

Bridge Functions Consortium

TRILL Interoperability Test Suite

Technical Document

Revision 0.7



**University of New Hampshire
InterOperability Laboratory
Bridge Functions Consortium
<http://www.iol.unh.edu>**

**121 Technology Drive, Suite 2
Durham, NH 03824-4716
Phone: +1-603-862-3525
Fax: +1-603-862-4181**

© 2010 University of New Hampshire InterOperability Laboratory

MODIFICATION RECORD

Draft Version 0.1 Complete	December 25 th , 2009
	<ul style="list-style-type: none">• Initial Release
Draft Version 0.6 Complete	March 16 th , 2010
	<ul style="list-style-type: none">• Public Release Candidate
Draft Version 0.7 Complete	March 30 th , 2010
	<ul style="list-style-type: none">• Added M Bit check to Test End Node Address Learning and Forgetting• Changed the Bridge Priorities to legal values in Test BPDU Handling• Renumbered VLANs on Test Multiple VLANs• Numerous typos fixed and clarification added

ACKNOWLEDGMENTS

The University of New Hampshire would like to acknowledge the efforts of the following individuals in the development of this test suite.

Authors:

David Bond University of New Hampshire

Commentators:

Timothy Carlin University of New Hampshire
Benjamin Long University of New Hampshire
Stephen Murphy University of New Hampshire
Aaron Stewart University of New Hampshire
Timothy Winters University of New Hampshire
Kari Younsi University of New Hampshire

INTRODUCTION

Overview

The University of New Hampshire's InterOperability Laboratory (IOL) is an institution designed to improve the interoperability of standards-based products by providing an environment where a product can be tested against other implementations of a standard. This suite of tests has been developed to help implementers evaluate the functioning of their TRansparent Interconnection of Lots of Links (TRILL) implementations. The tests do not determine if a product conforms to the specifications, nor are they purely interoperability tests. Rather, they provide a method to isolate problems within a device. Successful completion of all tests contained in this suite does not guarantee that the tested device will operate with other TRILL devices. However, these tests provide a reasonable level of confidence that the RBridge system will function well in most multi-vendor TRILL environments.

Acronyms

Acronyms used in this Test Suite:

BPDU: Bridge Protocol Data Unit
CSNP: Complete Sequence Number Packet
DRB: Designated RBridge
ESADI: End Station Address Distribution Information
IGMP: Internet Group Management Protocol
IP: Internet Protocol
IS-IS: Intermediate System to Intermediate System
LAN: Local Area Network
LSP: Link State Packet
MAC: Media Access Control
MLD: Multicast Listener Discovery
MTU: Maximum Transmission Unit
STP: Spanning Tree Protocol
TES: Testing End Station
TRB: Testing Routing Bridge
TRILL: TRansparent Interconnection of Lots of Links
VLAN: Virtual LAN

When several entities of the same type are present in a test configuration, a number is appended to the acronym to yield a label for each entity. For example, if there were three testing routing bridges in the test configuration, they would be labeled TRB0, TRB1 and TRB2.

TEST ORGANIZATION

This document organizes tests by group based on related test methodology or goals. Each group begins with a brief set of comments pertaining to all tests within that group. This is followed by a series of description blocks; each block describes a single test. The format of the description block is as follows:

- Test Label:** The test label and title comprise the first line of the test block. The test label is composed by concatenating the short test suite name, the group number, and the test number within the group, separated by periods. The **Test Number** is the group and test number, also separated by a period.
- Purpose:** The Purpose is a short statement describing what the test attempts to achieve. It is usually phrased as a simple assertion of the feature or capability to be tested.
- References:** The References section lists cross-references to the specifications and documentation that might be helpful in understanding and evaluating the test and results.
- Test Setup:** The Test Setup section describes the configuration of all devices prior to the start of the test. Different parts of the procedure may involve configuration steps that deviate from what is given in the test setup. If a value is not provided for a protocol parameter, then the protocol's default is used for that parameter.
- Procedure:** This section of the test description contains the step-by-step instructions for carrying out the test. These steps include such things as enabling links, unplugging devices from the network, or sending frame from a test station. The test procedure also cues the tester to make observations, which are interpreted in accordance with the observable results given for that test part.
- Observable Results:** This section lists observable results that can be examined by the tester to verify that the system is operating properly. When multiple observable results are possible, this section provides a short discussion on how to interpret them. The determination of a pass or fail for each test is usually based on how the systems behavior compares to the results described in this section.
- Possible Problems:** This section contains a description of known issues with the test procedure, which may affect test results in certain situations.

REFERENCES

The following documents are referenced in this text:

- [802.1Q-2005] IEEE Standard for Local and metropolitan area networks / Virtual Bridged Local Area Networks, 802.1Q-2005
- [IGMPv3] Request for Comments 3376 – Internet Group Management Protocol, Version 3
- [IPv4] Request for Comments 791 – Internet Protocol DARPA Internet Program Protocol Specification
- [IPv6] Request for Comments 2460 - Internet Protocol, Version 6 (IPv6) Specification
- [IS-IS] Request for Comments 1142- OSI IS-IS Intra-domain Routing Protocol
- [MLDv2] Request for Comments 3810 - Multicast Listener Discovery Version 2 (MLDv2) for IPv6
- [SNOOP] Request for Comments 4541 - Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches
- [TRILL] draft-ietf-trill-rbridge-protocol-14 – Rbridges: Base Protocol Specification

TABLE OF CONTENTS

MODIFICATION RECORD	2
ACKNOWLEDGMENTS	3
INTRODUCTION	4
REFERENCES	6
TABLE OF CONTENTS.....	7
COMMON TEST SETUP	9
Common Topology.....	9
Extended Common Topology.....	10
Transit Topology	10
Extended Transit Topology	11
Common Test Setup	12
Common Test Cleanup.....	12
GROUP 1: BASIC INTEROPERABILITY	13
Test TRILL_Interop.1.1: IS-IS	14
Test TRILL_Interop.1.2: Designated RBridge Election.....	15
Test TRILL_Interop.1.3: Incremental Deployment Functionality	17
Test TRILL_Interop.1.4: End Node Address Learning and Forgetting.....	18
Test TRILL_Interop.1.5: Nickname Collision	20
Test TRILL_Interop.1.6: Campus Wide MTU.....	22
Test TRILL_Interop.1.7: IP Snooping.....	23
Test TRILL_Interop.1.8: BPDU Handling.....	25
Test TRILL_Interop.1.9: Hop Count Handling.....	27
Test TRILL_Interop.1.10: RBridge Loss and Link Loss Handling.....	28
GROUP 2: FORWARDING.....	31
Test TRILL_Interop.2.1: Ingress Encapsulation	32
Test TRILL_Interop.2.2: Transit Forwarding	34
Test TRILL_Interop.2.3: Egress Decapsulation.....	36
GROUP 3: DISTRIBUTION TREES.....	38
Test TRILL_Interop.3.1: Shortest Path First Calculation.....	39
Test TRILL_Interop.3.2: Root Choice	42
Test TRILL_Interop.3.3: Number of Trees to Calculate	45
Test TRILL_Interop.3.4: Set of Trees to Calculate	46
Test TRILL_Interop.3.5: Tie Breaking	48
Test TRILL_Interop.3.6: No Receivers Pruning	49
GROUP 4: VLANS.....	50
Test TRILL_Interop.4.1: VLAN Ingress Encapsulation	51
Test TRILL_Interop.4.2: VLAN Transit Forwarding	53

*University of New Hampshire
InterOperability Laboratory*

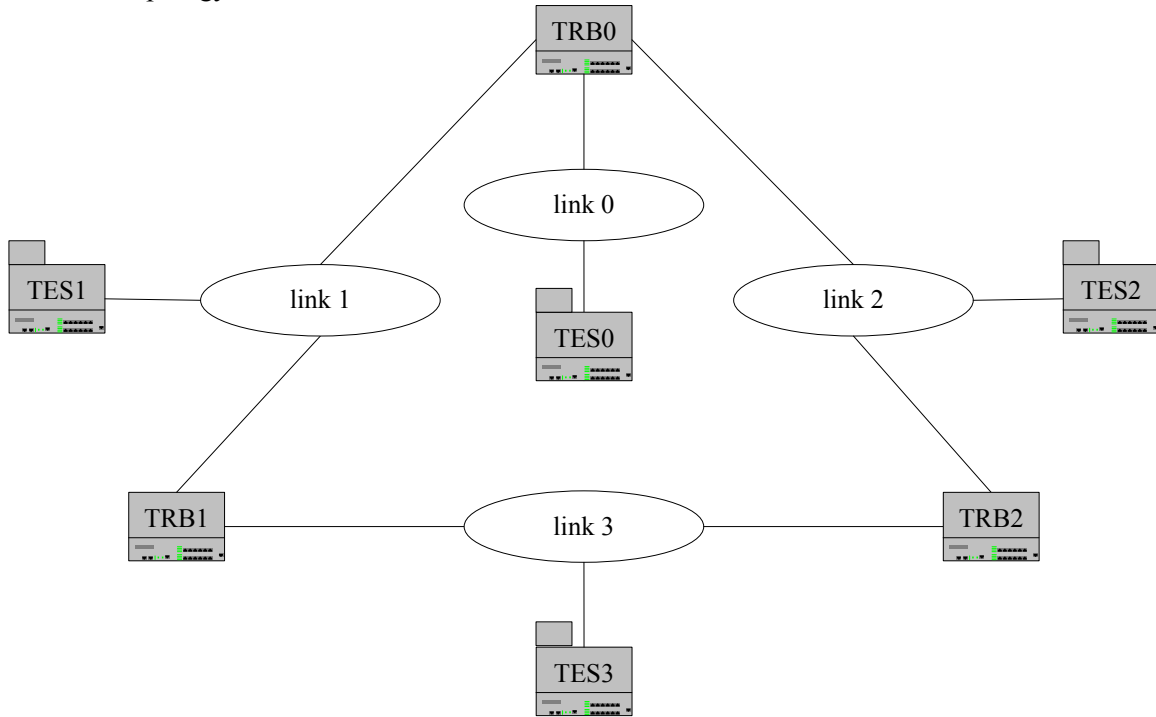
Test TRILL_Interop.4.3: VLAN Egress Decapsulation.....	55
Test TRILL_Interop.4.4: VLAN Incremental Deployment Functionality	57
Test TRILL_Interop.4.5: Multiple VLANs	58
Test TRILL_Interop.4.6: VLAN Mapping.....	59

Common Test Setup

Tests in this test suite may refer to the common test setup procedure defined in this section.

