Cisco Release 12.1(8b)E12 Safe Harbor Testing for Financial Enterprise Customers

Version History

Version Number	Date	Notes
1	November 15, 2002	This document was created.

Executive Summary

Cisco IOS Safe Harbor is an initiative that provides the global financial services enterprise customer with a stable Cisco IOS 12.1 E version-of-choice. Safe Harbor focuses on satisfying customer quality requirements in key vertical markets. This program links and expands upon several Cisco testing projects, including development, regression testing, and systems testing critical to the success of the financial services business. Safe Harbor is the successful completion of extensive validated testing for each release targeting the financial enterprise market.

The Cisco nEverest program will integrate testing for many diverse vertical markets and will leverage Safe Harbor testing to ensure a "holistic" approach to the general improvement of Cisco IOS software.

This document describes the testing environment, test plans, and results.

This document contains the following sections:

CISCO SYSTEMS

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Copyright © 2002. Cisco Systems, Inc. All rights reserved.

About Cisco IOS Safe Harbor

The goal of Cisco IOS Safe Harbor is to provide improved network stability, reliability, and performance with respect to Cisco IOS software. Safe Harbor involves testing the feature sets and protocols in a particular Cisco IOS Release 12.1 E image on the Catalyst 6500 platform to provide high quality code for the financial services business. This combination of features, hardware, and image is tested in a laboratory environment that simulates the financial services business network environment using regularly updated topologies and configurations provided by the financial customer. For information on the hardware tested and the network setup of the test environment, see the "Financial Lab Topology" section on page 2.

The groups of feature sets that are tested include the following: hardware redundancy, Layer 2 features, hardware forwarding features, Layer 3 routing features, network management features, and several miscellaneous features. Regression tests are conducted to validate existing features and ensure that functionality is maintained. Negative tests are designed and conducted to stress the features and their interoperability. For information on each feature and its testing, see the "Feature Sets Testing" section on page 12.

During the testing, the network is placed under loads that are consistent with those in a financial services network. A standard suite of tools (for example, Netcom Smartbits, IXIA packet generator, or Cisco Pagent) is used to generate network traffic. Network testing includes a combination of automated and manual tests. Simple Network Management Protocol (SNMP) is used to poll the network during the tests, and all tests are analyzed. For a summary of the test results, see the "Test Results Summary" section on page 9.

Note Safe Harbor testing does not address issues that might exist in the customer change control and operations processes.

Financial Lab Topology

Figure 1 through Figure 5 show the base Native IOS financial lab topology. The financial services network environment configured in the lab includes the following hardware:

- Forteen Catalyst 6500 switches running Cisco Native IOS Release 12.1(8b)E12 (SH1-97 to SH1-110)
- Two Catalyst 6500 switches that are running Hybrid CatOS 6.3(8) with no routing (Dist A-1 and Dist A-2)
- Pagent test devices to simulate the Internet Service Provider's (ISPs), Area 3, and Area 4, injecting Border Gateway Protocol (BGP), Open Shortest Path First (OSPF), and Enhanced Interior Gateway Routing Protocol (EIGRP) routes
- IXIA test devices to generate simulated customer traffic

The hardware configuration in the financial test lab includes a combination of distributed fabric, fabric-capable, and nonfabric modules.

Note The Switch Fabric Module is supported only with the Supervisor Engine 2 in the Catalyst 6500 series switch.

Basic Topology: Port Channel Deployment

Figure 1 through Figure 5 show the port channel deployment for the Safe Harbor testing. Catalyst 6500 series switches running Native Cisco IOS support both Layer 2 (L2) and Layer 3 (L3) EtherChannels, with up to eight ports aggregated in a single Etherchannel interface. All interfaces in each EtherChannel must be identically configured (the same speed, all Layer 2 or Layer 3, and so on).

EtherChannel load balancing can use either MAC addresses or IP addresses, and either source or destination or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch.

EtherChannel is a trunking technology that groups together multiple full-duplex 802.3 Ethernet interfaces to provide fault-tolerant high-speed links between switches, routers, and servers. An EtherChannel interface (consisting of up to eight Ethernet interfaces) is treated as a single interface; this is called a port channel.

The port channels configured for Safe Harbor testing are Gigabit EtherChannels (GECs). The following types of GEC port channels are configured and tested for Safe Harbor:

- Layer 3 GEC distributed forwarding card (DFC)
- Layer 3 GEC DFC and non-DFC mixed
- Layer 3 GEC using fabric-capable modules, nonfabric modules, and combinations of both
- Layer 2 GEC using fabric-capable modules, nonfabric modules, and combinations of both

Basic Topology: Routing Protocols

The following routing protocols are configured for Safe Harbor testing:

- BGP
  - External Border Gateway Protocol (eBGP)
  - Interior Border Gateway Protocol (iBGP)
- EIGRP
- OSPF

Figure 1 through Figure 5 shows the following:

- Where each routing protocol is configured in the basic test lab topology.
- The eBGP and iBGP routing protocol deployment for the Safe Harbor testing.
- The EIGRP routing protocol deployment for Safe Harbor testing.
- OSPF routing protocol areas configured for Safe Harbor testing.

Figure 1 Routing Topology

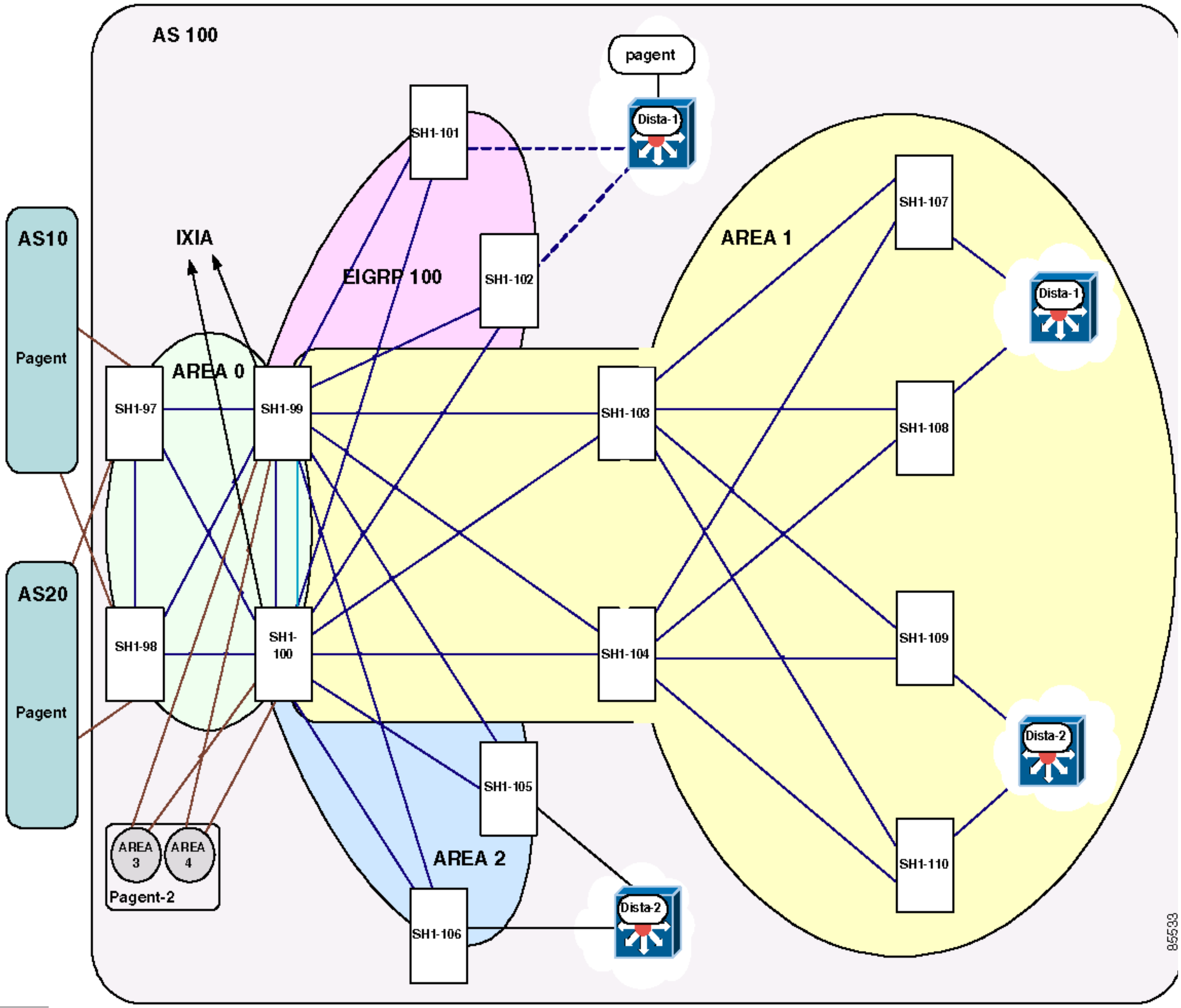


Figure 2 Quadrant 1 Detailed Interconnections Information

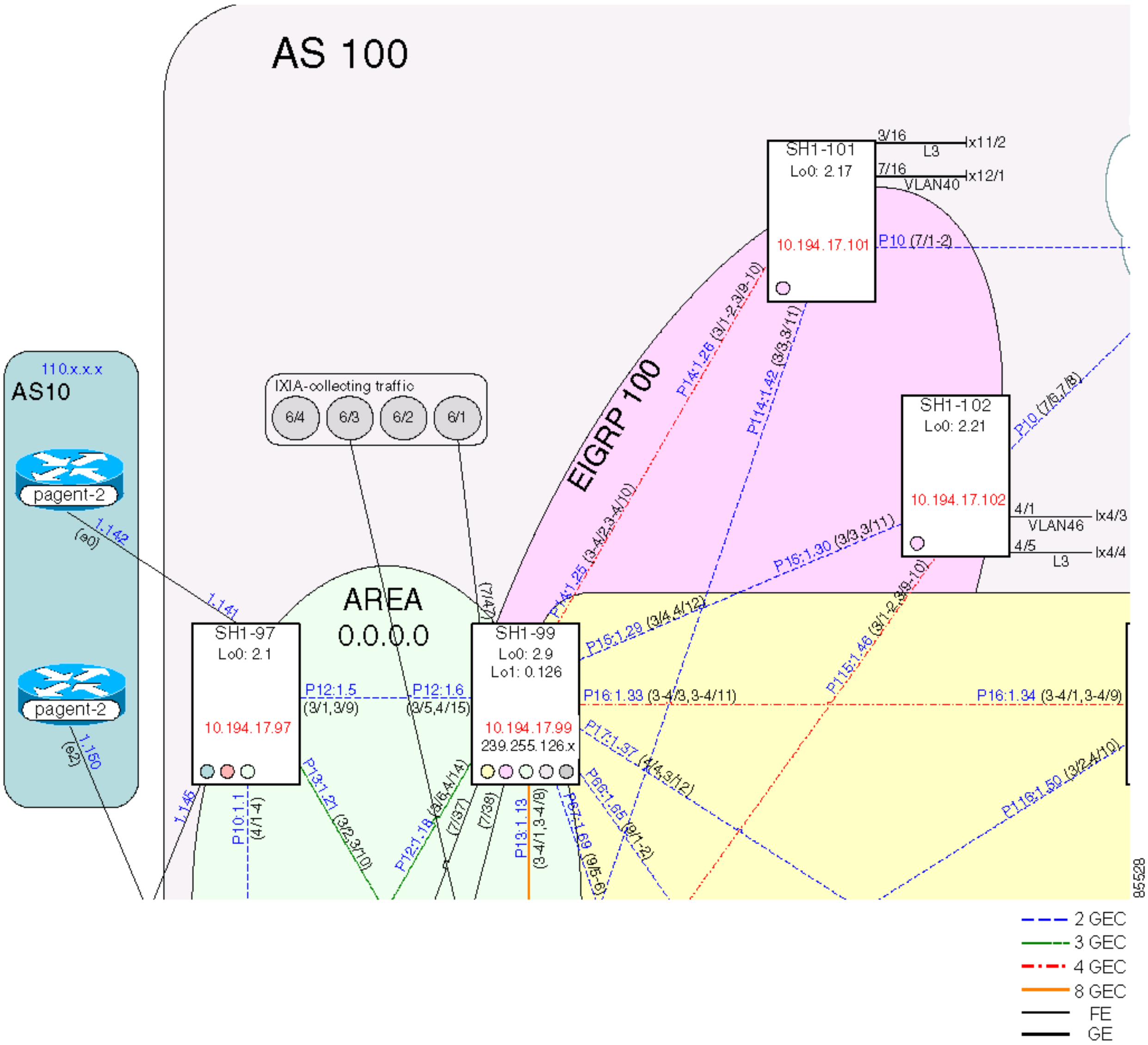


Figure 3 Quadrant 2 Detailed Interconnections Information

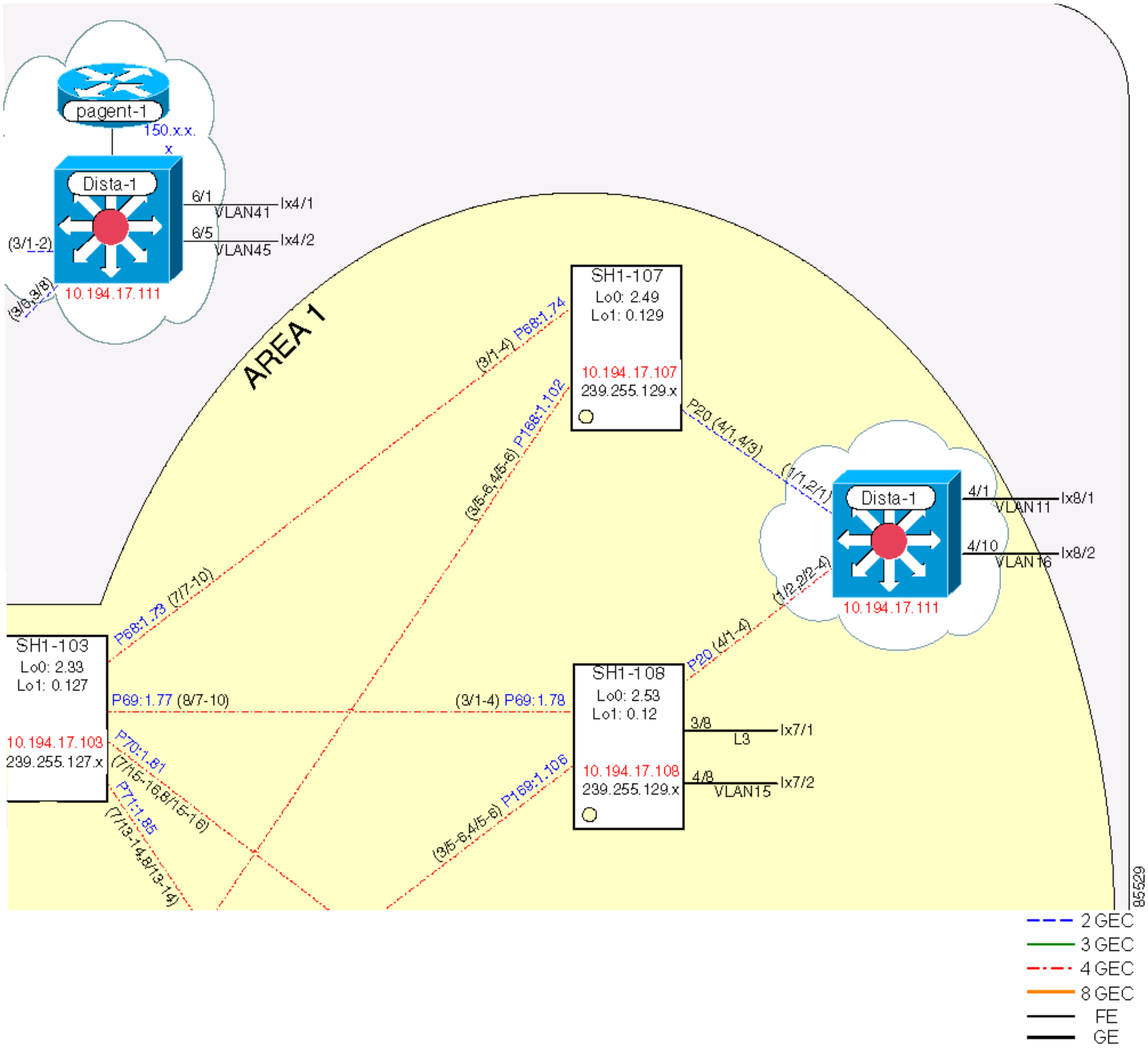


Figure 4 Quadrant 3 Detailed Interconnections Information

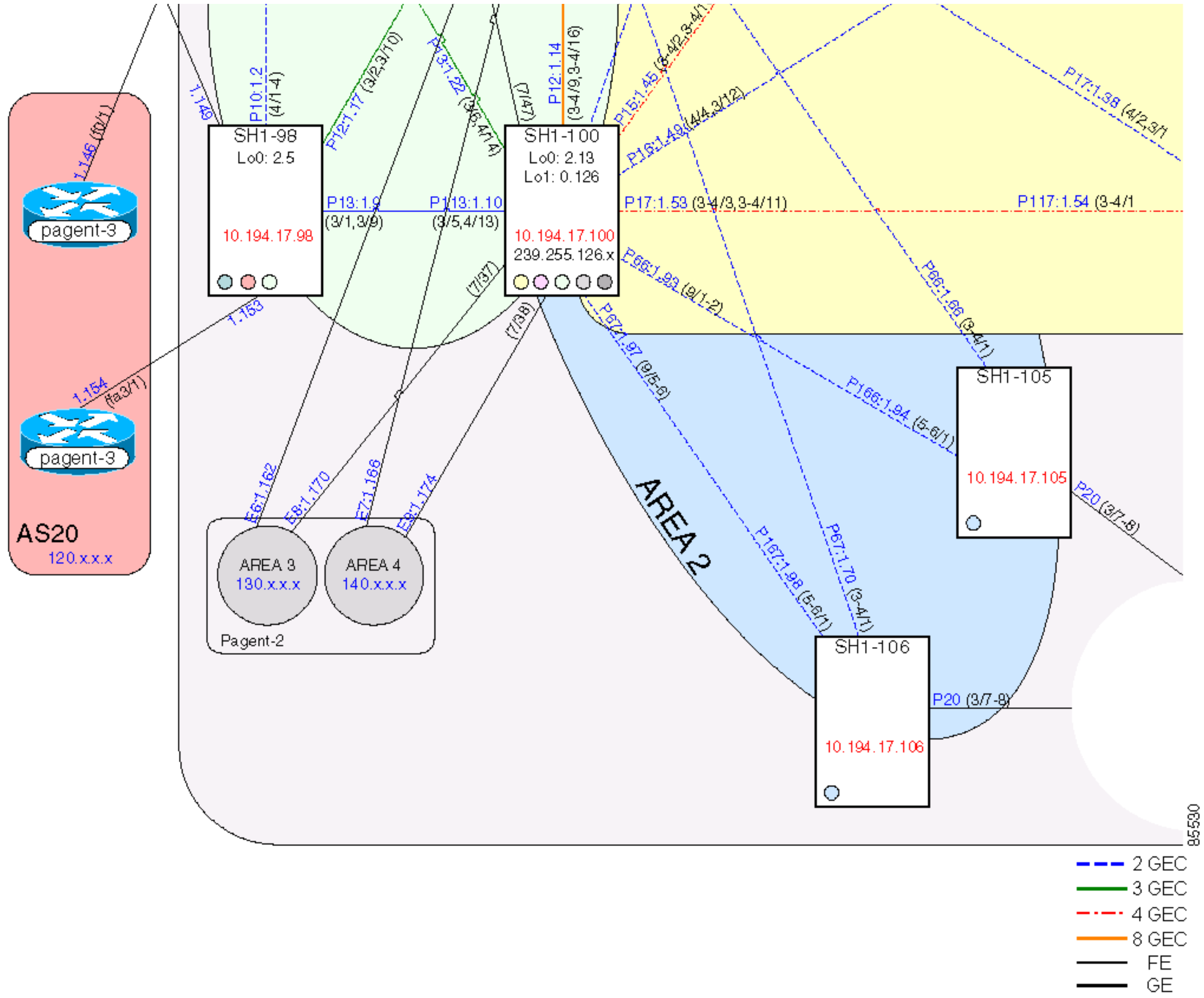
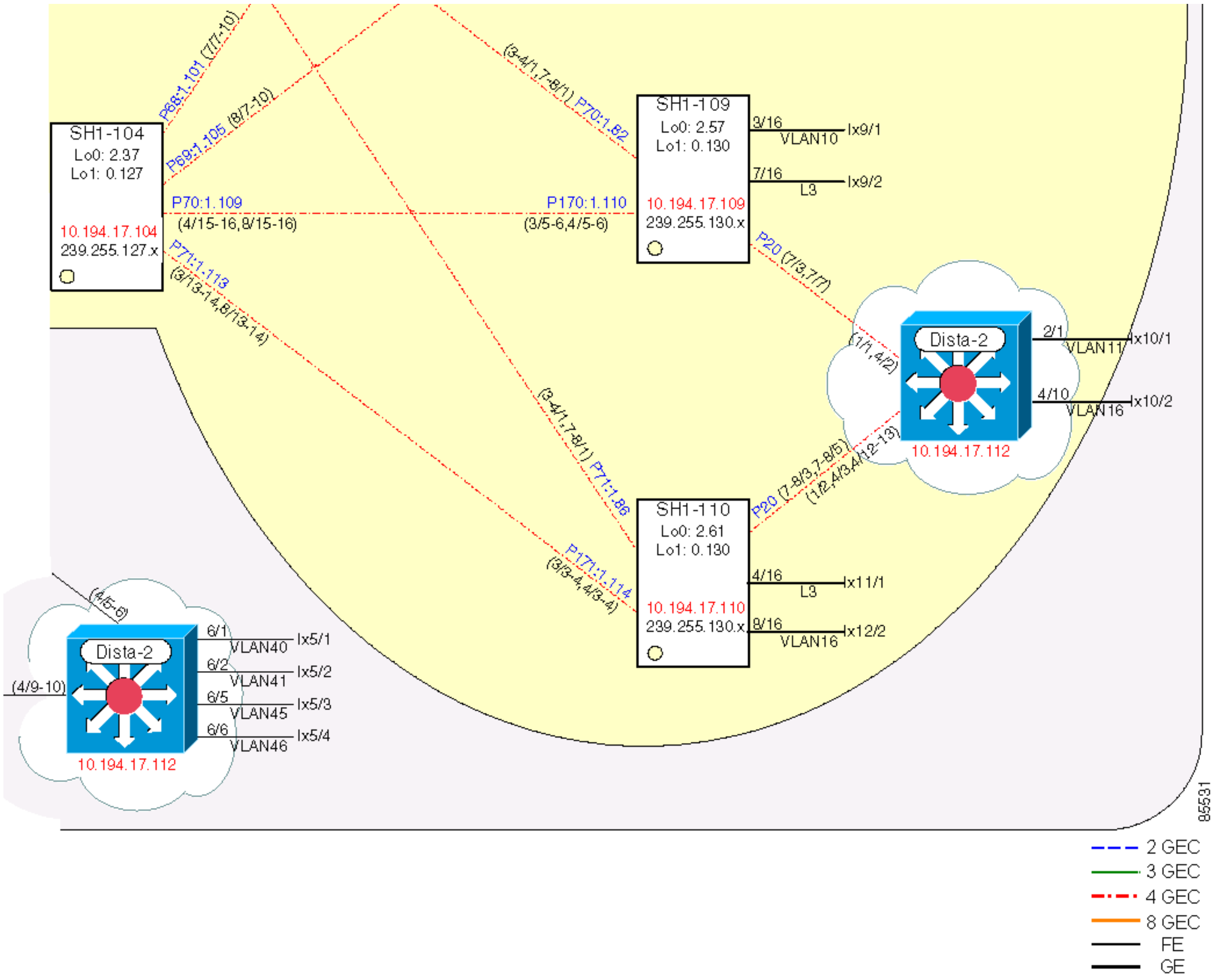


Figure 5 Quadrant 4 Detailed Interconnections Information



Test Results Summary

Table 1 summarizes the results of all completed testing as part of the Cisco IOS Safe Harbor initiative. Table 1 includes the following information: The feature or function tested, the section that describes the feature set to which the feature or function belongs, the results of the feature or function tests (pass or fail), the component tests for each feature/function, and any DDTS found during the Safe Harbor testing.

