INTERNET-DRAFT
Intended Status: Proposed Standard
Expires: July 7, 2013

Radia Perlman
Intel Labs
Fanwei Hu
ZTE Corporation
Donald Eastlake
Huawei
Kesava Vijaya Krupakaran
Dell
January 3, 2013

TRILL Smart Endnodes
draft-perlman-trill-smart-endnodes-01

Abstract

This draft addresses the problem of the size and freshness of the endnode learning table in access RBridges, by allowing endnodes to volunteer for endnode learning and encapsulation/decapsulation. Such an endnode is known as a "smart endnode". Only the attached RBridge can distinguish a "smart endnode" from a "normal endnode". The smart endnode uses the nickname of the attached RBridge, so this solution does not consume extra nicknames.

Status of this Memo

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

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

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

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

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

Copyright and License Notice

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

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

Table of Contents

1 Introduction

The IETF TRILL (Transparent Interconnection of Lots of Links) protocol implemented by devices called RBridges (Routing Bridges, [RFC6325]), provides optimal pair-wise data frame forwarding without configuration, safe forwarding even during periods of temporary loops, and support for multipathing of both unicast and multicast traffic. TRILL accomplishes this by using IS-IS([RFC1195]) ([RFC6165]) ([RFC6326bis])link state routing and encapsulating traffic using a header that includes a hop count. Devices that implement TRILL are called "RBridges" (Routing Bridges) or TRILL Switches.

An RBridge that attaches to endnodes is called an "edge RBridge", whereas one that exclusively forwards encapsulated frames is known as a "transit RBridge". An edge RBridge traditionally is the one that encapsulates a native Ethernet packet with a TRILL header, or that receives a TRILL-encapsulated packet and removes the TRILL header. To encapsulate, the edge RBridge must keep an "endnode table" consisting of (MAC, TRILL egress switch nickname) pairs, for those MAC addresses currently communicating with endnodes to which the edge RBridge is attached.

These table entries might be configured, received from ESADI, looked up in a directory, or learned from received traffic. If the edge RBridge has many attached endnodes, this table could become large. Also, if one of the MAC addresses in the table has moved to a different switch, it might be difficult for the edge RBridge to notice this quickly, and because the edge RBridge is tunneling to the incorrect egress RBridge, the traffic will get lost.

For these reasons, it is desirable for an endnode E (whether it be server, hypervisor, or VM) to maintain the endnode table for nodes that E is corresponding with. This eliminates the need for the attached RBridge R to know about those nodes (unless some non-smart endnode attached to R is also corresponding with those nodes), and it enables E to immediately discard an entry of (D, egress nickname), if E cannot talk to D. Then E can attempt to acquire a fresh entry for D by flooding to D, listening for ESADI, or consulting a directory.

The mechanism in this draft has E issue a TRILL-Hello (even though E is just an endnode), indicating E's desire to act as a smart endnode, together with the set of MAC addresses that E owns, and whether E would like to receive ESADI. E learns from R's Hello, whether R is capable of having a smart endnode neighbor, what R's nickname is, and which trees R can use when R ingresses frames. Although E transmits TRILL-Hellos, E does not transmit or receive LSPs.

R will accept already-encapsulated packets from E (perhaps verifying that the source MAC is indeed one of the ones that E owns, that the ingress RBridge field is R's, and if the packet is an encapsulated multidestination frame, whether the tree selected is one of the ones that R has claimed it will choose). When R receives (from the campus) a TRILL-encapsulated packet with R's nickname as egress, R checks whether the MAC address in the inner packet is one of the MAC addresses that E owns, and if so, R forwards the packet onto E's port, keeping it encapsulated.

1.1 Terminology

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

2. Added information in TRILL-Hello

Suppose endnode E is attached to RBridge R. In order for E to act as a smart endnode, both E and R have to be signaled. The logical choice of message to do this in is a TRILL-Hello.

For smart endnode operation, R's TRILL-Hello must contain the following information:

* flag indicating willingness to have an attached smart endnode

* R's nickname (already included)

* trees that R can use when ingressing frames

* new TLV for smart endnode neighbor list

* set of { ({set of RBridge nicknames}, pseudonode nickname) pairs}, which is a pseudonode nickname that can be used if the smart endnode is multihomed to all of the RBridge nicknames listed.

E's TRILL-Hello must contain the following information:

* I don't want to form an RB-adjacency; merely to be a smart endnode

* For each VLAN
  (1) The set of MAC addresses I own
  (2) Whether I wish to receive ESADI for that VLAN

Note that smart endnode E does not issue LSPs, nor does it receive LSPs or calculate topology. E does the following:

o E maintains an endnode table of (MAC, nickname) of end nodes with which the smart endnode is communicating. If E is attached to multiple VLANs (traditional 12 bit VLANs or 24-bit FGL Fine Grained Labels), there would be a separate (MAC, nickname) table for each VLAN/FGL that E is attached to. Entries in this table are populated the same way that an edge RBridge populates the entries in its table:

  * learning from (source, ingress) on packets it decapsulates

  * from ESADI([TRILL-ESADI])

  * by querying a directory

  * by having some entries configured

o When E wishes to transmit to unicast destination D, if (D, nickname) is in E's endnode table, E encapsulates with ingress nickname=R, egress nickname as indicated in D's table entry. If D is unknown, D either queries a directory or encapsulates the packet as a multidestination frame, using one of the trees that R has specified in R's TRILL-Hello.

o When E wishes to transmit to a multicast destination, E encapsulates the packet using one of the trees that R has specified.