Common Topology

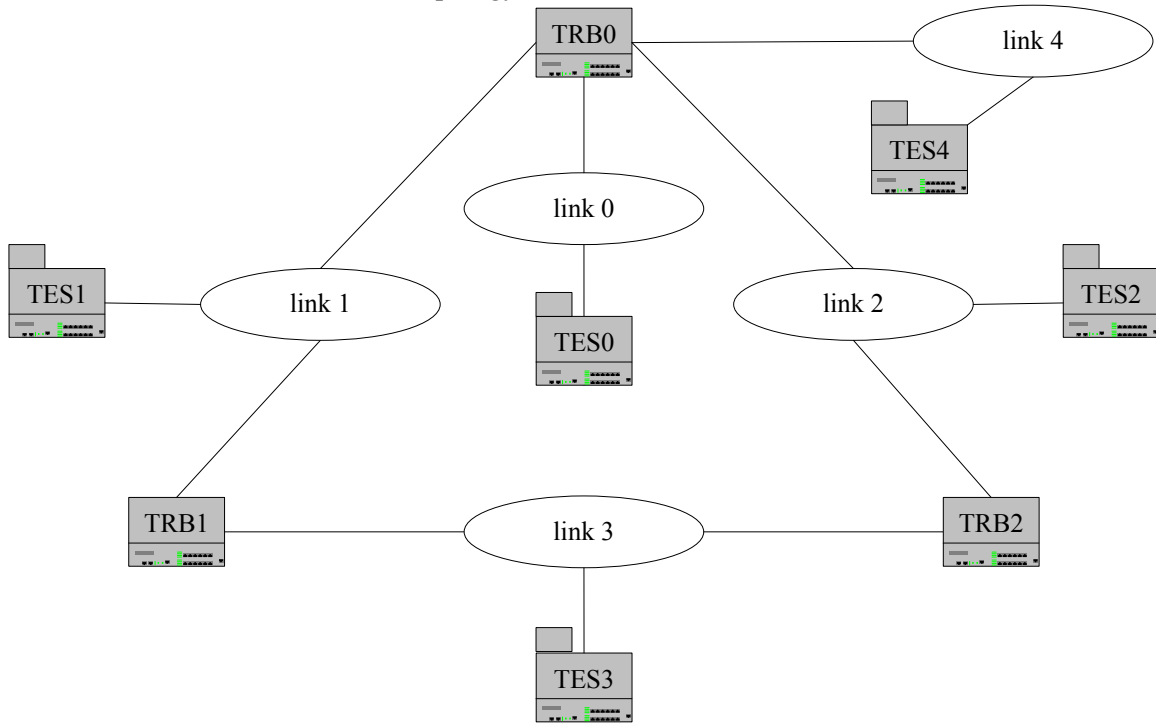
The common topology is:



For ease of transition between test setups TRB1 and TRB2 should also be physically wired to link 0. TRB1's and TRB2's ports to link 0 should be disabled. All transmissions of multicast ethernet data in this test suite refers to non-IPv4 and non-IPv6 multicast ethernet data unless otherwise specified.

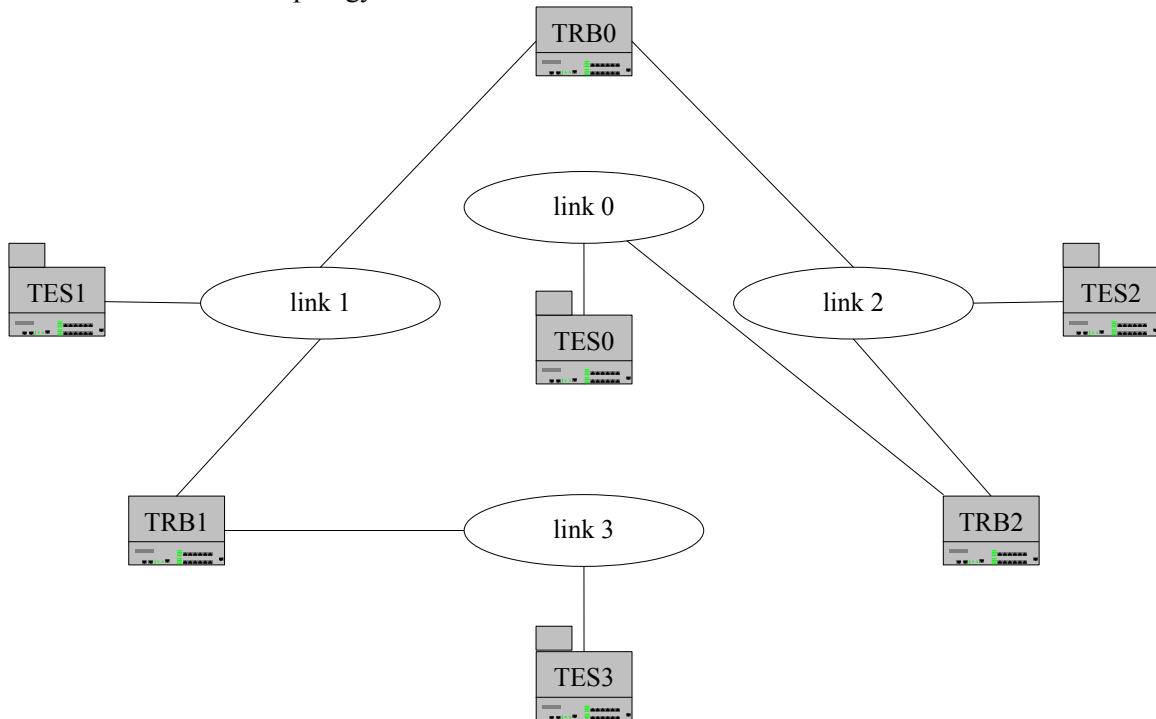
Extended Common Topology

A few tests use the extended common topology:



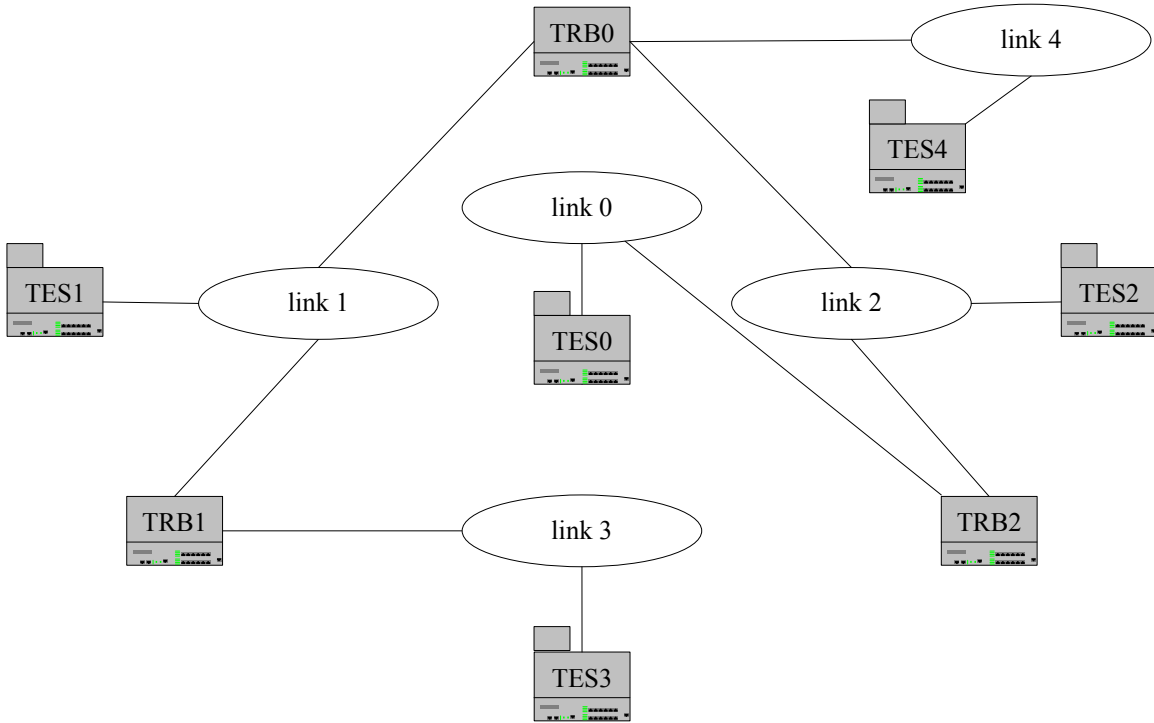
Transit Topology

A few tests use the transit topology:



Extended Transit Topology

A few tests use the extended transit topology:



Common Test Setup

Summary: This basic setup procedure configures the routers with the base TRILL settings for use with the [Common Topology](#), [Extended Common Topology](#), [Transit Topology](#), and [Extended Transit Topology](#). The Common Topology is used unless otherwise specified.

