Cisco Data Center Assurance Program (DCAP) 2.0

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco Data Center Assurance Program (DCAP) 2.0
© 2007 Cisco Systems, Inc. All rights reserved.

Preface i-i

Data Center Assurance Program (DCAP) i-i

This DCAP Book i-ii

Chapter 1: LAN (Layer 2-3) Infrastructure i-ii

Chapter 2: LAN (Layer 4-7) Services i-ii

Chapter 3: Storage Area Networking (SAN) i-ii

Chapter 4: Global Site Selector (GSS) i-ii

Chapter 5: Wide Area Application Services (WAAS) i-iii

CHAPTER 1 **Overview** 1-1

DCAP Testing Methodology 1-1

DCAP Testing Overview 1-2

DCAP Topology 1-4

CHAPTER 2 **Layer 2-3 Infrastructure** 2-1

Layer 2 Topology 2-2

Layer 3 Topology 2-4

Layer 2-3 Test Results Summary 2-5

Layer 2-3 DDTS Summary 2-7

Layer 2-3 Test Cases 2-7

Baseline 2-7

Topology Baseline 2-8

Topology Baseline 2-8

Device Management 2-9

Upgrade of Supervisor 720 System in Core Layer 2-10

Upgrade of Supervisor 720 System in Aggregation Layer 2-11

Upgrade of Supervisor 720 System in Access Layer 2-11

Upgrade of Catalyst 4948-10GE System in Access Layer 2-12

Upgrade of Content Switching Module (CSM) 2-13

Upgrade of Firewall Services Module (FWSM) 2-14

Upgrade of Secure Socket Layer Services Module (SSLSM) 2-15

General On-Line Diagnostics (GOLD) 2-16

SNMP MIB Tree Walk 2-18

Local SPAN 2-18

Remote SPAN (rSPAN) 2-19
Device Access 2-21
Repeated Logins Using SSH Version 1 2-21
Repeated Logins Using SSH Version 2 2-22
CLI Functionality 2-23
CLI Parser Functionality Using SSHv1 2-23
CLI Parser Functionality Using SSHv2 2-24
CLI Parser Functionality Using SSHv1 on 4948 2-24
CLI Parser Functionality Using SSHv2 on 4948 2-25
Security 2-25
Malformed SNMP Polling 2-25
Malformed SSH Packets 2-26
NMAP Open Port Scan 2-27
Traffic Forwarding 2-28
Zero Packet Loss 2-28
Distributed FIB Consistency 2-29
Layer 2 Protocols 2-30
Link Aggregation Control Protocol (LACP) 2-31
LACP Basic Functionality 2-31
LACP Load Balancing 2-32
Trunking 2-33
802.1q Trunking Basic Functionality 2-33
Spanning Tree 2-34
Rapid PVST+ Basic Functionality 2-34
Root Guard 2-36
Unidirectional Link Detection (UDLD) 2-38
UDLD Detection on 10GE Links 2-38
Layer 3 Protocols 2-39
Hot Standby Router Protocol (HSRP) 2-39
HSRP Basic Functionality 2-40
Open Shortest Path First (OSPF) 2-41
OSPF Route Summarization 2-41
OSPF Database Verification 2-42
Negative Testing 2-43
Hardware Failure 2-43
Access Layer Supervisor Failover Using SSO with NSF 2-44
Repeated Reset of Standby Supervisor in Access Layer 2-45
Reset of Aggregation Layer Device dca-agg-1 2-46
Reset of Aggregation Layer Device dca-agg-2 2-47

Reset of Core Layer Device dca-core-1 2-49
Reset of Core Layer Device dca-core-2 2-49
Spanning-Tree Primary Root Failure and Recovery 2-50
HSRP Failover with Fast Timers 2-54
HSRP Recovery From System Failure 2-57
Failure of EtherChannel Module on dca-agg-1 2-58
Failure of EtherChannel Module on dca-agg-2 2-59
Link Failure 2-61
Failure of Single Bundled 10-Gigabit Ethernet Link Between dca-agg-1 and dca-agg-2 2-61
Failure of 10 Gigabit Ethernet Link Between dca-agg-1 and dca-acc-6k-2 2-63
Failure of 10 Gigabit Ethernet Link Between dca-agg-2 and dca-acc-6k-2 2-63
Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-agg-1 2-64
Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-agg-2 2-65
Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-core-2 2-66
Failure of 10-Gigabit Ethernet Link Between dca-core-2 and dca-agg-1 2-66
Failure of 10-Gigabit Ethernet Link Between dca-core-2 and dca-agg-2 2-67
Network Resiliency Test 2-68

CHAPTER 3 **Layer 4-7 Services** 3-1

Integrated Bundle Vs. Service Switch Models 3-1
Traffic Pathways Through the Bundle 3-3
Integrated Bundle Configuration 3-5
Service Switch Configuration 3-8
Layer 4-7 Test Results Summary 3-9
Layer 4-7 DDTS Summary 3-10
Layer 4-7 Test Cases 3-10
Aggregation Switch Bundle Testing 3-11
CSM/FWSM Integration 3-11
Active FTP Through FWSM and CSM 3-11
Passive FTP Through FWSM and CSM 3-14
ICMP to a CSM Layer 3 and Layer 4 Vserver 3-16
DNS Query Through CSM and FWSM 3-18
FWSM and CSM Layer 4 SYN Attack 3-20
Idle Timeout UDP 3-22
CSM/SSLSM Integration 3-23
Backend SSL 3-24
SSL Sticky 3-28
URL Rewrite 3-30
Redundancy 3-32

FWSM Redundancy 3-32
CSM Redundancy 3-34
SSLM Reset 3-37
Sorry Server 3-41
HSRP Failover 3-42
Service Switch Bundle Testing 3-44
CSM/FWSM Integration 3-44
Active FTP Through FWSM and CSM 3-44
Passive FTP Through FWSM and CSM 3-47
ICMP to a CSM Layer 3 and Layer 4 Vserver 3-49
DNS Query Through CSM and FWSM 3-51
FWSM CSM Layer4 SYN Attack 3-53
Idle Timeout UDP 3-55
CSM/SSLSM Integration 3-56
Backend SSL 3-57
SSL Sticky 3-61
URL Rewrite 3-63
Redundancy 3-65
FWSM Redundancy 3-65
CSM Redundancy 3-67
SSLM Reset 3-70
Sorry Server 3-73
HSRP Failover 3-75

CHAPTER 4 **Storage Area Networking (SAN) 4-1**

SAN Topology 4-1
Transport Core 4-3
Test Results Summary 4-14
DDTS Summary 4-17
SAN Test Cases 4-17
Baseline 4-17
A.1: Device Check 4-18
Device Access—CLI and Device Manager 4-18
Device Hardware Check—CLI 4-19
Device Hardware Check—Device Manager 4-19
Device Network Services Check—CLI 4-20
Device Network Services Check—Device Manager 4-21
A.2: Infrastructure Check 4-22
Host and Storage Fabric Connectivity—NetApp 4-22

Host and Storage Fabric Connectivity—EMC 4-23
Intra-Fabric Connectivity 4-24
Topology Discovery—Fabric Manager 4-24
A.3: Host to Storage Traffic—EMC 4-25
Base Setup—VSANs EMC 4-25
Base Setup—Zoning EMC 4-26
Host To Storage IO Traffic—EMC 4-27
Replication FC Sync—EMC 4-28
Replication FCIP ASync—EMC 4-29
A.4: Host to Storage Traffic—NetApp 4-30
Base Setup—VSANs NetApp 4-31
Base Setup—Zoning NetApp 4-32
Host To Storage IO Traffic—NetApp 4-33
Replication FC-Sync—NetApp 4-34
Replication FCIP-Async—NetApp 4-35
Domain Parameters 4-36
Principal Switch Selection 4-36
FSPF Functionality 4-37
Basic FSPF Load Balancing 4-37
Path Selection—Cost Change on Equal Cost Paths 4-38
Primary Path Failure 4-39
Primary Path Removal—VSAN Remove 4-40
Fabric Extension 4-40
Async Replication—EMC 4-41
FCIP COMP 100Km EMC 4-41
FCIP ENCRP 100Km EMC 4-42
FCIP NONE 100Km EMC 4-43
FCIP WA 100Km EMC 4-44
FCIP WA COMP ENCRP 100Km EMC 4-46
FCIP Portchannel Failure 100Km EMC 4-47
Async Replication—NetApp 4-48
FCIP COMP 100Km NETAPP 4-48
FCIP ENCRP 100Km NETAPP 4-50
FCIP NONE 100Km NETAPP 4-51
FCIP WA 100Km NETAPP 4-52
FCIP WA COMP ENCRP 100Km NETAPP 4-53
FCIP Portchannel Failure 100Km NETAPP 4-54
Async Replication—HP 4-56
FCIP COMP 100Km HP 4-56
FCIP ENCRP 100Km HP 4-57

FCIP NONE 100Km HP 4-58
FCIP WA 100Km HP 4-59
FCIP WA COMP ENCRP 100Km HP 4-61
Sync Replication—EMC 4-62
FC Sync—DST=100Km, WA=OFF - EMC 4-62
FC Sync—DST=100Km, WA=ON - EMC 4-63
FC Sync—Portchannel Failure, DST=100Km - EMC 4-64
Sync Replication—NetApp 4-65
FC Sync—DST=100Km, WA=OFF - NetApp 4-66
FC Sync—DST=100Km, WA=ON - NetApp 4-67
FC Sync—Portchannel Failure, DST=100Km - NetApp 4-68
Sync Replication—HP 4-69
FC Sync—DST=100Km, WA=OFF - HP 4-69
FC Sync—DST=100Km, WA=ON - HP 4-70
Inter-VSAN Routing Functionality 4-71
Basic IVR Implementation 4-72
Basic IVR-NAT Implementation 4-72
Portchannel Functionality 4-73
Basic Portchannel Load Balancing 4-74
Multiple Link ADD to Group 4-74
Multiple Links Failure in Group 4-75
Multiple Links Remove to Group 4-76
Single Link Add to Group 4-77
Single Link Failure in Group 4-78
Single Link Remove from Group 4-79
Resiliency Functionality 4-80
EMC 4-80
Host Link Failure (Link Pull)—EMC 4-80
Host Link Failure (Port Shutdown)—EMC 4-81
Host Facing Module Failure (OIR)—EMC 4-82
Host Facing Module Failure (Reload)—EMC 4-83
NetApp 4-84
Host Link Failure (Link Pull)—NETAPP 4-84
Host Link Failure (Port Shutdown)—NETAPP 4-85
Host Facing Module Failure (OIR)—NETAPP 4-86
Host Facing Module Failure (Reload)—NETAPP 4-87
MDS 4-88
Active Crossbar Fabric Failover (OIR) 4-89
Active Supervisor Failover (OIR) 4-90
Active Supervisor Failover (Reload) 4-90

Active Supervisor Failover (Manual CLI) 4-91
Back Fan-Tray Failure (Removal) 4-92
Core Facing Module Failure (OIR) 4-93
Core Facing Module Failure (Reload) 4-94
Front Fan-Tray Failure (Removal) 4-95
Node Failure (Power Loss) 4-96
Node Failure (Reload) 4-97
Power Supply Failure (Cord Removal) 4-98
Power Supply Failure (Power Off) 4-99
Power Supply Failure (Removal) 4-100
Standby Supervisor Failure (OIR) 4-100
Standby Supervisor Failure (Reload) 4-101
Unused Module Failure (OIR) 4-102
Security Functionality 4-103
FC SP Authentication Failure 4-103
Port Security Basic Implementation 4-104
User Access—TACACS Basic Test 4-105
User Access—TACACS Servers Failure 4-106
End-to-End Functionality 4-107
HTTP Functionality 4-107
End-to-End Get 4-107
End-to-End Put 4-109

CHAPTER 5 **Global Site Selector (GSS)** 5-1
GSS Topology 5-2
Test Results Summary 5-3
GSS Test Cases 5-3
GSS Backup Restore 5-3
GSS DNS Request Processing 5-5
GSS DNS Static Proximity 5-10
GSS DNS Global Sticky 5-11
GSS DNS Keepalive Verification 5-13
GSS Load Balancing Methods 5-14

CHAPTER 6 **Wide Area Application Services (WAAS)** 6-1
WAAS Topology 6-2
WAAS Test Results Summary 6-3
WAAS DDTS Summary 6-4
WAAS Test Cases 6-5

Baseline 6-5
Device Management 6-5
SNMP MIB Walk WAE512 6-6
SNMP MIB Walk WAE7326 6-6
SNMP MIB Walk-WAE612 6-7
Reliability 6-7
Core Reload WAE7326 6-8
WAAS Edge Reload WAE512 6-9
CLI Functionality 6-9
Parser via Telnet 3845 6-10
WCCP 6-10
WCCPv2 on Core WAE7326 6-11
WCCPv2 on EDGE-WAE 512 6-11
WCCPv2 on Edge-3845 6-12
WCCPv2 on CORE-Sup720 6-14
NTP 6-15
NTP Configuration and Functionality 6-15
Authentication (Kerberos) 6-17
Core Kerberos Authentication WAE7326 6-17
Edge Kerberos Authentication WAE512 6-18
Optimization (DRE/TFO/LZ) 6-20
WAFS 6-20
WAFS Configuration Verification 6-21
CIFS/Performance 6-22
CIFS Windows Acceleration Verification WAE512 6-23
CIFS Cache Hit Benchmark 6-24
CIFS Cache Miss Benchmark 6-25
CIFS Native WAN Benchmark 6-26

APPENDIX A **SAN Considerations** A-1
EMC A-1
EMC DMX3 Host Device Information A-3
Network Appliance A-6
Network Appliance FAS6070 Device Information A-8
Linux host dcap-san-hst-04 A-8
Windows host dcap-san-hst-03 A-9
Hewlett Packard A-11
HP XP10000 Device Information A-13

APPENDIX B **Cisco GSS Deployment** A-1

APPENDIX C **WAAS Implementation** C-1

Initial Configuration of the WAAS Central Manager C-2
Initial Core WAE Configuration in the Data Center C-2
Initial Edge WAE Configuration at Remote Branch C-3
WAN Connection C-3
Basic Server/Client Configuration Overview C-4
WAAS Network Configuration Via the Central Manager C-4
Core cluster Settings C-5
Configure WAE Devices for Domain Name System (DNS) C-6
Configure WAE Devices for Windows Name Services (WINS) C-6
Configure NTP on the Central Manager C-6
Configure NTP on Core and Edge WAE Devices C-7
Defining the Core WAE C-7
Defining the Edge WAE C-7
Configure WAE Authentication Methods C-8
Configure a File Server C-8
Create a New Connection C-9
WCCPv2 Overview C-9
WCCPv2 Implementation C-9
Testing Concept C-10

APPENDIX D **The Voodoo Solution** G-1

Emulating 2000 Servers in DCAP G-1
What is Voodoo? G-1
Why the Need for Voodoo? G-1
What are the Necessary Components? G-1
What Features are Used to Make Voodoo Work? G-3
The Voodoo Solution in Full Scale G-5
Configuration Details G-7

APPENDIX E **DCAP Defects** E-1

CSCsh30404 E-1
CSCsg11506 E-2
CSCsg36646 E-2

CSCec74017 E-3
CSCsg22134 E-3
CSCsh25315 E-4
CSCek26222 E-4
CSCsf23499 E-5
CSCsg96362 E-5
CSCsc81109 E-6
CSCsf05420 E-6
CSCsd00731 E-7
CSCsh02565 E-7
CSCsg11203 E-8
CSCsg12760 E-8

APPENDIX F **Safe Harbor Technology Releases** F-1
Native (Classic) IOS 12.2(18)SXF6 F-2
Firewall Services Module (FWSM) 2.3.3.2 F-14
Single-Routed Firewall Services Module (FWSM) 2.3.3.2 F-14
Single-Transparent Firewall Services Module (FWSM) 2.3.3.2 F-17
Multi-Routed Firewall Services Module (FWSM) 2.3.3.2 F-19
Multi-Transparent Firewall Services Module (FWSM) 2.3.3.2 F-21
Content Switching Module (CSM) 4.1.6 F-24
Secure Socket Layer Module (SSLM) 3.1.1 F-26

Preface

Data Center Assurance Program (DCAP)

The Data Center Assurance Program (DCAP) was created to provide a data center design solution that is tested persistently, completely, and objectively. This second phase of the testing builds on the LAN infrastructure elements covered in the first phase, and adds additional features and coverage for the SAN, GSS, and WAAS areas. Future phases will repeat the testing executed in DCAP 2.0 as well as add testing for additional features and coverage. Testing is executed and results are reported as they were experienced. In short, the goal of DCAP is to provide transparency in testing so that our customers feel comfortable deploying these recommended designs.

The DCAP team does not exist as a stand-alone entity. Rather, it maintains close relationships with many successful teams within the Cisco testing community. The Enterprise Solutions Engineering (ESE) data center team supplies the starting point for datacenter topology design through its various SRND documents, which have been created through a close collaboration with marketing organizations and customer feedback sources. Testing direction is also guided by the Financial Test Labs (FTL), SAN Test Labs (STL) and Advanced Services (AS) teams, all of whom maintain tight relationships with customers and have a solid track record of relevant and useful testing. Testing performed as part of Cisco DCAP 2.0 was undertaken by members of the Safe Harbor, NSITE and STL test teams.

The Safe Harbor testing team provides the starting point for DCAP software candidate selection through its proven methodology and code-hardening testing. Where applicable, each software image used in the DCAP test topology has been tested and passed, or is under test, by the Safe Harbor team in their own test topologies.

The key to the DCAP program is the customer involvement, whether direct or indirect. Customer interaction is maintained directly through DCAP team presence at forums such as Cisco Technical Advisory Board (TAB) conferences and through customer feedback through direct polling and conversations. Indirectly, the various customer account teams provide valuable insight into the data center-related issues that are concerning our customers and the direction that customers are moving as data center technologies evolve.

To help maintain this culture of customer feedback, the DCAP team invites the reader to subscribe to one of the following email aliases:

- safeharbor-external-info—provided for Cisco's customers interested in the DCAP program; customers can sign up at http://www.cisco.com/go/safeharbor
- safeharbor-release-info—provided for Cisco sales engineers, CA engineers, account managers, or anyone with a customer that might benefit from DCAP testing.

This DCAP Book

Though all of the elements in the data center function as a whole, these elements can also be viewed individually. DCAP 2.0 testing was performed on this granular basis, and that is how the results are reported. This book contains five chapters and an appendices. Each chapter focuses on a particular component of the data center. The appendix documents procedures and methods used in support of the testing, not directly related to testing.

Chapter 1: LAN (Layer 2-3) Infrastructure

The DCAP LAN infrastructure is built around the Catalyst 6500 switching platform that provides for various features such as 10-Gigabit Ethernet connectivity, hardware switching, and distributed forwarding. The Catalyst 4948 switch is also deployed to provide top-of-rack access to data center servers. The LAN infrastructure design is tested for both functionality and response to negative events.

Chapter 2: LAN (Layer 4-7) Services

The modular Catalyst 6500 switching platform supports various modules which provide services at Layers 4-7. Several of these Service Modules are bundled together and tested in the DCAP 2.0 topology, including the Content Switching Module (CSM), Firewall Services Module (FWSM) and Secure Socket Layer Module (SSLM). The tests in this chapter focus on the ability of these three Service Modules to work together to provide load-balancing, security and encryption services to data center traffic.

Chapter 3: Storage Area Networking (SAN)

The DCAP SAN topology incorporates Cisco MDS fabric director products and design guides, industry best practices, and storage vendor implementation guidelines to provide a SAN infrastructure that is representative of the typical enterprise data center environment. The centerpiece of the topology is the Cisco MDS 9513 multiprotocol SAN director running SAN-OS version 3.0(3).

The topology provides redundant fiber channel connectivity for Linux and Windows hosts using QLogic and Emulex host bus adaptors to three different types of fiber channel enterprise storage arrays, namely the Network Appliance FAS6070, Hewlett Packard XP10000, and EMC DMX3. The topology also provides redundant fiber channel connectivity for synchronous storage replication and fiber channel over IP connectivity for asynchronous storage replication. Delay simulators allow modeling of a redundant data center environment for disaster recovery and business continuance testing. The topology is designed to use actual hosts and applications to generate test traffic to model actual customer environments as closely as possible.

Chapter 4: Global Site Selector (GSS)

The Global Site Selector (GSS) leverages DNS's distributed services to provide high availability to existing data center deployments by incorporating features above and beyond today's DNS services.

The GSS's are integrated into the existing DCAP 2.0 topology along with BIND Name Servers and tested using various DNS rules configured on the GSS . Throughout the testing, the GSS receives DNS queries sourced from client machines as well as via DNS proxies (D-Proxies). The Name Server zone files on

the D-Proxies are configured to nsfoward DNS queries to the GSS to obtain authoritative responses. Time To Live (TTL) values associated with the various DNS resource records are observed and taken into consideration throughout the testing.

GSS testing focuses on the fundamental ability of the GSS to work together with existing BIND Name Servers to provide global server load-balancing.

Chapter 5: Wide Area Application Services (WAAS)

Cisco Wide Area Application Services 4.0.3 (WAAS) is an application acceleration and WAN optimization solution for the branch office that improves the performance of any TCP-based application operating in a Wide Area Network (WAN) environment. With Cisco WAAS, enterprises can consolidate costly branch office servers and storage into centrally managed data centers, while still offering LAN-like service levels for remote users. The DCAP WAAS topology incorporates a single Wide-area Application Engine (WAE) into the data center as well as a single WAE at a remote branch office. The tests in this chapter focus on the basic functionality of the WAAS software on the WAE devices as well as the data center and branch routers ability to intercept and redirect TCP-based traffic.

Overview

DCAP Testing Methodology

The Safe Harbor team is a key partner for the DCAP team. The methodology and approach to testing that Safe Harbor uses ensures that the testing is relevant and the software is more stable. That is why this methodology has been adopted by the DCAP team for use in its testing.

There are several elements of the Safe Harbor methodology that set it apart and provide for a higher level of reliability in software releases. First is the deference that Safe Harbor gives to Cisco's customers. The results of every test are viewed from the perspective of how they might impact the end-user. The same goes for the bug scrubs that the Safe Harbor team conducts on a given release candidate. Bugs are monitored prior to a release and during the entire testing cycle. Any defects that may impact a customer are evaluated and scrutinized. Severity 3 defects are given the same level of consideration as Severity 1 and 2 defects, as they might be just as impacting to a customer.

A fix for a given defect always has the potential of causing problems in the same area of code, or even a different area. Because of this possibility of "collateral damage," Safe Harbor will never begin a final run of testing until the last fix has been committed. Only FCS code makes it into the test bed for the final test run. Because the software candidate is already available to the customer, the Safe Harbor team can maintain a Time-to-Quality focus, rather than responding to time-to-market pressures.

Lastly, and perhaps most importantly, the Safe Harbor team anchors its testing philosophy with an unqualified openness. Safe Harbor reports the results, as they occurred, so that customers have the opportunity to evaluate them based on their requirements. That is why DCAP aligns itself so closely with this successful Safe Harbor approach.

DCAP Testing Overview

This document presents the results of Cisco DCAP 2.0 testing.

Cisco DCAP 2.0 testing passed with exception. See DDTS summary tables per chapter for details on the defects that were encountered during testing.

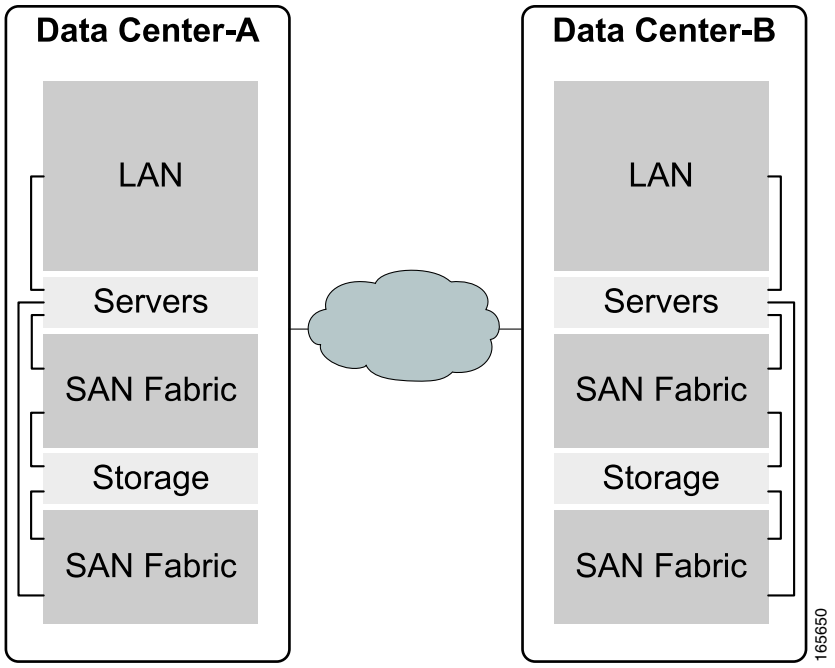
DCAP 2.0 testing builds on the previous phase by bringing in more of the overall data center elements, including:

- LAN Service Switch model
- SAN with replication
- Global Site Selector (GSS)
- Wide Area Application Services (WAAS)

While the previous phase of testing focused solely on the Local Area Network (LAN) design and functionality in a single data center, DCAP 2.0 gets closer to an actual end-to-end data center deployment. The addition of a Storage Area Networking (SAN) component was a key deliverable in DCAP 2.0. Also, a second data center was added in DCAP 2.0, enabling various storage replication scenarios. This phase of testing also added another element to the LAN deployment, in the form of a Service Switch deployment of the Catalyst 6500-based service modules. On the Wide Area Network (WAN) side, the WAAS product set was deployed, allowing the testing of caching and WAN optimization features between the data center and branch offices. The Global Site Selector (GSS) was also introduced in DCAP 2.0, paving the way for future testing on the load balancing of application availability between the two data centers.

Figure 1-1 gives a very high-level view of the DCAP 2.0 test topology components, (excluding the WAAS and GSS elements). Each of the two data centers is similar in the components. Each has a LAN infrastructure consisting of Core, Aggregation, and Access Layers. Servers form the bridge between the LAN and the SAN components, being both LAN-attached (via Ethernet) and SAN-attached (via FibreChannel). The servers are dual-homed into redundant SAN fabrics and the redundant SAN fabrics are, in turn, connected to the storage arrays. The storage layers in both data centers are connected for replication purposes.

Figure 1-1 DCAP Logical Topology



Where possible, DCAP testing tries to stay away from emulation, in favor of real hardware and applications. This is where the relationships with certain vendors become key. The DCAP team has worked closely with Network Appliance, Hewlett-Packard and EMC to deploy Tier 1 storage arrays from each of those vendors in the DCAP test topology.

The DCAP testing effort often relies on testing performed by other teams, particularly the Safe Harbor team. As mentioned above, the determination of which software to run in the various systems in the DCAP topology is made based on Safe Harbor software recommendations. Many of the tests executed in regular Safe Harbor testing are applicable to the DCAP topology and are leveraged for the final DCAP product. While those test results are considered in the final result, they are not reported in this document.

Table 1-1 lists the EDCS document numbers so that the reader can locate and review the results of relevant Safe Harbor testing. A comprehensive list of the test cases executed in these other projects is provided in the Appendix to this document.

Table 1-1 Safe Harbor Technology Releases in EDCS

Platform	Software Version	EDCS Doc. No.
Supervisor 720	12.2(18)SXF6	566444
Firewall Services Module	2.3(3.2)	523606
Content Switching Module	4.1(6)	514342
Secure Socket Layer Module	3.1(1)	504160
Catalyst 4948-10GE	12.2(31)SGA	N/A
MDS9500	3.0(3)	N/A

Note This documentation stipulates that the tests either Pass, Pass with Exception, or Fail. If a test Fails, and the impact to our customer base is determined to be broad enough, the entire release fails (resulting from 1 or more unresolved defects, notwithstanding unresolved cosmetic, minor, or test-specific defects, which are scrutinized by the DCAP engineering team as being a non-show stopping defect. If a test Fails, and the impact to our customer base is determined to be minor, the release as a whole may still Pass, with defects noted. Exceptions to any particular test are noted for disclosure purposes and incidental noteworthy clarification. Customers are advised to carefully review tests selected, by test suite and feature, particular to their environment.

DCAP Topology

Several separate test beds were used for Phase Two of the DCAP testing. The DC-A test bed was used to test the functionality of the IP infrastructure at Layers 2 and 3. Test and background traffic on this test bed was generated using traffic generators and racks of real servers running Linux. The DCb test bed was used to test the functionality of the IP infrastructure (services) at Layers 4 through 7, including the service switch model.

- For this second phase of DCAP testing, all Supervisor 720 platforms were running NativeIOS release 12.2(18)SXF6.
- The Catalyst 4948 access-layer switches were running NativeIOS release 12.2(31)SGA.
- The Content Switching Modules (CSM) were running CSM release 4.1(6).
- The Firewall Services Modules (FWSM) were running FWSM release 2.3(3.2).
- The Secure Socket Layer Service Modules (SSLSM) were running SSLSM release 3.1(1).

CHAPTER 2

Layer 2-3 Infrastructure

The DCAP 2.0 test topology consists of two separate data centers, DC-A and DC-B. Each data center has its own LAN, SAN and storage components. The tests performed with regards to Layer 2-3 Infrastructure verification were executed against the LAN topology in DC-A. Figure 2-1 shows this portion of the DCAP 2.0 test topology. It is divided into three distinct, logical layers called the Core, Aggregation, and Access Layers offering the Layer 2-3 services listed in Table 2-1.

Figure 2-1 DCAP Logical Topology

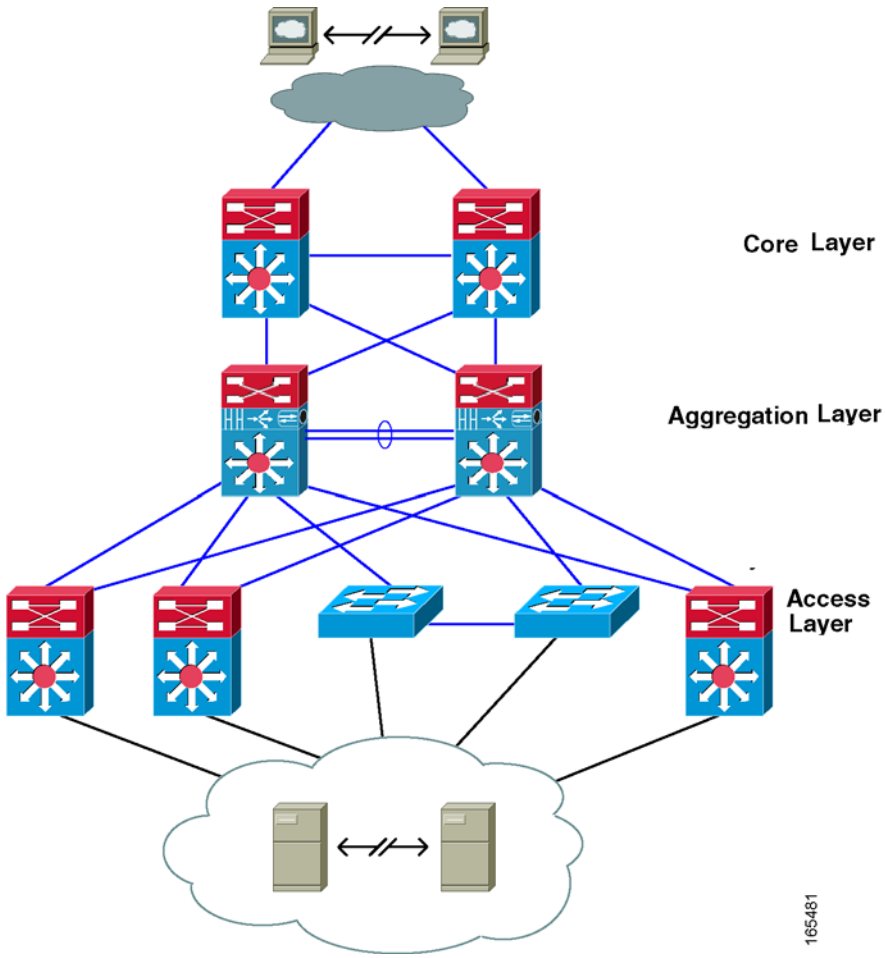


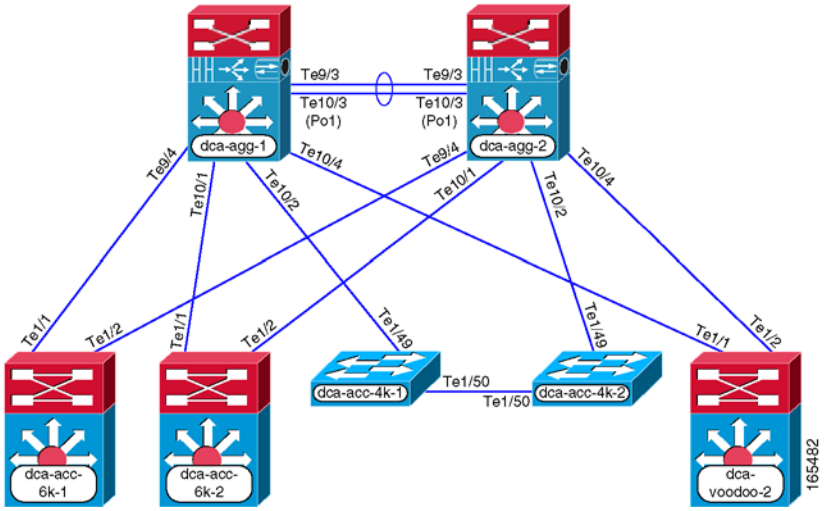
Table 2-1 DCAP 2.0 Logical Layer Services

Logical Layer	Sservices
Core	OSPF, CEF
Aggregation	Default Gateway Redundancy (HSRP), OSPF, Rapid PVST+ Spanning-Tree, UDLD, LACP, 802.1q Trunking
Access	Rapid PVST+ Spanning-Tree, 802.1q Trunking

Layer 2 Topology

Figure 2-2 shows the Layer 2 topology configuration of the DCAP test topology. There are seven devices that operate at Layer 2 in the test topology: dca-agg-1, dca-agg-2, dca-acc-6k-1, dca-acc-6k-2, dca-acc-4k-1, dca-acc-4k-2, and dca-voodoo-2.

Figure 2-2 Layer 2 Topology



All interswitch links in the Layer 2 domain are TenGigabitEthernet. For Cisco DCAP 2.0 testing, there are two groups of VLANs. The first group includes VLANs that are actually used for data traffic in the DCAP test plan. There are about 75 VLANs that are actually passing test traffic. In addition to these 75, there are roughly 170 additional VLANs in the DCAP Layer 2 domain that have been included to provide some scaling for Spanning-Tree and HSRP.

Each of the seven devices in the Layer 2 domain participate in Spanning-Tree. The Aggregation Layer device dca-agg-1 is configured as the primary Spanning-Tree Protocol (STP) root device for all VLANs in the Layer 2 domain, and dca-agg-2 is configured as the secondary STP root. The Spanning-Tree Protocol (STP) that is used in the DCAP topology is PVST+ plus the rapid convergence enhancements of IEEE 802.1w (collectively referred to as Rapid PVST+ or rPVST+).

As discussed in Chapter 3, "Layer 4-7 Services," the Aggregation Layer devices provide a number of services to the data traffic in the network. The Firewall Services Module (FWSM), installed in each of the two Aggregation Layer devices, provide some of these services. In the DCAP topology, the FWSM

is operating in multi-context transparent mode and bridges traffic between the outside VLAN to the inside VLAN. As such, only a subset of VLANs (the inside VLANs) are propagated down to the Access Layer devices, and the servers that reside on them.

While only a subset of VLANs are carried on the trunks connecting the Access Layer to the Aggregation Layer, the trunk between dca-agg-1 and dca-agg-2 carries all VLANs in the Layer 2 domain. This includes the same subset of inside VLANs that are carried to the Access Layer, their counterpart subset of outside VLANs, as well as a small subset of management VLANs.

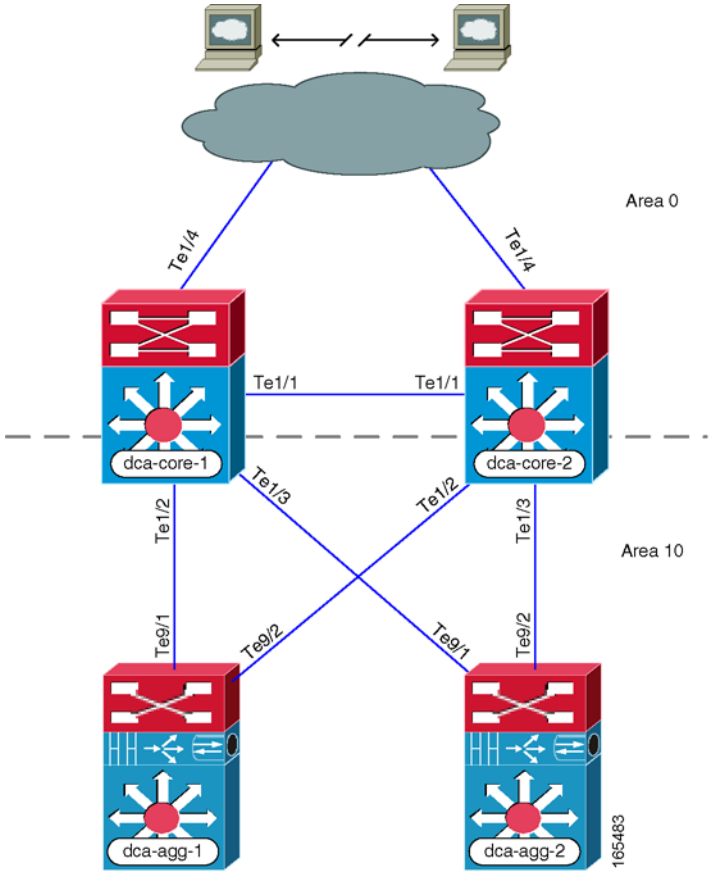
Some of these management VLANs carried between dca-agg-1 and dca-agg-2 carry keepalive traffic for the service modules in these two devices. The active and standby Content Switching Module (CSM) and FWSM pass heartbeat messages between each other so that, should the active become unavailable, the standby can transition itself to take over the active role for those services. If communication between the active and standby peers is lost, and the hardware has not been impacted, an “active/active” condition will likely result. This can wreak havoc on a service-based network and the data traffic that it carries. The reliability of communication between the two peers, then, is important.

The criticality of these heartbeat messages mandates a high level of redundancy for the link carrying these heartbeats. For this reason, two TenGigabitEthernet links are bundled together using Link Aggregation Control Protocol (LACP) to form an etherchannel between dca-agg-1 and dca-agg-2. Having two links provides one level of redundancy. Having these links split between two separate modules on each device provides an additional level of redundancy.

Layer 3 Topology

Figure 2-3 shows the general Layer 3 configuration of the DCAP 2.0 test topology. There are four devices that operate at Layer 3 of the OSI stack: dca-core-1, dca-core-2, dca-agg-1, and dca-agg-2.

Figure 2-3 Layer 3 Topology



The Layer 3 portion of the topology is fully meshed with TenGigabitEthernet, with OSPF running as the interior gateway protocol. The devices dca-core-1 and dca-core-2 serve as Area Border Routers (ABR) between Area 0 and Area 10. The link between these two Core Layer devices is in OSPF Area 0. The links between the Core Layer devices and the Aggregation Layer devices are in OSPF Area 10.

In the DCAP test topology, each of the Core Layer devices links up towards the Client cloud. These links are also in Area 0 and this is how the Layer 3 devices in the test topology know about the Client subnets.

The devices dca-agg-1 and dca-agg-2 provide default gateway redundancy via Hot Standby Router Protocol (HSRP). An HSRP default gateway is provided for each of the subnets defined by VLANs in the Layer 2 domain. By configuration, dca-agg-1 is the Active HSRP Router and dca-agg-2 the Standby. Preempt is configured for each VLAN on each of these two devices.

Layer 2-3 Test Results Summary

Table 2-2 summarizes tests executed as part of the Cisco DCAP 2.0 testing initiative. Table 2-2 includes the feature or function tested, the section that describes the feature set the feature or function belongs to, the component tests for each feature or function, and whether the test is new in this phase of DCAP testing.

Note Test results are unique to technologies covered and actual scenarios in which they were tested. DCAP is designed to cover critical path areas and augment ongoing regression and systems testing.

Table 2-2 Cisco DCAP 2.0 L2-3 Testing Summary

Test Suites	Features/Functions	Tests	Results
Baseline, page 2-7	Topology Baseline, page 2-8	1. Topology Baseline	
	Device Management, page 2-9	1. Upgrade of Supervisor 720 System in Core Layer 2. Upgrade of Supervisor 720 System in Aggregation Layer 3. Upgrade of Supervisor 720 System in Access Layer 4. Upgrade of Catalyst 4948-10GE System in Access Layer 5. Upgrade of Content Switching Module (CSM) 6. Upgrade of Firewall Services Module (FWSM) 7. Upgrade of Secure Socket Layer Services Module (SSLSM) 8. General On-Line Diagnostics (GOLD) 9. SNMP MIB Tree Walk 10. Local SPAN 11. Remote SPAN (rSPAN)	
	Device Access, page 2-21	1. Repeated Logins Using SSH Version 1 2. Repeated Logins Using SSH Version 2	
	CLI Functionality, page 2-23	1. CLI Parser Functionality Using SSHv1 2. CLI Parser Functionality Using SSHv2 3. CLI Parser Functionality Using SSHv1 on 4948 4. CLI Parser Functionality Using SSHv2 on 4948	CSCsc81109 CSCsc81109
	Security, page 2-25	1. Malformed SNMP Polling 2. Malformed SSH Packets 3. NMAP Open Port Scan	
	Traffic Forwarding, page 2-28	1. Zero Packet Loss 2. Distributed FIB Consistency	
Layer 2 Protocols	Link Aggregation Control Protocol (LACP), page 2-31	1. LACP Basic Functionality 2. LACP Load Balancing	
	Trunking, page 2-33	1. 802.1q Trunking Basic Functionality	

Table 2-2 Cisco DCAP 2.0 L2-3 Testing Summary (continued)

Test Suites	Features/Functions	Tests	Results
Layer 2 Protocols	Spanning Tree, page 2-34	1. Rapid PVST+ Basic Functionality 2. Root Guard	
	Unidirectional Link Detection (UDLD), page 2-38	1. UDLD Detection on 10GE Links	
Layer 3 Protocols	Hot Standby Router Protocol (HSRP), page 2-39	1. HSRP Basic Functionality	
	Open Shortest Path First (OSPF), page 2-41	1. OSPF Route Summarization 2. OSPF Database Verification	
Negative Testing	Hardware Failure, page 2-43	1. Access Layer Supervisor Failover Using SSO with NSF 2. Repeated Reset of Standby Supervisor in Access Layer 3. Reset of Aggregation Layer Device dca-agg-1 4. Reset of Aggregation Layer Device dca-agg-2 5. Reset of Core Layer Device dca-core-1 6. Reset of Core Layer Device dca-core-2 7. Spanning-Tree Primary Root Failure and Recovery 8. HSRP Failover with Fast Timers 9. HSRP Recovery From System Failure 10. Failure of EtherChannel Module on dca-agg-1 11. Failure of EtherChannel Module on dca-agg-2	CSCsf23499 CSCek26222 CSCek26222
	Link Failure, page 2-61	1. Failure of Single Bundled 10-Gigabit Ethernet Link Between dca-agg-1 and dca-agg-2 2. Failure of 10 Gigabit Ethernet Link Between dca-agg-1 and dca-acc-6k-2 3. Failure of 10 Gigabit Ethernet Link Between dca-agg-2 and dca-acc-6k-2 4. Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-agg-1 5. Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-agg-2 6. Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-core-2 7. Failure of 10-Gigabit Ethernet Link Between dca-core-2 and dca-agg-1 8. Failure of 10-Gigabit Ethernet Link Between dca-core-2 and dca-agg-2 9. Network Resiliency Test	

Layer 2-3 DDTS Summary

Table 2-3 lists Development Defect Tracking System (DDTS) software bugs with descriptions, and comments filed by the DCAP testing team during Cisco DCAP 2.0 L2-3 testing. Table 2-4 lists DDTS with descriptions encountered during Cisco DCAP 2.0 L2-3 testing.

Table 2-3 Summary of DDTS Filed During Cisco DCAP 2.0 L2-3 Testing

CSCsh02565	Misleading error message for slowstart feature.
CSCsg97985	CSM to track the backup serverfarm by default if configured.

Table 2-4 Summary of DDTS Encountered During Cisco DCAP 2.0 L2-3 Testing

CSCsd00731	Show mod csm X conns displays the active and passive ftp data port has 0.
CSCdz22187	CSM mark vserver OUTOFSERVICE when backup serverfarm is OPERATIONAL.
CSCsf23499	VLAN int goes DOWN when etherchannel trunk comes up.
CSCsc81109	show crypto pki server cli command causes Traceback
CSCek26222	mem leak at tm_dbg_msg_queue_init

Layer 2-3 Test Cases

Functionality critical to global enterprises inCisco DCAP 2.0 Layer 2-3 testing is described in the following sections. Refer to Cisco Data Center Assurance Program (DCAP) 2.0 Configurations document for test device configurations.

- Baseline, page 2-7
- Layer 2 Protocols, page 2-30
- Layer 3 Protocols, page 2-39
- Negative Testing, page 2-43

Baseline

The baseline tests are focused on various aspects of administering the devices in the DCAP 2.0 test topology, as well as the verification of the most basic features such as distributed forwarding and security.

The following test features were conducted:

- Topology Baseline, page 2-8
- Device Management, page 2-9
- Device Access, page 2-21
- CLI Functionality, page 2-23
- Security, page 2-25
- Traffic Forwarding, page 2-28

Topology Baseline

In all of DCAP 2.0 testing, system resources of all the Layer 2/3 devices in the test topology are monitored, including CPU and memory utilization. When an issue is suspected, manifest as a sustained CPU spike or consumed memory for example, it is helpful to have a steady-state baseline of what the network resources look like for comparison purposes. The tests in this section help to establish a baseline level of expected behavior so that real problems can be more easily identified.

The following test was performed:

- Topology Baseline, page 2-8

Topology Baseline

This test verified the network operation during steady state. While all background traffic and background routes are running, the network is allowed to run without perturbation to quantify the baseline cpu and memory of each device.

Test Procedure

The procedure used to perform the Topology Baseline test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Baseline all CDP neighbor relationships. Run the CDP crawler script verifying all expected CDP neighbors are reported.

The purposed of the CDP crawler script is to crawl through the network continuously, noting any changes that occur between traversals in CDP information. It parses information gathered from select CDP and IOS commands.

Step 3 Baseline all EtherChannel members. Run the channel crawler script verifying that all interfaces expected to be in channels are reported.

The purpose of the channel crawler script is to run through a network and verify that EtherChannels are in a proper state. It parses information gathered from select EtherChannel and IOS commands.

Step 4 Baseline all trunk interfaces. Run the trunk crawler script verifying that all expected trunking interfaces, configuration, and status are reported.

The purpose of the trunk crawler script is to run through a network and verify that trunking is in a proper state. It parses information gathered from select trunking and IOS commands.

Step 5 Baseline all interface states and counters. Run the interface crawler script recording interface counters and states.

The interface crawler script crawls through a network continually. All up/up interfaces are checked for various errors. Initially all non-zero error counters will be logged, then any counters that increment from that point on.

Step 6 Baseline all interface UDLD states. Run the UDLD crawler script recording the UDLD state of all interfaces.

The UDLD crawler script gathers a list of UDLD ports from a list of devices and traverses their neighbors continuously checking for UDLD problems or inconsistencies. It parses information gathered from select UDLD and IOS commands.

Step 7 Baseline all linecards used in the topology. Run the module crawler script recording module counters and state.

The module crawler script gathers a list of MODULES from a list of devices and looks for problems or inconsistencies. It parses information gathered from select module and IOS commands.

Step 8 Begin the test traffic. Allow it to run for two hours.

Step 9 Execute the CDP crawler script to verify that the CDP feature is operating in the Data Center test network as it was before background traffic was started.

Step 10 Execute the channel crawler script to verify that the EtherChannel feature is operating in the Data Center test network as it was before background traffic was started.

Step 11 Execute the trunk crawler script to verify that the trunking feature is operating in the Data Center test network as it was before background traffic was started.

Step 12 Execute the interface crawler script to verify that the basic functionality of the interface is operating in the Data Center test network as it was before background traffic was started.

Step 13 Execute the UDLD crawler script to verify that the UDLD feature is operating in the Data Center test network as it was before background traffic was started.

Step 14 Execute the module crawler script to verify that the line cards in the Data Center test network are still operating correctly after background traffic was started.

Step 15 Stop background scripts to collect final status of network devices and analyze for error.

Step 16 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Topology Baseline passed.

Device Management

Device Management tests cover some of the common procedures and features used in the normal operation of a network, including the upgrading of network devices and the use of various features that may be used in troubleshooting.

This test verified that the Cisco IOS upgrade process worked correctly.

The following tests were performed:

- Upgrade of Supervisor 720 System in Core Layer, page 2-10
- Upgrade of Supervisor 720 System in Aggregation Layer, page 2-11
- Upgrade of Supervisor 720 System in Access Layer, page 2-11
- Upgrade of Catalyst 4948-10GE System in Access Layer, page 2-12
- Upgrade of Content Switching Module (CSM), page 2-13
- Upgrade of Firewall Services Module (FWSM), page 2-14
- Upgrade of Secure Socket Layer Services Module (SSLSM), page 2-15

- General On-Line Diagnostics (GOLD), page 2-16
- SNMP MIB Tree Walk, page 2-18
- Local SPAN, page 2-18
- Remote SPAN (rSPAN), page 2-19

Upgrade of Supervisor 720 System in Core Layer

This test verified the ability for code to be upgraded for the latest version of Safe Harbor certified SXF code to the version of SXF that is under test. Core layer device dca-core-1 was upgraded from 12.2(18)SXF4 Native IOS to 12.2(18)SXF6 Native IOS to ensure that all hardware and configurations at the core layer were upgraded without issue.

Test Procedure

The procedure used to perform the Upgrade of Supervisor 720 System in Core Layer test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Record the start time of this test using the show clock command.

Step 3 Verify that dca-core-1 is running the old Native Cisco IOS image using the show version command.

Step 4 Verify that the Supervisor 720 image under test is on the proper file device on dca-core-1 using the dir disk0: command.

Step 5 Use the show running-config | include boot command to verify that the boot string points to the proper device and filename for the test image. If any changes are necessary, make them and then save them to NVRAM when done.

Step 6 Issue the reload command on dca-core-1, causing the supervisor to reboot. Report any error messages seen during reload.

Step 7 Use the show module and show version commands to verify that dca-core-1 came online successfully and that the new image is running.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the Cisco IOS Upgrade of Supervisor 720 System in Core Layer upgrade process to complete without error.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Upgrade of Supervisor 720 System in Core Layer passed.

Upgrade of Supervisor 720 System in Aggregation Layer

This test verified the ability for code to be upgraded for the latest version of Safe Harbor certified SXF code to the version of SXF that is under test. Aggregation layer device dca-agg-1 was upgraded from 12.2(18)SXF4 Native IOS to 12.2(18)SXF6 Native IOS to ensure that all hardware and configurations at the core layer were upgraded without issue.

Test Procedure

The procedure used to perform the Upgrade of Supervisor 720 System in Aggregation Layer test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Record the start time of this test using the show clock command.

Step 3 Verify that the dca-agg-1 is running the old Native Cisco IOS image using the show version command.

Step 4 Verify that the Supervisor 720 image under test is on the proper file device on dca-agg-1 using the dir disk0: command.

Step 5 Use the show running-config | include boot command to verify that the boot string points to the proper device and filename for the test image. If any changes are necessary, make them and then save them to NVRAM when done.

Step 6 Issue the reload command on dca-agg-1, causing the supervisor to reboot. Report any error messages seen during reload.

Step 7 Use the show module and show version commands to verify that dca-agg-1 came online successfully and that the new image is running.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the Cisco IOS Upgrade of Supervisor 720 System in Aggregation Layer upgrade process to complete without error.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Upgrade of Supervisor 720 System in Core Layer passed.

Upgrade of Supervisor 720 System in Access Layer

This test verified the ability for code to be upgraded for the latest version of Safe Harbor certified SXF code to the version of SXF that is under test. Access layer device dca-acc-6k-1 was upgraded from 12.2(18)SXF4 Native IOS to 12.2(18)SXF6 Native IOS to ensure that all hardware and configurations at the access layer were upgraded without issue.

Test Procedure

The procedure used to perform the Upgrade of Supervisor 720 System in Access Layer test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Record the start time of this test using the show clock command.

Step 3 Verify that dca-acc-6k-1 is running the old Native Cisco IOS image using the show version command.

Step 4 Verify that the Supervisor 720 image under test is on the proper file devices on dca-acc-6k-1 using the dir disk0: and dir slavedisk0: commands.

The device dca-acc-6k-1 is configured with dual supervisors. It is therefore necessary that each of these supervisors has the new image in their respective file systems.

Step 5 Use the show running-config | include boot command to verify that the boot string points to the proper device and filename for the test image. If any changes are necessary, make them and then save them to NVRAM when done.

Step 6 Issue the reload command on dca-acc-6k-1, causing both supervisors to reboot. Report any error messages seen during reload.

Step 7 Use the show module and show version commands to verify that dca-acc-6k-1 came online successfully and that the new image is running.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the Cisco IOS Upgrade of Supervisor 720 System in Access Layer upgrade process to complete without error.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Upgrade of Supervisor 720 System in Access Layer passed.

Upgrade of Catalyst 4948-10GE System in Access Layer

This test verified the ability for code to be upgraded to the version of code that is under test. Access layer device dca-acc-4k-1 was upgraded from 12.2(31)SG Native IOS to 12.2(31)SGA Native IOS to ensure that all hardware and configurations at the access layer were upgraded without issue.

