "interest" information must be provided — and propagated — by all RBridges. Mechanisms for learning and propagating multicast group participation by RBridges is out of scope in this document but may be defined in RBridge protocol specification(s).

Note that, because the multicast optimization would — in principle — further scope and reduce broadcast traffic, two things may be said:

- It is not necessary that all implementations in a deployment implement the optimization (though all must support the data required to implement it in RBridge peers) in order for any local multicast optimization (consistent with the above description) to work;
- Introduction of a multicast optimization will not result in potential forwarding loops where broadcast forwarding would not do so.

In the simplest case, the ingress RBridge for a given multicast frame will re-use the MAC destination group address of a received multicast frame. However this may not be required as it is possible that the mechanisms specified to support multicast will require examination of the decapsulated MAC destination group address at each RBridge that implements the optimization.

Specifics of multicast forwarding are to be defined in protocol specifications.

5.3.2-3. Flooding

Flooding is similarly reducible to broadcast forwarding in the simplest (default) case — with the exception that a frame being flooded across the TRILL Campus is typically a unicast frame for which no Unicast TRILL Forwarding Database entry exists at the ingress RBridge. This is not a minor distinction, however, because it impacts the way that addressing may be used to accomplish flooding within the TRILL Campus.

An ingress RBridge that does not have a Unicast TRILL Forwarding Database entry for a received frame MAC destination address, will inject the frame onto the ingress RBridge Tree by — for instance — encapsulating the frame with a MAC destination broadcast address, and forwarding it according to its local Multi-destination TRILL Forwarding Database. Without
optimization, each RBridge along the path toward all egress
RBridges will similarly forward the frame according to their
local Multi-destination TRILL Forwarding Database.

Using this approach results in one and only one copy of
the flooded frame being delivered to all egress RBridges.

However implementations may choose to optimize flooding. A
Flooding optimization will only work at any specific RBridge if
that RBridge re-evaluates the original (decapsulated) unicast
frame.

Any flooding optimization would operate similarly to the
multicast optimization described above, except that - instead of
requiring local information about multicast distribution - each
RBridge implementing the optimization will need only to lookup
the MAC destination address of the original (decapsulated) frame
in its local Unicast TRILL Forwarding Database. If an entry is
found, the frame could then be forwarded only if the specific
RBridge is on the shortest path between the originating ingress
RBridge and the appropriate egress RBridge. This could be
implemented - for example - as a specialized Multi-destination
TRILL Forwarding Database entry.

Note that, because a flooding optimization would - in principle
- further scope and reduce flooded traffic, two things may be said:

- It is not necessary that all implementations in a deployment
  support the optimization in order for any local flooding
  optimization (consistent with the above description) to work
  (hence such an optimization is optional);
- Introduction of the flooding optimization will not result in
  potential forwarding loops where flooded forwarding would not
  do so.

Because a forwarding decision can be made at each hop, it is
possible to terminate flooding early if a Unicast TRILL
Forwarding Database for the original MAC destination was in the
process of being propagated when flooding for the frame was
started. It is therefore possible to reduce the amount of
flooding to some degree in this case.

Specifics of a flooding optimization - beyond the above proof of
the concept that such a thing could be done safely - is out of
scope for this document and should be out of scope generally in
all protocol specifications for which the above analysis holds.
5.4. Routing Protocol Operation

The details of routing protocol operation are determined by the choice to use IS-IS routing. These details would be defined in appropriate protocol specification(s). Protocol specifications in this case may include both RBridge protocols (such as [3]), and specifications offering a generalized enhancement to IS-IS.

Protocol specifications should identify the means by which IS-IS meets the peer and topology discovery, and path computation needs of the specific protocol - including which IS-IS optional features and enhancements (if any) are required for support of specified RBridge operations.

5.5. Other Bridging and Ethernet Protocol Operations

In defining this architecture, several interaction models have been considered for protocol interaction between RBridges and other L2 forwarding devices - in particular, 802.1D bridges. Whatever model we adopt for these interactions must allow for the possibility of other types of L2 forwarding devices. Hence, a minimal participation model is most likely to be successful over the long term, assuming that RBridges are used in a L2 topology that would be functional if RBridges were replaced by other types of L2 forwarding devices.

Toward this end, RBridges - and the TRILL Campus as a whole - could (in theory) participate in Ethernet link protocols, notably the spanning tree protocol (STP) on the ingress/egress links using exactly one of the following interaction models:

- Transparent Participation (Transparent-STP)
- Active Participation (Participate-STP)
- Blocking Participation (Block-STP)

Only one of these variants would be supported by an instance of this architecture. All RBridges within a single TRILL Campus must use the same model for interacting with non-RBridge protocols. Furthermore, it is the explicit intent that only one of these models is ultimately supported - at least as a default mode of compliant implementations.

This architecture assumes RBridges block STP.
5.5.1. Wiring Closet Problem

There is at least one remaining issue with this assumption and that has been referred to as the "wiring closet problem." The essential problem is described in this subsection.

Given this configuration of bridges in a wiring closet, and an RBridge core:

```
-----> B-1 <---------------> RB-a <-----.
                  |                                  > RBridge CORE
                  |                                  /
-----> B-2 <---------------> RB-b <-----'
```

The link between (802.1D) bridges B-1 and B-2 is meant to be disabled by STP. In the RBridge case, however, there is no indication (from STP) that this link is redundant. Moreover, in order to avoid breaking bridge learning, either RB-a or RB-b will be the DR and - as a result, only one of the links (B-1<=>RB-a, B-2<=>RB-b) will get used.