1. Configure TRB0, TRB1 and TRB2 to:
 - a. Have a link cost of 1 on all links.
 - b. Have a distribution tree root priority of 0x8000 (32,768).
 - c. Be part of a single level 1 Layer 2 IS-IS instance with a fixed area-ID zero.
 - d. Have a priority of 0xC0 (192) for keeping its nickname.
 - e. Have a priority of 1 for becoming DRB except where modified below.
 - f. Have no VLANs configured on each link.
 - g. Have a holding time of 60 seconds.
 - h. Have a MTU of 1470 on all links.
 - i. Have a maximum and desired number of 2 distribution trees to compute.
 - j. Not have ESADI configured.
 - k. Have ageing time of 300 seconds.
 - l. Have an inhibition time of 30 seconds.
2. Configure TRB0 to:
 - a. Have a nickname of 0x8000 (32,768).
 - b. Have a priority of 2 on link 1 so that it becomes DRB and appointed forwarder on link 1.
 - c. Have a IS-IS System ID of 1111.1111.0000.
3. Configure TRB1 to:
 - a. Have a nickname of 0x8001 (32,769).
 - b. Have a priority of 2 on link 3 so that it becomes DRB and appointed forwarder on link 3.
 - c. Have a IS-IS System ID of 1111.1111.0001.
4. Configure TRB2 to:
 - a. Have a nickname of 0x8002 (32,770).
 - b. Have a priority of 2 on link 2 so that it becomes DRB and appointed forwarder on link 2.
 - c. Have a IS-IS System ID of 1111.1111.0002.

Common Test Cleanup

Summary: The cleanup procedure causes the devices to remove any TRILL information.

1. Disable TRILL on all devices.
2. Disable Layer 2 IS-IS on all devices.
3. Return all timers and variables to their defaults values.

Group 1: Basic Interoperability

Scope:

The following tests are designed to verify basic TRILL interoperability between RBridges.

Overview:

These tests verify the functionality of a system of RBridges on a fundamental level. These tests include IS-IS functionality, DRB Election, Incremental Deployment with 802.1Q-2005, End Node Learning, End Node Forgetting, ESADI participation, Nickname Collision, Campus Wide MTU determination, IP Snooping, BPDU Handling, Hop Count Handling, and RBridge Loss and Link Loss Handling.

Test TRILL_Interop.1.1: IS-IS

Purpose: To verify that the TRILL IS-IS instance functions correctly.

References:

- [TRILL] – 4.2.3
- [IS-IS]

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part.

Procedure:

1. Enable TRILL on all RBridges.
2. Observe the frames on all links.

Observable Results:

Step 2: All RBridges must become adjacent with one another. TRB0 and TRB1 must list each other as neighbors in their TRILL Hellos on link 1. TRB1 and TRB2 must list each other as neighbors in their TRILL Hellos on link 3. TRB0 and TRB2 must list each other as neighbors in their TRILL Hellos on link 2.

Possible Problems:

- None

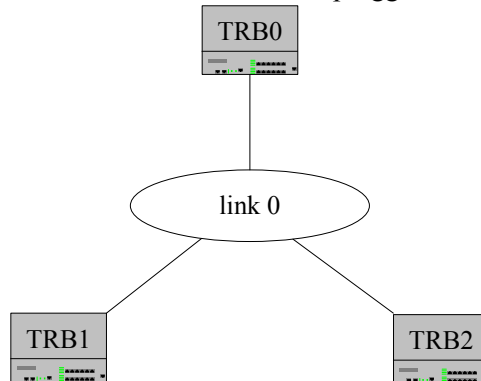
Test TRILL_Interop.1.2: Designated RBridge Election

Purpose: To verify that the Designated RBridge Election Process is properly performed.

References:

- [TRILL] – 4.2.4

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part. TRB0 is unplugged from link 1 and 2. TRB1 is unplugged from link 1 and 3. TRB2 is unplugged from link 2 and 3. TRB1 and TRB2 are plugged into link 0.



Procedure:

Part A: Higher Priority Designated RBridge Election

1. Configure TRB0 with a priority of 3 for becoming DRB on link 0.
2. Configure TRB1 with a priority of 2 for becoming DRB on link 0.
3. Configure TRB2 with a priority of 1 for becoming DRB on link 0.
4. Enable TRILL on all RBridges.
5. Observe the frames on all links.

Part B: Lower Priority Designated RBridge Election

6. Configure TRB0 with a priority of 2 for becoming DRB on link 0.
7. Configure TRB1 with a priority of 3 for becoming DRB on link 0.
8. Configure TRB2 with a priority of 1 for becoming DRB on link 0.
9. Enable TRILL on all RBridges.
10. Observe the frames on all links.

Part C: Equal Priority, Higher MAC Address Designated RBridge Election

11. Configure TRB0 with a priority of 2 for becoming DRB on link 0.
12. Configure TRB1 with a priority of 2 for becoming DRB on link 0.
13. Configure TRB2 with a priority of 1 for becoming DRB on link 0.
14. The MAC address values for each device on link 0 are such that TRB2 > TRB0 > TRB1.
15. Enable TRILL on all RBridges.
16. Observe the frames on all links.

Part D: Equal Priority Lower MAC Address Designated RBridge Election

17. Configure TRB0 with a priority of 2 for becoming DRB on link 0.
18. Configure TRB1 with a priority of 2 for becoming DRB on link 0.
19. Configure TRB2 with a priority of 1 for becoming DRB on link 0.

*University of New Hampshire
InterOperability Laboratory*

20. The MAC address values for each device on link 0 are such that $TRB1 > TRB0 > TRB2$.
21. Enable TRILL on all RBridges.
22. Observe the frames on all links.

Observable Results:

- *Part A*
Step 5: TRB0 must become the DRB on link 0 after holding time (60 seconds). TRB0 must begin issuing CSNPs.
- *Part B*
Step 10: TRB1 must become the DRB on link 0 after holding time (60 seconds). TRB1 must begin issuing CSNPs.
- *Part C*
Step 16: TRB0 must become the DRB on link 0 after holding time (60 seconds). TRB0 must begin issuing CSNPs.
- *Part D*
Step 22: TRB1 must become the DRB on link 0 after holding time (60 seconds). TRB1 must begin issuing CSNPs.

Possible Problems:

- For parts C or D devices may not allow the MAC address to be configured. If one device does not allow this then that device shall be TRB2 in this test. If more than one device does not allow the MAC address to be set and the MAC address inequalities do not hold then this test cannot be performed.

Test TRILL_Interop.1.3: Incremental Deployment Functionality

Purpose: To verify that the TRILL functions when incrementally deployed with 802.1Q.

References:

- [TRILL] – 2.6.2
- [802.1Q-2005]

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part.

Procedure:

1. TRB0 is disabled on link 2.
2. The latest version of STP supported is enabled on TRB1. TRB1 is no longer an RBridge.
3. Enable TRILL on all RBridges.
4. TES2 transmits unicast ethernet data to TES0 on link 2.
5. Observe the frames on all links.
6. TES0 transmits unicast ethernet data to TES2 on link 0.
7. Observe the frames on all links.

Observable Results:

Step 5: The unknown ethernet data must travel to all links in the 802.1Q cloud.

Step 7: The known ethernet data must travel from TES0 to TES3 thru the 802.1Q cloud.

Possible Problems:

- TRB1 may not support 802.1Q-2005. If this is the case then a switch that does should be substituted.

Test TRILL_Interop.1.4: End Node Address Learning and Forgetting

Purpose: To verify an RBridge learns and forgets end node address correctly.

References:

- [TRILL] – 4.8.1, 4.8.2, 4.2.5, 4.2.5.2

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part.

Procedure:

Part A: End Node Address Learning

1. Enable TRILL on all RBridges.
2. TES3 transmits unicast ethernet data to TES0 on link 3.
3. TES0 transmits unicast ethernet data to TES3 on link 0.
4. Observe the frames on all links.

Part B: End Node Address Forgetting

5. Enable TRILL on all RBridges.
6. TES3 transmits unicast ethernet data to TES0 on link 3.
7. Observe the frames on all links.
8. Wait the aging time of TES0 (300 Seconds).
9. TES0 transmits unicast ethernet data to TES3 on link 0.
10. Observe the frames on all links.

Part C: ESADI Address Learning

11. Configure ESADI on TRB0 and TRB1 so that TRB0 learns of TES3's MAC address.
12. Enable TRILL on all RBridges.
13. TES0 transmits unicast ethernet data to TES3 on link 0.
14. Observe the frames on all links.

Part D: ESADI Address Forgetting

15. Configure ESADI on TRB0 and TRB1 so that TRB0 learns of TES3's MAC address.
16. Enable TRILL on all RBridges.
17. Observe the frames on all links.
18. Unplug TRB0.
19. Reset TRILL and IS-IS on TRB0. Remove any ESADI configuration on TRB1.
20. Wait the timeout value specified in TRB0's ESADI information for TES3.
21. Plug in TRB0.
22. TES0 transmits unicast ethernet data to TES3 on link 0.
23. Observe the frames on all links.

Part E: End Node Management Configuration

24. Configure TRB0 to know TES3's location.
25. Enable TRILL on all RBridges.
26. TES0 transmits unicast ethernet data to TES3 on link 0.
27. Observe the frames on all links.

Observable Results:

- *Part A*

*University of New Hampshire
InterOperability Laboratory*

Step 4: TRB0 must add TES3 to its learned MAC address table. The data from TES0 to TES3 must be sent as a known unicast TRILL frame with the M bit clear.

- *Part B*

Step 7: TRB0 must add TES3 to its learned MAC address table.

Step 10: TRB0 must remove TES3 from its learned MAC address table. The data from TES0 to TES3 must be sent as an unknown unicast TRILL frame with the M bit set.