Test Procedure

The procedure used to perform the Upgrade of Catalyst 4948-10GE System in Access Layer test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Record the start time of this test using the show clock command.

Step 3 Verify that dca-acc-4k-1 is running the old Native Cisco IOS image using the show version command.

Step 4 Verify that the image under test is on the proper file device on dca-acc-4k-1 using the dir bootflash: command.

The new image needs to be the first image on the bootflash: device in order for it to boot. The 4948-10GE system will boot the first image in bootflash:.

Step 5 Issue the reload command on dca-acc-4k-1, causing the system to reboot. Report any error messages seen during reload.

Step 6 Use the show module and show version commands to verify that dca-acc-4k-1 came online successfully and that the new image is running.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the Cisco IOS Upgrade of Catalyst 4948-10GE System in Access Layer upgrade process to complete without error.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Upgrade of Catalyst 4948-10GE System in Access Layer passed.

Upgrade of Content Switching Module (CSM)

This test verified the ability for code to be upgraded for the latest version of Safe Harbor certified CSM code to the version of CSM code that is under test.

In DCAP 2.0, the candidate CSM version is the same as was tested in DCAP 1.0. For this reason, an actual upgrade from earlier code was not performed. Instead, the download and upgrade procedure was performed on the CSM which was starting with code version 4.1(6).

Test Procedure

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Record the start time of this test using the show clock command on dca-agg-1.

Step 3 Verify that the CSM in dca-agg-1 is running the old CSM image using the show module 2 command.

Step 4 Verify that the CSM image under test is on the proper file device on dca-agg-1 using the dir sup-bootflash: command.

Step 5 Use the show running-config | include tftp-server command to verify that dca-agg-1 is configured to be a TFTP server for that image. This will make the image downloadable to the CSM directly from the Supervisor.

Step 6 Set up a session between the supervisor engine and the CSM using the session slot 2 processor 0 command.

Step 7 Load the image from the supervisor engine to the CSM using the upgrade command.

Step 8 Once the new image has completed the download, exit the session and reboot the CSM module from the supervisor CLI, using the hw-module module 2 reset command.

Step 9 When the CSM module comes back online, use the show module 2 command on dca-agg-1 to verify that the new image has been loaded.

Step 10 Stop background scripts to collect final status of network devices and analyze for error.

Step 11 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the Cisco IOS Upgrade of Content Switching Module (CSM) upgrade process to complete without error.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Upgrade of Content Switching Module (CSM) passed.

Upgrade of Firewall Services Module (FWSM)

This test verified the ability for code to be upgraded for the latest version of Safe Harbor certified FWSM code to the version of FWSM code that is under test.

Note The FWSM code that is under test in DCAP 2.0 is 2.3(3.2). This is the first FWSM code to be Safe Harbor certified. For this reason, in this test, there will be no actual upgrade. Instead, a reload will be performed. The entire procedure for the upgrade remains below, and where steps were skipped, it was noted.

Test Procedure

The procedure used to perform the Upgrade of Firewall Services Module (FWSM) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Record the start time of this test using the show clock command on dca-agg-1.

Step 3 Verify that the FWSM in dca-agg-1 is running the old FWSM image using the show module 1 command.

In this run, the FWSM is running the FWSM image currently under test.

Step 4 Verify that the FWSM image under test is on the proper file device on dca-agg-1 using the dir sup-bootflash: command.

Step 5 Use the show running-config | include tftp-server command to verify that dca-agg-1 is configured to be a TFTP server for that image. This will make the image downloadable to the FWSM directly from the Supervisor.

Step 6 Set up a session between the supervisor engine and the FWSM using the session slot 1 processor 1 command.

Step 7 Verify connectivity from the FWSM to the supervisor using the ping command to the loopback address of the supervisor, 127.0.0.71.

Step 8 Use the copy tftp://127.0.0.71/image_name flash: command to download the new image from the TFTP server on the supervisor.

Step 9 Issue the reload command on the FWSM to reboot the blade.

Step 10 Once the FWSM has come back online, verify that the new image is running using the show module 1 command.

Step 11 Stop background scripts to collect final status of network devices and analyze for error.

Step 12 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the Cisco IOS Upgrade of Firewall Services Module (FWSM) upgrade process to complete without error.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Upgrade of Firewall Services Module (FWSM) passed.

Upgrade of Secure Socket Layer Services Module (SSLSM)

This test verified the ability for code to be upgraded for the latest version of Safe Harbor certified SSLSM code to the version of SSLSM code that is under test.

Note In DCAP 2.0, the candidate SSLSM version is the same as was tested in DCAP 1.0. For this reason, an actual upgrade from earlier code was not performed. Instead, the download and upgrade procedure was performed on the SSLSM which was starting with code version 3.1(1).

Test Procedure

The procedure used to perform the Upgrade of Secure Socket Layer Services Module (SSLSM) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Record the start time of this test using the show clock command on dca-agg-1.

Step 3 Verify that the SSLSM in dca-agg-1 is running the old SSLSM image using the show module 3 command.

Step 4 Verify that the SSLSM image under test is on the proper file device on dca-agg-1 using the dir sup-bootflash: command.

Step 5 Use the show running-config | include tftp-server command to verify that dca-agg-1 is configured to be a TFTP server for that image. This will make the image downloadable to the SSLSM directly from the Supervisor.

Step 6 Boot the SSLSM blade in slot 3 on dca-agg-1 into the maintenance partition using the hw-module module 3 reset cf:1 command.

Step 7 Verify that the slot 3 SSLSM is in the maintenance partition using the show module 3 command.

The Sw field in this command output should be appended with a lower-case m and the Status field should real Ok.

Step 8 Download the new image onto the slot 3 SSLSM using the copy tftp: pclc#3-fs: command. Enter 127.0.0.71 when prompted for the TFTP location.

Step 9 Once the download has completed successfully, you will see a message printed to the console of dca-agg-1 reading You can now reset the module. Reboot the slot 3 SSLSM using the hw-module module 3 reset command.

Step 10 Once the SSLSM has come back online, verify that it is running the new image using the show module 3 command.

Step 11 Stop background scripts to collect final status of network devices and analyze for error.

Step 12 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the Cisco IOS Upgrade of Secure Socket Layer Services Module (SSLSM) upgrade process to complete without error.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Upgrade of Secure Socket Layer Services Module (SSLSM) passed.

General On-Line Diagnostics (GOLD)

General online diagnostics (GOLD) is a software tool which tests and verifies the hardware functionality of the supervisor engine, modules, and switch while the switch is connected to a live network. There are disruptive and non-disruptive online diagnostic tests, including a subset of the GOLD tests which is run upon bootup of a hardware component. These are referred to as bootup diagnostics and are run during bootup, module OIR, or switch up to a redundant supervisor.

Each device in the datacenter topology is configured for a complete diagnostics run on bootup. This test verifies that each device in the datacenter topology is configured to run complete diagnostics on bootup, and that the complete set of diagnostics was run on each module at the last boot event.

Test Procedure

The procedure used to perform the General On-Line Diagnostics (GOLD) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Log into dca-core-1 and use the show diagnostic bootup level command to verify that the current level is set to complete.

The "current diagnostic bootup level" should be "complete."

Step 3 On dca-core-1, use the show diagnostic result all command to verify that the complete set of diagnostics was run against each module in the box, and that there were no failed tests.

There should be no tests with a result marked "F," or Failed.

Step 4 Log into dca-core-2 and use the show diagnostic bootup level command to verify that the current level is set to complete.

The "current diagnostic bootup level" should be "complete."

Step 5 On dca-core-2, use the show diagnostic result all command to verify that the complete set of diagnostics was run against each module in the box, and that there were no failed tests.

There should be no tests with a result marked "F," or Failed.

Step 6 Log into dca-agg-1 and use the show diagnostic bootup level command to verify that the current level is set to complete.

The "current diagnostic bootup level" should be "complete."

Step 7 On dca-agg-1, use the show diagnostic result all command to verify that the complete set of diagnostics was run against each module in the box, and that there were no failed tests.

There should be no tests with a result marked "F," or Failed.

Step 8 Log into dca-agg-2 and use the show diagnostic bootup level command to verify that the current level is set to complete.

The "current diagnostic bootup level" should be "complete."

Step 9 On dca-agg-2, use the show diagnostic result all command to verify that the complete set of diagnostics was run against each module in the box, and that there were no failed tests.

There should be no tests with a result marked "F," or Failed.

Step 10 Log into dca-acc-6k-1 and use the show diagnostic bootup level command to verify that the current level is set to complete.

The "current diagnostic bootup level" should be "complete."

Step 11 On dca-acc-6k-1, use the show diagnostic result all command to verify that the complete set of diagnostics was run against each module in the box, and that there were no failed tests.

There should be no tests with a result marked "F," or Failed.

Step 12 Log into dca-acc-6k-2 and use the show diagnostic bootup level command to verify that the current level is set to complete.

The "current diagnostic bootup level" should be "complete."

Step 13 On dca-acc-6k-2, use the show diagnostic result all command to verify that the complete set of diagnostics was run against each module in the box, and that there were no failed tests.

There should be no tests with a result marked "F," or Failed.

Step 14 Stop background scripts to collect final status of network devices and analyze for error.

Step 15 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that the complete set of online diagnostics will have been run on all modules in the systems under test, as configured.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

General On-Line Diagnostics (GOLD) passed.

SNMP MIB Tree Walk

Simple Network Management Protocol (SNMP) is ubiquitous as a tool used by network administrators to manage network performance, find and solve network problems, and plan for network growth.

This test verified that a SNMP walk of the MIB tree of dca-agg-1 did not cause any memory loss, tracebacks, or crashes. From a server, five version 1 SNMP walks were performed.

Test Procedure

The procedure used to perform the SNMP MIB Tree Walk test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 If the background test traffic is not already running, start it now.

Step 3 Verify the SNMP configuration of dca-agg-1 using the show running-config command.

Step 4 From the server CLI perform five SNMP walks on the DUT using the snmpwalk utility.

Step 5 Stop background scripts to collect final status of network devices and analyze for error.

Step 6 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that no tracebacks or crashes to occur on the DUT.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

SNMP MIB Tree Walk failed CSCsf23499.

Local SPAN

Local SPAN selects network traffic to send to a network analyzer. SPAN should not affect the switching of network traffic on source ports or VLANs. SPAN sends a copy of the packets received or transmitted by the source ports and VLANs to a destination port dedicated for SPAN use.

This test verified that normal traffic forwarding was maintained when a local SPAN session was configured on dca-acc-6k-2. Interface 10-Gigabit Ethernet 1/1 was used as the SPAN source. The destination was a Gigabit Ethernet interface connected to a server running Knoppix. This server was running the tethereal program to capture the traffic. The network was monitored for traffic irregularities and the DUT was monitored for CPU or memory stability.

Test Procedure

The procedure used to perform the Local SPAN test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 On dca-acc-6k-2, use the show monitor command to verify that there are no SPAN sessions present.

Step 3 Configure the SPAN source to be interface Te1/1, using the monitor session 1 source interface Te1/1 both command. By specifying both, the session will SPAN ingress and egress traffic on Te1/1.

Step 4 Configure the SPAN destination to be interface Gi2/41 using the monitor session 1 destination interface Gi2/41 command.

Step 5 Clear the traffic counters on dca-acc-6k-2 using the clear counters command.

Step 6 Begin the capture session on the Knoppix server.

Step 7 Run the background test traffic for a period of 10 minutes.

Step 8 When the background test traffic finishes, verify that it does not report any more than the normal amount of errors.

The script that is used to run the background test traffic will report statistics in the form of HTTP return codes. The Zero Packet Loss test indicates that the normal number of errors is below 0.01% (comparing, in that test, "500" return codes to "200" return codes).

Step 9 Compare the counters of the SPAN source interface with those of the SPAN destination interface using the show interface interface counters command.

The SPAN source is monitoring both transmit and receive of the source interface. The SPAN destination interface egress counters should reflect the combination of both directions of traffic on the SPAN source.

Step 10 Look for any errors on the SPAN destination interface using the show interfaces Gi2/41 command.

Step 11 Remove the SPAN configuration from dca-acc-6k-2 using the no monitor session 1 command.

Step 12 Stop background scripts to collect final status of network devices and analyze for error.

Step 13 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the SPAN utility to operate soundly under load.
- We expect the SPAN utility will not interfere with normal network traffic.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Local SPAN passed.

Remote SPAN (rSPAN)

With remote SPAN, the SPAN destination is a VLAN, rather than a physical interface. This VLAN is configured as a remote VLAN throughout the network. Traffic that is copied to the SPAN VLAN is tagged with that VLAN ID and sent through the network to a traffic analyzer attached to a network device that is remote to the SPAN source.

This test verified that normal traffic forwarding was maintained when a remote SPAN session was configured on dca-agg-1. Interface Te9/4 was used as the SPAN source. The destination was remote-vlan 900. This VLAN is configured throughout the Layer 2 domain in the DCAP test network. The traffic collector was a locally installed Network Analysis Module (NAM). The network was monitored for traffic irregularities and the DUT was monitored for CPU or memory stability.

Test Procedure

The procedure used to perform the Remote SPAN (rSPAN) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify that VLAN 900 is configured on all the DCAP devices in the Layer 2 domain, and that it is a remote VLAN using the show vlan id 900 command.

VLAN 900 should be present on dca-agg-1, dca-agg-2, dca-acc-6k-1, dca-acc-6k-2, dca-acc-4k-1, and dca-acc-4k-2. In the output for each of these devices, the "Remote SPAN VLAN" field should indicate "Enabled".

Step 3 On dca-agg-1, use the show monitor command to verify that there are no SPAN sessions present.

There may be a monitor session with ID=1 present on either of the Aggregation layer devices in the DCAP test topology as a result of having service modules in the system. If that is the case, session ID=2 may be used.

Step 4 On dca-agg-1, configure the SPAN source to be interface Te9/4, using the monitor session 2 source interface Te9/4 both command. By specifying both, the session will SPAN ingress and egress traffic on Te9/4.

Step 5 Configure the SPAN destination to be remote SPAN VLAN 900 using the monitor session 2 destination remote vlan 900 command.

Step 6 On device dca-acc-6k-2, verify that there are no SPAN sessions present using the show monitor command.

Step 7 On device dca-acc-6k-2, configure the SPAN source to be remote SPAN VLAN 900, using the monitor session 1 source remote vlan 900 command.

Step 8 Configure the SPAN destination to be interface Gi2/41 using the monitor session 1 destination interface Gi2/41 command.

Step 9 Clear the traffic counters on dca-agg-1 and dca-acc-6k-2 using the clear counters command.

Step 10 Begin the capture session on the Knoppix server.

Step 11 Run the background test traffic for a period of 10 minutes.

Step 12 When the background test traffic finishes, verify that it does not report any more than the normal amount of errors.

The script that is used to run the background test traffic will report statistics in the form of HTTP return codes. The Zero Packet Loss test indicates that the normal number of errors is below 1% (comparing, in that test, "500/400/402" return codes to "200" return codes).

Step 13 Compare the counters of the SPAN source interface (Te9/4 on dca-agg-1) with those of the SPAN destination interface (Gi2/41 on dca-acc-6k-2) using the show interface interface counters command.

The SPAN source is monitoring both transmit and receive of the source interface. The SPAN destination interface egress counters should reflect the combination of both directions of traffic on the SPAN source.

It is important to note that the SPAN source interface is a 10-Gigabit Ethernet interface while the destination interface is only Gigabit Ethernet. Packet loss is expected.

Step 14 Look for any errors on the SPAN destination interface using the show interfaces Gi2/41 command on dca-acc-6k-2.

It is important to note that the SPAN source interface is a 10-Gigabit Ethernet interface while the destination interface is only Gigabit Ethernet. Packet loss is expected.

Step 15 Remove the SPAN configurations from dca-agg-1 and dca-acc-6k-2 using the no monitor session session_id command.

Step 16 Stop background scripts to collect final status of network devices and analyze for error.

Step 17 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Figure 2-4 Remote SPAN (rSPAN) Usage

Expected Results

- We expect that the SPAN utility will operate soundly under load.
- We expect that the SPAN utility will not interfere with normal network traffic.
- We expect no sustained or unexpected impact on the CPU or memory.

Results

Remote SPAN (rSPAN) passed.

Device Access

The DCAP test topology includes dedicated out-of-band management links on all of the network devices. The access protocol used on all of these devices is SSH, for security purposes. These tests stress the access protocols used.

The following tests were performed:

- Repeated Logins Using SSH Version 1, page 2-21
- Repeated Logins Using SSH Version 2, page 2-22

Repeated Logins Using SSH Version 1

The device dca-agg-2 was subjected to 1000 login attempts, using version 1 of the SSH protocol, from each of six iterations of the login script. This was done to max out the available VTY interfaces on dca-agg-2. The full profile of background traffic (HTTP and FTP requests) was running during this test.

Test Procedure

The procedure used to perform the Repeated Logins Using SSH Version 1 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify that the HTTP and FTP background traffic is running.

Step 3 Verify that dca-agg-2 is configured for SSH login using the show ip ssh command.

The show ip ssh command should show SSH Enabled - version 1.99 in the output.

Step 4 Initiate 6 iterations of the test script. Each iteration will attempt to log into dca-agg-2 1000 times, successively, using SSH version 1.

Step 5 Stop background scripts to collect final status of network devices and analyze for error.

Step 6 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect no system error messages resulting from the multiple, repeated SSH login attempts.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Repeated Logins Using SSH Version 1 passed.

Repeated Logins Using SSH Version 2

The device dca-agg-1 was subjected to 1000 login attempts, using version 2 of the SSH protocol, from each of six iterations of the login script. This was done to max out the available VTY interfaces on dca-agg-1. The full profile of background traffic (HTTP and FTP requests) was running during this test.

Test Procedure

The procedure used to perform the Repeated Logins Using SSH Version 2 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify that the HTTP and FTP background traffic is running.

Step 3 Verify that dca-agg-1 is configured for ssh login using the show ip ssh command.

The show ip ssh command should show SSH Enabled - version 1.99 in the output.

Step 4 Initiate 6 iterations of the test script. Each iteration will attempt to log into dca-agg-1 1000 times, successively, using SSH version 2.

Step 5 Stop background scripts to collect final status of network devices and analyze for error.

Step 6 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect no system error messages resulting from the multiple, repeated SSH login attempts.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Repeated Logins Using SSH Version 2 passed.

CLI Functionality

Parser testing exercises the command line interface (CLI) of a router. The testing walks the parser tree, executing completed commands and filling in options as it comes to them. Certain branches of the parser tree were left out due to time constraints of the testing (eg. show tag-switching tdp, show mpls).

The following tests were performed:

- CLI Parser Functionality Using SSHv1, page 2-23
- CLI Parser Functionality Using SSHv2, page 2-24
- CLI Parser Functionality Using SSHv1 on 4948, page 2-24
- CLI Parser Functionality Using SSHv2 on 4948, page 2-25

CLI Parser Functionality Using SSHv1

An automated script was used to test the valid **show** and **clear** commands on dca-agg-2. The commands that were tested were a select subset of those tested in the full Native IOS Safe Harbor releases. These commands were chosen based on their relation to differentiating hardware and software features between the traditional Safe Harbor Native IOS topologies and the DCAP topology. SSH version 1 was used as the access protocol.

Test Procedure

The procedure used to perform the CLI Parser Functionality Using SSHv1 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Begin executing the show and clear commands on the device under test.

Step 3 Stop background scripts to collect final status of network devices and analyze for error.

Step 4 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Figure 2-5 CLI Parser Functionality Using SSHv1 Usage

Expected Results

- We expect all commands to be executed without negative impact on the DUT.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

CLI Parser Functionality Using SSHv1 passed with exception CSCsc81109.

CLI Parser Functionality Using SSHv2

This test verified **show** and **clear** commands on dca-agg-2 through an automated script. The commands that were tested were a select subset of those tested in the full Native IOS Safe Harbor releases. These commands were chosen based on their relation to differentiating hardware and software features between the traditional Safe Harbor Native IOS topologies and the DCAP topology. SSH version 2 was used as the access protocol.

Test Procedure

The procedure used to perform the CLI Parser Functionality Using SSHv2 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Begin executing the show and clear commands on the device under test.

Step 3 Stop background scripts to collect final status of network devices and analyze for error.

Step 4 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect all commands to be executed without negative impact on the DUT.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

CLI Parser Functionality Using SSHv2 passed with exception CSCsc81109.

CLI Parser Functionality Using SSHv1 on 4948

This test verified **show** and **clear** commands on dca-acc-4k-1 through the use of an automated script. SSH version 1 was used as the access protocol.

Test Procedure

The procedure used to perform the CLI Parser Functionality Using SSHv1 on 4948 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Begin executing the show and clear commands on the device under test.

Step 3 Stop background scripts to collect final status of network devices and analyze for error.

Step 4 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect all commands to be executed without negative impact on the DUT.

- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

CLI Parser Functionality Using SSHv1 on 4948 passed.

CLI Parser Functionality Using SSHv2 on 4948

This test verified **show** and **clear** commands on dca-acc-4k-2 through the use of an automated script. SSH version 2 was used as the access protocol.

Test Procedure

The procedure used to perform the CLI Parser Functionality Using SSHv2 on 4948 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Begin executing the show and clear commands on the device under test.

Step 3 Stop background scripts to collect final status of network devices and analyze for error.

Step 4 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect all commands to be executed without negative impact on the DUT.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

CLI Parser Functionality Using SSHv2 on 4948 passed.

Security

Resistance to outside attacks is critical to the operation of any data center. This section includes tests that measure the response of the network devices to various common attacks and techniques.

The following tests were performed:

- Malformed SNMP Polling, page 2-25
- Malformed SSH Packets, page 2-26
- NMAP Open Port Scan, page 2-27

Malformed SNMP Polling

Each network device in the Data Center test topology is configured for both read-only and read-write access via SNMP. The availability of SNMP access of certain network devices to the outside world leaves them vulnerable to certain attacks. One possible attack is through the use of malformed SNMP packets.

This test relies on the Protos (http://www.ee.oulu.fi/research/ouspg/protos/) test suite for SNMP. This test application subjected the DUT to many hundreds of misconfigured SNMP packets in an attempt to disrupt system activity. The Protos SNMP test was run against device dca-agg-1 while that device was being monitored for errors and disruptions to CPU and memory stability.

Test Procedure

The procedure used to perform the Malformed SNMP Polling test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 If the background test traffic is not already running, start it now.

Step 3 Verify the SNMP community string settings default using the show running-config | include snmp command on dca-agg-1.

The read-only password is "public" (default).

Step 4 Execute the two protos traffic generation scripts on dca-agg-1.

Step 5 Stop background scripts to collect final status of network devices and analyze for error.

Step 6 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect all DUTs to not hang, crash, or give tracebacks.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Malformed SNMP Polling passed.

Malformed SSH Packets

Similar to its vulnerability to outside attacks via corrupt SNMP traffic, a network device may be susceptible to outside attacts via corrupt SSH traffic. This test relies on the Protos (http://www.ee.oulu.fi/research/ouspg/protos/) test suite for SSH. This test application subjects the DUT too many hundreds of misconfigured SSH packets in an attempt to disrupt system activity.

The Protos SSH test was run against the datacenter test network device dca-agg-1 while that device was being monitored for errors and disruptions to CPU and memory stability.

Test Procedure

The procedure used to perform the Malformed SSH Packets test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 If the background test traffic is not already running, start it now.

Step 3 Verify that dca-agg-1 is configured with a hostname, domain name, and TACACS authentication on the VTY lines using the following commands:

show running-config | include hostname|domain|aaa|tacacs

show running-config | begin line vty 0

The lines that should be present are as follows:

```
hostname dca-agg-1
aaa new-model
aaa authentication login default group tacacs+ local
aaa authorization exec default group tacacs+ if-authenticated local
aaa session-id common
ip domain-name example.com
tacacs-server host 172.18.177.132
tacacs-server host 172.18.179.180
tacacs-server directed-request
tacacs-server key cisco
line vty 0 4
 transport input telnet ssh
```

Step 4 Verify the SSH server on dca-agg-1 is enabled using the show ip ssh command and that dca-agg-1 is accepting SSH connections.

Step 5 Send malformed SSH packets to the device while monitoring device. Monitor the device ensuring it does not hang, crash or reload.

Step 6 Stop background scripts to collect final status of network devices and analyze for error.

Step 7 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that SSH vulnerability testing does not cause the router to reload, hang, or crash.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Malformed SSH Packets passed.

NMAP Open Port Scan

A common way for hackers to wreak havoc on a network is to scan a network device (or an endpoint) for open TCP or UDP ports using the freely available NMAP tool. If an open port is found, the hacker may be able to exploit it and disrupt system activity. It is important, therefore, that a network device only leave those ports open that need to be for normal network services.

The test devices in the Data Center test topology have certain ports open by design. These include Telnet (port 23) and SSH (22). This test runs the NMAP port scan tool against each device in the test topology, verifying that there are no ports open other than the ones expected. The DUTs are monitored for errors and CPU and memory stability during this procedure.

Test Procedure

The procedure used to perform the NMAP Open Port Scan test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Begin a port span on the Supervisor 720 devices in the testbed using the NMAP tool.

The command, run as root, that was used to execute this step was nmap -v -p 1-65535 target_ip.

Step 3 Verify that all open ports (as revealed by the port scan) are expected.

Each of the devices in the datacenter test topology have telnet (TCP port 23) and SSH (TCP 22) open. These are the only ports we expect to see open.

Step 4 Stop background scripts to collect final status of network devices and analyze for error.

Step 5 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect open ports revealed by the NMAP tool as presumed.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

NMAP Open Port Scan passed.

Traffic Forwarding

Traffic forwarding measures basic traffic forwarding features and abilities of the DCAP 1.0 test topology.

The following tests were performed:

- Zero Packet Loss, page 2-28
- Distributed FIB Consistency, page 2-29

Zero Packet Loss

This test verified that the network devices in the Data Center topology are able to forward basic network traffic, without loss, in a steady-state condition. Web (HTTP/HTTPS) and FTP traffic consisting of varying frame sizes is sent between client devices and web servers. There are no negative, or failure, events introduced during this test. The network devices will all be monitored for errors, as well as for CPU and memory usage stability.

Test Procedure

The procedure used to perform the Zero Packet Loss test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Begin the background test traffic that will send 30 minutes' worth of HTTP, HTTPS, and FTP traffic between the clients and the servers.

Step 3 When the traffic completes, measure the percentage of connection attempts that resulted in error codes. This percentage should be less than 1%.

Step 4 Stop background scripts to collect final status of network devices and analyze for error.

Step 5 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Figure 2-6 Zero Packet Loss Usage

Expected Results

- We expect traffic loss due to background test traffic to be tolerable.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Zero Packet Loss passed.

Distributed FIB Consistency

Hardware forwarding in the Catalyst 6500 is accomplished by providing a specialized forwarding ASIC with a copy of the switch routing table. This Forwarding Information Base (FIB), located on the PFC3 of the Supervisor 720 Engine, contains only the information from the routing table that is necessary for making a forwarding decision, including the network prefix of the route, the next-hop address, and the egress interface. Because this FIB is located on the Supervisor 720 Engine itself, the traffic must go here to be forwarded. This type of hardware forwarding is referred to as centralized.

The Catalyst 6500 switch family also allows for distributed forwarding in hardware through the use of Distributed Forwarding Cards (DFC). These daughter cards, which, in the Data Center topology are located on the WS-X6704-10GE modules in the Aggregation Layer, are equipped with their own FIB, again a forwarding ASIC. This distributed FIB is synchronized with the FIB residing on the PFC3. The end result is faster forwarding, since forwarding lookups can be done locally on the line card and aren't needed from the Supervisor Engine.

This test verified that the FIBs on the WS-X6704-10GE linecards in the Aggregation Layer are properly synchronized. In each Aggregation device, dca-agg-1 and dca-agg-2, the central FIB is inspected and compared to each of the five distributed FIBs. The devices under test were monitored for errors as well as CPU and memory utilization issues.

Test Procedure

The procedure used to perform the LACP Basic Functionality test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Log into dca-agg-1 and use the show module command to verify the location of any DFCs in the system.

There are DFCs in each of slots 9-13 in dca-agg-1.

Step 3 Use the show ip cef command to verify that there is no difference between the FIB on the PFC and the FIB on the DFC in slot 9.

Step 4 Use the show ip cef command to verify that there is no difference between the FIB on the PFC and the FIB on the DFC in slot 10.

Step 5 Use the show ip cef command to verify that there is no difference between the FIB on the PFC and the FIB on the DFC in slot 11.

Step 6 Use the show ip cef command to verify that there is no difference between the FIB on the PFC and the FIB on the DFC in slot 12.

Step 7 Use the show ip cef command to verify that there is no difference between the FIB on the PFC and the FIB on the DFC in slot 13.

Step 8 Log into dca-agg-2 and use the show module command to verify the location of any DFCs in the system.

There are DFCs in each of slots 9-13 in dca-agg-2.

Step 9 Use the show ip cef command to verify that there is no difference between the FIB on the PFC and the FIB on the DFC in slot 9.

Step 10 Use the show ip cef command to verify that there is no difference between the FIB on the PFC and the FIB on the DFC in slot 10.

Step 11 Use the show ip cef command to verify that there is no difference between the FIB on the PFC and the FIB on the DFC in slot 11.

Step 12 Use the show ip cef command to verify that there is no difference between the FIB on the PFC and the FIB on the DFC in slot 12.

Step 13 Use the show ip cef command to verify that there is no difference between the FIB on the PFC and the FIB on the DFC in slot 13.

Step 14 Stop background scripts to collect final status of network devices and analyze for error.

Step 15 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect total parity between the FIB maintained on the supervisor (PFC) and the FIBs maintained on the DFCs.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

LACP Basic Functionality passed.

Layer 2 Protocols

Layer 2 testing looks at several key protocols running at the Data Link Layer (Layer 2) of the OSI model, in the Cisco DCAP 2.0 test topology.

The following test features were conducted:

- Link Aggregation Control Protocol (LACP), page 2-31
- Trunking, page 2-33
- Spanning Tree, page 2-34
- Unidirectional Link Detection (UDLD), page 2-38

Link Aggregation Control Protocol (LACP)

There are several ways that a channel can be formed using the LACP protocol. The channel that is used in the Data Center test topology is configured using LACP active mode, in which the port initiates negotiations with other ports by sending LACP packets.

The following tests were performed:

- LACP Basic Functionality, page 2-31
- LACP Load Balancing, page 2-32

LACP Basic Functionality

There are several ways that a channel can be formed using the LACP protocol. The channel that is used in the Data Center test topology is configured using LACP active mode, in which the port initiates negotiations with other ports by sending LACP packets. This test verified that the channel is formed correctly between dca-agg-1 and dca-agg-2. The CPU and memory utilization are monitored for stability during this test.

Test Procedure

The procedure used to perform the LACP Basic Functionality test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 On both dca-agg-1 and dca-agg-2, it is interfaces 10-Gigabit Ethernet 9/3 and 10-Gigabit Ethernet 10/3 that are bundled to form Port-channel 1 using LACP. Use the show running-config interface command on each of these interfaces to verify that LACP is configured for active mode.

The following lines are present on each of these four interfaces:

```
channel-protocol lacp
channel-group 1 mode active
```

Step 3 Use the show interfaces Port-channel 1 etherchannel command on both dca-agg-1 and dca-agg-2 to verify that both interfaces Te9/3 and Te10/3 are bundled and active in the port-channel.

The "Number of ports" in each case should be given as "2". Further, each of the two interfaces should be listed as "Ports in the Port-channel" and their "EC state" should be "Active".

Step 4 Stop background scripts to collect final status of network devices and analyze for error.

Step 5 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the LACP-formed channels to build correctly.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

LACP Basic Functionality passed.

LACP Load Balancing

When the next-hop for network traffic is out an EtherChannel, the switch must decide which of the bundled physical interfaces to send the network traffic out. Further, the switch must have the ability to balance any traffic going out an EtherChannel across the multiple available physical interfaces (anything less would be a waste of available bandwidth). In Native IOS, there are several EtherChannel load-balancing algorithms available for the network administrator to use to get the best balance of traffic across all available physical interfaces.

The algorithm used in the Data Center test topology makes the load balancing decision (which physical port to send the traffic out) based on a combination of the source and destination Layer 4 ports. This test verified that both physical interfaces in the EtherChannel between dca-agg-1 and dca-agg-2 passed traffic when a diversity of traffic was sent across the EtherChannel. The Aggregation layer devices were monitored for any errors. The CPU and memory utilization were monitored for stability.

Test Procedure

The procedure used to perform the LACP Load Balancing test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Use the show interfaces Port-channel 1 etherchannel command on dca-agg-1 and dca-agg-2 to verify that a two-port channel is active between the two devices.

The channel shows ports Te9/3 and Te10/3 in Active state.

Step 3 Use the show running-config | include load-balance command to verify that dca-agg-1 and dca-agg-2 are configured to do Layer 4 source/destination load-balancing.

The configuration command port-channel load-balance src-dst-port is present on both devices.

Step 4 Clear the traffic counters on dca-agg-1 and dca-agg-2 using the clear counters command.

Step 5 Begin a 5-minute period of the background test traffic.

Step 6 When the traffic has finished, use the show interfaces Port-channel 1 counters etherchannel command on dca-agg-1 and dca-agg-2 to verify that traffic was sent on both ports of the EtherChannel.

The distribution of traffic may or may not be equal, depending on the distribution of source and destination ports for ingress and egress traffic.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect traffic to be distributed between both links of the EtherChannel connecting dca-agg-1 and dca-agg-2.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

LACP Load Balancing passed.

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. The table lists and describes the five modes of trunking on Cisco switches.

The following test was performed:

- 802.1q Trunking Basic Functionality, page 2-33

802.1q Trunking Basic Functionality

On Cisco Catalyst 6500 and Catalyst 4900 switches, trunks can be formed in multiple ways. Trunking can either be dynamic, in which trunking is negotiated between the two sides of the link, or it can be set to on or off, statically. In the case of the Data Center test topology, trunk links are set to on, meaning they trunk VLANs regardless of what the remote side of the link is doing.

The trunk encapsulation can also be either dynamically negotiated or set statically. In the Data Center test topology, the encapsulation is set statically to 802.1q, or dot1q.

This test verified that the links that are configured as trunk links between the Data Center devices actually form trunks correctly. The links looked at include those between two Catalyst 6500s (dca-agg-2 and dca-acc-6k-1) and those between a Catalyst 6500 and a Catalyst 4900 (dca-agg-2 and dca-acc-4k-2). The CPU and memory utilization of the DUTs was monitored for stability.

Test Procedure

The procedure used to perform the 802.1q Trunking Basic Functionality test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 The devices dca-agg-2 and dca-acc-6k-1, both Catalyst 6500s are connected by a static trunk. Use the show running-config interface interface and show interfaces interface trunk commands to verify that this is the current configuration and the trunk is currently working.

Step 3 Using the shutdown and no shutdown commands, flap the Te9/4 interface on dca-agg-2.

Step 4 Use the show interfaces interface trunk command to verify that the trunk between dca-agg-2 and dca-acc-6k-1 has re-formed correctly.

Step 5 The devices dca-agg-2 (a Catalyst 6500) and dca-acc-4k-2 (a Catalyst 4900) are connected by a static trunk. Use the show running-config interface interface and show interfaces interface trunk commands to verify that this is the current configuration and the trunk is currently working.

Step 6 Using the shutdown and no shutdown commands, flap the Te10/2 interface on dca-agg-2.

Step 7 Use the show interfaces interface trunk command to verify that the trunk between dca-agg-2 and dca-acc-4k-2 has re-formed correctly.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect 802.1q trunks to be formed correctly between the two Catalyst 6500 devices.
- We expect 802.1q trunks to be formed correctly between the Catalyst 6500 and the Catalyst 4900.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

802.1q Trunking Basic Functionality passed.

Spanning Tree

Each of the seven devices in the Cisco DCAP 2.0 topology Layer 2 domain participates in Spanning-Tree. The Spanning-Tree Protocol (STP) that is used in the DCAP topology is PVST+ plus the rapid convergence enhancements of IEEE 802.1w (collectively referred to as Rapid PVST+ or rPVST+). This group of tests looks at the basic functionality of rPVST+ as well as some of the commonly-used STP features.

The following tests were performed:

- Rapid PVST+ Basic Functionality, page 2-34
- Root Guard, page 2-36

Rapid PVST+ Basic Functionality

In the Data Center test topology dca-agg-1 is configured to be the primary root switch for all VLANs, while dca-agg-2 is configured to be the secondary root switch. This test does not focus so much on the ability of rPVST+ to converge quickly and accurately as it does on the fundamental mechanics of STP. It verifies that the correct switch is root and that all Layer 2 interfaces in the Data Center Layer 2 domain are in the correct STP state, with the correct switch identified as root, for all VLANs.

Test Procedure

The procedure used to perform the Rapid PVST+ Basic Functionality test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify that spanning-tree configurations on all Layer 2 devices in the DCAP test topology using the show running-configuration | include spanning-tree.

On devices dca-agg-1, dca-agg-2, dca-acc-6k-1, dca-acc-6k-2, dca-acc-4k-1, dca-acc-4k-2 and dca-voodoo-2, the following lines are present:

spanning-tree mode rapid-pvst

spanning-tree extend system-id

spanning-tree pathcost method long

On dca-agg-1 (which is configured as the root switch for all VLANs) the following configuration line should be present:

spanning-tree vlan 1-4094 priority 24576

On dca-agg-2 (which is configured as the secondary root switch for all VLANs) the following configuration line should be present:

spanning-tree vlan 1-4094 priority 28672

Step 3 Verify that the system with MAC address 0015.c719.bf80 is root for all VLANs on all systems using the show spanning-tree root command.

A different system may be shown as root for VLAN 1 on some systems. This is expected as VLAN 1 is active only because it is allowed on the CSM port-channel (Po258) in both dca-agg-1 and dca-agg-2. For this reason, each of those two devices will report their respective local MAC addresses as being root for VLAN 1.

Step 4 Verify that dca-agg-1 is the system that owns the root MAC address 0015.c719.bf80 using the show catalyst6000 chassis-mac-addresses command.

Note On the Catalyst 4900 systems, dca-acc-4k-1 and dca-acc-4k-2, the show module command will be used to verify that this root MAC address does not fall into the range of system MAC addresses.

Step 5 Use the show spanning-tree vlan 2101 command to map the spanning-tree for VLAN 2101 as an example of what the per-VLAN STP topology should look like.

- The device dca-agg-1, which is the STP root for VLAN 2101, should report all interfaces in "FWD" state. This list of interfaces includes Te9/4, Te10/1, Te10/2, Te10/4, Po1 and Po270 (the FWSM/backplane interface). The Root ID Address should show the root MAC address "0015.c719.bf80" as should the Bridge ID Address (this switch is root). The Root ID Priority and the Bridge ID Priority should also be the same value, "25677", which is the configured priority of "24576" plus the VLAN, 2101.
- The device dca-agg-2, which is the secondary STP root for VLAN 2101, should report all interfaces in "FWD" state. This list of interfaces includes Te9/4, Te10/1, Te10/2, Te10/4, Po1 and Po270 (the FWSM/backplane interface). The Root ID Address should show the root MAC address "0015.c719.bf80". The Bridge ID Address should show the local system MAC address "0015.c734.9d80". The Root ID Priority should be "25677", while the Bridge ID Priority should be "30773", which is the configured priority of 28672 plus the VLAN, 2101.
- The device dca-acc-6k-1, should report interface Te1/1 in "FWD" state and Te1/2 in "BLK" state. All other interfaces (connected to the servers in the DCAP test topology) should be in "FWD" state. The Root ID Address should show the root MAC address "0015.c719.bf80". The Bridge ID Address should show the local system MAC address "0016.9cb5.c000". The Root ID Priority should be "25677", while the Bridge ID Priority should be "34869", which is the default priority of 32768 plus the VLAN, 2101.
- The device dca-acc-6k-2, should report interface Te1/1 in "FWD" state and Te1/2 in "BLK" state. All other interfaces (connected to the servers in the DCAP test topology) should be in "FWD" state. The Root ID Address should show the root MAC address "0015.c719.bf80". The Bridge ID Address should show the local system MAC address "0016.9c9e.a000". The Root ID Priority should be "25677", while the Bridge ID Priority should be "34869", which is the default priority of 32768 plus the VLAN, 2101.
- The device dca-acc-4k-1, should report both interfaces Te1/49 and Te1/50 in "FWD" state. All other interfaces (connected to the servers in the DCAP test topology) should also be in "FWD" state. The Root ID Address should show the root MAC address "0015.c719.bf80". The Bridge ID Address should show the local system MAC address "0015.fa80.4f80". The Root ID Priority should be "25677", while the Bridge ID Priority should be "34869", which is the default priority of 32768 plus the VLAN, 2101.

- The device dca-acc-4k-2, should report interface Te1/49 in "FWD" state and Te1/50 in "BLK" state. All other interfaces (connected to the servers in the DCAP test topology) should be in "FWD" state. The Root ID Address should show the root MAC address "0015.c719.bf80". The Bridge ID Address should show the local system MAC address "0015.fa80.4f40". The Root ID Priority should be "25677", while the Bridge ID Priority should be "34869", which is the default priority of 32768 plus the VLAN, 2101.
- The device dca-voodoo-2, should report interface Te1/1 in "FWD" state and Te1/2 in "BLK" state. All other interfaces (connected to the servers in the DCAP test topology) should be in "FWD" state. The Root ID Address should show the root MAC address "0015.c719.bf80". The Bridge ID Address should show the local system MAC address "000f.f827.4d80". The Root ID Priority should be "25677", while the Bridge ID Priority should be "34869", which is the default priority of 32768 plus the VLAN, 2101.

Step 6 Use the show spanning-tree summary command to verify that all VLANs on the primary root (dca-agg-1) and secondary root (dca-agg-2) are in "Forwarding" state.

The very last line of the output for this command gives a summary for all VLANs. There should be no VLANs in any state other than "Forwarding".

Step 7 Use the show spanning-tree summary command to verify that all VLANs on the access switches (dca-acc-6k-1, dca-acc-6k-2, dca-acc-4k-1, dca-acc-4k-2 and dca-voodoo-2) have a single port in "Blocking" state (with the exception of dca-acc-4k-1, which will have all "Forwarding").

In each VLAN row, in the output of this command, there should be a "1", indicating a single port is "Blocking" for that VLAN.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the spanning-trees for each VLAN to be converged correctly, with the appropriate ports in the appropriate STP state.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Rapid PVST+ Basic Functionality passed.

Root Guard

Spanning-tree best practices dictate that the root switch (and even backup root switch) should be designated and forced into that role by the network designer. This is done by configuring the STP priority to an appropriate value lower than the default of 32768. A deterministic root switch results in deterministic network traffic and predictable behavior.

This predictable behavior can be disrupted, however, should a switch be inserted into the network topology (accidentally or maliciously) with a lower STP priority or Bridge ID than the configured root switch. The STP Root Guard feature helps to protect against such a situation by building a wall of protection around the configured root switch(es).

Interfaces that are connecting the root switch(es) to the access layer switches are configured locally with the Root Guard feature. Should the root (or secondary root) receive a BPDU on any of these interfaces with a lower Bridge ID than it has, the interface will be transitioned into a "Root Inconsistent" state. This is essentially a perpetual "Listening" state in which the interface can continue to monitor the link for errant BPDUs (or their absence), but not forward any data traffic. When the interface stops receiving such BPDUs, it transitions back to the appropriate STP state.

In the DCAP test topology, dca-agg-1 is configured to be the primary STP root while dca-agg-2 is configured to be the secondary STP root. The interfaces that connect these two switches to the Layer 2 access devices are configured with STP Root Guard enabled. The Port-channel interface connecting these two aggregation devices has Root Guard disabled.

In this test, Port-channel 1 (connecting dca-agg-1 and dca-agg-2) will be broken. When this happens, the interfaces connecting the access switches to dca-agg-2 will transition from Blocking to Forwarding state and begin to forward BPDUs from dca-agg-1 to dca-agg-2. The device dca-agg-2, now receiving BPDUs of a lower priority (from the STP root), will move the links on which it is receiving such BPDUs to Root Inconsistent state. When Port-channel 1 is reconnected, the links will return to Forwarding state.

Test Procedure

The procedure used to perform the Root Guard test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Use the show running-config | include spanning-tree command to verify that dca-agg-1 is configured to be the primary STP root switch and dca-agg-2 is configured to be the secondary root switch.

The device dca-agg-1 is configured with a priority of 24576, the lowest of any switches in the DCAP test topology. It will therefore assume the primary STP root role. The device dca-agg-2 is configured with a priority of 28672, the lowest of any switches in the DCAP test topology. It will therefore assume the secondary STP root role.

Step 3 Use the show running-config interface interface command to verify that interfaces Te9/4, Te10/1, Te10/2 and Te10/4 on both dca-agg-1 and dca-agg-2 are configured with spanning-tree guard root.

Step 4 Use the show spanning-tree interface interface command to verify that interfaces Te9/4, Te10/1, Te10/2 and Te10/4 on both dca-agg-1 and dca-agg-2 are in STP Forwarding state for all VLANs.

Step 5 Verify that there are no interfaces in "Root Inconsistent" state on either dca-agg-1 or dca-agg-2 using the show spanning-tree inconsistentports command.

Step 6 Shutdown Port-channel 1 on dca-agg-2.

Step 7 Verify that Te9/4, Te10/1, Te10/2, Te10/4 on dca-agg-2 are in "Root Inconsistent" state using the show spanning-tree inconsistentports command.

Each of these interfaces should be listed per VLAN in the output of this command.

Step 8 Bring Port-channel 1 on dca-agg-2 back online using the no shutdown command.

Step 9 Use the show spanning-tree interface interface command to verify that interfaces Te9/4, Te10/1, Te10/2 and Te10/4 on both dca-agg-1 and dca-agg-2 are again in STP Forwarding state for all VLANs.

Step 10 Verify that there are again no interfaces in "Root Inconsistent" state on either dca-agg-1 or dca-agg-2 using the show spanning-tree inconsistentports command.

Step 11 Stop background scripts to collect final status of network devices and analyze for error.

Step 12 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the interfaces with STP root guard enabled to transition to "Root Inconsistent" state when they begin receiving BPDUs with lower priority than the local BPDU.
- We expect the interfaces with STP root guard enabled to return to Forwarding state when they stop receiving such BPDUs.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Root Guard passed.

Unidirectional Link Detection (UDLD)

The Unidirectional Link Detection (UDLD) protocol allows devices connected through fiber-optic or copper Ethernet cables (for example, Category 5 cabling) to monitor the physical status 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.

The following test was performed:

- UDLD Detection on 10GE Links, page 2-38

UDLD Detection on 10GE Links

This test forced a unidirectional link condition on one of the 10-Gigabit Ethernet links in the Data Center test topology and verified that the link was put into a UDLD down state correctly.

Devices dca-agg-1 and dca-agg-2 are connected via two 10-Gigabit Ethernet links, Te13/1 and Te13/2 (on each device). On dca-agg-1, the tx fibers of Te13/1 and Te13/2 were switched, creating a crossed-fiber situation.

This test verified that the DUTs were monitored for errors during this test. The CPU and memory utilization on the DUTs were also monitored for stability.

Test Procedure

The procedure used to perform the UDLD Detection on 10GE Links test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Use the show udld interface command to verify the current UDLD state of interfaces Te13/1 and Te13/2 on both dca-agg-1 and dca-agg-2.

The operational state for all interfaces should be "Enabled / in aggressive mode". The current bidirectional state should be "Bidirectional". There should also be a single neighbor entry showing "Bidirectional" as the current bidirectional state.

Step 3 Switch the transmit fibers of interfaces Te13/1 and Te13/2 on dca-agg-1. Verify that the system log contains at least one UDLD link detection interface disable message.

Step 4 Verify the interface status for all four interfaces using the show interface interface status command.

Interface Te13/2 on dca-agg-1 and dca-agg-2 are put into errdisable state.

Step 5 Use the show udld interface command to verify the current UDLD state of interfaces Te13/1 and Te13/2 on dca-agg-1 and dca-agg-2.

Step 6 Return the transmit fibers to their original location and flap interface Te13/2 on dca-agg-1 and dca-agg-2 using the shutdown and no shutdown commands.

Step 7 Use the show udld interface command to verify that interfaces Te13/1 and Te13/2 on both dca-agg-1 and dca-agg-2 have returned to the original UDLD states.

The operational state for all interfaces should be "Enabled / in aggressive mode". The current bidirectional state should be "Bidirectional". There should also be a single neighbor entry showing "Bidirectional" as the current bidirectional state.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect one or more switches to have ports in errdisable state when UDLD determines a fiber has been crossed between ports.
- We expect UDLD Link State to be shut down for the port determined to be unidirectional.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

UDLD Detection on 10GE Links passed.

Layer 3 Protocols

Layer 3 tests look at several key protocols running at the Network Layer (Layer 3) of the OSI model in the DCAP test topology.

The following test features were conducted:

- Hot Standby Router Protocol (HSRP), page 2-39
- Open Shortest Path First (OSPF), page 2-41

Hot Standby Router Protocol (HSRP)

Hot Standby Router Protocol (HSRP) is used to provide a redundant gateway IP address to clients on a particular subnet. In the DCAP test topology, the virtual IP address (gateway) is shared by two routers, dca-agg-1 and dca-agg-2. Each of these two routers is configured with two IP addresses per HSRP subnet, one that is unique to that router, and one that is shared with the peer HSRP router. The router with the highest HSRP priority will assume Active state and respond to queries on the Virtual IP. The other router will assume Standby state and ignore such queries, while in Standby state.

The following test was performed:

- HSRP Basic Functionality, page 2-40

HSRP Basic Functionality

There are 200 HSRP groups in the DCAP test topology, providing virtual gateways for over 200 subnets. This test verified that the Aggregation Layer devices were able to scale to this number of standby groups. It verified that the correct router was in Active HSRP state and that only that router was displaying the HSRP MAC address.

Test Procedure

The procedure used to perform the HSRP Basic Functionality test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify the HSRP configuration for VLANs 1101-1300 on dca-agg-1 and dca-agg-2 using the show running-config | begin interface Vlan1101 command.

On dca-agg-1, these 200 VLANs are configured with a standby priority of 120. On dca-agg-2, they are configured with a standby priority of 110. Each VLAN has a standby IP address, and each belongs to a separate standby group (specified by the number directly following standby).

Step 3 Use the show standby brief command to verify that dca-agg-1 is "Active" for VLANs1101-1300 and that dca-agg-2 is "Standby".

Step 4 Verify that dca-agg-1 has a virtual MAC address running on each of the standby VLANs for which it is the active HSRP router using the show standby | include Vlan|Virtual mac.

Each VLAN has a virtual MAC address assigned to it.

Step 5 Verify that dca-agg-2 does not have a virtual MAC address running on the standby VLANs for which it is the standby HSRP router using the show standby | include Vlan|Virtual mac.

None of the VLANs have a virtual MAC address assigned to it.

Step 6 Stop background scripts to collect final status of network devices and analyze for error.

Step 7 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect all HSRP groups to reflect their configuration in their active and standby states.
- We expect all Active HSRP groups to have an associated Virtual MAC address and that no Standby HSRP groups will have a MAC address.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

HSRP Basic Functionality passed.

Open Shortest Path First (OSPF)

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 test was performed:

- OSPF Route Summarization, page 2-41
- OSPF Database Verification, page 2-42

OSPF Route Summarization

In the DCAP test topology, in OSPF Area 10, there are several /30 subnets configured for the inter-switch links. These all share a 172.31.1.x prefix. All servers in Area 10 are on /24 subnets and share a prefix of 101.1.x.x.

The Core routers in the Data Center test topology, dca-core-1 and dca-core-2, as part of the default configuration, summarize the subnets in Area 10 for advertisement into Area 0. The 172.31.1.x/30 subnets are configured to summarize as 172.31.1.0/24 networks while the 101.1.x.x/24 subnets are configured to summarize as 101.1.0.0/16 networks.

This test verified that summarization was occurring by looking at the routing table of a device in Area 20. The memory and CPU utilization were monitored for stability.

Test Procedure

The procedure used to perform the OSPF Route Summarization test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Use the show running-config | begin router ospf command on dca-core-1 and dca-core-2 to verify that summarization is configured.

Each of the devices will have the following two lines as part of their OSPF configuration:

area 10 range 101.1.0.0 255.255.0.0 area 10 range 172.31.1.0 255.255.255.0

The first line will cause any networks in Area 10 matching 101.1.x.x to be summarized into a /16. The second line will cause any networks in Area 10 matching 172.31.1.x to be summarized into a /24.

Step 3 On dca-agg-1, use the show running-config interface Te9/2 to verify that this device has an interface with a /30 address matching the 172.31.1.x format.

The IP address of interface Te9/2 is 172.31.1.14/30.

Step 4 On dca-voodoo-2, an Area Border Router in Area 20, use the show ip route 172.31.1.14 command to verify that the route to this address has been summarized as a /24.

The output of this command will show a "Routing entry for 172.31.1.0/24".

Step 5 On dca-agg-1, use the show running-config interface Vlan1101 to verify that this device has an interface with a /24 address matching the 101.1.x.x format.

The IP address of interface Vlan1101 is 101.1.1.2/24.

Step 6 On dca-voodoo-2, use the show ip route 101.1.1.2 command to verify that the route to this address has been summarized as a /16.

The output of this command will show a "Routing entry for 101.1.0.0/16".

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the appropriate networks to be summarized correctly by the OSPF Area Border Routers.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

OSPF Route Summarization passed.

OSPF Database Verification

Each Layer 3 device in an Autonomous System (AS) running OSPF maintains a detailed view, or database, of the entire OSPF network. This database contains information gathered about networks that have been advertised via OSPF, and OSPF routers in the network. Information about networks and OSPF routers is propagated through the OSPF network using several different types of Link State Algorithm (LSA) messages.

This test verified that the OSPF database contains the information that would be expected for this particular test topology.

Test Procedure

The procedure used to perform the OSPF Database Verification test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 From each router in the DCAP test topology that is running OSPF, gather information about the OSPF interfaces and the OSPF processes that are running using the show ip ospf database, show ip ospf database database-summary and show ip ospf interfaces commands.