Note These test results are specific to the technologies covered and the actual test scenarios in which they were tested. Safe Harbor is designed to cover critical path areas and augment ongoing regression and systems testing.

For a compete list of IOS commands and usage in this document, refer to http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/index.htm.

Table 1 Safe Harbor Test Results Summary

Test Suites	Feature/Function	Tests	Results	DDTS
Hardware Redundancy, page 12	Hardware Redundancy, page 12	Fabric Flap, page 12 Supervisor Failover, page 14	Pass	None
Layer 2 Features, page 16	Spanning Tree Protocol, page 16	Basic Spanning Tree Protocol Configuration, page 16	Pass	None
	Unidirectional Link Detection-Aggressive Mode, page 17	Basic UDLD Test on Layer 2 Link, page 18 Basic UDLD Test on Layer 3 Link, page 20	Pass	None
	Trunking, page 21	Basic Trunk Configuration, page 22 Failure and Recovery, page 23	Pass	None
	Port Aggregation Protocol (Channeling), page 24	Basic Layer 2 Channeling Configuration, page 25 Basic Layer 3 Channeling Configuration, page 26 Layer 2 and Layer 3 EtherChannel Load Balance, page 28 Gigabit Ethernet Module Reset, page 29	Pass	None
	VLAN Trunking Protocol, page 31	Basic VLAN Trunking Protocol Configuration, page 31	Pass	None
Hardware Forwarding Features, page 33	IP Unicast, page 33	Layer 2 Gigabit EtherChannel Failover, page 33 Layer 3 Gigabit EtherChannel Failover, page 35	Pass	None

Table 1 Safe Harbor Test Results Summary (continued)

Test Suites	Feature/Function	Tests	Results	DDTS
	IP Multicast, page 36	Basic Multicast and Multicast Source Discovery Protocol, page 37 Basic IGMP and CGMP Functionality, page 38 Core Multicast Source Discovery Protocol, page 40 Non-Reverse Path Forwarding Rate Limiting and Multicast Stub, page 41 Gigabit EtherChannel Failover: Non-dCEF GEC Failover, page 45 Gigabit EtherChannel Failover: Mixed GEC Failover, page 47 Gigabit EtherChannel Failover: dCEF GEC Failover, page 49 Switch Fabric Module Failover, page 51 Gigabit Ethernet Module Failover, page 53 Protocol Independent Module-Designated Router Failover, page 55 Auto-Rendezvous Point Functionality and Failover, page 57 Layer 3 Interface Multicast Negative, page 60 Unicast and Multicast Test with 130K Injected IP Routes, page 62 IP PIM Neighbor-Filter Command, page 64 Dual Sources, page 65 Secondary Subnet, page 66 MSDP Failover, page 68	Pass	None
Layer 3 Routing Features, page 71	Cisco Express Forwarding, page 71	Cisco Express Forwarding Packet-Switching, page 71 Cisco Express Forwarding Table Entries, page 72 CEF Load Balance, page 73	Pass	None

Table 1 Safe Harbor Test Results Summary (continued)

Test Suites	Feature/Function	Tests	Results	DDTS
	Open Shortest Path First, page 75	Autocost, page 75 Passive Interface, page 76 Filtering, page 78 OSPF Redistribution, page 79 OSPF Topology Database, page 80	Pass	None
	Border Gateway Protocol, page 82	Scale to Ten BGP Neighbors in Core, page 82 Route Redistribution, page 83 BGP Neighbor Flap, page 83	Pass	None
	Hot Standby Routing Protocol, page 84	Basic HSRP, page 85 HSRP Failover, page 86 HSRP Failover Using Fast Timers, page 87	Pass	None
	Enhanced Interior Gateway Routing Protocol, page 88	EIGRP Summarization, page 89 EIGRP Redistribution, page 89	Pass	None
Network Management Features, page 91	Simple Network Management Protocol, page 91	Basic Functionality Shut/No Shut Interface, page 91 Protos Request Application, page 92	Pass	None CSCDW62852
	TACACS, page 93	Verify User Authentication, page 93	Pass	None
Miscellaneous Features, page 94	NTP Basic Functionality, page 94	NTP Basic Functionality, page 94	Pass	None
	Syslog Basic Functionality, page 95	Syslog Basic Functionality, page 95	Pass	None
	User Datagram Protocol Broadcast Flooding, page 96	UDP Broadcast Flooding, page 96	Pass	None
	System Upgrade, page 98	System Upgrade, page 98	Pass	None

Feature Sets Testing

Functionality critical to the global financial service business tested for the Cisco IOS Safe Harbor release is described in the following sections:

- Hardware Redundancy, page 12
- Layer 2 Features, page 16
- Hardware Forwarding Features, page 33
- Layer 3 Routing Features, page 71
- Network Management Features, page 91
- Miscellaneous Features, page 94

Hardware Redundancy

Whenever a fault is encountered, the redundant module takes over the functions of the failed hardware module. Testing hardware redundancy for Safe Harbor involves performing various failover scenarios to verify that internal hardware redundancy fails over as expected. The tests described in the following sections were performed:

- Fabric Flap, page 12
- Supervisor Failover, page 14

Fabric Flap

This test reset the active SFM in the system repeatedly to verify that SFM failover operates as designed. The device under test (DUT) was SH1-103. With an IXIA traffic stream passing through it, the DUT SFM was failed 20 times. Test results included measures of the period during which no traffic was passed and the time required for the failed SFM to come back online to the "standby" state.

Test Plan

The procedure used to perform the hardware redundancy fabric flap test follows:

Step 1 Enter the **show memory summary** command to take an initial memory snapshot of the DUT. Final memory and CPU utilization results are provided in the last step of this procedure.

Step 2 Verify that the background unicast traffic is flowing (rate = 10,000 packets per second [pps}) from devices Dista-1 (IXIA 13/1) and Dista-2 (IXIA 13/2) to SH1-105 (IXIA 1/1).

Step 3 Configure the **ip ospf cost 100** command on the following interfaces to verify that traffic coming from Dista-1 and Dista-2 chooses the path through SH1-103: SH1-104 port-channel 69, port-channel 71; SH1-108 port-channel 169; SH1-110 port-channel 171:

Commands

- **ip ospf cost 100**

Step 4 Verify that traffic is being switched using the crossbar fabric on device SH1-103:

Commands

- **show fabric switching-mode**
- **show interfaces** *interface* **counters**

Step 5 Enter the following commands to determine the active SFM in the DUT and reset it:

Commands

- **show module**
- **hw-module module** *5|6* **reset**

Step 6 Track the time stamps between the time it is reset and the time it comes back online and the time required for the standby SFM to become active. Record these values.

Step 7 Repeat Step 5 and Step 6 nineteen times.

Step 8 Determine the total number of packets dropped and calculate the mean for the total test period.

Step 9 Determine the mean down time for the reset module and the mean time that traffic is not passed (Table 2).

Table 2 Reset Module Mean Down Time and Mean Time Traffic Not Passed

Component	Fastest (sec)	Average (sec)	Slowest (sec)
Standby >> Active	0.008	0.012	0.009
Reset Recovery	25.365	0.012	25.371
Traffic Loss (from Dista-1)	-	-	~1.5
Traffic Loss (from Dista-2)	-	-	~1.5

Step 10 Use the **show memory summary** and **show processes cpu** commands to verify that memory and CPU did not suffer severe or sustained impact during the test on the DUT.

Expected Results

We expect that SFM failover operates as designed after repeatedly resetting the active SFM in the system.

Results

Table 3 shows the fabric flap test results.

Table 3 Fabric Flap Test Results

Tests	Results
Fabric Flap	Pass

Supervisor Failover

This test verified the proper operation of redundant supervisors during a series of twenty continual resets. The goal of this test was to verify that the box failed between the supervisors as designed, recovered, and forwarded traffic. This test was further designed to verify that failure operations were within the design guidelines for the applicable hardware and software versions under test. Although this test measured time, it by no means measures the speed at which failover can take place (which is dependent upon the configuration and line cards in the system).

Because both supervisor engines 1 and 2 are covered in this test, two devices were used during this procedure. These devices were SH1-107 (Sup1) and SH1-109 (Sup2). A traffic stream was sent from Dista-1 to Dista-2 at a fixed rate. Configurations were altered so that this traffic stream can go only through SH1-108 or SH1-110, depending on the test being run. The amount of traffic loss during device resets was used to determine the recovery time (failover time).

Test Plan

The procedure used to perform the supervisor failover test follows.

Step 1 For Sup1, enter the **show ip route summary** command to display some of the factors that may contribute to reload times. Such factors may include configuration size, memory used, size of the IP routing table:

Commands

- **show ip route summary**
- **show processes memory | include Used**
- **directory nvram:**
- **show running-config**

Step 2 Shut down interface port-channel 10 on device SH1-107, which will ensure that the traffic generated in Step 3 goes through SH1-108 to get to its destination.

Step 3 Begin an IXIA traffic stream (1.6 million packets) from Dista-1 to Dista-2. Send this traffic at a rate of 10,000 pps.

Step 4 Issue the **reload** command on SH1-108.

Step 5 Measure the period of time between when traffic stopped and it was started again. This period of time is considered the failover time.

Step 6 Repeat Step 1 through Step 4 ten times, recording the failover times. The following is raw data for repeated Steps 2 to 10 (see Table 4).

Table 4 Supervisor1 (SH1-108) Summarized Results

Component	Fastest (sec)	Average (sec)	Slowest (sec)
Failover time	138.3	139.0	139.8

Step 1 For Sup2, enter the **show ip route summary** command to display some of the factors that may contribute to reload times. Such factors may include configuration size, memory used, and size of the IP routing table:

Commands

- **show ip route summary**
- **show processes memory | include used**
- **directory nvram:**
- **show running-config**

Step 2 Shut down interface port-channel 20 on device SH1-109 which will ensure that the traffic generated in Step 3 goes through SH1-110 to get to its destination.

Step 3 Begin an IXIA traffic stream (1.2 million packets) from Dista-1 to Dista-2. Send this traffic at a rate of 10,000 pps.

Step 4 Enter the **reload** command on SH1-110.

Step 5 Measure the period of time between when the traffic stopped and when it was started again. This period of time is considered the failover time.

Step 6 Repeat Step 1 through Step 4 ten times, recording the failover times. The following is raw data for repeated Steps 2 to 20 (see Table 5).

Table 5 Supervisor2 (SH1-110) Summarized Results

Component	Fastest (sec)	Average (sec)	Slowest (sec)
Failover time	87.1	94.8	108.9

Expected Results

We expect that failure operations are within the design guidelines for the given hardware and software versions under test with no configuration or functionality loss.

Results

Table 6 shows the supervisor failover test results.

Table 6 Supervisor Failover Test Results

Tests	Results
Supervisor Failover	Pass

Layer 2 Features

Layer 2 feature testing for Safe Harbor involves the features:

- Spanning Tree Protocol, page 16
- Unidirectional Link Detection-Aggressive Mode, page 17
- Trunking, page 21
- Port Aggregation Protocol (Channeling), page 24
- VLAN Trunking Protocol

Spanning Tree Protocol

The spanning-tree algorithm provides path redundancy by defining a tree that spans all of the switches in an extended network and then forces certain redundant data paths into a standby (blocked) state. At regular intervals, the switches in the network send and receive spanning-tree packets that they use to identify the path. If one network segment becomes unreachable, or if spanning-tree costs change, the spanning-tree algorithm reconfigures the spanning-tree topology and reestablishes the link by activating the standby path. Because this feature has limited use in the financial customer networks, it received limited coverage in testing here.

The following test was performed:

- Basic Spanning Tree Protocol Configuration, page 16

Basic Spanning Tree Protocol Configuration

This test tested the basic functionality of the Spanning Tree Protocol (STP), including verifying that the various STP states occurred within the defined times; that STP properly converged, with all switches pointing to the correct device as root; and that the CPU did not reach unreasonable levels. The DUTs are SH1-109, SH1-110, and Dista-2.

Note The coverage of STP in Safe Harbor testing for Native IOS is limited because the implementation of STP is limited in the network of Cisco customers.

Test Plan

The procedure used to perform the basic Spanning Tree Protocol configuration test follows.

Step 1 Enter the **show memory summary** command to take an initial memory snapshot of devices SH1-109 and SH1-110. Final memory and CPU utilization results are provided in the last step of this procedure.

Step 2 Verify that trunk links are configured between SH1-109 and Dista-2 and also between SH1-110 and Dista-2, and that they are trunking VLANs 10-20 by using the following commands:

Commands

- **show interfaces** *interface* **trunk**
- **show trunk** *mod/port*

Step 3 Verify that SH1-110 is the root switch for VLANs 10-20 using the **show spanning-tree root** command:

Commands

- **show spanning-tree root**

Step 4 Disable the EtherChannel between SH1-110 and Dista-2.

Step 5 Verify that STP has converged with the new topology (no SH1-110) and that SH1-109 is now root for VLANs 10-20 using the **show spanning-tree root** command:

Commands

- **show spanning-tree root**

Step 6 Reenable the channel that was disabled in Step 4.

Step 7 Verify that STP reconverges with SH1-110 once again as root. Make certain that the convergence times did not exceed configured specifications:

Commands

- **show spanning-tree root**

Step 8 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results for devices SH1-109 and SH1-110.

Expected Results

We expect spanning-tree recalculation occurs in an anticipated time frame. This value depends on the parameters of the spanning-tree domain.

Results

Table 7 shows the basic Spanning Tree Protocol configuration test results.

Table 7 Basic Spanning Tree Protocol Configuration Test Results

Tests	Results
Basic Spanning Tree Protocol Configuration	Pass

Unidirectional Link Detection-Aggressive Mode

The Unidirectional Link Detection-Aggressive Mode (UDLD-AM) protocol allows devices connected through fiber-optic or copper Ethernet cables (for example, Category 5 cabling) to monitor the physical configuration of the cables and detect when a unidirectional link exists. When a unidirectional link is detected, UDLD shuts down the affected port and alerts the user. Unidirectional links can cause a variety of problems, including spanning-tree topology loops and erroneous Layer 3 routing.

Note The lowest value of the UDLD-AM message interval can be only 7 seconds, and the hold down time can be 21 seconds. By default, the HSRP hello timer is 3 seconds and the hold down timer is 10 seconds, and the EIGRP hello timer is 5 seconds and hold down timer is 15 seconds. When the link becomes unidirectional, before the UDLD-AM can shut down the port, the HSRP and EIGRP neighbor will flap. After UDLD-AM shut down the unidirectional port, the HSRP and EIGRP neighbor will stay up stable.

Note If UDLD mode or UDLD-AM is enabled globally on Safe Harbor switches, the interface shows the UDLD message interval as 7 seconds, which is actually the "running message interval." Once the UDLD neighbor is established, the message interval changes to 60 seconds. See CSCdv74001.

The following tests were performed:

- Basic UDLD Test on Layer 2 Link, page 18
- Basic UDLD Test on Layer 3 Link, page 20

Basic UDLD Test on Layer 2 Link

This test created a unidirectional link between SH1-109 (the device under test [DUT]) and Dista-2. Console messages were logged during the mock failure, and port states were recorded.

Test Plan

The procedure used to perform the basic UDLD test on Layer 2 link test follows.

Step 1 Enter the **show memory summary** command to take an initial memory snapshot of device SH1-109. Final memory and CPU utilization results are provided in the last step of this procedure.

Step 2 Verify that UDLD is configured globally on the DUT and that UDLD aggressive mode is configured locally on the interface g7/3 by using the following commands:

Commands

- **show running-config | include udld**
- **show running-config interfaces g7/3**
- **show udld g7/3**

Step 3 Verify that errdisable recovery is configured for a UDLD cause on SH1-109 which means that the system will attempt to recover the interface after the timer interval (30 seconds) by using the following commands:

Note If the system attempts to recover an interface if errdisable state fails, the interface will no longer be shown on the list of err disabled interfaces, but will show an up or down condition.

Commands

- **show running-config | include udld**
- **show errdisable recovery**

Step 4 Change the timer interval of the errdisable recovery to 90 seconds. This change is for testing purposes only so that there is enough time to gather data before the timer expires.

Commands

- errdisable recovery interval 90

- show errdisable recovery

Step 5 Verify that the terminal monitor is enabled on the DUT.

Step 6 With the interface g7/3 in the up/up state (Dista-2 port 1/1 showing "connected"), and both g6/3 and g7/3 bundled in port-channel 20, pull the receive fiber from port 1/1 on Dista-2. Log any messages on the DUT. Determine the interface state of g7/3 on the DUT.

Commands

- **show interfaces g7/3**
- **show udld g7/3**

Step 7 Check the errdisable recovery interface list to verify that g7/3 is on it. Watch as the timer diminishes, and log any messages after the timer reaches zero. Display the current state of the interfaces:

Commands

- **show errdisable recovery**
- **show interfaces g7/3**
- **show udld g7/3**

Step 8 Reinsert the receive fiber into port 1/1 of Dista-2. Copy any log messages here and determine the port/interface states. Verify that interface g7/3 returns to the up/up state and that it resumes its role in port-channel 20:

Commands

- **show interfaces g7/3**
- **show udld g7/3**

Step 9 Reset the errdisable recovery interval to 30 seconds by entering the following commands:

Commands

- **errdisable recovery interval 30**

Step 10 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results for device SH1-109.

Expected Results

We expect that UDLD-AM will detect a unidirectional Layer 2 link, shut down the affected port, and alert the user. We also expect that the link is reestablished when physical connectivity is restored and UDLD-disabled ports are reset.

Results

Table 8 shows the basic UDLD test on Layer 2 link test results.

Table 8 Basic UDLD Test on Layer 2 Link Test Results

Tests	Results
Basic UDLD Test on Layer 2 Link	Pass

Basic UDLD Test on Layer 3 Link

This test involved the emulation of a unidirectional link on a Layer 3 interface. Console messages and interface states were logged throughout the process. The DUTs are SH1-104 and SH1-109.

Test Plan

The procedure used to perform the basic UDLD test on Layer 3 link test follows.

Step 1 Enter the **show memory summary** command to take an initial memory snapshot of devices SH1-109 and SH1-104. Final memory and CPU utilization results are provided in the last step of this procedure.

Step 2 Verify that UDLD is configured globally on both DUTs and that UDLD-AM is configured locally on the interfaces g4/15 of SH1-104 and g3/5 of SH1-109 by using the following commands:

Commands

- **show running-config | include udld**
- **show running-config interfaces** *interface*
- **show udld** *interface*

Step 3 Verify that errdisable recovery is configured for a UDLD cause on both SH1-104 and SH1-109, which means that the system will attempt to recover the interface after the timer interval (30 seconds):

Note If the system attempts to recover an interface in errdisable state fails, the interface will no longer be shown on the list of errdisable interfaces, but will show an up or down condition.

Commands

- **show running-config | include udld**
- **show errdisable recovery**

Step 4 Change the timer interval of the errdisable recovery to 90 seconds. This change is for testing purposes only so that there is enough time to gather data before the timer expires.

Commands

- **errdisable recovery interval 90**
- **show errdisable recovery**

Step 5 Verify that terminal monitoring is enabled on both DUTs.

Step 6 With the interfaces connecting SH1-104 and SH1-109 in the up/up state, and g3/5 of SH1-109 as part of interface port-channel 170, pull the transmit fiber from g3/5 on SH1-109. Log any messages on both devices. Determine the interface state of g3/5 on SH1-109 and g4/15 on SH1-104.

Commands

- **show interfaces** *interface*
- **show udld** *interface*

Step 7 Check the errdisable recovery interface list to verify that g3/5 of SH1-109 is on it. Watch as the timer diminishes, and log any messages after the timer reaches zero. Display the current state of the interfaces:

Commands

- **show errdisable recovery**
- **show interfaces** *interface*

Step 8 Reinsert the transmit fiber into g3/5 of SH1-109. Copy any log messages here and determine the port and interface states. Verify that g3/5 of SH1-109 returns to the up/up state and that it is rebundled with port-channel 170:

Commands

- **show interfaces** *interface*
- **show udld** *interface*

Step 9 Reset the errdisable recovery interval to 30 seconds by using the following commands:

Commands

- **errdisable recovery interval 30**

Step 10 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results for device SH1-109.

Expected Results

We expect that UDLD-AM will detect a unidirectional Layer 3 link, shut down the affected port, and alert the user. We also expect that the link is reestablished when physical connectivity is restored and UDLD-disabled ports are reset.

Results

Table 9 shows the basic UDLD test on Layer 3 link test results.

Table 9 Basic UDLD Test on Layer 3 Link Test Results

Tests	Results
Basic UDLD Test on Layer 3 Link	Pass

Trunking

A trunk is a point-to-point link between one or more switch ports and another networking device such as a router or a switch. Trunks carry the traffic of multiple VLANs over a single link and allow VLANs to be extended across an entire network. Table 10 lists and describes the five modes of trunking on Cisco switches.

Table 10 Trunking Modes on Cisco Switches

Mode	Description
On	Local interface trunks. Sends Dynamic Trunking Protocol (DTP) packets. Puts the port into permanent trunking mode and negotiates to convert the link to a trunk link. The port becomes a trunk port even if the neighboring port does not agree to the change.

Table 10 Trunking Modes on Cisco Switches (continued)

Off	Local interface does not trunk. Puts the port into nontrunking mode and negotiates to convert the link into a nontrunk link. The port becomes a nontrunk port even if the neighboring port does not agree to the change.
Auto	Local interface trunks if it receives DTP packets. Enables the port to convert the link to a trunk link. The port becomes a trunk port if the neighboring port is set to on or desirable mode. This is the default mode for Fast Ethernet and Gigabit Ethernet ports.
Desirable	Local interface sends DTP packets. Makes the port actively attempt to convert the link to a trunk line. The port becomes a trunk port if the neighboring port is set to on, desirable, or auto mode.
Nonnegotiate	Local interface forms a trunk and does not send DTP packets. Puts the port into permanent trunking mode, but prevents the port from generating DTP frames. You must configure the neighboring port normally as a trunk port to establish a trunk link.