- *Part C*

Step 14: TRB0 must add TES3 to its learned MAC address table through ESADI. The data from TES0 to TES3 must be sent as a known unicast TRILL frame with the M bit clear

- *Part D*

Step 17: TRB0 must add TES3 to its learned MAC address table.

Step 23: TRB0 must remove TES3 from its learned MAC address table. The data from TES0 to TES3 must be sent as an unknown unicast TRILL frame with the M bit set.

- *Part E*

Step 27: TRB0 must add TES3 to its learned MAC address table through management configuration. The data from TES0 to TES3 must be sent as a known unicast TRILL frame with the M bit clear.

Possible Problems:

- TRB0 or TRB1 might not support any, or the same, ESADI protocol or Layer 2 Registration Protocol. If only TRB1 does not support the same ESADI or Layer 2 Registration Protocol then TRB1 can be swapped with TRB2 for this test. Also, a device being unplugged from a network must not cause any TRILL or IS-IS frames to be sent on the link the device was unplugged from. Unplugging a device must also not cause any other devices to lose link.

Test TRILL_Interop.1.5: Nickname Collision

Purpose: To verify that RBridges properly deal with a nickname collision.

References:

- [TRILL] – 3.7.3

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part.

Part A: Higher Priority Nickname Collision

1. Configure all RBridges with a nickname of 0x8000 (32,768).
2. Configure TRB0 with a priority of 0xC1 (193) for keeping its nickname.
3. Configure TRB1 with a priority of 0xC0 (192) for keeping its nickname.
4. Configure TRB2 with a priority of 0xBF (191) for keeping its nickname.
5. Enable TRILL on all RBridges.
6. Observe the frames on all links.

Part B: Middle Priority Nickname Collision

7. Configure all RBridges with a nickname of 0x8000 (32,768).
8. Configure TRB0 with a priority of 0xC0 (192) for keeping its nickname.
9. Configure TRB1 with a priority of 0xC1 (193) for keeping its nickname.
10. Configure TRB2 with a priority of 0xBF (191) for keeping its nickname.
11. Enable TRILL on all RBridges.
12. Observe the frames on all links.

Part C: Lower Priority Nickname Collision

13. Configure all RBridges with a nickname of 0x8000 (32,768).
14. Configure TRB0 with a priority of 0xBC (191) for keeping its nickname.
15. Configure TRB1 with a priority of 0xC1 (193) for keeping its nickname.
16. Configure TRB2 with a priority of 0xC0 (192) for keeping its nickname.
17. Enable TRILL on all RBridges.
18. Observe the frames on all links.

Part D: Equal Priority Higher IS-IS System ID Nickname Collision

19. Configure all RBridges with a nickname of 0x8000 (32,768).
20. Configure TRB0 with a priority of 0xC0 (192) for keeping its nickname.
21. Configure TRB0 with an IS-IS System ID of 1111.1111.0010.
22. Configure TRB1 with a priority of 0xC0 (192) for keeping its nickname.
23. Configure TRB1 with an IS-IS System ID of 1111.1111.0001.
24. Configure TRB2 with a priority of 0xBF (191) for keeping its nickname.
25. Configure TRB2 with an IS-IS System ID of 1111.1111.0012.
26. Enable TRILL on all RBridges.
27. Observe the frames on all links.

Part E: Equal Priority Lower IS-IS System ID Nickname Collision

28. Configure all RBridges with a nickname of 0x8000 (32,768).
29. Configure TRB0 with a priority of 0xC0 (192) for keeping its nickname.
30. Configure TRB0 with an IS-IS System ID of 1111.1111.0000.
31. Configure TRB1 with a priority of 0xC0 (192) for keeping its nickname.
32. Configure TRB1 with an IS-IS System ID of 1111.1111.0011.

*University of New Hampshire
InterOperability Laboratory*

33. Configure TRB2 with a priority of 0xBF (191) for keeping its nickname.
34. Configure TRB2 with an IS-IS System ID of 1111.1111.0012.
35. Enable TRILL on all RBridges.
36. Observe the frames on all links.

Observable Results:

- *Part A*
Step 6: TRB0 must keep the nickname 0x8000 (32,768) in its TRILL Hello frames. The other RBridges must choose new nicknames.
- *Part B*
Step 12: TRB1 must keep the nickname 0x8000 (32,768) in its TRILL Hello frames. The other RBridges must choose new nicknames.
- *Part C*
Step 18: TRB1 must keep the nickname 0x8000 (32,768) in its TRILL Hello frames. The other RBridges must choose new nicknames.
- *Part D*
Step 27: TRB0 must keep the nickname 0x8000 (32,768) in its TRILL Hello frames. The other RBridges must choose new nicknames.
- *Part E*
Step 36: TRB1 must keep the nickname 0x8000 (32,768) in its TRILL Hello frames. The other RBridges must choose new nicknames.

Possible Problems:

- None

Test TRILL_Interop.1.6: Campus Wide MTU

Purpose: To verify that the RBridges choose the appropriate campus wide MTU.

References:

- [TRILL] – 4.3.1

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part.

Procedure:

1. Configure TRB1 and TRB2 to have an MTU of 1280 on link 3.
2. Enable TRILL on all RBridges.
3. Observe the frames on all links.

Observable Results:

Step 3: The Campus Wide MTU Sz value must be 1280 on all RBridges. The originatingLSPBufferSize in each RBridge's LSP must be set to 1280.

Possible Problems:

- None

Test TRILL_Interop.1.7: IP Snooping

Purpose: To verify that RBridges perform IP Snooping for multicast data.

References:

- [TRILL] – 4.7
- [IPv4]
- [IGMPv3]
- [IPv6]
- [MLDv2]
- [SNOOP]

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part.

Procedure:

Part A: IPv4 Multicast Data Not Joined

1. Enable TRILL on all RBridges.
2. TES3 sends multicast data for IPv4 multicast group 224.0.6.130 on link 3.
3. Observe the frames on all links.

Part B: IPv4 Multicast Data Joined

4. Enable TRILL on all RBridges.
5. TES0 sends an IGMPv3 to exclude nothing for multicast group 224.0.6.130 on link 0.
6. TES3 sends multicast data for IPv4 multicast group 224.0.6.130 on link 3.
7. Observe the frames on all links.

Part C: IPv4 Multicast Data Left

8. Enable TRILL on all RBridges.
9. TES0 sends an IGMPv3 to exclude nothing for multicast group 224.0.6.130 on link 0.
10. TES3 sends multicast data for IPv4 multicast group 224.0.6.130 on link 3.
11. Observe the frames on all links.
12. TES0 sends an IGMPv3 to include nothing for multicast group 224.0.6.130 on link 0.
13. TES3 sends multicast data for IPv4 multicast group 224.0.6.130 on link 3.
14. Observe the frames on all links.

Observable Results:

- *Part A*
Step 3: The multicast data must not reach TES0.
- *Part B*
Step 7: The multicast data must reach TES0.
- *Part C*
Step 11: The multicast data must reach TES0.
Step 14: The multicast data must not reach TES0.

Possible Problems:

*University of New Hampshire
InterOperability Laboratory*

- An RBridge may not support IP Snooping. IPv6 and MLD may be used instead of IPv4 and IGMP.

Test TRILL_Interop.1.8: BPDU Handing

Purpose: To verify that RBridges receive and transmit BPDUs correctly.

References:

- [TRILL] – 4.9.3
- [802.1Q-2005]
- [IS-IS]

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part.

Procedure:

Part A: Inhibits on Root Bridge Change

1. The latest version of STP supported is enabled on TRB1 and TRB2. TRB1 and TRB2 are no longer RBridges.
2. Configure TRB1 with a Bridge Priority of 0x8000 (32,768).
3. Configure TRB2 with a Bridge Priority of 0x9000 (36,864).
4. Enable TRILL on TRB0.
5. TES0 transmits broadcast ethernet data on link 0.
6. Observe the frames on all links.
7. Configure TRB1 with a Bridge Priority of 0xA000 (40,960).
8. TES0 transmits broadcast ethernet data on link 0.
9. Observe the frames on all links.
10. Wait inhibition time (30 seconds).
11. TES0 transmits broadcast ethernet data on link 0.
12. Observe the frames on all links.

Part B: Sends Topology Change BPDU on change of Appointed Forwarder

13. The latest version of STP supported is enabled on TRB1. TRB1 is no longer an RBridge.
14. Enabled TRB1 and TRB2 on link 0.
15. Configure TRB0 to have a priority of 2 on link 0 so that it becomes DRB and appointed forwarder on link 0.
16. Configure TRB2 to have a priority of 1 on link 0.
17. Enable TRILL on all RBridges.
18. Observe the frames on all links.
19. Configure TRB2 to have a priority of 3 on link 0 so that it becomes DRB and appointed forwarder on link 0.
20. Observe the frames on all links.

Observable Results:

- *Part A*
 - Step 6:** The broadcast ethernet data must be forwarded by TRB0 onto link 1 and then TRB0 must encapsulate the broadcast ethernet data with ethernet and TRILL headers and then transmit the encapsulated frame on link 1. TRB1 must forward both the encapsulated and un-encapsulated frame onto link 3. TRB2 must forward both of these onto link 2.
 - Step 9:** TRB0 must inhibit itself. The broadcast ethernet data or an encapsulated frame must

*University of New Hampshire
InterOperability Laboratory*

not reach links 1, 2, or 3.

Step 12: The broadcast ethernet data must be forwarded by TRB0 onto link 1 and then TRB0 must encapsulate the broadcast ethernet data with ethernet and TRILL headers and then transmit the encapsulated frame on link 1. TRB1 must forward both the encapsulated and un-encapsulated frame onto link 3. TRB2 must forward both of these onto link 2.