The devices in the DCAP test topology that are running OSPF include dca-agg-1, dca-agg-2, dca-core-1, dca-core-2, and dca-voodoo-2.

Step 3 Verify that the number of Router (Type 1) LSAs in each area is what is expected.

For each router in a particular area, there should be a single router LSA given.

Step 4 Verify that the number of Network (Type 2) LSAs in each area is what is expected.

A Type 2 LSA is generated when there exists a network segment that has both a DR and a BDR. In other words, when there is a network shared by two devices both running OSPF on that network, a network LSA is generated. The number of Network LSAs is determined by how many of those types of links are in a given area.

Step 5 Verify that the number of Summary Network (Type 3) LSAs in each area is what is expected.

The ABR (Area Border Router), or the router that is at the border of two or more areas, sends a summary network LSA into Area X for all other are other than X for which it is a participant. So if you have Areas X, Y and Z and RouterA is a participant in all of them, it will generate a Type-3 LSA and send it into Area X. This LSA will contain a summary of all the networks found in Areas Y and Z.

Step 6 Verify that the number of Summary ASBR (Type 4) LSAs in each area is what is expected.

An ABR between areas X and Y will generate a Summary ASBR into X for an ASBR which exists in Y. However, an ABR which is an ASBR will not advertise itself. In each area, verify there is a Summary ASBR LSA from each ABR in that area for each ASBR outside this area.

Step 7 Verify that the number of AS-External (Type 5) LSAs in each area is what is expected.

This is pretty much the number of networks that are being redistributed from another routing protocol into OSPF times the number of ABRs that share redistribution responsibilities.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the OSPF database to populated with the correct number of LSAs of each type for the topology being used.

Results

OSPF Database Verification passed.

Negative Testing

Negative tests executed measure the response of the test topology to various negative events, such as simulated hardware failures, link failures and whole device failures.

The following test features were conducted:

- Hardware Failure, page 2-43
- Link Failure, page 2-61

Hardware Failure

Hardware failure testing measures the ability of the DCAP test topology to absorb various hardware failures, including system crashes and module resets.

The following test was performed:

- Access Layer Supervisor Failover Using SSO with NSF, page 2-44
- Repeated Reset of Standby Supervisor in Access Layer, page 2-45
- Reset of Aggregation Layer Device dca-agg-1, page 2-46
- Reset of Aggregation Layer Device dca-agg-2, page 2-47
- Reset of Core Layer Device dca-core-1, page 2-49

- Reset of Core Layer Device dca-core-2, page 2-49
- Spanning-Tree Primary Root Failure and Recovery, page 2-50
- HSRP Failover with Fast Timers, page 2-54
- HSRP Recovery From System Failure, page 2-57
- Failure of EtherChannel Module on dca-agg-1, page 2-58
- Failure of EtherChannel Module on dca-agg-2, page 2-59

Access Layer Supervisor Failover Using SSO with NSF

Of the failover protocols that Cisco Catalyst 6500 series switches support, SSO is the most aggressive and provides the shortest downtimes. With the Stateful Switchover protocol, the standby supervisor is fully booted and ready, with a copy of the synchronized configuration received from the active supervisor.

Coupled with the Non-Stop Forwarding feature, which allows the forwarding of traffic even while forwarding tables are being rebuilt by the new supervisor, SSO has the ability to provide sub-second failovers.

In the DCAP test topology, the only devices with supervisor redundancy are the Catalyst 6500 access switches, dca-acc-6k-1 and dca-acc-6k-2. This test measures the effect of an SSO/NSF failover on connections facilitated by dca-acc-6k-2. A series of ten SSO/NSF failovers will be performed on dca-acc-6k-2 while background test traffic and a measurable traffic stream are being run.

Test Procedure

The procedure used to perform the Access Layer Supervisor Failover Using SSO with NSF test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify that both supervisors in dca-acc-6k-2 are online using the show module command.

The supervisors are listed in slots 5 and 6. One is listed as "Active" and the other as "Hot".

Step 3 Verify that the system is ready for an SSO failover using the show redundancy states command.

The operational redundancy mode is "sso" and the peer state is "STANDBY HOT".

Step 4 Begin running about an hour's worth of traffic. This will include both the background test traffic and the measurable traffic stream.

Step 5 Once traffic is running, force a failover of the active supervisor on dca-acc-6k-2 using the redundancy force-switchover command.

Step 6 When the failed supervisor reboots and comes back online, verify that it is online using the show module command and that it is ready for another SSO redundancy failover using the show redundancy states command.

Step 7 Use the show logging command to verify that no errors occurred during the failover and recovery.

Step 8 Repeat the failover scenario 9 more times (for a total of ten). After each failover, verify the system status using the show module, show redundancy states and show logging commands.

Step 9 When all of the failovers are complete, analyze the traffic statistics for excessive errors and to verify that none of the failovers resulted in more than 3 seconds of traffic loss.

Step 10 Stop background scripts to collect final status of network devices and analyze for error.

Step 11 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the system to recover from each forced failover successfully.
- We expect the failed supervisor to come back online in the SSO standby role following each failover.
- We expect each failover to result in less than 3 seconds of traffic loss.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Access Layer Supervisor Failover Using SSO with NSF passed.

Repeated Reset of Standby Supervisor in Access Layer

The Catalyst 6500 systems used in the Access Layer in the DCAP test topology are equipped with dual supervisors for intra-chassis redundancy. The resetting of the standby supervisor in either of these systems should not result in any errors in the system or impact to traffic. This test verified that neither resulted from a repeated reset of the standby supervisor in dca-acc-6k-2.

Test Procedure

The procedure used to perform the Repeated Reset of Standby Supervisor in Access Layer test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify that both supervisors in dca-acc-6k-2 are online using the show module command.

The supervisors are listed in slots 5 and 6. One is listed as "Active" and the other as "Hot".

Step 3 Begin running about an hour's worth of traffic. This will include both the background test traffic and the measurable traffic stream.

Step 4 Once traffic is running, reset the standby supervisor on dca-acc-6k-2 (the one shown as "Hot" in the show module output) using the hw-module module module reset command.

Step 5 When the standby supervisor reboots and comes back online, verify that it is online using the show module and again in the "Hot" standby state.

Step 6 Repeat the standby supervisor reset scenario 9 more times (for a total of ten). After each reset, verify the system status using the show module command.

Step 7 When all of the failovers are complete, analyze the traffic statistics for excessive errors and to verify that none of the failovers resulted in more than 3 seconds of traffic loss.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the resetting of the standby supervisor in the Access Layer device will not cause any ill-effects to the system or to traffic in the DCAP test topology.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Repeated Reset of Standby Supervisor in Access Layer passed.

Reset of Aggregation Layer Device dca-agg-1

The Aggregation Layer provides the bulk of services for traffic coming into the datacenter. Services such as server load balancing, SSL decryption and encryption, and firewalling are provided by the Aggregation Layer devices dca-agg-1 and dca-agg-2.

Redundancy in the aggregation layer is important for providing a high level of availability for these services. The DCAP test topology was designed to provide redundancy through a pair of Aggregation Layer devices rather than redundant supervisors because the failover timers for many of the services are set very low. This means that a service failover could be triggered even by a very fast SSO/NSF failover. This is in-line with the goals of predictability with regards to traffic in the datacenter. If anything less than all services fail over, the result is a very unclean picture of traffic using both Aggregation Layer boxes for the various services before arriving at the destination.

By configuration, dca-agg-1 is the primary provider of services in the DCAP test topology. It is the STP root, the Active HSRP router, and it houses the active CSM and FWSM. The standby services lay in wait on dca-agg-2 for a failover event.

This test verified the impact on traffic due to the failure of the Aggregation device dca-agg-1. Background traffic was run using Linux servers as well as measurable traffic using the Shenick test tool. A total of five system resets were performed.

Test Procedure

The procedure used to perform the Reset of Aggregation Layer Device dca-agg-1 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Start the background traffic and the measurable Shenick traffic.

Step 3 Once traffic is running, verify that dca-agg-1 is passing traffic through interfaces Te9/1, Te9/2, Te9/4, Te10/1, Te10/2, and Te10/4 using the show interfaces counters | include Port|Te9|Te10 command.

Step 4 Use the show interfaces counters | include Port|Te9|Te10 command to verify that dca-agg-2 is not passing any substantial traffic.

There will be a decent amount of management traffic being passed by dca-agg-2.

Step 5 Reload dca-agg-1.

Step 6 Use the show interfaces counters | include Port|Te9|Te10 command to verify that dca-agg-2 is now passing traffic.

Step 7 When dca-agg-1 comes back online, verify that it is passing traffic through interfaces Te9/1, Te9/2, and Po1 using the show interfaces counters | include Port|Te9|Te10|Po1 command.

The command in this step asks to look at the traffic counters on Port-channel 1 as well. This is because, following the recovery of dca-agg-1, the FWSM will remain active in dca-agg-2. The version of FWSM code that is running in DCAP Phase One does not have a preempt feature, and so the FWSM in dca-agg-1 never resumes the active role. This is why, below, a manual reset of the FWSM in dca-agg-2 is called for.

This is also the reason that no traffic is seen from dca-agg-1 down to the Access Layer switches, through interfaces Te9/4, Te10/1, Te10/2, or Te10/4. In order for traffic to pass between client and server, it must go through the active FWSM, which is now in dca-agg-2 (and will remain so indefinitely).

Step 8 Use the show interfaces counters | include Port|Te9|Te10|Po1 command to verify that dca-agg-2 is passing traffic over Port-channel 1 as well as the interfaces that connect to the downstream Access Layer devices, Te9/4, Te10/1, Te10/2, and Te10/4.

Step 9 Reboot the FWSM in dca-agg-2 using the hw-module module 1 reset command.

This will force the FWSM in dca-agg-1 back into active mode, the starting point for this failover test.

Step 10 Verify that dca-agg-1 is again passing traffic through interfaces Te9/4, Te10/1, Te10/2, and Te10/4 using the show interfaces counters | include Port|Te9|Te10|Po1 command.

Step 11 Repeat the above sequence of reloading the DUT and checking counters four more times.

Step 12 Verify the traffic that was lost via the Shenick test tool and the Linux-generated tool.

Step 13 Stop background scripts to collect final status of network devices and analyze for error.

Step 14 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect traffic loss of five seconds or less for each failover performed.
- We expect traffic forwarding to resume when the reset device comes back online.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Reset of Aggregation Layer Device dca-agg-1 passed.

Reset of Aggregation Layer Device dca-agg-2

The Aggregation Layer provides the bulk of services for traffic coming into the datacenter. Services such as server load balancing, SSL decryption and encryption, and fire walling are provided by the Aggregation Layer devices dca-agg-1 and dca-agg-2.

Redundancy in the aggregation layer is important for providing a high level of availability for these services. The DCAP test topology was designed to provide redundancy through a pair of Aggregation Layer devices rather than redundant supervisors because the failover timers for many of the services are set very low. This means that a service failover could be triggered even by a very fast SSO/NSF failover. This is in-line with the goals of predictability with regards to traffic in the datacenter. If anything less than all services fail over, the result is a very unclean picture of traffic using both Aggregation Layer boxes for the various services before arriving at the destination.

By configuration, dca-agg-2 is the standby provider of services in the DCAP test topology. It is the STP secondary root, the Standby HSRP router, and it houses the standby CSM and FWSM. These standby services lay in wait on dca-agg-2 for a failover event.

This test verified the impact on traffic due to the failure of the Aggregation device dca-agg-2. Background traffic was run using Linux servers as well as measurable traffic using the Shenick test tool. A total of five system resets were performed.

Test Procedure

The procedure used to perform the Reset of Aggregation Layer Device dca-agg-2 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Start the background traffic and the measurable Shenick traffic.

Step 3 Once traffic is running, verify that dca-agg-2 is not passing any significant data traffic through interfaces Te9/4, Te10/1, Te10/2, Te10/4, or Po1 using the show interfaces counters | include Port|Te9|Te10|Po1 command.

The four 10-Gigabit Ethernet interfaces are connected downstream to the Access Layer switches. Since dca-agg-2 is not providing any active services for datacenter traffic, other than SSL decryption/encryption, there should not be any data traffic transiting dca-agg-2. The EtherChannel Po1 connects dca-agg-2 to dca-agg-1 and, in steady-state, is used solely for management traffic.

Note that in the output, it will show significant traffic being sent to the Access Layer devices. In truth, this is traffic from the Shenick test tool that is being flooded as a result of the tool not answering periodic ARP requests.

Step 4 Reload dca-agg-2.

Step 5 When dca-agg-2 comes back online, verify that it is, again, not passing any substantial traffic using the show interfaces counters | include Port|Te9|Te10|Po1 command.

Significant traffic being sent to the Access Layer devices. In truth, this is traffic from the Shenick test tool that is being flooded as a result of the tool not answering periodic ARP requests.

Step 6 Repeat the above sequence of reloading dca-agg-2 and checking counters four more times.

Step 7 Verify the traffic that was lost via the Shenick test tool and the Linux-generated tool.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect traffic loss of three seconds or less for each failover performed.
- We expect failed supervisors to come back online without manual intervention.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Reset of Aggregation Layer Device dca-agg-2 passed.

Reset of Core Layer Device dca-core-1

The Core Layer is the first stop for traffic coming into the datacenter. As such, redundancy in the core layer is important for maintaining a high level of availability. Having redundant devices running OSPF allows for failover times nearly as fast as could be achieved through redundant supervisors running SSO with NSF. Further, redundant devices provides load balancing mechanisms.

This test verified the impact on traffic due to the failure of the core device dca-core-1. Background traffic was run using Linux servers as well as measurable traffic using the Shenick test tool. A total of five system resets were performed.

Test Procedure

The procedure used to perform the Reset of Core Layer Device dca-core-1 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Start the background traffic and the measurable Shenick traffic.

Step 3 Once traffic is running, verify that dca-core-1 is passing traffic through interfaces Te1/2 and Te1/4 using the show interfaces counters | include Port|Te1 command.

Step 4 Reload dca-core-1.

Step 5 When dca-core-1 comes back online, verify that it is again passing traffic through interfaces Te1/2 and Te1/4 using the show interfaces counters | include Port|Te1 command.

Step 6 Repeat the above sequence of reloading the DUT and checking counters four more times.

Step 7 Verify the traffic that was lost via the Shenick test tool and the Linux-generated tool.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect raffic loss of three seconds or less for each failover performed.
- We expect traffic forwarding to resume when the reset device comes back online.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Reset of Core Layer Device dca-core-1 passed.

Reset of Core Layer Device dca-core-2

The Core Layer is the first stop for traffic coming into the datacenter. As such, redundancy in the core layer is important for maintaining a high level of availability. Having redundant devices running OSPF allows for failover times nearly as fast as could be achieved through redundant supervisors running SSO with NSF. Further, redundant devices provides load balancing mechanisms.

This test verified the impact on traffic due to the failure of the core device dca-core-2. Background traffic was run using Linux servers as well as measurable traffic using the Shenick test tool. A total of five system resets were performed.

Test Procedure

The procedure used to perform the Reset of Core Layer Device dca-core-2 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Start the background traffic and the measurable Shenick traffic.

Step 3 Once traffic is running, verify that dca-core-2 is passing traffic through interfaces Te1/2 and Te1/4 using the show interfaces counters | include Port|Te1 command.

Step 4 Reload dca-core-2.

Step 5 When dca-core-2 comes back online, verify that it is again passing traffic through interfaces Te1/2 and Te1/4 using the show interfaces counters | include Port|Te1 command.

Step 6 Repeat the above sequence of reloading the DUT and checking counters four more times.

Step 7 Verify the traffic that was lost via the Shenick test tool and the Linux-generated tool.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect raffic loss of three seconds or less for each failover performed.
- We expect traffic forwarding to resume when the reset device comes back online.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Reset of Core Layer Device dca-core-2 passed.

Spanning-Tree Primary Root Failure and Recovery

In the DCAP test topology, dca-agg-1 is configured as the primary spanning-tree root switch. This means that all traffic flowing between clients and servers will find its way through dca-agg-1.

The spanning-tree protocol has certain rules governing the STP link states (Forwarding and Blocking) of root and non-root devices. This test verified that the interfaces in the Layer 2 domain of the DCAP test topology were in the proper STP state initially, following a primary root failure, and after the primary root recovery.

The purpose of this test is to look specifically at the functionality and behavior of the Rapid PVST+ spanning-tree protocol during the failure of the primary root switch. No traffic is run in this test for this reason. There are other tests in this suite that measure the impact of the failure of certain devices on traffic.

Test Procedure

The procedure used to perform the Spanning-Tree Primary Root Failure and Recovery test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 In spanning-tree, the bridge with the lowest configured priority becomes the STP root switch. If all priorities are the same (i.e. default), the bridge with the lowest MAC address becomes root. Use the show spanning-tree bridge command to verify that dca-agg-1 is configured with the lowest STP priority for all VLANs in the Layer 2 domain. Also verify that dca-agg-2 is configured with the second-lowest priority.

The configured STP priority of dca-agg-1 is 24576 for all VLANs. Because spanning-tree extend system-id is configured, the real priority of dca-agg-1 for a particular VLAN will be the sum of the configured priority plus the value of that VLAN.

The configured STP priority of dca-agg-1 is 28672 for all VLANs. The adjustment to real/advertised priority due to the spanning-tree extend system-id command applies to this switch as well. Note that since dca-agg-2 has the second-highest STP priority of all the switches in the L2 domain, it is next-in-line to become STP root should dca-agg-1 fail.

The STP priority of the other switches in the L2 domain was left as default, with the advertised priority being the default value plus the VLAN value.

Step 3 Use the show spanning-tree summary command on dca-agg-1 and dca-agg-2 to verify that all VLANs in each of these switches are in STP Forwarding state.

Step 4 Use the show spanning-tree summary command on dca-acc-6k-1, dca-acc-6k-2, and dca-voodoo-2 to verify that there is one interface in each VLAN on these switches that is in STP Blocking state.

Each of these three switches shows 202 VLANs. Each switch is dual-homed to the two Aggregation Layer switches, so that their Layer 2 design is a "triangle-shaped looped topology." As such, one uplink interface will be Forwarding for all VLANs (the one connected to the STP Root) and one uplink will be Blocking for all VLANs (the one connected to the STP Secondary Root). So, 202 interfaces are in Blocking state.

Step 5 Use the show spanning-tree summary command on dca-acc-4k-1 and dca-acc-4k-2 to verify the STP states of the VLANs in these switches.

Both of these switches show 202 VLANs. These two switches are connected to the Aggregation Layer switches and each other, such that their Layer 2 design is a "U-shaped looped topology." As such, all the interfaces on one of the two (dca-acc-4k-1) will be Forwarding for all VLANs and one switch (dca-acc-4k-2) will have one interface in Blocking state for all VLANs. So, 202 interfaces are in Blocking state.

Step 6 Issue the show spanning-tree vlan 2101 command on all 7 Layer 2 devices to verify their steady-state status. This VLAN will be looked at as a representative of the whole set when the failover and recovery is performed in later steps.

All 7 devices show that the MAC address of the Root switch is 0015.c719.bf80 (dca-agg-1).

- The device dca-agg-1 should show six interfaces in STP "FWD" state. Te9/4, Te10/1, Te10/2 and Te10/4 are trunks to the Access Layer devices. Po1 is the trunk connecting dca-agg-1 to dca-agg-2, used to carry the Fault Tolerant and synchronization information for the CSM and FWSM. Po270 is the internal EtherChannel interface by which the FWSM connects to the system backplane.

- The device dca-agg-2 should also show six interfaces in STP "FWD" state. Te9/4, Te10/1, Te10/2 and Te10/4 are trunks to the Access Layer devices. Po1 is the trunk connecting dca-agg-2 to dca-agg-1, used to carry the Fault Tolerant and synchronization information for the CSM and FWSM. Po270 is the internal EtherChannel interface by which the FWSM connects to the system backplane.
- The device dca-acc-6k-1 should show several interfaces in STP "FWD" state and one in "BLK" state. Te1/1 is the Root Port for this device, connecting it to the STP root (dca-agg-1), and is Forwarding. Te1/2 is the Alternate Port, connecting dca-acc-6k-1 to the secondary root (dca-agg-2), and is Blocking. The remainder of the ports are connected to servers, and are all in Forwarding state.
- The device dca-acc-6k-2 should show several interfaces in STP "FWD" state and one in "BLK" state. Te1/1 is the Root Port for this device, connecting it to the STP root (dca-agg-1), and is Forwarding. Te1/2 is the Alternate Port, connecting dca-acc-6k-2 to the secondary root (dca-agg-2), and is Blocking. The remainder of the ports are connected to servers, and are all in Forwarding state.
- The device dca-acc-4k-1 should show all interfaces in STP "FWD" state. Te1/49 is the Root Port for this device, connecting it to the STP root (dca-agg-1), and is Forwarding. Te1/50 connects dca-acc-4k-1 to the other Cat4k access switch (dca-acc-4k-2), and is also Forwarding. The remainder of the ports are connected to servers, and are all in Forwarding state.
- The device dca-acc-4k-2 should show only one interface in STP "FWD" state and one in STP "BLK" state. Te1/49 is the Root Port for this device, connecting it to the secondary STP root (dca-agg-2), and is Forwarding. Te1/50 connects dca-acc-4k-1 to the other Cat4k access switch (dca-acc-4k-1), and is Blocking. There are no server ports in VLAN 2101 on this Access switch.
- The device dca-voodoo-2 should show two interfaces in STP "FWD" state and one in "BLK" state. Te1/1 is the Root Port for this device, connecting it to the STP root (dca-agg-1), and is Forwarding. Te1/2 connects dca-voodoo-2 to the secondary root switch (dca-agg-2), and is Blocking. The only other interface (Gi6/1) in the list is a server port and is Forwarding.

Step 7 Issue the reload command on the primary STP root, dca-agg-1.

Step 8 About 5 seconds after reloading dca-agg-1, check the STP states for VLAN 2101 again, on the 6 remaining Access Layer switches using the show spanning-tree vlan 2101 command. Verify that there are no ports in the Blocking state anymore and that all devices show the MAC address of the Root switch as being 0015.c734.9d80 (dca-agg-2).

The spanning-tree protocol that is being used is Rapid PVST+, or rPVST+. It facilitates sub-second state change times. Normal spanning-tree (IEEE 802.1d) takes roughly 30 seconds to reconverge following a topology change.

Step 9 Use the show spanning-tree summary command on all Access Layer devices to verify that there are no VLANs in Blocking state.

Step 10 Wait for the original STP Root device, dca-agg-1, to come back online.

Step 11 Once dca-agg-1 is online again and operational, issue the show spanning-tree vlan 2101 command on all 7 Layer 2 devices to verify their status has returned to steady-state conditions.

All 7 devices again show that the MAC address of the Root switch is 0015.c719.bf80 (dca-agg-1).

- The device dca-agg-1 should again show six interfaces in STP "FWD" state. Te9/4, Te10/1, Te10/2 and Te10/4 are trunks to the Access Layer devices. Po1 is the trunk connecting dca-agg-1 to dca-agg-2, used to carry the Fault Tolerant and synchronization information for the CSM and FWSM. Po270 is the internal EtherChannel interface by which the FWSM connects to the system backplane.

- The device dca-agg-2 should again show six interfaces in STP "FWD" state. Te9/4, Te10/1, Te10/2 and Te10/4 are trunks to the Access Layer devices. Po1 is the trunk connecting dca-agg-2 to dca-agg-1, used to carry the Fault Tolerant and synchronization information for the CSM and FWSM. Po270 is the internal EtherChannel interface by which the FWSM connects to the system backplane.
- The device dca-acc-6k-1 should again show several interfaces in STP "FWD" state and one in "BLK" state. Te1/1 is the Root Port for this device, connecting it to the STP root (dca-agg-1), and is Forwarding. Te10/2 is the Alternate Port, connecting dca-acc-6k-1 to the secondary root (dca-agg-2), and is Blocking. The remainder of the ports are connected to servers, and are all in Forwarding state.
- The device dca-acc-6k-2 should again show several interfaces in STP "FWD" state and one in "BLK" state. Te1/1 is the Root Port for this device, connecting it to the STP root (dca-agg-1), and is Forwarding. Te10/2 is the Alternate Port, connecting dca-acc-6k-2 to the secondary root (dca-agg-2), and is Blocking. The remainder of the ports are connected to servers, and are all in Forwarding state.
- The device dca-acc-4k-1 should again show all interfaces in STP "FWD" state. Te1/49 is the Root Port for this device, connecting it to the STP root (dca-agg-1), and is Forwarding. Te1/50 connects dca-acc-4k-1 to the other Cat4k access switch (dca-acc-4k-2), and is also Forwarding. The remainder of the ports are connected to servers, and are all in Forwarding state.
- The device dca-acc-4k-2 should again show only one interface in STP "FWD" state and one in STP "BLK" state. Te1/49 is the Root Port for this device, connecting it to the secondary STP root (dca-agg-2), and is Forwarding. Te1/50 connects dca-acc-4k-1 to the other Cat4k access switch (dca-acc-4k-1), and is Blocking. There are no server ports in VLAN 2101 on this Access switch.
- The device dca-voodoo-2 should again show two interfaces in STP "FWD" state and one in "BLK" state. Te1/1 is the Root Port for this device, connecting it to the STP root (dca-agg-1), and is Forwarding. Te1/2 connects dca-voodoo-2 to the secondary root switch (dca-agg-2), and is Blocking. The only other interface (Gi6/1) in the list is a server port and is Forwarding.

Step 12 Use the show spanning-tree summary command on dca-agg-1 and dca-agg-2 to verify that all VLANs in each of these switches are in STP Forwarding state.

Step 13 Use the show spanning-tree summary command on dca-acc-6k-1, dca-acc-6k-2, and dca-voodoo-2 to verify that there is one interface in each VLAN on these switches that is in STP Blocking state.

Each of these three switches shows 202 VLANs. Each switch is dual-homed to the two Aggregation Layer switches, so that their Layer 2 design is a "triangle-shaped looped topology." As such, one uplink interface will be Forwarding for all VLANs (the one connected to the STP Root) and one uplink will be Blocking for all VLANs (the one connected to the STP Secondary Root). So, 202 interfaces are in Blocking state.

Step 14 Use the show spanning-tree summary command on dca-acc-4k-1 and dca-acc-4k-2 to verify the STP states of the VLANs in these switches.

Both of these switches show 202 VLANs. These two switches are connected to the Aggregation Layer switches and each other, such that their Layer 2 design is a "U-shaped looped topology." As such, all the interfaces on one of the two (dca-acc-4k-1) will be Forwarding for all VLANs and one switch (dca-acc-4k-2) will have one interface in Blocking state for all VLANs. So, 202 interfaces are in Blocking state.

Step 15 Repeat the above sequence four more times, gathering the information necessary to verify correct STP behavior each time.

Step 16 Stop background scripts to collect final status of network devices and analyze for error.

Step 17 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect spanning-tree to reconverge appropriately when the primary root switch is taken offline.
- We expect spanning-tree to reconverge appropriately when the primary root switch is recovered.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Spanning-Tree Primary Root Failure and Recovery passed.

HSRP Failover with Fast Timers

Hot Standby Router Protocol (HSRP) is used to provide gateway redundancy to hosts on a given network. The Aggregation Layer devices dca-agg-1 and dca-agg-2 are configured to share a virtual IP address (VIP) that serves as the gateway for the hosts on that network. The router with the highest priority becomes the Active HSRP router for that network, while the other becomes the Standby. The two communicate through and exchange of Hello packets. Should the Standby router stop receiving Hello packets from the Active for a period of time (called the Dead Time), it will assume the Active is no longer available and transition itself from Standby state to Active state. This provides a high-availability gateway to the hosts on the network.

In the DCAP test topology, there are 201 VLANs configured with HSRP. The device dca-agg-1 is the Active HSRP router, with a priority of 120 on each VLAN interface. The device dca-agg-2 is the Standby HSRP router, with a priority of 110 on each VLAN interface. Each VLAN interface is configured with a separate HSRP group. This allows for each interface to have a unique MAC address. These two devices multicast their status with Hello packets sent every 1 second. The configured Dead Timer for each VLAN is 3 seconds.

Should dca-agg-1 fail, dca-agg-2 will assume the Active HSRP router role after 3 seconds. If dca-agg-1 fails, dca-agg-2 will also assume the active role for other services, such as those provided by the CSM, FWSM, and SSLSM modules. Traffic will failover completely to dca-agg-2 and its services modules.

When dca-agg-1 comes back online, it is intended that it will resume the active role for these various services, including HSRP. To avoid a flood of management traffic at the point when dca-agg-1 becomes available again, the HSRP configuration on the VLANs specifies a wait period of 60 seconds. Even though dca-agg-2 is receiving Hello packets again from dca-agg-1, it will not give up the Active role until it receives the Coup message that dca-agg-1 sends about 60 seconds after it comes back online.

This test verified that the HSRP protocol functions as configured in the DCAP test topology. First, to prove the function on a small scale, one VLAN interface on dca-agg-1 was shut down. It was verified that dca-agg-2 took over the Active role approximately 3 seconds after the VLAN on dca-agg-1 was shut down. The VLAN interface on dca-agg-1 was then brought back online, and it was verified that a Coup message was not sent until about 60 seconds after dca-agg-1 begins sending Hellos again.

The second part of this test verified that HSRP with these "fast timers" functioned correctly on a large scale, but shutting down 200 VLAN interfaces at once on dca-agg-1. The state transitions were monitored on dca-agg-2 to verify that all VLANs transitioned after about 3 seconds. When the VLANs on dca-agg-1 were brought back online, it was verified that, 60 seconds later, dca-agg-1 became the Active HSRP router again and dca-agg-2 transitioned back to Standby, for all of the VLANs.

Note that traffic was not used in this test case since it looks more at the functionality of the protocol. Impacts on traffic were looked at during the Reset of Aggregation Layer Device dca-agg-1 test case.

Test Procedure

The procedure used to perform the HSRP Failover with Fast Timers test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify the HSRP configuration of the VLAN 1102 interface on dca-agg-1 using the show running-config interface Vlan1102 command.

VLAN 1102 is configured with a real IP address of 101.1.2.2/24 as well as a virtual IP address of 101.1.2.1/24. The standby group for this VLAN is 2, so the virtual MAC address for this gateway will be 0000.0c07.ac02. The priority is configured as 120 and the timers are configured as 1-second for the Hellos and 3-seconds for the Dead Timer. Preempt is configured, with a delay of 60 seconds.

Step 3 Verify the HSRP configuration of the VLAN 1102 interface on dca-agg-2 using the show running-config interface Vlan1102 command.

VLAN 1102 is configured with a real IP address of 101.1.2.3/24 as well as a virtual IP address of 101.1.2.1/24. The standby group for this VLAN is 2, so the virtual MAC address for this gateway will be 0000.0c07.ac02. The priority is configured as 110 and the timers are configured as 1-second for the Hellos and 3-seconds for the Dead Timer. Preempt is configured, with a delay of 60 seconds.

Step 4 Verify that dca-agg-1 is the Active HSRP router by issuing the show standby vlan 1102 command on both dca-agg-1 and dca-agg-2.

On dca-agg-1, the output will show "Local state is Active" as well as a virtual MAC address of 0000.0c07.ac02. The output will also show that the router with address 101.1.2.3 is the Standby HSRP router.

On dca-agg-2, the output will show "Local state is Standby" and no virtual MAC will be displayed. The output will also show that the router with address 101.1.2.2 is the Active HSRP router.

Step 5 On dca-agg-2, set the debug condition interface Vlan1102 so that only activity on this VLAN will be logged in the debugging that is about to take place.

Step 6 On dca-agg-2, turn on debugging for HSRP Hello and Coup packets using the debug standby packets hello and debug standby packets coup commands.

Step 7 While the debugs are active on dca-agg-2, shutdown the VLAN 1102 interface on dca-agg-1.

Step 8 Using the debugs, verify that, on dca-agg-2, VLAN 1102 moves from Standby to Active state in 2-3 seconds.

In steady-state operation, there is a 1 for 1 exchange of Hello PDUs. When VLAN 1102 is shut down on dca-agg-1, dca-agg-2 will stop receiving Hellos. There should be, therefore, a period of 2-3 unilateral Hellos shown on dca-agg-2 before the state transition occurs.

Step 9 Verify that dca-agg-2 is now the Active HSRP router by issuing the show standby vlan 1102 command on both dca-agg-1 and dca-agg-2.

On dca-agg-1, the output will show "Local state is Init (interface down)".

On dca-agg-2, the output will show "Local state is Active" and, now, a virtual MAC of 0000.0c07.ac02 will be displayed.

Step 10 Bring the VLAN 1102 interface back online on dca-agg-1 using the no shutdown command.

Step 11 Verify that dca-agg-2 starts receiving Hello packets from dca-agg-1 again and that, after about 60 seconds, a Coup message is received and a state transition occurs.

About a minute after dca-agg-2 begins receiving Hellos again from dca-agg-1, it will receive a Coup message. When this Coup message is received, dca-agg-2 will transition from Standby to Speak state.

Step 12 Verify that dca-agg-1 is again the Active HSRP router by issuing the show standby vlan 1102 command on both dca-agg-1 and dca-agg-2.

On dca-agg-1, the output will show "Local state is Active" as well as a virtual MAC address of 0000.0c07.ac02. The output will also show that the router with address 101.1.2.3 is the Standby HSRP router.

On dca-agg-2, the output will show "Local state is Standby" and no virtual MAC will be displayed. The output will also show that the router with address 101.1.2.2 is the Active HSRP router.

Step 13 On dca-agg-2, turn off all debugging and unset the debug condition using the undebug all and no debug condition all commands.

Step 14 Use the show standby brief command on dca-agg-1 and dca-agg-2 to verify that dca-agg-1 is the Active HSRP router for all VLANs.

On dca-agg-1, under the "State" column heading, each VLAN should show "Active". On dca-agg-2, each VLAN should show "Standby".

Step 15 Use the interface range and shutdown commands to shut down VLANs 1101-1300 on dca-agg-1.

Step 16 By watching the error log messages, verify visually that the VLANs transition to Init state on dca-agg-1 and to Active state on dca-agg-2 within the 3-second Dead Timer window.

Step 17 Verify that the 200 VLANs are now in "Init" state on dca-agg-1 and "Active" state on dca-agg-2, using the show standby brief command.

Step 18 Use the interface range and no shutdown commands to bring online VLANs 1101-1300 on dca-agg-1.

Step 19 By watching the error log messages, verify visually that the VLANs transition to Active state on dca-agg-1 and to Standby state on dca-agg-2 after about 60 seconds.

Step 20 Verify that the 200 VLANs are again in "Active" state on dca-agg-1 and "Standby" state on dca-agg-2, using the show standby brief command.

Step 21 Stop background scripts to collect final status of network devices and analyze for error.

Step 22 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect Standby HSRP router to become Active following a loss of 2-3 hello PDUs.
- We expect proper failover to occur on both a small scale and a large scale.
- We expect the HSRP router with **preempt** configured to resume the Active role after the configured wait time.
- We expect preemption to work correctly on both a small scale and a large scale.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

HSRP Failover with Fast Timers passed.

HSRP Recovery From System Failure

The device dca-agg-1 is, by configuration, the Active HSRP router for 201 VLANs in the DCAP test topology. It is also configured with HSRP preempt, and a preempt delay of 60 seconds. This means that should a failover occur, and dca-agg-2 becomes the Active router, when dca-agg-1 comes back online, it will restore Active state 60 seconds after it becomes available.

This test verified the behavior of HSRP during a system failure. The first part of the test looked at what occurs when dca-agg-1 is rebooted. By design, dca-agg-2 became the new Active HSRP router once dca-agg-1 went offline. This happened within 3 seconds (the configured Dead Timer) of dca-agg-1 leaving the topology.

The second part of this test verified that dca-agg-1, once it became available again, preempted for Active HSRP status after 60 seconds of participating in the Hello exchange with dca-agg-2.

No traffic was used during this test as it is focused on looking at the HSRP behavior aspect of a failover scenario. The test Reset of Aggregation Layer Device dca-agg-1 looks at the effects of a failover on traffic.

Test Procedure

The procedure used to perform the HSRP Recovery From System Failure test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify that dca-agg-1 is the Active HSRP router for VLANs 301 and 1101-1300 using the show standby brief command.

In the output, under the column headed "State", each VLAN should be listed as "Active".

Step 3 Verify that dca-agg-2 is the Standby HSRP router for VLANs 301 and 1101-1300 using the show standby brief command.

In the output, under the column headed "State", each VLAN should be listed as "Standby".

Step 4 Reload dca-agg-1.

Step 5 Verify that the VLANs in dca-agg-2 transitioned from Standby to Active state within 2-3 seconds following the reload using the show standby brief command.

Step 6 Wait for dca-agg-1 to come back online.

Step 7 When dca-agg-1 comes back online, verify that it preempts dca-agg-2 and resumes the Active HSRP role after about 60 seconds of being online. Use the show standby brief command on both dca-agg-1 and dca-agg-2 for this.

Device dca-agg-1 should show "Active" for all VLANs again, which dca-agg-2 should show "Standby".

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect HSRP to failover correctly when the Active HSRP router is taken out of service.

- We expect that, when the router is brought back online, it will resume its Active HSRP role after the configured 60-second delay.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

HSRP Recovery From System Failure passed with exception CSCsf23499.

Failure of EtherChannel Module on dca-agg-1

The Port-channel between the two Aggregation Layer devices, dca-agg-1 and dca-agg-2, is critical for health monitoring and connection synchronization connections between the service modules. This link, an 802.1q trunk, carries the VLANs that communicate the heartbeat messages between the CSMs and the FWSMs. Further, it replicates the connection states between the peer service modules so that downtime due to failover is minimized. With that in mind, the redundancy of this port-channel is essential.

In the DCAP test topology, this port-channel is composed of two 10-Gigabit Ethernet links, each link on a separate WS-X6704-10GE module. This test verified that if one of these modules were to reset, the impact to traffic would be minimal. Each of the two modules will be flapped multiple times. Client-to-server TCP traffic will be monitored to verify that there are no adverse effects.

Test Procedure

The procedure used to perform the Failure of EtherChannel Module on dca-agg-1 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Begin sending HTTP test traffic using the Shenick test tool.

Step 3 Verify that the Port-channel 1 interface on dca-agg-1 and dca-agg-2 consists of two bundled 10-Gigabit Ethernet interfaces, and that they are active, using the show etherchannel 1 summary command.

Two interfaces, Te9/3 and Te10/3, should be listed as ports belonging to Po1. They should each have a P-flag next to them (in parentheses), indicating that they are bundled in the port-channel. This applies to the output seen from each device.

Step 4 Use the show module command to verify that the WS-X6704-10GE modules in slots 9 and 10 are online.

Step 5 On dca-agg-1, reset the 10-Gigabit Ethernet module in slot 9 using the hw-module module 9 reset command.

Step 6 When module 9 comes back online, verify that the Port-channel 1 interface on dca-agg-1 and dca-agg-2 again consists of two bundled 10-Gigabit Ethernet interfaces, and that they are active, using the show etherchannel 1 summary command.

Two interfaces, Te9/3 and Te10/3, should be listed as ports belonging to Po1. They should each have a P-flag next to them (in parentheses), indicating that they are bundled in the port-channel. This applies to the output seen from each device.

Step 7 Repeat the reset of module 9 on dca-agg-1 nine more times for a total of ten flaps.

Step 8 Measure any traffic loss due to the module being reset.

Step 9 Verify that the Port-channel 1 interface on dca-agg-1 and dca-agg-2 still consists of two bundled 10-Gigabit Ethernet interfaces, and that they are active, using the show etherchannel 1 summary command.

Two interfaces, Te9/3 and Te10/3, should be listed as ports belonging to Po1. They should each have a P-flag next to them (in parentheses), indicating that they are bundled in the port-channel. This applies to the output seen from each device.

Step 10 Use the show module command to verify that the WS-X6704-10GE modules in slots 9 and 10 are online.

Step 11 On dca-agg-1, reset the 10-Gigabit Ethernet module in slot 10 using the hw-module module 10 reset command.

Step 12 When module 10 comes back online, verify that the Port-channel 1 interface on dca-agg-1 and dca-agg-2 again consists of two bundled 10-Gigabit Ethernet interfaces, and that they are active, using the show etherchannel 1 summary command.

Two interfaces, Te9/3 and Te10/3, should be listed as ports belonging to Po1. They should each have a P-flag next to them (in parentheses), indicating that they are bundled in the port-channel. This applies to the output seen from each device.

Step 13 Repeat the reset of module 10 on dca-agg-1 nine more times for a total of ten flaps.

Step 14 Measure any traffic loss due to the module being reset.

Step 15 Stop background scripts to collect final status of network devices and analyze for error.

Step 16 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the port-channel interface to maintain normal operation as a logical interface when one of the hardware modules is reloaded.
- We expect traffic loss due to the module reload to be minimal.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Failure of EtherChannel Module on dca-agg-1 passed with exception CSCek26222.

Failure of EtherChannel Module on dca-agg-2

The Port-channel between the two Aggregation Layer device, dca-agg-1 and dca-agg-2, is critical for the monitoring of health and the synchronization of connections between the service modules. This link, an 802.1q trunk, carries the VLANs that communicate the heartbeat messages between the CSMs and the FWSMs. Further, it replicates the connection states between the peer service modules so that downtime due to failover is minimized. With that in mind, the redundancy of this port-channel is essential.

In the DCAP test topology, this port-channel is composed of two 10-Gigabit Ethernet links, each link on a separate WS-X6704-10GE module. This test verified that if one of these modules were to reset, the impact to traffic would be minimal. Each of the two modules will be flapped multiple times. Client-to-server TCP traffic will be monitored to verify that there are no adverse effects.

Test Procedure

The procedure used to perform the Failure of EtherChannel Module on dca-agg-2 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Begin sending HTTP test traffic using the Shenick test tool.

Step 3 Verify that the Port-channel 1 interface on dca-agg-1 and dca-agg-2 consists of two bundled 10-Gigabit Ethernet interfaces, and that they are active, using the show etherchannel 1 summary command.

Two interfaces, Te9/3 and Te10/3, should be listed as ports belonging to Po1. They should each have a P-flag next to them (in parentheses), indicating that they are bundled in the port-channel. This applies to the output seen from each device.

Step 4 Use the show module command to verify that the WS-X6704-10GE modules in slots 9 and 10 are online.

Step 5 On dca-agg-2, reset the 10-Gigabit Ethernet module in slot 9 using the hw-module module 9 reset command.

Step 6 When module 9 comes back online, verify that the Port-channel 1 interface on dca-agg-1 and dca-agg-2 again consists of two bundled 10-Gigabit Ethernet interfaces, and that they are active, using the show etherchannel 1 summary command.

Two interfaces, Te9/3 and Te10/3, should be listed as ports belonging to Po1. They should each have a P-flag next to them (in parentheses), indicating that they are bundled in the port-channel. This applies to the output seen from each device.

Step 7 Repeat the reset of module 9 on dca-agg-2 nine more times for a total of ten flaps.

Step 8 Measure any traffic loss due to the module being reset.

Step 9 Verify that the Port-channel 1 interface on dca-agg-1 and dca-agg-2 still consists of two bundled 10-Gigabit Ethernet interfaces, and that they are active, using the show etherchannel 1 summary command.

Two interfaces, Te9/3 and Te10/3, should be listed as ports belonging to Po1. They should each have a P-flag next to them (in parentheses), indicating that they are bundled in the port-channel. This applies to the output seen from each device.

Step 10 Use the show module command to verify that the WS-X6704-10GE modules in slots 9 and 10 are online.

Step 11 On dca-agg-2, reset the 10-Gigabit Ethernet module in slot 10 using the hw-module module 10 reset command.

Step 12 When module 10 comes back online, verify that the Port-channel 1 interface on dca-agg-1 and dca-agg-2 again consists of two bundled 10-Gigabit Ethernet interfaces, and that they are active, using the show etherchannel 1 summary command.

Two interfaces, Te9/3 and Te10/3, should be listed as ports belonging to Po1. They should each have a P-flag next to them (in parentheses), indicating that they are bundled in the port-channel. This applies to the output seen from each device.

Step 13 Repeat the reset of module 10 on dca-agg-2 nine more times for a total of ten flaps.

Step 14 Measure any traffic loss due to the module being reset.

Step 15 Stop background scripts to collect final status of network devices and analyze for error.

Step 16 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the port-channel interface to maintain normal operation as a logical interface when one of the hardware modules is reloaded.
- We expect traffic loss due to the module reload to be minimal.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Failure of EtherChannel Module on dca-agg-2 passed with exception CSCek26222.

Link Failure

Link failure testing measures the impact of a link failure occurring in the data path. The ability of the data center infrastructure to respond favorably to such scenarios is critical to the high availability of network resources.

The following tests were performed:

- Failure of Single Bundled 10-Gigabit Ethernet Link Between dca-agg-1 and dca-agg-2, page 2-61
- Failure of 10 Gigabit Ethernet Link Between dca-agg-1 and dca-acc-6k-2, page 2-63
- Failure of 10 Gigabit Ethernet Link Between dca-agg-2 and dca-acc-6k-2, page 2-63
- Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-agg-1, page 2-64
- Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-agg-2, page 2-65
- Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-core-2, page 2-66
- Failure of 10-Gigabit Ethernet Link Between dca-core-2 and dca-agg-1, page 2-66
- Failure of 10-Gigabit Ethernet Link Between dca-core-2 and dca-agg-2, page 2-67
- Network Resiliency Test, page 2-68

Failure of Single Bundled 10-Gigabit Ethernet Link Between dca-agg-1 and dca-agg-2

The Port-channel between the two Aggregation Layer devices, dca-agg-1 and dca-agg-2, is critical for the monitoring of health and the synchronization of connections between the service modules. This link, an 802.1q trunk, carries the VLANs that communicate the heartbeat messages between the CSMs and the FWSMs. Further, it replicates the connection states between the peer service modules so that downtime due to failover is minimized. With that in mind, the redundancy of this port-channel is essential.

This test verified that if a single link of that port-channel were to go down, the impact to traffic would be minimal. Each of the two 10-Gigabit Ethernet links in the port-channel will be flapped multiple times. Client-to-server TCP traffic will be monitored to verify that there are no adverse effects.

Test Procedure

The procedure used to perform the Failure of Single Bundled 10-Gigabit Ethernet Link Between dca-agg-1 and dca-agg-2 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Begin sending HTTP test traffic using the Shenick test tool.

Step 3 Verify that the Port-channel 1 interface on dca-agg-1 and dca-agg-2 consists of two bundled 10-Gigabit Ethernet interfaces, and that they are active, using the show etherchannel 1 summary command.

Two interfaces, Te9/3 and Te10/3, should be listed as ports belonging to Po1. They should each have a P-flag next to them (in parentheses), indicating that they are bundled in the port-channel. This applies to the output seen from each device.

Step 4 On dca-agg-1, shutdown interface Te9/3.

Step 5 After a minute, bring interface Te9/3 back online using the no shutdown command on dca-agg-1.

Step 6 Verify that the Port-channel 1 interface on dca-agg-1 and dca-agg-2 again consists of two bundled 10-Gigabit Ethernet interfaces, and that they are active, using the show etherchannel 1 summary command.

Two interfaces, Te9/3 and Te10/3, should be listed as ports belonging to Po1. They should each have a P-flag next to them (in parentheses), indicating that they are bundled in the port-channel. This applies to the output seen from each device.

Step 7 Repeat the flap of interface Te9/3 on dca-agg-1 nine more times for a total of ten flaps.

Step 8 Measure any traffic loss due to the interface being shut down and being brought back online.

Step 9 Verify that the Port-channel 1 interface on dca-agg-1 and dca-agg-2 still consists of two bundled 10-Gigabit Ethernet interfaces, and that they are active, using the show etherchannel 1 summary command.

Two interfaces, Te9/3 and Te10/3, should be listed as ports belonging to Po1. They should each have a P-flag next to them (in parentheses), indicating that they are bundled in the port-channel. This applies to the output seen from each device.

Step 10 On dca-agg-1, shutdown interface Te10/3.

Step 11 After a minute, bring interface Te10/3 back online using the no shutdown command on dca-agg-1.

Step 12 Verify that the Port-channel 1 interface on dca-agg-1 and dca-agg-2 again consists of two bundled 10-Gigabit Ethernet interfaces, and that they are active, using the show etherchannel 1 summary command.

Two interfaces, Te9/3 and Te10/3, should be listed as ports belonging to Po1. They should each have a P-flag next to them (in parentheses), indicating that they are bundled in the port-channel. This applies to the output seen from each device.

Step 13 Repeat the flap of interface Te10/3 on dca-agg-1 nine more times for a total of ten flaps.

Step 14 Measure any traffic loss due to the interface being shut down and being brought back online.

Step 15 Stop background scripts to collect final status of network devices and analyze for error.

Step 16 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the port-channel interface to maintain normal operation as a logical interface when a single bundled link is flapped.
- We expect traffic loss due to the link failure to be minimal.
- We expect traffic loss due to the link recovery to be minimal.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Failure of Single Bundled 10-Gigabit Ethernet Link Between dca-agg-1 and dca-agg-2 passed.

Failure of 10 Gigabit Ethernet Link Between dca-agg-1 and dca-acc-6k-2

This test verified the impact on traffic from a failure of the 10-Gigabit Ethernet link between an Aggregation Layer device, dca-agg-1, and an Access Layer device, dca-acc-6k-2. Web traffic is sent using the Shenick test tool from 100 clients to a VIP on the CSM. Te10/1 on dca-agg-1 is shut down for 2 minutes then brought back online for 2 minutes. This is repeated for a total of 10 interface flaps.

Test Procedure

The procedure used to perform the Failure of 10 Gigabit Ethernet Link Between dca-agg-1 and dca-acc-6k-2 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Begin sending HTTP test traffic using the Shenick test tool.

Step 3 Log in to dca-agg-1 and shutdown interface Te10/1.

Step 4 After two minutes, bring interface Te10/1 back online using the no shutdown command on dca-agg-1.

Step 5 Repeat the interface flap nine more times for a total of ten flaps.

Step 6 Measure any traffic loss due to the interface being shut down and being brought back online.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect traffic loss due to the link failure to be minimal.
- We expect traffic loss due to the link recovery to be minimal.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Failure of 10 Gigabit Ethernet Link Between dca-agg-1 and dca-acc-6k-2 passed.

Failure of 10 Gigabit Ethernet Link Between dca-agg-2 and dca-acc-6k-2

This test verified the impact on traffic from a failure of the 10-Gigabit Ethernet link between an Aggregation Layer device, dca-agg-2, and an Access Layer device, dca-acc-6k-2. Web traffic is sent using the Shenick test tool from 100 clients to a VIP on the CSM. Te10/1 on dca-agg-2 is shut down for 2 minutes then brought back online for 2 minutes. This is repeated for a total of 10 interface flaps.

Test Procedure

The procedure used to perform the Failure of 10 Gigabit Ethernet Link Between dca-agg-2 and dca-acc-6k-2 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Begin sending HTTP test traffic using the Shenick test tool.

Step 3 Log in to dca-agg-2 and shutdown interface Te10/1.

Step 4 After a minute, bring interface Te10/1 back online using the no shutdown command on dca-agg-2.

Step 5 Repeat the interface flap nine more times for a total of ten flaps.

Step 6 Measure any traffic loss due to the interface being shut down and being brought back online.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect traffic loss due to the link failure to be minimal.
- We expect traffic loss due to the link recovery to be minimal.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Failure of 10 Gigabit Ethernet Link Between dca-agg-2 and dca-acc-6k-2 passed.

Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-agg-1

This test verified the impact on traffic from a failure of the 10-Gigabit Ethernet link between an Aggregation Layer device, dca-agg-1, and a Core Layer device, dca-core-1. Web traffic is sent using the Shenick test tool from 100 clients to a VIP on the CSM. Te1/2 on dca-core-1 is shut down for 2 minutes then brought back online for 2 minutes. This is repeated for a total of 10 interface flaps.

Test Procedure

The procedure used to perform the Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-agg-1 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Begin sending HTTP test traffic using the Shenick test tool.

Step 3 Log in to dca-core-1 and shutdown interface Te1/2.

Step 4 After a minute, bring interface Te1/2 back online using the no shutdown command on dca-core-1.

Step 5 Repeat the interface flap nine more times for a total of ten flaps.

Step 6 Measure any traffic loss due to the interface being shut down and being brought back online.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect traffic loss due to the link failure to be minimal.
- We expect traffic loss due to the link recovery to be minimal.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-agg-1 passed.

Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-agg-2

This test verified the impact on traffic from a failure of the 10-Gigabit Ethernet link between an Aggregation Layer device, dca-agg-2, and a Core Layer device, dca-core-1. Web traffic is sent using the Shenick test tool from 100 clients to a VIP on the CSM. Te1/3 on dca-core-1 is shut down for 2 minutes then brought back online for 2 minutes. This is repeated for a total of 10 interface flaps.

Test Procedure

The procedure used to perform the Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-agg-2 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Begin sending HTTP test traffic using the Shenick test tool.

Step 3 Log in to dca-core-1 and shutdown interface Te1/3.

Step 4 After a minute, bring interface Te1/3 back online using the no shutdown command on dca-core-1.

Step 5 Repeat the interface flap nine more times for a total of ten flaps.

Step 6 Measure any traffic loss due to the interface being shut down and being brought back online.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect traffic loss due to the link failure to be minimal.
- We expect traffic loss due to the link recovery to be minimal.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-agg-2 passed.

Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-core-2

This test verified the impact on traffic from a failure of the 10-Gigabit Ethernet link between the two Core Layer devices, dca-core-1 and dca-core-2. Web traffic is sent using the Shenick test tool from 100 clients to a VIP on the CSM. Te1/1 on dca-core-1 is shut down for 2 minutes then brought back online for 2 minutes. This is repeated for a total of 10 interface flaps.

Test Procedure

The procedure used to perform the Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-core-2 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Begin sending HTTP test traffic using the Shenick test tool.

Step 3 Log in to dca-core-1 and shutdown interface Te1/1.

Step 4 After a minute, bring interface Te1/1 back online using the no shutdown command on dca-core-1.