The following trunking tests were performed:

- Basic Trunk Configuration, page 22
- Failure and Recovery, page 23

Basic Trunk Configuration

This test verified the basic functionality of the various trunking configurations. A static trunking configuration was tested between devices SH1-107 and Dista-1. A dynamic trunking configuration also was tested between devices SH1-108 and Dista-1.

Test Plan

The procedure used to perform the basic trunk configuration test follows.

Step 1 Verify that static trunking is configured and working on each side of the connection between SH1-107 and Dista-1 by using the following commands. The ports and interfaces are SH1-107 g4/1, g4/3, and Dista-1 1/1, 2/1.

Commands

- **show running-config interfaces** *interface*
- **show trunk** *mod/port*
- **show interfaces** *interface* **trunk**

Step 2 Verify that dynamic trunking is configured and working on each side of the connection between SH1-108 and Dista-1 by using the following commands. The ports and interfaces are SH1-108 g4/1-4 and Dista-1 1/2, 2/2-4.

Commands

- **show running-config interfaces** *interface*
- **show trunk** *mod/port*
- **show interfaces** *interface* **trunk**

Expected Results

We expect basic trunking functionality to work properly and perform correctly in failure and recovery scenarios.

Results

Table 11 shows the basic trunk configuration test results.

Table 11 Trunking Test Results

Tests	Results
Basic Trunk Configuration	Pass

Failure and Recovery

This test verified failure and recovery of both statically and dynamically configured trunks. The static trunk links were between SH1-107 and Dista-1. The dynamic trunk links were between SH1-108 and Dista-1. These links were failed and then recovered, to verify that trunking was reestablished.

Test Plan

The procedure used to perform the failure and recovery test follows.

Step 1 Enter the **show memory summary** command to take an initial memory snapshot of SH1-107 and SH1-108, and use the **show processes cpu** command to begin tracking CPU utilization. Final memory and CPU utilization results are provided in the last step of this procedure.

Step 2 Verify that static trunking is configured between SH1-107 and Dist A-1, and that the ports are trunking by using the following commands:

Commands

- **show interfaces** *interface* **trunk**
- **show trunk**

Step 3 Verify that dynamic trunking is configured between SH1-108 and Dist A-1, and that the ports are trunking by using the following commands:

Commands

- **show interfaces** *interface* **trunk**
- **show trunk**

Step 4 Shut down interface g4/1 on SH1-107 (static trunk) and verify that it is down by using the following commands:

Commands

- **shutdown**
- **show interfaces**
- **show interfaces** *interface* **trunk**

Step 5 Bring up interface g4/1 on SH1-107 and verify that the trunk is reestablished by using the following commands:

Commands

- **no shutdown**
- **show interfaces**
- **show interfaces** *interface* **trunk**

Step 6 Shut down interface g4/1 on SH1-108 (dynamic trunk) and verify that it is down by using the following commands:

Commands

- **shutdown**
- **show interfaces**
- **show interfaces** *interface* **trunk**

Step 7 Bring up interface g4/1 on SH1-108 and verify that the trunk is reestablished by using the following commands:

Commands

- **no shutdown**
- **show interfaces**
- **show interfaces** *interface* **trunk**

Step 8 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results for devices SH1-107 and SH1-108.

Expected Results

We expect basic trunking functionality to work properly and perform correctly in failure and recovery scenarios.

Results

Table 12 shows the failure and recovery test results.

Table 12 Failure and Recovery Test Results

Tests	Results
Failure and Recovery	Pass

Port Aggregation Protocol (Channeling)

The port aggregation protocol (PAgP) facilitates the automatic creation of EtherChannels by exchanging packets between Ethernet ports. PAgP packets are exchanged only between ports in **auto** and **desirable** modes. Ports configured in **on** or **off** mode do not exchange PAgP packets. The protocol learns the capabilities of port groups dynamically and informs the other ports. Once PAgP identifies correctly matched EtherChannel links, it groups the ports into an EtherChannel. The EtherChannel is then added to the spanning tree as a single bridge port.

EtherChannel includes four user-configurable modes: **on**, **off**, **auto**, and **desirable**. Only **auto** and **desirable** are PAgP modes. The auto and desirable modes can be modified with the **silent** and **non-silent** keywords. By default, ports are in **auto silent** mode.

An EtherChannel distributes frames across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel.

EtherChannel frame distribution is based on a Cisco-proprietary hashing algorithm. The algorithm is deterministic; given the same addresses and session information, you always hash to the same port in the channel, preventing out-of-order packet delivery.

The following tests were performed:

- Basic Layer 2 Channeling Configuration, page 25
- Basic Layer 3 Channeling Configuration, page 26
- Layer 2 and Layer 3 EtherChannel Load Balance, page 28
- Gigabit Ethernet Module Reset, page 29

Basic Layer 2 Channeling Configuration

This test verified the basic aspects of Layer 2 PAgP configuration to verify that the basic functionality worked correctly. For this test, a set of links between SH1-109 and Dista-1 was given a configuration for static channeling. Dynamic channeling was tested in the bundling of links between SH1-110 and Dista-2.

Test Plan

The procedure used to perform the Basic Layer 2 Channeling Configuration test follows.

Step 1 Verify the configuration of static channeling between SH1-109 and Dist A-1 by using the following commands:

Note In Native IOS, a port-channel interface is created, and then individual interfaces are joined to that port-channel interface. The configuration of those individual interfaces determines whether the channel is static or dynamic.

Commands

- **show running-config interfaces** *interface*

Step 2 Verify the configuration of dynamic channeling between SH1-110 and Dist A-1 by using the following commands:

Commands

- **show running-config interfaces** *interface*

Step 3 Verify that all ports are bundled that are supposed to be in each channel, and that channels are working by using the following commands:

Commands

- **show interfaces** *interface* **status**
- **show interfaces** *interface* **etherchannel**

- **show interfaces** *interface*

Expected Results

We expect that EtherChannels are created and frames exchange across Layer 2 links properly.

Results

Table 13 shows the basic Layer 2 channeling test results.

Table 13 Basic Layer 2 Channeling Test Results

Tests	Results
Basic Layer 2 Channeling Configuration	Pass

Basic Layer 3 Channeling Configuration

This test verified that Layer 3 port-channel functionality was tested. This test verified static and dynamic channel configurations for each of the following combinations: dCEF::dCEF channels, non-dCEF::non-dCEF channels, and mixed::mixed (involving both non-dCEF and dCEF modules). Table 14 maps each of the six steps to the combination covered.

Table 14 Basic Layer 3 Channeling Configuration Matrix

Ste p	dCEF	Channelin g
1	Yes	Static
2	Yes	Dynamic
3	Mixed	Static
4	Mixed	Dynamic
5	No	Static
6	No	Dynamic

Test Plan

The procedure used to perform the basic Layer 3 channeling configuration test follows.

Step 1 The port-channel (4 ports) between SH1-103 (Po16) and SH1-99 (Po16) resides on dCEF-only modules. Verify that it is configured for static channeling and that it is functioning correctly by using the following commands:

Commands

- **show running-config interfaces** *interface*
- **show interfaces** *interface* **status**
- **show interfaces** *interface* **etherchannel**
- **show module**

Step 2 The port-channel (4 ports) between SH1-104 (Po117) and SH1-99 (Po17) resides on dCEF-only modules. Verify that it is configured for dynamic channeling and that it is functioning correctly by using the following commands:

Commands

- **show running-config interfaces** *interface*
- **show interfaces** *interface* **status**
- **show interfaces** *interface* **etherchannel**
- **show module**

Step 3 The port-channel (4 ports) between SH1-103 (Po70) and SH1-109 (Po70) resides on mixed modules. Verify that it is configured for static channeling and that it is functioning correctly by using the following commands:

Commands

- **show running-config interfaces** *interface*
- **show interfaces** *interface* **status**
- **show interfaces** *interface* **etherchannel**
- **show module**

Step 4 The port-channel (4 ports) between SH1-103 (Po71) and SH1-110 (Po71) resides on mixed modules. Verify that it is configured for dynamic channeling and that it is functioning correctly by using the following commands:

Commands

- **show running-config interfaces** *interface*
- **show interfaces** *interface* **status**
- **show interfaces** *interface* **etherchannel**
- **show module**

Step 5 The port-channel (4 ports) between SH1-103 (Po68) and SH1-107 (Po68) resides on non-dCEF modules. Verify that it is configured for static channeling and that it is functioning correctly by using the following commands:

Commands

- **show running-config interfaces** *interface*
- **show interfaces** *interface* **status**
- **show interfaces** *interface* **etherchannel**
- **show module**

Step 6 The port-channel (4 ports) between SH1-104 (Po68) and SH1-107 (Po168) resides on non-dCEF modules. Verify that it is configured for dynamic channeling and that it is functioning correctly by using the following commands:

Commands

- **show running-config interfaces** *interface*
- **show interfaces** *interface* **status**
- **show interfaces** *interface* **etherchannel**

- **show module**

Expected Results

We expect that EtherChannels are created and frames exchange across Layer 3 links properly.

Results

Table 15 shows the basic Layer 3 channeling test results.

Table 15 Basic Layer 3 Channeling Test Results

Tests	Results
Basic Layer 3 Channeling Configuration	Pass

Layer 2 and Layer 3 EtherChannel Load Balance

This test verified that load distribution took place across the individual interfaces in an EtherChannel. An IXIA traffic stream was generated, sending traffic from 20 emulated sources to a single destination. This traffic was sent from one side of the SH1 network to the other, passing through several GECs along the way. Along each hop in the path to the destination, load distribution was verified by examining the traffic statistics on individual interfaces. All traffic sent from the multiple sources was received on the other end of the network, though balanced across many interfaces. Traffic was forwarded from sources to destination via hardware shortcuts.

Traffic (10 million packets) was sent at a rate of 25,000 pps from Dista-1 to Dista-2. Devices SH1-107 and SH1-108 used supervisor 1 engines, and devices SH1-109 and SH1-110 used supervisor 2 engines. Because traffic was forwarded through all devices (except SH1-107), coverage of both Sup1 and the Sup2 was implied. Both Layer 2 and Layer 3 EtherChannels were tested. The Layer 2 channels in this network were those between SH1-107, SH1-108, and Dista-1, and between SH1-109, SH1-110, and Dista-2.

Test Plan

The procedure used to perform the Layer 2 and Layer 3 EtherChannel load balance test follows.

Step 1 Enter the **show memory summary** command to take an initial memory snapshot of devices SH1-103, SH1-104, SH1-107, SH1-108, SH1-109, and SH1-110. Final memory and CPU utilization results are provided in the last step of this procedure.

Step 2 Clear the counters on all of the devices listed in Step 1, and on Dista-1 and Dista-2 to verify an accurate packet count for the purposes of testing. Using the following command:

Commands

- **clear counters**

Step 3 Begin the IXIA traffic stream from Dista-1 (20 IP addresses) to Dista-2 (1 IP address). Configure the stream so that only 10 million packets are sent.

Step 4 Once the stream has finished its transmission, take account of all unicast packets into and out of each device in the path, verifying that the traffic was load-balanced across all ports and interfaces within the EtherChannels:

Commands

- **show ma**
- **show interfaces counters module** *module*

Note The counters will not always equal exactly 10,000,000 due to the presence of management traffic on some interfaces. To determine which interface of a channel to send the traffic out, the software uses an algorithm that considers such things as destination and source IP addresses. Given a destination and source IP pair, the switch will determine which interface should be used for a particular packet. Some individual physical interfaces may show an outgoing counter of "0" because only 20 source IP addresses are being used, and not all physical interfaces will be selected by the algorithm and pass traffic.

Step 5 Verify that the unicast traffic was forwarded in hardware in the Layer 3 switches, or those devices listed in Step 1:

Commands

- **show mls ip**

Note For the purposes of this test, we are not concerned with exactly how many packets were hardware-switched along the traffic path, but rather only that they were.

Step 6 Verify that all traffic that was sent was received.

Note Run the TCL test script to initiate the IXIA traffic stream and to gather the packets received data.

Step 7 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results for all devices listed in Step 1.

Expected Results

We expect load distribution to take place across the individual interfaces in the EtherChannel.

Results

Table 16 shows the Layer 2 and Layer 3 EtherChannel load balance test results.

Table 16 Layer 2 and Layer 3 EtherChannel Load Balance Test Results

Tests	Results
Layer 2 and Layer 3 EtherChannel Load Balance	Pass

Gigabit Ethernet Module Reset

This test verified the ability of Layer 2 and Layer 3 EtherChannels to handle module resets and failures. The supervisor 1 and supervisor 2 were both tested. With IXIA traffic (configured for 20 source IP addresses and one destination IP address) sent from Dista-1 to Dista-2, two modules were reset, individually, on device SH1-108 (Sup1). One of the modules was involved in a Layer 2 GEC and the other in a Layer 3 GEC. For Sup2 coverage, a single module was reset on SH1-110. This module has interfaces involved in both Layer 2 and Layer 3 GECs.

Test Plan

The procedure used to perform the Gigabit Ethernet Module Reset test follows.

Step 1 Enter the **show memory summary** command to take an initial memory snapshot of devices SH1-108 and SH1-110. Final memory and CPU utilization results are provided in the last step of this procedure.

Step 2 Begin an IXIA traffic stream, sourcing from 20 IP addresses, on Dista-1, destined for a single IP address connected to Dista-2. Verify that some traffic is going through channel interfaces port-channel 10 and port-channel 69 of SH1-108, and port-channel 20 and port-channel 71 of SH1-110:

Commands

- **show interfaces** *interface* **counters**

Step 3 Reset module 4 of SH1-108. This module has interfaces (g4/1-g4/4) that make up a Layer 2 channel. Verify that the channel re-forms correctly after the module comes back online:

Commands

- **show interfaces** *interface* **status**
- **show interface**s *interface* **etherchannel**
- **hw-module module** *module* **reset**

Step 4 Reset module 3 of SH1-108. This module has interfaces involved in two separate Layer 3 GECs (g3/1-g3/4 of port-channel 69 and g3/5-g3/6 of port-channel 169). Verify that the channels re-form correctly after the module comes back online:

Commands

- **show interfaces** *interface* **status**
- **show interface**s *interface* **etherchannel**
- **hw-module module** *module* **reset**

Step 5 Reset module 7 of SH1-110. This module has interfaces (g7/3 and g7/5) that are involved in an Layer 2 channel (port-channel 20). This module also has an interface (g7/1) that is involved in a Layer 3 GEC (port-channel 71). Verify that the channel re-forms correctly after the module comes back online:

Commands

- **show interfaces** *interface* **status**
- **show interface**s *interface* **etherchannel**
- **hw-module module** *module* **reset**

Step 6 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results for devices SH1-108 and SH1-110.

Expected Results

We expect the GEC and GEC ports to work properly if the Gigabit Ethernet module gets reset.

Results

Table 17 shows the Gigabit Ethernet module reset test results.

Table 17 Gigabit Ethernet Module Reset Test Results

Tests	Results
Gigabit Ethernet Module Reset	Pass

VLAN Trunking Protocol

VLAN Trunking Protocol (VTP) is a Layer 2 messaging protocol that maintains VLAN configuration consistency by managing the addition, deletion, and renaming of VLANs on a network-wide basis. VTP minimizes misconfigurations and configuration inconsistencies that can result in a number of problems, such as duplicate VLAN names, incorrect VLAN-type specifications, and security violations.

You can use VTP to manage VLANs 1 to 1005 in your network. (Note that VTP does not support VLANs 1025 to 4094.) With VTP, you can make configuration changes centrally on one switch and have those changes automatically communicated to all the other switches in the network.

The following test was performed:

- Basic VLAN Trunking Protocol Configuration, page 31

Basic VLAN Trunking Protocol Configuration

This test configured VLANs and verified normal behavior in various combinations of VTP client, server, and transparent modes between devices SH1-107 and Dista-1.

Test Plan

The procedure used to perform the basic VLAN trunking protocol configuration test follows.

Step 1 Enter the **show memory summary** command to take an initial memory snapshot of device SH1-107. Final memory and CPU utilization results are provided in the last step of this procedure.

Step 2 Verify that SH1-107 and Dista-1 are in VTP transparent mode by using the following commands:

Commands

- **show vtp status**
- **show vtp domain**

Step 3 Verify that a trunk link is configured between SH1-107 (g4/1, g4/3) and Dista-1 (1/1, 2/1) by using the following commands:

Commands

- **show interfaces** *interface* **trunk**
- **show trunk** *mod/port*

Step 4 Run the TCL test script which will complete the remainder of the steps.

Step 5 Check the output of the script to verify that both cases passed.

Step 6 With both SH1-107 and Dista-1 in transparent mode, configure VLAN 200 on SH1-107. Verify that VLAN 200 is created on SH1-107 but not on Dista-1.

Step 7 Configure SH1-107 for VTP server mode (Dista-1 still in transparent mode) and add VLAN 201 to the VLAN database of SH1-107. Verify that VLAN 201 is created on SH1-107 but not on Dista-1.

Step 8 Configure Dista-1 for VTP Client mode (SH1-107 is the server) and add VLAN 202 to the VLAN database on SH1-107. Verify that VLAN 202 was added to both SH1-107 and Dista-1.

Step 9 Configure Dista-1 for VTP Server mode (SH1-107 is the server) and add VLAN 203 to the VLAN database on SH1-107. Verify that VLAN 203 was added to both SH1-107 and Dista-1.

Step 10 With SH1-107 and Dista-1 both in server mode, configure VTP Version 2 on Dista-1 (VTP V2 is disabled on SH1-107). Configure VLAN 204 on SH1-107. Verify that VLAN 204 is created on both SH1-107 and Dista-1.

Step 11 With the same VTP configurations as in Step 10, create VLAN 205 on Dista-1. Verify that VLAN 205 is created on both SH1-107 and Dista-1.

Step 12 With SH1-107 in server mode, configure Dista-1 for VTP transparent mode and SH1-108 for VTP client mode. Configure VLAN 206 on SH1-107. Verify that VLAN 206 is not created on Dista-1, but is created on SH1-108.

Step 13 Return the VTP modes of SH1-107, SH1-108, and Dista-1 to transparent, and remove VLANs 200 to 206 from each of those three devices.

Step 14 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results for device SH1-107.

Expected Results

We expect that VLAN configuration consistency through management of the addition, deletion, and renaming of VLANs on a network-wide basis will be maintained.

Results

Table 18 shows the basic VLAN trunking protocol configuration test results.

Table 18 Basic VLAN Trunking Protocol Configuration Test Results

Tests	Results
Basic VLAN Trunking Protocol Configuration	Pass

Hardware Forwarding Features

Hardware forwarding testing for Safe Harbor involves these features:

- IP Unicast, page 33
- IP Multicast, page 36

IP Unicast

The Internet Protocol (IP) is a packet-based protocol used to exchange data over computer networks. IP handles addressing, fragmentation, reassembly, and protocol demultiplexing. It is the foundation on which all other IP protocols (collectively referred to as the IP Protocol suite) are built. A network-layer protocol, IP contains addressing and control information that allows data packets to be routed.

The Transmission Control Protocol (TCP) is built upon the IP layer. TCP is a connection-oriented protocol that specifies the format of data and acknowledgments used in the transfer of data. TCP also specifies the procedures that the networking devices use to verify that the data arrives correctly. TCP allows multiple applications on a system to communicate concurrently because it handles all demultiplexing of the incoming traffic among the application programs.

The following tests were performed:

- Layer 2 Gigabit EtherChannel Failover, page 33
- Layer 3 Gigabit EtherChannel Failover, page 35

Layer 2 Gigabit EtherChannel Failover

This test sent traffic over a Layer 2 GEC, and the links in that GEC were failed one at a time until the traffic was forced to find a different path to the destination. There were two separate test procedures, one for supervisor 1 coverage, and another for supervisor 2 coverage. The device under test (DUT) for the first test was SH1-108, which had a Layer 2 GEC connecting it with Dista-1. The DUT for the second test was SH1-110, which had a Layer 2 GEC connecting it with Dista-2.

Test Plan

The procedure used to perform the IP unicast Layer 2 Gigabit EtherChannel failover test follows.

Step 1 Enter the **show memory summary** command to take an initial memory snapshot of both DUTs. Final memory results are provided at the end of both procedures, and a table showing CPU utilization for the period of both procedures is also provided.

Step 2 For supervisor 1, verify that the GEC (interface port-channel 10) between the DUT and Dista-1 is up and the appropriate physical interfaces are active within that channel by using the following commands:

Commands

- **show interfaces** *interface* **status**
- **show interfaces** *interface* **etherchannel**

Step 3 Start an IXIA traffic stream emulating 20 IP sources on Dista-1 sending to a single destination on Dista-2. Verify that the first hop for this traffic is the DUT (HSRP active router):

Commands

- **show mac**

Step 4 Shut down one interface of the GEC on the DUT and measure any traffic loss.

Step 5 Bring up the interface shut down in Step 4.

Step 6 Fail the entire port-channel from the SH1-108 side. Measure any traffic loss.

Step 7 Bring all the interfaces that have been shut down back up, so that port-channel 10 is up/up and all traffic flows through the DUT again.

Step 8 For Supervisor 2, verify that the GEC (interface port-channel 20) between the DUT (now SH1-110) and Dista-2 is up and the appropriate physical interfaces are active within that channel:

Commands

- **show interfaces** *interface* **status**
- **show interfaces** *interface* **etherchannel**

Step 9 Start an IXIA traffic stream emulating 20 IP sources on Dista-2 sending to a single destination on Dista-1. Verify that the first hop for this traffic is the DUT (HSRP active router):

Commands

- **show mac**

Step 10 Shut down one interface of the GEC on the DUT and measure any traffic loss.

Step 11 Bring up the interface shut down in Step 4.

Step 12 Fail the entire port-channel from the SH1-110 side. Measure any traffic loss.

Step 13 Bring all the interfaces that have been shut down back up, so that port-channel 20 is up/up and all traffic flows through the DUT again.

Step 14 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results.

Expected Results

We expect that traffic interruption will be acceptable when an L2 GEC is failed and that the traffic will resume on its original path when the L2 GEC is brought back up.

Results

Table 19 shows the Layer 2 Gigabit EtherChannel failover test results.

Table 19 Layer 2 Gigabit EtherChannel Failover Test Results

Tests	Results
Layer 2 Gigabit EtherChannel Failover	Pass

Layer 3 Gigabit EtherChannel Failover

This test verified the ability of a system to cope with a Layer 3 GEC failure, with a minimum amount of traffic loss. IXIA traffic was directed through the Layer 3 GEC, and the Layer 3 GEC was forced to fail, first a single interface, and then the entire channel. Because the GEC between SH1-108 and SH1-103 was tested, supervisor 1 (SH1-108) and the supervisor 2 (SH1-103) were included in the same test.

Test Plan

The procedure used to perform the IP Unicast Layer 3 Gigabit EtherChannel Failover test follows.

Step 1 Enter the **show memory summary** command to take an initial memory snapshot of devices SH1-103 and SH1-108. Final memory and CPU utilization results are provided in the last step of this procedure.