- *Part B*

Step 18: TRB0 must become the DRB on link 0 after holding time (60 seconds). TRB0 will begin issuing CSNPs.

Step 20: TRB2 must become the DRB on link 0 after holding time (60 seconds). TRB2 will begin issuing CSNPs. TRB0 will cease issuing CSNPs. TRB0 must transmit a Topology Change BPDU on link 0.

Possible Problems:

- TRB1 and/or TRB2 may not support 802.1Q-2005. If this is the case then other switches that do should be substituted.

Test TRILL_Interop.1.9: Hop Count Handling

Purpose: To verify that an RBridge properly forwards TRILL data through the campus decrementing the hop count along the way.

References:

- [TRILL] – 4.6.2.4, 4.6.2.5

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part. The [Transit Topology](#) is used.

Procedure:

1. TRB2 is disabled on link 3 and enabled on link 0. TRB0 is disabled on link 0.
2. Enable TRILL on all RBridges.
3. TES3 transmits unicast ethernet data to TES0 on link 3.
4. Observe the frames on all links.
5. TES0 transmits unicast ethernet data to TES3 on link 0.
6. Observe the frames on all links.

Observable Results:

Step 4: TRB2 must learn TES3's MAC address. TRB0 must decrement the TRILL hop count of the encapsulated frames from link 1 sent on link 2.

Step 6: TRB0 must decrement the TRILL hop count of the encapsulated frames from link 2 sent on link 1.

Possible Problems:

- None

Test TRILL_Interop.1.10: RBridge Loss and Link Loss Handling

Purpose: To verify that the RBridges recover from the loss of an RBridge in the campus or the loss of a link on a RBridge in the campus.

References:

- [TRILL] – 4.2.4, 4.5.5, 4.6

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part.

Procedure:

Part A: Unicast Pathway RBridge Loss

1. Configure TRB0's link 2 port to have a link cost of 2.
2. Enable TRILL on all RBridges.
3. TES3 transmits unicast ethernet data to TES0 on link 3.
4. TES0 transmits unicast ethernet data to TES3 on link 0.
5. Observe the frames on all links.
6. Unplug TRB1 from all links.
7. Wait holding time (60 seconds).
8. TES0 transmits unicast ethernet data to TES3 on link 0.
9. Observe the frames on all links.

Part B: Unicast Pathway Link Loss

10. Configure TRB2's link 3 port to have a link cost of 2.
11. Enable TRILL on all RBridges.
12. TES3 transmits unicast ethernet data to TES0 on link 3.
13. TES0 transmits unicast ethernet data to TES3 on link 0.
14. Observe the frames on all links.
15. Unplug TRB1 from link 3.
16. TES0 transmits unicast ethernet data to TES3 on link 0.
17. Observe the frames on all links.

Part C: Distribution Tree Root Loss

18. Configure TRB1's distribution tree root priority to 0x8002 (32,770).
19. Configure TRB2's distribution tree root priority to 0x8001 (32,769).
20. Enable TRILL on all RBridges.
21. TES0 transmits broadcast ethernet data on link 0.
22. Observe the frames on all links.
23. Unplug TRB1 from all links.
24. Wait holding time (60 seconds).
25. TES0 transmits broadcast ethernet data on link 0.
26. Observe the frames on all links.

Part D: Distribution Tree Root Link Loss

27. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
28. Enable TRILL on all RBridges.
29. TES0 transmits broadcast ethernet data on link 0.
30. Observe the frames on all links.
31. Unplug TRB1 from link 3.

32. TES0 transmits broadcast ethernet data on link 0.
33. Observe the frames on all links.

Part E: Distribution Tree RBridge Loss

34. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
35. Enable TRILL on all RBridges.
36. TES0 transmits broadcast ethernet data on link 0.
37. Observe the frames on all links.
38. Unplug TRB2 from all links.
39. Wait holding time (60 seconds).
40. TES0 transmits broadcast ethernet data on link 0.
41. Observe the frames on all links.

Part F: Distribution Tree RBridge Link Loss

42. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
43. Enable TRILL on all RBridges.
44. TES0 transmits broadcast ethernet data on link 0.
45. Observe the frames on all links.
46. Unplug TRB2 from link 3.
47. TES0 transmits broadcast ethernet data on link 0.
48. Observe the frames on all links.

Observable Results:

- *Part A*
 - Step 5:** TRB0 must ingress the ethernet data from TES0 to TES3 and TRB1 must egress the data. The data must not be flooded to every link.
 - Step 9:** TRB0 must ingress the ethernet data and TRB2 must egress the data.
- *Part B*
 - Step 15:** TRB0 must ingress the ethernet data from TES0 to TES3 and TRB1 must egress the data. The data must not be flooded to every link.
 - Step 17:** TRB1 must notify TRB0 of the link failure through transmission of an updated IS-IS LSP. TRB0 must ingress the ethernet data and TRB2 must egress the data.
- *Part C*
 - Step 22:** TRB0 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0 and 1. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.
 - Step 26:** TRB0 must ingress the ethernet data. TRB2 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0 and 1. TRB2 must be the appointed forwarder on link 2 and 3.
- *Part D*
 - Step 30:** TRB0 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0 and 1. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.
 - Step 33:** TRB1 must notify TRB0 of the link failure through transmission of an updated IS-IS LSP. TRB0 must ingress the ethernet data. TRB2 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0 and 1. TRB2 must be the appointed forwarder on link 2 and 3.
- *Part E*
 - Step 37:** TRB0 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0 and 1. TRB1 must be the appointed

*University of New Hampshire
InterOperability Laboratory*

forwarder on link 3. TRB2 must be the appointed forwarder on link 2.

Step 41: After holding time (60 seconds) TRB0 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, 1 and 2. TRB1 must be the appointed forwarder on link 3.

- *Part F*

Step 45: TRB0 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0 and 1. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.

Step 48: TRB2 must notify TRB0 of the link failure through transmission of an updated IS-IS LSP. TRB0 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0 and 1. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.

Possible Problems:

- A device being unplugged from a network must not cause any TRILL or IS-IS frames to be sent on the link the device was unplugged from. Unplugging a device must also not cause any other devices to lose link.

Group 2: Forwarding

Scope:

The following tests are designed to verify TRILL interoperability between RBridges in forwarding.

Overview:

These tests verify that RBridges can handle known unicast, unknown unicast, multicast, and broadcast data in the ingress, transit, and egress situations. All the forwarding tested in this section is based on no VLAN tagging.

Test TRILL_Interop.2.1: Ingress Encapsulation

Purpose: To verify that an RBridge properly encapsulates data on ingress into the campus.

References:

- [TRILL] – 2.4.1, 4.6.1.1

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part. The [Extended Common Topology](#) is used.

Procedure:

Part A: Known Unicast Ingress Encapsulation

1. Configure TRB0's link 2 port to have a link cost of 2.
2. Enable TRILL on all RBridges.
3. TES3 transmits unicast ethernet data to TES0 on link 3.
4. TES0 transmits unicast ethernet data to TES3 on link 0.
5. Observe the frames on all links.

Part B: Unknown Unicast Ingress Encapsulation

6. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
7. Enable TRILL on all RBridges.
8. TES0 transmits unicast ethernet data to TES3 on link 0.
9. Observe the frames on all links.

Part C: Broadcast Ingress Encapsulation

10. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
11. Enable TRILL on all RBridges.
12. TES0 transmits broadcast ethernet data on link 0.
13. Observe the frames on all links.

Part D: Multicast Ingress Encapsulation

14. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
15. Enable TRILL on all RBridges.
16. TES0 transmits multicast ethernet data on link 0.
17. Observe the frames on all links.

Observable Results:

- *Part A*
Step 5: TRB0 must ingress the ethernet data from TES0 to TES3 and TRB1 must egress the data. The data must not be flooded to every link.
- *Part B*
Step 9: TRB0 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.
- *Part C*
Step 13: TRB0 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.
- *Part D*

*University of New Hampshire
InterOperability Laboratory*

Step 17: TRB0 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.

Possible Problems:

- None

Test TRILL_Interop.2.2: Transit Forwarding

Purpose: To verify that an RBridge properly forwards TRILL data through the campus.

References:

- [TRILL] – 4.6.2, 4.6.2.4, 4.6.2.5

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part. The [Extended Transit Topology](#) is used.

Procedure:

Part A: Known Unicast Transit Forwarding

1. TRB2 is disabled on link 3 and enabled on link 0. TRB0 is disabled on link 0.
2. Enable TRILL on all RBridges.
3. TES3 transmits unicast ethernet data to TES0 on link 3.
4. TES0 transmits unicast ethernet data to TES3 on link 0.
5. Observe the frames on all links.

Part B: Unknown Unicast Transit Forwarding

6. TRB2 is disabled on link 3 and enabled on link 0. TRB0 is disabled on link 0.
7. Enable TRILL on all RBridges.
8. TES0 transmits unicast ethernet data to TES3 on link 0.
9. Observe the frames on all links.

Part C: Broadcast Transit Forwarding

10. TRB2 is disabled on link 3 and enabled on link 0. TRB0 is disabled on link 0.
11. Enable TRILL on all RBridges.
12. TES0 transmits broadcast ethernet data on link 0.
13. Observe the frames on all links.

Part D: Multicast Transit Forwarding

14. TRB2 is disabled on link 3 and enabled on link 0. TRB0 is disabled on link 0.
15. Enable TRILL on all RBridges.
16. TES0 transmits multicast ethernet data on link 0.
17. Observe the frames on all links.

Part E: ESADI Transit Forwarding

18. TRB2 is disabled on link 3 and enabled on link 0. TRB0 is disabled on link 0.
19. Configure ESADI on TRB2 and TRB1.
20. Enable TRILL on all RBridges.
21. Observe the frames on all links.