Step 5 Repeat the interface flap nine more times for a total of ten flaps.

Step 6 Measure any traffic loss due to the interface being shut down and being brought back online.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect traffic loss due to the link failure to be minimal.
- We expect traffic loss due to the link recovery to be minimal.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Failure of 10-Gigabit Ethernet Link Between dca-core-1 and dca-core-2 passed.

Failure of 10-Gigabit Ethernet Link Between dca-core-2 and dca-agg-1

This test verified the impact on traffic from a failure of the 10-Gigabit Ethernet link between an Aggregation Layer device, dca-agg-1, and a Core Layer device, dca-core-2. Web traffic is sent using the Shenick test tool from 100 clients to a VIP on the CSM. Te1/2 on dca-core-2 is shut down for 2 minutes then brought back online for 2 minutes. This is repeated for a total of 10 interface flaps.

Test Procedure

The procedure used to perform the Failure of 10-Gigabit Ethernet Link Between dca-core-2 and dca-agg-1 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Begin sending HTTP test traffic using the Shenick test tool.

Step 3 Log in to dca-core-2 and shutdown interface Te1/2.

Step 4 After a minute, bring interface Te1/2 back online using the no shutdown command on dca-core-2.

Step 5 Repeat the interface flap nine more times for a total of ten flaps.

Step 6 Measure any traffic loss due to the interface being shut down and being brought back online.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect traffic loss due to the link failure to be minimal.
- We expect traffic loss due to the link recovery to be minimal.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Failure of 10-Gigabit Ethernet Link Between dca-core-2 and dca-agg-1 passed.

Failure of 10-Gigabit Ethernet Link Between dca-core-2 and dca-agg-2

This test verified the impact on traffic from a failure of the 10-Gigabit Ethernet link between an Aggregation Layer device, dca-agg-2, and a Core Layer device, dca-core-2. Web traffic is sent using the Shenick test tool from 100 clients to a VIP on the CSM. Te1/3 on dca-core-2 is shut down for 2 minutes then brought back online for 2 minutes. This is repeated for a total of 10 interface flaps.

Test Procedure

The procedure used to perform the Failure of 10-Gigabit Ethernet Link Between dca-core-2 and dca-agg-2 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Begin sending HTTP test traffic using the Shenick test tool.

Step 3 Log in to dca-core-2 and shutdown interface Te1/3.

Step 4 After a minute, bring interface Te1/3 back online using the no shutdown command on dca-core-2.

Step 5 Repeat the interface flap nine more times for a total of ten flaps.

Step 6 Measure any traffic loss due to the interface being shut down and being brought back online.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect traffic loss due to the link failure to be minimal.
- We expect traffic loss due to the link recovery to be minimal.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Failure of 10-Gigabit Ethernet Link Between dca-core-2 and dca-agg-2 passed.

Network Resiliency Test

This is a single test that encompasses a suite of scripts used to take an initial snapshot of the test network, introduce mayhem, and take a final snapshot of the test network. The initial and final snapshots should be nearly identical, save for some cosmetic differences.

The class of test scripts used are called spiders, or crawlers. Each script is given a seed device to start at. It takes a snapshot of the information for that crawler, then discovers the neighboring devices, and moves on to those neighboring devices to take snapshots of them, and so forth until all the devices in the network have been covered. There are nine individual crawler scripts that are run at the beginning of this test. They look at the module status in the devices, the interface status, the trunk and channel status, as well as the status of certain protocols (CDP, UDLD, PIM, HSRP, and OSPF). Information is gathered from the device, and saved in a file, during this initial run.

After the initial snapshot is taken, a script called Rolling Havoc is run. This crawler logs into each network device flaps each inter-switch link a configured number of times.

After Rolling Havoc has wrought its harm on the network, the nine other crawler scripts are run again, gathering post-havoc information about the same aspects of the network. Being that no other negative tests are taking place during this test sequence, the network should return to an identical state to that it was in before the Rolling Havoc script was run.

Test Procedure

The procedure used to perform the Network Resiliency Test test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Baseline all CDP neighbor relationships. Run the CDP crawler script verifying all expected CDP neighbors are reported.

The purposed of the CDP crawler script is to crawl through the network continuously, noting any changes that occur between traversals in CDP information. It parses information gathered from select CDP and IOS commands.

Step 3 Baseline all EtherChannel members. Run the channel crawler script verifying that all interfaces expected to be in channels are reported.

The purpose of the channel crawler script is to run through a network and verify that EtherChannels are in a proper state. It parses information gathered from select EtherChannel and IOS commands.

Step 4 Baseline all trunk interfaces. Run the trunk crawler script verifying that all expected trunking interfaces, configuration, and status are reported.

The purpose of the trunk crawler script is to run through a network and verify that trunking is in a proper state. It parses information gathered from select trunking and IOS commands.

Step 5 Baseline all interface states and counters. Run the interface crawler script recording interface counters and states.

The interface crawler script crawls through a network continually. All up/up interfaces are checked for various errors. Initially all non-zero error counters will be logged, then any counters that increment from that point on.

Step 6 Baseline all interface UDLD states. Run the UDLD crawler script recording the UDLD state of all interfaces.

The UDLD crawler script gathers a list of UDLD ports from a list of devices and traverses their neighbors continuously checking for UDLD problems or inconsistencies. It parses information gathered from select UDLD and IOS commands.

Step 7 Baseline all linecards used in the topology. Run the module crawler script recording module counters and state.

The module crawler script gathers a list of MODULES from a list of devices and looks for problems or inconsistencies. It parses information gathered from select module and IOS commands.

Step 8 Baseline the HSRP feature in the topology. Run the HSRP crawler script recording HSRP state.

Step 9 Flap each of the active non-management interfaces in the SH3 network 5 times each.

Step 10 Execute the CDP crawler script to verify that the CDP feature is still operating correctly in the Data Center test network.

Step 11 Execute the channel crawler script to verify that the EtherChannel feature is still operating correctly in the Data Center test network.

Step 12 Execute the trunk crawler script to verify that the trunking feature is still operating correctly in the Data Center test network.

Step 13 Execute the interface crawler script to verify that the basic functionality of the interface is still operating correctly in the Data Center test network.

Step 14 Execute the UDLD crawler script to verify that the UDLD feature is still operating correctly in the Data Center test network.

Step 15 Execute the module crawler script to verify that the line cards in the Data Center test network are still operating correctly.

Step 16 Execute the HSRP crawler script to verify that HSRP in the Data Center test network is still operating correctly.

Step 17 Stop background scripts to collect final status of network devices and analyze for error.

Step 18 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the EtherChannel feature to work correctly before and after the interface flapping.
- We expect the HSRP feature to work correctly before and after the interface flapping.
- We expect the CDP feature to work correctly before and after the interface flapping.
- We expect the trunking feature to work correctly before and after the interface flapping.
- We expect the PIM feature to work correctly before and after the interface flapping.
- We expect the UDLD feature to work correctly before and after the interface flapping.
- We expect the modules to work correctly before and after the interface flapping.

- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Network Resiliency Test passed.

Layer 4-7 Services

The modular Catalyst 6500 switching platform supports various modules which provide services at Layers 4-7. Several of these Service Modules are bundled together and tested in the DCAP 2.0 topology, including the Content Switching Module (CSM), Firewall Services Module (FWSM) and Secure Socket Layer Module (SSLM). The tests in this chapter focus on the ability of these three Service Modules to work together to provide load-balancing, security and encryption services to data center traffic.

Services at Layers 4-7 were handled by Service Modules installed in the Catalyst 6500 platform at the Aggregation Layer. The Content Switching Module (CSM) was installed to provide server load-balancing services. The Firewall Services Module (FWSM) was used to provide basic firewall security services. The Secure Socket Layer Module (SSLM) was used to provide encryption services, both client-facing, and in the backend, as well.

The modular design of the Catalyst 6500 platform allows for each of these Service Modules to be installed in the same chassis. Each of these Service Modules has been subjected to, and passed, standalone testing performed by the Safe Harbor test team. To avoid a duplication of efforts, the Service Module testing performed in DCAP 2.0 was focused on the interoperability of all three Service Modules.

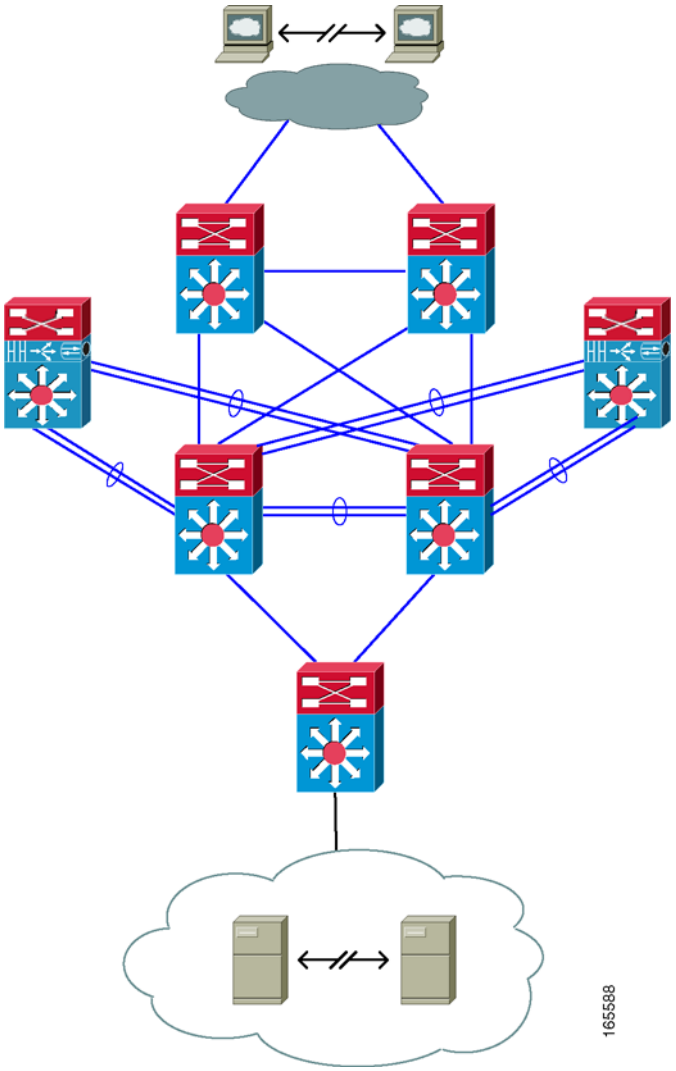
Integrated Bundle Vs. Service Switch Models

In the first phase of DCAP testing, DCAP 1.0, the Service Module bundle was contained directly in the Aggregation Layer switches, dca-agg-1 and dca-agg-2, themselves. DCAP 2.0 testing repeats the testing of this integrated bundle configuration and adds a new configuration. The integrated bundle testing is performed in the data center DC-A LAN, while the new configuration is implemented and tested in the DC-B LAN.

The new configuration is referred to as the Service Switch configuration. In the integrated bundle configuration, the Aggregation Layer devices perform double duty, providing Layer 4-7 services to data center traffic and providing aggregation density to the Access Layer switches. In the Service Switch configuration, these two functions are cleaved.

A dedicated Catalyst 6500, called the Service Switch, provides Layer 4-7 services to data center traffic, as seen in Figure 3-1. This switch is typically a Catalyst 6509 with one slot used for the Supervisor 720 and a second slot used for a 10-Gigabit Ethernet module (WS-X6704-10GE) to provide connectivity to the Aggregation Layer switches. This leaves a full seven slots available for installing Catalyst 6500 Service Modules. While only the CSM, FWSM, and SSLM were tested in DCAP 2.0, many other Service Modules exist, such as the Intrusion Detection Services Module (IDSM), Network Analysis Module (NAM), and Communications Media Module (CMM), to name a few.

Figure 3-1 Service Switch Overview Topology in DC-B



In the integrated bundle model, as configured in the DC-A LAN topology, a Catalyst 6513 chassis was used as the Aggregation Layer switch, so that a full 5 slots could be allocated for 10-Gigabit Ethernet linecards. (If a Catalyst 6509 had been used, 5 slots would have been used for the four Service Modules and Supervisor, leaving only 4 slots for 10-Gigabit Ethernet modules.)

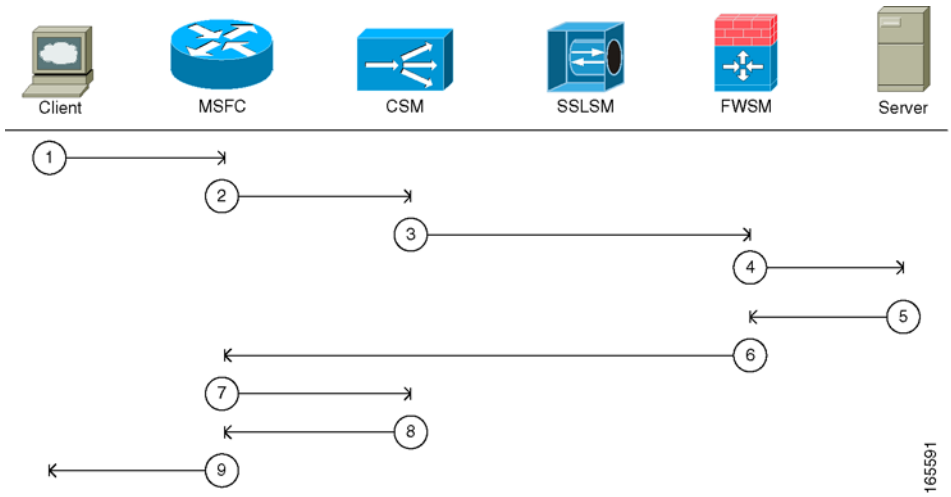
With the Service Modules in their own chassis in the Service Switch deployment, the Aggregation Layer switch is freed up to provide additional 10-Gigabit Ethernet density to the Access Layer. The Catalyst 6513 still only provides 5 slots of 10-Gigabit Ethernet density, though, since only slots 9-13 are dual-fabric capable, a requirement for the 10-Gigabit Ethernet modules. Therefore, a Catalyst 6509 was used as the Aggregation Layer devices, dcb-agg-1 and dcb-agg-2, providing a full 8 slots for 10-Gigabit Ethernet density.

Traffic Pathways Through the Bundle

While later sections of this chapter will discuss the specifics of each Service Module bundle deployment, the way that they handle data center traffic and provide services is the same.

The operation of the CSM for the majority of the traffic it passes is fairly straightforward as seen in Figure 3-2. A client request is sent to the advertised VIP (1) and reaches that destination after being routed by the MSFC (2). The CSM sends the request on to a real server in the serverfarm that is matched with the advertised VIP (3). The request hits the FWSM and is bridged from the outside VLAN to the inside VLAN and forwarded along to the real server (4), which services the request. The server sends the reply to its default gateway on the MSFC (5). After being bridged again by the FWSM (6), this reply hits the policy map configured on the MSFC which tells it to send the reply along to the CSM (7). The CSM replaces the real server's IP address with that of the VIP and sends the reply on to the client (8), again via the MSFC (9).

Figure 3-2 HTTP Request Packet Path Through Service Module Bundle



The DCAP test topology is set up to handle clear text backend and SSL backend. The first case is described above. Figure 3-3 clearly shows an encrypted client request sent to the advertised VIP (1) which reaches that destination after being routed by the MSFC (2). The CSM sends the request on to one of the operational SSLSMs in the SSLSM serverfarm for decryption (3). The decrypted traffic is sent back to the CSM (4) where it hits another VIP and is thus sent on, in clear text, to the selected real server (5). The request hits the FWSM and is bridged from the outside VLAN to the inside VLAN and forwarded along to the real server (6), which services the request. The server sends the reply to its default gateway on the MSFC (7). After being bridged again by the FWSM (8), this reply hits the policy map configured on the MSFC which tells it to send the reply along to the CSM (9). The CSM works with the SSLSM in reverse order, sending the reply first to the SSLSM for encryption (10). The SSLSM encrypts the reply, using the same client keypair that it used to decrypt it earlier, and sends it back to the CSM (11). The CSM replaces the real server's IP address with that of the VIP and sends the encrypted reply on to the client (12), again via the MSFC (13).

Figure 3-3 HTTPS Request Packet Path Through Service Module Bundle with No Backend Encryption

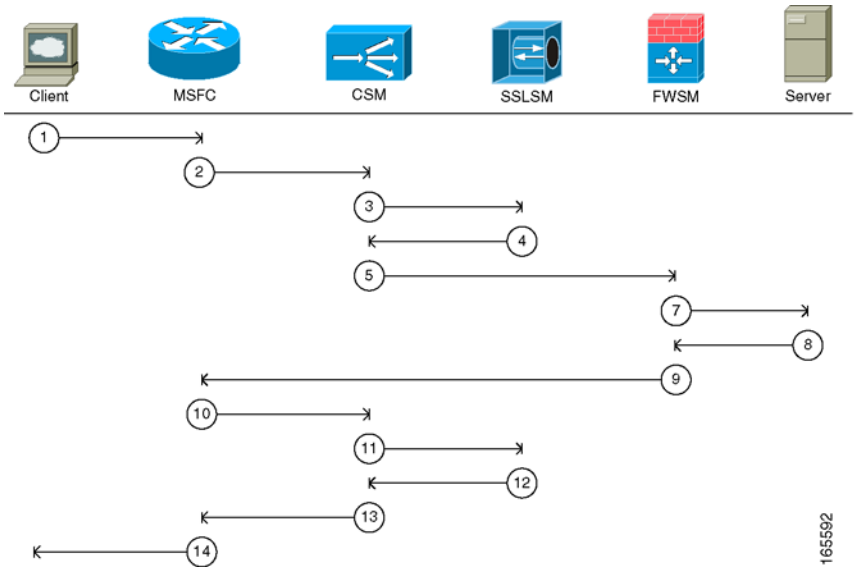
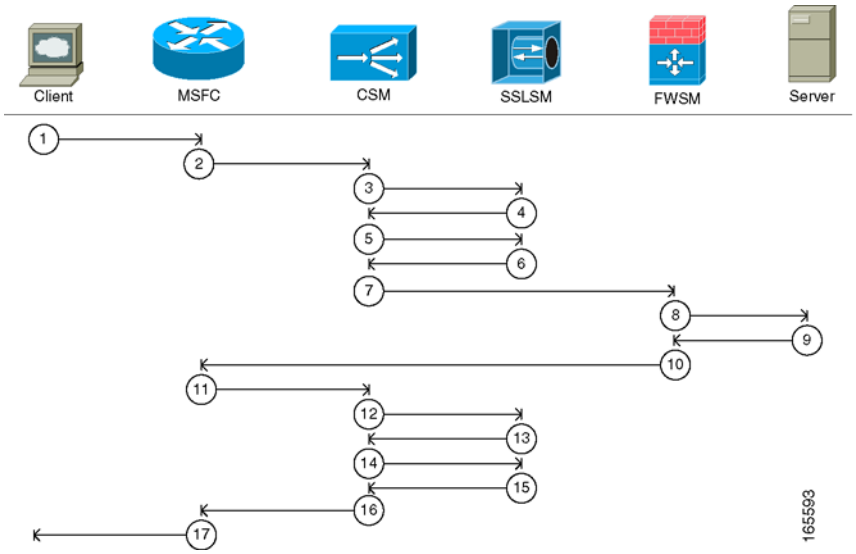


Figure 3-4 show extra steps for traffic that requires backend encryption services. An encrypted client request is sent to the advertised VIP (1) and reaches that destination after being routed by the MSFC (2). The CSM sends the request on to one of the operational SSLSMs in the SSLSM serverfarm for decryption (3). The decrypted traffic is sent back to the CSM (4) where it hits another VIP that has it sent back to the SSLSM for backend encryption (5). The SSLSM re-encrypts the request, this time with the server keypair, and sends it back to the CSM (6). The CSM sends the request along to the appropriate real server, encrypted (7). This encrypted request hits the FWSM and is bridged from the outside VLAN to the inside VLAN and forwarded along to the real server (8), which services the request. The server sends the reply to its default gateway on the MSFC (9). After being bridged again by the FWSM (10), this reply hits the policy map configured on the MSFC which tells it to send the reply along to the CSM (11). The CSM works with the SSLSM in reverse order, sending the reply first to the SSLSM for decryption (12). The SSLSM decrypts the reply using the server keypair and sends the reply, in clear text, back to the CSM (13). The reply hits another VIP on the CSM which sends it back to the SSLSM (14), one final time, for re-encryption using the client keypair. Once re-encrypted, the reply is sent back to the CSM (15). The CSM replaces the real server's IP address with that of the VIP and sends the encrypted reply on to the client (16), again via the MSFC (17).

Figure 3-4 HTTPS Request Packet Path Through Service Module Bundle with Backend Encryption

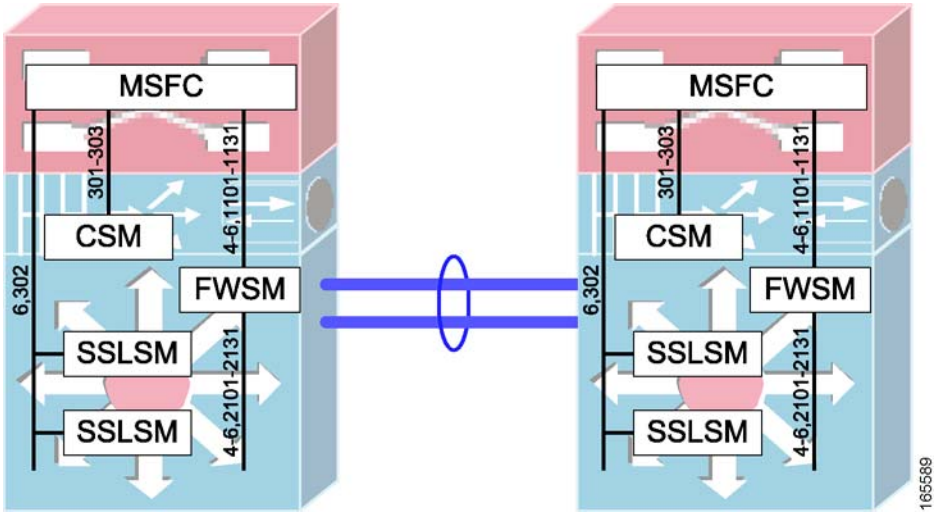


Integrated Bundle Configuration

Though the Service Switch configuration is physically different than the integrated bundle configuration, logically, they provide the Layer 4-7 services to the data center traffic in the same way. The internal workings of the integrated bundle will be explained here first, then the differences in the Service Switch will be outlined.

Each of the aggregation devices in the integrated bundle contains an equal number of Service Modules: one CSM, one FWSM and two SSLMs, as illustrated in Figure 3-5. There needs to be communication between the Service Modules not only in the same chassis, but also between chassis, as will be explained below. The 10-Gigabit Etherchannel provides the Layer 2 adjacency needed to facilitate this communication.

Figure 3-5 Logical Functionality of Services in the Integrated Bundle Configuration



There are several modes that the CSM can be configured to operate in, including bridge and routed modes. In the DCAP test topology, it is configured to operate in one-arm mode, which is a variation of routed mode. Essentially, the CSM can be viewed as an appliance connected via a single link to the Aggregation device. Being in routed mode, it serves as a gateway for both client and server traffic.

The CSM is connected, internally, via an etherchannel consisting of four GigabitEthernet interfaces. This etherchannel is a trunk carrying VLANs 301-303. VLAN 301 is used for traffic that needs to be load-balanced without the use of the SSL services provided by the SSLSM blades. VLAN 302, which runs on the link connecting the SSLSMs to the MSFC, is used for traffic that needs to be load-balanced and also requires the encryption or decryption services of the SSLSM blades. VLAN 303 is used to communicate with the peer CSM in the other Aggregation Layer device via the heartbeat messages.

As discussed earlier, there are two CSMs in the DCAP DC-A LAN test topology, one in dca-agg-1 and one in dca-agg-2. With CSM version 4.1(6), only one CSM can effectively be active at a time. The CSM in dca-agg-1 is configured with a priority such that it is the active CSM, when both CSMs are able to communicate with each other. In steady-state, each CSM sends heartbeat messages to its peer every second. If 3 seconds pass between subsequent heartbeat messages, a CSM will consider its peer to be unavailable. If the active CSM stops hearing from the standby, nothing will happen other than learning that the standby is unavailable. If the standby stops hearing from the active, though, it will transition itself to active state and begin to advertise its services and gateways to the network. As mentioned previously, an active/active condition can wreak havoc on a network when both CSMs begin to advertise the same service. This is why the etherchannel between the two Aggregation Layer devices is so critical.

There are useful statistics listed below regarding the configuration of the CSM services in the Cisco DCAP 2.0 test topology. For redundancy to be operational, each CSM must have exactly the same configuration, with regards to the services that it provides.

- 2000 real servers are configured and operational
- 31 server farms group the real servers (30 server farms with 64 reals, 1 with 80)
- 31 vservers providing one VIP/gateway apiece
- SSLSM serverfarms and vservers facilitating the correct handling of HTTPS traffic
- Various flavors of server load-balancing algorithms (round robin, least connections, most connections, weighted server, etc.)
- HTTP, Passive FTP and Active FTP

- TCP probes used in each serverfarm to monitor the health of each of the 2000 real servers in the Layer 2 domain

There are two modes that the FWSM can operate in, Routed and Transparent. There is also the option of configuring more than one operational context. Different contexts provide virtual firewalls to different traffic. The FWSMs in the DCAP test topology are configured to operate in transparent mode using 31 separate contexts (in addition to the default "system" and "admin" contexts).

In transparent mode, the firewall is actually bridging traffic from an outside VLAN to an inside VLAN, and vice versa. In the DCAP test topology, the outside VLANs used for data are VLANs 1101-1131. The corresponding inside VLANs are 2101-2131. Client traffic whose destination is a real server on VLAN 2101 on the inside, will hit the firewall from the outside on VLAN 1101 and be bridged onto 2101. The opposite will take place in the other direction.

Like the CSM, the FWSM also uses heartbeat messages to communicate with its peer, verifying its existence. Also like the CSM, only one FWSM can be active at a time, with FWSM software version 2.3(3.2). The heartbeat messages are sent on VLAN 4 each second. If 3 seconds pass between subsequent heartbeat messages, a FWSM will consider its peer to be unavailable. As with the CSM, if the active FWSM stops hearing from the standby, nothing will happen other than it will learn that the standby is unavailable. If the standby stops hearing from the active, though, it will transition itself to active state and begin to advertise its services and gateways to the network. An active/active condition is a dangerous possibility with the FWSM too.

VLAN 5 is used to communicate configuration information between the two peer FWSMs. Outside of certain elements of the "admin" and "system" contexts, there is only one configuration shared between the two FWSMs. The active will use the connection with the standby on VLAN 5 to synchronize the configuration between the two devices (with the active overwriting the standby whenever there are differences). VLAN 5 is also used by the active FWSM to replicate the current connections to the standby. In the event that a failover does occur, the standby will not lose time re-learning all of the connections that the active had established. Note that this is only good for long-term connections.

There are 31 contexts in the FWSM configuration, one for each of the VLANs. They are named "VLAN 1101-2101" through "VLAN 1131-2131" to reflect the VLANs they are associated with (outside-inside). These contexts govern the traffic that is allowed to access the inside from the outside, and vice versa. In Cisco DCAP 2.0 test topology, each context is essentially the same, save some minor differences. All contexts are very much promiscuous with regards to what traffic they let through.

VLAN 6 is used for management (telnet) access to the FWSM.

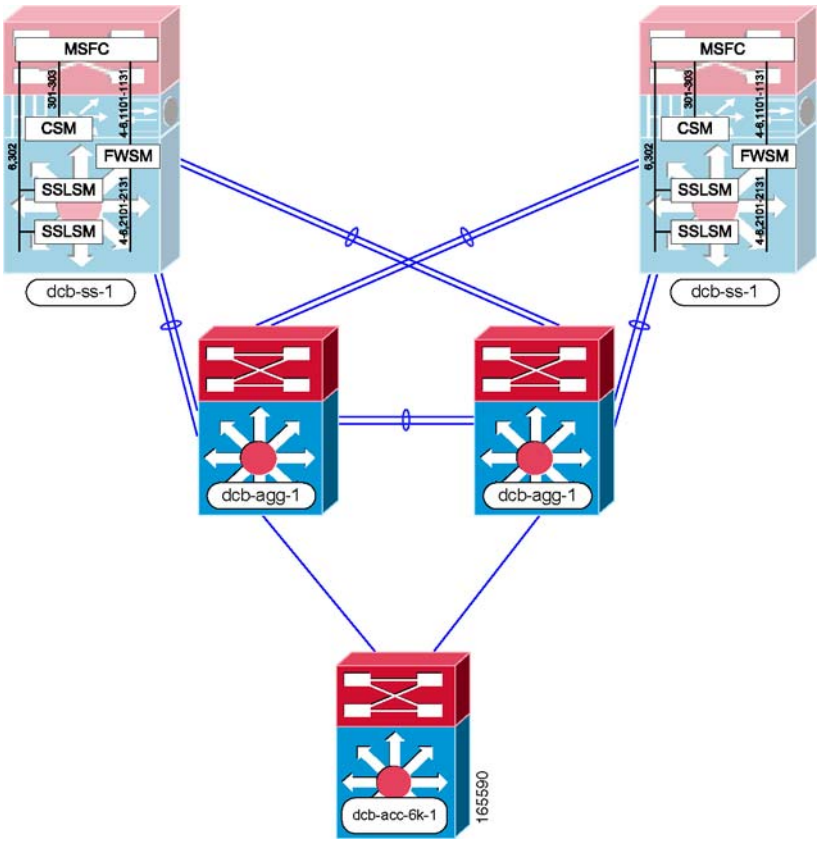
There are four SSLSMs in the DCAP DC-A LAN test topology, two in each of the Aggregation Layer devices. The SSLSMs are neither active nor standby; they work in a pool to handle encryption services for data center traffic. Each of the CSMs is configured with a serverfarm called "SSLSM" which contains four "real servers." The IP addresses of these real servers are the inband IP addresses of the four SSLSMs. When HTTPS traffic comes into the CSM to be load-balanced, it is handed off to this serverfarm and the decryption request is handled by one of the four SSLSMs.

Though the four SSLSMs are located in separate physical switches (dca-agg-1 and dca-agg-2 or dcb-ss-1 and dcb-ss-2) they are used as if they were in one location. The encryption and decryption requests traverse the interswitch etherchannels, if need be, to reach the necessary SSLSM.

Service Switch Configuration

Logically, the Service Module bundle in the Service Switch deployment handles traffic in the same manner as in the integrated bundle. The Service Module Configurations are nearly identical, down to the VLAN. Physically, the Service Switch is quite different, as can be seen in Figure 3-6.

Figure 3-6 Logical and Physical Configuration of the Service Switch Deployment in DC-B



There are two Service Switches, dcb-ss-1 and dcb-ss-2. As in the integrated bundle switches dca-agg-1 and dca-agg-2, these Service Switches have a single CSM, a single FWSM, and two SSLSMs. Internally, they are connected in the same way. Externally, each Service Switch chassis is dual-homed to the pair of Aggregation Layer devices, dcb-agg-1 and dcb-agg-2. If they were connected to the Aggregation Layer via a single etherchannel, and that channel was broken, a failover event would occur with regards to the active CSM and FWSM. It would take 3 seconds plus the amount of time necessary to re-build any TCP connections to restore traffic if such a failover event occurred. With dual-homing, the sub-second convergence time of the Rapid PVST+ Spanning-Tree Protocol can be relied on to ensure that such a failover event does not occur.

The four etherchannels connecting the Service Switches to the Aggregation Layer switches carry all of the inside and outside data VLANs, 1101-1131 and 2101-2131, as well as the VLANs necessary for connection replication, redundancy, and out-of-band management. VLAN 10 is also configured to facilitate OSPF adjacency between the two Service Switches and the two Aggregation Layer devices and help make the networks residing on dcb-ss-1 and dcb-ss-2 known to the rest of the DC-A LAN topology. The Spanning-Tree configuration remains unchanged, with dcb-agg-1 as the primary STP root and dcb-agg-2 as the secondary root.

It is important to mention some changes that are necessary at Layer 3 to support the Service Switch model. For those inside VLANs whose networks are advertised out to the world, the Service Switches share default gateway duties via HSRP. The device dcb-ss-1 is configured with the higher HSRP priority and would thus be the active HSRP router in a steady-state condition, with dcb-ss-2 waiting as standby. (For those VLANs carrying traffic that is not serviced by the Service Switch bundle, dcb-agg-1 and dcb-agg-2 share the HSRP responsibilities, as in the integrated bundle setup.)

Layer 4-7 Test Results Summary

Table 3-1 summarizes tests executed as part of the Cisco DCAP 2.0 testing initiative. Table 3-1 includes the feature or function tested, the section that describes the feature set the feature or function belongs to, the component tests for each feature or function, and whether the test is new in this phase of DCAP testing.

Note Test results are unique to technologies covered and actual scenarios in which they were tested. DCAP is designed to cover critical path areas and augment ongoing regression and systems testing.

Table 3-1 Cisco DCAP 2.0 L4-7 Testing Summary

Test Suites	Features/Functions	Tests	Results
Aggregation Switch Bundle Testing	CSM/FWSM Integration, page 3-11	1. Active FTP Through FWSM and CSM 2. Passive FTP Through FWSM and CSM 3. ICMP to a CSM Layer 3 and Layer 4 Vserver 4. DNS Query Through CSM and FWSM 5. FWSM and CSM Layer 4 SYN Attack 6. Idle Timeout UDP	CSCsd00731 CSCsd00731
	CSM/SSLSM Integration, page 3-23	1. Backend SSL 2. SSL Sticky 3. URL Rewrite	CSCse24138
	Redundancy, page 3-32	1. FWSM Redundancy 2. CSM Redundancy 3. SSLM Reset 4. Sorry Server 5. HSRP Failover	CSCdz22187 CSCsg97985
Service Switch Bundle Testing	CSM/FWSM Integration, page 3-11	1. Active FTP Through FWSM and CSM 2. Passive FTP Through FWSM and CSM 3. ICMP to a CSM Layer 3 and Layer 4 Vserver 4. DNS Query Through CSM and FWSM 5. FWSM CSM Layer4 SYN Attack 6. Idle Timeout UDP	

Table 3-1 Cisco DCAP 2.0 L4-7 Testing Summary (continued)

Test Suites	Features/Functions	Tests	Results
Service Switch Bundle Testing	CSM/SSLSM Integration, page 3-23	1. Backend SSL 2. SSL Sticky 3. URL Rewrite	
	Redundancy, page 3-32	1. FWSM Redundancy 2. CSM Redundancy 3. SSLM Reset 4. Sorry Server 5. HSRP Failover	CSCdz22187 CSCsg97985

Layer 4-7 DDTS Summary

Table 2 lists Development Defect Tracking System (DDTS) software bugs with descriptions, and comments filed by the DCAP testing team during Cisco DCAP 2.0 L4-7 testing. Table 3 lists DDTS with descriptions encountered during Cisco DCAP 2.0 L4-7 testing.

Table 2 Summary of DDTS Filed During Cisco DCAP 2.0 L4-7 Testing

CSCsh25315	Standby CSM crashed while issuing vserver inservice command.

Table 3 Summary of DDTS Encountered During Cisco DCAP 2.0 L4-7 Testing

CSCsd00731	Show mod csm X conns displays the active and passive ftp data port has 0.
CSCdz22187	CSM mark vserver OUTOFSERVICE when backup serverfarm is OPERATIONAL.
CSCsg97985	CSM to track the backup serverfarm by default if configured.
CSCse24138	CSM to track the backup serverfarm by default if configured.

Layer 4-7 Test Cases

Functionality critical to global enterprises in Cisco DCAP 2.0 L4-7 testing is described in the following sections. Refer to Cisco Data Center Assurance Program (DCAP) 2.0 Configurations document for test device configurations.

- Aggregation Switch Bundle Testing, page 3-11
- Service Switch Bundle Testing, page 3-44

Aggregation Switch Bundle Testing

The following tests verified various functional aspects of the three Service Modules (CSM, FWSM and SSLSM) in a bundled fashion; that is, working together in the same chassis to provide services to data center traffic. Three Service Modules are bundled together in the Aggregation Layer switches in DC-A, dca-agg-1 and dca-agg-2.

The following test features were conducted:

- CSM/FWSM Integration, page 3-11
- CSM/SSLSM Integration, page 3-23
- Redundancy, page 3-32

CSM/FWSM Integration

CSM/FWSM integration looks at interoperability capacities of the CSM and FWSM, in terms of how they work together to handle data traffic.

The following tests were performed:

- Active FTP Through FWSM and CSM, page 3-44

Active FTP Through FWSM and CSM

This test verified that the FWSM and CSM properly handled active FTP traffic when the **ftp fixup protocol 21** was enabled and disabled on the FWSM. FTP traffic was sent from an outside client to vserver VIP-ACTIVE-FTP and from an inside client to an outside server.

Relevant CSM configuration

```
 real P1_ETH1.3001
  address 101.1.1.11
  location dca-penguin-1
  inservice
 real P1_ETH1.3002
  address 101.1.1.12
  location dca-penguin-1
  inservice
!
 serverfarm FARM1-A
  nat server
  no nat client
  predictor leastconns
  real name P1_ETH1.3001
   inservice
  real name P1_ETH1.3002
   inservice
!
 vserver VIP-ACTIVE-FTP
  virtual 101.40.1.251 tcp ftp
  vlan 301
  serverfarm FARM1-A
  advertise active
  replicate csrp connection
  persistent rebalance
  inservice
!
```

Test Procedure

The procedure used to perform the Active FTP Through FWSM and CSM test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the following commands on the FWSM to clear connections and counters in the VLAN 1101-2101 context:

change context Vlan1101-2101

clear xlate

clear conn

clear access-list ACL-in count

clear log

Step 3 Issue the following commands on the FWSM to verify connections and counters have been cleared:

change context Vlan1101-2101

show xlate

show conn

show access-list ACL-in

Step 4 Issue the following commands on the Active CSM to clear connections and counters:

clear mod csm 2 counters

clear mod csm 2 connections

Step 5 Issue the following commands on the Active CSM to verify the counters have been cleared:

show mod csm 2 vserver name vip-active-ftp detail

show mod csm 2 real sfarm farm1-a detail

show mod csm 2 stats

show mod csm 2 conns

Step 6 Send ACTIVE FTP traffic to vserver VIP-ACTIVE-FTP from an outside client.

Step 7 Issue the following command on the FWSM to verify the FTP control and data channels were successfully created:

change context Vlan1101-2101

show xlate

show conn

show log

Step 8 Issue the **show mod csm 2 conns** command to verify the FTP control and data connections have been established.

Step 9 When the FTP traffic has completed issue the following command on the FWSM to verify a match on the correct access-list:

show access-list ACL-in | include extended permit tcp any 101.1.1.0 255.255.255.0 eq ftp

Step 10 Issue the following command on the Active CSM to verify the FTP traffic was properly load balanced:

show mod csm 2 vserver name vip-active-ftp detail

show mod csm 2 real sfarm farm1-a detail

show mod csm 2 stats

Step 11 On the FWSM context VLAN 1101-2101, configure **no fixup protocol ftp 21**.

The **fixup protocol ftp 21** configuration is part of the default configuration for the DCAP test topology.

Step 12 Send an active FTP request from an inside client to an outside server.

This connection should fail. When **no fixup protocol ftp 21** has been configured only passive mode ftp is allowed from an inside interface to an outside interface.

Step 13 Issue the following command on the FWSM to verify the FTP data channel was not successfully created:

change context Vlan1101-2101

show xlate

show conn

show log

Step 14 Reconfigure **fixup protocol ftp 21** on the VLAN 1101-2101 context to enable the fixup protocol for FTP on port 21 and use the **show fixup protocol ftp** command to verify it is now been enabled.

Step 15 Issue the following commands on the FWSM to clear connections and counters:

change context Vlan1101-2101

clear xlate

clear conn

clear access-list ACL-in count

clear access-list outgoing count

clear log

Step 16 Send active FTP traffic to vserver VIP-ACTIVE-FTP from an outside client.

Step 17 Issue the following command on the FWSM to verify the FTP control and data channels were successfully created:

change context Vlan1101-2101

show xlate

show conn

show log

Step 18 Issue the **show mod csm 2 conns** command to verify the FTP control and data connections have been established.

Step 19 Stop background scripts to collect final status of network devices and analyze for error.

Step 20 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the FWSM to permit Active FTP on the Outside Interface.
- We expect the FWSM would deny Active FTP on the Inside to Outside interface when fix up protocol ftp 21 is disabled.

- We expect the CSM vserver to properly load balance Active FTP.

Results

Active FTP Through FWSM and CSM passed with exception CSCsd00731.

Passive FTP Through FWSM and CSM

This test verified that the FWSM and CSM properly handled PASSIVE FTP traffic when the FTP fix up was enabled and disabled on the FWSM. FTP traffic was sent from outside client to vserver VIP-PASSIVE-FTP with FTP fix up enabled on the FWSM and when it was disabled. The same was done for FTP GET requests coming from an inside client to an outside server.

Relevant CSM configuration

```
!
module csm 2
 real P1_ETH1.3001
  address 101.1.1.11
  location dca-penguin-1
  inservice
 real P1_ETH1.3002
  address 101.1.1.12
  location dca-penguin-1
  inservice
!
 serverfarm FARM1-A
  nat server
  no nat client
  predictor leastconns
  real name P1_ETH1.3001
   inservice
  real name P1_ETH1.3002
   inservice
!
 vserver VIP-PASSIVE-FTP
  virtual 101.40.1.252 tcp ftp service ftp
  vlan 301
  serverfarm FARM1-A
  advertise active
  replicate csrp connection
  persistent rebalance
  inservice
!
```

Relevant FWSM configuration (context VLAN 1101-2101)

```
!
fixup protocol ftp
!
```

Test Procedure

The procedure used to perform the Passive FTP Through FWSM and CSM test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 On dca-agg-1, using the **show module csm 2 vserver name vip-passive-ftp detail** command, verify that the CSM vserver VIP-PASSIVE-FTP is configured for "service ftp" and that it is pointing to the serverfarm FARM1.

The output of the command shows that the serverfarm that is being used is **FARM1** and that **service = ftp** is enabled.

Step 3 Using the **show module csm 2 serverfarm name farm1-a detail** command, verify that there are two real servers in serverfarm FARM1-A and that they are both OPERATIONAL.

Step 4 On the active FWSM, in context VLAN 1101-2101, use the **show fixup** command to verify that **fixup protocol ftp 21** is not configured. If it is configured, use the **no fixup protocol ftp** command to disable it.

Step 5 From an outside client, send a single passive FTP GET to vserver VIP-PASSIVE-FTP and verify that it fails.

The connection fails because the **fixup protocol ftp** has been disabled on the active FWSM.

Step 6 Send a single passive FTP request from an inside client to the outside server.

This connection should succeed. When FTP fixups have been disabled, only passive mode FTP is allowed from an inside interface to an outside interface (active FTP is disallowed).

Step 7 Configure **fixup protocol ftp 21** on the active FWSM context VLAN 1101-2101 to enable the fix up protocol for FTP on port 21.

Step 8 Issue the following commands on the active FWSM context VLAN 1101-2101 to clear connections and counters:

clear xlate

clear conn

clear log

Step 9 Issue the following commands on the active CSM to clear connections and counters:

clear module csm 2 counters

clear module csm 2 connections

Step 10 Send a single passive FTP GET request for a very large file from an outside client to the CSM vserver VIP-PASSIVE-FTP.

Step 11 While the GET is underway, issue the following commands on the active FWSM context VLAN 1101-2101 to verify the FTP Control and Data Channels were successfully created:

show conn

show xlate

show log

Step 12 While the GET is underway, issue the **show module csm 2 conn** command to verify the FTP control and data connections have been established.

Step 13 Send 20 passive FTP GETs from an outside client to the CSM vserver VIP-PASSIVE-FTP.

Each of these should succeed.

Step 14 On the active CSM, use the **show module csm 2 real sfarm farm1-a detail** to verify that the previous GET requests have been load-balanced evenly across both servers in serverfarm FARM1-A.

Each real server listed in the output should show about the same number of "total connections established".

Step 15 Send a single passive FTP request from inside client to the outside server.

This connection should succeed.

Step 16 Stop background scripts to collect final status of network devices and analyze for error.

Step 17 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the FWSM to permit Passive FTP on the Inside Interface.
- We expect the FWSM to deny Passive FTP on the Outside interface when fixup protocol ftp 21 is disabled.
- We expect the CSM vserver to properly load balance Active FTP.
- We expect no unacceptable impact on the CPU or memory of the DUT.

Results

Passive FTP Through FWSM and CSM passed with exception CSCsd00731.

ICMP to a CSM Layer 3 and Layer 4 Vserver

This test verified ICMP PING traffic to Mulitple Layer 4 vservers and a Layer 3 vserver all configured with the same virtual IP address. The CSM virtual address is located on the outside of the FWSM and the CSM reals or located on the inside of the CSM.

Relevant CSM configuration

```
!
 vserver DMZ1-FTP
  virtual 201.40.40.240 tcp ftp service ftp
  vlan 301
  serverfarm FARM1
  advertise active
  persistent rebalance
  inservice
!
vserver VIP-DNS
  virtual 201.40.40.240 udp dns service per-packet
  vlan 301
  serverfarm FARM1
  advertise active
  persistent rebalance
  inservice
!
 vserver VIP-L3
  virtual 201.40.40.240 any
  vlan 301
  serverfarm FARM1-A
  advertise active
  persistent rebalance
  inservice
!
 vserver VIP-WWW
  virtual 201.40.40.240 tcp www
  vlan 301
  serverfarm FARM1-A
  advertise active
  persistent rebalance
```

```
  inservice
!
```

Test Procedure

The procedure used to perform the ICMP to a CSM Layer 3 and Layer 4 Vserver test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the **clear module csm 2 counters** command to clear all CSM statistics.

Step 3 Issue the following commands on the Active FWSM in context VLAN 1101-2101 to clear connections and counters.

clear xlate

clear conn

clear log

Step 4 Suspend CSM vserver VIP-L3 with the **no inservice** command.

Step 5 From an outside linux client send ICMP PING to CSM vserver VIP-WWW. This ping should be successful.

Step 6 On the Active FWSM issue the **show xlate** command. You should see zero global entries because only Layer 3 vservers load balance pings to reals.

Step 7 Verify the following vservers have not recorded any policy matches or packets received by issuing the following commands.

show module csm 2 vservers name DMZ1-FTP detail

show module csm 2 vservers name vip-dns detail

show module csm 2 vservers name vip-www detail

show module csm 2 vservers name vip-l3 detail

Step 8 Enable CSM vserver VIP-L3 with the **inservice** command and verify its now operational with the **show module csm 2 vserver vip-l3 detail** command.

Step 9 From an outside Linux client send ICMP PING to CSM vserver VIP-L3. This ping should be successful.

Step 10 On the Active FWSM issue the **show xlate** command. You should see a global entry for each real in the serverfarm because only Layer 3 vservers load balance pings request to reals.

Step 11 Verify only vserver VIP-L3 has recorded policy match and packets received by issuing the following commands.

show module csm 2 vservers name DMZ1-FTP detail

show module csm 2 vservers name vip-dns detail

show module csm 2 vservers name vip-www detail

show module csm 2 vservers name vip-l3 detail

Step 12 Suspend the following vservers with the **no inservice** command: DMZ1-FTP, VIP-DNS, VIP-WWW, and VIP-L3.

Step 13 From an outside Linux client send ICMP PING to CSM vserver VIP-WWW. This ping should be unsuccessful because all four vserver configured with the same virtual IP have been taken out of service.

Step 14 Enable the following vservers with the **inservice** command

Vsever DMZ1-FTP

VIP-DNS

VIP-WWW

VIP-L3

Step 15 Stop background scripts to collect final status of network devices and analyze for error.

Step 16 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the CSM will NOT Load Balance ICMP for a layer 4 vserver.
- We expect the CSM to Load Balance ICMP for a layer 3 vserver.
- We expect the FWSM to create a connection for ICMP when fixup protocol ICMP is configured.
- We expect vservers to respond to ICMP when operational.
- We expect a vserver not to respond to ICMP when not operational.

Results

ICMP to a CSM Layer 3 and Layer 4 Vserver passed.

DNS Query Through CSM and FWSM

This test verified that the FWSM and CSM properly handled DNS traffic when fix up protocol DNS was enabled. In this test the CSM virtual is on the outside of the FWSM and the reals are on the inside of the FWSM.

Relevant CSM configuration

```
vserver VIP-DNS
  virtual 201.40.40.240 udp dns service per-packet
  vlan 301
  serverfarm FARM1
  advertise active
  persistent rebalance
  inservice
```

Relevant FWSM Access-list

```
access-list ACL-in extended permit udp any 201.1.1.0 255.255.255.0
```

Test Procedure

The procedure used to perform the DNS Query Through CSM and FWSM test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the following commands on the Active FWSM in context VLAN 1101-2101 to clear connections and counters:

clear xlate

clear conn

clear access-list ACL-in count

clear log

Step 3 Issue the **clear module csm 2 counters** and **clear module csm 2 connections** commands on the Active CSM to clear connections and counters.

Step 4 Use Spirent Avalanche to send a small set of DNS queries to vserver VIP-DNS for domain name www.datacenter.com.

Step 5 Issue the **show xlate** command on the Active FWSM to verify that a global entry was created for each real in serverfarm FARM1.

Step 6 Issue the **show show access-list ACL-in** command to verify there are matches on the portion of the access-list that permits UDP DNS queries.

The ACL line that permits this traffic is:

access-list ACL-in extended permit udp any 201.1.1.0 255.255.255.0

Step 7 Issue the following commands on the Active CSM to verify the DNS traffic was properly load balanced:

show module csm 2 vserver name vip-dns detail

show module csm 2 stats

The "total conns" should approximate the number of hits that was seen on the FWSM access-list.

Step 8 Issue the **show module csm 2 real sfarm FARM1 detail** command to verify that each real server in the serverfarm has made some connections.

The "total conns established" counter is about 7 for each real in the farm.

Step 9 Issue the **clear module csm 2 counters** and **clear module csm 2 connections** commands on the Active CSM to clear connections and counters.

Step 10 Use Spirent Avalanche to send DNS quires to vserver VIP-DNS for domain name www.datacenter.com at rate of one thousand users per second.

Step 11 While traffic is running issue the **show xlate** and **show conn | include most** commands on the VLAN 1101-2101 FWSM context to verify the xlate table and number of open connections.

You should see 65 global entries in the xlate table.

Step 12 Verify the number of attempts and number of failures on the Spirent Avalanche run tab.

Step 13 Verify the DNS connections rate on the Spirent Avalanche client stats tab and the select DNS. DNS quires per second should be around 1,000.

Step 14 Issue the following commands on the Active CSM to verify the DNS traffic was properly load balanced. Counters Tot matches and L4 Load-Balanced Decisions should have the same value. Verify the Tot matches counter equals the number of attempts on the Spirent Avalanche run tab.

show module csm 2 vserver name vip-dns detail

show module csm 2 stats

Step 15 Issue the **clear module csm 2 counters** and **clear module csm 2 connections** commands on the Active CSM to clear connections and counters.

Step 16 Use Spirent Avalanche to send DNS quires to vserver VIP-DNS for domain name www.datacenter.com at rate of Fifteen hundred users per second. At this rate we expect a large number of UDP request to fail.

Step 17 While traffic is running issue the following two commands to verify the xlate table and number of open connections. You should see 65 global entries in the xlate table.

show xlate

show conn | in most

Step 18 Verify the number of attempts and number of failures on the Spirent Avalanche run tab.

Step 19 Verify the DNS connections rate on the Spirent Avalanche client stats tab and the select DNS. DNS quires per second should be around 1,500.

Step 20 Issue the following commands on the Active CSM to verify the DNS traffic was properly load balanced. Counters Tot matches and L4 Load-Balanced Decisions should have the same value. Verify the Tot matches counter equals the number of attempts on the Spirent Avalanche run tab.

show mod csm 2 vserver name vip-dns detail

show mod csm 2 stats

Step 21 Stop background scripts to collect final status of network devices and analyze for error.

Step 22 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the FWSM to permit DNS traffic to vserver VIP-DNS.
- We expect the FWSM to NAT the DNS response to the outside client when fixup protocol DNS is enabled.
- We expect the FWSM not to NAT the DNS response to the inside client when fixup protocol DNS is enabled.
- We expect the CSM vserver to properly load balance DNS traffic.

Results

DNS Query Through CSM and FWSM passed.

FWSM and CSM Layer 4 SYN Attack

Syn-flood attacks aim at preventing a TCP/IP server from servicing request. The SYN flag is set in a TCP segment when a connection request is sent by a computer. The target server responds back with an ACK and waits for a response from the initiator. The SYN-Flood attacker spoofs the source IP address so that the server never receives a response to the ACK. This causes the server to use up resources overloading the server and preventing it from responding to legitimate connection request.

TCP Intercept protects inside systems from a DoS attack perpetrated by flooding an interface with TCP SYN packets. Enable this feature by setting the maximum embryonic connections option of the Nat and static commands.

When the embryonic limit has been surpassed, the TCP intercept feature intercepts TCP SYN packets from clients to servers on a higher security level. The TCP intercept feature implements software to protect TCP servers from TCP SYN-flooding attacks, which are a type of denial-of-service attack. SYN cookies are used during the validation process and help to minimize the amount of valid traffic being dropped.

The embryonic limit is a feature that is enabled for any inbound connection (a connection that the FWSM considers from lower to higher security). In order for a connection to be inbound, you need to either hit a static or a global xlate.

This test verified the TCP Intercept feature by sending 1 million SYN packets generated on a Linux server using random source ip address. The SYN packets were sent to a CSM layer 4 server with 65 reals behind the FWSM.