Step 2 Verify that the GEC (interface Port-Channel 69) between the DUT (SH1-108) and SH1-103 is up and the appropriate physical interfaces are active within that channel:

Commands

- **show interfaces** *interface* **status**
- **show interfaces** *interface* **etherchannel**

Step 3 Configure the **ip ospf cost 100** command on SH1-104 port-channel 69 and SH1-108 port-channel 169 to ensure that all traffic forwarded by SH1-108 is forwarded out port-channel 69, or the channel under test:

Commands

- **ip ospf cost 100**

Step 4 Start an IXIA traffic stream emulating 20 IP sources on Dista-1 sending to a single destination on Dista-2. Verify that this traffic is being forwarded through the Layer 3 port channel under test (Po69) and not through port-channel 169:

Commands

- **show interfaces** *interface* **counters**

Step 5 Verify that unicast traffic is being load-balanced across all four links in port-channel 69 on SH1-108:

Commands

- **show interface counters module** *module*

Step 6 Shut down one interface of the GEC on the DUT and measure any traffic loss.

Step 7 Bring up the link that was shut down in Step 6.

Step 8 Shut down the entire interface port-channel 69 on SH1-108. Verify that all traffic now goes from SH1-108 to SH1-104 (out SH1-108 port-channel 169) and measure traffic loss:

Commands

- **show interfaces** *interface* **counters**

Step 9 Bring up interface port-channel 69 and verify that traffic flows through it again, and ceases to flow through port-channel 169:

Commands

- **show interface counters module** *module*

Step 10 Remove the **ip ospf cost 100** statements put on the interfaces in Step 3:

Commands

- **no ip ospf cost 100**

Step 11 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results for devices SH1-103 and SH1-108.

Expected Results

We expect that traffic interruption will be acceptable when an L2 GEC is failed and that the traffic will resume on its original path when the L2 GEC is brought back up.

Results

Table 20 shows the Layer 3 Gigabit EtherChannel failover test results.

Table 20 Layer 3 Gigabit EtherChannel Failover Test Results

Tests	Results
Layer 3 Gigabit EtherChannel Failover	Pass

IP Multicast

Traditional IP communication allows a host to send packets to a single host (*unicast transmission*) or to all hosts (*broadcast transmission*). IP multicast provides a third scheme, allowing a host to send packets to a subset of all hosts (*group transmission*). These hosts are known as group members.

Packets delivered to group members are identified by a single multicast group address. Multicast packets are delivered to a group using best-effort reliability, just like IP unicast packets.

The multicast environment consists of senders and receivers. Any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group receive the message.

A multicast address is chosen for the receivers in a multicast group. Senders use that address as the destination address of a datagram to reach all members of the group.

Membership in a multicast group is dynamic; hosts can join and leave at any time. There is no restriction on the location or number of members in a multicast group. A host can be a member of more than one multicast group at a time.

Note On PFC1, the (*, G) can only be software-switched. If the **ip pim spt-threshold infinity** command is used on PFC1, there might be high CPU usage and multicast packets might be lost under heavy traffic.

The following tests were performed:

- Basic Multicast and Multicast Source Discovery Protocol, page 37
- Basic IGMP and CGMP Functionality, page 38
- Core Multicast Source Discovery Protocol, page 40
- Non-Reverse Path Forwarding Rate Limiting and Multicast Stub, page 41
- Gigabit EtherChannel Failover: Non-dCEF GEC Failover, page 45

Basic Multicast and Multicast Source Discovery Protocol

This test verified multicast and Multicast Source Discovery Protocols (MSDP) basic functionality, including the creation of hardware shortcuts for multicast entries. In this test, IXIA was used to send traffic streams to five multicast groups, 239.255.127.100 through 239.255.127.104. The traffic entered the network at an interface on SH1-108, and exits the network at three points, one on SH1-108 and two on Dista-1. One million packets were sent at a rate of 10,000 pps. IP PIM was configured for sparse-mode and the **ip pim spt-threshold infinity** command was set.

Test Plan

The procedure used to perform the basic multicast and Multicast Source Discovery Protocol test follows.

Step 1 Enter the **show memory summary** command to take an initial memory snapshot of device SH1-108, and use the **show processes cpu** command to begin tracking CPU utilization by using the **show processes cpu** command. Final memory and CPU utilization results are provided in the last step of this procedure.

Step 2 Start the IGMP join traffic for the five multicast groups on IXIA ports 8/1 (Dista-1 4/1), 8/2 (Dista-1 4/6), and 7/1 (SH1-108 g3/8).

Step 3 Begin the IXIA stream (IXIA port 7/2).

Step 4 Verify that the proper (S, G) entries are created on SH1-108 by using the following command:

Commands

- **show ip mroute**

Step 5 Verify that the multicast traffic is being hardware-switched by using the following command:

Commands

- **show mls ip multicast**

Step 6 Verify that all multicast traffic that is sent is also received by the intended interfaces and ports.

Step 7 Verify that 100 percent of the unicast background traffic running during the test is received on the appropriate interfaces.

Step 8 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results for device SH1-108.

Expected Results

We expect multicast and MSDP basic functionality, including the creation of hardware shortcuts for multicast entries, to work properly.

Results

Table 21 shows the basic multicast and MSDP test results.

Table 21 Basic Multicast and MSDP Test Results

Tests	Results
Basic Multicast and Multicast Source Discovery Protocol	Pass

Basic IGMP and CGMP Functionality

Internet Group Management Protocol (IGMP) software components run on both the Cisco router and the switch. An IGMP-capable IP multicast router sees all IGMP packets and can inform the switch when specific hosts join or leave IP multicast groups.

When the IGMP-capable router receives an IGMP control packet, it creates an IGMP packet that contains the request type (either join or leave), the multicast group address, and the MAC address of the host. The router sends the packet to a well-known address to which all switches listen. When a switch receives the packet, the supervisor engine interprets the packet and modifies the forwarding table automatically. Cisco Group Management Protocol (CGMP) should seamlessly integrate with IGMP and performs the same function.

This test verified IGMP and CGMP basic functionality, including the IGMP and CGMP status, and verified that no multicast traffic was flooded into ports that do not have a multicast client. IGMP joins were sourced by IXIA-8/2 and sent to Dista-1 4/6 (VLAN 16) for multicast groups 239.255.127.100 through 239.255.127.104. Traffic for these groups was sourced by IXIA 8/1 and sent to Dista-1 4/1 (VLAN 11). The traffic was 1.5 million packets sent at a rate of 10,000 pps. Success of the test was measured by 100 percent of traffic being received by the single receiver interface, no traffic going to unintended interfaces, and no sustained or abnormal impact to the memory of CPU utilization.

Test Plan

The procedure used to perform the basic IGMP and CGMP functionality test follows.

Step 1 Verify that SH1-107 and SH1-108 are running Multicast Source Discovery Protocol (MSDP) Anycast and peering to each other. If they are peering to one another, each will have an MSDP statement pointing to the other's loop back address:

Commands

- **show running-config | include msdp**
- **show running-config interfaces lo0**

Step 2 Start the IGMP join messages on IXIA port 8/2 (Dista-1 4/6).

Step 3 Enter the **show memory summary** command to take an initial memory snapshot of device SH1-108. Final memory and CPU utilization results are provided in the last step of this procedure.

Step 4 Begin sending multicast traffic using IXIA port 8/1 into Dista-1 port 4/1.

Step 5 On Dist A-1, analyze output from the following commands to determine IGMP status:

Commands

- **show multicast router**
- **show multicast group**
- **show igmp statistics 11**
- **show igmp statistics 16**

Step 6 Verify that the standby rendezvous point, SH1-107, has the necessary entries in its mroute table (a null outgoing interface list) and the necessary MMLS entries:

Commands

- **show ip mroute**
- **show mls ip multicast**

Step 7 Verify that the primary rendezvous point, SH1-108, has the necessary entries in its mroute table (including the correct outgoing interface list) and the necessary MMLS entries:

Commands

- **show ip mroute**
- **show mls ip multicast**

Step 8 Verify that 100 percent of packets sent were received by the appropriate port and not by the unintended interfaces.

Step 9 Verify that 100 percent of the unicast background traffic running during the test is received on the appropriate interfaces.

Step 10 Use the **show memory summary** and **show processes cpu** commands to verify that these processes caused no major impact on the memory of CPU utilization.

Expected Results

We expect IGMP and CGMP basic functionality, including the IGMP and CGMP status to work properly, and that no multicast traffic is flooded into ports that do not have a multicast client.

Results

Table 22 shows the basic IGMP and CGMP functionality test results.

Table 22 Basic IGMP and CGMP Functionality Test Results

Tests	Results
Basic IGMP and CGMP Functionality	Pass

Core Multicast Source Discovery Protocol

This test verified the functionality of Core MSDP at the core level. SH1-103 and SH1-104 were configured as MSDP peers for multicast groups 239.255.127.100 through 239.255.127.104. Multicast data traffic for these five groups was sent from IXIA connected to Dista-2 to two receiver ports, one on Dista-1 and one on Dista-2. This test was successful if the traffic used the path through the active MSDP peer (SH1-104) to get to the receiver.

SH1-107 and SH1-108 were configured with the **ip pim spt-threshold infinity** command to verify that the (*, G) path was used, forcing the traffic to go through the active MSDP router. Multicast traffic was sent at a rate of 10,000 pps. All packets were received by each of the two ports. The unicast background traffic must be fully received. There were no negative impact on memory or CPU utilization, on any of the Layer 3 devices.

Test Plan

The procedure used to perform the core MSDP test follows.

Step 1 Verify that SH1-103 and SH1-104 are running MSDP Anycast for multicast groups under test. Verify that SH1-104 is the active MSDP router. Verify that the **ip pim spt-threshold infinity** command is configured on SH1-107 and SH1-108:

Commands

- **show running-config | include msdp**
- **show running-config | include spt**

Step 2 Verify that loopback 1 interface on SH1-103 is configured with the **ip ospf cost 10** command, so that SH1-104 is chosen as preferred rendezvous point:

Commands

- **show running-config interfaces lo1**

Step 3 Begin the IGMP joins on IXIA ports10/2 and 8/1.

Step 4 Enter the **show memory summary** command to take an initial memory snapshot of devices SH1-109, SH1-110, SH1-103, SH1-104, SH1-107, and SH1-108. Final memory and CPU utilization results are provided in the last step of this procedure.

Step 5 Begin the IXIA traffic streams for multicast groups 239.255.127.100 to 239.255.127.104 (IXIA port 10/1).

Step 6 Verify that traffic is using SH1-104 as the MSDP router, and not SH1-103 by using the following commands:

Commands

- **show ip mroute**
- **show mac**

Step 7 Verify that all multicast packets sent were received on the appropriate ports.

Step 8 Verify that 100 percent of the unicast background traffic running during the test is received on the appropriate interfaces.

Step 9 Use the **show memory summary** and **show processes cpu** commands to verify that no significant or lasting impact on the memory of CPU utilization occurred for devices.

Expected Results

We expect functionality of core MSDP at the core level to work properly.

Results

Table 23 shows the core MSDP test results.

Table 23 Core MSDP Test Results

Tests	Results
Core Multicast Source Discovery Protocol	Pass

Non-Reverse Path Forwarding Rate Limiting and Multicast Stub

Stub multicast routing allows you to configure remote and stub routers as IGMP proxy agents. Instead of fully participating in PIM, these stub routers simply forward IGMP messages from the hosts to the upstream multicast router.

This test verified that the multicast stub command worked and that non-Reverse Path Forwarding (RPF) rate limiting functionality worked on Policy Feature Cards (PFC1 and PFC2), and dCEF, including a hardware shortcut.

PFC1, PFC2, and the DFCs support ACL-based filtering of RPF failures for sparse mode stub networks. When you enable the ACL-based method of filtering Reverse Path Forwarding failures by entering the **mls ip multicast stub** command on the redundant router, the following ACLs automatically download to the PFC and are applied to the interface you specify:

```
access-list 100 permit ip A.B.C.0 0.0.0.255 any
access-list 100 permit ip A.B.D.0 0.0.0.255 any
access-list 100 permit ip any 224.0.0.0 0.0.0.255
access-list 100 permit ip any 224.0.1.0 0.0.0.255
access-list 100 deny ip any 224.0.0.0 15.255.255.255
```

The ACLs filter Reverse Path Forwarding failures and drop them in hardware so that they are not forwarded to the router. This test has two parts. The first part tests multicast stub functionality on PFC1 (on SH1-107). The second part tests multicast stub functionality on the PFC2 (on SH1-109). In each, multicast traffic was sent from one side of the network (from Dista-1 and Dista-2) to the other side. As a component of each part, the non-Reverse Path Forwarding rate limiting functionality was observed.

Test Plan

The procedure used to perform the non-TFP rate limiting and multicast stub test follows.

Supervisor 1

In this part of the test, SH1-107 was configured to be the multicast stub router. Five groups of multicast traffic (groups 239.255.127.100 to 239.255.127.104) were sent from Dista-2. Joins for these five groups were sent from Dista-1. The IP address of the remote interface connecting SH1-107 to SH1-104 was configured as the IGMP helper address. The IGMP join packets that were received by SH1-107 were forwarded via this address statement to SH1-104. All five groups should appear on SH1-104, when the **show ip igmp groups** command is used.

Coupled with the **mls ip multicast stub** command on SH1-107 is the **ip pim neighbor-filter** *access-list* command configured on SH1-104. This configuration blocked PIM neighbor updates coming from SH1-107, and made SH1-107 truly transparent to the rest of multicast functionality.

Step 1 Use a SNMP monitoring utility to begin monitoring memory and CPU utilization for devices SH1-107 and SH1-104.

Step 2 Configure SH1-107 VLAN 11 (source of IGMP joins) as a multicast stub by using the following command:

Commands

- **mls ip multicast stub**

Step 3 Configure the IP helper address for SH1-107, the address on SH1-104 to which SH1-107 will forward the IGMP join packets:

Commands

- **ip igmp helper-address** *IP_address*

Step 4 Configure SH1-104 to filter out IP PIM neighbor updates coming from SH1-107. Verify that access list 6 exists and points to the correct IP address of the blocked PIM updates:

Commands

- **ip pim neighbor-filter 6**
- **show access-list 6**

Step 5 USe the **ip pim spt-threshold infinity** command on SH1-107 and SH1-108 (Sup 12) to ensure that the multicast routing state (*, G) is used, and not (S, G):

Commands

- **ip pim spt-threshold infinity**

Step 6 Enter the **ip ospf cost 10** command on loopback interface 1 of SH1-103 to ensure that SH1-104 is selected as the primary PIM rendezvous point:

Commands

- **ip ospf cost 10**

Step 7 Begin the IGMP join messages for the five multicast groups on Dista-1 (IXIA port 8/1).

Step 8 Begin the IXIA traffic stream for the five multicast groups on Dista-2 (IXIA port 10/1).

Step 9 Verify that the IGMP join messages have been forwarded from SH1-107 to SH1-104 and that SH1-104 does not register SH1-107 as an IP PIM neighbor:

Commands

- **show ip igmp groups**
- **show ip pim neighbor**

Step 10 Use the **show mls ip multicast summary** command to verify that the traffic flowing through the stub router is being hardware-switched:

Commands

- **show mls ip multicast summary**

Step 11 Determine the path of traffic to the receiver, starting at SH1-104 (the rendezvous point). Traffic is expected to be sent out both interfaces of port-channel 68 and port-channel 69 from SH1-104, to SH1-107 and SH1-108, respectively, because there are IGMP joins coming from port-channel 68 (forwarded from SH1-107) and PIM requests coming from port-channel 69 (from SH1-108). The traffic should be forwarded, beyond that, only by SH1-108, because that is the PIM-DR for that segment. SH1-107 should not be forwarding traffic.

Commands

- **show ip mroute**
- **show interfaces** *interface* **counters**
- **show ip pim neighbor**

Step 12 Verify that 100 percent of the multicast traffic that was sent is received on the appropriate port (zero packet loss).

Step 13 Verify that 100 opercent of the unicast background traffic running during the test is received on the appropriate interfaces.

Step 14 Verify that no major or sustained impact to the memory or CPU occurred for test period.

Step 15 Negate the commands configured in Step 2 through Step 4.

Supervisor 2

In this part of the test, SH1-109 was configured to be the multicast stub router. Five groups of multicast traffic (groups 239.255.129.100 through 239.255.129.104) were sent from Dista-1. Joins for these five groups were sent from Dista-2. The IP address of the remote interface connecting SH1-109 to SH1-104 was configured as the IGMP helper address. The IGMP join packets that are received by SH1-109 were forwarded via this address statement to SH1-104. All five groups should appear on SH1-104, when the **show ip igmp groups** command is used.

Coupled with the **mls ip multicast stub** command on SH1-109 is the **ip pim neighbor-filter** *access-list* command configured on SH1-104. This configuration blocked PIM neighbor updates coming from SH1-109, and made SH1-109 truly transparent to the rest of multicast functionality.

Step 1 Use a SNMP monitoring utility to begin monitoring memory and CPU utilization for devices SH1-109 and SH1-104.

Step 2 Configure SH1-109 VLAN 16 (source of IGMP joins) as a multicast stub by using the following commands:

Commands

- **mls ip multicast stub**

Step 3 Configure the IP helper address for SH1-109, the address on SH1-104 to which SH1-109 will forward the IGMP join packets by using the following commands:

Commands

- **ip igmp helper-address** *IP_address*

Step 4 Configure SH1-104 to filter out IP PIM neighbor updates coming from SH1-109. Verify that access list 5 exists and points to the correct IP address of the blocked PIM updates:

Commands

- **ip pim neighbor-filter 5**
- **show access-lists 5**

Step 5 Use the **ip pim spt-threshold infinity** command on SH1-107 and SH1-108 (Sup12) to ensure that the multicast routing state (*, G) is used, and not (S, G):

Commands

- **ip pim spt-threshold infinity**

Step 6 Verify that **ip ospf cost 10** is configured on the loopback interface 1 of SH1-103 to ensure that SH1-104 is selected as the primary PIM rendezvous point:

Commands

- **show running-config interfaces Lo1**

Step 7 Begin the IGMP join messages for the five multicast groups on Dista-2 (IXIA port 10/2).

Step 8 Begin the IXIA traffic stream for the five multicast groups on Dista-1 (IXIA port 8/1).

Step 9 Verify that the IGMP join messages have been forwarded from SH1-109 to SH1-104 and that SH1-104 (port-channel 70) does not register SH1-109 as an IP PIM neighbor.

Commands

- **show ip igmp groups**
- **show ip pim neighbor**

Step 10 Verify that traffic flowing through the stub router is being hardware-switched.

Commands

- **show mls ip multicast summary**

Step 11 Determine the path of traffic to the receiver, starting at SH1-104 (the rendezvous point). Traffic should be sent out both interfaces port-channel 70 and port-channel 71 from SH1-104, to SH1-109 and SH1-110, respectively, because there are IGMP joins coming from port-channel 70 (forwarded from SH1-109) and PIM requests coming from port-channel 71 (from SH1-110). The traffic should be forwarded, beyond that, only by SH1-110, because that is the PIM-DR for that segment. SH1-109 should not be forwarding traffic.

Commands

- **show ip mroute**
- **show interfaces** *interface* **counters**
- **show ip pim neighbor**

Step 12 Verify that 100 percent of the multicast traffic that was sent is received on the appropriate port (zero packet loss).

Step 13 Verify that 100 percent of the unicast background traffic running during the test is received on the appropriate interfaces.

Step 14 Verify that no major or sustained impact to the memory or CPU occurred for the test period.

Step 15 Negate the configurations of Step 2 through Step 4.

Expected Results

We expect the **multicast stub** command to work and non-RPF rate limiting functionality to work on Policy Feature Cards (PFC1 and PFC2), and dCEF, including a hardware shortcut.

Results

Table 24 shows the non-RPF rate limiting and multicast stub test results.

Table 24 Non-RFP Rate Limiting and Multicast Stub Test Results

Tests	Results
Non-Reverse Path Forwarding Rate Limiting and Multicast Stub	Pass

Gigabit EtherChannel Failover: Non-dCEF GEC Failover

This test verified multicast and MSDP functionality during a non-distributed Cisco Express Forwarding (dCEF) GEC failover.

Traffic for multicast groups 239.255.127.100 through 239.255.127.104 was sent out via IXIA from Dista-1 and received at Dista-2. Traffic was sent at a rate of 10,000 pps. The port channel between devices SH1-103 and SH1-108 was failed, first two of four ports, then all four ports.

Note Background unicast traffic remained running for this test at a rate of 25,000 pps. Results for zero-packet loss of this traffic are not given, though, because the test involved failovers and traffic was certain to be lost.

Test Plan

The procedure used to perform the Gigabit EtherChannel failover: non-dCEF GEC failover test follows.

Step 1 Begin monitoring CPU utilization for devices SH1-103, SH1-104, SH1-107, SH1-108, SH1-109, and SH1-110. Enter the **show memory summary** command to take an initial memory snapshot of SH1-103 and SH1-108.

Step 2 Verify that core routers SH1-103 and SH1-104 are running MSDP Anycast by using the following command:

Commands

- **show running-config | include msdp**

Step 3 Use the **ip pim spt-threshold infinity** command for SH1-109 and SH1-110 to ensure that the multicast routing state (*, G) is used, and not (S, G):

Commands

- **show running-config | include spt**

Step 4 Start IGMP join traffic from IXIA port 10/2 on Dista-2 4/10.

Step 5 Set SH1-103 as the primary PIM RP by using the **ip ospf cost 10** command on loopback interface 1 of SH1-104:

Commands

- **ip ospf cost 10**
- **show running-config | interfaces Lo1**

Step 6 Use IXIA to send multicast traffic to multicast groups 239.255.127.100 through 239.255.127.104 on SH1-108 VLAN 15 (IXIA port 7/2). The receiving port is on Dist A-2 VLAN 16 (IXIA port 10/2). Reconfigure the IXIA stream so that the host portion of the IP source is incremented four times, which will allow the traffic to be evenly distributed over all four links within the GEC.

Step 7 Verify that SH1-103 is the primary PIM rendezvous point for the given multicast groups and that SH1-108 is forwarding traffic to SH1-103:

Commands

- **show ip pim rp**
- **show ip route 172.31.0.127**
- **show ip mroute**

Step 8 Verify that the modules connecting SH1-103 and SH1-108 are non-dCEF modules. The ports connecting them are g8/7-g8/10 on the SH1-103 side, and g3/1-g3/4 on the SH1-108 side:

Commands

- **show module**

Step 9 Use the **test etherchannel load-balance interface port-channel 69 ip** command on the supervisor (SP, not Route Processor) to verify that each link of the GEC should pass some multicast flow. Use the interface counters to verify that traffic is actually being passed across each of the physical interfaces:

Commands

- **test etherchannel load-balance interface port-channel** *id* **ip** *source-ip*
- **show interfaces counters module 3**

Step 10 Fail two links of the 4-port GEC between SH1-103 and SH1-108, forcing some multicast traffic to move to two other links.

Step 11 Bring up the two links failed in Step 10 and verify the multicast frame loss.