Observable Results:

- *Part A*
Step 5: TRB2 must ingress the ethernet data from TES0 to TES3 and TRB1 must egress the data. The data must not be flooded to every link.
- *Part B*
Step 9: TRB2 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.

*University of New Hampshire
InterOperability Laboratory*

- *Part C*
Step 13: TRB2 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.
- *Part D*
Step 17: TRB2 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.
- *Part E*
Step 21: TRB1 and TRB2 must ingress TRILL encapsulated ESADI frames into the network. TRB1 must be the distribution tree root for these frames. TRB0 must be the appointed forwarder on link 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.

Possible Problems:

- TRB1 or TRB2 might not support any, or the same, ESADI protocols.

Test TRILL_Interop.2.3: Egress Decapsulation

Purpose: To verify that an RBridge properly decapsulates data on egress out of the campus.

References:

- [TRILL] – 4.6.2, 4.6.2.4, 4.6.2.5

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part. The [Extended Common Topology](#) is used.

Procedure:

Part A: Known Unicast Egress Decapsulation

1. Configure TRB2's link 2 port to have a link cost of 2.
2. Enable TRILL on all RBridges.
3. TES0 transmits unicast ethernet data to TES3 on link 3.
4. TES3 transmits unicast ethernet data to TES0 on link 0.
5. Observe the frames on all links.

Part B: Unknown Unicast Egress Decapsulation

6. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
7. Enable TRILL on all RBridges.
8. TES0 transmits unicast ethernet data to TES3 on link 0.
9. Observe the frames on all links.

Part C: Broadcast Egress Decapsulation

10. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
11. Enable TRILL on all RBridges.
12. TES0 transmits broadcast ethernet data on link 0.
13. Observe the frames on all links.

Part D: Multicast Egress Decapsulation

14. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
15. Enable TRILL on all RBridges.
16. TES0 transmits multicast ethernet data on link 0.
17. Observe the frames on all links.

Observable Results:

- *Part A*
Step 5: TRB1 must ingress the ethernet data from TES0 to TES3 and TRB0 must egress the data. The data must not be flooded to every link.
- *Part B*
Step 9: TRB1 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.
- *Part C*
Step 13: TRB1 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.
- *Part D*

*University of New Hampshire
InterOperability Laboratory*

Step 17: TRB1 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.

Possible Problems:

- None

Group 3: Distribution Trees

Scope:

The following tests are designed to verify RBridges properly handles distributions trees.

Overview:

These tests verify that RBridges calculated distribution trees with the shortest path first algorithm, choose the correct RBridges for roots in distribution trees, correctly choose the campus wide K value for the number of trees to calculate, use the set of S roots the highest priority RBridge wishes to use as roots, use the correct tie breaker in distribution tree calculations, and prune links from the distribution tree that do not have any receivers.

Test TRILL_Interop.3.1: Shortest Path First Calculation

Purpose: To verify that TRILL distribution trees are properly calculated with the shortest path first algorithm.

References:

- [TRILL] – 4.5.1

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part.

Procedure:

Part A: Balanced Tree Calculation, Tie Breaking

1. Configure TRB0 to have a distribution tree root priority of 0x8001 (32,769).
2. Enable TRILL on all RBridges.
3. TES0 transmits multicast ethernet data on link 0.
4. Observe the frames on all links.

Part B: Unbalanced Tree Calculation, Direct Link

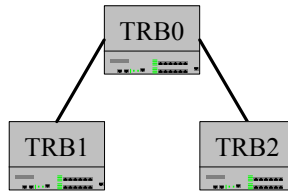
5. Configure TRB0's link 2 to have a link cost of 10.
6. Configure TRB0 to have a distribution tree root priority of 0x8001 (32,769).
7. Enable TRILL on all RBridges.
8. TES0 transmits multicast ethernet data on link 0.
9. Observe the frames on all links.
10. Configure TRB0's link 1 to have a link cost of 20.
11. TES0 transmits multicast ethernet data on link 0.
12. Observe the frames on all links.

Part C: Unbalanced Tree Calculation, Indirect Link

13. Configure TRB1's link 3 to have a link cost of 10.
14. Configure TRB1 to have a distribution tree root priority of 0x8001 (32,769).
15. Enable TRILL on all RBridges.
16. TES0 transmits multicast ethernet data on link 0.
17. Observe the frames on all links.
18. Configure TRB2's link 3 to have a link cost of 20.
19. TES0 transmits multicast ethernet data on link 0.
20. Observe the frames on all links.

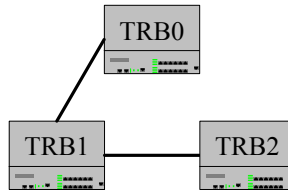
Observable Results:

- *Part A*
Step 4: TRB0 must ingress the ethernet data. TRB0 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, and 1. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2. The calculated distribution tree must be:

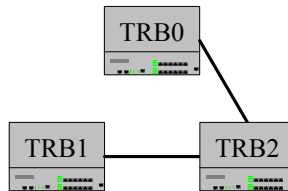


• *Part B*

Step 8: TRB0 must ingress the ethernet data. TRB0 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, and 1. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2. The calculated distribution tree must be:

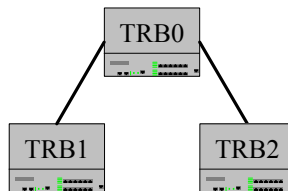


Step 12: TRB0 must ingress the ethernet data. TRB0 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, and 1. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2. The calculated distribution tree must be:

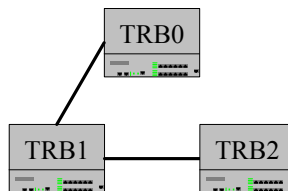


• *Part C*

Step 17: TRB0 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, and 1. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2. The calculated distribution tree must be:



Step 20: TRB0 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, and 1. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2. The calculated distribution tree must be:



Possible Problems:

- None

Test TRILL_Interop.3.2: Root Choice

Purpose: To verify that RBridges correctly choose distribution tree roots.

References:

- [TRILL] – 4.5

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part.

Procedure:

Part A: Higher Priority Root Choice

1. Configure TRB0 with a distribution tree root priority of 0x8002 (32,770).
2. Configure TRB1 with a distribution tree root priority of 0x8001 (32,769).
3. Configure TRB2 with a distribution tree root priority of 0x8000 (32,768).
4. Enable TRILL on all RBridges.
5. TES0 transmits multicast ethernet data on link 0.
6. Observe the frames on all links.

Part B: Middle Priority Root Choice

7. Configure TRB0 with a distribution tree root priority of 0x8001 (32,769).
8. Configure TRB1 with a distribution tree root priority of 0x8002 (32,770).
9. Configure TRB2 with a distribution tree root priority of 0x8000 (32,768).
10. Enable TRILL on all RBridges.
11. TES0 transmits multicast ethernet data on link 0.
12. Observe the frames on all links.

Part C: Lower Priority Root Choice

13. Configure TRB0 with a distribution tree root priority of 0x8000 (32,768).
14. Configure TRB1 with a distribution tree root priority of 0x8002 (32,770).
15. Configure TRB2 with a distribution tree root priority of 0x8001 (32,769).
16. Enable TRILL on all RBridges.
17. TES0 transmits multicast ethernet data on link 0.
18. Observe the frames on all links.

Part D: Equal Priority Higher IS-IS System Id Root Choice

19. Configure TRB0 with a distribution tree root priority of 0x8001 (32,769).
20. Configure TRB1 with a distribution tree root priority of 0x8001 (32,769).
21. Configure TRB2 with a distribution tree root priority of 0x8000 (32,768).
22. Configure TRB0 with a IS-IS System ID of 1111.1111.0010
23. Configure TRB1 with a IS-IS System ID of 1111.1111.0001
24. Configure TRB2 with a IS-IS System ID of 1111.1111.0022
25. Enable TRILL on all RBridges.
26. TES0 transmits multicast ethernet data on link 0.
27. Observe the frames on all links.

Part E: Equal Priority Lower IS-IS System Id Root Choice

28. Configure TRB0 with a distribution tree root priority of 0x8001 (32,769).
29. Configure TRB1 with a distribution tree root priority of 0x8001 (32,769).
30. Configure TRB2 with a distribution tree root priority of 0x8000 (32,768).
31. Configure TRB0 with a IS-IS System ID of 1111.1111.0010

32. Configure TRB1 with a IS-IS System ID of 1111.1111.0021
33. Configure TRB2 with a IS-IS System ID of 1111.1111.0002
34. Enable TRILL on all RBridges.
35. TES0 transmits multicast ethernet data on link 0.
36. Observe the frames on all links.

Part F: Equal Priority Middle IS-IS System Id Root Choice

37. Configure all RBridges with a distribution tree root priority of 0x0000 (32768).
38. Configure TRB0 with a IS-IS System ID of 1111.1111.0001
39. Configure TRB1 with a IS-IS System ID of 1111.1111.0000
40. Configure TRB2 with a IS-IS System ID of 1111.1111.0002
41. Enable TRILL on all RBridges.
42. TES0 transmits multicast ethernet data on link 0.
43. Observe the frames on all links.

Part G: Higher Priority Root Choice, Max Bound

44. Configure TRB0 with a distribution tree root priority of 0xFFFF(65,535).
45. Configure TRB1 with a distribution tree root priority of 0x8001 (32,769).
46. Configure TRB2 with a distribution tree root priority of 0x8000 (32,768).
47. Enable TRILL on all RBridges.
48. TES0 transmits multicast ethernet data on link 0.
49. Observe the frames on all links.