Relevant CSM configuration

```
vserver VIP-WWW
  virtual 201.40.40.240 tcp www
  vlan 301
  serverfarm FARM1
  advertise active
  persistent rebalance
  inservice
```

Test Procedure

The procedure used to perform the FWSM and CSM Layer 4 SYN Attack test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Configure context VLAN 1101-2101 on the Active FWSM with the following static command to enable the limitation of embryonic connections to 20.

static (inside,outside) 201.1.1.0 201.1.1.0 netmask 255.255.255.0 tcp 0 20

Step 3 Issue the **clear module csm 2 counters** command to clear all CSM statistics and **clear module csm 2 conn** to clear all connections.

Step 4 Verify CSM utilization by issuing the **show module csm 2 tech-support utilization** command.

Step 5 On the FWSM system context, clear the Fast Path Syn Cookie Statistics Counters for NP-1 and NP-2 with the **clear np 1 syn** and **clear np 2 syn** commands.

Step 6 Verify CPU and memory utilization on the FWSM by issuing the **show cpu** and **show memory** commands from the system context.

Step 7 From the outside client send 10,000,000 syn packets to vserver VIP-WWW with random source ip addresses.

Step 8 While the syn attack traffic is being sent verify the rate of the syn attack on the FWSM by issuing the **show perfmon | inc TCP Intercept** command. Issue the command several times to obtain a good baseline.

Step 9 While syn attack traffic is being sent verify CSM utilization by issuing the **show module csm 2 tech-support utilization** command.

Step 10 Verify there are no errors on the CSM by issuing the following commands.

show mod csm 2 vserver name vip-www detail

show mod csm 2 reals sfarm farm1-a det

show mod csm 2 stats

Step 11 Verify the FWSM is issued a syn cookie and verify the number of syn packets intercepted by issuing the following commands.

show np 1 syn

show np 2 syn

Step 12 Verify FWSM CPU and memory utilization were not adversely impacted by the syn attack by issuing the **show cpu** and **show memory** commands.

Step 13 Verify the FWSM log contains message number FWSM-6-106015 by issuing the **show log** command in context VLAN 1101-2101.

Step 14 Remove static statement from VLAN 1101-2101 on the Active FWSM with the following command.

no static (inside,outside) 201.1.1.0 201.1.1.0 netmask 255.255.255.0 tcp 0 20

Step 15 Stop background scripts to collect final status of network devices and analyze for error.

Step 16 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the FWSM to intercept syn packets being sent to the CSM reals by issuing a syn cookie.
- We except CPU and memory utilization on the CSM and FWSM not to be adversely impacted by the syn attack.
- We expect the CSM to evenly load balance packets across all reals in the serverfarm.

Results

FWSM and CSM Layer 4 SYN Attack passed.

Idle Timeout UDP

The test verified the CSM will remove idle UDP connections at 60 seconds and the FWSM will remove them after two minutes. It also verified that the CSM load-balanced the UDP connections.

The CSM vserver VIP-TFTP has been configured with a 60 second idle timer. A TFTP copy request (UDP port 69) was generated on a Linux client, to the VIP-TFTP, to create a connection on the CSM and FWSM. It was verified that these connections were load-balanced properly to the real servers in the serverfarm. It was then verified that these connections timed out after 60 seconds on the CSM and 2 minutes on the FWSM.

Relevant CSM configuration

```
!
 vserver VIP-TFTP
  virtual 101.40.40.244 udp 0
  vlan 301
  serverfarm FARM1-A
  advertise active
  idle 60
  persistent rebalance
  inservice
!
```

Test Procedure

The procedure used to perform the Idle Timeout UDP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify CSM vserver VIP-TFTP is operational and the idle time out is set to 60 by issuing the **show mod csm 2 vserver name vip-tftp detail** command.

Step 3 Verify all reals are operational for CSM serverfarm FARM1-A by issuing the **show mod csm 2 real sfarm farm1-a detail** command.

Step 4 Clear all counters and connections on the CSM by issuing the **clear mod csm 2 counters** and **clear mod csm 2 conn** commands.

On the linux client dca-penguin-15, perform a single TFTP copy request to the VIP-TFTP using the tftp 101.40.40.244 -c get file.txt command.

Step 5 On the active CSM, use the **show mod csm 2 serverfarm name farm1-a detail** command to verify that UDP connections have been load balanced across the two real servers in serverfarm FARM1-A.

Each of the two real servers shows one connections apiece.

Step 6 On the active CSM, use the **show mod csm 2 conn vserver vip-tftp** command to verify that UDP connections have been created for the TFTP transfer.

Step 7 Use the **show clock** and **show mod csm 2 conn vserver vip-tftp** commands to verify that the UDP connections time out after one minute.

Step 8 Issue the **show timeout** command on the active FWSM in context VLAN 1101-2101 to verify timeout udp is set to two minutes.

Step 9 Issue the **clear conn** command on the Active FWSM in context VLAN 1101-2101 to clear connections.

On the linux client dca-penguin-15, perform a single TFTP copy request to the VIP-TFTP using the tftp 101.40.40.244 -c get file.txt command.

Step 10 On the active FWSM, use the **show conn** command to verify that UDP connections have been created for the TFTP transfer.

Step 11 Use the **show clock** and **show conn** commands to verify that the UDP connections on the FWSM time out after two minutes.

Step 12 Stop background scripts to collect final status of network devices and analyze for error.

Step 13 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect flows that exceed the idle timeout will be cleared.
- We expect the CSM vserver to properly load balance UDP traffic.

Results

Idle Timeout UDP passed.

CSM/SSLSM Integration

CSM/SSLSM integration looks at interoperability capacities of the CSM and SSLSM, in terms of how they work together to handle data traffic.

The following tests were performed:

- Backend SSL, page 3-24
- SSL Sticky, page 3-28
- URL Rewrite, page 3-30

Backend SSL

This test verified that the CSM and SSLM successfully work together to load balance SSL traffic on the client side internally decrypted the traffic for advanced layer 7 processing then re-encrypt the traffic load balancing to the backend servers. This test also verified the CSM is able to stick clients to the same real based on cookies.

The CSM and SSLM communicate together on an internal VLAN in Routed Mode. The CSM communicates with the clients and reals in bridged mode. Clients access the CSM virtual addresses through static NAT mappings on the FWSM.

Relevant CSM Configuration

```
Vlan 100 client
  description DMZ1
  ip address 192.168.100.4 255.255.255.0
  gateway 192.168.100.1
!
 vlan 200 server
  description DMZ1
  ip address 192.168.100.4 255.255.255.0
  alias 192.168.100.3 255.255.255.0
!
 vlan 172 server
  description SSLM
  ip address 172.16.100.2 255.255.255.0
  alias 172.16.100.1 255.255.255.0
!
 probe TCP tcp
  interval 5
  failed 2
!
 probe ICMP icmp
!
 map 1.GIF url
  match protocol http url /1.gif
!
 map 2.GIF url
 match protocol http url /2.gif
!
 real BACKEND-SSL33
  address 172.16.100.133
  inservice
 real BACKEND-SSL34
  address 172.16.100.134
  inservice
 real BACKEND-SSL35
  address 172.16.100.135
  inservice
 real BACKEND-SSL36
  address 172.16.100.136
  inservice
real FRONTEND-SSL1
  address 172.16.100.103
  inservice
 real FRONTEND-SSL2
  address 172.16.100.113
  inservice
!
 serverfarm BACKEND-SSL
  nat server
  no nat client
  real name BACKEND-SSL33
```

```
   inservice
  real name BACKEND-SSL34
   inservice
  real name BACKEND-SSL35
   inservice
  real name BACKEND-SSL36
   inservice
  probe TCP
!
 serverfarm FRONTEND-SSL
  nat server
  no nat client
  real name FRONTEND-SSL1
   inservice
  real name FRONTEND-SSL2
   inservice
  probe TCP
!
 serverfarm ROUTE
  nat server
  no nat client
  predictor forward
!
 sticky 3 cookie backend-ssl timeout 30
!
 policy 1.GIF
  url-map 1.GIF
  sticky-group 3
  serverfarm BACKEND-SSL
!
 policy 2.GIF
  url-map 2.GIF
  sticky-group 3
  serverfarm BACKEND-SSL
!
vserver BACKEND-SSL
  virtual 172.16.100.203 tcp www
  vlan 172
  serverfarm BACKEND-SSL
  sticky 30 group 3
  persistent rebalance
  slb-policy 1.GIF
  slb-policy 2.GIF
  inservice
!
 vserver FORWARD
  virtual 192.168.100.0 255.255.255.0 any
  vlan 172
  serverfarm ROUTE
  persistent rebalance
  inservice
!
 vserver FRONTEND-SSL
  virtual 192.168.100.203 tcp https
  serverfarm FRONTEND-SSL
  ssl-sticky offset 20 length 6
  sticky 30 group 30
  persistent rebalance
  inservice
!
```

Relevant SSLM1 Configuration

```
ssl-proxy policy http-header backend-ssl
```

```
 custom "backend-ssl:on"
 client-ip-port
!
ssl-proxy service backend-ssl33 client
 virtual ipaddr 172.16.100.133 protocol tcp port 80
 server ipaddr 192.168.100.100 protocol tcp port 443
 trusted-ca backend-ssl
 authenticate verify signature-only
 inservice
!
ssl-proxy service backend-ssl34 client
 virtual ipaddr 172.16.100.134 protocol tcp port 80
 server ipaddr 192.168.100.110 protocol tcp port 443
 trusted-ca backend-ssl
 authenticate verify signature-only
 inservice
!
ssl-proxy service frontend-ssl
 virtual ipaddr 172.16.100.103 protocol tcp port 443
 server ipaddr 172.16.100.203 protocol tcp port 80
 certificate rsa general-purpose trustpoint backend-ssl
 policy http-header backend-ssl
 inservice
!
ssl-proxy pool ca backend-ssl
 ca trustpoint bundle-root
!
crypto ca trustpoint backend-ssl
 enrollment terminal pem
 fqdn www.backend-ssl.com
 subject-name C=US, ST=Massachusetts, L=Boxborough, O=Cisco, OU=Safeharbor,
CN=www.backend-ssl.com
 rsakeypair backend-ssl
!
crypto ca trustpoint bundle-root
 enrollment terminal pem
!
```

Relevant SSLM2 Configuration

```
 ssl-proxy policy http-header backend-ssl
 custom "backend-ssl:on"
 client-ip-port
!
ssl-proxy service backend-ssl35 client
 virtual ipaddr 172.16.100.133 protocol tcp port 80
 server ipaddr 192.168.100.100 protocol tcp port 443
 trusted-ca backend-ssl
 authenticate verify signature-only
 inservice
!
ssl-proxy service backend-ssl36 client
 virtual ipaddr 172.16.100.134 protocol tcp port 80
 server ipaddr 192.168.100.110 protocol tcp port 443
 trusted-ca backend-ssl
 authenticate verify signature-only
 inservice
!
ssl-proxy service frontend-ssl
 virtual ipaddr 172.16.100.103 protocol tcp port 443
 server ipaddr 172.16.100.203 protocol tcp port 80
 certificate rsa general-purpose trustpoint backend-ssl
 policy http-header backend-ssl
 inservice
```

```
!
ssl-proxy pool ca backend-ssl
 ca trustpoint bundle-root
!
crypto ca trustpoint backend-ssl
 enrollment terminal pem
 fqdn www.backend-ssl.com
 subject-name C=US, ST=Massachusetts, L=Boxborough, O=Cisco, OU=Safeharbor,
CN=www.backend-ssl.com
 rsakeypair backend-ssl
!
crypto ca trustpoint bundle-root
 enrollment terminal pem
!
```

Test Procedure

The procedure used to perform the Backend SSL test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the following command on all 4 SSLMs to clear statistics:

clear ssl-proxy stats service

clear ssl-proxy stats hdr

clear ssl-proxy stats ssl

Step 3 Issue the following commands on all 4 SSLMs to verify statistics have been cleared:

show ssl-proxy stats service

show ssl-proxy stats hdr

show ssl-proxy stats ssl client

Step 4 Issue the **show ssl-proxy service** command on all four SSLSMs to verify ssl-proxy services are operational.

Step 5 Issue the **clear mod csm 2 counters** command on the active CSM to clear counters.

Step 6 Issue the following commands to verify the counters have been cleared:

show mod csm 2 vserver name SSL30 detail

show mod csm 2 vserver name VIP30 detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 stats

Step 7 Issue the **clear mod csm 2 sticky all** command on the active CSM to clear the sticky table.

Step 8 Issue the **show mod csm 2 sticky** command on the active CSM to verify the sticky table.

Step 9 Send multiple HTTPS get requests for 1.html, 2.html and 3.html from the outside client to vserver FRONTEND-SSL. The client emulation tool will generate the traffic using 3 different cookies.

Step 10 Wait until client emulation traffic has completed then issue the **show mod csm 2 vservers name SSL30 detail** command to verify the Tot matches counter equals 600.

Step 11 Issue the **show mod csm 2 vservers name VIP30 detail** command to verify the Tot matches counter has incremented for the following 3 policies:

100 times for 1.GIF

200 times for 2.GIF

300 times for (default)

Step 12 Issue the **show mod csm 2 real sfarm FARM30 detail** command on the active CSM to verify the load balancing of connections.

Step 13 Issue the **show mod csm 2 stats** command on the active CSM to verify there are no errors.

Step 14 Issue the **show mod csm 2 sticky** command on the active CSM to verify the sticky table.

Step 15 Issue the **show ssl-proxy stats service BACKEND30** command on all SSLMs to verify the following two counters equal 600:

conns attempted

conns completed

Step 16 Issue the following commands on sslm-1 to very conns attempted and conns completed counter have incremented and there are no errors.

show ssl-proxy stats service BACKEND30

show ssl-proxy stats service SSL-backend

Step 17 Stop background scripts to collect final status of network devices and analyze for error.

Step 18 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the CSM to correctly load balance SSL traffic.
- We expect the CSM to apply the correct L7 policy on clear text traffic.
- We expect the CSM to be able to stick based on the client cookie.
- We expect the SSLSM to re-encrypt the clear text traffic and forward through the CSM to the backend server.
- We expect the SSLSM to insert client IP and Port information
- We expect the SSLM to insert the customer header.

Results

Backend SSL passed with exception CSCse24138.

SSL Sticky

This test verified the ability of the CSM to extract SSL Session ID and add a SSL entry to the sticky table. Subsequent SSL request containing the same SSL Session ID were sent to the same real server associated with that sticky entry. The real servers used in this test are SSL modules.

Test Procedure

The procedure used to perform the SSL Sticky test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the **clear mod csm 2 counters** command on the active CSM. Issue the following commands to verify the counters have been cleared:

show mod csm 2 vservers name SSL30 detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 stats

Step 3 Issue the **clear mod csm 2 sticky all** command on the active CSM to clear all sticky entries.

Step 4 Issue the **show mod csm 2 sticky** command to verify all ssl sticky entries have been cleared.

Step 5 Issue the **show ssl-proxy service** command on all four SSLMs to verify ssl-proxy service is operational.

Step 6 Issue the **clear ssl-proxy stats service** command on all four SSLMs to clear ssl-proxy service statistics.

Step 7 Issue the **show ssl-proxy stats service** command on all four SSLMs to verify statistics have been cleared.

Step 8 Begin initiating SSL GET requests to vserver SSL30. This involves a single user generating 240 HTTPS requests where a new SSL Session ID will be generated on every 30th request.

Step 9 Within 30 seconds after the traffic has started, issue the **show module csm 2 reals sfarm sslsm detail** command on the active CSM to verify that all of the connections up to this point are being sent ("stuck") to a single SSLSM "server."

The **total connections established** on one of the servers should be some value greater than 1 and less than 30. There should be no established connections on any of the other servers.

Step 10 When traffic has completed verify that connections were load balanced among the four SSLMs in serverfarm SSLMSM:

show mod csm 2 vservers name SSL30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 stats

Step 11 Use the **show module csm 2 sticky group 206** command on the active CSM to verify that the SSL sticky group has entries in it.

Step 12 Stop background scripts to collect final status of network devices and analyze for error.

Step 13 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the CSM to stick clients to the same real based on SSL Session ID.

Results

Backend SSL passed.

URL Rewrite

This test verified that the SSLM properly manipulated the data coming from the server with the use of the URL rewrite functionality. Server data that contains a 300 series redirect will be rewritten to HTTPS being forwarded to the client.

HTTPS and HTTP traffic for this test is load balanced by a CSM.

IE, Firefox and a client emulator will be used to test basic SSL Termination and URL Rewrite

Note Under the current time constraints we are not able to test every possible browser/version that exists today. The browsers were carefully selected to show any inconsistencies in SSL termination.

Relevant CSM Configuration

```
!
 sticky 241 ssl timeout 30
!
 vserver CLEAR-REWRITE
  virtual 201.40.40.241 tcp www
  vlan 302
  serverfarm FARM1-A
  persistent rebalance
  inservice
!
 vserver SSL-REWRITE
  virtual 201.40.40.241 tcp https
  vlan 301
  serverfarm SSLSM
  advertise active
  sticky 30 group 241
  persistent rebalance
  inservice
```

Relevant SSLM1 Configuration

```
!
 policy url-rewrite rewrite-test
  url 201.40.40.241
  url www.urlrewrite.com
  url www.urlrewrite1.com
  url www.urlrewrite2.com
  url www.urlrewrite3.com
  url www.urlrewrite4.com
  url www.urlrewrite5.com
  url www.urlrewrite6.com
  url www.urlrewrite7.com
  url www.urlrewrite8.com
  url www.urlrewrite9.com
  url www.urlrewrite10.com
  url www.urlrewrite11.com
  url www.urlrewrite12.com
  url www.urlrewrite13.com
  url www.urlrewrite14.com
  url www.urlrewrite15.com
  url www.urlrewrite16.com
  url www.urlrewrite17.com
  url www.urlrewrite18.com
  url www.urlrewrite19.com
  url www.urlrewrite20.com
  url www.urlrewrite21.com
  url www.urlrewrite22.com
```

```
 url www.urlrewrite23.com
 url www.urlrewrite24.com
 url www.urlrewrite25.com
 url www.urlrewrite26.com
 url www.urlrewrite27.com
 url www.urlrewrite28.com
 url www.urlrewrite29.com
 url www.urlrewrite30.com
!
service url-rewrite
 virtual ipaddr 201.40.40.241 protocol tcp port 443 secondary
 server ipaddr 200.1.0.20 protocol tcp port 80
 certificate rsa general-purpose trustpoint url-rewrite
 no nat server
 policy url-rewrite rewrite-test
 inservice
!
```

Test Procedure

The procedure used to perform the URL Rewrite test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 From the outside client use the client emulator to generate an HTTPS request to vserver SSL-REWRITE. Verify the location field of the HTTP 302 redirect packet was rewritten to HTTPS.

Step 3 Clear ssl-proxy service statistics and url statistics on each of the four SSLSMs in the topology by issuing the following:

clear ssl-proxy stats service url-rewrite context Default

clear ssl-proxy stats url

Step 4 Verify the ssl-proxy service statistics and url statistics have been cleared on each of the four SSLSMs in the topology by issuing the following commands:

show ssl-proxy stats service url-rewrite context Default

show ssl-proxy stats url

Step 5 Issue the **clear mod csm 2 count** command on the active CSM to clear csm counters.

Step 6 From the outside client use the client emulator to generate 1000 HTTPS request to vserver SSL-REWRITE.

Step 7 When client emulated traffic has completed issue the **show ssl-proxy stats url** command on all SSLMs to verify the Rewrites Succeeded counter has incremented for a combined total of 1000.

Step 8 Issue the **show ssl-proxy stats service url-rewrite** command on all SSLMs to verify the conns attempted and full handshakes counters have incremented to 1000.

Step 9 On the Active CSM verify the total matches counter for vserver SSL-REWRITE and vserver CLEAR-REWRITE equals 2000 by issuing the command **show mod csm 2 vserver name name detail** command.

Step 10 On the Active CSM verify traffic was evenly load balanced between all reals in serverfarm SSLSM and serverfarm FARM30 by issuing the **show mod csm 2 real sfarm *name* detail** command.

Step 11 Stop background scripts to collect final status of network devices and analyze for error.

Step 12 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that the SSLM can rewrite the server issued 300 series redirects from HTTP to HTTPS.

Results

URL Rewrite passed.

Redundancy

The resiliency of network resources and services to hardware and software compent failures is key to a successful high availability strategy in a data center network. Redundancy measures the effects of various failure scenarios on Layer 4-7 services and hardware.

The following tests were performed:

- FWSM Redundancy, page 3-32
- CSM Redundancy, page 3-34
- SSLM Reset, page 3-37
- Sorry Server, page 3-41
- HSRP Failover, page 3-42

FWSM Redundancy

This test verified that long lived flows being load balanced by the CSM and traversing the FWSM will be replicated between the Primary and Secondary FWSM. The ability of the system to successfully replicate flows and forward traffic after the failover was the criteria for a successful test run.

Test Procedure

The procedure used to perform the FWSM Redundancy test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the following commands on the Primary and Secondary FWSM:

show xlate

show conn

Step 3 Issue the **show failover** command on the Primary and Secondary FSWM to verify the Primary FWSM is in Active state.

Step 4 Issue the **clear mod csm 2 count** command on the active CSM to clear counters.

Step 5 Issue the following commands to verify the counters have been cleared:

show mod csm 2 vservers name VIP30 detail

show mod csm 2 vservers name SSL30 detail

show mod csm 2 vservers name VIP29 detail
show mod csm 2 vservers name SSL29 detail
show mod csm 2 real sfarm SSLSM detail
show mod csm 2 real sfarm FARM29 detail
show mod csm 2 real sfarm FARM30 detail
show mod csm 2 stats
show mod csm 2 conn

Step 6 Generate HTTPS traffic to vservers SSL30. Generate FTP traffic to vserver VIP1.

Step 7 Issue the following command on the Primary and Secondary FWSM to verify connections:

show xlate
show conn

Step 8 Issue the following commands on the active CSM to verify connections:

show mod csm 2 vservers name VIP30 detail
show mod csm 2 vservers name SSL30 detail
show mod csm 2 vservers name VIP29 detail
show mod csm 2 vservers name SSL29 detail
show mod csm 2 real sfarm SSLSM detail
show mod csm 2 real sfarm FARM29 detail
show mod csm 2 real sfarm FARM30 detail
show mod csm 2 stats
show mod csm 2 conn

Step 9 Issue the **reload** command on the Primary FWSM to force a reload.

Step 10 Issue the **show failover** command on the Secondary FSWM to verify it's now Active.

Step 11 Issue the following commands on the Secondary FWSM to verify connections:

show xlate
show conn

Step 12 Issue the following commands on the active CSM several times to verify connections:

show mod csm 2 vservers name VIP30 detail
show mod csm 2 vservers name SSL30 detail
show mod csm 2 vservers name VIP29 detail
show mod csm 2 vservers name SSL29 detail
show mod csm 2 real sfarm SSLSM detail
show mod csm 2 real sfarm FARM29 detail
show mod csm 2 real sfarm FARM30 detail
show mod csm 2 stats
show mod csm 2 conn

Step 13 When the Primary FWSM comes back on line issue the **show failover** command to verify it's in Standby state.

Step 14 Issue the following command on the Primary FWSM to verify connections have been replicated from the Secondary FWSM:

show xlate

show conn

Step 15 Issue the **reload** command on the Secondary FWSM to force a reload.

Step 16 Issue the **show failover** command on the Primary FSWM to verify it's now Active.

Step 17 Issue the following command on the Primary FWSM to verify connections:

show xlate

show conn

Step 18 Issue the following commands on the active CSM several times to verify connections:

show mod csm 2 vservers name VIP30 detail

show mod csm 2 vservers name SSL30 detail

show mod csm 2 vservers name VIP29 detail

show mod csm 2 vservers name SSL29 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM29 detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 stats

show mod csm 2 conn

Step 19 Wait for FTP traffic to complete and check for errors

Step 20 Stop background scripts to collect final status of network devices and analyze for error.

Step 21 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the FWSM to replicate flow information from active to standby FWSM.
- We expect the standby FWSM will transition to active state with the failure of the active FWSM.

Results

FWSM Redundancy passed.

CSM Redundancy

This test verified that flow information is replicated from the active CSM to the standby CSM. Upon a redundancy transition the standby CSM will become the new active CSM and process all flows which were originally created on the active CSM.

Test Procedure

The procedure used to perform the CSM Redundancy test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the **clear mod csm 2 counters** command to clear CSM counters on the active and standby CSM.

Step 3 Issue the following commands on the active and standby CSM to verify the counters have been cleared:

show mod csm 2 vservers name VIP30 detail

show mod csm 2 vservers name SSL30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 stats

show mod csm 2 conn

Step 4 Issue the **clear mod csm 2 sticky all** command on the active and standby CSM. Issue the **show mod csm 2 sticky** command to verify the sticky table was cleared.

Step 5 Issue the **clear ssl-proxy stats service** command on all SSLMs to clear statistics:

Step 6 Issue the **show ssl-proxy service** command on both SSLMs to verify all proxy services are operational.

Step 7 Generate HTTPS & FTP traffic to vservers VIP1 and SSL30.

Step 8 Issue the following commands on the active CSM to verify traffic flow and to determine which reals connections are being stuck to:

show mod csm 2 vservers name VIP1 detail

show mod csm 2 vservers name SSL30 detail

show mod csm 2 vservers name VIP30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 real sfarm FARM1 detail

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 9 Issue the following commands on the standby CSM to verify connection information and sticky information has been replicated. Verify that the standby CSM is not load balancing any connections:

show mod csm 2 vservers name VIP1 detail

show mod csm 2 vservers name SSL30 detail

show mod csm 2 vservers name VIP30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 real sfarm FARM1 detail

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 10 Issue the **show ssl-proxy stats service** command on all SSLSMs to verify the cons completed counter has incremented and that there are no handshake failures.

Step 11 Issue the **hw-module module 2 reset** command to rest the active CSM in slot 2.

Step 12 Issue the **show mod csm 2 ft** command on the standby to verify it is now the active CSM.

Step 13 Issue the following commands on the new active CSM to verify traffic flow and to determine if connections remained stuck to the same real:

show mod csm 2 vservers name VIP1 detail

show mod csm 2 vservers name SSL30 detail

show mod csm 2 vservers name VIP30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 real sfarm FARM1 detail

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 14 Issue the **show ssl-proxy stats service** command to verify the cons completed counter has incremented and that there are no handshake failures.

Step 15 When the reloaded CSM comes back online issue the **show mod csm 2 ft** command to verify it has preempted and is now the active CSM.

Step 16 Issue the following commands on the new active CSM to verify traffic flow and to determine if connections remained stuck to the same real:

show mod csm 2 vservers name VIP1 detail

show mod csm 2 vservers name SSL30 detail

show mod csm 2 vservers name VIP30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 real sfarm FARM1 detail

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 17 Issue the following commands on the standby CSM to verify connection information and sticky information has been replicated:

show mod csm 2 vservers name VIP1 detail

show mod csm 2 vservers name SSL30 detail

show mod csm 2 vservers name VIP30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 real sfarm FARM1 detail

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 18 Issue the **show ssl-proxy stats service** command to verify the cons completed counter has incremented and that there are no handshake failures.

Step 19 Issue the **show log** command on the FSWM to check for any errors.

Step 20 Stop background scripts to collect final status of network devices and analyze for error.

Step 21 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the CSM to replicate connections hitting the vserver.
- We expect the standby to become active and service the persistent replicated connection.
- We expect the CSM to preempt after a failure.
- We expect sticky connections to remain stuck to the same real after a failover.

Results

CSM Redundancy passed.

SSLM Reset

This test verified the effect of a SSL module reset on CSM load balancing. The CSM TCP Probe detects the module failure and stops load balancing traffic to it. The CSM continues to load balancing traffic to the remaining operational SSL Module. When the CSM TCP probe detects the SSLM Module is operational again it will start load balance traffic to it.

Relevant CSM Configuration

```
real DMZ1-SRV1
  address 192.168.100.100
  inservice
real DMZ1-SRV2
  address 192.168.100.110
  inservice
real SSLM1
  address 172.16.100.100
  inservice
real SSLM2
  address 172.16.100.110
  inservice
!
 serverfarm DMZ1-CLEAR
  nat server
  no nat client
  real name DMZ1-SRV1 80
   inservice
  real name DMZ1-SRV2 80
   inservice
  probe TCP
!
  serverfarm SSLM-445
   nat server
   no nat client
   failaction purge
   real name SSLM1 445
    inservice
```

```
   real name SSLM2 445
    inservice
   probe TCP
!
 sticky 200 ssl timeout 30
!
vserver DMZ1-CLEAR
  virtual 172.16.100.200 tcp 445
  vlan 172
  serverfarm DMZ1-CLEAR
  replicate csrp sticky
  replicate csrp connection
  persistent rebalance
  inservice
!
 vserver DMZ1-HTTPS
  virtual 192.168.100.200 tcp https
  serverfarm SSLM-445
  sticky 30 group 200
  replicate csrp sticky
  replicate csrp connection
  persistent rebalance
  inservice
!
```

Relevant SSLM1 Configuration

```
ssl-proxy service dmz1-web
 virtual ipaddr 172.16.100.100 protocol tcp port 445
 virtual policy ssl session-cache
 server ipaddr 172.16.100.200 protocol tcp port 445
 certificate rsa general-purpose trustpoint vip200
 inservice
```

Relevant SSLM2 Configuration

```
ssl-proxy service dmz1-web
 virtual ipaddr 172.16.100.110 protocol tcp port 445
 virtual policy ssl session-cache
 server ipaddr 172.16.100.200 protocol tcp port 445
 certificate rsa general-purpose trustpoint vip200
 inservice
```

Test Procedure

The procedure used to perform the SSLM Reset test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the following commands to verify they have been cleared:

show mod csm 2 vservers name SSL30 detail

show mod csm 2 vservers name VIP30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 serverfarms name SSLSM detail

show mod csm 2 probe name SSL detail | include SSLSM

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 3 Issue the following commands on both SSLMs to verify the services are operational.

show ssl-proxy service BACKEND30

show ssl-proxy service CLEAR29

show ssl-proxy service SSL-backend

Step 4 Issue the following commands on both SSLSMs to clear ssl-proxy service stats.

clear ssl-proxy stats service CLEAR29

clear ssl-proxy stats service BACKEND30

clear ssl-proxy stats service SSL-backend

Step 5 Issue the following commands to verify they have been cleared.

show ssl-proxy stats service CLEAR29

show ssl-proxy stats service BACKEND30

show ssl-proxy stats service SSL-backend

Step 6 From outside client initiate long lived HTTPS flow to vserver VIP30

Step 7 Issue the following commands on the active CSM to verify vservers SSL29, SSL30, VIP30 and VIP29 have open connections.

show mod csm 2 vservers name SSL30 detail

show mod csm 2 vservers name VIP30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 serverfarms name SSLSM detail

show mod csm 2 probe name SSL detail | include SSLSM

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 8 Issue the following commands on both SSLSMs to verify the conns attempted and conns completed counter has incremented and there are no errors:

show ssl-proxy stats service BACKEND30

show ssl-proxy stats service CLEAR29

show ssl-proxy stats service SSL-backend

Step 9 Issue the **hw-module module 3 reset** command on agg-1 reset SSLSM1.

Step 10 Monitor the client traffic when the CSM Probe dectects a failure it should reset one of the active connections.

Step 11 When you see the CSM log message indicating the probe failure send another HTTPS request from the client whose connections was reset.

Step 12 Issue the following commands on the active CSM to verify the TCP probe has failed real sslm1 and all traffic is being load balanced to sslm2:

1

show mod csm 2 vservers name SSL30 detail

show mod csm 2 vservers name VIP30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 serverfarms name SSLSM detail

sh mod csm 2 probe name SSL detail

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 13 Issue the following commands to verify the conns attempted and conns completed counter are still incrementing and there are no errors:

show ssl-proxy stats service BACKEND30

show ssl-proxy stats service CLEAR29

show ssl-proxy stats service SSL-backend

Step 14 After the SSLM becomes operational generate multiple HTTPS request to vserver SSL30.

Step 15 Issue the following commands to make sure traffic is being load balanced among the four SSLMs:

show mod csm 2 vservers name SSL30 detail

show mod csm 2 vservers name VIP30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 serverfarms name SSLSM detail

sh mod csm 2 probe name SSL detail

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 16 Stop background scripts to collect final status of network devices and analyze for error.

Step 17 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the CSM TCP probe to detect the SSLM failure.
- We expect the CSM to reset open connections when a probe fails a real.
- We expect the CSM to properly load balance traffic during a real failure.

Results

SSLM Reset passed.

Sorry Server

This test verified the CSM's ability to switchover incoming connections to a backup server farm, should the primary server farm become unavailable.

Relevant CSM Configuration

```
probe FORCED-FAIL http
 request method get url /notthere.html
 expect status 200 299
 interval 10
 retries 2
 failed 5
 open 3
 receive 5
!
serverfarm SORRY
  nat server
  no nat client
  predictor leastconns
  real name P1_ETH1.3001
   inservice
  real name P1_ETH1.3002
   inservice
!
serverfarm SORRY-BACK
  nat server
  no nat client
  predictor leastconns
  real name P1_ETH1.3003
   inservice
  real name P1_ETH1.3004
   inservice
!
vserver SORRY
 virtual 101.40.30.240 tcp www
 serverfarm SORRY backup SORRY-BACK
 persistent rebalance
 inservice
```

Test Procedure

The procedure used to perform the Sorry Server test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify that all real servers in each of the primary and backup serverfarms are operational.

show mod csm 2 reals sfarm sorry det

show mod csm 2 reals sfarm sorry-back det

Step 3 Clear the CSM counters

clear mod csm 2 counters

Step 4 Show the counters to make sure they are clear.

sh mod csm 2 serverfarms name SORRY-BACK det

sh mod csm 2 serverfarms name SORRY det

show mod csm 2 reals sfarm sorry det

show mod csm 2 reals sfarm sorry-back det

Step 5 Send 10 HTTPv1.0 GET requests to the vserver SORRY.

Step 6 Verify that the connections are being made to the primary serverfarm and not to the backup serverfarm.

show mod csm 2 vserver name sorry det

show mod csm 2 reals sfarm sorry det

show mod csm 2 reals sfarm sorry-back det

Step 7 Configure the probe FORCED-FAIL on the primary serverfarm, forcing all of its real servers to fail. probe forced-fail

Step 8 Send 10 HTTPv1.0 GET requests to the vserver SORRY.

Step 9 Verify that the real servers in serverfarm SORRY have failed and that the serverfarm SORRY-BACK is now hosting the new connections

show mod csm 2 vserver name sorry det

show mod csm 2 reals sfarm sorry det

show mod csm 2 reals sfarm sorry-back det

Step 10 Remove the probe FORCED-FAIL from the configuration of serverfarm SORRY, no probe forced-fail

Step 11 Verify that the real servers within serverfarm SORRY are again operational and accepting incoming connections, while the real servers within the backup serverfarm are no longer hosting incoming connections

show mod csm 2 vserver name sorry det

show mod csm 2 reals sfarm sorry det

show mod csm 2 reals sfarm sorry-back det

Step 12 Stop background scripts to collect final status of network devices and analyze for error.

Step 13 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect traffic to be forwarded to a backup server farm only when the probes fail on the primary server farm.

Results

Sorry Server passed with exception CSCdz22187, CSCsg97985.

HSRP Failover

This test verified HSRP failover when a system failure occurred. This test also verified that the HSRP preempt command worked when the system returned to an operational state, if the interface was configured with a higher priority than the current active router interface in the same HSRP group. HTTPS traffic was sent through a FWSM and load balanced via CSM and SSLM.

Test Procedure

The procedure used to perform the HSRP Failover test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the **show standby brief** command on both 6500s to verify agg1 is active for all VLANs.

Step 3 Issue the **clear mod csm 2 count** command on the active CSM to clear csm counters. Issue the following commands to verify they have been cleared and to verify state: show mod csm 2 vservers name VIP1 detail show mod csm 2 vservers name SSL30 detail show mod csm 2 vservers name VIP30 detail show mod csm 2 vservers name SSL29 detail show mod csm 2 vservers name VIP29 detail show mod csm 2 real sfarm SSLSM detail show mod csm 2 real sfarm FARM30 detail show mod csm 2 real sfarm FARM29 detail show mod csm 2 real sfarm FARM1 detail show mod csm 2 stats

Step 4 Issue the **show ssl-proxy service** command on all SSLSMs to verify the services are operational.

Step 5 Issue the **clear ssl-proxy stats service** command on all four SSLSMs to clear ssl-proxy service stats. Issue the **show ssl-proxy stats service** command to verify they have been cleared. Please note some counters might have incremented due to CSM probes.

Step 6 From outside client initiate HTTPS traffic to vserver VIP29 and VIP30.

Step 7 Issue the following commands on the active CSM to verify vservers SSL29, VIP29, SSL29 and VIP30 have open connections. show mod csm 2 vservers name VIP1 detail show mod csm 2 vservers name SSL30 detail show mod csm 2 vservers name VIP30 detail show mod csm 2 vservers name SSL29 detail show mod csm 2 vservers name VIP29 detail show mod csm 2 real sfarm SSLSM detail show mod csm 2 real sfarm FARM30 detail show mod csm 2 real sfarm FARM29 detail show mod csm 2 real sfarm FARM1 detail show mod csm 2 stats

Step 8 Issue the following commands on all four SSLSMs to verify the conns attempted and conns completed counter has incremented and there are no errors.

show ssl-proxy stats service

show ssl-proxy stats ssl

Step 9 Issue the **reload** command on agg1 to force a failover.

Step 10 Issue the **show standby brief** command on agg22 to verify its now active.

Step 11 Issue the following commands on the active CSM to verify vservers SSL29, VIP29, SSL29 and VIP30 have open connections. show mod csm 2 vservers name VIP1 detail show mod csm 2 vservers name SSL30 detail show mod csm 2 vservers name VIP30 detail show mod csm 2 vservers name SSL29 detail show mod csm 2 vservers name VIP29 detail show mod csm 2 real sfarm SSLSM detail show mod csm 2 real sfarm FARM30 detail show mod csm 2 real sfarm FARM29 detail show mod csm 2 real sfarm FARM1 detail show mod csm 2 stats

Step 12 Issue the following commands on both SSLSMs in agg2 to verify the conns attempted and conns completed counter are still incrementing and there are no errors.

show ssl-proxy stats service

show ssl-proxy stats ssl

Step 13 When agg1 becomes operational again issue the **show standby brief** command to verify it preempts and becomes active.

Step 14 Stop background scripts to collect final status of network devices and analyze for error.

Step 15 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the mean failover time for HSRP to be less then the default dead time of 10 seconds.
- We expect that when the failed system becomes operational again, it will resume HSRP active status and forward traffic.

Results

HSRP Failover passed.

Service Switch Bundle Testing

The following tests verified various functional aspects of the three Service Modules (CSM, FWSM and SSLSM) in a bundled fashion; that is, working together in the same chassis to provide services to data center traffic. Three Service Modules are bundled together in a pair of switches dedicated to providing services, separate from the Aggregation Layer switches.

The following test features were conducted:

- CSM/FWSM Integration, page 3-44
- CSM/SSLSM Integration, page 3-56
- Redundancy, page 3-65

CSM/FWSM Integration

CSM/FWSM integration looks at interoperability capacities of the CSM and FWSM, in terms of how they work together to handle data traffic.

The following tests were performed:

- Active FTP Through FWSM and CSM, page 3-44
- Passive FTP Through FWSM and CSM, page 3-47
- ICMP to a CSM Layer 3 and Layer 4 Vserver, page 3-49
- DNS Query Through CSM and FWSM, page 3-51
- FWSM CSM Layer4 SYN Attack, page 3-53
- Idle Timeout UDP, page 3-55

Active FTP Through FWSM and CSM

This test verified that the FWSM and CSM properly handled active FTP traffic when the **ftp fixup protocol 21** was enabled and disabled on the FWSM. FTP traffic was sent from an outside client to vserver VIP-ACTIVE-FTP and from an inside client to an outside server.

Relevant CSM Configuration

```
real P1_ETH1.3001
 address 101.1.1.11
 location dca-penguin-1
 inservice
real P1_ETH1.3002
```

```
  address 101.1.1.12
  location dca-penguin-1
  inservice
!
 serverfarm FARM1-A
  nat server
  no nat client
  predictor leastconns
  real name P1_ETH1.3001
   inservice
  real name P1_ETH1.3002
   inservice
!
 vserver VIP-ACTIVE-FTP
  virtual 101.40.1.251 tcp ftp
  vlan 301
  serverfarm FARM1-A
  advertise active
  replicate csrp connection
  persistent rebalance
  inservice
```

Test Procedure

The procedure used to perform the Active FTP Through FWSM and CSM test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the following commands on the FWSM to clear connections and counters in the VLAN 1101-2101 context:

```
change context Vlan1101-2101
clear xlate
clear conn
clear log
conf t
clear access-list ACL-in count
```

Step 3 Issue the following commands on the FWSM to verify connections and counters have been cleared:

change context VLAN 1101-2101

show xlate

show conn

show access-list ACL-in

Step 4 Issue the following commands on the Active CSM to clear connections and counters:

clear mod csm 2 counters

clear mod csm 2 connections

Step 5 Issue the following commands on the Active CSM to verify the counters have been cleared:

show mod csm 2 vserver name vip-active-ftp detail

show mod csm 2 real sfarm farm1-a detail

show mod csm 2 stats

show mod csm 2 conns

Step 6 Send ACTIVE FTP traffic to vserver VIP-ACTIVE-FTP from an outside client.

Step 7 Issue the following command on the FWSM to verify the FTP control and data channels were successfully created:

change context VLAN 1101-2101

show xlate

show conn

show log

Step 8 Issue the **show mod csm 2 conns** command to verify the FTP control and data connections have been established.

Step 9 When the FTP traffic has completed issue the following command on the FWSM to verify a match on the correct access-list:

show access-list ACL-in | include extended permit tcp any 201.1.1.0 255.255.255.0 eq ftp

Step 10 Issue the following command on the Active CSM to verify the FTP traffic was properly load balanced:

show mod csm 2 vserver name vip-active-ftp detail

show mod csm 2 real sfarm farm1-a detail

show mod csm 2 stats

Step 11 On the FWSM context VLAN 1101-2101, configure **no fixup protocol ftp 21**.

The **fixup protocol ftp 21** configuration is part of the default configuration for the DCAP test topology.

Step 12 Send an active FTP request from an inside client to an outside server.

This connection should fail. When **no fixup protocol ftp 21** has been configured only passive mode ftp is allowed from an inside interface to an outside interface.

Step 13 Issue the following command on the FWSM to verify the FTP data channel was not successfully created:

change context VLAN 1101-2101

show xlate

show conn

show log

Step 14 Reconfigure **fixup protocol ftp 21** on the VLAN 1101-2101 context to enable the fixup protocol for FTP on port 21 and use the **show fixup protocol ftp** command to verify it is now been enabled.

Step 15 Issue the following commands on the FWSM to clear connections and counters:

```
change context Vlan1101-2101
clear xlate
clear conn
conf t
clear access-list ACL-in count
clear access-list outgoing count
clear log
```

Step 16 Send active FTP traffic to vserver VIP-ACTIVE-FTP from an outside client.

Step 17 Issue the following command on the FWSM to verify the FTP control and data channels were successfully created:

change context Vlan1101-2101

show xlate

show conn

show log

Step 18 Issue the **show mod csm 2 conns** command to verify the FTP control and data connections have been established.

Step 19 Stop background scripts to collect final status of network devices and analyze for error.

Step 20 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the FWSM to permit Active FTP on the Outside Interface.
- We expect the FWSM would deny Active FTP on the Inside to Outside interface when fixup protocol ftp 21 is disabled.
- We expect the CSM vserver to properly load balance Active FTP.

Results

Active FTP Through FWSM and CSM passed.

Passive FTP Through FWSM and CSM

This test verified that the FWSM and CSM properly handled PASSIVE FTP traffic when the FTP fixup was enabled and disabled on the FWSM. FTP traffic was sent from outside client to vserver VIP-PASSIVE-FTP with FTP fixup enabled on the FWSM and when it was disabled. The same was done for FTP GET requests coming from an inside client to an outside server.

Relevant CSM Configuration

```
!
module csm 2
 real P1_ETH1.3001
  address 101.1.1.11
  location dca-penguin-1
  inservice
 real P1_ETH1.3002
  address 101.1.1.12
  location dca-penguin-1
  inservice
!
 serverfarm FARM1-A
  nat server
  no nat client
  predictor leastconns
  real name P1_ETH1.3001
   inservice
  real name P1_ETH1.3002
   inservice
!
 vserver VIP-PASSIVE-FTP
  virtual 101.40.1.252 tcp ftp service ftp
  vlan 301
  serverfarm FARM1-A
  advertise active
  replicate csrp connection
  persistent rebalance
  inservice
```

```
!
```

Relevant FWSM configuration (context VLAN 1101-2101)

```
!
fixup protocol ftp
!
```

Test Procedure

The procedure used to perform the Passive FTP Through FWSM and CSM test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 On dcb-ss-1, using the **show module csm 2 vserver name vip-passive-ftp detail** command, verify that the CSM vserver VIP-PASSIVE-FTP is configured for "service ftp" and that it is pointing to the serverfarm FARM1.

The output of the command shows that the serverfarm that is being used is **FARM1** and that **service = ftp** is enabled.

Step 3 Using the **show module csm 2 serverfarm name farm1-a detail** command, verify that there are two real servers in serverfarm FARM1-A and that they are both OPERATIONAL.

Step 4 On the active FWSM, in context VLAN 1101-2101, use the **show fixup** command to verify that **fixup protocol ftp 21** is not configured. If it is configured, use the **no fixup protocol ftp** command to disable it.

Step 5 From an outside client, send a single passive FTP GET to vserver VIP-PASSIVE-FTP and verify that it fails.

The connection fails because the **fixup protocol ftp** has been disabled on the active FWSM.

Step 6 Send a single passive FTP request from an inside client to the outside server.

This connection should succeed. When FTP fixups have been disabled, only passive mode FTP is allowed from an inside interface to an outside interface (active FTP is disallowed).

Step 7 Configure **fixup protocol ftp 21** on the active FWSM context VLAN 1101-2101 to enable the fixup protocol for FTP on port 21.

Step 8 Issue the following commands on the active FWSM context VLAN 1101-2101 to clear connections and counters:

clear xlate

clear conn

clear log

Step 9 Issue the following commands on the active CSM to clear connections and counters:

clear module csm 2 counters

clear module csm 2 connections

Step 10 Send a single passive FTP GET request for a very large file from an outside client to the CSM vserver VIP-PASSIVE-FTP.

The target file, "1G_file.zip" is 1-Gigabyte in size.

Step 11 While the GET is underway, issue the following commands on the active FWSM context VLAN 1101-2101 to verify the FTP Control and Data Channels were successfully created:

show conn

show xlate

show log

Step 12 While the GET is underway, issue the **show module csm 2 conn** command to verify the FTP control and data connections have been established.

Step 13 Send 20 passive FTP GETs from an outside client to the CSM vserver VIP-PASSIVE-FTP.

Each of these should succeed.

Step 14 On the active CSM, use the **show module csm 2 real sfarm farm1-a detail** to verify that the previous GET requests have been load-balanced evenly across both servers in serverfarm FARM1-A.

Each real server listed in the output should show about the same number of "total connections established".

Step 15 Send a single passive FTP request from inside client to the outside server.

This connection should succeed.

Step 16 Stop background scripts to collect final status of network devices and analyze for error.

Step 17 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the FWSM to permit Passive FTP on the Inside Interface.
- We expect the FWSM will deny Passive FTP on the Outside interface when fixup protocol ftp 21 is disabled.
- We expect the CSM vserver to properly load balance Active FTP.
- We expect no unacceptable impact on CPU or memory.

Results

Passive FTP Through FWSM and CSM passed.

ICMP to a CSM Layer 3 and Layer 4 Vserver

This test verified ICMP PING traffic to Mulitple Layer 4 vservers and a Layer 3 vserver all configured with the same virtual IP address. The CSM virtual address is located on the outside of the FWSM and the CSM reals or located on the inside of the CSM.

Relevant CSM Configuration

```
!
 vserver DMZ1-FTP
  virtual 201.40.40.240 tcp ftp service ftp
  vlan 301
  serverfarm FARM1
  advertise active
  persistent rebalance
  inservice
!
vserver VIP-DNS
  virtual 201.40.40.240 udp dns service per-packet
  vlan 301
  serverfarm FARM1
```

```
  advertise active
  persistent rebalance
  inservice
!
 vserver VIP-L3
  virtual 201.40.40.240 any
  vlan 301
  serverfarm FARM1-A
  advertise active
  persistent rebalance
  inservice
!
 vserver VIP-WWW
  virtual 201.40.40.240 tcp www
  vlan 301
  serverfarm FARM1-A
  advertise active
  persistent rebalance
  inservice
!
```

Test Procedure

The procedure used to perform the ICMP to a CSM Layer 3 and Layer 4 Vserver test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the **clear module csm 2 counters** command to clear all CSM statistics.

Step 3 Issue the following commands on the Active FWSM in context VLAN 1101-2101 to clear connections and counters.

clear xlate

clear conn

clear log

Step 4 Suspend CSM vserver VIP-L3 with the **no inservice** command.

Step 5 From an outside linux client send ICMP PING to CSM vserver VIP-WWW. This ping should be successful.

Step 6 On the Active FWSM issue the **show xlate** command. You should see zero global entries because only Layer 3 vservers load balance pings to reals.

Step 7 Verify the following vservers have not recorded any policy matches or packets received by issuing the following commands.

show module csm 2 vservers name DMZ1-FTP detail

show module csm 2 vservers name vip-dns detail

show module csm 2 vservers name vip-www detail

show module csm 2 vservers name vip-l3 detail

Step 8 Enable CSM vserver VIP-L3 with the **inservice** command and verify its now operational with the **show module csm 2 vserver vip-l3 detail** command.

Step 9 From an outside linux client send ICMP PING to CSM vserver VIP-L3. This ping should be successful.

Step 10 On the Active FWSM issue the **show xlate** command. You should see a global entry for each real in the serverfarm because only Layer 3 vservers load balance pings request to reals.

Step 11 Verify only vserver VIP-L3 has recorded policy match and packets received by issuing the following commands.

show module csm 2 vservers name DMZ1-FTP detail

show module csm 2 vservers name vip-dns detail

show module csm 2 vservers name vip-www detail

show module csm 2 vservers name vip-l3 detail

Step 12 Suspend the following vservers with the **no inservice** command: DMZ1-FTP, VIP-DNS, VIP-WWW, and VIP-L3.

Step 13 From an outside linux client send ICMP PING to CSM vserver VIP-WWW. This ping should be unsuccessful because all four vserver configured with the same virtual IP have been taken out of service.

Step 14 Enable the following vservers with the **inservice** command

Vsever DMZ1-FTP

VIP-DNS

VIP-WWW

VIP-L3

Step 15 Stop background scripts to collect final status of network devices and analyze for error.

Step 16 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the CSM NOT to Load Balance ICMP for a layer 4 vserver.
- We expect the CSM to Load Balance ICMP for a layer 3 vserver.
- We expect the FWSM to create a connection for ICMP when fixup protocol icmp is configured.
- We expect vservers to respond to ICMP when operational.
- We expect a vserver not to respond to ICMP when not operational.

Results

ICMP to a CSM Layer 3 and Layer 4 Vserver passed.

DNS Query Through CSM and FWSM

This test verified that the FWSM and CSM properly handled DNS traffic when fixup protocol DNS was enabled. In this Topology the CSM virtual is on the outside of the FWSM and the reals are on the inside of the FWSM. Spirent Avalanche is used to generate client and server traffic.

Relevant CSM Configuration

```
vserver VIP-DNS
  virtual 201.40.40.240 udp dns service per-packet
  vlan 301
  serverfarm FARM1
  advertise active
  persistent rebalance
  inservice
```

Relevant FWSM Access-list

```
access-list ACL-in extended permit udp any 201.1.1.0 255.255.255.0
```

Test Procedure

The procedure used to perform the DNS Query Through CSM and FWSM test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the following commands on the Active FWSM in context VLAN 1101-2101 to clear connections and counters:

clear xlate

clear conn

clear access-list ACL-in count

clear log

Step 3 Issue the **clear module csm 2 counters** and **clear module csm 2 connections** commands on the Active CSM to clear connections and counters.

Step 4 Use STT/TUB to send a DNS querie to vserver VIP-DNS for domain name www.datacenter.com.

Step 5 Issue the **show xlate** command on the Active FWSM to verify that a global entry was created for each real in serverfarm FARM1.

Step 6 Issue the **show show access-list ACL-in** command to verify there are matches on the portion of the access-list that permits UDP DNS queries.

The ACL line that permits this traffic is:

access-list ACL-in extended permit udp any 201.1.1.0 255.255.255.0

Step 7 Issue the following commands on the Active CSM to verify the DNS traffic was properly load balanced:

show module csm 2 vserver name vip-dns detail

show module csm 2 stats

The "total conns" should approximate the number of hits that was seen on the FWSM access-list.

Step 8 Issue the **show module csm 2 real sfarm FARM1 detail** command to verify that each real server in the serverfarm has made some connections.

Step 9 Issue the **clear module csm 2 counters** and **clear module csm 2 connections** commands on the Active CSM to clear connections and counters.

Step 10 Use Spirent Avalanche to send DNS quires to vserver VIP-DNS for domain name www.datacenter.com at rate of one thousand users per second.

Step 11 While traffic is running issue the **show xlate** and **show conn | include most** commands on the VLAN 1101-2101 FWSM context to verify the xlate table and number of open connections.

You should see 65 global entries in the xlate table.

Step 12 Verify the number of attempts and number of failures on the Spirent Avalanche run tab.

Step 13 Verify the DNS connections rate on the Spirent Avalanche client stats tab and the select DNS. DNS quires per second should be around 1,000.

Step 14 Issue the following commands on the Active CSM to verify the DNS traffic was properly load balanced. Counters Tot matches and L4 Load-Balanced Decisions should have the same value. Verify the Tot matches counter equals the number of attempts on the Spirent Avalanche run tab.

show module csm 2 vserver name vip-dns detail

show module csm 2 stats

Step 15 Issue the **clear module csm 2 counters** and **clear module csm 2 connections** commands on the Active CSM to clear connections and counters.