Step 12 Fail all four links of the GEC between SH1-103 and SH1-108, forcing multicast traffic to go through SH1-107 to get to SH1-103. Use the **show ip mroute** and **test etherchannel load-balance** commands, and examine the interface counter to verify that traffic is redirected through SH1-107. Determine the traffic loss:

Commands

- **test etherchannel load-balance interface port-channel** *id* **ip** *source-ip*
- **show ip mroute**
- **show interfaces count module 3**

Step 13 Bring up the four links between SH1-107 and SH1-103. Use the **show ip mroute** and **test etherchannel load-balance** commands, and examine the interface counters to verify that traffic is once again sent through SH1-108. Determine the traffic loss:

Commands

- **test etherchannel load-balance interface port-channel** *id* **ip** *source-ip*
- **show ip mroute**

- **show interfaces counters module 3**

Step 14 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU tracking results.

Expected Results

We expect that multicast and MSDP functionality during a non-distributed (dCEF) GEC failover to work properly.

Results

Table 25 shows the Gigabit EtherChannel failover: non-dCEF GEC failover test results.

Table 25 Gigabit EtherChannel Failover: Non-dCEF GEC Failover Test Results

Tests	Results
Gigabit EtherChannel Failover: Non-dCEF GEC Failover	Pass

Gigabit EtherChannel Failover: Mixed GEC Failover

This test verified multicast and MSDP functionality during a dCEF and non-dCEF GEC failover. There was a four-port channel between devices SH1-103 and SH1-110. On the SH1-103 side, those four ports were spread evenly between two modules, a dCEF-capable module, and a non-dCEF module. During this test, the two ports on the non-dCEF module were disabled and enabled, one port on each dCEF and non-dCEF module were disabled and enabled, both ports on the dCEF card were disabled and enabled, and, all four ports were disabled and enabled.

Note Applicable to this test, among others, was the static topology change in which SH1-103 interfaces g7/13 and g8/13, and SH1-110 g3/1 and g7/1, used the Texas 2 GBICs (WS-X5383) to run Gigabit EtherChannel over copper.

Note Background unicast traffic remained running for this test at a rate of 25,000 pps. Results for zero-packet loss of this traffic are not given, though, because the test involves failovers and traffic is certain to be lost.

Test Plan

The procedure used to perform the Gigabit EtherChannel failover: mixed GEC failover test follows.

Step 1 Begin monitoring CPU utilization for devices SH1-103, SH1-104, SH1-107, SH1-108, SH1-109, and SH1-110. Enter the **show memory summary** command to take an initial memory snapshot of SH1-103 and SH1-110.

Step 2 Verify that core routers SH1-103 and SH1-104 are running MSDP Anycast by using the following command:

Commands

- **show running-config | include msdp**

Step 3 Use the **ip pim spt-threshold infinity** command on SH1-107 and SH1-108 to ensure that the multicast routing state (*, G) is used, and not (S, G):

Commands

- **show running-config | include spt**

Step 4 Start IGMP join traffic from Dista-1 4/6 (IXIA port 8/2).

Step 5 Use IXIA port 10/1 to send multicast traffic to multicast groups 239.255.127.100 through 239.255.127.104 on Dista-2 VLAN 11 g2/1. The receiving port was the IXIA port 8/2 on Dist A-1 VLAN 16 g4/10. Reconfigure the IXIA stream so that the host portion of the IP source was incremented four times, which will allow the traffic to be evenly distributed over all four links within the GEC.

Step 6 Set SH1-103 as the primary PIM RP by using the **ip ospf cost 10** command on loopback interface 1 of SH1-104, causing SH1-103's default cost of five to be preferred. Verify that SH1-103 was the primary PIM RP for the given multicast groups and that SH1-110 was forwarding traffic through SH1-103 using the following commands:

Commands

- **ip ospf cost 10**
- **show running-config | interfaces Lo1**
- **show ip pim rp**
- **show ip route 172.31.0.127**
- **show ip mroute**

Step 7 Verify that the modules connecting SH1-103 and SH1-108 are mixed (dCEF and non-dCEF) modules. Modules 7 and 8 are used on the SH1-103 side, and modules 3, 4, 7, and 8 are used on the SH1-110 side:

Commands

- **show module**

Step 8 Use the **test etherchannel load-balance interface port-channel 71 ip** command on the supervisor (SP, not Route Processor) of SH1-110 to verify that each link of the GEC should pass some multicast flow. Use the counters to verify that traffic is actually being passed across each of the physical interfaces:

Commands

- **test etherchannel load-balance interface port-channel** *id* **ip** *source-ip*
- **show interfaces counters module** *3|4|7|8*

Step 9 Fail interfaces g3/1 and g4/1 of SH1-110, forcing some multicast traffic to move to two other links. Verify the multicast frame loss.

Step 10 Bring up the two links failed in Step 9 and verify the multicast frame loss.

Step 11 Fail interfaces g3/1 and g7/1 of SH1-110, forcing some multicast traffic to move to two other links. Verify the multicast frame loss.

Step 12 Bring up the two links failed in Step 11 and verify the multicast frame loss.

Step 13 Fail interfaces g7/1 and g8/1 of SH1-110, forcing some multicast traffic to move to two other links. Verify the multicast frame loss.

Step 14 Bring up the two links failed in Step 13 and verify the multicast frame loss.

Step 15 Fail all four links of the GEC between SH1-103 and SH1-110, forcing multicast traffic to go through SH1-109 to get to SH1-103. Verify the multicast frame loss. Use the **show ip mroute** and **test etherchannel load-balance** commands, and examine the interface counters to verify that traffic is redirected through SH1-109. Determine the traffic loss:

Commands

- **test etherchannel load-balance interface port-channel** *id* **ip** *source-ip*
- **show ip mroute**
- **show interfaces counters module** *3|4|7|8*

Step 16 Bring up the four links between SH1-107 and SH1-103. Use the **show ip mroute** and **test etherchannel load-balance** commands, and examine the interface counters to verify that traffic is once again sent through SH1-108. Determine the traffic loss:

Commands

- **test etherchannel load-balance interface port-channel** *id* **ip** *source-ip*
- **show ip mroute**

Step 17 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU tracking results.

Expected Results

We expect multicast and MSDP functionality during a dCEF and non-dCEF GEC failover to work properly.

Results

Table 26 shows the Gigabit EtherChannel failover: mixed GEC failover test results.

Table 26 Gigabit EtherChannel Failover: Mixed GEC Failover Test Results

Tests	Results
Gigabit EtherChannel Failover: Mixed GEC Failover	Pass

Gigabit EtherChannel Failover: dCEF GEC Failover

This test verified multicast and MSDP functionality during a dCEF GEC failover. There was a four-port channel between devices SH1-104 and SH1-110. On the SH1-110 side, those four ports were spread evenly between two dCEF-enabled modules. During this test, two ports of the dCEF GEC were disabled and enabled, then all four ports were disabled and enabled.

Test Plan

The procedure used to perform the IP Multicast Gigabit EtherChannel Failover: dCEF GEC Failover test follows.

Step 1 Begin monitoring CPU utilization for devices SH1-103, SH1-104, SH1-107, SH1-108, SH1-109, and SH1-110. Enter the **show memory summary** command to take an initial memory snapshot of devices SH1-103 and SH1-110.

Step 2 Verify that core routers SH1-103 and SH1-104 are running MSDP Anycast by using the following command:

Commands

- **show running-config | include msdp**

Step 3 Use the **ip pim spt-threshold infinity** command on SH1-107 and SH1-108 to ensure that the multicast routing state (*, G) is used, and not (S, G):

Commands

- **show running-config | include spt**

Step 4 Start IGMP join traffic from IXIA port 82 to Dista-1 g4/16.

Step 5 Use IXIA port 10/1 to send multicast traffic to multicast groups 239.255.127.100 through 239.255.127.104 on Dista-2 VLAN 11 g2/1. The receiving port is IXIA port 8/2 on Dist A-1 VLAN 16 g 4/10. Reconfigure the IXIA stream so that the host portion of the IP source is incremented four times which will allow the traffic to be evenly distributed over all four links within the GEC.

Step 6 Set SH1-104 as the primary Protocol Independent Multicast rendezvous point (PIM RP) by using the **ip ospf cost 10** command on the loopback interface 1 of SH1-103.

Commands

- **ip ospf cost 10**

Step 7 Verify that SH1-104 is the primary PIM rendezvous point for the given multicast groups and that SH1-110 is forwarding traffic through SH1-104:

Commands

- **show running-config | interfaces Lo1**
- **show ip pim rp**
- **show ip route 172.31.0.127**
- **show ip mroute**

Step 8 Verify that modules of SH1-110 that connect to SH1-104 are dCEF modules. Modules 3 and 8 are used on the SH1-104 side, and modules 3 and 4 are used on the SH1-110 side:

Commands

- **show module**

Step 9 Use the **test etherchannel load-balance interface port-channel 71 ip** command on the supervisor (SP, not Route Processor) of SH1-110 to verify that each link of the GEC should pass some multicast flow. Use the counters to verify that traffic is actually being passed across each of the physical interfaces:

Commands

- **test etherchannel load-balance interface port-channel** *id* **ip** *source-ip*
- **show interfaces counters module 3 | 4**

Step 10 Fail interfaces g3/3 and g4/3 of SH1-110, forcing some multicast traffic to move to two other links. Verify the multicast frame loss.

Step 11 Bring up the two links failed in Step 10 and verify the multicast frame loss.

Step 12 Fail all four links of the GEC between SH1-103 and SH1-108, forcing multicast traffic to go through SH1-107 to get to SH1-103. Verify the multicast frame loss. Use the **show ip mroute and test etherchannel load-balance** commands to verify that traffic is redirected through SH1-107. Determine the traffic loss:

Commands

- **test etherchannel load-balance interface port-channel** *id* **ip** *source-ip*
- **show ip mroute**

Step 13 Bring up the four links between SH1-107 and SH1-103. Use the **show ip mroute and test etherchannel load-balance** commands, and examine the interface counters to verify that traffic is once again sent through SH1-108:

Commands

- **test etherchannel load-balance interface port-channel** *id* **ip** *source-ip*
- **show ip mroute**
- **show interfaces counters module** *3* | *4*

Step 14 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU tracking results.

Expected Results

We expect multicast and MSDP functionality during a dCEF GEC failover to work properly.

Results

Table 27 shows the IP multicast Gigabit EtherChannel Failover: dCEF GEC Failover test results.

Table 27 IP Multicast Gigabit EtherChannel Failover: dCEF GEC Failover Test Results

Tests	Results
Gigabit EtherChannel Failover: dCEF GEC Failover	Pass

Switch Fabric Module Failover

This test verified multicast functionality during Switch Fabric Module (SFM) failover. Multicast traffic generated by IXIA was sent through the network from Dista-2 to Dista-1, through SH1-110. The legacy module on SH1-110 was powered down, ensuring compact-mode switching, and the active SFM was failed in a number of situations. Traffic was tracked in each case, and traffic loss was calculated.

In several of the following steps, verification that the traffic is using the SFM is requested. Verify by examining how the Medusa ASIC is handling traffic. The Medusa ASIC interfaces are between the line card local bus and the backplane of the Catalyst 6500. If traffic is using the SFM (no legacy, or nonfabric-enabled, cards present), the Medusa mode will be "Compact" for all line cards, indicating that it is sending a compact header on the backplane for the switching decision. Note that any WS-X6816 modules are always in "Compact" mode, because they are fabric-only cards. More on the switching modes involving the SFM can be found at the Cisco Catalyst 6500 Series Switches site:

http://www.cisco.com/en/US/products/hw/switches/ps708/products_configuration_guide_chapter09186a008007fb2a.html#xtocid17541

Test Plan

The procedure used to perform the fabric module failover test follows.

Step 1 Enter the **show memory summary** command to take an initial memory snapshot of device SH1-110. Begin CPU utilization monitoring for all involved devices using the **show processes cpu** command.

Step 2 Verify that SH1-103 and SH1-104 are running MSDP Anycast for multicast group 239.255.127.x by using the following command:

Commands

- **show running-config | include msdp**

Step 3 Use the **ip pim spt-threshold infinity** command on SH1-107 and SH1-108 (Supervisor Engine 1 and MultiLayer Switch Feature Card 2) to ensure that the multicast routing state (*, G) is used, and not (S, G):

Commands

- **show running-config | include spt**

Step 4 Set SH1-103 as the primary PIM rendezvous point by configuring the **ip ospf cost 10** command on the loopback1 interface of SH1-104:

Commands

- **show running-config interfaces lo1**

Step 5 Verify that SH1-108 and SH1-110 are the PIM-DRs on their respective segments:

Commands

- **show ip pim neighbor**

Step 6 Begin sending the IGMP join messages for multicast groups 239.255.127.100 through 239.255.127.104 to Dista-1 4/6 (IXIA port 8/2).

Step 7 Use IXIA port 10/1 to send multicast traffic to groups 239.255.127.100 through 239.255.127.104 on Dist A-2 VLAN 11 g2/1. Verify that SH1-103 is the primary PIM rendezvous point and is forwarding traffic:

Commands

- **show ip pim rp**
- **show ip route 172.31.0.127**
- **show ip mroute**

Step 8 On SH1-110, power down the legacy module (module 9). Verify that traffic is using the SFM:

Commands

- **no power enable module 9**
- **show module**
- **show fabric medusa mode**

Step 9 On SH1-110, fail over the active SFM. Check the traffic stream for lost traffic.

Commands

- **show fabric active**
- **hw-module module** *5|6* **reset**

Step 10 On SH1-110, power up the legacy module and display the fabric switching mode. Check the traffic stream for lost traffic.

Commands

- **power enable module 9**
- **show module**
- **show fabric medusa mode**

Step 11 On SH1-110, fail over the active SFM. Check the traffic stream for lost traffic.

Commands

- **show fabric active**
- **hw-module module** *5|6* **reset**

Step 12 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results for device SH1-110.

Expected Results

We expect multicast functionality during SFM failover to work properly.

Results

Table 28 shows the switch fabric module failover test results.

Table 28 Switch Fabric Module Failover Test Results

Tests	Results
Switch Fabric Module Failover	Pass

Gigabit Ethernet Module Failover

This test verified multicast functionality during Gigabit Ethernet (GE) module failover. The Gigabit Ethernet modules of SH1-110 and SH1-108 was reset (failed over) with traffic passing through them. The ability of the devices to compensate for these failovers was measured by traffic loss.

Test Plan

The procedure used to perform the Gigabit Ethernet module failover test follows.

Step 1 Enter the **show memory summary** command to take an initial memory snapshot of devices SH1-108, SH1-110, and SH1-103.

Step 2 Verify that core routers SH1-103 and SH1-104 are running MSDP Anycast by using the following command:

Commands

- **show running-config | include msdp**

Step 3 Enter the **ip pim spt-threshold infinity** command on SH1-107 and SH1-108 (Engine Supervisor 1 and MultiLayer Switch Feature Set 2) to verify that the multicast routing state (*, G) is used, and not (S, G):

Commands

- **show running-config | include spt**

Step 4 Set SH1-103 as the primary PIM rendezvous point by configuring the **ip ospf cost 10** command on loop back interface 1 of SH1-104. Verify that SH1-103 is the PIM rendezvous point:

Commands

- **show running-config interfaces lo1**

Step 5 Verify that SH1-108 and SH1-110 are the PIM-DRs on their respective segments:

Commands

- **show ip pim neighbor**

Step 6 Start IGMP join messages from IXIA to Dista-2 4/10 (IXIA port 10/2), SH1-110 g8/16 (IXIA port 12/2), and SH1-110 g4/16 (IXIA port 11/1) for multicast groups 239.255.127.100 through 239.255.127.104.

Step 7 Start IXIA traffic, sending multicast traffic to groups 239.255.127.100 through 239.255.127.104 on SH1-108 g4/8 (IXIA port 7/2). Set the traffic rate at 10,000 pps. Configure the traffic stream so that it is being sent from four different source IP addresses which will ensure that the traffic is spread across all four GEC links. Verify that SH1-103 is the rendezvous point and that traffic is passing through it, coming in from port-channel 69 and going out port-channel 71:

Commands

- **show ip mroute**
- **show ip pim rp**
- **show ip route 172.31.0.127**

Step 8 Reset SH1-110 GE module 7. Determine traffic loss.

Commands

- **hw-module module 7 reset**

Step 9 Reset SH1-110 Gigabit Ethernet module 8. Determine traffic loss.

Commands

- **hw-module module 8 reset**

Step 10 Reset SH1-110 Gigabit Ethernet module 3. Determine traffic loss.

Commands

- **hw-module module 3 reset**

Step 11 Reset SH1-110 Gigabit Ethernet module 4. Determine traffic loss.

Commands

- **hw-module module 4 reset**

Step 12 Reset SH1-108 Gigabit Ethernet module 3. Determine traffic loss.

Commands

- **hw-module module 3 reset**

Step 13 Stop the IXIA joins and traffic started in Step 6 and Step 7.

Step 14 Start IGMP join messages from IXIA to Dista-1 4/1 (IXIA port 8/1) for multicast groups 239.255.127.100 through 239.255.127.104.

Step 15 Start IXIA traffic, sending multicast traffic to groups 239.255.127.100 through 239.255.127.104 on Dista-2 2/1 (IXIA port 10/1). Set the traffic rate at 10,000 pps. Configure the traffic stream so that it is being sent from four different source IP addresses which will ensure that the traffic is spread across all four GEC links. Verify that traffic is passing through SH1-103, coming in interface port-channel 71 and going out port-channel 69:

Commands

- **show ip mroute**

Step 16 Reset Gigabit Ethernet module 7 on SH1-110. Determine traffic loss.

Commands

- **hw-module module 7 reset**

Step 17 Reset Gigabit Ethernet module 8 on SH1-110. Determine traffic loss.

Commands

- **hw-module module 8 reset**

Step 18 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results for devices SH1-103, SH1-108, and SH1-110.

Expected Results

We expect multicast functionality during GE module failover to work properly.

Results

Table 29 shows the Gigabit Ethernet module failover test results.

Table 29 Gigabit Ethernet Module Failover Test Results

Tests	Results
Gigabit Ethernet Module Failover	Pass

Protocol Independent Module-Designated Router Failover

This test verified multicast functionality during designated router (PIM-DR) failover. IXIA traffic for five multicast groups was sourced from the IXIA connected to Dista-1 and collected from Dista-2. The default PIM-DRs in this part of the network are SH1-108 (Sup12) and SH1-110 (Sup22). Their backups were SH1-107 and SH1-109, respectively. Giving coverage to the Sup22, SH1-110 was isolated from the network by shutting down its port-channel interfaces to its neighbors. SH1-109 was then forced to take over the PIM-DR role and forward traffic. Coverage was given for the Sup12 by performing similar actions on SH1-108 and SH1-107.

Test Plan

The procedure used to perform the Protocol Independent Module-designated router (PIM-DR) failover test follows.

Step 1 Enter the **show memory summary** command to take an initial memory snapshot of devices SH1-107, SH1-108, SH1-109, SH1-110, and SH1-103.

Step 2 Verify that SH1-103 and SH1-104 are running MSDP Anycast and are peered to each other through their loopback 0 interfaces by using the following commands.

Commands

- **show running-config | include msdp**
- **show running-config interfaces lo0**

Step 3 Use the **ip pim spt-threshold infinity** command on SH1-107 and SH1-108 (Supervisor Engine 1 and MultiLayer Switch Feature Set 2) to ensure that the multicast routing state (*, G) is used, and not (S, G):

Commands

- **show running-config | include spt**

Step 4 Set SH1-103 as the primary PIM rendezvous point by configuring the **ip ospf cost 10** command on loopback 1 interface of SH1-104, because SH1-103 has a default cost of 1:

Commands

- **show running-config interfaces lo1**

Step 5 Verify that SH1-108 and SH1-110 are the PIM-DRs on their respective segments:

Commands

- **show ip pim neighbor**

Step 6 Start IGMP join traffic for multicast groups 239.255.127.100 through 239.255.127.104 on Dista-2 2/1 (IXIA port 10/1) and Dista-2 4/10 (IXIA port 10/2).

Step 7 Start an IXIA traffic stream for multicast groups 239.255.127.100 through 239.255.127.104 on Dista-1 4/1 (IXIA port 8/1).

Step 8 Verify that the traffic stream is going through SH1-103, coming in interface port-channel 69 and going out interface port-channel 71:

Commands

- **show ip mroute**
- **show interfaces po69 counters**
- **show interfaces po71 counters**

Step 9 Shut down port channels 71, 171, and 20 on SH1-110, so that SH1-109 becomes the PIM-DR. Verify that traffic being routed through SH1-103 is now going out interface port-channel 70. Determine the amount of traffic loss:

Commands

- **show ip mroute**
- **show interfaces po70 counters**

Step 10 Bring up the interfaces that were shut down in the previous step, so that SH1-110 becomes the PIM-DR again. Verify that traffic being routed through SH1-103 is now going out interface port-channel 71. Verify that SH1-110 is again the PIM-DR:

Commands

- **show ip mroute**
- **show interfaces po71 counters**
- **show ip pim neighbor**

Step 11 Shut down port channels 69, 169, and 10 on SH1-108, so that SH1-107 becomes the PIM-DR. Verify that traffic being routed through SH1-103 is now coming in through interface port-channel 68. Determine the amount of traffic loss:

Commands

- **show ip mroute**
- **show interfaces po68 counters**

Step 12 Bring up the interfaces that were shut down in the previous step, so that SH1-108 becomes the PIM-DR again. Verify that traffic being routed through SH1-103 is now coming in interface port-channel 69. Verify that SH1-108 is again the PIM-DR for this segment. Determine the amount of traffic loss:

Commands

- **show ip mroute**
- **show interfaces po69 counters**
- **show ip pim neighbor**

Step 13 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results of all devices listed in Step 1.

Expected Results

We expect multicast functionality during PIM-DR failover to work properly.

Results

Table 30 shows the protocol independent module-designated router failover test results.

Table 30 Protocol Independent Module-Designated Router Failover Test Results

Tests	Results
Protocol Independent Module-Designated Router Failover	Pass

Auto-Rendezvous Point Functionality and Failover

Auto Rendezvous Point eliminates the need to statically (manually) configure rendezvous point information in every router in the network. Selected routers in the network can be configured to be candidate rendezvous points for a given set of multicast groups (as defined by access control lists). These routers advertise their candidacy in the form of rendezvous point-announce messages at a configurable interval (default is 60 seconds).

One or more (for redundancy) routers are configured as mapping agents, which will cache active Group-to rendezvous point information. Mapping agents learn which routers are candidate rendezvous points for given groups by joining the well-known Cisco-RP-Announce multicast group (224.0.1.39) to receive candidate rendezvous point announcements. Once an rendezvous point is selected (based on highest IP address) by the mapping agent, the other routers in the network will learn of the IP address of the rendezvous point for the given groups by listening to the well-known Cisco-RP-Discovery multicast group (224.0.1.40) which they had joined at startup.