Part H: Lower Priority Root Choice, Max Bound

50. Configure TRB0 with a distribution tree root priority of 0x8001 (32,769).
51. Configure TRB1 with a distribution tree root priority of 0xFFFF (65,535).
52. Configure TRB2 with a distribution tree root priority of 0x8000 (32,768).
53. Enable TRILL on all RBridges.
54. TES0 transmits multicast ethernet data on link 0.
55. Observe the frames on all links.

Observable Results:

- *Part A*
Step 6: TRB0 must be the root of the distribution tree calculated. TRB0 must encapsulate the multicast data in an ethernet and TRILL header and TRB0 must be egress RBridge.
- *Part B*
Step 12: TRB1 must be the root of the distribution tree calculated. TRB0 must encapsulate the multicast data in an ethernet and TRILL header and TRB1 must be egress RBridge.
- *Part C*
Step 18: TRB1 must be the root of the distribution tree calculated. TRB0 must encapsulate the multicast data in an ethernet and TRILL header and TRB1 must be egress RBridge.
- *Part D*
Step 27: TRB0 must be the root of the distribution tree calculated. TRB0 must encapsulate the multicast data in an ethernet and TRILL header and TRB0 must be egress RBridge.
- *Part E*
Step 36: TRB1 must be the root of the distribution tree calculated. TRB0 must encapsulate the multicast data in an ethernet and TRILL header and TRB1 must be egress RBridge.
- *Part F*
Step 43: TRB2 must be the root of the distribution tree calculated. TRB0 must encapsulate the multicast data in an ethernet and TRILL header and TRB2 must be egress RBridge.
- *Part G*

*University of New Hampshire
InterOperability Laboratory*

Step 49: TRB0 must be the root of the distribution tree calculated. TRB0 must encapsulate the multicast data in an ethernet and TRILL header and TRB0 must be egress RBridge.

- *Part H*

Step 55: TRB1 must be the root of the distribution tree calculated. TRB0 must encapsulate the multicast data in an ethernet and TRILL header and TRB1 must be egress RBridge.

Possible Problems:

- None

Test TRILL_Interop.3.3: Number of Trees to Calculate

Purpose: To verify that RBridges correctly choose the number of trees to calculate.

References:

- [TRILL] – 4.5

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part.

Procedure:

Part A: Highest Priority RBridge Desired K

1. Configure all RBridges to support a maximum of 3 computed distribution trees.
2. Configure TRB0 and TRB2 to desire 3 computed distribution trees.
3. Configure TRB1 to desire 2 computed distribution trees.
4. Enable TRILL on all RBridges.
5. Send multicast ethernet data from TES0 to TES3 until load balancing occurs.
6. Observe the frames on all links.

Part B: Non-Highest Priority RBridge Maximal K

7. Configure TRB0 and TRB1 to support a maximum of 3 computed distribution trees.
8. Configure TRB2 to support a maximum of 2 computed distribution trees.
9. Configure TRB0 and TRB1 to desire 3 computed distribution trees.
10. Configure TRB2 to desire 2 computed distribution trees.
11. Enable TRILL on all RBridges.
12. Send multicast ethernet data from TES0 to TES3 until load balancing occurs.
13. Observe the frames on all links.

Observable Results:

- *Part A*
Step 6: Once TRB0 begins to load balance the multicast ethernet data must be sent over a maximum of two distribution trees.
- *Part B*
Step 13: Once TRB0 begins to load balance the multicast ethernet data must be sent over a maximum of two distribution trees.

Possible Problems:

- TRB0 may not support load balancing and may only use one distribution tree. This test may be omitted by such devices. If multiple end stations sending multicast ethernet data to link 3 need to be introduced onto link 0 for load balancing to occur the configuration can be altered to account for this.

Test TRILL_Interop.3.4: Set of Trees to Calculate

Purpose: To verify that RBridges calculate the set of trees advertised by the highest priority RBridge first.

References:

- [TRILL] – 4.5

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part.

Procedure:

Part A: Less than K Roots Specified in S Set of Highest Priority RBridge

1. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
2. Configure all RBridges to support a maximum of and to desire 2 computed distribution trees.
3. Configure TRB1 to advertise a set, {TRB2}, of desired distribution tree roots.
4. Enable TRILL on all RBridges.
5. Send multicast ethernet data from TES0 to TES3 until load balancing occurs.
6. Observe the frames on all links.

Part B: K Roots Specified in S Set of Highest Priority RBridge

7. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
8. Configure all RBridges to support a maximum of and to desire 2 computed distribution trees.
9. Configure TRB1 to advertise a set, {TRB0, TRB2}, of desired distribution tree roots.
10. Enable TRILL on all RBridges.
11. Send multicast ethernet data from TES0 to TES3 until load balancing occurs.
12. Observe the frames on all links.

Part C: Greater than K Roots Specified in S Set by Highest Priority RBridge

13. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
14. Configure all RBridges to support a maximum of and to desire 2 computed distribution trees.
15. Configure TRB1 to advertise a set, {TRB0, TRB2, TRB1}, of desired distribution tree roots. This set shall be configured in that order.
16. Enable TRILL on all RBridges.
17. Send multicast ethernet data from TES0 to TES3 until load balancing occurs.
18. Observe the frames on all links.

Part D: Priority 0 Root Specified in S Set by Highest Priority RBridge

19. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
20. Configure all RBridges to support a maximum of and to desire 2 computed distribution trees.
21. Configure TRB2 with a distribution tree root priority of 0x0000 (0).
22. Configure TRB1 to advertise a set, {TRB2}, of desired distribution tree roots.
23. Enable TRILL on all RBridges.
24. Send multicast ethernet data from TES0 to TES3 until load balancing occurs.
25. Observe the frames on all links.

Observable Results:

- *Part A*

Step 6: Once TRB0 begins to load balance the multicast ethernet data must be sent over a

*University of New Hampshire
InterOperability Laboratory*

maximum of two distribution trees. Those distribution trees must be rooted at TRB2 and TRB1.

- *Part B*

Step 12: Once TRB0 begins to load balance the multicast ethernet data must be sent over a maximum of two distribution trees. Those distribution trees must be rooted at TRB2 and TRB0.

- *Part C*

Step 18: Once TRB0 begins to load balance the multicast ethernet data must be sent over a maximum of two distribution trees. Those distribution trees must be rooted at TRB2 and TRB0.

- *Part D*

Step 25: Once TRB0 begins to load balance the multicast ethernet data must be sent over a maximum of two distribution trees. Those distribution trees must be rooted at TRB0 and TRB1.

Possible Problems:

- TRB0 may not support load balancing and may only use one distribution tree. This test may be omitted by such devices. If multiple end stations, sending multicast ethernet data to link 3, need to be introduced onto link 0 for load balancing to occur, the configuration can be altered to account for this.

Test TRILL_Interop.3.5: Tie Breaking

Purpose: To verify that RBridges use the correct tie breaking mechanism for distribution tree calculation.

References:

- [TRILL] – 4.5.1

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part.

Procedure:

1. Configure TRB0 to have a distribution tree root priority of 0x8001(32,769).
2. Configure TRB0 to have a second nickname of 0x8010 (32,784). This nickname shall be referred to as TRB0-1.
3. Configure TRB1 to advertise a set, {TRB0, TRB0-1}, of desired distribution tree roots. This set shall be configured in that order.
4. Configure all RBridges to support a maximum of and to desire 2 computed distribution trees.
5. Configure TRB0's link 2 port to have a link cost of 2.
6. Enable TRILL on all RBridges.
7. Send multicast ethernet data from TES0 to TES3 until load balancing occurs.
8. Observe the frames on all links.

Observable Results:

Step 8: Once TRB0 begins to load balance the multicast ethernet data must be sent over a maximum of two distribution trees. Those distribution trees must be rooted at TRB0 and TRB0-1. The distribution tree rooted at TRB0 must go from TRB0 to TRB1 to TRB2. The distribution tree rooted at TRB0-1 must go from TRB0 then split to both TRB1 and TRB2.

Possible Problems:

- TRB0 may not support load balancing and may only use one distribution tree. This test may be omitted by such devices. If multiple end stations, sending multicast ethernet data to link 3, need to be introduced onto link 0, for load balancing to occur the configuration can be altered to account for this.

Test TRILL_Interop.3.6: No Receivers Pruning

Purpose: To verify that RBridges properly prune off links that do not have receivers on them.

References:

- [TRILL] – 4.5.3

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part.

Procedure:

1. Disable TES2's link 2.
2. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
3. Enable TRILL on all RBridges.
4. TES0 transmits multicast ethernet data on link 0.
5. Observe the frames on all links.

Observable Results:

Step 5: The multicast ethernet data and the TRILL encapsulated multicast ethernet data must not be sent on link 2.

Possible Problems:

- None

Group 4: VLANs

Scope:

The following tests are designed to verify TRILL interoperability between RBridges across VLAN tagged networks.

Overview:

These tests verify that TRILL RBridges can handle known unicast, unknown unicast, multicast, and broadcast data in the ingress, transit, and egress situations. All the forwarding tested in this section is based on single VLAN tagged ethernet data. There are also tests for 802.1 incremental deployment with a single VLAN involved, simple multiple VLAN TRILL topologies, and VLAN mapping.

Test TRILL_Interop.4.1: VLAN Ingress Encapsulation

Purpose: To verify that an RBridge properly encapsulates VLAN data on ingress into the campus.

References:

- [TRILL] – 2.4.1, 4.6.1.1

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part. The [Extended Common Topology](#) is used.

Procedure:

Part A: Known Unicast Ingress Encapsulation

1. Configure all devices with trunk ports for VLAN 1 on all links.
2. Configure TRB0's link 2 port to have a link cost of 2.
3. Enable TRILL on all RBridges.
4. TES3 transmits unicast ethernet data to TES0 on link 3.
5. TES0 transmits unicast ethernet data to TES3 on link 0.
6. Observe the frames on all links.