Step 16 Use Spirent Avalanche to send DNS quires to vserver VIP-DNS for domain name www.datacenter.com at rate of Fifteen hundred users per second. At this rate we expect a large number of UDP request to fail.

Step 17 While traffic is running issue the following two commands to verify the xlate table and number of open connections. You should see 65 global entries in the xlate table.

show xlate

show conn | in most

Step 18 Verify the number of attempts and number of failures on the Spirent Avalanche run tab.

Step 19 Verify the DNS connections rate on the Spirent Avalanche client stats tab and the select DNS. DNS quires per second should be around 1,500.

Step 20 Issue the following commands on the Active CSM to verify the DNS traffic was properly load balanced. Counters Tot matches and L4 Load-Balanced Decisions should have the same value. Verify the Tot matches counter equals the number of attempts on the Spirent Avalanche run tab.

show mod csm 2 vserver name vip-dns detail

show mod csm 2 stats

Step 21 Stop background scripts to collect final status of network devices and analyze for error.

Step 22 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the FWSM to permit DNS traffic to vserver VIP-DNS.
- We expect the FWSM to NAT the DNS response to the outside client when fixup protocol DNS is enabled.
- We expect the FWSM not to NAT the DNS response to the inside client when fixup protocol DNS is enabled.
- We expect the CSM vserver to properly load balance DNS traffic.

Results

DNS Query Through CSM and FWSM passed.

FWSM CSM Layer4 SYN Attack

Syn-flood attacks aim at preventing a TCP/IP server from servicing request. The SYN flag is set in a TCP segment when a connection request is sent by a computer. The target server responds back with an ACK and waits for a response from the initiator. The SYN-Flood attacker spoofs the source IP address so that the server never receives a response to the ACK. This causes the server to use up resources overloading the server and preventing it from responding to legitimate connection request.

TCP Intercept protects inside systems from a DoS attack perpetrated by flooding an interface with TCP SYN packets. Enable this feature by setting the maximum embryonic connections option of the nat and static commands.

When the embryonic limit has been surpassed, the TCP intercept feature intercepts TCP SYN packets from clients to servers on a higher security level. The TCP intercept feature implements software to protect TCP servers from TCP SYN-flooding attacks, which are a type of denial-of-service attack. SYN cookies are used during the validation process and help to minimize the amount of valid traffic being dropped.

The embryonic limit is a feature that is enabled for any inbound connection (a connection that the FWSM considers from lower to higher security). In order for a connection to be inbound, you need to either hit a static or a global xlate.

This test verified the TCP Intercept feature by sending 1 millon SYN packets generated on a linux server using random source ip address. The SYN packets were sent to a CSM layer 4 server with 65 reals behind the FWSM.

Relevant CSM Configuration

```
vserver VIP-WWW
  virtual 201.40.40.240 tcp www
  vlan 301
  serverfarm FARM1
  advertise active
  persistent rebalance
  inservice
```

Test Procedure

The procedure used to perform the FWSM CSM Layer4 SYN Attack test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Configure context VLAN 1101-2101 on the Active FWSM with the following static command to enable the limitation of embryonic connections to 20.

static (inside,outside) 201.1.1.0 201.1.1.0 netmask 255.255.255.0 tcp 0 20

Step 3 Issue the **clear module csm 2 counters** command to clear all CSM statistics and **clear module csm 2 conn** to clear all connections.

Step 4 Verify CSM utilization by issuing the **show module csm 2 tech-support utilization** command.

Step 5 On the FWSM system context, clear the Fast Path Syn Cookie Statistics Counters for NP-1 and NP-2 with the **clear np 1 syn** and **clear np 2 syn** commands.

Step 6 Verify CPU and memory utilization on the FWSM by issuing the **show cpu** and **show memory** commands from the system context.

Step 7 Clear all the translations on the FWSM by issuing the **clear xlate** command on the active FWSM in Vlan context

Step 8 From the outside client send 1,000,000 syn packets to vserver VIP-WWW with random source ip addresses.

Step 9 While the syn attack traffic is being sent verify the rate of the syn attack on the FWSM by issuing the **show perfmon | inc TCP Intercept** command. Issue the command several times to obtain a good baseline.

Step 10 While syn attack traffic is being sent verify CSM utilization by issuing the **show module csm 2 tech-support utilization** command.

Step 11 Verify there are no errors on the CSM by issuing the following commands.

show mod csm 2 vserver name vip-www detail

show mod csm 2 reals sfarm farm1-a det

show mod csm 2 stats

Step 12 Verify the FWSM is issued a syn cookie and verify the number of syn packets intercepted by issueing the following commands.

show np 1 syn

show np 2 syn

Step 13 Verify FWSM CPU and memory utilization were not adversely impacted by the syn attack by issuing the **show cpu** and **show memory** commands.

Step 14 Verify the FWSM log contains message number FWSM-6-106015 by issuing the **show log** command in context VLAN 1101-2101.

Step 15 Remove static statement from VLAN 1101-2101 on the Active FWSM with the following command.

no static (inside,outside) 201.1.1.0 201.1.1.0 netmask 255.255.255.0 tcp 0 20

Step 16 Stop background scripts to collect final status of network devices and analyze for error.

Step 17 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the FWSM to intercept syn packets being sent to the CSM reals by issuing a syn cookie.

  We except CPU and memory utilization on the CSM and FWSM not to be adversely impacted by the syn attack.
- We expect the CSM to evenly load balance packets across all reals in the serverfarm.

Results

FWSM CSM Layer4 SYN Attack passed.

Idle Timeout UDP

The test verified the CSM will remove idle UDP connections at 60 seconds and the FWSM will remove them after two minutes. It also verified that the CSM load-balanced the UDP connections.

The CSM vserver VIP-TFTP has been configured with a 60 second idle timer. A TFTP copy request (UDP port 69) was generated on a Linux client, to the VIP-TFTP, to create a connection on the CSM and FWSM. It was verified that these connections were load-balanced properly to the real servers in the serverfarm. It was then verified that these connections timed out after 60 seconds on the CSM and 2 minutes on the FWSM.

Relevant CSM Configuration

```
!
 vserver VIP-TFTP
  virtual 101.40.40.244 udp 0
  vlan 301
  serverfarm FARM1-A
  advertise active
  idle 60
  persistent rebalance
  inservice
```

Test Procedure

The procedure used to perform the Idle Timeout UDP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify CSM vserver VIP-TFTP is operational and the idle time out is set to 60 by issuing the **show mod csm 2 vserver name vip-tftp detail** command.

Step 3 Verify all reals are operational for CSM serverfarm FARM1-A by issuing the **show mod csm 2 real sfarm farm1-a detail** command.

Step 4 Clear all counters and connections on the CSM by issuing the **clear mod csm 2 counters** and **clear mod csm 2 conn** commands.

Step 5 On the active CSM, use the **show mod csm 2 serverfarm name farm1-a detail** command to verify that UDP connections have been created.

Step 6 On the active CSM, use the **show mod csm 2 conn vserver vip-tftp** command to verify that UDP connections have been created for the TFTP transfer.

Step 7 Use the **show clock** and **show mod csm 2 conn vserver vip-tftp** commands to verify that the UDP connections time out after one minute.

Step 8 Issue the **show timeout** command on the active FWSM in context VLAN 1101-2101 to verify timeout udp is set to two minutes.

Step 9 Issue the **clear conn** command on the Active FWSM in context VLAN 1101-2101 to clear connections.

Step 10 On the active FWSM, use the **show conn** command to verify that UDP connections have been created for the TFTP transfer.

Step 11 Use the **show clock** and **show conn** commands to verify that the UDP connections on the FWSM time out after two minutes.

Step 12 Stop background scripts to collect final status of network devices and analyze for error.

Step 13 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect flows that exceed the idle timeout will be cleared from both the CSM and the FWSM.
- We expect the CSM vserver to properly load balance UDP traffic.

Results

Idle Timeout UDP passed.

CSM/SSLSM Integration

CSM/SSLSM integration looks at interoperability capacities of the CSM and SSLSM, in terms of how they work together to handle data traffic.

The following tests were performed:

- Backend SSL, page 3-57

- SSL Sticky, page 3-61
- URL Rewrite, page 3-63

Backend SSL

This test verified that the CSM and SSLM successfully work together to load balance SSL traffic on the client side internally decrypted the traffic for advanced layer 7 processing then re-encrypt the traffic load balancing to the backend servers. This test also verified the CSM is able to stick clients to the same real based on cookies.

The CSM and SSLM communicate together on an internal VLAN in Routed Mode. The CSM communicates with the clients and reals in bridged mode. Clients access the CSM virtual addresses through static NAT mappings on the FWSM.

Relevant CSM Configuration

```
Vlan 100 client
  description DMZ1
  ip address 192.168.100.4 255.255.255.0
  gateway 192.168.100.1
!
 vlan 200 server
  description DMZ1
  ip address 192.168.100.4 255.255.255.0
  alias 192.168.100.3 255.255.255.0
!
 vlan 172 server
  description SSLM
  ip address 172.16.100.2 255.255.255.0
  alias 172.16.100.1 255.255.255.0
!
 probe TCP tcp
  interval 5
  failed 2
!
 probe ICMP icmp
!
 map 1.GIF url
  match protocol http url /1.gif
!
 map 2.GIF url
 match protocol http url /2.gif
!
 real BACKEND-SSL33
  address 172.16.100.133
  inservice
 real BACKEND-SSL34
  address 172.16.100.134
  inservice
 real BACKEND-SSL35
  address 172.16.100.135
  inservice
 real BACKEND-SSL36
  address 172.16.100.136
  inservice
real FRONTEND-SSL1
  address 172.16.100.103
  inservice
 real FRONTEND-SSL2
  address 172.16.100.113
  inservice
```

```
!
 serverfarm BACKEND-SSL
  nat server
  no nat client
  real name BACKEND-SSL33
   inservice
  real name BACKEND-SSL34
   inservice
  real name BACKEND-SSL35
   inservice
  real name BACKEND-SSL36
   inservice
  probe TCP
!
 serverfarm FRONTEND-SSL
  nat server
  no nat client
  real name FRONTEND-SSL1
   inservice
  real name FRONTEND-SSL2
   inservice
  probe TCP
!
 serverfarm ROUTE
  nat server
  no nat client
  predictor forward
!
 sticky 3 cookie backend-ssl timeout 30
!
 policy 1.GIF
  url-map 1.GIF
  sticky-group 3
  serverfarm BACKEND-SSL
!
 policy 2.GIF
  url-map 2.GIF
  sticky-group 3
  serverfarm BACKEND-SSL
!
vserver BACKEND-SSL
  virtual 172.16.100.203 tcp www
  vlan 172
  serverfarm BACKEND-SSL
  sticky 30 group 3
  persistent rebalance
  slb-policy 1.GIF
  slb-policy 2.GIF
  inservice
!
 vserver FORWARD
  virtual 192.168.100.0 255.255.255.0 any
  vlan 172
  serverfarm ROUTE
  persistent rebalance
  inservice
!
 vserver FRONTEND-SSL
  virtual 192.168.100.203 tcp https
  serverfarm FRONTEND-SSL
  ssl-sticky offset 20 length 6
  sticky 30 group 30
  persistent rebalance
  inservice
```

```
!
```

Relevant SSLM1 Configuration

```
ssl-proxy policy http-header backend-ssl
 custom "backend-ssl:on"
 client-ip-port
!
ssl-proxy service backend-ssl33 client
 virtual ipaddr 172.16.100.133 protocol tcp port 80
 server ipaddr 192.168.100.100 protocol tcp port 443
 trusted-ca backend-ssl
 authenticate verify signature-only
 inservice
!
ssl-proxy service backend-ssl34 client
 virtual ipaddr 172.16.100.134 protocol tcp port 80
 server ipaddr 192.168.100.110 protocol tcp port 443
 trusted-ca backend-ssl
 authenticate verify signature-only
 inservice
!
ssl-proxy service frontend-ssl
 virtual ipaddr 172.16.100.103 protocol tcp port 443
 server ipaddr 172.16.100.203 protocol tcp port 80
 certificate rsa general-purpose trustpoint backend-ssl
 policy http-header backend-ssl
 inservice
!
ssl-proxy pool ca backend-ssl
 ca trustpoint bundle-root
!
crypto ca trustpoint backend-ssl
 enrollment terminal pem
 fqdn www.backend-ssl.com
 subject-name C=US, ST=Massachusetts, L=Boxborough, O=Cisco, OU=Safeharbor,
CN=www.backend-ssl.com
 rsakeypair backend-ssl
!
crypto ca trustpoint bundle-root
 enrollment terminal pem
!
```

Relevant SSLM2 Configuration

```
 ssl-proxy policy http-header backend-ssl
 custom "backend-ssl:on"
 client-ip-port
!
ssl-proxy service backend-ssl35 client
 virtual ipaddr 172.16.100.133 protocol tcp port 80
 server ipaddr 192.168.100.100 protocol tcp port 443
 trusted-ca backend-ssl
 authenticate verify signature-only
 inservice
!
ssl-proxy service backend-ssl36 client
 virtual ipaddr 172.16.100.134 protocol tcp port 80
 server ipaddr 192.168.100.110 protocol tcp port 443
 trusted-ca backend-ssl
 authenticate verify signature-only
 inservice
!
ssl-proxy service frontend-ssl
```

```
 virtual ipaddr 172.16.100.103 protocol tcp port 443
 server ipaddr 172.16.100.203 protocol tcp port 80
 certificate rsa general-purpose trustpoint backend-ssl
 policy http-header backend-ssl
 inservice
!
ssl-proxy pool ca backend-ssl
 ca trustpoint bundle-root
!
crypto ca trustpoint backend-ssl
 enrollment terminal pem
 fqdn www.backend-ssl.com
 subject-name C=US, ST=Massachusetts, L=Boxborough, O=Cisco, OU=Safeharbor,
CN=www.backend-ssl.com
 rsakeypair backend-ssl
!
crypto ca trustpoint bundle-root
 enrollment terminal pem
!
```

Test Procedure

The procedure used to perform the Backend SSL test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the following command on all 4 SSLMs to clear statistics:

clear ssl-proxy stats service

clear ssl-proxy stats hdr

clear ssl-proxy stats ssl

Step 3 Issue the following commands on all 4 SSLMs to verify statistics have been cleared:

show ssl-proxy stats service

show ssl-proxy stats hdr

show ssl-proxy stats ssl client

Step 4 Issue the **show ssl-proxy service** command on all four SSLSMs to verify ssl-proxy services are operational.

Step 5 Issue the **clear mod csm 2 counters** command on the active CSM to clear counters.

Step 6 Issue the following commands to verify the counters have been cleared:

show mod csm 2 vserver name SSL30 detail

show mod csm 2 vserver name VIP30 detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 stats

Step 7 Issue the **clear mod csm 2 sticky all** command on the active CSM to clear the sticky table.

Step 8 Issue the **show mod csm 2 sticky** command on the active CSM to verify the sticky table.

Step 9 Send multiple HTTPS get requests for 1.html, 2.html and 3.html from the outside client to vserver FRONTEND-SSL. The client emulation tool will generate the traffic using 3 different cookies.

Step 10 Wait until client emulation traffic has completed then issue the **show mod csm 2 vservers name SSL30 detail** command to verify the Tot matches counter equals 600.

Step 11 Issue the **show mod csm 2 vservers name VIP30 detail** command to verify the Tot matches counter has incremented for the following 3 policies:

100 times for 1.GIF

200 times for 2.GIF

300 times for (default)

Step 12 Issue the **show mod csm 2 real sfarm FARM30 detail** command on the active CSM to verify the load balancing of connections.

Step 13 Issue the **show mod csm 2 stats** command on the active CSM to verify there are no errors.

Step 14 Issue the **show mod csm 2 sticky** command on the active CSM to verify the sticky table.

Step 15 Issue the **show ssl-proxy stats service BACKEND30** command on all SSLMs to verify the following two counters equal 600:

conns attempted

conns completed

Step 16 Issue the following commands on sslm-1 to very conns attempted and conns completed counter have incremented and there are no errors.

show ssl-proxy stats service BACKEND30

show ssl-proxy stats service SSL-backend

Step 17 Stop background scripts to collect final status of network devices and analyze for error.

Step 18 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the CSM to correctly load balance SSL traffic.
- We expect the CSM to apply the correct L7 policy on clear text traffic.
- We expect the CSM to be able to stick based on the client cookie.
- We expect the SSLSM to re-encrypt the clear text traffic and forward through the CSM to the backend server.
- We expect the SSLSM to insert client IP and Port information
- We expect the SSLM to insert the customer header.

Results

Backend SSL passed.

SSL Sticky

This test verified the ability of the CSM to extract SSL Session ID and add a SSL entry to the sticky table. Subsequent SSL request containing the same SSL Session ID were sent to the same real server associated with that sticky entry. The real servers used in this test are SSL modules.

Test Procedure

The procedure used to perform the SSL Sticky test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the **clear mod csm 2 counters** command on the active CSM. Issue the following commands to verify the counters have been cleared:

show mod csm 2 vservers name SSL30 detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 stats

Step 3 Issue the **clear mod csm 2 sticky all** command on the active CSM to clear all sticky entries.

Step 4 Issue the **show mod csm 2 sticky** command to verify all ssl sticky entries have been cleared.

Step 5 Issue the **show ssl-proxy service** command on all four SSLMs to verify ssl-proxy service is operational.

Step 6 Issue the **clear ssl-proxy stats service** command on all four SSLMs to clear ssl-proxy service statistics.

Step 7 Issue the **show ssl-proxy stats service** command on all four SSLMs to verify statistics have been cleared.

Step 8 Begin initiating SSL GET requests to vserver SSL30. This involves a single user generating 240 HTTPS requests where a new SSL Session ID will be generated on every 30th request.

Step 9 Within 30 seconds after the traffic has started, issue the **show module csm 2 reals sfarm sslsm detail** command on the active CSM to verify that all of the connections up to this point are being sent ("stuck") to a single SSLSM "server."

The **total connections established** on one of the servers should be some value greater than 1 and less than 30. There should be no established connections on any of the other servers.

Step 10 When traffic has completed verify that connections were load balanced among the four SSLMs in serverfarm SSLMSM:

show mod csm 2 vservers name SSL30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 stats

Step 11 Use the **show module csm 2 sticky group 206** command on the active CSM to verify that the SSL sticky group has entries in it.

Step 12 Stop background scripts to collect final status of network devices and analyze for error.

Step 13 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the CSM to stick clients to the same real based on SSL Session ID.

Results

SSL Sticky passed.

URL Rewrite

This test verified that the SSLM properly manipulated the data coming from the server with the use of the URL rewrite functionality. Server data that contains a 300 series redirect will be rewritten to HTTPS being forwarded to the client.

HTTPS and HTTP traffic for this test is load balanced by a CSM.

IE, Firefox and a client emulator will be used to test basic SSL Termination and URL Rewrite

Note Under the current time constraints we are not able to test every possible browser/version that exists today. The browsers were carefully selected to show any inconsistencies in SSL termination.

Relevant CSM Configuration

```
!
 sticky 241 ssl timeout 30
!
 vserver CLEAR-REWRITE
  virtual 201.40.40.241 tcp www
  vlan 302
  serverfarm FARM1-A
  persistent rebalance
  inservice
!
 vserver SSL-REWRITE
  virtual 201.40.40.241 tcp https
  vlan 301
  serverfarm SSLSM
  advertise active
  sticky 30 group 241
  persistent rebalance
  inservice
```

Relevant SSLM1 Configuration

```
!
 policy url-rewrite rewrite-test
  url 201.40.40.241
  url www.urlrewrite.com
  url www.urlrewrite1.com
  url www.urlrewrite2.com
  url www.urlrewrite3.com
  url www.urlrewrite4.com
  url www.urlrewrite5.com
  url www.urlrewrite6.com
  url www.urlrewrite7.com
  url www.urlrewrite8.com
  url www.urlrewrite9.com
  url www.urlrewrite10.com
  url www.urlrewrite11.com
  url www.urlrewrite12.com
  url www.urlrewrite13.com
  url www.urlrewrite14.com
  url www.urlrewrite15.com
  url www.urlrewrite16.com
  url www.urlrewrite17.com
  url www.urlrewrite18.com
  url www.urlrewrite19.com
  url www.urlrewrite20.com
  url www.urlrewrite21.com
  url www.urlrewrite22.com
```

```
 url www.urlrewrite23.com
 url www.urlrewrite24.com
 url www.urlrewrite25.com
 url www.urlrewrite26.com
 url www.urlrewrite27.com
 url www.urlrewrite28.com
 url www.urlrewrite29.com
 url www.urlrewrite30.com
!
service url-rewrite
 virtual ipaddr 201.40.40.241 protocol tcp port 443 secondary
 server ipaddr 200.1.0.20 protocol tcp port 80
 certificate rsa general-purpose trustpoint url-rewrite
 no nat server
 policy url-rewrite rewrite-test
 inservice
!
```

Test Procedure

The procedure used to perform the URL Rewrite test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 From the outside client use the client emulator to generate an HTTPS request to vserver SSL-REWRITE.

Step 3 Clear ssl-proxy service statistics and url statistics on each of the four SSLSMs in the topology by issuing the following:

clear ssl-proxy stats service url-rewrite context Default

clear ssl-proxy stats url

Step 4 Verify the ssl-proxy service statistics and url statistics have been cleared on each of the four SSLSMs in the topology by issuing the following commands:

show ssl-proxy stats service url-rewrite context Default

show ssl-proxy stats url

Step 5 Issue the **clear mod csm 2 count** command on the active CSM to clear csm counters.

Step 6 From the outside client use the client emulator to generate 1000 HTTPS request to vserver SSL-REWRITE.

Step 7 When client emulated traffic has completed issue the **show ssl-proxy stats url** command on all SSLMs to verify the Rewrites Succeeded counter has incremented for a combined total of 1000.

Step 8 Issue the **show ssl-proxy stats service url-rewrite** command on all SSLMs to verify the conns attempted and full handshakes counters have incremented to 1000.

Step 9 On the Active CSM verify the total matches counter for vserver SSL-REWRITE and vserver CLEAR-REWRITE equals 2000 by issuing the command **show mod csm 2 vserver name name detail** command.

Step 10 On the Active CSM verify traffic was evenly load balanced between all reals in serverfarm SSLSM and serverfarm FARM30 by issuing the **show mod csm 2 real sfarm *name* detail** command.

Step 11 Stop background scripts to collect final status of network devices and analyze for error.

Step 12 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that the SSLM can rewrite the server issued 300 series redirects from HTTP to HTTPS.

Results

URL Rewrite passed.

Redundancy

The resiliency of network resources and services to hardware and software compent failures is key to a successful high availability strategy in a data center network. Redundancy measures the effects of various failure scenarios on Layer 4-7 services and hardware.

The following tests were performed:

- FWSM Redundancy, page 3-65
- CSM Redundancy, page 3-67
- SSLM Reset, page 3-70
- Sorry Server, page 3-73
- HSRP Failover, page 3-75

FWSM Redundancy

This test verified that long lived flows being load balanced by the CSM and traversing the FWSM will be replicated between the Primary and Secondary FWSM. The ability of the system to successfully replicate flows and forward traffic after the failover was the criteria for a successful test run.

Test Procedure

The procedure used to perform the FWSM Redundancy test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the following commands on the Primary and Secondary FWSM:

show xlate

show conn

Step 3 Issue the **show failover** command on the Primary and Secondary FSWM to verify the Primary FWSM is in Active state.

Step 4 Issue the **clear mod csm 2 count** command on the active CSM to clear counters.

Step 5 Issue the following commands to verify the counters have been cleared:

show mod csm 2 vservers name VIP29 detail

show mod csm 2 vservers name SSL29 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM29 detail

show mod csm 2 stats

show mod csm 2 conn

Step 6 Generate HTTPS traffic to vserver SSL29. Generate FTP traffic to vserver VIP1.

Step 7 Issue the following command on the Primary and Secondary FWSM to verify connections:

show xlate

show conn

Step 8 Issue the following commands on the active CSM to verify connections:

show mod csm 2 vservers name VIP29 detail

show mod csm 2 vservers name SSL29 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM29 detail

show mod csm 2 stats

show mod csm 2 conn

Step 9 Issue the **reload** command on the Primary FWSM to force a reload.

Step 10 Issue the **show failover** command on the Secondary FSWM to verify it's now Active.

Step 11 Issue the following commands on the Secondary FWSM to verify connections:

show xlate

show conn

Step 12 Issue the following commands on the active CSM several times to verify connections:

show mod csm 2 vservers name VIP29 detail

show mod csm 2 vservers name SSL29 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM29 detail

show mod csm 2 stats

show mod csm 2 conn

Step 13 When the Primary FWSM comes back on line issue the **show failover** command to verify it's in Standby state.

Step 14 Issue the following command on the Primary FWSM to verify connections have been replicated from the Secondary FWSM:

show xlate

show conn

Step 15 Issue the **reload** command on the Secondary FWSM to force a reload.

Step 16 Issue the **show failover** command on the Primary FSWM to verify it's now Active.

Step 17 Issue the following command on the Primary FWSM to verify connections:

show xlate

show conn

Step 18 Issue the following commands on the active CSM several times to verify connections:

show mod csm 2 vservers name VIP29 detail

show mod csm 2 vservers name SSL29 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM29 detail

show mod csm 2 stats

show mod csm 2 conn

Step 19 Wait for FTP traffic to complete and check for errors

Step 20 Stop background scripts to collect final status of network devices and analyze for error.

Step 21 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the FWSM to replicate flow information from active to standby FWSM.
- We expect the standby FWSM will transition to active state with the failure of the active FWSM.

Results

FWSM Redundancy passed.

CSM Redundancy

This test verified that flow information is replicated from the active CSM to the standby CSM. Upon a redundancy transition, the standby CSM will become the new active CSM and process all flows which were originally created on the active CSM.

Test Procedure

The procedure used to perform the CSM Redundancy test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the **clear mod csm 2 counters** command to clear CSM counters on the active and standby CSM.

Step 3 Issue the following commands on the active and standby CSM to verify the counters have been cleared:

show mod csm 2 vservers name VIP29 detail

show mod csm 2 vservers name SSL29 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM29 detail

show mod csm 2 stats

show mod csm 2 conn

Step 4 Issue the **clear mod csm 2 sticky all** command on the active and standby CSM. Issue the **show mod csm 2 sticky** command to verify the sticky table was cleared.

Step 5 Issue the **clear ssl-proxy stats service** command on all SSLMs to clear statistics:

Step 6 Issue the **show ssl-proxy service** command on both SSLMs to verify all proxy services are operational.

Step 7 Generate HTTPS & FTP traffic to vservers VIP1 and SSL29.

Step 8 Issue the following commands on the active CSM to verify traffic flow and to determine which reals connections are being stuck to:

show mod csm 2 vservers name VIP-PASSIVE-FTP detail

show mod csm 2 vservers name SSL29 detail

show mod csm 2 vservers name VIP29 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM29 detail

show mod csm 2 real sfarm FARM1-A detail

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 9 Issue the following commands on the standby CSM to verify connection information and sticky information has been replicated. Verify that the standby CSM is not load balancing any connections:

show mod csm 2 vservers name VIP-PASSIVE-FTP detail

show mod csm 2 vservers name SSL29 detail

show mod csm 2 vservers name VIP29 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM29 detail

show mod csm 2 real sfarm FARM1-A detail

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 10 Issue the **show ssl-proxy stats service** command on all SSLSMs to verify the cons completed counter has incremented and that there are no handshake failures.

Step 11 Issue the **hw-module module 2 reset** command to rest the active CSM in slot 2.

Step 12 Issue the **show mod csm 2 ft** command on the standby to verify it is now the active CSM.

Step 13 Issue the following commands on the new active CSM to verify traffic flow and to determine if connections remained stuck to the same real:

show mod csm 2 vservers name VIP-PASSIVE-FTP detail

show mod csm 2 vservers name VIP29 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM29 detail

show mod csm 2 real sfarm FARM1-A detail

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 14 Issue the **show ssl-proxy stats service** command to verify the cons completed counter has incremented and that there are no handshake failures.

Step 15 When the reloaded CSM comes back online issue the **show mod csm 2 ft** command to verify it has preempted and is now the active CSM.

Step 16 Issue the following commands on the new active CSM to verify traffic flow and to determine if connections remained stuck to the same real:

show mod csm 2 vservers name VIP-PASSIVE-FTP detail

show mod csm 2 vservers name SSL29 detail

show mod csm 2 vservers name VIP29 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM29 detail

show mod csm 2 real sfarm FARM1-A detail

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 17 Issue the following commands on the standby CSM to verify connection information and sticky information has been replicated:

show mod csm 2 vservers name VIP-PASSIVE-FTP detail

show mod csm 2 vservers name SSL29 detail

show mod csm 2 vservers name VIP29 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM29 detail

show mod csm 2 real sfarm FARM1-A detail

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 18 Issue the **show ssl-proxy stats service** command to verify the cons completed counter has incremented and that there are no handshake failures.

Step 19 Issue the **show log** command on the FSWM to check for any errors.

Step 20 Stop background scripts to collect final status of network devices and analyze for error.

Step 21 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the CSM to replicate connections hitting the vserver.
- We expect the standby to become active and service the persistent replicated connection.
- We expect the CSM to preempt after a failure.
- We expect sticky connections to remain stuck to the same real after a failover.

Results

CSM Redundancy passed.

SSLM Reset

This test verified the effect of a SSL module reset on CSM load balancing. The CSM TCP Probe detects the module failure and stops load balancing traffic to it. The CSM continues to load balancing traffic to the remaining operational SSL Module. When the CSM TCP probe detects the SSLM Module is operational again it will start load balance traffic to it.

Relevant CSM Configuration

```
real DMZ1-SRV1
  address 192.168.100.100
  inservice
real DMZ1-SRV2
  address 192.168.100.110
  inservice
real SSLM1
  address 172.16.100.100
  inservice
real SSLM2
  address 172.16.100.110
  inservice
!
 serverfarm DMZ1-CLEAR
  nat server
  no nat client
  real name DMZ1-SRV1 80
   inservice
  real name DMZ1-SRV2 80
   inservice
  probe TCP
!
  serverfarm SSLM-445
   nat server
   no nat client
   failaction purge
   real name SSLM1 445
    inservice
   real name SSLM2 445
    inservice
   probe TCP
!
 sticky 200 ssl timeout 30
!
vserver DMZ1-CLEAR
  virtual 172.16.100.200 tcp 445
  vlan 172
  serverfarm DMZ1-CLEAR
  replicate csrp sticky
  replicate csrp connection
  persistent rebalance
  inservice
!
 vserver DMZ1-HTTPS
  virtual 192.168.100.200 tcp https
  serverfarm SSLM-445
  sticky 30 group 200
  replicate csrp sticky
  replicate csrp connection
  persistent rebalance
  inservice
!
```

Relevant SSLM1 Configuration

```
ssl-proxy service dmz1-web
 virtual ipaddr 172.16.100.100 protocol tcp port 445
 virtual policy ssl session-cache
 server ipaddr 172.16.100.200 protocol tcp port 445
 certificate rsa general-purpose trustpoint vip200
 inservice
```

Relevant SSLM2 Configuration

```
ssl-proxy service dmz1-web
 virtual ipaddr 172.16.100.110 protocol tcp port 445
 virtual policy ssl session-cache
 server ipaddr 172.16.100.200 protocol tcp port 445
 certificate rsa general-purpose trustpoint vip200
 inservice
```

Test Procedure

The procedure used to perform the SSLM Reset test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the following commands to verify they have been cleared:

show mod csm 2 vservers name SSL30 detail

show mod csm 2 vservers name VIP30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 serverfarms name SSLSM detail

show mod csm 2 probe name SSL detail | include SSLSM

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 3 Issue the following commands on both SSLMs to verify the services are operational.

show ssl-proxy service BACKEND30

show ssl-proxy service CLEAR29

show ssl-proxy service SSL-backend

Step 4 Issue the following commands on both SSLSMs to clear ssl-proxy service stats.

clear ssl-proxy stats service CLEAR29

clear ssl-proxy stats service BACKEND30

clear ssl-proxy stats service SSL-backend

Step 5 Issue the following commands to verify they have been cleared.

show ssl-proxy stats service CLEAR29

show ssl-proxy stats service BACKEND30

show ssl-proxy stats service SSL-backend

Step 6 From outside client initiate long lived HTTPS flow to vserver VIP30

Step 7 Issue the following commands on the active CSM to verify vservers SSL29, SSL30, VIP30 and VIP29 have open connections.

show mod csm 2 vservers name SSL30 detail

show mod csm 2 vservers name VIP30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 serverfarms name SSLSM detail

show mod csm 2 probe name SSL detail | include SSLSM

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 8 Issue the following commands on both SSLSMs to verify the conns attempted and conns completed counter has incremented and there are no errors:

show ssl-proxy stats service BACKEND30

show ssl-proxy stats service CLEAR29

show ssl-proxy stats service SSL-backend

Step 9 Issue the **hw-module module 3 reset** command on agg-1 reset SSLSM1.

Step 10 Monitor the client traffic when the CSM Probe dectects a failure it should reset one of the active connections.

Step 11 When you see the CSM log message indicating the probe failure send another HTTPS request from the client whose connections was reset.

Step 12 Issue the following commands on the active CSM to verify the TCP probe has failed real sslm1 and all traffic is being load balanced to sslm2:

l

show mod csm 2 vservers name SSL30 detail

show mod csm 2 vservers name VIP30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 serverfarms name SSLSM detail

sh mod csm 2 probe name SSL detail

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 13 Issue the following commands to verify the conns attempted and conns completed counter are still incrementing and there are no errors:

show ssl-proxy stats service BACKEND30

show ssl-proxy stats service CLEAR29

show ssl-proxy stats service SSL-backend

Step 14 After the SSLM becomes operational generate multiple HTTPS request to vserver SSL30.

Step 15 Issue the following commands to make sure traffic is being load balanced among the four SSLMs:

show mod csm 2 vservers name SSL30 detail

show mod csm 2 vservers name VIP30 detail

show mod csm 2 real sfarm SSLSM detail

show mod csm 2 real sfarm FARM30 detail

show mod csm 2 serverfarms name SSLSM detail

sh mod csm 2 probe name icmp detail

show mod csm 2 stats

show mod csm 2 sticky

show mod csm 2 conn

Step 16 Stop background scripts to collect final status of network devices and analyze for error.

Step 17 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the CSM TCP probe to detect the SSLM failure.
- We expect the CSM to reset open connections when a probe fails a real.
- We expect the CSM to properly load balance traffic during a real failure.

Results

SSLM Reset passed.

Sorry Server

This test verified the CSM ability to switchover incoming connections to a backup server farm, should the primary server farm become unavailable.

Relevant CSM Configuration

```
probe FORCED-FAIL http
 request method get url /notthere.html
 expect status 200 299
 interval 10
 retries 2
 failed 5
 open 3
 receive 5
!
serverfarm SORRY
  nat server
  no nat client
  predictor leastconns
  real name P1_ETH1.3001
   inservice
  real name P1_ETH1.3002
   inservice
!
```

```
serverfarm SORRY-BACK
  nat server
  no nat client
  predictor leastconns
  real name P1_ETH1.3003
   inservice
  real name P1_ETH1.3004
   inservice
!
vserver SORRY
 virtual 101.40.30.240 tcp www
 serverfarm SORRY backup SORRY-BACK
 persistent rebalance
 inservice
```

Test Procedure

The procedure used to perform the Sorry Server test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify that all real servers in each of the primary and backup serverfarms are operational.

show mod csm 2 reals sfarm sorry det

show mod csm 2 reals sfarm sorry-back det

Step 3 Clear the CSM counters

clear mod csm 2 counters

Step 4 Show the counters to make sure they are clear.

sh mod csm 2 serverfarms name SORRY-BACK det

sh mod csm 2 serverfarms name SORRY det

show mod csm 2 reals sfarm sorry det

show mod csm 2 reals sfarm sorry-back det

Step 5 Send 10 HTTPv1.0 GET requests to the vserver SORRY.

Step 6 Verify that the connections are being made to the primary serverfarm and not to the backup serverfarm.

show mod csm 2 vserver name sorry det

show mod csm 2 reals sfarm sorry det

show mod csm 2 reals sfarm sorry-back det

Step 7 Configure the probe FORCED-FAIL on the primary serverfarm, forcing all of its real servers to fail. probe forced-fail

Step 8 Send 10 HTTPv1.0 GET requests to the vserver SORRY.

Step 9 Verify that the real servers in serverfarm SORRY have failed and that the serverfarm SORRY-BACK is now hosting the new connections

show mod csm 2 vserver name sorry det

show mod csm 2 reals sfarm sorry det

show mod csm 2 reals sfarm sorry-back det

Step 10 Remove the probe FORCED-FAIL from the configuration of serverfarm SORRY, no probe forced-fail

Step 11 Verify that the real servers within serverfarm SORRY are again operational and accepting incoming connections, while the real servers within the backup serverfarm are no longer hosting incoming connections

show mod csm 2 vserver name sorry det

show mod csm 2 reals sfarm sorry det

show mod csm 2 reals sfarm sorry-back det

Step 12 Stop background scripts to collect final status of network devices and analyze for error.

Step 13 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect traffic to be forwarded to a backup server farm only when the probes fail on the primary server farm.

Results

Sorry Server passed with exception CSCdz22187, CSCsg97985.

HSRP Failover

This test verified HSRP failover when a system failure occurred. This test also verified that the HSRP preempt command worked when the system returned to an operational state, if the interface was configured with a higher priority than the current active router interface in the same HSRP group. HTTPS traffic was sent through a FWSM and load balanced via CSM and SSLM.

Test Procedure

The procedure used to perform the HSRP Failover test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Issue the **show standby brief** command on both 6500s to verify agg1 is active for all VLANs.

Step 3 Issue the **clear mod csm 2 count** command on the active CSM to clear csm counters. Issue the following commands to verify they have been cleared and to verify state: show mod csm 2 vservers name SSL29 detail show mod csm 2 vservers name VIP29 detail show mod csm 2 real sfarm SSLSM detail show mod csm 2 real sfarm FARM29 detail show mod csm 2 stats

Step 4 Issue the **show ssl-proxy service** command on all SSLSMs to verify the services are operational.

Step 5 Issue the **clear ssl-proxy stats service** command on all four SSLSMs to clear ssl-proxy service stats. Issue the **show ssl-proxy stats service** command to verify they have been cleared. Please note some counters might have incremented due to CSM probes.

Step 6 From outside client initiate HTTPS traffic to vserver VIP29

Step 7 Issue the following commands on the active CSM to verify vservers SSL29, VIP29, SSL29 and VIP30 have open connections. show mod csm 2 vservers name SSL29 detail show mod csm 2 vservers name VIP29 detail show mod csm 2 real sfarm SSLSM detail show mod csm 2 real sfarm FARM29 detail show mod csm 2 stats

Step 8 Issue the following commands on all four SSLSMs to verify the conns attempted and conns completed counter has incremented and there are no errors.

show ssl-proxy stats service

show ssl-proxy stats ssl

Step 9 Issue the **reload** command on agg1 to force a failover.

Step 10 Issue the **show standby brief** command on agg22 to verify its now active.

Step 11 Issue the following commands on the active CSM to verify vservers SSL29, have open connections. show mod csm 2 vservers name SSL29 detail show mod csm 2 vservers name VIP29 detail show mod csm 2 real sfarm SSLSM detail show mod csm 2 real sfarm FARM29 detail show mod csm 2 stats

Step 12 Issue the following commands on both SSLSMs in agg2 to verify the conns attempted and conns completed counter are still incrementing and there are no errors.

show ssl-proxy stats service

show ssl-proxy stats ssl

Step 13 When agg1 becomes operational again issue the **show standby brief** command to verify it preempts and becomes active.

Step 14 Stop background scripts to collect final status of network devices and analyze for error.

Step 15 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the mean failover time for HSRP to be less then the default dead time of 10 seconds.
- We expect that when the failed system becomes operational again, it will resume HSRP active status and forward traffic.

Results

HSRP Failover passed.

Storage Area Networking (SAN)

DCAP SAN testing incorporates Cisco MDS fabric director products and design guides, industry best practices, and storage vendor implementation guidelines to provide a SAN infrastructure that is representative of the typical enterprise data center environment. The centerpiece of the topology configuration is the Cisco MDS 9513 multiprotocol SAN director running SAN-OS version 3.0(3).

SAN Topology

The topology provides redundant fiber channel connectivity for Linux and Windows hosts using QLogic and Emulex host bus adaptors to three different types of fiber channel enterprise storage arrays, namely the EMC DMX3, Network Appliance FAS6070, and Hewlett Packard XP10000. The topology also provides redundant fiber channel connectivity for synchronous storage replication and fiber channel over IP connectivity for asynchronous storage replication. Delay simulators allow modeling of a redundant data center environment for disaster recovery and business continuance testing. The topology is designed to use actual hosts and applications to generate test traffic to model actual customer environments as close as possible.

Figure 4-1 depicts the entire SAN topology, including MDS switches, end devices, and test devices. The MDS switches are mostly MDS9513s with these components:

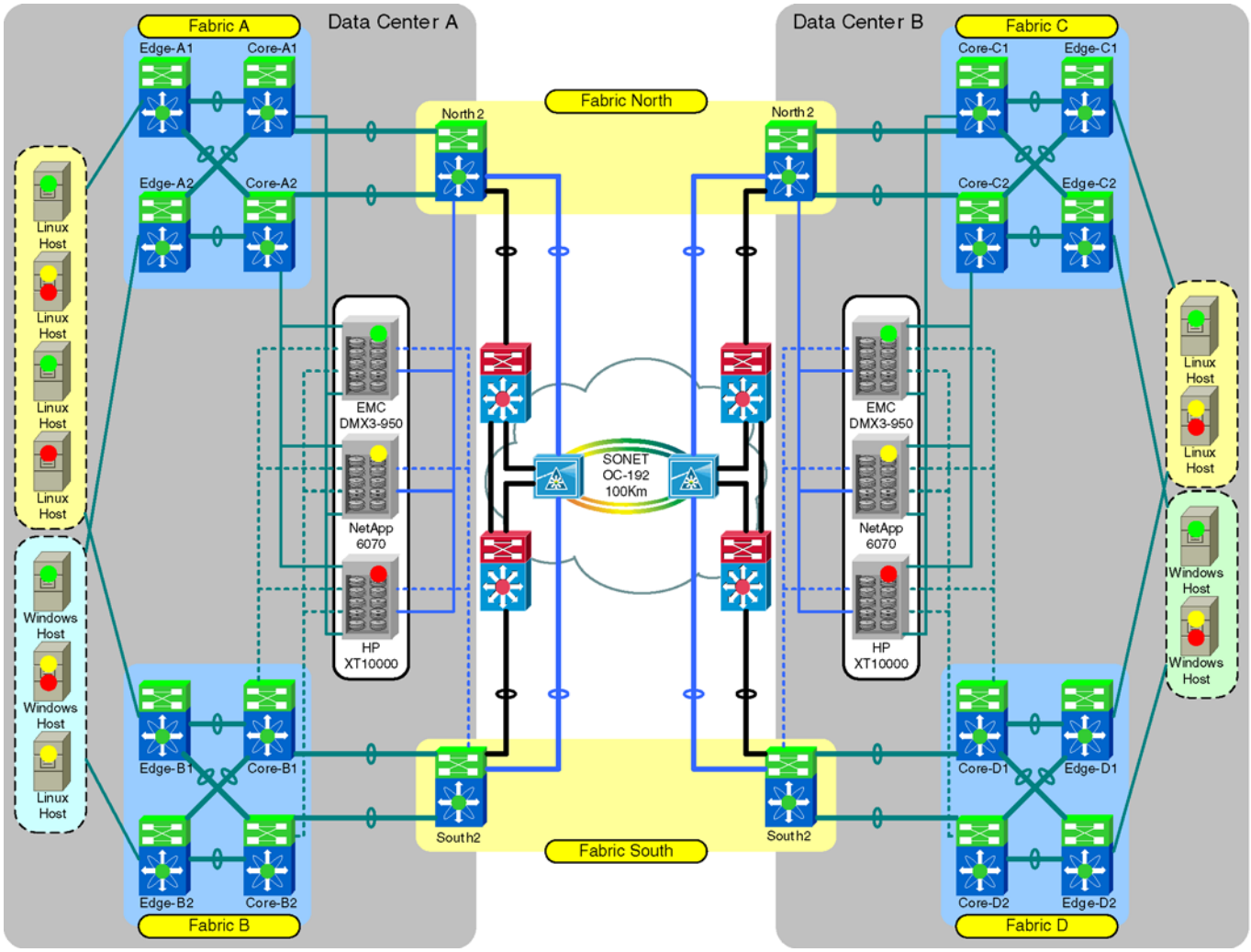
- Redundant version 2 supervisors for high availability and nondisruptive upgrade capability.
- FC modules with 12, 24, and 48 4-Gbps capable ports used for host and storage connectivity as well as interswitch links.
- Storage Service Modules with 32 2-Gbps capable FC ports used for testing replication with FC write acceleration.
- 14+2 modules with 14 2-Gbps capable FC ports and 2 Gigabit Ethernet ports for FCIP.

The MDS switches provide dual, redundant FC fabrics for connecting hosts to storage as well as dual FC fabrics for synchronous replication and FCIP fabrics for asynchronous storage replication over a transit core consisting of FC and Gigabit Ethernet SONET links. All FC fabrics are implemented using Cisco's VSAN technology. All interswitch links belong to port channels for redundancy.

The end devices include hosts and storage arrays. The hosts are running Linux and Windows and have either 4-Gbps Qlogic or Emulex HBAs providing at least two redundant paths to storage devices. The storage arrays include EMC DMX3, Hewlett Packard XP10000s, and Network Appliance FAS6070s. Hosts connect to edge switches and storage arrays connect to core switches.

The test devices include an Agilent SAN tester with two 4-port N2X 4 Gbps blades and two Anue delay generators. The test devices are used in favor of end devices only when end devices are unsuitable or incapable of supplying the required test conditions.

Figure 4-1 DCAP SAN Test Topology Overview



Transport Core

Figure 4-2 displays the infrastructure used by the storage arrays to replicate data between data centers. The heart of the transit core is a pair of Cisco ONS 15454s which provide eight STS-24c bidirectional, unprotected circuits over two OC-192 SONET links. Four circuits comprise the "north" transit fabric and the other four the "south" transit fabric. Two circuits in each fabric support native FC traffic for synchronous replication and the other two support IP for FCIP traffic for asynchronous replication.

Figure 4-3 isolates the virtual SANs (VSANs) used to implement the dual fabric configuration for host to storage connectivity and storage to storage replication. Separate VSANs facilitate logical isolation of traffic by storage frame vendor and allow tuning FC and FCIP protocols independently.

Figure 4-4 shows the host and storage ports in Fabric A.

Figure 4-5 shows the host and storage ports in Fabric B.

Figure 4-6 shows the host and storage ports in Fabric C.

Figure 4-7 shows the host and storage ports in Fabric D.

Note The dotted orange lines over dcap-m9509-Core-D1 and dcap-m9509-Edge-D1 indicate only one version 2 supervisor is installed.

Figure 4-8 shows the storage replication ports in the North Transit Fabric.

Figure 4-9 shows the storage replication ports in the South Transit Fabric.