The command to configure a router as a candidate rendezvous point is:

```
ip pim send-rp-announce interface scope ttl
```

The command to configure the mapping agent is:

```
ip pim send-rp-discovery scope ttl
```

This test verified that two candidates were configured with each of those routers also acting as the mapping agents. Test success was based on whether the candidate with the highest IP address was correctly advertised as the rendezvous point for the active groups. A test involving the failure of the elected candidate rendezvous point wasl also performed to verify traffic failover of the candidate with the lower IP address, and to verify that traffic reverted its path through the candidate with the higher IP address, once it comes back online.

Test Plan

The procedure used to perform the auto-RP functionality and failover test follows.

Step 1 Enter the **write memory** command on SH1-103, SH1-104, SH1-107, SH1-108, SH1-109, and SH1-110.

Step 2 Enter the **show memory summary** command to take an initial memory snapshot of devices SH1-103 and SH1-104.

Step 3 There are four commands on each of the devices listed in Step 1 that must be removed, or negated. The following commands represent the configuration of a static rendezvous point, which are not used during this procedure:

Commands

- **no ip pim rp-address 172.31.0.127 1 override**
- **no ip pim rp-address 172.31.0.128 2 override**
- **no ip pim rp-address 172.31.0.129 3 override**
- **no ip pim rp-address 172.31.0.130 4 override**

Step 4 Configure SH1-103 and SH1-104 with the following commands (in global configuration mode):

Commands

- **ip pim send-rp-announce loopback 0 scope 16**
- **ip pim send-rp-discovery scope 16**

Step 5 Configure switches SH1-107, SH1-108, SH1-109, and SH1-110 to accept rendezvous point statements from auto rendezvous point mapping agents, which will define the active rendezvous point router for these devices:

Commands

- **ip pim accept-rp auto-rp**

Step 6 Verify that the commands entered in Step 4 and Step 5 are the only ones defining rendezvous point behavior on the devices listed in Step 1:

Commands

- **show running-config | include rp**

Step 7 Begin IGMP join messages sent for multicast groups 239.255.127.100 through 239.255.127.104 on SH1-108 g3/8 (IXIA port 7/1) and SH1-108 g4/8 (IXIA port 7/2).

Step 8 Begin IXIA traffic streams for multicast groups 239.255.127.100 through 239.255.127.104 on Dista-2 2/1 (IXIA port 10/1).

Step 9 Verify that SH1-104 is the elected rendezvous point, because of its higher IP address of its loopback 0 interface. Verify that all the proper devices recognize SH1-104 as the rendezvous point:

Commands

- **show running-config interfaces lo0**
- **show ip pim rp**
- **show ip pim rp mapping**

Step 10 Verify that all traffic that is sent is received on the proper ports, and that the traffic is passing through the candidate rendezvous point with the highest IP address (SH1-104):

Commands

- **show ip mroute**

Step 11 Fail (power off/reset/isolation) the switch with the higher IP address, or the current elected rendezvous point. Verify that traffic is successfully diverted through the other candidate rendezvous point. Measure the time required for this event occur:

Commands

- **show ip mroute**
- **show ip pim rp**
- **show ip pim rp mapping**

Step 12 Reintroduce SH1-104 into the network. When SH1-104 is back online, verify that traffic once again passes through it. Measure any traffic loss:

Commands

- **show ip mroute**
- **show ip pim rp**
- **show ip pim rp mapping**

Step 13 Negate the commands configured in Step 4 and Step 5 on the appropriate switches:

Commands

- **no ip pim send-rp-announce loopback 0 scope 16**
- **no ip pim send-rp-discovery scope 16**
- **no ip pim accept-rp auto-rp**

Step 14 Copy the configuration stored in NVRAM to the current running configuration:

Commands

- **copy startup running-config**

Step 15 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results for devices SH1-103 and SH1-104.

Expected Results

We expect the Auto Rendezvous Point feature to eliminate the need to manually configure rendezvous point information in every router in the network. We also expect to see traffic fail over of the candidate with the lower IP address and when the elected candidate rendezvous point fails, that traffic reverts its path through the candidate with the higher IP address when it comes back online.

Results

Table 31 shows the auto-RP functionality and failover test results.

Table 31 Auto-RP Functionality and Failover Test Results

Tests	Results
Auto-Rendezvous Point Functionality and Failover	Pass

Layer 3 Interface Multicast Negative

This test introduced faults into the topology and verified that multicast functionality remained consistent with functional specifications on Layer 3 ports. We introduced online insertion and removal (OIR) of line cards, reset individual line cards, and reload the switch and supervisor engine or Switch Fabric Module failover.

Test Plan

The procedure used to perform the Layer 3 interface multicast negative test follows.

Step 1 Begin tracking memory usage and CPU utilization for device SH1-110 for the test procedure period.

Step 2 Verify that interface g4/16 of SH1-110 is configured as a Layer 3 interface by using the following command. This interface is connected to IXIA port 11/1:

Commands

- **show running-config interfaces g4/16**

Step 3 Use the **ip pim spt-threshold infinity** command on SH1-107 and SH1-108 (supervisor engine 1 and MultiLayer Switch Feature Card 2) to ensure that the multicast routing state (*, G) is used, and not (S, G):

Commands

- **show running-config | include spt**

Step 4 Configure SH1-103 as the rendezvous point for the groups by configuring the **ip ospf 10** command on the loopback 1 interface of SH1-104:

Commands

- **show running-config interfaces lo1**

Step 5 Verify that the IXIA port (11/1) connected to SH1-110 is receiving traffic destined for multicast groups 239.255.129.100 through 239.255.129.104 and is sending to multicast groups 239.255.127.100 through 239.255.127.104:

Commands

- **show interfaces g4/16 counters**
- **show ip mroute**

Step 6 Verify that the IXIA port (7/1) connected to SH1-108 (g3/8) is configured exactly the opposite, that it is receiving traffic destined for multicast groups 239.255.127.100 through 239.255.127.104 and sending to multicast groups 239.255.129.100 through 239.255.129.104:

Commands

- **show interfaces g3/8 counters**
- **show ip mroute**

Step 7 Demonstrate the path for each set of multicast groups, 239.255.127.x and 239.255.129.x, between SH1-108 and SH1-110:

Commands

- **show ip mroute**

Step 8 Reload the active supervisor in SH1-110, and check the mroute entries of SH1-103 and SH1-104 (after SH1-110 comes back online) to verify that they are passing traffic in the same manner as reported in Step 7. Measure any traffic loss:

Commands

- **show ip mroute**

Step 9 Perform OIR on the active supervisor on SH1-110, and check the mroute entries of SH1-103 and SH1-104 to verify that they are passing traffic in the same manner as reported in Step 6. Measure any traffic loss:

Commands

- **show ip mroute**

Step 10 Reset module 4 on SH1-110, and check the mroute entries of SH1-103 and SH1-104 to verify that they are passing traffic in the same manner as reported in Step 6. Measure any traffic loss:

Commands

- **hw-module module 4 reset**
- **show ip mroute**

Step 11 Perform OIR on module 4 on SH1-110, and check the mroute entries of SH1-103 and SH1-104 to verify that they are passing traffic in the same manner as reported in Step 6. Measure any traffic loss:

Commands

- **show ip mroute**

Step 12 Disable power to module 9, a legacy module, on SH1-110 and verify that the traffic on SH1-110 is being switched in compact mode. Fail over the active SFM on SH1-110, and check the mroute entries of SH1-103 and SH1-104 to verify that they are passing traffic in the same manner as reported in Step 6. Measure any traffic loss:

Commands

- **no power enable module 9**
- **show fabric medusa mode**
- **show fabric active**
- **show ip mroute**

Step 13 Cycle interface g4/16 (**shut** or **no shut** command) on SH1-110, and check the mroute entries of SH1-103 and SH1-104 to verify that they are passing traffic in the same manner as reported in Step 6. Measure any traffic loss:

Commands

- **show ip mroute**

Step 14 Reenable power to legacy module 9:

Commands

- **no power enable module 9**

Step 15 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results.

Expected Results

We expect multicast functionality to remain consistent with functional specifications on Layer 3 ports when introducing faults into the topology.

Results

Table 32 shows the Layer 3 interface multicast negative test results.

Table 32 Layer 3 Interface Multicast Negative Test Results

Tests	Results
Layer 3 Interface Multicast Negative	Pass

Unicast and Multicast Test with 130K Injected IP Routes

This test verified that the switch functioned correctly when both unicast and multicast traffic were added with 100K BGP, 20K OSPF, and 10K EIGRP routes injected into the network.

Test Plan

The procedure used to perform the unicast and multicast test with 130K injected IP routes test follows.

Step 1 Use an SNMP monitoring utility to begin monitoring memory and CPU utilization for devices SH1-97, SH1-98, SH1-99, SH1-100, SH1-101, SH1-102, SH1-103, SH1-104, SH1-105, SH1-106, SH1-107, SH1-108, SH1-109, and SH1-110.

Step 2 Verify that the following devices are running Multicast Source Discovery Protocol (MSDP) Anycast and are peered as indicated: SH1-99 peered with SH1-100; SH1-103 peered with SH1-104; and SH1-107 peered with SH1-108:

Commands

- **show running-config | include msdp**
- **show running-config interfaces Lo0**

Step 3 Verify that SH1-99 and SH1-100 are configured rendezvous points for multicast groups 239.255.126.x:

Commands

- **show running-config | include rp**
- **show running-config interfaces Lo1**
- **show access-lists 1**
- **show ip interfaces brief | include** *ip-address*

Step 4 Verify that SH1-103 and SH1-104 are configured rendezvous points for multicast groups 239.255.127.x:

Commands

- **show running-config | include rp**
- **show running-config interfaces Lo1**
- **show access-lists 1**
- **show ip interfaces brief | include** *ip-address*

Step 5 Verify that SH1-107 and SH1-108 are configured rendezvous points for multicast groups 239.255.129.x:

Commands

- **show running-config | include rp**
- **show running-config interfaces Lo1**
- **show access-lists 1**
- **show ip interfaces brief | include** *ip-address*

Step 6 Verify that SH1-102, SH1-106, SH1-108, and SH1-110 are PIM-DRs on their respective segments:

Commands

- **show ip pim neighbor**

Step 7 Generate IP routes, as follows:

a. Start IXIA's OSPF Protocol Server on IXIA ports 6/1 to 6/4 to generate 2000 OSPF routes.

b. Start BGP route generation on Pagent device to generate 35,000 BGP routes.

c. Start OSPF route generation on Pagent device to generate about 18,000 OSPF routes.

d. Start BGP route generation on Pagent device to generate about 65,000 BGP routes.

Step 8 Verify that all routes have been fed into the network and propagated. Devices SH1-97 through SH1-100 should have about 100,000 BGP routes and 20,000 OSPF routes, for a total of about 120,000 routes. Devices SH1-101 and SH1-102 should have about 20,000 EIGRP routes. Devices SH1-103 through SH1-110 should have about 20,000 OSPF routes each:

Commands

- **show ip route summary**

Step 9 Begin the following IXIA unicast traffic flows:

a. A bidirectional 18,000-pps flow between IXIA-4/2 (Dista-1 6/5) and IXIA-5/2 (Dista-2 6/2);

b. A unidirectional 1000 pps flow from IXIA-11/1 (SH1-110 g4/16) to the 130.1.1.0 and 140.77.45.0 networks in OSPF areas 3 and 4, generated by Pagent. To measure packet receipt for these flows, examine the unicast output to Fa7/37 on SH1-100 (IXIA 6/2) for hosts 130.1.1.x and Fa7/38 on SH1-99 (IXIA 6/4) for hosts 140.77.45.x.

Step 10 Send multicast traffic, as follows:

a. Use IXIA port 11/2 (SH1-101 g3/16) to send 74 bytes of multicast traffic at 15K-pps to group 239.255.126.100. The receivers are IXIA ports 4/1 (Dista-1 6/1) and IXIA port 5/1 (Dista-2 6/1).

b. Use IXIA port 5/1 (Dista-2 6/1) to send 74 bytes of multicast traffic 15K-pps to group 239.255.126.101. The receivers are IXIA port 4/1 (Dista-1 6/1) and IXIA port 5/2 (Dista-2 6/2).

Step 11 Use the TCL test script to send and collect a given amount of traffic on each of the streams (and the background unicast stream: Ix-13/1 and Ix-13/2 to Ix-1/1). Send this traffic at the rates shown for 20 minutes. Verify that all traffic is received where it should have been with zero packet loss.

Step 12 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results.

Expected Results

We expect that switch to function correctly when both unicast and multicast traffic are added with 100K BGP, 20K OSPF, and 10K EIGRP routes injected into the network.

Results

Table 33 shows the unicast and multicast test with 130K injected IP routes test results.

Table 33 Unicast and Multicast Test with 130K Injected IP Routes Test Results

Tests	Results
Unicast and Multicast Test with 130K Injected IP Routes	Pass

IP PIM Neighbor-Filter Command

DDTS CSCdw63676 describes a case in which the router crashes when the **ip pim neighbor-filter** *access-list* command is entered on a VLAN interface, regardless of whether that ACL is present.

This test verified that the solution for this DDTS was effective and present in Cisco IOS Release 12.1(8b)E12. The DUT is SH1-103. This command was configured on a VLAN on the DUT. This test verified that no negative effect resulted from this command being entered.

Test Plan

The procedure used to perform the IP PIM **neighbor-filter** command test follows.

Step 1 On the DUT, enter the commands on a VLAN interface:

Commands

- **show ip interfaces br | include Vlan**
- **ip pim neighbor-filter 55**

Step 2 Configure the **neighbor-filter** command on that Vlan and watch for a crash:

Step 3 Verify that a crash does not occur.

Expected Results

We expect that no negative effect results from entry of the **ip pim neighbor-filter** *access-list* command in Cisco IOS Release 12.1(8b)E12.

Results

Table 34 shows the IP multicast IP PIM neighbor-filter command test results.

Table 34 IP Multicast IP PIM Neighbor-Filter Command Test Results

Tests	Results
IP PIM Neighbor-Filter Command	Pass

Dual Sources

DDTS CSCdx55659 describes a case in which an (S, G) entry on a router was deleted if that router was receiving traffic for a single multicast group from two separate, legitimate sources. In this case, for which a solution was verified in this test, traffic for multicast group 239.255.127.100 entered the network at Dista-2. A multicast receiver was on SH1-109 and successfully received this traffic (creating an (S, G) entry in its mroute table. If a second source for group 239.255.127.100 was started, entering at SH1-108, this (S, G) entry on SH1-109 was deleted after about 2 minutes. If the traffic source was stopped, the (S, G) entry will not recover on its own; instead, a **clear ip mroute *** command needed to be entered.

A fix for this problem was integrated into Cisco IOS Release 12.1(8b)E12. A successful test was one in which the problem fails to reproduce, and all traffic sent by both sources was received.

Test Plan

The procedure used to perform the dual sources test follows.

Step 1 Use an SNMP monitoring utility to begin monitoring memory and CPU utilization for devices SH1-108, SH1-109, and SH1-110.

Step 2 Start IP IGMP join messages with the IXIA protocol view on IXIA ports 9/1, 9/2, 11/1, and 7/1.

Step 3 Start an IXIA multicast traffic stream for multicast group 239.255.127.100, entering at Dista-2 (IXIA port 8/1).

Step 4 Verify that traffic for the mcast group is received on all appropriate interfaces and that a valid (S, G) entry exists in the mroute table for SH1-109:

Commands

- **show ip mroute**

Step 5 Start an IXIA multicast traffic stream for multicast group 239.255.127.100, entering at SH1-108 (IXIA port 7/2).

Step 6 Allow traffic from both sources to run, uninterrupted, for about 5 minutes. Verify that all traffic is being received on the appropriate interfaces after those 5 minutes and that the (S, G) entry is present in the mroute table of SH1-109:

Commands

- **show ip mroute**

Step 7 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results.

Expected Results

We expect this DDTS to fail to reproduce, and all traffic sent by both sources to be received.

Results

Table 35 shows the dual sources test results.

Table 35 Dual Sources Test Results

Tests	Results
Dual Sources	Pass

Secondary Subnet

DDTS CSCdu71914 provides a fix for a case in which MLS entries are not created for IP multicast traffic on secondary subnets. This test verified resolution of the problem. On device SH1-108, a secondary subnet was defined for VLAN 15, to which the IXIA-generated multicast traffic came, from Dista-2. The primary IP address for VLAN 15 was 172.31.15.69/24. The secondary IP address for VLAN 15 on SH1-108 was 172.31.150.69/24. The IXIA port 7/2, which was the source of the IGMP joins and the destination for the multicast traffic, configured for IGMP joins coming from IP sources 172.31.15.72 and 172.31.150.72.

With these configurations in place, an IXIA traffic stream for groups 239.255.127.100 through 239.255.127.104 was started from Dista-2. IGMP joins were started on Dista-1 for those groups (Ix7/2). This test verified that an MLS entry exists on SH1-108 for the secondary subnet.

Test Plan

The procedure used to perform the secondary subnet test follows.

Step 1 Start monitoring CPU and memory for SH1-110.

Step 2 Verify that a secondary IP address is configured on Gigabit interface 4/16 of SH1-110:

Commands

- **show running-config interfaces g4/16**

Step 3 Verify that both IP addresses 172.31.3.111 and 172.31.33.111 are configured in the IXIA IP table (in the Protocol Window) for port 11/1 and that both are associated with MAC address 0001.1101.0000.

Step 4 In the IXIA IGMP table (in the Protocol Window) for port 11/1, verify that the joins are being sent from the secondary IP subnet, or address 172.31.3.111.

Step 5 Start IP IGMP join messages with the IXIA protocol view on IXIA port 11/1 (SH1-110 g4/16).

Step 6 Start an IXIA multicast traffic stream (at 50,000 pps) for multicast groups 239.255.127.100 through 239.255.127.104, entering at Dista-1 (IXIA port 8/1).

Step 7 Verify that traffic for the mcast groups is received on the appropriate interface:

Commands

- **show interfaces g4/16 counters**

Step 8 Verify that IGMP join messages are being received on SH1-110, coming from the secondary subnet address:

Commands

- **show ip igmp groups**

Step 9 Verify that multicast traffic is being hardware-switched:

Commands

- **show mls ip multicast group** *group*
- **show ip mroute**

Step 10 Turn off MLS IP multicast on SH1-110, and verify that the CPU utilization increases because multicast traffic is now being software-switched:

Commands

- **show processes cpu | include CPU utilization**
- **no mls ip multicast**

Step 11 Turn MLS IP multicast back on and verify that CPU utilization returns to near zero, as multicast traffic is once again being hardware-switched:

Commands

- **show processes cpu | include CPU utilization**
- **mls ip multicast**

Step 12 Send a fixed number of packets in a newly started stream and verify that all multicast traffic that was sent has been received.

Step 13 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results.

Expected Results

We expect MLS entries will be created for IP multicast traffic on secondary subnets.

Results

Table 36 shows the secondary subnet test results.

Table 36 Secondary Subnet Test Results

Tests	Results
Secondary Subnet	Pass

MSDP Failover

This test verified MSDP functionality during an MSDP PIM-rendezvous point (RP) failover. A multicast traffic stream was sent from Dista-1 to Dista-2. SH1-103, was configured as the PIM rendezvous point, and traffic flowed through it. The rendezvous point (SH1-103) was failed over in three ways. First, it was isolated by shutting down its port-channel interfaces connecting it to the rest of the network (simulating a device failure). Second, the loopback1 interface on SH1-103 will be shut down, forcing SH1-104 to become the rendezvous point. Third, the port-channel between SH1-103 and SH1-108 was shut down, forcing SH1-104 to become the rendezvous point.

Test Plan

The procedure used to perform the MSDP failover test follows.

Step 1 Enter the **show memory summary** command to take an initial memory snapshot of devices SH1-103, SH1-104, SH1-108, and SH1-110.

Step 2 Verify that core routers SH1-103 and SH1-104 are running MSDP Anycast and are peered to each other by their respective loopback interface 0:

Commands

- **show running-config | include msdp**
- **show running-config interfaces lo0**

Step 3 Enter the **ip pim spt-threshold infinity** command on SH1-107, SH1-108, SH1-109, and SH1-110 to ensure that the multicast routing state (*, G) is used, and not (S, G):

Commands

- **show running-config | include spt**

Step 4 Set SH1-103 as the primary PIM rendezvous point by configuring the **ip ospf cost 10** command on the loopback 1 interface of SH1-104:

Commands

- **ip ospf cost 10**

Step 5 Verify that SH1-108 and SH1-110 are the PIM-DRs on their respective segments:

Commands

- **show ip pim neighbor**

Step 6 Begin sending IGMP join protocol messages into Dista-2 4/10 (IXIA port 10/2) and SH1-110 g8/16 (IXIA port 12/2) for multicast groups 239.255.127.100 through 239.255.127.104.

Step 7 Use IXIA to send multicast traffic to groups 239.255.127.100 through 239.255.127.104 on SH1-108 g4/8 (IXIA port 7/2). Verify that traffic is flowing through SH1-103, coming in interface port-channel 69 and going out interface port-channel 71:

Note For this step and all subsequent steps, the RPF-MFD flag in the mroute entries indicates that particular flow is being completely hardware-switched.

Commands

- **show ip mroute**

Step 8 Shut down the loopback 1 interface on SH1-103, so that SH1-104 becomes the PIM rendezvous point. Verify that traffic is now using SH1-104 to reach its destination and that it is being hardware-switched. Determine the amount of traffic loss:

Commands

- **show ip mroute**
- **show interfaces** *interface* **counters**

Step 9 Bring up the loopback 1 interface on SH1-103, so that SH1-103 becomes the PIM rendezvous point again. Verify that traffic is now using SH1-103 to reach its destination and that it is being hardware-switched. Determine the amount of traffic loss:

Commands

- **show ip mroute**
- **show interfaces** *interface* **counters**

Step 10 Shut down the link (port-channel 71 on SH1-103) between SH1-103 and SH1-110, so that SH1-104 becomes the PIM rendezvous point. Verify that traffic is now using SH1-104 to reach its destination and that it is being hardware-switched. Determine the amount of traffic loss:

Commands

- **show ip mroute**
- **show interfaces** *interface* **counters**

Step 11 Bring up the link between SH1-103 and SH1-110, so that SH1-103 becomes the PIM rendezvous point. Verify that traffic is now using SH1-104 to reach its destination and that it is being hardware-switched. Determine the amount of traffic loss:

Commands

- **show ip mroute**
- **show interfaces** *interface* **counters**

Step 12 Isolate SH1-103 by shutting down its interfaces: port-channel 68, port-channel 69, port-channel 70, and port-channel 71. Verify that traffic is now using SH1-104 to reach its destination and that it is being hardware-switched. Determine the amount of traffic loss:

Commands

- **show ip mroute**

- **show interfaces** *interface* **counters**

Step 13 Bring up the interfaces on SH1-103 that were failed in Step 12, so that SH1-103 becomes the PIM rendezvous point again. Verify that traffic is now using SH1-104 to reach its destination and that it is being hardware-switched. Determine the amount of traffic loss:

Commands

- **show ip mroute**
- **show interfaces** *interface* **counters**

Step 14 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results for all devices listed in Step 1.

Expected Results

We expect that multicast traffic will be forwarded in the event of an MSDP failover.