The attached RBridge R does the following:

o When receiving an encapsulated frame from a port with a smart endnode, with R's nickname as ingress, R forwards the packet to the specified egress nickname, as with any encapsulated packet. However, R MAY enforce that the inner source MAC and VLAN (or FGL) are as specified for the smart endnode, by dropping if the MAC (or VLAN/FGL) are not among the expected set from the smart endnode.

3. Hello Exchange with RBridges

The smart endnode E need not send Hellos as frequently as normal RBridges. These hellos MAY be periodically unicast to the Appointed Forwarder R. In case R crashes and restarts, or the DRB changes, and E sees a Hello without mentioning E, then E SHOULD send a Hello immediately. If R is AF for any of the VLANs that E claims, R MUST list E in its Hellos as a smart endnode neighbor.

4. Multi-homing

Now suppose E is attached to the TRILL campus in two places; to RBridges R1 and R2.

There are two ways for this to work:

(1) E can choose either R1 or R2's nickname, when encapsulating a frame, whether the encapsulated frame is sent via R1 or R2. If E wants to do active-active load splitting, and uses R1's nickname when forwarding through R1, and R2's nickname when forwarding through R2, this will cause distant RBridges (or smart endnodes) to keep changing their endnode table entry for D between (D, R1's nickname) and (D, R2's nickname). So it would be preferable for E to always encapsulate using the same nickname (R1 or R2) unless E detects a problem with connectivity using that nickname. And in this case, R1 and R2 need to be informed that the smart endnode might encapsulate with a different nickname, i.e., R1 might receive an encapsulated packet from smart endnode E using ingress nickname "R2".

(2) R1 and R2 might indicate, in their Hello, another nickname that attached end nodes may use if they are multihomed to R1 and R2, separate from R1 and R2's nicknames (which they would also list in their Hello). This would be useful if there were many end nodes multihomed to the same set of RBridges. This would be analogous to a pseudonode nickname; return traffic would go via the shortest path from the source to the endnode, whether it is R1 or R2. If E loses connectivity to R2, then E would revert to using R1's nickname. This does use a nickname, but hopefully would be shared by many end nodes multihomed to the same set of RBridges.

5. Encapsulation and Decapsulation

Consider a smart endnode E on a shared LAN wishing to communicate with D. First suppose D is not on the shared LAN. The draft already handles that case.

Suppose D is on the same shared LAN as smart node E. If E does not know where D is, the packet needs to be flooded BOTH on the shared LAN as a native packet, and throughout the campus, encapsulated.

(1) If E does not know where D is, then E sends two copies of the packet; one native, and one encapsulated.

(2) If the Appointed Forwarder R receives a native packet on a port with smart endnode E, and the source MAC is one that E owns, then

R MUST discard the packet.

(3) If R receives a native packet on a port with smart endnode E, and the destination MAC is one that E owns, then R MUST discard the packet.

(4) The other non-AFs in the shared LAN behave as usual - they don't encapsulate native frames.

This solution works regardless of whether D is a smart endnode or not. Smart endnode E will learn that D is on the shared link, and keep in its table (D, native on my link). So in the future, E will send to D by transmitting natively. R MUST discard the packet because it notices the source MAC is owned by E. D will transmit to E natively, whether or not D is a smart endnode. R will also discard the packet in this case because the destination MAC is owned by E. So D and E will talk natively.

If R receives a multicast from a remote RBridge, and the exit interface includes hybrid endnodes, it should send two copies of mulicast frames, one as native and the other as TRILL encapsulated frame. When smart endnode receives the encapsulated frame, it learns the remote address.

6. Security Considerations

For general TRILL Security Considerations, see([RFC6325]).

7. IANA Considerations

This document requires no IANA actions.

8. References

8.1 Normative References

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

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

[RFC6325] R. Perlman, D. Eastlake, et al, "RBridges: Base Protocol Specification", RFC 6325, July 2011.

[RFC6165] Banerjee, A. and D. Ward, "Extensions to IS-IS for Layer-2 Systems", RFC 6165, April 2011.

[RFC6326bis] D. Eastlake, A. Banerjee, et al, "Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS", draft-eastlake-isis-rfc6326bis-09.txt, work in progress.

[Directory] Linda, D., Eastlake, D., Perlman, R., and I. Gashinsky, "TRILL Edge Directory Assistance Framework", trill-directory-framework-01 (work in process).

[TRILL-ESADI] Zhai, H., Hu, F., Perlman, R., and D. Eastlake, "TRILL(Transparent Interconnection of Lots of Links): The ESADI (End Station Address Distribution Information) Protocol", draft-ietf-trill-esadi-01(work in process).

Authors' Addresses

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

Phone: +1-408-765-8080

Email: Radia@alum.mit.edu

Fangwei Hu
ZTE Corporation
No.889 Bibo Rd
Shanghai, 201203
China

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

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

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

Kesava Vijaya Krupakaran
Dell
Olympia Technology Park,
Guindy Chennai 600 032
India

Phone: +91 44 4220 8496
Email: Kesava_Vijaya_Krupak@Dell.com