Figure 4-2 DCAP SAN Test Topology Transit Core Detail

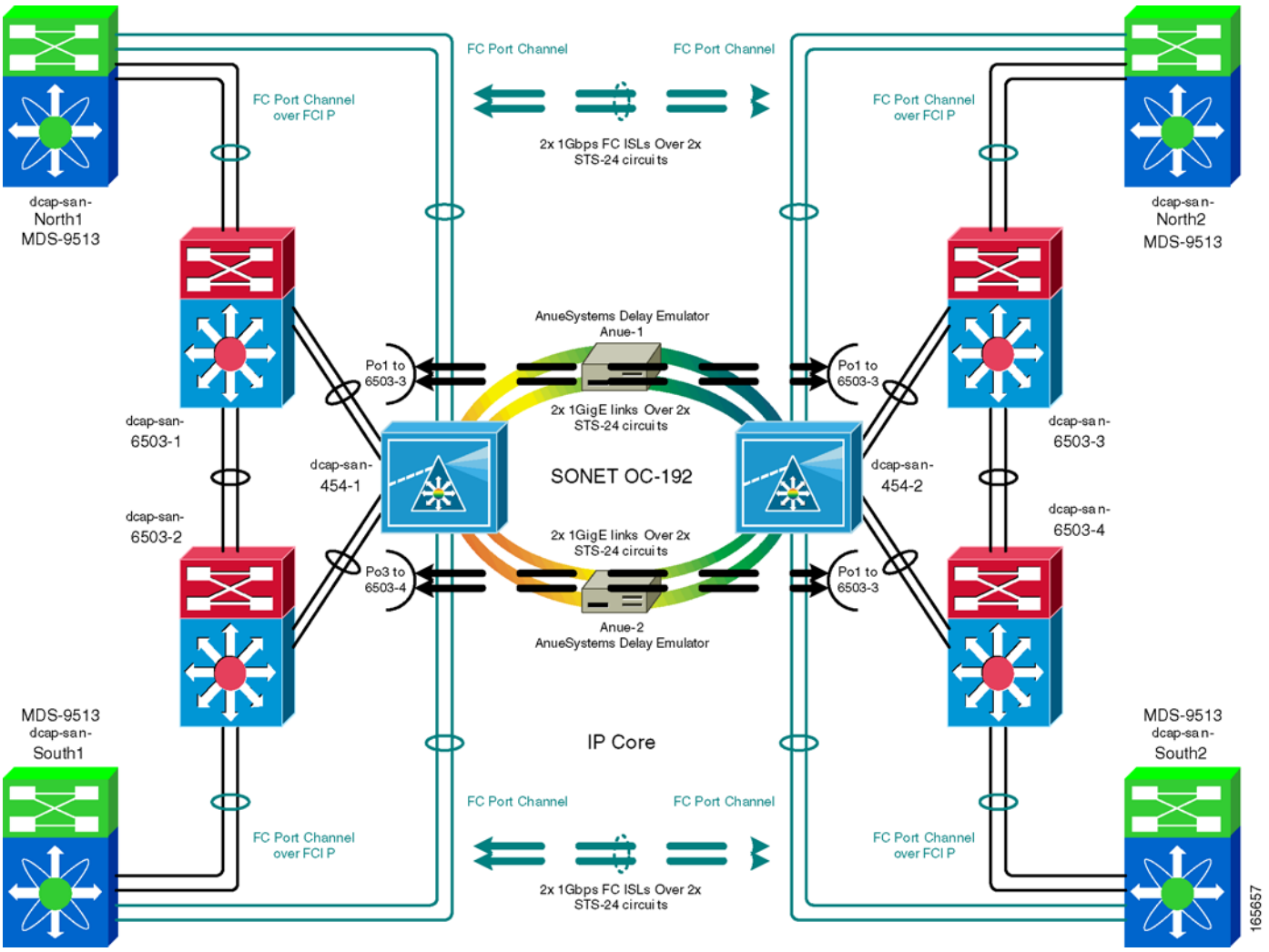


Figure 4-3 DCAP SAN VSAN Traffic Flow

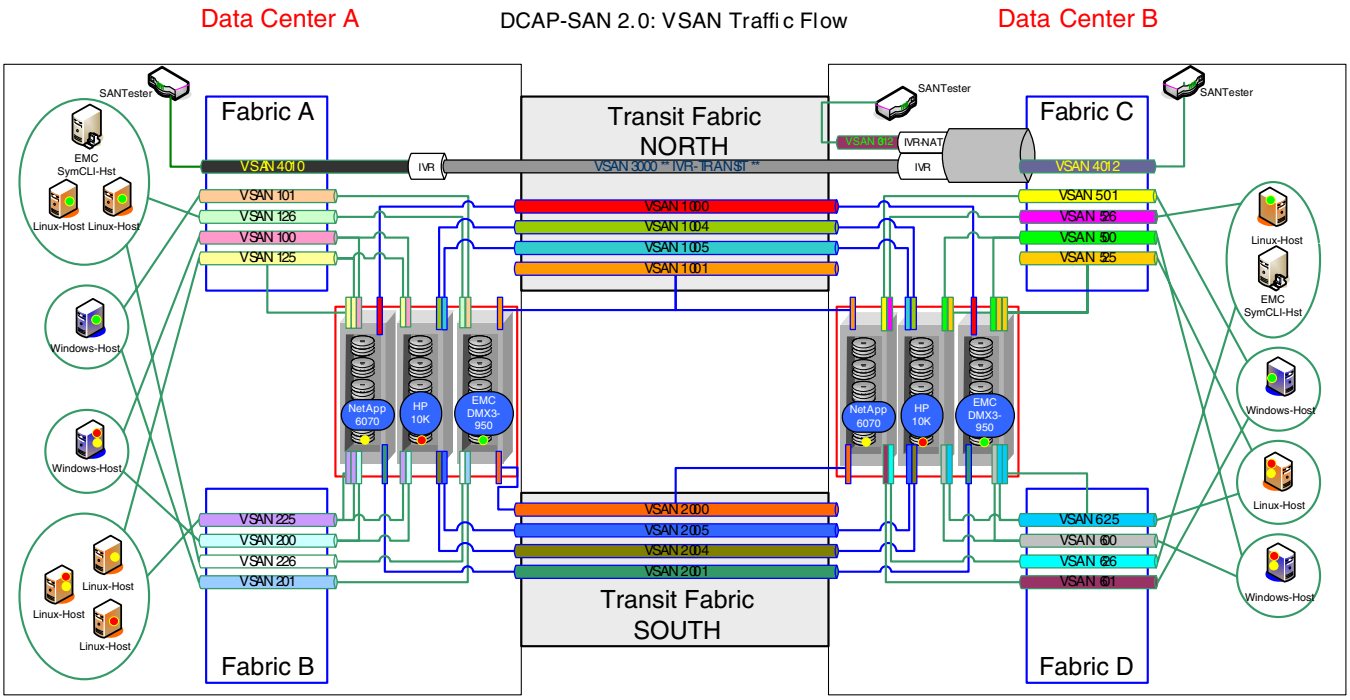


Figure 4-4 Fabric Manager Topology Map for Fabric A

:

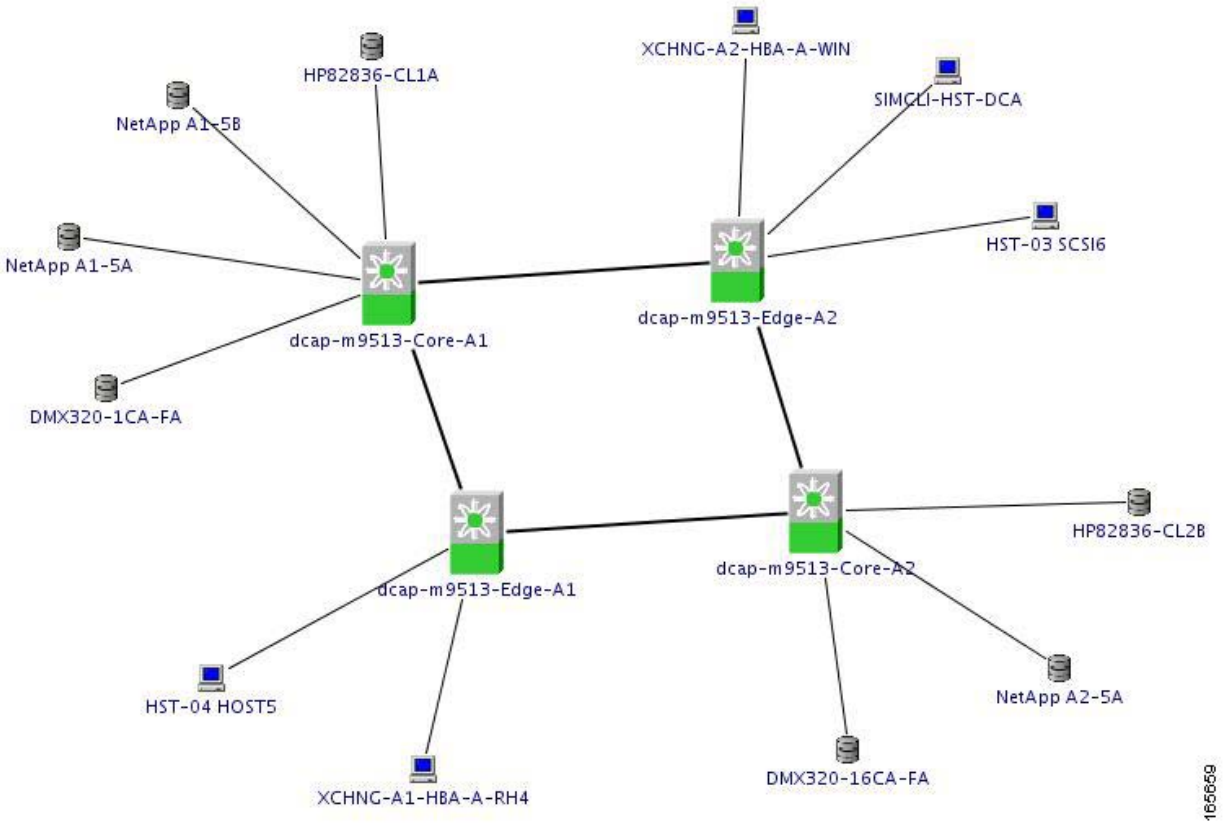


Figure 4-5 Fabric Manager Topology Map for Fabric B

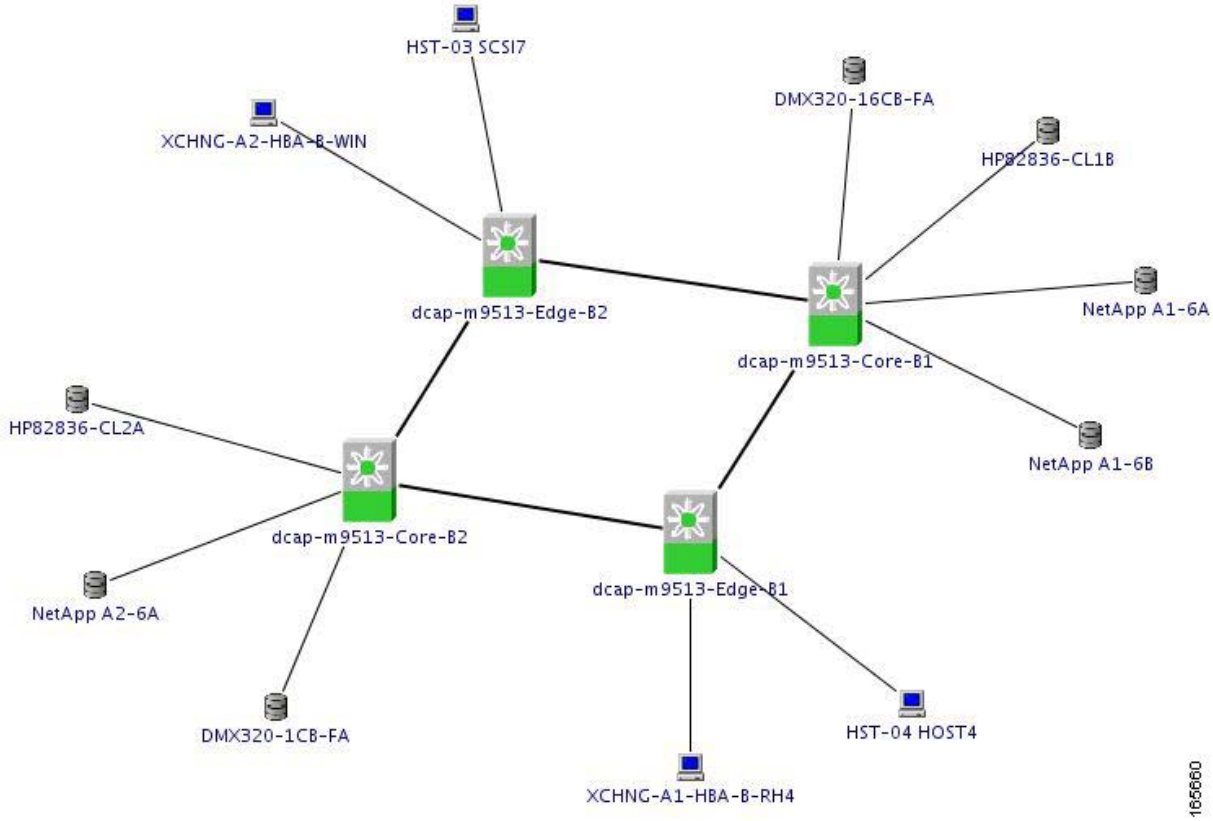


Figure 4-6 Fabric Manager Topology Map for Fabric C

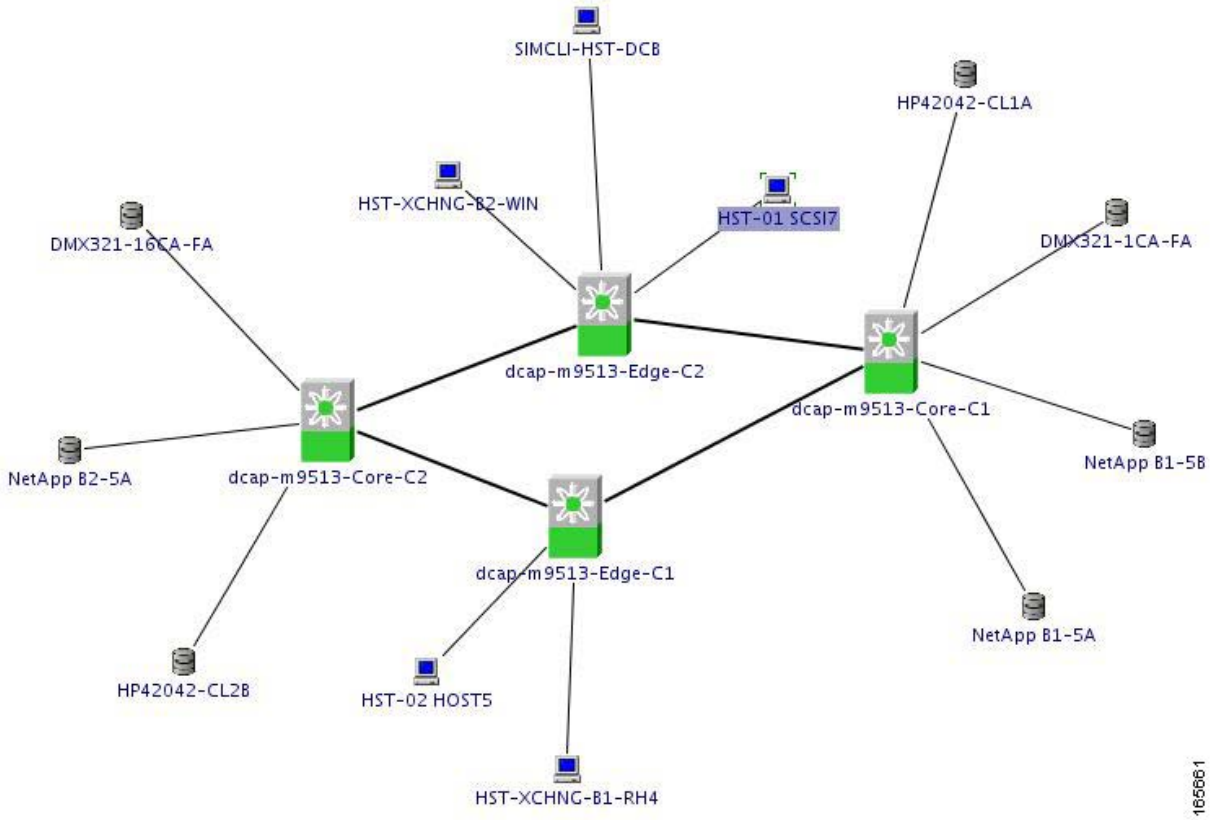


Figure 4-7 Fabric Manager Topology Map for Fabric D

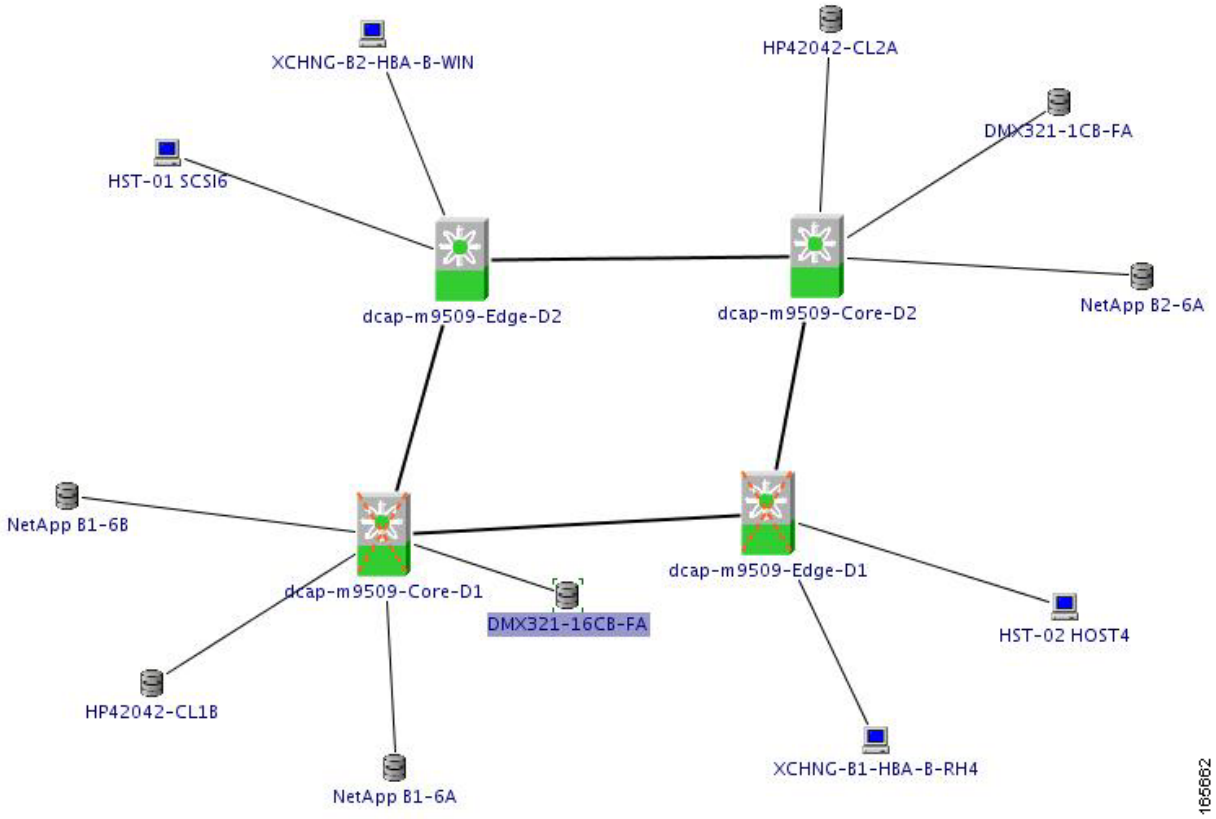


Figure 4-8 Fabric Manager Topology Map for North Transit Fabric

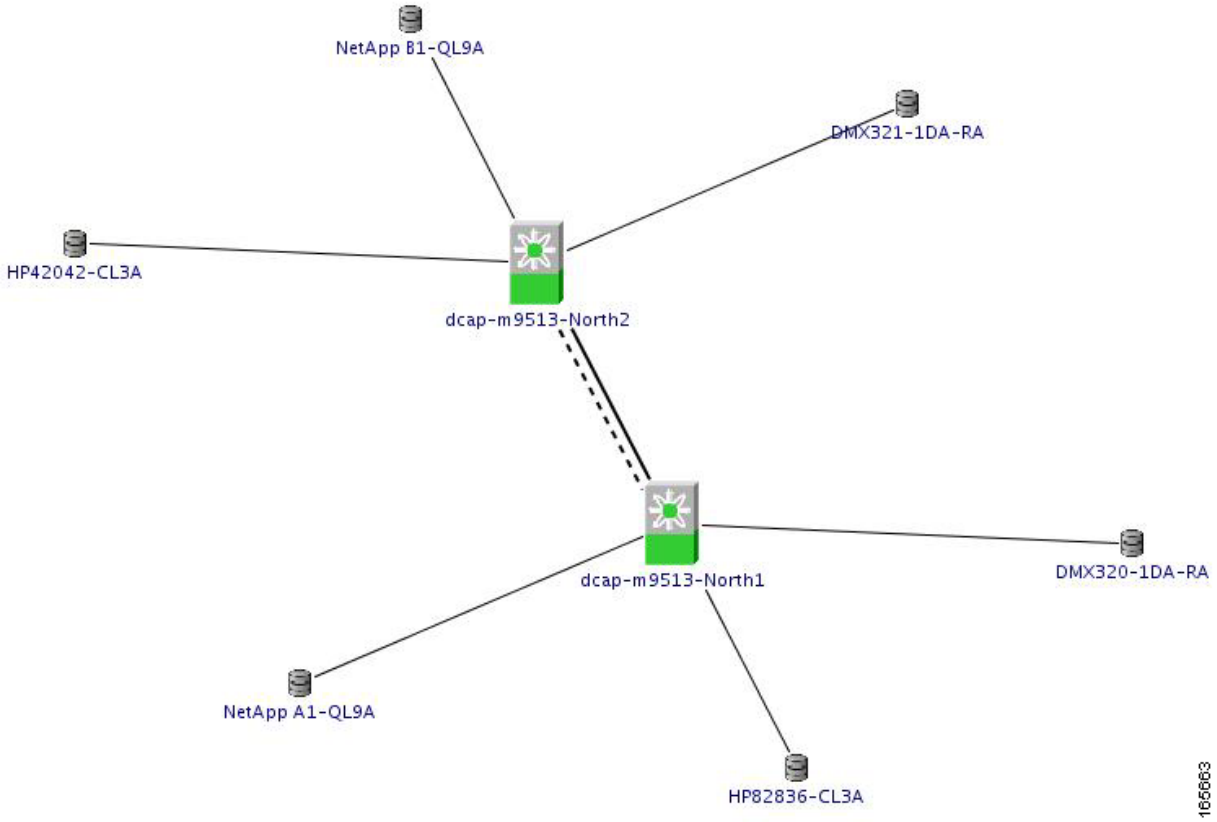
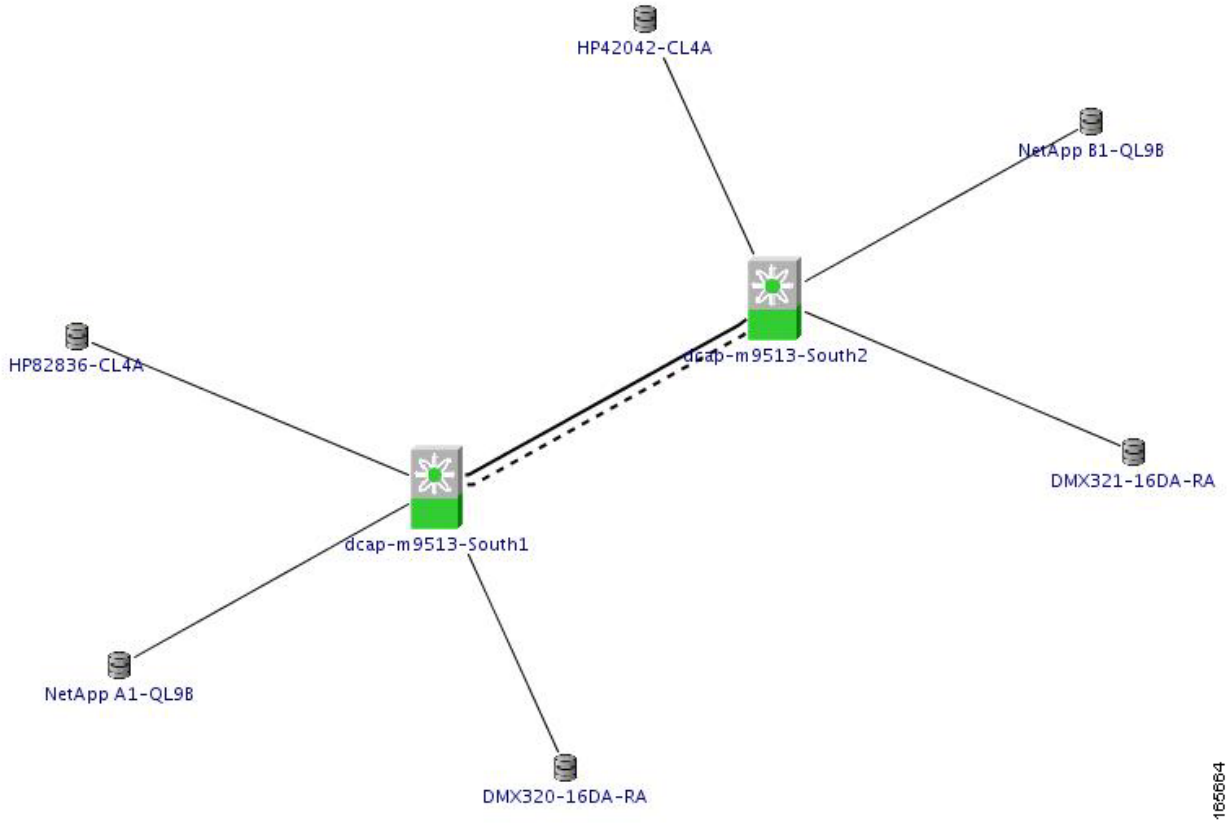


Figure 4-9 Fabric Manager Topology Map for South Transit Fabric



The Cisco DCAP 2.0 Storage Area Network (SAN) tests cover the MDS9500 platform and fall into four major categories: Baseline Testing, Functionality Testing, Resilience Testing, and SAN Extension Testing.

Baseline Testing looks at the general functionality and configuration of the devices in the DCAP SAN test topology. This includes ensuring each test host has redundant paths to each storage array and verifying replication paths between pairs of storage arrays are working properly. Configurations follow best practices for VSAN configuration, port allocation, zoning, and storage device mapping and masking. Functionality Testing covers key device management and Fiber Channel (FC) protocols such as virtual SAN (VSAN) configuration, port channels, Fabric Shortest Path First (FSPF), Inter-Virtual SAN routing (IVR), and security. Resilience Testing measures the response of the DCAP SAN topology to various failure conditions like cable pulls, power outage, component failures, and supervisor and module reloads and online insertions and replacements (OIRs). SAN Extension Testing focuses on replication of data between actual storage frames from EMC, Hewlett Packard, and Network Appliance using both the native FC protocol and the FC over Internet Protocol (FCIP) with simulated latencies. Advanced capabilities like FC write acceleration and FCIP compression, write acceleration, and encryption are included.

The DCAP SAN topology is divided into three distinct, logical layers called the Transit, Core, and Edge layers offering services listed in Table 4-1.

Table 4-1 DCAP SAN Logical Layer Services

Logical Layer	Services
Transit	FC over SONET fabric extension (for synchronous storage replication), FCIP fabric extension (for asynchronous storage replication), DPVM, IVR (with and without NAT), FCIP compression and encryption, FC and FCIP write acceleration
Core	Storage fabric connectivity
Edge	Host fabric connectivity
All Layers	VSANs, FSPF, port channels, zoning, port security, FC-SP, TACACS+ authentication

The DCAP SAN topology incorporates Cisco MDS fabric director products and design guides, industry best practices, and storage vendor implementation guidelines to provide a SAN infrastructure that is representative of the typical enterprise data center environment. The infrastructure provides both fundamental and advanced fiber channel SAN services for both host to storage connectivity and storage replication. The topology includes two separate data centers, each with dual, redundant core/edge fabrics for highly available and resilient connectivity, and dual transit fabrics that allow storage frame-based replication between the data centers for disaster recovery and business continuance. Delay generators from Anue Systems allow simulation of distance between the data centers. For Phase 2, the tested distance is 100 km (62 miles), which corresponds to a round trip time of 1 ms. All fabrics use Cisco's Virtual SAN (VSAN) technology.

The transit fabric is the focal point of the topology and testing, since Cisco SAN extension capabilities are key differentiators for many customers. SAN extension is a key enabler for modern data center designs which call for redundant data centers. Cisco SAN extension supports both synchronous and asynchronous replication. The use of these terms follows industry practice; that is, synchronous replication means a host write transaction is not acknowledged by the primary or local storage frame until the secondary or remote storage frame confirms it has safely stored a replica of the data, and asynchronous replication means a successful host write is immediately acknowledged by the primary frame.

The heart of the transit core is a pair of Cisco ONS 15454s which provide eight STS-24c bidirectional, unprotected circuits over two OC-192 SONET links. Four circuits comprise the "north" transit fabric and the other four the "south" transit fabric. Two circuits in each fabric support native FC traffic for synchronous replication and the other two support IP for FCIP traffic for asynchronous replication.

The primary SAN switch used in the topology is the Cisco MDS9513 (model DS-C9513) with dual version 2 supervisors (model DS-X9530-SF2-K9) for maximum scaleability and nondisruptive operation and upgrades. Host connectivity is through high density 48-port FC modules (model DS-X9148) which provide 4 Gbps of bandwidth on an oversubscribed basis. Storage connectivity is through lower density 12- and 24-port FC modules (models DS-X9112 and DS-X9124). Inter-switch links (ISLs) use 12-port FC modules to maximize bandwidth. All ISLs are in port channels. Connectivity between data centers relies on 32-port Storage Services Modules (model DS-X9032-SSM) for synchronous replication using FC over SONET and 14+2 port modules (model DS-X9302-14K9) for asynchronous replication using FC over IP (FCIP). The 14+2 modules have 14 FC interfaces capable of 2 Gbps and 2 Gigabit Ethernet interfaces.

The topology is designed to support testing that's as realistic as possible. Although test devices such as an Agilent SAN tester and Anue delay generators are part of the topology, these are only used when actual hosts and storage devices cannot provide suitable test conditions. To support this testing approach, each data center has FC storage frames from EMC (model DMX3), Network Appliance (model FAS6070), and Hewlett Packard (model XP10000). Each storage frame provides devices for primary application access as well as replication between data centers. The EMC frames use Synchronous Replication Data Facility/Synchronous (SRDF/S) for synchronous replication and SRDF/Asynchronous

(SRDF/A) for asynchronous replication. The Network Appliance frames use SnapMirror for both types of replication. The Hewlett Packard frames use XP Continuous Access (CA) Synchronous for synchronous replication and and XP CA Asynchronous for asynchronous replication. More details for each frame vendor are in the appendix section.

Each data center also has both Windows and Linux servers with multipath software and either 4-Gbps Qlogic or Emulex HBAs providing at least two redundant paths to storage devices. Hosts accessing EMC storage use PowerPath for both Windows and Linux. Hosts with Network Appliance or Hewlett Packard storage use Veritas DMP for Windows and native MPIO (device-mapper-multipath) for Linux. In Phase 2, the majority of the tests are based on I/O generated by iometer on Windows and iorate on Linux. These tools are desirable because they're readily available in source code and binary distributions, easy to deploy and use, and provide basic performance information that helps determine the success or failure of tested MDS features. For Phase 2, both iometer and iorate were configured to do 8 KB sequential writes to generate load.

Test Results Summary

Table 4-2 summarizes tests executed as part of the Cisco DCAP 2.0 testing initiative. Table 4-2 includes the feature or function tested, the section that describes the feature set the feature or function belongs to, the component tests for each feature or function, and whether the test is new in this phase of DCAP testing.

Note Test results are unique to technologies covered and actual scenarios in which they were tested. DCAP is designed to cover critical path areas and augment ongoing regression and systems testing.

Table 4-2 Cisco DCAP 2.0 SAN Testing Summary

Test Suites	Features/Functions	Tests	Results
Baseline, page 4-17	A.1: Device Check, page 4-18	1. Device Access—CLI and Device Manager 2. Device Hardware Check—CLI 3. Device Hardware Check—Device Manager 4. Device Network Services Check—CLI 5. Device Network Services Check—Device Manager	
	A.2: Infrastructure Check, page 4-22	1. Host and Storage Fabric Connectivity—NetApp 2. Host and Storage Fabric Connectivity—EMC 3. Intra-Fabric Connectivity 4. Topology Discovery—Fabric Manager	
	A.3: Host to Storage Traffic—EMC, page 4-25	1. Base Setup—VSANs EMC 2. Base Setup—Zoning EMC 3. Host To Storage IO Traffic—EMC 4. Replication FC Sync—EMC 5. Replication FCIP ASync—EMC	
	A.4: Host to Storage Traffic—NetApp, page 4-30	1. Base Setup—VSANs NetApp 2. Base Setup—Zoning NetApp 3. Host To Storage IO Traffic—NetApp 4. Replication FC-Sync—NetApp 5. Replication FCIP-Async—NetApp	
Device Access—CLI and Device Manager, page 4-18	Device Access—CLI and Device Manager, page 4-18	1. Principal Switch Selection	
FSPF Functionality, page 4-37	FSPF Functionality, page 4-37	1. Basic FSPF Load Balancing 2. Path Selection—Cost Change on Equal Cost Paths 3. Primary Path Failure 4. Primary Path Removal—VSAN Remove	

Table 4-2 Cisco DCAP 2.0 SAN Testing Summary (continued)

Test Suites	Features/Functions	Tests	Results
Fabric Extension, page 4-40	Async Replication—EMC, page 4-41	1. FCIP COMP 100Km EMC 2. FCIP ENCRP 100Km EMC 3. FCIP NONE 100Km EMC 4. FCIP WA 100Km EMC 5. FCIP WA COMP ENCRP 100Km EMC 6. FCIP Portchannel Failure 100Km EMC	
	Async Replication—NetApp, page 4-48	1. FCIP COMP 100Km NETAPP 2. FCIP ENCRP 100Km NETAPP 3. FCIP NONE 100Km NETAPP 4. FCIP WA 100Km NETAPP 5. FCIP WA COMP ENCRP 100Km NETAPP 6. FCIP Portchannel Failure 100Km NETAPP	
	Async Replication—HP, page 4-56	1. FCIP COMP 100Km HP 2. FCIP ENCRP 100Km HP 3. FCIP NONE 100Km HP 4. FCIP WA 100Km HP 5. FCIP WA COMP ENCRP 100Km HP	
	FCIP COMP 100Km HP, page 4-56	1. FC Sync—DST=100Km, WA=OFF - EMC 2. FC Sync—DST=100Km, WA=ON - EMC 3. FC Sync—Portchannel Failure, DST=100Km - EMC	
	Sync Replication—NetApp, page 4-65	1. FC Sync—DST=100Km, WA=OFF - NetApp 2. FC Sync—DST=100Km, WA=ON - NetApp 3. FC Sync—Portchannel Failure, DST=100Km - NetApp	
	Sync Replication—HP, page 4-69	1. FC Sync—DST=100Km, WA=OFF - HP 2. FC Sync—DST=100Km, WA=ON - HP	
Inter-VSAN Routing Functionality, page 4-71	Inter-VSAN Routing Functionality, page 4-71	1. Basic IVR Implementation 2. Basic IVR-NAT Implementation	
Portchannel Functionality, page 4-73	Portchannel Functionality, page 4-73	1. Basic Portchannel Load Balancing 2. Multiple Link ADD to Group 3. Multiple Links Failure in Group 4. Multiple Links Remove to Group 5. Single Link Add to Group 6. Single Link Failure in Group 7. Single Link Remove from Group	

Table 4-2 Cisco DCAP 2.0 SAN Testing Summary (continued)

Test Suites	Features/Functions	Tests	Results
Device Access—CLI and Device Manager, page 4-18	EMC, page 4-80	1. Host Link Failure (Link Pull)—EMC 2. Host Link Failure (Port Shutdown)—EMC 3. Host Facing Module Failure (OIR)—EMC 4. Host Facing Module Failure (Reload)—EMC	
	NetApp, page 4-84	1. Host Link Failure (Link Pull)—NETAPP 2. Host Link Failure (Port Shutdown)—NETAPP 3. Host Facing Module Failure (OIR)—NETAPP 4. Host Facing Module Failure (Reload)—NETAPP	
	Host Link Failure (Link Pull)—NETAPP, page 4-84	1. Active Crossbar Fabric Failover (OIR) 2. Active Supervisor Failover (OIR) 3. Active Supervisor Failover (Reload) 4. Active Supervisor Failover (Manual CLI) 5. Back Fan-Tray Failure (Removal) 6. Core Facing Module Failure (OIR) 7. Core Facing Module Failure (Reload) 8. Front Fan-Tray Failure (Removal) 9. Node Failure (Power Loss) 10. Node Failure (Reload) 11. Power Supply Failure (Cord Removal) 12. Power Supply Failure (Power Off) 13. Power Supply Failure (Removal) 14. Standby Supervisor Failure (OIR) 15. Standby Supervisor Failure (Reload) 16. Unused Module Failure (OIR)	
Security Functionality, page 4-103	Security Functionality, page 4-103	1. FC SP Authentication Failure 2. Port Security Basic Implementation 3. User Access—TACACS Basic Test 4. User Access—TACACS Servers Failure	
Device Access—CLI and Device Manager, page 4-18	HTTP Functionality, page 4-107	1. End-to-End Get 2. End-to-End Put	

DDTS Summary

Table 4-3 lists Development Defect Tracking System (DDTS) software bugs with descriptions, and comments filed by the DCAP testing team during Cisco DCAP 2.0 SAN testing. Table 4-4 lists DDTS with descriptions encountered during Cisco DCAP 2.0 SAN testing.

Table 4-3 Summary of DDTS Filed During Cisco DCAP 2.0 SAN Testing

CSCsh11203	MDS9500 show hardware internal debug-info has invalid packet-flow syntax. Discovered during troubleshooting NetApp replication issue.

Table 4-4 Summary of DDTS Encountered During Cisco DCAP 2.0 SAN Testing

CSCsg22134	FMS startup dialog repeateadly shown when ipaddress set in svr.hostname.
CSCsg96362	FM java.lang.Nullpointer when editing port-channel in topology map.

SAN Test Cases

Functionality critical to global enterprises in Cisco DCAP 2.0 Storage Area Network (SAN) testing is described in the following sections. Refer to Cisco Data Center Assurance Program (DCAP) 2.0 Configurations document for test device configurations.

- Baseline, page 4-17
- Domain Parameters, page 4-36
- FSPF Functionality, page 4-37
- Fabric Extension, page 4-40
- Inter-VSAN Routing Functionality, page 4-71
- Portchannel Functionality, page 4-73
- Resiliency Functionality, page 4-80
- Security Functionality, page 4-103
- End-to-End Functionality, page 4-107

Baseline

Baseline tests ensure the SAN topology is configured properly for testing and management tools and devices are operating properly. These tests were not formally documented in Phase 2 for HP due to hardware issues and time constraints.

The following test features were conducted:

- A.1: Device Check, page 4-18
- A.2: Infrastructure Check, page 4-22
- A.3: Host to Storage Traffic—EMC, page 4-25
- A.4: Host to Storage Traffic—NetApp, page 4-30

A.1: Device Check

Device check tests ensure management tools and services like the MDS CLI, MDS Fabric Manager, NTP, SMTP, syslog, and so on are accessible and functioning properly.

The following tests were performed:

- Device Access—CLI and Device Manager, page 4-18
- Device Hardware Check—CLI, page 4-19
- Device Hardware Check—Device Manager, page 4-19
- Device Network Services Check—CLI, page 4-20
- Device Network Services Check—Device Manager, page 4-21

Device Access—CLI and Device Manager

Individual devices/nodes in the testbed require multiple access methods for management purposes. These access methods include CLI and NMS. This test validated the support and availability of access via the console , Telnet, SSH, and the Device_Manager application. Access methods were executed from a management station with IP access to the devices and Terminal-Servers.

Test Procedure

The procedure used to perform the Device Access—CLI and Device Manager test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Console] Verify that each node can be accessed and logged into via the console.

Step 3 [Telnet] Verify that each node can be accessed and logged into via Telnet.

Step 4 [SSH] Verify that each node can be accessed and logged into via SSH.

Step 5 [DM] Verify that each node can be accessed and logged into via Device_Manager.

Step 6 Stop background scripts to collect final status of network devices and analyze for error.

Step 7 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that access support for console, Telnet, SSH, and Device_Manager is active and operational.
- We expect local authorization/authentication to be supported by the access methods under validation without problems or issues.
- We expect no unacceptable impact on the CPU or memory.

Results

Device Access—CLI and Device Manager passed.

Device Hardware Check—CLI

All hardware components of each device must be in active and operational status prior to the start any test activities. This test verified that linecards or modules, redundant components, power supply units, and other environmental conditions were without problems or issues in each device.

Test Procedure

The procedure used to perform the Device Hardware Check—CLI test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Access each MDS node in each Fabric and verify that all linecards and modules are in operational 'OK' condition.

Step 3 Check and verify that all environmental related hardware and conditions are in operational 'OK' status.

Step 4 Stop background scripts to collect final status of network devices and analyze for error.

Step 5 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that all linecards or modules are in 'OK' operational condition.
- We expect that all environmental related hardware (e.g., fans, power supply units) are in 'OK' condition and fully operational.

Results

Device Hardware Check—CLI passed.

Device Hardware Check—Device Manager

All hardware components in each device must be in active and operational status prior to the start any test activities. Using the Device_Manager application this test verified that linecards or modules, redundant components, power supply units, and other environmental conditions were without problems or issues in each device prior to test.

Test Procedure

The procedure used to perform the Device Hardware Check—Device Manager test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Use Device_Manager to access each MDS node in each Fabric and verify that all linecards and modules are in operational 'OK' condition.

Step 3 Use Device_Manager to check and verify that all power and environmental related hardware and status are in operational 'OK' condition.

Step 4 Stop background scripts to collect final status of network devices and analyze for error.

Step 5 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that hardware and environmental status of the nodes can be reviewed and validated via Device_Manager.
- We expect that all linecards or modules are in 'OK' operational condition.
- We expect that all environmental related hardware (e.g., fans, power supply units) are in 'OK' condition and fully operational.

Results

Device Hardware Check—Device Manager passed.

Device Network Services Check—CLI

Devices or nodes in the Fabrics are required to have a common clock source vi NTP, SNMP services for remote management and traps, and logging capabilities to remote SYSLOG servers. This test verified that network services (NTP, SNMP, and SYSLOG) are configured and operational in all nodes.

Test Procedure

The procedure used to perform the Device Network Services Check—CLI test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [NTP] Verify that NTP is configured properly (i.e., peers are the correct NTP servers) and operational (i.e., statistics are working) in all nodes.

Step 3 [NTP] Verify that time stamping is synchronized in all nodes.

Step 4 [SNMP] Verify that SNMP (and traps) is configured properly (i.e., SNMP server IP addresses are correct, traps are enabled, communities set) and operational.

Step 5 [SNMP] Verify that IP connectivity to SNMP servers.

Step 6 [SYSLOG] Verify that SYSLOG (logging server) is configured properly (i.e., SYSLOG server IP addresses are correct, timestamps = milliseconds) and operational.

Step 7 [SYSLOG] Verify that IP connectivity to SYSLOG servers.

Step 8 [SYSLOG] Verify SYSLOG functionality by getting in 'configuration mode' and out. This will generate a syslog message. Check the log in the syslog server to validate its delivery.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that NTP services are configured and operational in each node within the testbed.
- We expect that SNMP services are configured and operational in each node within the testbed.

- We expect that SYSLOG services are configured and operational in each node within the testbed.

Results

Device Network Services Check—CLI passed.

Device Network Services Check—Device Manager

Devices or nodes in the Fabrics are required to have a common clock source via NTP, SNMP services for remote management and traps, and logging capabilities to remote SYSLOG servers. Using the Device Manager application this test case verified that network services (NTP, SNMP, and SYSLOG) are configured and operational in all nodes.

Test Procedure

The procedure used to perform the Device Network Services Check—Device Manager test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [NTP] Verify that NTP is configured properly (i.e., peers are the correct NTP servers) and operational (i.e., statistics are working) in all nodes.

Step 3 [NTP] Verify that time stamping is synchronized in all nodes.

Step 4 [SNMP] Verify that SNMP (and traps) is configured properly (i.e., SNMP server IP addresses are correct, traps are enabled, communities set) and operational.

Step 5 [SNMP] Verify trap-generation functionality by checking for recent Fabric Manager events. If there aren't any, try resetting a port (be sure this doesn't cause unexpected disruption of traffic). This will generate a trap - check events in FM or DM.

Step 6 [SYSLOG] Verify that SYSLOG (logging server) is configured properly (i.e., SYSLOG server IP addresses are correct, timestamps = milliseconds) and operational.

Step 7 [SYSLOG] Verify IP connectivity to SYSLOG servers.

Step 8 [SYSLOG] Verify SYSLOG functionality by getting into and then out of configuration mode. Check the log in the syslog server to validate its delivery.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that Device_Manager can accurately verify the active operational state of the services in question.
- We expect that NTP services are configured and operational in each node within the testbed.
- We expect that SNMP services are configured and operational in each node within the testbed.
- We expect that SYSLOG services are configured and operational in each node within the testbed.

Results

Device Network Services Check—Device Manager passed.

A.2: Infrastructure Check

Infrastructure check tests ensures all hosts and storage devices are logged into the appropriate fabric, storage replication works, and Fabric Manager properly discovers all fabrics and reports no anomalies.

The following tests were performed:

- Host and Storage Fabric Connectivity—NetApp, page 4-22
- Host and Storage Fabric Connectivity—EMC, page 4-23
- Intra-Fabric Connectivity, page 4-24
- Topology Discovery—Fabric Manager, page 4-24

Host and Storage Fabric Connectivity—NetApp

The connectivity between test hosts, storage arrays, and the Fabrics was tested to ensure a problem free infrastructure prior to testing. The verification was done by means of checking port status/conditions and complete fabric logins from the part of the end devices in all links available (e.g., devices are dual-homed in many instances). This was done via the Fabric Manager application (part of the discovery process) with CLI validation.

Test Procedure

The procedure used to perform the Host and Storage Fabric Connectivity—NetApp test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [FM] Verify that all test hosts and storage arrays' connections are active and displayed correctly in the topology map.

Step 3 [FM] Verify successful fabric logins, fcns registration, and Device Aliases by checking correct PWWN, HBAs' IDs, aliases.

Step 4 [FM] Verify correct HBAs' manufacturers information (model, firmware, driver).

Step 5 [FM] Check all hosts and storage arrays fabric ports against errors.

Step 6 [CLI] Validate FM information (previous steps) via CLI.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that Fabric_Manager's fabric discovery process accurately presents host and storage arrays' connectivity information.
- We expect all links between hosts and corresponding fabric nodes to be active (UP/UP). The same applies to the storage arrays links.

- We expect all test hosts and storage arrays to successfully log-in into the Fabrics (flogi).
- We expect no unacceptable impact on the CPU or memory.

Results

Host and Storage Fabric Connectivity—NetApp passed.

Host and Storage Fabric Connectivity—EMC

The connectivity between test hosts, storage arrays, and the Fabrics was tested to ensure a problem free infrastructure prior to testing. The verification was done by means of checking port status/conditions and complete fabric logins from the part of the end devices in all links available (e.g., devices are dual-hommed in many instances). This was done via the Fabric_Manager application (part of the discovery process) with CLI validation.

Test Procedure

The procedure used to perform the Host and Storage Fabric Connectivity—EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [FM] Verify that all test hosts and storage arrays' connections are active and displayed correctly in the topology map.

Step 3 [FM] Verify successful fabric logins, fcns registration, and Device Aliases by checking correct PWWN, HBAs' IDs, aliases.

Step 4 [FM] Check all hosts and storage arrays fabric ports against errors.

Step 5 [CLI] Validate FM information (previous steps) via CLI.

Step 6 Stop background scripts to collect final status of network devices and analyze for error.

Step 7 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that Fabric_Manager's fabric discovery process accurately presents host and storage arrays' connectivity information.
- We expect that all links between hosts and corresponding fabric nodes are active (UP/UP). The same applies to the storage arrays links.
- We expect that all test hosts and storage arrays successfully log-in into the Fabrics (flogi).

Results

Host and Storage Fabric Connectivity—EMC passed.

Intra-Fabric Connectivity

Intra-Fabrics' connections (e.g., links, Port-Channels) and operational conditions were tested to ensure proper end-to-end connectivity within stable fabrics. The accuracy or proper discovery / representation of such conditions via Fabric_Manager was part of this test. The test and validation was executed using Fabric_Manager and verified via CLI.

Test Procedure

The procedure used to perform the Intra-Fabric Connectivity test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [FM] Verify that all links between Fabric nodes are active and operational.

Step 3 [FM] Verify that all Port-Channels are active (i.e., all intended members are active in the channel).

Step 4 [FM]Verify connectivity within the transport fabrics (North and South) - over IP and over SONET.

Step 5 [CLI] Validate/confirm, via CLI, all intra-Fabric connectivity as verified with FM in the previous steps. Run "show port-channel summary" and verify total and operational port counts match.

Step 6 Stop background scripts to collect final status of network devices and analyze for error.

Step 7 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that Fabric_Manager successfully discovers all intra-Fabric links/connections with accurate status.
- We expect that all links between fabric nodes are active and operating without problems (e.g., no errors).
- We expect for all Port-Channels to be fully operational.
- We expect for all fabric nodes to be able to see each other.

Results

Intra-Fabric Connectivity passed.

Topology Discovery—Fabric Manager

This test verified that Fabric Manager and Device Manager can accurately and completely discover all 6 Fabrics and devices attached to them. The appropriate "Host and Storage Fabric Connectivity" test cases and the "Intra-Fabric Connectivity" test case must have been executed before this one.

Test Procedure

The procedure used to perform the Topology Discovery—Fabric Manager test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Start Fabric Manager and open each fabric.

Step 3 Verify that all MDS nodes, hosts, storage arrays are discovered and accurately identified.

Step 4 Open Device Manager sessions to each node in each fabric and verify proper hardware layout and ID information.

Step 5 Stop background scripts to collect final status of network devices and analyze for error.

Step 6 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that Fabric Manager will be able to discover all nodes in all 6 fabrics without problem or issue.
- We expect nounacceptable impact on the CPU or memory.

Results

Topology Discovery—Fabric Manager passed.

A.3: Host to Storage Traffic—EMC

Host-to-storage traffic tests for EMC ensure hosts can access storage devices and that both SRDF/S for synchronous replication and SRDF/A for asynchronous replication are working properly.

The following tests were performed:

- Base Setup—VSANs EMC, page 4-25
- Base Setup—Zoning EMC, page 4-26
- Host To Storage IO Traffic—EMC, page 4-27
- Replication FC Sync—EMC, page 4-28
- Replication FCIP ASync—EMC, page 4-29

Base Setup—VSANs EMC

Host-to-Storage communication is the first most essential and basic service that a SAN must provide followed by Replication (Storage-to-Storage for Business Continuance). These services are made up of building blocks which include: VSAN ports membership, zone membership, zoneset activation, LUN masking, etc. This test verified the basic configuration and activation of all VSANs needed for Host-to-Storage and Replication communication between hosts (w/ multiple operating systems), storage arrays, and storage array pair. VSANs were configured and verified via Fabric_Manager (w/ CLI validation).

Test Procedure

The procedure used to perform the Base Setup—VSANs EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Two Test-Hosts are selected each with a different Operating System (Windows Enterprise 2003 Server and Linux RedHat Enterprise). Both hosts are dual-hommed to two separate fabrics (as per test topology - Fabric A and B -OR- Fabric C and D). [Pre-Test Condition #2] Storage Arrays are dual-hommed to the Host fabrics and to the Replication fabrics. [Pre-Test Condition #3] Storate_Array's LUN Masking should be configured to allow access from the Test-Hosts to the proper (non-replicating) LUNs. [Pre-Test Condition #4] Storage Array's Replication services must be enabled for sync and async replication between selected LUNs.

Step 3 [VSAN] Create one(1) Windows_Hosts VSAN per fabric. Add the Windows_Host and matching Storage_Arrays fabric ports to that VSAN as members.

Step 4 [VSAN] Check that Windows_Host and matching Storage_Arrays fabric ports re-login into the fabrics and into the FC Name Server under the correct Windows_Host VSAN.

Step 5 [VSAN] Create one(1) Linux_Hosts VSAN per fabric. Add the Linux_Host and matching Storage_Arrays fabric ports to that VSAN as members.

Step 6 [VSAN] Check that Linux_Host and matching Storage_Arrays fabric ports re-login into the fabrics and into the FC Name Server under the correct Linux_Host VSAN.

Step 7 [VSAN] Create two(2) Replication VSANs per Transport/Replication fabric. Add the Storage Array's fabric ports to those VSANs as members.

Step 8 [VSAN] Check that the Storage Array and matching Storage_Arrays fabric replication ports re-login into the Transport/Replication fabrics and into the FC Name Server under the correct Replication VSANs.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that Fabric_Manager is able to configure all VSANs between hosts and storage arrays and between storage arrays in the replication ports without problems or issues.
- We expect no problems or issues with the configuration and verification of services' VSANs.
- We expect all created VSANs to be allowed and active in all Port-Channel / trunk ISLs / Fabric Extension links.

Results

Base Setup—VSANs EMC passed.

Base Setup—Zoning EMC

Host-to-Storage communication is the first most essential and basic service that a SAN must provide followed by Replication (Storage-to-Storage for Business Continuance). These services are made up of building blocks which include: VSAN ports membership, zone membership, zoneset activation, LUN masking, etc. This test verified the base Zoning configuration to enable communication between hosts (w/ multiple operating systems) and a storage arrays and between storage array pairs. Zones and Zonesets were configured and verified via Fabric_Manager (w/ CLI validation).

Test Procedure

The procedure used to perform the Base Setup—Zoning EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case.

Step 3 [Zone] Per-fabric: Create one(1) Windows_Hosts Zone for the Windows_Hosts VSAN. Add the Windows_Host and matching Storage_Arrays fabric ports to that Zone as members (i.e., two member zone).

Step 4 [Zone] Per-fabric: Create one(1) Linux_Hosts Zone for the Linux_Hosts VSAN. Add the Linux_Host and matching Storage_Arrays fabric ports to that Zone as members (i.e., two member zone).

Step 5 [Zone] Per-Replication-fabric: Create one(1) Sync-Replication Zone for the Sync-Replication VSAN. Add the Storage Array and matching Storage_Arrays fabric ports to that Zone as members (i.e., two member zone).

Step 6 [Zoneset] Per-fabric: Create a Hosts Zoneset and add the created Zones. Activate and distribute the Zoneset.

Step 7 [Zoneset] Per-Replication-fabric: Create a Replication Zoneset and add the created Zones. Activate and distribute the Zoneset.

Step 8 [Zoneset] Per-fabric: Verify Zoneset distribution and activation across the Fabric.

Step 9 [Zoneset] Per-fabric: Verify that all Zone members are active in the recently activated Zoneset.

Step 10 Verify that each Test-Host can see the matching LUN and NO other.

Step 11 Verify that each Storage Array can see the remote pair within the Replication services.

Step 12 Verify that each Test-Host's multi-pathing software can see the redundant paths available to it.

Step 13 Stop background scripts to collect final status of network devices and analyze for error.

Step 14 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that Fabric_Manager is able to configure all Zones between hosts and storage arrays and between storage arrays in the replication ports without problems or issues.
- We expect no problems or issues with the configuration and verification of services' Zones.
- We expect all Zone and Zone members to be active and all Zones distributed among nodes within the Fabrics.

Results

Base Setup—Zoning EMC passed.

Host To Storage IO Traffic—EMC

Host-to-Storage communications is based on IOs where the host reads from and writes to the LUNs in the storage array. This test verified the communication (i.e., I/Os) between hosts (w/ multiple operating systems) and a storage array. Traffic is generated tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by the CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- Iteration time : 5 minutes

Test Procedure

The procedure used to perform the Host To Storage IO Traffic—EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from a test hosts (Windows or Linux) to the corresponding non-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss on the interfaces and load balanced.

Step 5 Stop background scripts to collect final status of network devices and analyze for error.

Step 6 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport and delivery of all IO traffic between Test-Hosts and Storage-Array
- We expect for Fabric_Manager and the CLI to be able to present accurate link utilization.
- We expect a logical distribution between Read/Write ratios vs. IOPS.

Results

Host To Storage IO Traffic—EMC passed.

Replication FC Sync—EMC

Replication (synchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored real-time (IO is not complete until R2 acknowledges it). This test verified the basic functionality of sync replication between hosts (w/ multiple operating systems) and a storage array pair. Traffic is generated in multiple block sizes with tools like IOMETER and IORATE. Different Read/Write ratios were used. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- No FCWA, Encryption, Compression
- Iteration time : 5 minutes
- Distance : 0 Km

Test Procedure

The procedure used to perform the Replication FC Sync—EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case. [Pre-Test Condition #3] Host-To-Storage test successfully executed.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for the CLI and Fabric Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall out-of-sync due to MDS related issues.
- We expect for the IO delay statistics to be higher (i.e., longer delay) and less IOPS than the Host-To-Storage scenario.

Results

Replication FC Sync—EMC passed.

Replication FCIP ASync—EMC

Replication (Asynchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored in time offsets (IO is not complete after R1 acknowledges it but cached data is replicated to R2 in time intervals). This test verified the basic functionality of Async replication between hosts (w/ multiple operating systems) and a storage array pair. Traffic is generated in multiple block sizes with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- No FCIP-WA, Encryption, Compression
- Iteration time : 5 minutes

- Distance : 0 Km

Test Procedure

The procedure used to perform the Replication FCIP ASync—EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case. [Pre-Test Condition #3] Host-To-Storage test successfully executed.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for the CLI and Fabric Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fail their replication procedures due to MDS related issues.
- We expect for the IO delay statistics to be similar (i.e., similar delay) and IOPS to the Host-To-Storage scenario.

Results

Replication FCIP ASync—EMC passed.

A.4: Host to Storage Traffic—NetApp

Host-to-storage traffic tests for NetApp ensure hosts can access storage devices and that synchronous SnapMirror for synchronous replication and asynchronous SnapMirror for asynchronous replication are working properly.

The following tests were performed:

- Base Setup—VSANs NetApp, page 4-31
- Base Setup—Zoning NetApp, page 4-32
- Host To Storage IO Traffic—NetApp, page 4-33

- Replication FC-Sync—NetApp, page 4-34
- Replication FCIP-Async—NetApp, page 4-35

Base Setup—VSANs NetApp

Host-to-Storage communication is the first most essential and basic service that a SAN must provide followed by Replication (Storage-to-Storage for Business Continuance). These services are made up of building blocks which include: VSAN ports membership, zone membership, zoneset activation, LUN masking, etc. This test verified the basic configuration and activation of all VSANs needed for Host-to-Storage and Replication communication between hosts (w/ multiple operating systems), storage arrays, and storage array pair. VSANs were configured and verified via Fabric_Manager (w/ CLI validation).

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- No FCIP-WA, Encryption, Compression
- Iteration time : 5 minutes
- Distance : 0 Km

Test Procedure

The procedure used to perform the Base Setup—VSANs NetApp test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Two test hosts are selected each with a different operating system (Windows Enterprise 2003 Server and Linux RedHat Enterprise). Both hosts are dual-homed to two separate fabrics (as per test topology - Fabric A and B -OR- Fabric C and D). [Pre-Test Condition #2] Storage Arrays are dual-homed to the Host fabrics and to the Replication fabrics. [Pre-Test Condition #3] Storage Array's LUN Masking should be configured to allow access from the test hosts to the proper (non-replicating) LUNs. [Pre-Test Condition #4] Storage Array's Replication services must be enabled for sync and async replication between selected LUNs.

Step 3 [VSAN] Create one(1) Windows_Hosts VSAN per fabric. Add the Windows_Host and matching Storage_Arrays fabric ports to that VSAN as members.

Step 4 [VSAN] Check that Windows_Host and matching Storage_Arrays fabric ports re-login into the fabrics and into the FC Name Server under the correct Windows_Host VSAN.

Step 5 [VSAN] Create one(1) Linux_Hosts VSAN per fabric. Add the Linux_Host and matching Storage_Arrays fabric ports to that VSAN as members.