One solution to this problem is to include - as a configuration option, for instance - the ability to enable negotiation of (or use of a pre-defined, or configurable) pseudo-bridge identifier to be used in any of the variations of STP.

One - (near) zero-configuration - option we've considered would be to use a well-known bridge identifier that each RBridge would use as a common pseudo-bridge identifier. Such an ID, used in combination with other STP configuration parameters, would most likely have to be guaranteed to win the root bridge election process in order to be a reasonable and useful default.

However, because this architecture assumes RBridges block STP, participation in any form of STP is assumed to take place in an in-line, co-located bridge function. Such a bridge function is in addition to RBridge architectural functionality described in this document. Implementations may include such functionality and will very likely require some minimal configuration to turn it on, in vendor specific RBridge implementations. An example of a minimal configuration would be to assign a pseudo-bridge identifier to (the local in-line co-located bridge associated with) a specific RBridge port.

For reasons of interoperability, specific protocol proposals to address the needs of this architecture may specify exactly how a
co-located bridge will operate in this case (if such co-located bridge functionality is included in an implementation), as well as whether or not inclusion of such co-location is required.

As a further note, one of the problems that should be addressed - assuming that this problem is to be resolved - is how to make certain the solution is robust against configuration error. In any solution that requires configuration of a pseudo-bridge ID that is common across a TRILL Campus, for example, it is possible to guard against configuration errors by using an election process (based on the root bridge election process) to determine which configured ID will be used by all RBridges in common - assuming that multiple pseudo-bridge IDs are inadvertently configured.

Finally, note that there is a chicken-and-egg problem associated with RBridge participation in STP where RBridges may themselves be connected by spanning trees.

6. How RBridges Address the TRILL Problem Space

The RBridge architecture addresses the following aspects of the requirements identified in reference [2] through the use of a link-state routing protocol and defined forwarding behaviors:

- Inefficient Paths
- Robustness to Link Interruption

In addition, using a logical model of "separation of functions" this architecture allows specifications and implementations to address existing and developing Ethernet extensions and enhancements, and provides a background against which protocol specifications may address: concerns about convergence under dynamic network changes, and optimizations for VLAN, ARP/ND, Multicast, etc.

7. Conclusions

This document discusses options considered and factors affecting any protocol specific choices that may be made in instantiating the TRILL architecture using RBridges.

Specific architectural and protocol instantiations should take these into consideration. In particular, protocol, encapsulation and procedure specifications should allow for potential
optimizations described in the architectural document to the maximum extent possible.

Also, this document addresses considerations relative to interaction with existing technology and "future-proofing" solutions. For both simplicity in description, and robust long term implementation of the technology, this document recommends the use of clear distinction - at all possible points - of definitions, protocols, procedures, etc. from related (but not identical) specifications and interactions.

In particular, this document recommends the use of a "collocation model" in addressing issues with combining RBridge, Router and 802.1D bridge behavior.

8. Security Considerations

As one stated requirement of this architecture is the need to be able to provide an L2 delivery mechanism that is potentially configuration free, the default operation mode for instances of this architecture should assume a trust model that does not require configuration of security information. This is - in fact - an identical trust model to that used by Ethernet devices in general.

In consequence, the default mode does not require - but also does not preclude - the use of established security mechanisms associated with the existing protocols that may be extended or enhanced to satisfy this document’s architectural definitions.

In general, this architecture suggest the use of a link-state routing protocol - modified as required to support L2 reachability and link state between RBridges. Any mechanisms defined to support secure protocol exchanges between link-state routing peers may be extended to support this architecture as well.

This architecture also suggests use of additional encapsulation mechanisms and - to the extent that any proposed mechanism may include (or be extended to include) secure transmission - it may be desirable to provide such (optional) extensions.

To the extent possible, any extensions of protocol or encapsulation should allow for at least one mode of operation that doesn’t require configuration - if necessary, for limited use in a physically secure deployment.
9. IANA Considerations

This document has no direct IANA considerations. It does suggest, that protocols that instantiate the architecture use a TRILL header as a wrapper on the payload for RBridge to RBridge traffic, and this TRILL header may be identified by a new Ethertype in the tunneled Ethernet link header. This Ethertype, identified in an Ethernet header, could be allocated by the IEEE.

10. Acknowledgments

The initial work for this document was largely done by Joe Touch, based on work he and Radia Perlman completed earlier. Subsequent changes are not to be blamed on either of them.

In addition, the current text has been helped substantially by comments and suggestions from the TRILL working group, working group chairs, and from Ron Bonica, Stewart Bryant, Joel Halpern, Guillermo Ibanez and Russ White in particular. Also, a great deal of work was recently done - by Joe Touch, Radia Perlman, Dinesh Dutt and Silvano Gai - in an effort to align terminology and concepts used in this document with those also used in the other TRILL documents.

11. References

11.1. Normative References

None.

11.2. Informative References


12. Author's Addresses

Editor:

Eric Gray
Ericsson
900 Chelmsford Street
Lowell, MA, 01851
Phone: +1 (978) 275-7470
EMail: Eric.Gray@Ericsson.com

Contributors:

Joe Touch
USC/ISI
4676 Admiralty Way
Marina del Rey, CA 90292-6695, U.S.A.
Phone: +1 (310) 448-9151
EMail: touch@isi.edu
URL: http://www.isi.edu/touch

Radia Perlman
Sun Microsystems
EMail: Radia.Perlman@sun.com

Intellectual Property Statement

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

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.

Disclaimer of Validity

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

Copyright Statement

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.

Acknowledgment

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