Results

Table 37 shows the MSDP failover test results.

Table 37 MSDP Failover Test Results

Tests	Results
MSDP Failover	Pass

Layer 3 Routing Features

Layer 3 routing feature testing for Safe Harbor involves these features:

- Cisco Express Forwarding, page 71
- Open Shortest Path First, page 75
- Border Gateway Protocol, page 82
- Hot Standby Routing Protocol, page 84
- Enhanced Interior Gateway Routing Protocol, page 88

Cisco Express Forwarding

Cisco Express Forwarding (CEF) evolved to best accommodate the changing network dynamics and traffic characteristics resulting from increasing numbers of short period flows typically associated with web-based applications and interactive type sessions. Existing Layer 3 switching paradigms use a route-cache model to maintain a fast lookup table for destination network prefixes. The route-cache entries are traffic-driven in that the first packet to a new destination was routed via routing table information and, as part of that forwarding operation, a route-cache entry for that destination was then added. This behavior allows subsequent packet flows to that same destination network to be switched based on an efficient route cache match. These entries are periodically aged out to keep the route cache current and can be immediately invalidated if the network topology changes. This "demand-caching" scheme—maintaining a very fast access subset of the routing topology information—is optimized for scenarios whereby the majority of traffic flows are associated with a subset of destinations. However, given that traffic profiles at the core of the Internet (and potentially within some large enterprise networks) are no longer resembling this model, a new switching paradigm was required that would eliminate the increasing cache maintenance resulting from growing numbers of topologically dispersed destinations and dynamic network changes.

CEF avoids the potential overhead of continuous cache churn by instead using a Forwarding Information Base (FIB) for the destination switching decision that mirrors the entire contents of the IP routing table; that is, there was a one-to-one correspondence between FIB table entries and routing table prefixes, Therefore, there was no need to maintain a route cache.

The following tests were performed:

- Cisco Express Forwarding Packet-Switching, page 71
- Cisco Express Forwarding Table Entries, page 72
- CEF Load Balance, page 73

Cisco Express Forwarding Packet-Switching

This test verified that IP unicast traffic was hardware-switched. If traffic was not hardware-switched (by the PFC), the traffic was fast-switched by the MSFC (see Table 38).

Results Summary

Table 38 Packets Hardware-Switched

Packets Sent by SH1-104	Packets Received by SH1-104	Packets Switched by SH1-99	Percentage Switched
500	500	1000	100

Test Plan

The procedure used to perform the Layer 3 CEF Cisco Express Forwarding Packet Switching follows.

Note Safe Harbor routers SH1-99, SH1-103 and SH1-104 were used for the test. Interfaces port-channel 16 was used to connect SH1-99 to SH1-103. Interface port-channel 17 was used to connect SH1-99 through SH1-104.

Step 1 Send a fixed number of pings from SH1-104 to SH1-103, across SH1-99. Compare the pings and responses totals against the MLS CEF Layer 3 packets switched. Use an extended ping of 500 packets to calculate the number of packets hardware-switched across the interim switch, SH1-99. The result should be 1,000 FIB-switched packets on SH1-99.

The Safe Harbor test bed has many redundant paths and the resulting asymmetric routes. For this test, block those routes to force a symmetric route between the test switches.

As the result of those interfaces being shut, all traffic is switched across module 3 on SH1-99.

Step 2 Collect the initial packet count. The ability to clear these counters is introduced in a later version of Cisco IOS software.

```
SH1-99#sho mls statistics | begin Module 3
Statistics for Earl in Module 3
```

Expected Results

We expect that IP unicast traffic will be hardware-switched.

Results

Table 39 shows the Cisco express forwarding packet-switching test results.

Table 39 Cisco Express Forwarding Packet-Switching Test Results

Tests	Results
Cisco Express Forwarding Packet-Switching	Pass

Cisco Express Forwarding Table Entries

This test verified that correct CEF table entries were used to forward IP unicast traffic passing through a switch on a WS-X6816 module with Distributed Forwarding Card (DFC).

The correct MSFC2 CEF table entries were created from the routing information obtained using the OSPF routing protocol. The proper CEF table entries were created for connected routes and remote routes.

Test Plan

The procedure used to perform the Layer 3 Cisco Express Forwarding Table Entries test follows.

Step 1 Use Safe Harbor routers SH1-99, SH1-103, and SH1-104 for the test. Use port-channel 16 interface to connect SH1-99 to SH1-103, and port-channel 17 interface to connect SH1-99 to SH1-104.

Step 2 Verify that the values are correct on all three switches and that there is a network entry and 4 host entries for each directly connected network and a single entry to the next hop for a remote network. The connected interfaces have the corresponding five CEF entries and the corresponding entry for the route learned, in this case by the routing protocol OSPF.

Expected Results

We expect that correct CEF table entries were used to forward IP unicast traffic passing through a switch on a WS-X6816 module with DFC.

Results

Table 40 shows the CEF table entries test results.

Table 40 Cisco Express Forwarding Table Entries Test Results

Tests	Results
Cisco Express Forwarding Table Entries	Pass

CEF Load Balance

These tests verified that HW shortcuts and CEF distribution functioned properly with IP unicast traffic. Two tests were performed that monitored unicast traffic streams of "many-to-one" with 20 incremented IP addresses sending traffic to a single destination IP address and "many-to-many" with 100 source and 100 destination address pairs.

Test One

The first test, many-to-one, confirmed that traffic sourced from multiple source addresses to a single used multiple paths if they were available. Traffic was shown to be hardware-switched on both the supervisor 1 and supervisor 2 modules.

Test Two

The second test, many-to-many, confirmed that traffic sourced from multiple source IP addresses to multiple destination IP addresses was switched in hardware and used multiple paths. Depending on the network configuration, the supervisor2/MSFC2 will fail to use all shared paths (polarization) because the hash algorithm will always group the traffic using the ipSrc, ipDst, and pathCount values in the same group.

The polarization problem was not present when SUP1 and SUP2 are mixed because they use different hash algorithms.

Test Plan

The procedure used to perform the Layer 3 CEF load-balance test follows.

Step 1 Verify that all port-channel interfaces are up/up with the proper interfaces bundled in them:

The first traffic topology, many-to-one, used 20 incrementing source IP addresses destined to a single IP address. The traffic source was IXIA-1 port Ix8/2 connected to switch Dista-1. The destination IP address was IXIA-1 port Ix-10/2 connected to switch Dista-2. Router SH1-108 received the traffic from Dista-1 and distributed it into the test network.

Step 2 Send 10 million packets at 100,000 packets per second. Verify that interface counters on all of the test switches showed that traffic used the multiple paths available.

Look at the MLS entries on the routers to "prove" that CEF is working and that the traffic is being hardware-switched. Also look at the CPU utilization by the devices during this portion of the test. A full 100 MB of traffic will be sent, and little if any impact on the CPU should be seen.

Step 3 Send 17 million packets at a rate of about 170,000 pps (about 100 Mbps as measured by IXIA).

In the second test, many-to-many, send IXIA unicast traffic from Dista-2 (Ix-10/2) to Dista-1 (Ix-8/2). The traffic consists of destination and source IP address pairs unicast traffic. The traffic rate is 170,000 74-byte packets per second. Use the observance of zero-impact on the CPU to confirm that traffic is being hardware-switched. Examine interface counters to follow the path of the stream from source to destination.

Traffic Source Connections

```
IXIA_10/2 ---- 2/6_DISTA2_(1/2,4/3,4/12-13) ----
(gi7/3,gi8/5,gi7/5,gi8/3-Port-channel20)_SH1-110
```

Test Traffic

```
Source Addresses -172.31.26.102-201 (172.31.26.0)
Destination Addresses - 172.31.16.82-181 (172.31.16.0)
```

Polarization

The Sup2/MSFC2 uses the ipSrc, ipDst and pathcount to make frame distribution decisions. Under certain network configurations traffic may not always use all paths available in the CEF table. In the test configuration switches 110, 103, and 104, all have the same pathcount.

Expected Results

We expect HW shortcuts and CEF distribution to function properly with IP unicast traffic.

Results

Table 41 shows the CEF load balance test results.

Table 41 CEF Load Balance Test Results

Tests	Results
CEF Load Balance	Pass

Open Shortest Path First

Open Shortest Path First (OSPF) is an Interior Gateway Protocol (IGP) developed by the OSPF working group of the Internet Engineering Task Force (IETF). Designed expressly for IP networks, OSPF supports IP subnetting and tagging of externally derived routing information. OSPF also allows packet authentication and uses IP multicast when sending and receiving packets.

The following tests were performed:

Autocost

This test verified that the **auto-cost reference-bandwidth** command functioned correctly. OSPF used the interface cost to compute routing matrics. By default:

cost = 100/bandwidth (in Mbps)

The lowest cost is 1, which is for 100 Mbps interfaces. To let GE, GEC or 10 GE, and so on interfaces have a cost that will allow comparison of the higher speed connections, the **autocost** command was used to set up a higher reference bandwidth other than 100 Mbps. The default autocost reference bandwidth for the Native IOS Safe Harbor testing was 100,000.

Test Plan

The procedure used to perform the Layer 3 OSPF autocost test follows.

Step 1 Use an SNMP monitoring utility to begin monitoring memory and CPU utilization for device SH1-97.

Step 2 Verify the autocost configuration for SH1-97:

Commands

- **show running-config | beg router ospf**

Step 3 Verify the OSPF interface costs for interfaces port-channel 1 and 13. Verify that port-channel 11 is a two-port channel, and port-channel 13 is a four-port channel. The autocost configuration allows the 4-port port-channel to have half of the cost of the 2-port port-channel:

Commands

- **show interfaces** *interface* **etherchannel**
- **show ospf interfaces** *interface*

Step 4 Reset the reference bandwidth to the default value of 100. Verify that both port-channel interfaces now have the same minimum cost of one and would be compared equally with a single fast-Ethernet interface:

Commands

- **auto-cost reference-bandwidth 10**
- **show ospf interfaces** *interface*

Step 5 Change the autocost value back to 100000, and verify that the interface costs have changed back to their original values:

Commands

- **auto-cost reference-bandwidth 100000**
- **show ospf interfaces** *interface*

Step 6 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results.

Expected Results

We expect the **auto-cost reference-bandwidth** command to function correctly.

Results

Table 42 shows the OSPF autocost test results.

Table 42 OSPF Autocost Test Results

Tests	Results
Autocost	Pass

Passive Interface

This test verified that the **passive-interface** command functioned correctly. Routers should not send routing updates out passive interfaces. In the case of OSPF, hello packets will not be sent out passive interfaces, so a neighbor will not be formed.

For this test, SH1-106 was an OSPF neighbor to devices SH1-99 and SH1-100 in OSPF area 2. SH1-100 and SH1-99 were sent OSPF routes by Pagent. SH1-100 and SH1-106 were connected by a port-channel, interface port-channel 67 on SH1-100 and interface port-channel 167 on SH1-106. SH1-99 and SH1-106 were connected by a port-channel, interface port-channel 67 on SH1-100 and interface port-channel 67 on SH1-106. The **passive-interface** command was configured on port-channel 67 on SH1-100 and on port-channel 67 on SH1-99. The neighbor tables on SH1-100 and SH1-99 were checked to verify that the command worked (SH1-106 should be removed). SH1-106 lost about 20,000 OSPF routes out of its routing table. The **passive-interface** commands were removed and the neighbor relationships and route table updates were verified.

Test Plan

The procedure used to perform the Layer 3 OSPF Passive Interface test follows.

Step 1 Use an SNMP monitoring utility to begin monitoring memory and CPU utilization for devices SH1-99, SH1-100, and SH1-106.

Step 2 Verify that SH1-99, SH1-100, and SH1-106 are running OSPF (process ID 1) and that both belong to area 2:

Commands

- **show running-config interface**

- **show running-config | beg router ospf**

Step 3 Verify the neighbor relationship between SH1-100 and SH1-106 and between SH1-99 and SH1-106:

Commands

- **show ip ospf neighbor**

Step 4 Verify the number of OSPF routes that SH1-106 has in its routing table:

Commands

- **show ip route summary**

Step 5 Configure a passive interface on port-channel 67 on SH1-99 and on port-channel 67 on SH1-100:

Commands

- **passive-interfaces** *interface*

Step 6 Verify that the neighbor relationship no longer exists between SH1-100 and SH1-106 or between SH1-99 and SH1-106. Verify that the OSPF routes SH1-106 was receiving from its neighbors are no longer in its routing table:

Commands

- **show ip ospf neighbor**
- **show ip route summary**

Step 7 Remove the passive interface configured in Step 4:

Commands

- **no passive-interface** *interface*

Step 8 Verify that the neighbor relationship between SH1-100 and SH1-99 and SH1-106 has been reestablished and verify that SH1-106 again received routing updates from its neighbors:

Commands

- **show ip ospf neighbor**
- **show ip route summary**

Step 9 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results.

Expected Results

We expect the **passive-interface** command to function correctly and routers to not send routing updates out passive interfaces.

Results

Table 43 shows the OSPF passive interface test results.

Table 43 OSPF Passive Interface Test Results

Tests	Results
Passive Interface	Pass

Filtering

This test verified the capability of OSPF to filter out routes using the **distribute-list** command. SH1-97, SH1-100, and SH1-106 were connected as such: SH1-97, SH1-100, and SH1-106. These connections were Layer 3 links (channels). The link between SH1-100 and SH1-106 (port-channel 13) was network 172.31.1.96/30. SH1-97 knows, through OSPF, that this network via port-channel 13. In this test, SH1-97 was configured with an access list and the **distribute-list** command, so that SH1-97 no longer knew this route; it was filtered out.

Test Plan

The procedure used to perform the Layer 3 OSPF filtering test follows:

Step 1 Enter the **show memory summary** command to take an initial memory snapshot of device SH1-97.

Step 2 Verify that SH1-97, SH1-100, and SH1-106 are running OSPF:

Commands

- **show running-config | begin router ospf**

Step 3 Verify that SH1-97 knows the route to network 172.31.1.96 via port-channel 13:

Commands

- **show ip route** *network*

Step 4 Verify that access list 30 is configured, on SH1-97, to deny network 172.31.1.96:

Commands

- **show access-lists** *acl_#*

Step 5 Configure the **distribute-list** command for access list 30 on SH1-97, OSPF process ID 1:

Commands

- *no* **distribute-list** *acl_#* **in** *interface*

Step 6 Verify that the route to the 172.31.1.96 network is no longer known via Po13 in SH1-97:

Note SH1-97 has connections to other routers running OSPF and, so, may learn the route to that network via another interface.

Commands

- **show ip route** *network*

Step 7 Remove the **distribute-list** command configured on SH1-97:

Commands

- *no* **distribute-list** *acl_#* **in** *interface*

Step 8 Verify that the route to the 172.31.1.96 network is, once again, in the routing table of SH1-97, known via port-channel 13:

Commands

- **show ip route** *network*

Step 9 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results for device SH1-97.

Expected Results

We expect OSPF to filter out routes using the **distribute-list** command.

Results

Table 44 shows the OSPF filtering test results.

Table 44 OSPF Filtering Test Results

Tests	Results
Filtering	Pass

OSPF Redistribution

This test verified that redistribution of EIGRP into OSPF functioned properly.

Test Plan

The procedure used to perform the Layer 3 OSPF redistribution test follows.

Step 1 Use an SNMP monitoring utility to begin monitoring memory and CPU utilization for devices SH1-101 and SH1-102.

Step 2 Verify that SH1-100 has OSPF configured (process ID 1) and that SH1-101 is running EIGRP for autonomous system (AS) 1320:

Commands

- **show running-config | begin ospf**
- **show running-config | begin eigrp**

Step 3 Display output from the routing table of SH1-101 prior to redistribution:

Commands

- **show ip route summary**

Step 4 Display output from the OSPF routing table of SH1-100:

Commands

- **show ip route summary**

Step 5 Redistribute EIGRP routes into OSPF 1 using the **redistribute eigrp 1320 subnets** command:

Commands

- **redistribute ospf 1 metric 100000 10 255 1 1500 match internal route-map OSPF2EIGRP**
- **show route-map OSPF2EIGRP**
- **show access-lists 17**

Step 6 Display output from the routing table of SH1-101 after redistribution:

Commands

- **show ip route summary**

Step 7 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results for devices SH1-101 and SH1-102.

Expected Results

We expect redistribution of EIGRP into OSPF to function properly.

Results

Table 45 shows the OSPF redistribution test results.

Table 45 OSPF Redistribution Test Results

Tests	Results
OSPF Redistribution	Pass

OSPF Topology Database

This test verified that the OSPF topology database functioned correctly.

Test Plan

The procedure used to perform the Layer 3 OSPF topology database test follows.

Step 1 Verify that the OSPF database was correct using the **show ip ospf database database-summary** command. Each area should have a router LSA (type 1) for each router, a network LSA (type 2) for each VLAN segment and a loopback interface, a summary net LSA (type 3), a summary ASBR LSAs describing routes to an ASBR (type 4), and autonomous system external LSAs (type 5).

Note Pagent injection of 30k in area 3 and area 4 were turned off.

Results Summary

Note On the summary net LSA, each router LSA in the equation signifies the loopback or OOB VLAN from which the router ID was taken. These stub networks are included in the summary net LSAs, but not the network LSAs.

Area 0

Table 46 Area 0 Database Summary Information

LSA	Expected	Found	Passed	Notes
Router	4	4	Yes	4 routers in area 0
Network	6	6	Yes	6 transit networks
Summary Net	133	133	Yes	All networks in area 1 and 2
Summary ASBR	1	1	Yes	SH1-100 was the only ASBR configured for this test

Area 1

Table 47 Area 1 Database Summary Information

LSA	Expected	Found	Passed	Notes
Router	8	8	Yes	8 routers in area 1
Network	12	12	Yes	12 transit networks between devices in area 1
Summary Net	82	82	Yes	All networks in area 0 and 2
Summary ASBR	1	1	Yes	SH1-100 was the only ASBR configured for this test

Area 2

Table 48 Area 2 Database Summary Information

LSA	Expected	Found	Passed	Notes
Router	4	4	Yes	4 routers in area 2
Network	4	4	Yes	4 transit networks
Summary Net	142	142	Yes	All networks in area 0 and 1
Summary ASBR	1	1	Yes	SH1-100 was the only ASBR configured for this test

Note An ABR in any area will create a type 4 LSA in that area for an ASBR in that area, which was expected behavior in case that specific area gets partitioned.

Expected Results

We expect the OSPF topology database to function correctly.

Results

Table 49 shows the OSPF topology database test results.

Table 49 OSPF Topology Database Test Results

Tests	Results
OSPF Topology Database	Pass

Border Gateway Protocol

Border Gateway Protocol (BGP) is an exterior gateway protocol designed to exchange network reachability information with other BGP systems in other autonomous systems. BGP exchanges routing information in the form of routing updates. An update includes a network number, a list of autonomous systems that the routing information has passed through (the autonomous system path), and a list of other path attributes.

The following tests were performed:

- Scale to Ten BGP Neighbors in Core, page 82
- Route Redistribution, page 83
- BGP Neighbor Flap, page 83

Scale to Ten BGP Neighbors in Core

This test verified that no memory leaks or unexpected CPU load occurred with eight BGP neighbors in the core (four core routers and four Pagent neighbors) and a total of 100K BGP routes, 20K OSPF routes, and 10K EIGRP routes.

Test Plan

The procedure used to perform the Layer 3 scale to ten BGP neighbors in core test follows.

Step 1 Using Pagent, IXIA, and the 4 core switches (SH1-97, SH1-98, SH1-99, and SH1-100), inject 130K routes—100K BGP, 20K OSPF, and 10K EIGRP routes.

Step 2 Monitor CPU and memory.

Expected Results

We expect no memory leaks or unexpected CPU load to occur with 8 BGP neighbors in the core.

Results

Table 50 shows scale to ten BGP neighbors in core test results.

Table 50 Scale to Ten BGP Neighbors in Core Test Results

Tests	Results
Scale to Ten BGP Neighbors in Core	Pass

Route Redistribution

This test verified that route redistribution worked correctly between BGP and both OSPF and EIGRP.

Test Plan

The procedure used to perform the Layer 3 BGP route redistribution test follows.

Step 1 Configure redistribution from BGP into OSPF.

Add the **redistribute ospf 1 metric 100** command to SH1-100.

Step 2 Apply an inbound distribute list using access list 16 under router OSPF configuration on SH1-97 to block this route from being known by OSPF. Verify that the route is now known via BGP.

Step 3 Configure redistribution in BGP for EIGRP.

Step 4 Apply the route map on SH1-100 based on access list 18, which allows only the even subnets under 200.1.0.0:

Step 5 Leaving the EIGRP redistribution in place, add redistribution for OSPF to test redistribution in BGP for both OSPF and EIGRP.

Step 6 Put a distribute list on SH1-97 to block OSPF learned route 140.68.240.0.

Expected Results

We expect route redistribution to function correctly between BGP and both OSPF and EIGRP.

Results

Table 51 shows BGP route redistribution test results.

Table 51 BGP Route Redistribution Test Results

Tests	Results
Route Redistribution	Pass

BGP Neighbor Flap

This test verified that a flapping nondampened BGP peer did not cause any memory leaks or prolonged high CPU utilization, and that the device under test (DUT) functioned properly after the peer stopped flapping. With BGP routes being fed into devices SH1-97 and SH1-98, this procedure simulated constant flapping of those BGP neighbors. Specifically, the Pagent device rtp-wbu-te-p4 was feeding 35,000 BGP routes into SH1-97 from interface fa1/0. The Pagent configuration on this interface was modified to enable flapping and the test was run for a period of 8 hours.

Test Plan

The procedure used to perform the Layer 3 BGP neighbor flap test follows.

Step 1 Using an SNMP monitoring utility, monitor memory and CPU utilization on core routers SH1-97, SH1-98, SH1-99, and SH1-100.

Commands

- **show processes memory**

Step 2 Verify that SH1-97 is an eBGP peer with a Pagent router. The Pagent router is sending updates for 35,000 routes from autonomous system 10:

Commands

- **show ip bgp summary**

Step 3 Log in to the Pagent device and use the **lne bgp interface fa1/0** command (BGP route feed) to flap every 30 to 60 seconds followed by a nonflapping period of 60 to 120 seconds. Turn router flap on and allow flapping to continue for a period of 8 hours:

Commands

- **router-flap on**

Step 4 Check memory, CPU utilization, and BGP summary before and after running the flapping test for 8 hours:

Commands

- **show processes memory**
- **show processes cpu**
- **show ip bgp summary**

Expected Results

We expect no memory or CPU issues.

Results

Table 52 shows BGP neighbor flap test results.

Table 52 BGP Neighbor Flap Test Results

Tests	Results
BGP Neighbor Flap	Pass

Hot Standby Routing Protocol

For IP, the Hot Standby Router Protocol (HSRP) allows one router to automatically assume the function of the second router if the second router fails. HSRP is particularly useful when the users on one subnet require continuous access to resources in the network.