Part B: Unknown Unicast Ingress Encapsulation

7. Configure all devices with trunk ports for VLAN 1 on all links.
8. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
9. Enable TRILL on all RBridges.
10. TES0 transmits unicast ethernet data to TES3 on link 0.
11. Observe the frames on all links.

Part C: Broadcast Ingress Encapsulation

12. Configure all devices with trunk ports for VLAN 1 on all links.
13. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
14. Enable TRILL on all RBridges.
15. TES0 transmits broadcast ethernet data on link 0.
16. Observe the frames on all links.

Part D: Multicast Ingress Encapsulation

17. Configure all devices with trunk ports for VLAN 1 on all links.
18. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
19. Enable TRILL on all RBridges.
20. TES0 transmits multicast ethernet data on link 0.
21. Observe the frames on all links.

Observable Results:

- *Part A*
Step 6: TRB0 must ingress the ethernet data from TES0 to TES3 and TRB1 must egress the data. The data must not be flooded to every link.
- *Part B*
Step 11: TRB0 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.
- *Part C*

*University of New Hampshire
InterOperability Laboratory*

Step 16: TRB0 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.

- *Part D*

Step 21: TRB0 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.

Possible Problems:

- None

Test TRILL_Interop.4.2: VLAN Transit Forwarding

Purpose: To verify that an RBridge properly forwards TRILL VLAN data through the campus.

References:

- [TRILL] – 4.6.2, 4.6.2.4

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part. The [Extended Transit Topology](#) is used.

Procedure:

Part A: Known Unicast Transit Forwarding

1. Configure all devices with trunk ports for VLAN 1 on all links.
2. TRB2 is disabled on link 3 and enabled on link 0. TRB0 is disabled on link 0.
3. Enable TRILL on all RBridges.
4. TES3 transmits unicast ethernet data to TES0 on link 3.
5. TES0 transmits unicast ethernet data to TES3 on link 0.
6. Observe the frames on all links.

Part B: Unknown Unicast Transit Forwarding

7. Configure all devices with trunk ports for VLAN 1 on all links.
8. TRB2 is disabled on link 3 and enabled on link 0. TRB0 is disabled on link 0.
9. Enable TRILL on all RBridges.
10. TES0 transmits unicast ethernet data to TES3 on link 0.
11. Observe the frames on all links.

Part C: Broadcast Transit Forwarding

12. Configure all devices with trunk ports for VLAN 1 on all links.
13. TRB2 is disabled on link 3 and enabled on link 0. TRB0 is disabled on link 0.
14. Enable TRILL on all RBridges.
15. TES0 transmits broadcast ethernet data on link 0.
16. Observe the frames on all links.

Part D: Multicast Transit Forwarding

17. Configure all devices with trunk ports for VLAN 1 on all links.
18. TRB2 is disabled on link 3 and enabled on link 0. TRB0 is disabled on link 0.
19. Enable TRILL on all RBridges.
20. TES0 transmits multicast ethernet data on link 0.
21. Observe the frames on all links.

Part E: ESADI Transit Forwarding

22. Configure all devices with trunk ports for VLAN 1 on all links.
23. TRB2 is disabled on link 3 and enabled on link 0. TRB0 is disabled on link 0.
24. Enable TRILL on all RBridges.
25. Observe the frames on all links.

Observable Results:

- *Part A*
Step 6: TRB2 must ingress the ethernet data from TES0 to TES3 and TRB1 must egress the data. The data must not be flooded to every link.

*University of New Hampshire
InterOperability Laboratory*

- *Part B*
Step 11: TRB2 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.
- *Part C*
Step 16: TRB2 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.
- *Part D*
Step 21: TRB2 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.
- *Part E*
Step 25: TRB1 and TRB2 must ingress TRILL encapsulate ESADI frames into the network. TRB1 must be the distribution tree root for these frames. TRB0 must be the appointed forwarder on link 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.

Possible Problems:

- TRB1 or TRB2 might not support any, or the same, ESADI protocols.

Test TRILL_Interop.4.3: VLAN Egress Decapsulation

Purpose: To verify that an RBridge properly decapsulates VLAN data on egress out of the campus.

References:

- [TRILL] – 4.6.2, 4.6.2.4, 4.6.2.5

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part. The [Extended Common Topology](#) is used.

Procedure:

Part A: Known Unicast Egress Decapsulation

1. Configure all devices with trunk ports for VLAN 1 on all links.
2. Configure TRB2's link 2 port to have a link cost of 2.
3. Enable TRILL on all RBridges.
4. TES0 transmits unicast ethernet data to TES3 on link 3.
5. TES3 transmits unicast ethernet data to TES0 on link 0.
6. Observe the frames on all links.

Part B: Unknown Unicast Egress Decapsulation

7. Configure all devices with trunk ports for VLAN 1 on all links.
8. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
9. Enable TRILL on all RBridges.
10. TES0 transmits unicast ethernet data to TES3 on link 0.
11. Observe the frames on all links.

Part C: Broadcast Egress Decapsulation

12. Configure all devices with trunk ports for VLAN 1 on all links.
13. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
14. Enable TRILL on all RBridges.
15. TES0 transmits broadcast ethernet data on link 0.
16. Observe the frames on all links.

Part D: Multicast Egress Decapsulation

17. Configure all devices with trunk ports for VLAN 1 on all links.
18. Configure TRB1's distribution tree root priority to 0x8001 (32,769).
19. Enable TRILL on all RBridges.
20. TES0 transmits multicast ethernet data on link 0.
21. Observe the frames on all links.

Observable Results:

- *Part A*
Step 6: TRB1 must ingress the ethernet data from TES0 to TES3 and TRB0 must egress the data.
- *Part B*
Step 11: TRB1 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.
- *Part C*

*University of New Hampshire
InterOperability Laboratory*

Step 16: TRB1 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.

- *Part D*

Step 21: TRB1 must ingress the ethernet data. TRB1 must be the distribution tree root. TRB0 must be the appointed forwarder on link 0, 1, and 4. TRB1 must be the appointed forwarder on link 3. TRB2 must be the appointed forwarder on link 2.

Possible Problems:

- None

Test TRILL_Interop.4.4: VLAN Incremental Deployment Functionality

Purpose: To verify that the TRILL functions when incrementally deployed with 802.1Q and VLANs are in use.

References:

- [TRILL] – 2.6.2
- [802.1Q-2005]

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part.

Procedure:

1. Configure all devices with trunk ports for VLAN 1 on all links.
2. TRB0 is disabled on link 2.
3. The latest version of STP supported is enabled on TRB1.
4. Enable TRILL on all RBridges except for TRB1.
5. TES2 transmits unicast ethernet data to TES0 on link 2.
6. Observe the frames on all links.
7. TES0 transmits unicast ethernet data to TES2 on link 0.
8. Observe the frames on all links.

Observable Results:

Step 6: The unknown ethernet data must travel to all links in the 802.1Q cloud.

Step 8: The known ethernet data must travel from TES0 to TRES3 thru the 802.1Q cloud.

Possible Problems:

- TRB1 may not support 802.1Q. If this is the case then a switch that does should be used instead.

Test TRILL_Interop.4.5: Multiple VLANs

Purpose: To verify that the RBridge system functions when multiple VLANs are used.

References:

- [TRILL] – 2.4.1, 4.6.1.1

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part.

Procedure:

1. Configure all devices with trunk ports for VLAN 2 and 3 on all links except link 1 should not have VLAN 2 configured and link 2 should not have VLAN 3 configured.
2. All links should have a link cost of 1.
3. Enable TRILL on all RBridges.
4. TES3 transmits unicast ethernet data to TES0 on link 3 and tagged with VLAN 2.
5. TES3 transmits unicast ethernet data to TES0 on link 3 and tagged with VLAN 3.
6. TES0 transmits unicast ethernet data to TES3 on link 0 and tagged with VLAN 2.
7. TES0 transmits unicast ethernet data to TES3 on link 0 and tagged with VLAN 3.
8. Observe the frames on all links.

Observable Results:

Step 8: TRB0 must ingress the VLAN 2 ethernet data and TRB2 must egress the data. TRB0 must ingress the VLAN 3 ethernet data and TRB1 must egress the data.

Possible Problems:

- None

Test TRILL_Interop.4.6: VLAN Mapping

Purpose: To verify that the RBridges can properly map VLANs over a TRILL campus.

References:

- [TRILL] – 4.4.5

Test Setup: The [Common Test Setup](#) is performed after each step. The [Common Test Cleanup](#) is performed after each part.

Procedure:

1. Configure all devices with trunk ports for VLAN 1 on link 0.
2. Configure all devices with trunk ports for VLAN 2 on links 1, 2, and 3.
3. Configure TRB0 to map VLAN 1 to VLAN 2 from links 0 and 1 to link 2.
4. Enable TRILL on all RBridges.
5. TES3 transmits unicast ethernet data to TES0 on link 3 and tagged with VLAN 2.
6. Observe the frames on all links.
7. TES0 transmits unicast ethernet data to TES3 on link 0 and tagged with VLAN 1.
8. Observe the frames on all links.

Observable Results:

Step 6: The VLAN tagging must be mapped correctly, from VLAN 2 on link 1 to VLAN 1 on link 0, as the unicast data is forwarded.

Step 8: The VLAN tagging must be mapped correctly, from VLAN 1 on link 0 to VLAN 2 on link 1, as the unicast data is forwarded.

Possible Problems:

- Some devices may not support VLAN mapping. The end stations may not support VLAN tagging. If they do not configure the routing bridges to tag and un-tag frames coming and going to them respectively.