The following tests were performed:

- Basic HSRP, page 85
- HSRP Failover, page 86
- HSRP Failover Using Fast Timers, page 87

Basic HSRP

This test verified the basic functionality of HSRP. SH1-109 and SH1-110 were configured as HSRP routers for VLANs 10 to 20, peered to each other through Layer 2 switch Dista-2. This test verified that the configuration on each Layer 3 switch produced the desired results. The unicast traffic stream that was running in the background from Dista-2 to SH1-105 was used to prove that traffic chose the active HSRP router.

Test Plan

The procedure used to perform the Layer 3 Basic HSRP test follows.

Step 1 Verify that SH1-109 and SH1-110 are configured with HSRP on their interfaces for VLANs 10 through 20:

Commands

- **show running-config | begin interface Vlan10**

Step 2 Verify that trunking is configured for VLANs 10 to 20 on the links between SH1-109 and Dista-2, and SH1-110 and Dista-2. The trunking of these VLANs is what allows the two HSRP routers to negotiate for status:

Commands

- **show interfaces trunk module**

Step 3 Verify that SH1-109 and SH1-110 are running multiple HSRP groups on VLAN 10 through VLAN 20:

Commands

- **show standby brief**

Expected Results

We expect basic HSRP functionality to function according to specifications.

Results

Table 53 shows the basic HSRP test results.

Table 53 Basic Hot Standby Routing Protocol Test Results

Tests	Results
Basic HSRP	Pass

HSRP Failover

This test verified HSRP failover when a link was down. This test also verified that the HSRP **preempt** command worked when the link returned to an up/up state, if the interface was configured with a higher priority than the currently active router interface in the same HSRP group.

Test Plan

The procedure used to perform the Layer 3 HSRP failover test follows.

Step 1 Use the **show memory summary** command to take a memory snapshot of devices SH1-101 and SH1-102, and use the **show processes cpu** command to begin tracking CPU utilization.

Step 2 Verify that the links between SH1-101 and Dista-1 and the links between SH1-102 and Dista-1 are trunks and that VLANs 40 to 50 are active within them. Check interfaces g7/1-2 on the SH1-101 side, and g7/6 and g7/8 on the SH1-102 side:

Commands

- **show interfaces trunk module** *module*

Step 3 Begin an IXIA unicast traffic stream on Dista-1 6/5 (IXIA port 4/2). The destination for this stream is Dista-2 6/2 (IXIA port 5/2).

Step 4 Verify that HSRP is active on SH1-102 for VLAN 45 (this is the VLAN on which traffic will be sent by IXIA), and standby on SH1-101 for VLAN 45. Verify that traffic is being sent to the active HSRP router:

Commands

- **show standby brief**
- **show interfaces Po10 counters**

Step 5 On SH1-102, shut down the port-channel 10 interface which should force the unicast traffic stream to fail over to SH1-101. Verify that this failover does happen, and measure the time (using traffic loss) required for that switchover to occur:

Commands

- **show standby brief**

Step 6 Bring the port-channel 10 interface back up on SH1-102. Verify that HSRP on SH1-102 preempts SH1-101 and once again becomes the active HSRP router. Determine the amount of time (using traffic loss) that required for this action to occur:

Commands

- **show standby brief**

Step 7 Perform a series of nine more port cycles on the port-channel 10 interface of SH1-102. Measure the average amount of time required for switchover following interface shutdown (include the results from Step 4).

Step 8 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results.

Expected Results

We expect HSRP failover when a link is down.

Results

Table 54 shows the HSRP failover test results.

Table 54 Hot Standby Routing Protocol Failover Test Results

Tests	Results
HSRP Failover	Pass

HSRP Failover Using Fast Timers

This test verified HSRP failover when a link is down, using fast timers. This test also verified that the HSRP **preempt** command worked when the link returned to an up/up state, if the interface was configured with a higher priority than the currently active router interface in the same HSRP group.

This test is similar to the HSRP failover test with standard timers. However, the hello timer and hold timer for VLAN45 on SH1-102 and SH1-102 were reduced to 1 and 3 seconds, respectively. By default, they are 3 and 10 seconds.

Test Plan

The procedure used to perform the Layer 3 HSRP failover (using fast timers) test follows.

Step 1 Use the **show memory summary** command to take a memory snapshot of devices SH1-101 and SH1-102, and use the **show processes cpu** command to begin tracking CPU utilization.

Step 2 Verify that the links between SH1-101 and Dista-1 and the links between SH1-102 and Dista-1 are trunks and that VLANs 40 to 50 are active within them. Check interfaces g7/1-2 on the SH1-101 side, and g7/6 and g7/8 on the SH1-102 side:

Commands

- **show interfaces trunk module** *module*

Step 3 Begin an IXIA unicast traffic stream on Dista-1 6/5 (IXIA port 4/2). The destination for this stream is Dista-2 6/2 (IXIA port 5/2).

Step 4 Verify that HSRP is active on SH1-102 for VLAN 45 (this is the VLAN on which traffic will be sent by IXIA), and standby on SH1-101 for VLAN 45. Verify that traffic is being sent to the active HSRP router:

Commands

- **show standby brief**
- **show interfaces Po10 counters**

Step 5 Configure the fast timers on VLAN45 of SH1-101 and SH1-102 and confirm that they have been changed:

Commands

- **standby timers 1 3**
- **show standby vlan 45**

Step 6 On SH1-102, shutdown interface port-channel 10, which should force the unicast traffic stream to fail over to SH1-101. Verify that this failover does happen, and measure the time (using traffic loss) required for that switchover to occur:

Commands

- **show standby brief**

Step 7 Bring the port-channel 10 interface back up on SH1-102. Verify that HSRP on SH1-102 preempts SH1-101 and once again becomes the active HSRP router. Determine the amount of time (using traffic loss) required for this action to occur:

Commands

- **show standby brief**

Step 8 Perform a series of nine more port cycles on the port-channel 10 interface of SH1-102. Measure the average amount of time required for switchover following interface shutdown (include the results from Step 4).

Step 9 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results.

Expected Results

We expect HSRP failover when a link is down.

Results

Table 55 shows the HSRP failover using fast timers test results.

Table 55 HSRP Failover Using Fast Timers Test Results

Tests	Results
HSRP Failover Using Fast Timers	Pass

Enhanced Interior Gateway Routing Protocol

The Enhanced Interior Gateway Routing Protocol (EIGRP) is an enhanced version of the IGRP protocol developed by Cisco Systems. Enhanced IGRP uses the same distance vector algorithm and distance information as IGRP. However, the convergence properties and the operating efficiency of Enhanced IGRP have improved significantly over IGRP.

The convergence technology is based on research conducted at SRI International and employs an algorithm referred to as the Diffusing Update Algorithm (DUAL). This algorithm guarantees loop-free operation at every instant throughout a route computation and allows all devices involved in a topology change to synchronize at the same time. Routers that are not affected by topology changes are not involved in recomputations.

The following tests were performed:

- EIGRP Summarization, page 89
- EIGRP Redistribution, page 89

EIGRP Summarization

This test verified manual EIGRP summarization by using the **ip summary-address eigrp** *as-number* interface configuration command.

There are a few /24 networks directly connected to SH1-101 and SH1-102 that can be summarized as /22 or /21 on the port-channels of SH1-100 (port channels 14 and 15).

- 172.31.20.0/24 through 172.31.23.0/24, summarized as 172.31.20.0/22
- 172.31.24.0/24 through 172.31.30.0/24, summarized as 172.31.24.0/21

The procedure used to perform the Layer 3 EIGRP summarization test follows.

Test Plan

Step 1 Verify that routes are not summarized:

Step 2 Add the **ip summary-address eigrp** command on the following ports:

- Interface port-channel 14 and port-channel 15 on SH1-100

Step 3 Verify the output of the **show ip route** command on both the distributions. Verify that the network shows up as a /22 and not a /24:

Expected Results

We expect EIGRP summarization to function according to specifications.

Results

Table 56 shows the EIGRP summarization test results.

Table 56 EIGRP Summarization Test Results

Tests	Results
EIGRP Summarization	Pass

EIGRP Redistribution

This test verified that EIGRP route redistribution worked correctly, with and without access lists and route map filtering.

Five /24 loop backs were directly connected to SH1-97. These loop backs were to be redistributed into EIGRP. After loop backs were redistributed into the EIGRP domain, the loopback addresses were filtered to allow only the odd subnets.

Test Plan

The procedure used to perform the Layer 3 EIGRP redistribution test follows.

Step 1 Display the output of the routing table pertaining to EIGRP routes prior to redistribution:

Step 2 Configure redistribution from OSPF into EIGRP with no filtering:

Step 3 Compare current routing table with initial (gathered at Step 1) and verify that routes exist in the EIGRP domain:

Step 4 Configure redistribution from OSPF into EIGRP using a route map to filter:

- Remove the **redistribution** command.
- Display the route entry.

Expected Results

We expect EIGRP redistribution to function according to specifications.

Results

Table 57 shows the EIGRP redistribution test results.

Table 57 EIGRP Redistribution Test Results

Tests	Results
EIGRP Redistribution	Pass

Network Management Features

Network management feature testing for Safe Harbor involves the following sections:

- Simple Network Management Protocol, page 91
- TACACS, page 93

Simple Network Management Protocol

The Simple Network Management Protocol (SNMP) system consists of an SNMP manager, an SNMP agent, and a Management Information Base (MIB).

SNMP is an application-layer protocol that provides a message format for communication between SNMP managers and agents.

The following tests were performed:

- Basic Functionality Shut/No Shut Interface, page 91
- Protos Request Application, page 92

Basic Functionality Shut/No Shut Interface

This test verified that basic SNMP functionality on the DUT functioned according to specification.

Test Plan

The procedure used to perform the SNMP Basic Functionality Shut/No Shut Interface test follows.

Step 1 Verify the SNMP configuration on SH1-101:

Commands

- **show running-config | include snmp**

Step 2 Shut down the interface VLAN 40 on SH1-101.

Step 3 Remove the shut down with the **no shut** command configured for the interface VLAN 40 on SH1-101.

Step 4 Verify that the traps are received by a machine that is set up as the SNMP trap receiver. Display the output from the log files of that machine.

Expected Results

We expect basic functionality shut/no Shut interface to function according to specifications.

Results

Table 58 shows the basic functionality shut/no Shut interface test results.

Table 58 Basic Functionality Shut/No Shut Interface Test Results

Tests	Results
Basic Functionality Shut/No Shut Interface	Pass

Protos Request Application

PROTOS is an technology industry project standard that researches different approaches of testing implementations of protocols using functional testing methods. The goal is to support proactive elimination of faults with information security implications.

Protos testing involves sending erroneous SNMP packets to the DUT and running these tests on all permutations of supervisor and MSFC. This is a negative test, so the only verification is that the DUT was unaffected.

This test verified that there were no SNMP vulnerabilities in the Native Cisco IOS code from the perspective of malformed SNMP packets or intentional hacks. This test did not check for existing MIBs and did not verify the results from polling every possible MIB.

Test Plan

The procedure used to perform the SNMP Protos request application test follows.

Step 1 Verify that the SNMP **read-only** password is public on both the supervisor and MSFC of SH1-103, SH1-105, and SH1-107.

Step 2 Monitor CPU and Memory of each router. Monitor the console of each router for failures, tracebacks, or infinite pauses.

Note Be sure to enable the **logging console** command while monitoring, or display the log when the scripts finish running.

Step 3 Execute the proper TCL test scripts on each host.

Step 4 After tests are completed, verify that CPU and memory did not spike during these tests, and that the router did not fail, pause indefinitely, or give tracebacks. Router configuration should not change as a result of these tests.

Expected Results

We expect the Protos request application to function according to specifications.

Results

Table 59 shows the Protos request application test results.

Table 59 Protos Request Application Test Results

Tests	Results
Protos Request Application	Pass

TACACS

Login authentication increases the security of the system by keeping unauthorized users from guessing the password. The user is limited to a specific number of attempts to successfully log in to the switch. If the user fails to authorize the password, the system delays accesses and captures the user ID and the IP address of the station in the syslog and in the SNMP trap.

The following test was performed:

- Verify User Authentication, page 93

Verify User Authentication

This test verified that the Terminal Access Controller Access Control System (TACACS) login authentication worked correctly and identified TACACS server for the DUT to peer with. The DUT was configured to point to that server for authentication information. Telnet was used to reach the DUT and verify the authentication function.

Test Plan

The procedure used to perform the TACACS user authorization test follows.

Step 1 Verify that SH1-101 is configured for TACACS authentication:

Commands

- **aaa new-model**
- **aaa authentication login default enable**
- **aaa authentication login sh1-testcase group tacacs+ enable**
- **tacacs-server host 172.18.177.132**
- **show running-config | include aaa|tacacs**

Step 2 Turn on TACACS authentication for Telnet sessions:

Commands

- **login authentication sh1-testcase**

Step 3 Verify that user authentication works by logging in to SH1-101:

Step 4 Remove TACACS authentication for Telnet sessions:

Commands

- **login authentication default**

Expected Results

We expect TACACS login authentication to function correctly.

Results

Table 60 shows the verify user authorization test results.

Table 60 Verify User Authorization Test Results

Tests	Results
Verify User Authentication	Pass

Miscellaneous Features

Miscellaneous features tested for Safe Harbor are described in the following sections:

- NTP Basic Functionality, page 94
- Syslog Basic Functionality, page 95
- User Datagram Protocol Broadcast Flooding, page 96
- System Upgrade, page 98

NTP Basic Functionality

Network Time Protocol (NTP) synchronizes timekeeping among a set of distributed time servers and clients. This synchronization allows events to be correlated when system logs are created and other time-specific events occur.

An NTP server must be accessible by the client switch. NTP runs over User Datagram Protocol (UDP), which runs over IP. NTP is extremely efficient; no more than one packet per minute is necessary to synchronize two machines to within a millisecond of one another.

This test verified basic functionality. It used a local Sun server (IP address: 172.18.177.132) as the NTP server. SH1-104 was configured as the NTP client.

Test Plan

The procedure used to perform the NTP functionality test follows.

Step 1 Verify that the server with IP address 172.18.177.132 is configured as the NTP server on SH1-104:

Commands

- **show running-config | include ntp**

Step 2 Verify that the association between the NTP server and the client is active:

Commands

- **show ntp associations**

Step 3 Verify that the NTP status is "synchronized" and that the time and date shown on SH1-104 are the same as those shown on the server (NTP is working):

Commands

- **show ntp status**
- **show clock**

- **date** (on server—UNIX command)

Expected Results

We expect synchronized timekeeping among a set of distributed time servers and clients.

Results

Table 61 shows the basic NTP functionality test results.

Table 61 Basic NTP Functionality Test Results

Tests	Results
NTP Basic Functionality	Pass

Syslog Basic Functionality

The System Log (syslog) protocol provides a transport to allow a machine to send event notification messages across IP networks to event message collectors, which are also known as syslog servers.

This test verified Syslog functionality.

Test Plan

The procedure used to perform the basic syslog functionality test follows.

Step 1 Verify that syslog is configured on SH1-101 and that the designated server is 172.18.177.132:

Commands

- **show running-config | include logg**

Step 2 Enable the **terminal monitor** command on SH1-101 and enable IP PIM debugging:

Commands

- **terminal monitor**
- **debug ip pim**

Step 3 Once you receive debug messages on SH1-101, turn debugging off:

Commands

- **undebug all**

Step 4 Display output from the syslog server. Compare it to messages received on SH1-101:

Commands

- **tail local7 | grep 10.194.17.110** (on server)

Expected Results

We expect an established transport to send event notification messages across IP networks to event message collectors.

Results

Table 62 shows the basic syslog functionality test results.

Table 62 Basic Syslog Functionality Test Results

Tests	Results
Syslog Basic Functionality	Pass

User Datagram Protocol Broadcast Flooding

A *broadcast* is a data packet that is destined for multiple hosts. Broadcasts can occur at the data link layer and the network layer. Data-link broadcasts are sent to all hosts attached to a particular physical network. Network layer broadcasts are sent to all hosts attached to a particular logical network. TCP/IP supports the following types of broadcast packets:

- All ones—By setting the broadcast address to all ones (255.255.255.255), all hosts on the network receive the broadcast.
- Network—By setting the broadcast address to a specific network number in the network portion of the IP address and setting all ones in the host portion of the broadcast address, all hosts on the specified network receive the broadcast. For example, when a broadcast packet is sent with the broadcast address of 131.108.255.255, all hosts on network number 131.108 receive the broadcast.
- Subnet—By setting the broadcast address to a specific network number and a specific subnet number, all hosts on the specified subnet receive the broadcast. For example, when a broadcast packet is set with the broadcast address of 131.108.4.255, all hosts on subnet 4 of network 131.108 receive the broadcast.

Because broadcasts are recognized by all hosts, a significant goal of router configuration is to control unnecessary proliferation of broadcast packets. Cisco routers support two kinds of broadcasts: *directed* and *flooded*. A directed broadcast is a packet sent to a specific network or series of networks, whereas a flooded broadcast is a packet sent to every network. In IP internetworks, most broadcasts take the form of User Datagram Protocol (UDP) broadcasts.

The following test was performed:

- UDP Broadcast Flooding, page 96

UDP Broadcast Flooding

This test verified many functional aspects of UDP broadcast flooding. An IXIA broadcast stream was used for each aspect of the procedure. First, the broadcast stream was sent into the network at a single device. It was verified that this device forwarded the broadcast out all physical interfaces. Next the device was configured to use an IP helper address, which caused the IP destination portion of the header, originally a broadcast address, to be replaced by a unicast address, destined for the IP helper address.

The IXIA streams were configured with UDP port 53 (DNS). The next step of the test verified the functionality of the **no ip forward-protocol** command, which should halt the forwarding of any broadcasts using the protocol specified (in this case DNS).

Test Plan

The procedure used to perform the UDP broadcast flooding test follows.

Step 1 Use the **show memory summary** command to take a memory snapshot of device SH1-101, and use the **show processes cpu** command to begin tracking CPU utilization.

Step 2 Verify that the **ip helper-address** and **no ip forward-protocol** commands are not configured on the device, for the beginning of this procedure:

Commands

- **show running-config | include helper-address|forward-protocol**

Step 3 Begin the broadcast IXIA stream on Ix-12/1 (SH1-101 g7/16). The stream should have a length of 74 bytes and be sent at a rate of 5000 packets per second. The broadcast stream has a DMAC of FFFF.FFFF.FFFF and a DIP of 255.255.255.255. This stream will come in on VLAN 40 and should be broadcast out all interfaces carrying VLAN 40. Verify that it is.

a. Determine which trunks are carrying VLAN40. There is only one trunk, port-channel 10:

b. Determine which interfaces on the device are configured as access ports with VLAN40. Only g7/16 is configured in such a way. This is the interface to IXIA.

c. The broadcast traffic stream is coming in on interface g7/16:

Step 4 Configure an IP helper address of 172.31.64.112 on the VLAN40 interface of SH1-101. This IP address belongs to IXIA port 11/2, also connected to SH1-101 via interface g3/16.

Step 5 Verify that the traffic coming in is now being sent out interface g3/16, as unicast traffic:

Step 6 Configure the **no ip forward-protocol udp domain** command globally on SH1-101, which will cause the router to stop forwarding any UDP traffic with an Layer 4 port of 53 (DNS). Verify that traffic stops being sent out interface g3/16:

Step 7 Allow IP forwarding of DNS packets again, and verify that unicast traffic is being sent to the IP helper address again:

Step 8 Stop the IXIA traffic for about 2 minutes. Reconfigure the IXIA stream to send at a rate of 2500 pps, instead of the previous 5000 pps.

Step 9 Send traffic for about 2 minutes, enough to get a measure of what halving the traffic rate does for helping CPU utilization.

Step 10 Stop the IXIA traffic for about 2 minutes. Reconfigure the IXIA stream to send at a rate of 5000 pps, but with a packet size of 500 bytes (was 100 bytes previously).

Step 11 Send traffic for about 2 minutes, enough to get a measure of what increasing the packet size does for the CPU utilization.

Step 12 Use the **show memory summary** and **show processes cpu** commands to verify memory and CPU utilization results.

Expected Results

We expect devices to forward and halt all appropriate broadcasts.

Results

Table 63 shows the UDP broadcast flooding test results.

Table 63 UDP Broadcast Flooding Test Results

Tests	Results
UDP Broadcast Flooding	Pass

System Upgrade

This test measured the basic ability of the code to boot correctly. The boot process was followed in each case from the console and screened for any errors, and so on. The output shows three separate sections, one for each supervisor and MSFC combination: Sup11, Sup12, and Sup22.

This test verified that the Cisco IOS upgrade process worked correctly.

Test Plan

The procedure used to perform the system upgrade test follows.

Step 1 On Sup 11, verify that SH1-106 is running the old Native Cisco IOS image:

Commands

- **show version**

Step 2 Verify that the Sup11 image under test is on the proper file devices:

Commands

- **dir** *device*
- **dir slave** *device*

Step 3 Verify that the boot string points to the proper device and filename for the boot image. Save any changes to NVRAM when done:

Commands

- **show running-config | include boot**
- *no* **boot system flash** *old/new image*

Step 4 Either power-cycle the device or perform a reset on the standby supervisor followed by a reload, to verify that both primary and standby supervisors reload with the new image. Report any error messages seen during reload:

Commands

- **hw-module module** *module* **reset**
- **reload**

Step 5 Verify that both supervisors come online successfully and that the new image is running:

Commands

- **show module**

- **show version**

Step 6 On Sup12, verify that SH1-102 is running the old Native Cisco IOS image:

Commands

- **show version**

Step 7 Verify that the Sup12 image under test is on the proper file devices:

Commands

- **directory** *device*
- **directory slave** *device*

Step 8 Verify that the boot string points to the proper device for the boot image. It is not necessary to specify a filename, if the desired boot image is the first listed under that device directory. Save any changes to NVRAM when done:

Commands

- **show running-config | include boot**
- *no* **boot system flash** *old/new image*

Step 9 Either power-cycle the device or perform a reset of the standby supervisor followed by a reload, to verify both primary and standby supervisors reload with the new image. Report any error messages seen during reload:

Commands

- **hw-module module** *module* **reset**
- **reload**

Step 10 Verify that both supervisors come online successfully and that the new image is running:

Commands

- **show module**
- **show version**

Step 11 On Sup22, verify that SH1-102 is running the old Native Cisco IOS image:

Commands

- **show version**

Step 12 Verify that the new Sup22 image under test is on the proper file devices:

Commands

- **directory** *device*
- **directory slave** *device*

Step 13 Verify that the boot string points to the proper device for the boot image. It is not necessary to specify a filename, if the desired boot image is the first listed under that device directory. Save any changes to NVRAM when done:

Commands

- **show running-config | include boot**
- *no* **boot system flash** *old/new image*

Step 14 Either power-cycle the device or perform a reset on the standby supervisor followed by a reload, to verify that both primary and standby supervisors reload with the new image. Report any error messages seen during reload:

Commands

- **hw-module module** *module* **reset**
- **reload**

Step 15 Verify that both supervisors come online successfully and that the new image is running:

Commands

- **show module**
- **show version**

Expected Results

We expect the Cisco IOS system upgrade process to work properly and the code to boot correctly.

Results

Table 64 shows the system upgrade test results.

Table 64 System Upgrade Test Results

Tests	Results
System Upgrade	Pass