Step 6 [VSAN] Check that Linux_Host and matching Storage_Arrays fabric ports re-login into the fabrics and into the FC Name Server under the correct Linux_Host VSAN.

Step 7 [VSAN] Create two(2) Replication VSANs per Transport/Replication fabric. Add the Storage Array's fabric ports to those VSANs as members.

Step 8 [VSAN] Check that the Storage Array and matching Storage_Arrays fabric replication ports re-login into the Transport/Replication fabrics and into the FC Name Server under the correct Replication VSANs.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that Fabric_Manager is able to configure all VSANs between hosts and storage arrays and between storage arrays in the replication ports without problems or issues.
- We expect no problems or issues with the configuration and verification of services' VSANs.
- We expect all created VSANs to be allowed and active in all Port-Channel / trunk ISLs / Fabric Extension links.
- We expect nounacceptable impact on the CPU or memory.

Results

Base Setup—VSANs NetApp passed.

Base Setup—Zoning NetApp

Host-to-Storage communication is the first most essential and basic service that a SAN must provide followed by Replication (Storage-to-Storage for Business Continuance). These services are made up of building blocks which include: VSAN ports membership, zone membership, zoneset activation, LUN masking, etc. This test verified the base Zoning configuration to enable communication between hosts (w/ multiple operating systems) and a storage arrays and between storage array pairs. Zones and Zonesets were configured and verified via Fabric_Manager (w/ CLI validation).

Test Procedure

The procedure used to perform the Base Setup—Zoning NetApp test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case.

Step 3 [Zone] Per-fabric: Create one (1) Windows_Hosts Zone for the Windows_Hosts VSAN. Add the Windows_Host and matching Storage_Arrays fabric ports to that Zone as members (i.e., two member zone).

Step 4 [Zone] Per-fabric: Create one (1) Linux_Hosts Zone for the Linux_Hosts VSAN. Add the Linux_Host and matching Storage_Arrays fabric ports to that Zone as members (i.e., two member zone).

Step 5 [Zone] Per-Replication-fabric: Create one(1) Sync-Replication Zone for the Sync-Replication VSAN. Add the Storage Array and matching Storage_Arrays fabric ports to that Zone as members (i.e., two member zone).

Step 6 [Zoneset] Per-fabric: Create a Hosts Zoneset and add the created Zones. Activate and distribute the Zoneset.

Step 7 [Zoneset] Per-Replication-fabric: Create a Replication Zoneset and add the created Zones. Activate and distribute the Zoneset.

Step 8 [Zoneset] Per-fabric: Verify Zoneset distribution and activation across the Fabric.

Step 9 [Zoneset] Per-fabric: Verify that all Zone members are active in the recently activated Zoneset.

Step 10 Verify that each Test-Host can see the required LUNs.

Step 11 Verify that each Storage Array can see the remote pair within the Replication services.

Step 12 Verify that each Test-Host's multi-pathing software can see the redundant paths available to it.

Step 13 Stop background scripts to collect final status of network devices and analyze for error.

Step 14 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that Fabric_Manager is able to configure all Zones between hosts and storage arrays and between storage arrays in the replication ports without problems or issues.
- We expect no problems or issues with the configuration and verification of services' Zones.
- We expect all Zone and Zone members to be active and all Zones distributed among nodes within the Fabrics.
- We expect nounacceptable impact on the CPU or memory.

Results

Base Setup—Zoning NetApp passed.

Host To Storage IO Traffic—NetApp

Host-to-Storage communication is based on IOs where the host reads from and writes to the LUNs in the storage array. This test verified the communication (i.e., IOs) between hosts (w/ multiple operating systems) and a storage array. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with FM validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- Iteration time : 5 minutes

Test Procedure

The procedure used to perform the Host To Storage IO Traffic—NetApp test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding non-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Stop background scripts to collect final status of network devices and analyze for error.

Step 7 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport and delivery of all IO traffic between Test-Hosts and Storage-Array.
- We expect for the CLI and Fabric_Manager to be able to present accurate link utilization.
- We expect a logical distribution between Read/Write ratios vs. IOPS.
- We expect nounacceptable impact on the CPU or memory.

Results

Host To Storage IO Traffic—NetApp passed.

Replication FC-Sync—NetApp

Replication (synchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored real-time (IO is not complete until R2 acknowledges it). This test verified the basic functionality of sync replication between hosts (w/ multiple operating systems) and a storage array pair. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- No FCWA, Encryption, Compression
- Iteration time : 5 minutes
- Distance : 0 Km

Test Procedure

The procedure used to perform the Replication FC-Sync—NetApp test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case. [Pre-Test Condition #3] Host-To-Storage test successfully executed.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding sync-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for the CLI and Fabric Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall out-of-sync due to MDS related issues.
- We expect for the IO delay statistics to be higher (i.e., longer delay) and less IOPS than the Host-To-Storage scenario.
- We expect nounacceptable impact on the CPU or memory.

Results

Replication FC-Sync—NetApp passed.

Replication FCIP-Async—NetApp

Replication (Asynchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored in time offsets (IO is not complete after R1 acknowledges it but cached data is replicated to R2 in time intervals). This test verified the basic functionality of Async replication between hosts (w/ multiple operating systems) and a storage array pair. Traffic is generated in multiple block sizes with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- No FCIP-WA, Encryption, Compression
- Iteration time : 5 minutes
- Distance : 0 Km

Test Procedure

The procedure used to perform the Replication FCIP-Async—NetApp test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case. [Pre-Test Condition #3] Host-To-Storage test successfully executed.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding sync-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for the CLI and Fabric Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fail their replication procedures due to MDS related issues.
- We expect for the IO delay statistics to be similar (i.e., similar delay) and IOPS than the Host-To-Storage scenario.
- We expect nounacceptable impact on the CPU or memory.

Results

Replication FCIP-Async—NetApp passed.

Domain Parameters

Domain parameters testing ensures fiber channel domain configuration works as expected.

The following tests were performed:

- Device Access—CLI and Device Manager, page 4-18

Principal Switch Selection

The configuration and verification of Principal Switch Selection static parameters was tested. All configuration and verification was done via Fabric Manager with confirmation through CLI.

Test Procedure

The procedure used to perform the Principal Switch Selection test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Identify core Principal Switch within the target test Fabric.

Step 3 Configure non-Principal switch as the new Principal switch (configure higher priority).

Step 4 Verify Principal Switch configuration.

Step 5 Perform a Domain Restart to apply configuration.

Step 6 Verify new Principal Switch is active as Primary. Check that previous principal switch to be secondary.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that hard configuration of Principal Switches across the fabric is successful (no problems or issues).
- We expect that detection, reporting, validation, verification was successfully done with Fabric Manager with confirmation.
- We expect nounacceptable impact on the CPU or memory.

Results

Principal Switch Selection passed.

FSPF Functionality

FSPF functionality tests determine whether the Fabric Shortest Path First protocol properly handles load balancing, manual path cost configuration, and path failures and removals.

The following tests were performed:

- Basic FSPF Load Balancing, page 4-37
- Path Selection—Cost Change on Equal Cost Paths, page 4-38
- Primary Path Failure, page 4-39
- Primary Path Removal—VSAN Remove, page 4-40

Basic FSPF Load Balancing

The configuration and verification of FSPF parallel paths load balancing was tested. Redundant parallel paths with equal cost were configured for storage traffic to traverse. All configuration and verification was done via CLI.

Test Procedure

The procedure used to perform the Basic FSPF Load Balancing test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Identify target parallel paths of equal cost in the topology.

Step 3 Generate storage traffic (with random OXIDs) using the SANTester tool from an Edge MDS across parallel port-channels out the corresponding North or South MDS.

Step 4 Verify storage traffic is flowing without loss or problems over the available paths.

Step 5 Verify that storage traffic traversing the equal cost parallel paths is evenly distributed across them.

Step 6 Stop background scripts to collect final status of network devices and analyze for error.

Step 7 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that FSPF allow for storage traffic to be successfully load balanced over parallel paths of equal cost.
- We expect that detection, reporting, validation, verification was successfully done with Fabric Manager with CLI confirmation.

Results

Basic FSPF Load Balancing passed.

Path Selection—Cost Change on Equal Cost Paths

The FSPF's capability of changing priority or cost to paths with equal cost was tested. Redundant parallel paths with equal cost were configured so FSPF would select only one as the primary for storage traffic to traverse. All configuration and verification was done via Fabric Manager with confirmation through CLI.

Test Procedure

The procedure used to perform the Path Selection—Cost Change on Equal Cost Paths test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Identify target parallel paths of equal cost in the topology.

Step 3 Generate storage traffic (with random OXIDs) using the SANTester tool from an Edge MDS across parallel port-channels out the corresponding North or South MDS.

Step 4 Verify storage traffic is flowing without loss or problems over the available paths prior, during, and after the cost change.

Step 5 Verify that storage traffic traversing the equal cost parallel paths is evenly distributed across them.

Step 6 Change FSPF cost to one of the parallel equal cost paths to be higher than the other.

Step 7 Confirm that traffic changed from load balanced across the parallel paths to only traversing the path with the better metric.

Step 8 Confirm that storage traffic suffered no loss or problems during the path selection configuration.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that FSPF will select a path with better cost than another after cost was change between parallel and equal paths.
- We expect no traffic loss or problems after the path selection has been configured.

- We expect that detection, reporting, validation, verification was successfully done with CLI confirmation.

Results

Path Selection—Cost Change on Equal Cost Paths passed.

Primary Path Failure

This test verified FSPF's detection and re-routing of storage traffic after the primary path is shut down. Storage traffic was generated and traversing a primary path then link is physically disabled. All configuration and verification was done via CLI.

Test Procedure

The procedure used to perform the Primary Path Failure test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Identify target redundant paths in the topology (within a Fabric).

Step 3 Generate storage traffic (with random OXIDs) using the SANTester tool from an Edge MDS across parallel port-channels out the corresponding North or South MDS.

Step 4 Verify storage traffic is flowing without loss or problems over the available paths prior, during, and after the failure.

Step 5 Shutdown the link where the primary path traverses.

Step 6 Confirm the detection of the primary path loss and FSPF re-routed the traffic over the available redundant path.

Step 7 Confirm that storage traffic suffered no loss or problems during the path selection configuration.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that FSPF will detect the FAILURE(removal) of the primary path and will re-route the traffic over a secondary path.
- We expect no traffic loss or problems after the path selection has been configured.
- We expect that detection, reporting, validation, verification was successfully done with CLI confirmation.

Results

Primary Path Failure passed.

Primary Path Removal—VSAN Remove

This test verified FSPF's detection and re-routing of storage traffic after the VSAN was removed from the primary path. Storage traffic was generated and traversing a primary path - the VSAN was removed from the path. All configuration and verification was done via CLI.

Test Procedure

The procedure used to perform the Primary Path Removal—VSAN Remove test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Identify target redundant paths of unequal cost in the topology (within a Fabric).

Step 3 Generate storage traffic (with random OXIDs) using the SANTester tool from an Edge MDS across parallel port-channels out the corresponding North or South MDS.

Step 4 Verify storage traffic is flowing without loss or problems over the primary path and over the paths prior, during, and after the path removal.

Step 5 Remove test VSAN from ISL trunk which is the primary path.

Step 6 Confirm that FSPF detected the primary path and re-routed the traffic over the available redundant path.

Step 7 Confirm that storage traffic suffered no loss or problems during the path selection configuration.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that FSPF will detect the removal of the primary path and will re-route the traffic over a secondary path.
- We expect no traffic loss or problems after the path selection has been configured.
- We expect that detection, reporting, validation, verification was successfully done with CLI confirmation.

Results

Primary Path Removal—VSAN Remove passed.

Fabric Extension

Fabric extension tests check synchronous and asynchronous storage replication over the SAN topology. For each vendor, synchronous tests over FC and asynchronous tests over FCIP are performed with different capabilities enabled on the transit fabric.

The following test features were conducted:

- Async Replication—EMC, page 4-41
- Async Replication—NetApp, page 4-48

- Async Replication—HP, page 4-56
- FCIP COMP 100Km HP, page 4-56
- Sync Replication—NetApp, page 4-65
- Sync Replication—HP, page 4-69

Async Replication—EMC

Asynchronous replication test for EMC tests SRDF/A over FCIP without any advanced services, with just FCIP write acceleration, with just FCIP compression, with just FCIP encryption, and with all three advanced services at the same time.

The following tests were performed:

- FCIP COMP 100Km EMC, page 4-41
- FCIP ENCRP 100Km EMC, page 4-42
- FCIP NONE 100Km EMC, page 4-43
- FCIP WA 100Km EMC, page 4-44
- FCIP WA COMP ENCRP 100Km EMC, page 4-46
- FCIP Portchannel Failure 100Km EMC, page 4-47

FCIP COMP 100Km EMC

Replication (Asynchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored in time offsets (IO is not complete after R1 acknowledges it but cached data is replicated to R2 in time intervals). This test verified the basic functionality of Async replication between hosts (with multiple operating systems) and a storage array pair with the compression feature active in the MDS'es. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- Write-Acceleration : OFF
- Compression : ON
- Encryption : OFF
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FCIP COMP 100Km EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify Compression statistics to show the feature is operational.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect the CLI and Fabric Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fail their replication procedures due to MDS related issues.
- We expect for the IO BW statistics to improve with the use of the Compression feature.
- We expect no unacceptable impact on the CPU or memory.

Results

FCIP COMP 100Km EMC passed.

FCIP ENCRP 100Km EMC

Replication (Asynchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored in time offsets (IO is not complete after R1 acknowledges it but cached data is replicated to R2 in time intervals). This test verified the basic functionality of Async replication between hosts (with multiple operating systems) and a storage array pair with the IP-Encryption feature active in the MDS'es. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- Write-Acceleration : OFF
- Compression : OFF
- Encryption : ON
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FCIP ENCRP 100Km EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify IP-Encryption is operational.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall their replication procedures due to MDS related issues.
- We expect for the IP-Encryption feature not to have an adverse effect on the IO statistics.
- We expect no unacceptable impact on the CPU or memory.

Results

FCIP ENCRP 100Km EMC passed.

FCIP NONE 100Km EMC

Replication (Asynchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored in time offsets (IO is not complete after R1 acknowledges it but cached data is replicated to R2 in time intervals). This test verified the basic functionality of Async replication between hosts (with multiple operating systems) and a storage array pair. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- Write-Acceleration : OFF

- Compression : OFF
- Encryption : OFF
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FCIP NONE 100Km EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case. [Pre-Test Condition #3] Host-To-Storage test successfully executed.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fail their replication procedures due to MDS related issues.
- We expect for the IO delay statistics to be similar (i.e., similar delay) and IOPS than the Host-To-Storage scenario.
- We expect no unacceptable impact on the CPU or memory.

Results

FCIP NONE 100Km EMC passed.

FCIP WA 100Km EMC

Replication (Asynchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored in time offsets (IO is not complete after R1 acknowledges it but cached data is replicated to R2 in time intervals). This test verified the basic functionality of Async replication between hosts (w/ multiple operating systems) and a storage array pair with the Write-Acceleration feature active in the MDS'es. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs

and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by Fabric_Manager (w/ CLI validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- Write-Acceleration : ON
- Compression : OFF
- Encryption : OFF
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FCIP WA 100Km EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify FCIP-WA statistics to show the feature is operational.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall their replication procedures due to MDS related issues.
- We expect for the IO delay statistics to be improve with the use of the FCIP Write Acceleration feature.
- We expect no unacceptable impact on the CPU or memory.

Results

FCIP WA 100Km EMC passed.

FCIP WA COMP ENCRP 100Km EMC

Replication (Asynchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored in time offsets (IO is not complete after R1 acknowledges it but cached data is replicated to R2 in time intervals). This test verified the basic functionality of Async replication between hosts (with multiple operating systems) and a storage array pair with Write-Acceleration, Compression, and IP-Encryption features active in the MDS'es. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- Write-Acceleration : ON
- Compression : ON
- Encryption : ON
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FCIP WA COMP ENCRP 100Km EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify FCIP-WA, Compression, and Encryption are operational.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fail their replication procedures due to MDS related issues.

- We expect for the IO delay statistics to improve with the use of the FCIP Write Acceleration feature.
- We expect for the IO BW statistics to improve with the use of the Compression feature.
- We expect for the IP-Encryption feature not to have an adverse effect on the IO statistics.
- We expect for the combination of WA, Compression, and Encryption not to have an negative effect on FCIPs functionality or the storage traffic delivery.
- We expect no unacceptable impact on the CPU or memory.

Results

FCIP WA COMP ENCRP 100Km EMC passed.

FCIP Portchannel Failure 100Km EMC

This test verified the resilience of the Fabric Extension network (over IP) when one Port-Channel link fails while Async replication is active. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- Write-Acceleration : ON
- Compression : ON
- Encryption : ON
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FCIP Portchannel Failure 100Km EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding async-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify FCIP-WA, Compression, and Encryption are operational.

Step 8 Fail one link of the FCIP Port-Channel towards the Transport (IP Fabric Extension) Fabric.

Step 9 Confirm that the Port-Channel link is down. Verify that the failure is detected and reported by the nodes to the management applications.

Step 10 Verify that replication traffic is traversing the remaining port-channel link.

Step 11 Re-establish the port-channel failed link.

Step 12 Verify failed link is re-established as a member of the port-channel and that the recovery was detected and reported to the management applications.

Step 13 Verify that replication traffic is load balanced across the port-channel including the recovered link.

Step 14 Verify the storage arrays' asynchronous replication state throughout the failure and recovery. State of "consistent" is normal.

Step 15 Stop background scripts to collect final status of network devices and analyze for error.

Step 16 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fail Async replication due to MDS related issues like the Fabric Extension port-channel failure.
- We expect for the Port-Channel link failure to be detected and reported to the management applications.
- We expect no unacceptable impact on the CPU or memory.

Results

FCIP Portchannel Failure 100Km EMC passed.

Async Replication—NetApp

Asynchronous replication test for NetApp tests asynchronous SnapMirror over FCIP without any advanced services, with just FCIP write acceleration, with just FCIP compression, with just FCIP encryption, and with all three advanced services at the same time.

The following tests were performed:

- FCIP COMP 100Km NETAPP, page 4-48
- FCIP ENCRP 100Km NETAPP, page 4-50
- FCIP NONE 100Km NETAPP, page 4-51
- FCIP WA 100Km NETAPP, page 4-52
- FCIP WA COMP ENCRP 100Km NETAPP, page 4-53
- FCIP Portchannel Failure 100Km NETAPP, page 4-54

FCIP COMP 100Km NETAPP

Replication (Asynchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored in time offsets (IO is not complete after R1 acknowledges it but cached data is replicated to R2 in time intervals). This test verified the basic functionality of Async replication between hosts (with multiple operating systems) and a storage array pair with the Compression feature active in the

MDS'es. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by Fabric_Manager (w/ CLI validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 4K, 8K, 16K, 32K
- Write-Acceleration : OFF
- Compression : ON
- Encryption : OFF
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FCIP COMP 100Km NETAPP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify Compression statistics to show the feature is operational.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall their replication procedures due to MDS related issues.
- We expect for the IO BW statistics to improve with the use of the Compression feature.
- We expect no unacceptable impact on the CPU or memory.

Results

FCIP COMP 100Km NETAPP passed.

FCIP ENCRP 100Km NETAPP

Replication (Asynchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored in time offsets (IO is not complete after R1 acknowledges it but cached data is replicated to R2 in time intervals). This test verified the basic functionality of Async replication between hosts (with multiple operating systems) and a storage array pair with the IP-Encryption feature active in the MDS'es. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- Write-Acceleration : OFF
- Compression : OFF
- Encryption : ON
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FCIP ENCRP 100Km NETAPP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify IP-Encryption is operational.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall their replication procedures due to MDS related issues.

- We expect for the IP-Encryption feature not to have an adverse effect on the IO statistics.
- We expect no unacceptable impact on the CPU or memory.

Results

FCIP ENCRP 100Km NETAPP passed.

FCIP NONE 100Km NETAPP

Replication (Asynchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored in time offsets (IO is not complete after R1 acknowledges it but cached data is replicated to R2 in time intervals). This test verified the basic functionality of Async replication between hosts (with multiple operating systems) and a storage array pair. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- Write-Acceleration : OFF
- Compression : OFF
- Encryption : OFF
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FCIP NONE 100Km NETAPP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case. [Pre-Test Condition #3] Host-To-Storage test successfully executed.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager and CLI to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fail their replication procedures due to MDS related issues.
- We expect for the IO delay statistics to be similar (i.e., similar delay) and IOPS than the Host-To-Storage scenario.
- We expect no unacceptable impact on the CPU or memory.

Results

FCIP NONE 100Km NETAPP passed.

FCIP WA 100Km NETAPP

Replication (Asynchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored in time offsets (IO is not complete after R1 acknowledges it but cached data is replicated to R2 in time intervals). This test verified the basic functionality of Async replication between hosts (w/ multiple operating systems) and a storage array pair with the Write-Acceleration feature active in the MDS'es. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by Fabric_Manager (w/ CLI validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- Write-Acceleration : ON
- Compression : OFF
- Encryption : OFF
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FCIP WA 100Km NETAPP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify FCIP-WA statistics to show the feature is operational.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall their replication procedures due to MDS related issues.
- We expect for the IO delay statistics to be improve with the use of the FCIP Write Acceleration feature.
- We expect no unacceptable impact on the CPU or memory.

Results

FCIP WA 100Km NETAPP passed.

FCIP WA COMP ENCRP 100Km NETAPP

Replication (Asynchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored in time offsets (IO is not complete after R1 acknowledges it but cached data is replicated to R2 in time intervals). This test verified the basic functionality of Async replication between hosts (w/ multiple operating systems) and a storage array pair with Write-Acceleration, Compression, and IP-Encryption features active in the MDS'es. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- Write-Acceleration : ON
- Compression : ON
- Encryption : ON
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FCIP WA COMP ENCRP 100Km NETAPP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify FCIP-WA, Compression, and Encryption are operational.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fail their replication procedures due to MDS related issues.
- We expect for the IO delay statistics to be improve with the use of the FCIP Write Acceleration feature.
- We expect for the IO BW statistics to improve with the use of the Compression feature.
- We expect for the IP-Encryption feature not to have an adverse effect on the IO statistics.
- We expect for the combination of WA, Compression, and Encryption not to have an negative effect on FCIPs functionality or the storage traffic delivery.
- We expect no unacceptable impact on the CPU or memory.

Results

FCIP WA COMP ENCRP 100Km NETAPP passed.

FCIP Portchannel Failure 100Km NETAPP

This test verified the resilience of the Fabric Extension network (over IP) when one Port-Channel link fails while Async replication is active. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- Write-Acceleration : ON
- Compression : ON
- Encryption : ON

- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FCIP Portchannel Failure 100Km NETAPP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding async-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify FCIP-WA, Compression, and Encryption are operational.

Step 8 Fail one link off the FCIP Port-Channel towards the Transport (IP Fabric Extension) Fabric.

Step 9 Confirm that the Port-Channel link is down. Verify that the failure is detected and reported by the nodes to the management applications.

Step 10 Verify that replication traffic is traversing the remaining port-channel link.

Step 11 Re-establish the port-channel failed link.

Step 12 Verify failed link is re-established as a member of the port-channel and that the recovery was detected and reported to the management applications.

Step 13 Verify that replication traffic is load balanced across the port-channel including the recovered link.

Step 14 Verify the storage arrays' asynchronous replication state throughout the failure and recovery. States of "Transferring" or "Idle" are normal.

Step 15 Stop background scripts to collect final status of network devices and analyze for error.

Step 16 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fail Async replication due to MDS related issues like the Fabric Extension port-channal failure.
- We expect for the Port-Channel link failure to be detected and reported to the management applications.
- We expect no unacceptable impact on the CPU or memory.

Results

FCIP Portchannel Failure 100Km NETAPP passed.

Async Replication—HP

Asynchronous replication test for HP tests HP XP Continuous Access Asynchronous replication over FCIP without any advanced services, with just FCIP write acceleration, with just FCIP compression, with just FCIP encryption, and with all three advanced services at the same time.

The following tests were performed:

- FCIP COMP 100Km HP, page 4-56
- FCIP ENCRP 100Km HP, page 4-57
- FCIP NONE 100Km HP, page 4-58
- FCIP WA 100Km HP, page 4-59
- FCIP WA COMP ENCRP 100Km HP, page 4-61

FCIP COMP 100Km HP

Replication (Asynchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored in time offsets (IO is not complete after R1 acknowledges it but cached data is replicated to R2 in time intervals). This test verified the basic functionality of Async replication between hosts (with multiple operating systems) and a storage array pair with the Compression feature active in the MDS'es. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by Fabric_Manager (w/ CLI validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 4K, 8K, 16K, 32K
- Write-Acceleration : OFF
- Compression : ON
- Encryption : OFF
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FCIP COMP 100Km HP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify Compression statistics to show the feature is operational.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall their replication procedures due to MDS related issues.
- We expect for the IO BW statistics to improve with the use of the Compression feature.
- We expect no unacceptable impact on the CPU or memory.

Results

FCIP COMP 100Km HP passed.

FCIP ENCRP 100Km HP

Replication (Asynchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored in time offsets (IO is not complete after R1 acknowledges it but cached data is replicated to R2 in time intervals). This test verified the basic functionality of Async replication between hosts (with multiple operating systems) and a storage array pair with the IP-Encryption feature active in the MDS'es. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- Write-Acceleration : OFF
- Compression : OFF
- Encryption : ON
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FCIP ENCRP 100Km HP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify IP-Encryption is operational.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall their replication procedures due to MDS related issues.
- We expect for the IP-Encryption feature not to have an adverse effect on the IO statistics.
- We expect no unacceptable impact on the CPU or memory.

Results

FCIP ENCRP 100Km HP passed.

FCIP NONE 100Km HP

Replication (Asynchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored in time offsets (IO is not complete after R1 acknowledges it but cached data is replicated to R2 in time intervals). This test verified the basic functionality of Async replication between hosts (with multiple operating systems) and a storage array pair. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- Write-Acceleration : OFF
- Compression : OFF
- Encryption : OFF
- Iteration time : 5 minutes

- Distance : 100 Km

Test Procedure

The procedure used to perform the FCIP NONE 100Km HP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case. [Pre-Test Condition #3] Host-To-Storage test successfully executed.

Step 3 Generate host IO traffic to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager and CLI to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fail their replication procedures due to MDS related issues.
- We expect for the IO delay statistics to be similar (i.e., similar delay) and IOPS than the Host-To-Storage scenario.
- We expect no unacceptable impact on the CPU or memory.

Results

FCIP NONE 100Km HP passed.

FCIP WA 100Km HP

Replication (Asynchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored in time offsets (IO is not complete after R1 acknowledges it but cached data is replicated to R2 in time intervals). This test verified the basic functionality of Async replication between hosts (w/ multiple operating systems) and a storage array pair with the Write-Acceleration feature active in the MDS'es. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by Fabric_Manager (w/ CLI validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- Write-Acceleration : ON
- Compression : OFF
- Encryption : OFF
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FCIP WA 100Km HP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify FCIP-WA statistics to show the feature is operational.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall their replication procedures due to MDS related issues.
- We expect for the IO delay statistics to be improve with the use of the FCIP Write Acceleration feature.
- We expect no unacceptable impact on the CPU or memory.

Results

FCIP WA 100Km HP passed.

FCIP WA COMP ENCRP 100Km HP

Replication (Asynchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored in time offsets (IO is not complete after R1 acknowledges it but cached data is replicated to R2 in time intervals). This test verified the basic functionality of Async replication between hosts (w/ multiple operating systems) and a storage array pair with Write-Acceleration, Compression, and IP-Encryption features active in the MDS'es. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- Write-Acceleration : ON
- Compression : ON
- Encryption : ON
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FCIP WA COMP ENCRP 100Km HP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate host IO traffic to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify FCIP-WA, Compression, and Encryption are operational.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fail their replication procedures due to MDS related issues.

- We expect for the IO delay statistics to be improve with the use of the FCIP Write Acceleration feature.
- We expect for the IO BW statistics to improve with the use of the Compression feature.
- We expect for the IP-Encryption feature not to have an adverse effect on the IO statistics.
- We expect for the combination of WA, Compression, and Encryption not to have an negative effect on FCIPs functionality or the storage traffic delivery.
- We expect no unacceptable impact on the CPU or memory.

Results

FCIP WA COMP ENCRP 100Km HP passed.

Sync Replication—EMC

Synchronous replication test for EMC tests SRDF/S with and without FC write acceleration.

The following tests were performed:

- FC Sync—DST=100Km, WA=OFF - EMC, page 4-62
- FC Sync—DST=100Km, WA=ON - EMC, page 4-63
- FC Sync—Portchannel Failure, DST=100Km - EMC, page 4-64

FC Sync—DST=100Km, WA=OFF - EMC

Replication (synchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored real-time (IO is not complete until R2 acknowledges it). This test verified the basic functionality of sync replication between hosts (w/ multiple operating systems) and a storage array pair. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- FCWA : OFF
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FC Sync—DST=100Km, WA=OFF - EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall out-of-sync due to MDS related issues.
- We expect for the IO delay statistics to be higher (i.e., longer delay) and less IOPS than the Host-To-Storage scenario.
- We expect no unacceptable impact on the CPU or memory.

Results

FC Sync—DST=100Km, WA=OFF - EMC passed.

FC Sync—DST=100Km, WA=ON - EMC

Replication (synchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored real-time (IO is not complete until R2 acknowledges it). This test verified the basic functionality of sync replication between hosts (with multiple operating systems) and a storage array pair with the Write-Acceleration feature active. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- FCWA : ON
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FC Sync—DST=100Km, WA=ON - EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify FCWA feature is operational.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall out-of-sync due to MDS related issues.
- We expect for the IO delay statistics to be lower than the non-FCWA scenario at the same distance.
- We expect no unacceptable impact on the CPU or memory.

Results

FC Sync—DST=100Km, WA=ON - EMC passed.

FC Sync—Portchannel Failure, DST=100Km - EMC

This test verified the resilience of the Fabric Extension network when one Port-Channel link fails while sync replication is active. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by Fabric_Manager (w/ CLI validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- FCWA : ON
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FC Sync—Portchannel Failure, DST=100Km - EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify FCWA statistics to show the feature is operational.

Step 8 Fail one link off the FC Port-Channel within the Transport (Fabric Extension) Fabric.

Step 9 Confirm that the Port-Channel link is down. Verify that the failure is detected and reported by the nodes to the management applications.

Step 10 Verify that replication traffic is traversing the remaining port-channel link.

Step 11 Re-establish the port-channel failed link.

Step 12 Verify failed link is re-established as a member of the port-channel and that the recovery was detected and reported to the management applications.

Step 13 Verify that replication traffic is load balanced across the port-channel including the recovered link.

Step 14 Verify that storage arrays remained in synch throughout the failure and recovery. No traffic loss during each iteration within the test.

Step 15 Stop background scripts to collect final status of network devices and analyze for error.

Step 16 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall out-of-sync due to MDS related issues like the Fabric Extension port-channal failure.
- We expect for the Port-Channel link failure to be detected and reported to the management applications.
- We expect no unacceptable impact on the CPU or memory.

Results

FC Sync—Portchannel Failure, DST=100Km - EMC passed.

Sync Replication—NetApp

Synchronous replication test for NetApp tests synchronous SnapMirror with and without FC write acceleration.

The following tests were performed:

- FC Sync—DST=100Km, WA=OFF - NetApp, page 4-66
- FC Sync—DST=100Km, WA=ON - NetApp, page 4-67
- FC Sync—Portchannel Failure, DST=100Km - NetApp, page 4-68

FC Sync—DST=100Km, WA=OFF - NetApp

Replication (synchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored real-time (IO is not complete until R2 acknowledges it). This test verified the basic functionality of sync replication between hosts (with multiple operating systems) and a storage array pair. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by Fabric_Manager (w/ CLI validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- FCWA : OFF
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FC Sync—DST=100Km, WA=OFF - NetApp test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall out-of-sync due to MDS related issues.

- We expect for the IO delay statistics to be higher (i.e., longer delay) and less IOPS than the Host-To-Storage scenario.
- We expect no unacceptable impact on the CPU or memory.

Results

FC Sync—DST=100Km, WA=OFF - NetApp passed.

FC Sync—DST=100Km, WA=ON - NetApp

Replication (synchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored real-time (IO is not complete until R2 acknowledges it). This test verified the basic functionality of sync replication between hosts (with multiple operating systems) and a storage array pair with the Write-Acceleration feature active. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- FCWA : ON
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FC Sync—DST=100Km, WA=ON - NetApp test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify FCWA feature is operational.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.

- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall out-of-sync due to MDS related issues.
- We expect for the IO delay statistics to be lower than the non-FCWA scenario at the same distance.
- We expect no unacceptable impact on the CPU or memory.

Results

FC Sync—DST=100Km, WA=ON - NetApp passed.

FC Sync—Portchannel Failure, DST=100Km - NetApp

This test verified the resilience of the Fabric Extension network when one Port-Channel link fails while sync replication is active. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by Fabric_Manager (w/ CLI validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- FCWA : ON
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FC Sync—Portchannel Failure, DST=100Km - NetApp test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate IO traffic from both hosts (Windows and Linux) to the corresponding synch-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify FCWA statistics to show the feature is operational.

Step 8 Fail one link off the FC Port-Channel within the Transport (Fabric Extension) Fabric.

Step 9 Confirm that the Port-Channel link is down. Verify that the failure is detected and reported by the nodes to the management applications.

Step 10 Verify that replication traffic is traversing the remaining port-channel link.

Step 11 Re-establish the port-channel failed link.

Step 12 Verify failed link is re-established as a member of the port-channel and that the recovery was detected and reported to the management applications.

Step 13 Verify that replication traffic is load balanced across the port-channel including the recovered link.

Step 14 Verify that storage arrays remained in synch throughout the failure and recovery. No traffic loss during each iteration within the test.

Step 15 Stop background scripts to collect final status of network devices and analyze for error.

Step 16 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall out-of-sync due to MDS related issues like the Fabric Extension port-channal failure.
- We expect for the Port-Channel link failure to be detected and reported to the management applications.
- We expect no unacceptable impact on the CPU or memory.

Results

FC Sync—Portchannel Failure, DST=100Km - NetApp passed.

Sync Replication—HP

Synchronous replication test for HP tests HP XP Continuous Access Synchronous replication with and without FC write acceleration.

The following tests were performed:

- FC Sync—DST=100Km, WA=OFF - HP, page 4-69
- FC Sync—DST=100Km, WA=ON - HP, page 4-70

FC Sync—DST=100Km, WA=OFF - HP

Replication (synchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored real-time (IO is not complete until R2 acknowledges it). This test verified the basic functionality of sync replication between hosts (with multiple operating systems) and a storage array pair. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by Fabric_Manager (w/ CLI validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- FCWA : OFF
- Iteration time : 5 minutes
- Distance : 100 Km

Test Procedure

The procedure used to perform the FC Sync—DST=100Km, WA=OFF - HP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate host IO traffic to the corresponding sync-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall out-of-sync due to MDS related issues.
- We expect for the IO delay statistics to be higher (i.e., longer delay) and less IOPS than the Host-To-Storage scenario.
- We expect no unacceptable impact on the CPU or memory.

Results

FC Sync—DST=100Km, WA=OFF - HP passed.

FC Sync—DST=100Km, WA=ON - HP

Replication (synchronous) services is based on IOs from hosts to array-R1 to array-R2 where LUNs are mirrored real-time (IO is not complete until R2 acknowledges it). This test verified the basic functionality of sync replication between hosts (with multiple operating systems) and a storage array pair with the Write-Acceleration feature active. Traffic is generated with tools like IOMETER and IORATE. All test traffic ran over the VSANs and Zones already configured and tested in previous tests. The traffic statistics (IO Delay, IO per second) were observed, validated, and collected by CLI (with Fabric_Manager validation) and the test tools. Traffic Generation Characteristics:

- Read/Write ratios : 0/100
- Block Sizes per r/w ratio : 8K
- FCWA : ON
- Iteration time : 5 minutes

- Distance : 100 Km

Test Procedure

The procedure used to perform the FC Sync—DST=100Km, WA=ON - HP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Base VSANs configuration has been executed and validated in the 'Base Setup - VSANs' test case. [Pre-Test Condition #2] Base Zones configuration has been executed and validated in the "Base Setup - Zoning' test case.

Step 3 Generate host IO traffic to the corresponding sync-replicated LUNs using the traffic characteristics defined in this test case.

Step 4 Verify (using Fabric_Manager and CLI) that traffic is flowing without loss.

Step 5 Verify that the hosts are making use of the dual-paths.

Step 6 Verify that the Storage-Array pair is replicating the LUNs without problems or issues.

Step 7 Verify FCWA feature is operational.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect complete transport, delivery, and replication of all IO traffic between Test-Hosts and Storage-Array pair for all block size iterations within the Read/Write ratio.
- We expect for Fabric_Manager to be able to present accurate link utilization.
- We expect for the Storage-Array pair not to fall out-of-sync due to MDS related issues.
- We expect for the IO delay statistics to be lower than the non-FCWA scenario at the same distance.
- We expect no unacceptable impact on the CPU or memory.

Results

FC Sync—DST=100Km, WA=ON - HP passed.

Inter-VSAN Routing Functionality

Inter-VSAN (IVR) routing functionality tests make sure IVR works as expected both with and without NAT.

The following tests were performed:

- Basic IVR Implementation, page 4-72
- Basic IVR-NAT Implementation, page 4-72

Basic IVR Implementation

IVR basic Inter-VSAN routing functionality was tested. Traffic generation generated across the Transport Fabrics (transit) in other VSANs. All configuration done using Fabric Manager and verification was done via CLI.

Test Procedure

The procedure used to perform the Basic IVR Implementation test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Fibre Channel Connectivity between DC-A and DC-B over the transport core most be active.

Step 3 In the SRC fabric configure IVR-Zones to route between the Test-Host VSAN and the Transit VSAN. Configure the DST fabric with IVR-Zones to route between the Transit VSAN and the storage array VSAN.

Step 4 Verify creation and activation of IVR-Zones. Check that Test-Hosts and storage arrays are active in the IVR-zone.

Step 5 Generate traffic using the SANTester tool from Edge in DC-A to Edge in DC-B. Traffic must use random OX_IDs.

Step 6 Verify that storage traffic is delivered without loss or problems to the remote storage array over IVR.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that IVR will successfully route storage traffic from the Source VSAN over the Transit/Transport Fabric (transit vsan) to the remote Fabric (destination).
- We expect no loss of traffic or problems with the Inter-VSAN routing.

Results

Basic IVR Implementation passed.

Basic IVR-NAT Implementation

IVR-NAT basic Inter-VSAN routing functionality was tested between Nodes/VSANs with same Domain ID. Fiber Channel traffic generation was configured to communicate from a source device to a destination device in different VSANs with same Domain-IDs. All configuration was done via Fabric Manager with validation through CLI.

Test Procedure

The procedure used to perform the Basic IVR-NAT Implementation test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Configure a source VSAN X (Domain ID A) in an Edge switch and trunk it across the port-channel to the corresponding Core switch. Configure a destination VSAN Y (Domain ID A) in the corresponding Core switch.

Step 3 Configure IVR-NAT to route VSAN X traffic to VSAN Y on the Core switch (IVR border switch).

Step 4 Verify creation and activation of IVR-NAT Zones. Check that test devices are active in the IVR-NAT zones.

Step 5 Generate traffic using the SANTester tool from Edge to Core. Traffic must use random OX_IDs.

Step 6 Verify that storage traffic is delivered without loss or problems to the remote storage array over IVR-NAT.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that IVR-NAT will successfully route Fiber Channel traffic from the Source VSAN over the to the Destination VSAN when both have the same Domain ID.
- We expect no loss of traffic or problems with the Inter-VSAN routing.
- We expect that detection, reporting, validation, verification was successfully done with CLI confirmation.

Results

Basic IVR-NAT Implementation passed.

Portchannel Functionality

Portchannel functionality tests look at whether the port channel protocol correctly load balances, allows link additions and removals, and handles link failures.

The following tests were performed:

- Basic Portchannel Load Balancing, page 4-74
- Multiple Link ADD to Group, page 4-74
- Multiple Links Failure in Group, page 4-75
- Multiple Links Remove to Group, page 4-76
- Single Link Add to Group, page 4-77
- Single Link Failure in Group, page 4-78
- Single Link Remove from Group, page 4-79

Basic Portchannel Load Balancing

This test verified the Port-Channel's load-balancing capability (based on OXID) across the 12xISL inter-fabric channels. Storage traffic was generated to cross the Port-Channels. All configuration and verification was done via CLI.

Test Procedure

The procedure used to perform the Basic Portchannel Load Balancing test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] All Port-Channels within the Fabrics must be configured and active.

Step 3 Generate storage traffic (with random OXIDs) using the SANTester tool from an Edge MDS across a 12xISL port-channel out a Core MDS.

Step 4 Verify storage traffic is flowing without loss or problems over Port-Channel(s).

Step 5 Verify that storage traffic traversing one of the 12xISL Intra-Fabric Port-Channels is evenly distributed across all channel members.

Step 6 Stop background scripts to collect final status of network devices and analyze for error.

Step 7 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that storage traffic is evenly distributed across all members of the Port-Channels without loss or problems (based on OXID).
- We expect no unacceptable impact on the CPU or memory.

Results

Basic Portchannel Load Balancing passed.

Multiple Link ADD to Group

This test verified the support for the addition of multiple links to an active Port-Channel group without disrupting active traffic or services over the channel. Storage traffic was generated to cross the Port-Channels prior, during, and after the links were added. All configuration and verification was done via Fabric Manager with confirmation through CLI.

Test Procedure

The procedure used to perform the Multiple Link ADD to Group test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Remove 6 member links from one of the intra-fabric port-channels (via configuration) - target port-channel must have 6xISL members after removal.

Step 3 Generate storage traffic (with random OXIDs) using the SANTester tool from an Edge MDS across a 11xISL port-channel out a Core MDS.

Step 4 Verify storage traffic is flowing without loss or problems over Port-Channel(s).

Step 5 Verify that storage traffic traversing the target 6xISL Intra-Fabric Port-Channel is evenly distributed across all channel members.

Step 6 Execute a 'Force Interface Addition' of 6 links to the target port-channel.

Step 7 Verify that the newly added port-channel members becomes active in the group. The addition must be detected and reported to the management applications (e.g., syslog messages).

Step 8 Verify that storage traffic traversing the target 12xISL Intra-Fabric Port-Channel is evenly distributed across all channel members including the new members.

Step 9 Confirm that storage traffic traversing the target port-channel was not affected during or after the addition of the single link to the group.

Step 10 Stop background scripts to collect final status of network devices and analyze for error.

Step 11 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that the multiple link addition/integration is executed successfully without errors, issues, or problems.
- We expect that the multiple link addition didn't affect active storage traffic over the Port-Channel.
- We expect for the active storage traffic to be load-balanced over the newly added ports.
- We expect no unacceptable impact on the CPU or memory.

Results

Multiple Link ADD to Group passed.

Multiple Links Failure in Group

This test verified the that the physical failure of multiple links in an active Port-Channel group does not disrupt active traffic or services over the channel. Storage traffic was generated to cross the Port-Channels prior, during, and after the link failure. All configuration and verification was done via CLI.

Test Procedure

The procedure used to perform the Multiple Links Failure in Group test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Target test port-channel must be an intra-fabric 12xISL member group.

Step 3 Generate storage traffic (with random OXIDs) using the SANTester tool from an Edge MDS across a 12xISL port-channel out a Core MDS.

Step 4 Verify storage traffic is flowing without loss or problems over Port-Channel(s) (prior, during, and after the failure).

Step 5 Verify that storage traffic traversing the target 12xISL Intra-Fabric Port-Channel is evenly distributed across all channel members.

Step 6 Physically remove (fiber pull) a 6 members off the target port-channel.

Step 7 Verify that the removed port-channel members becomes inactive in the group. The removal must be detected and reported to the management applications (e.g., syslog messages).

Step 8 Verify that storage traffic traversing the target 6xISL Intra-Fabric Port-Channel is evenly distributed across all remaining channel members.

Step 9 Confirm that storage traffic traversing the target port-channel was not affected during or after the removal of multiple links from the group.

Step 10 Stop background scripts to collect final status of network devices and analyze for error.

Step 11 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that a multiple links failure doesn't affect active storage traffic over the Port-Channel.
- We expect for the failure to be detected and reported by the nodes to the management applications/servers.
- We expect for the active storage traffic to be load-balanced over the remaining port-channel members.
- We expect no unacceptable impact on the CPU or memory.

Results

Multiple Links Failure in Group passed.

Multiple Links Remove to Group

This test verified the support for the removal of multiple links from an active Port-Channel group without disrupting active traffic or services over the channel. Storage traffic was generated to cross the Port-Channels prior, during, and after the link was removed. All configuration and verification was done via CLI.

Test Procedure

The procedure used to perform the Multiple Links Remove to Group test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Target test port-channel must be an intra-fabric 12xISL member group.

Step 3 Generate traffic using the SANTester tool from Edge to Core device over the 12xISL port-channel. Traffic must use random OX_IDs.

Step 4 Verify storage traffic is flowing without loss or problems over Port-Channel(s) - before, during, and after the port removal.

Step 5 Verify that storage traffic traversing the target 12xISL Intra-Fabric Port-Channel is evenly distributed across all channel members.

Step 6 Remove (via configuration) 6 members off the target port-channel.

Step 7 Verify that the removed port-channel members becomes inactive in the group. The removal must be detected and reported to the management applications (e.g., syslog messages).

Step 8 Verify that storage traffic traversing the target 6xISL Intra-Fabric Port-Channel is evenly distributed across all channel members.

Step 9 Confirm that storage traffic traversing the target port-channel was not affected during or after the removal of multiple links from the group.

Step 10 Stop background scripts to collect final status of network devices and analyze for error.

Step 11 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that a multiple links removal is executed successfully without errors, issues, or problems.
- We expect that the multiple links removal didn't affect active storage traffic over the Port-Channel.
- We expect for the active storage traffic to be load-balanced over the remaining port-channel members.
- We expect no unacceptable impact on the CPU or memory.

Results

Multiple Links Remove to Group passed.

Single Link Add to Group

This test verified the support for the addition of a single link to an active Port-Channel group without disrupting active traffic or services over the channel. Storage traffic was generated to cross the Port-Channels prior, during, and after the link was added. All configuration and verification was done via Fabric Manager with confirmation through CLI.

Test Procedure

The procedure used to perform the Single Link Add to Group test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Remove a member link from one of the intra-fabric port-channels (via configuration) - target port-channel must have 11xISL members after removal.

Step 3 Generate storage traffic (with random OXIDs) using the SANTester tool from an Edge MDS across a 11xISL port-channel out a Core MDS.

Step 4 Verify storage traffic is flowing without loss or problems over Port-Channel(s) - before, during, and after the port integration.

Step 5 Verify that storage traffic traversing the target 11xISL Intra-Fabric Port-Channel is evenly distributed across all channel members.

Step 6 Add a single additional port to the port-channel (force).

Step 7 Verify that the newly added port-channel member becomes active in the group. The addition must be detected and reported to the management applications (e.g., syslog messages).

Step 8 Verify that storage traffic traversing the target 12xISL Intra-Fabric Port-Channel is evenly distributed across all channel members including the new member.

Step 9 Confirm that storage traffic traversing the target port-channel was not affected during or after the addition of the single link to the group.

Step 10 Stop background scripts to collect final status of network devices and analyze for error.

Step 11 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that the single link addition/integration is executed successfully without errors, issues, or problems.
- We expect that the single link addition didn't affect active storage traffic over the Port-Channel.
- We expect for the active storage traffic to be load-balanced over the newly added port.

Results

Single Link Add to Group passed.

Single Link Failure in Group

This test verified the that the physical failure of a single link in an active Port-Channel group does not disrupt active traffic or services over the channel. Storage traffic was generated to cross the Port-Channels prior, during, and after the link failure. All configuration and verification was done via CLI.

Test Procedure

The procedure used to perform the Single Link Failure in Group test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Target test port-channel must be an intra-fabric 12xISL member group.

Step 3 Generate storage traffic (with random OXIDs) using the SANTester tool from an Edge MDS across a 12xISL port-channel out a Core MDS.

Step 4 Verify storage traffic is flowing without loss or problems over Port-Channel(s).

Step 5 Verify that storage traffic traversing the target 12xISL Intra-Fabric Port-Channel is evenly distributed across all channel members.

Step 6 Physically remove (fiber pull) a member off the target port-channel.

Step 7 Verify that the removed port-channel member becomes inactive in the group. The removal must be detected and reported to the management applications (e.g., snmp-traps, syslog messages).

Step 8 Verify that storage traffic traversing the target 11xISL Intra-Fabric Port-Channel is evenly distributed across all remaining channel members.

Step 9 Confirm that storage traffic traversing the target port-channel was not affected during or after the removal of the single link from the group.

Step 10 Stop background scripts to collect final status of network devices and analyze for error.

Step 11 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that the single link failure doesn't affect active storage traffic over the Port-Channel.
- We expect for the failure to be detected and reported by the nodes to the management applications/servers.
- We expect for the active storage traffic to be load-balanced over the remaining port-channel members.
- We expect no unacceptable impact on the CPU or memory.

Results

Single Link Failure in Group passed.

Single Link Remove from Group

This test verified the support for the removal (shut down) of a single link from an active Port-Channel group without disrupting active traffic or services over the channel. Storage traffic was generated to cross the Port-Channels prior, during, and after the link was removed. All configuration and verification was done via CLI.

Test Procedure

The procedure used to perform the Single Link Remove from Group test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 [Pre-Test Condition #1] Target test port-channel must be an intra-fabric 12xISL member group.

Step 3 Generate storage traffic (with random OXIDs) using the SANTester tool from an Edge MDS across a 12xISL port-channel out a Core MDS.

Step 4 Verify storage traffic is flowing without loss or problems over Port-Channel(s) - before, during, and after the port removal.

Step 5 Verify that storage traffic traversing the target 12xISL Intra-Fabric Port-Channel is evenly distributed across all channel members.

Step 6 Remove (via configuration) a member off the target port-channel by SHUT-DOWN.

Step 7 Verify that the removed port-channel member becomes inactive in the group.

Step 8 Verify that storage traffic traversing the target 11xISL Intra-Fabric Port-Channel is evenly distributed across all channel members.

Step 9 Confirm that storage traffic traversing the target port-channel was not affected during or after the removal of the single link from the group.

Step 10 Stop background scripts to collect final status of network devices and analyze for error.

Step 11 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that the single link removal (shut down) is executed successfully without errors, issues, or problems.
- We expect that the single link removal didn't affect active storage traffic over the Port-Channel.
- We expect for the active storage traffic to be load-balanced over the remaining port-channel members.
- We expect no unacceptable impact on the CPU or memory.

Results

Single Link Remove from Group passed.

Resiliency Functionality

Resiliency functionality tests examine whether single component failures cause unexpected storage access disruption. These tests were not performed in Phase 2 for HP due to hardware issues and time constraints.

The following test features were conducted:

- EMC, page 4-80
- NetApp, page 4-84
- Host Link Failure (Link Pull)—NETAPP, page 4-84

EMC

EMC resiliency functionality tests involve failing a path by disabling an edge link, disconnecting an edge link, and reloading and replacing an edge switch module and making sure the fabric design and the PowerPath multipath software transparently route all traffic over the remaining link.

The following tests were performed:

- Host Link Failure (Link Pull)—EMC, page 4-80
- Host Link Failure (Port Shutdown)—EMC, page 4-81
- Host Facing Module Failure (OIR)—EMC, page 4-82
- Host Facing Module Failure (Reload)—EMC, page 4-83

Host Link Failure (Link Pull)—EMC

This test verified the Fabric and Host resiliency to a physical link failure when multi-pathing software is running in a host with redundant paths. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. One of the test-host links was pulled to verify traffic re-route to redundant connections. This link was later re-connected to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager.

Test Procedure

The procedure used to perform the Host Link Failure (Link Pull)—EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Physically remove link connection between test-host (windows) and fabric.

Step 4 Verify that the failure is detected and reported to the management applications.

Step 5 Verify traffic flows completely over redundant path and record any traffic loss.

Step 6 Reconnect link and confirm that it recovers without problems.

Step 7 Verify storage traffic flow recovers over the reconnected link.

Step 8 Verify link recovery is detected and reported by the devices to the management applications.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the failure as multi-pathing software takes care of failing it over to the redundant host link.
- We expect traffic loss to be null or minimal during failure and recovery.
- We expect traffic from non-test-hosts is not affected by the failure or recovery.
- We expect all systems to recover completely from the link failure. Traffic rates and delays return to pre-failure status.
- We expect all the failure and recovery to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Host Link Failure (Link Pull)—EMC passed.

Host Link Failure (Port Shutdown)—EMC

This test verified the Fabric and Host resiliency to a port shut down when multi-pathing software is running in a host with redundant paths. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. One of the test-host links was shut down to verify traffic re-route to redundant connections. This link was later re-enabled to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager .

Test Procedure

The procedure used to perform the Host Link Failure (Port Shutdown)—EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Shutdown a link connection between test-host (windows) and fabric.

Step 4 Verify that the shut down is detected and reported to the management applications.

Step 5 Verify traffic flows completely over redundant path and record any traffic loss.

Step 6 Re-enable the link and confirm that it recovers without problems.

Step 7 Verify storage traffic flow recovers over the re-enabled link.

Step 8 Verify link recovery is detected and reported by the devices to the management applications.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the shut down as multi-pathing software takes care of failing it over to the redundant host link.
- We expect traffic loss to be null or minimal during failure and recovery.
- We expect traffic from non-test-hosts is not affected by the failure or recovery.
- We expect all systems to recover completely from the link shut down. Traffic rates and delays return to pre-failure status.
- We expect the failure and recovery to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Host Link Failure (Port Shutdown)—EMC passed.

Host Facing Module Failure (OIR)—EMC

This test verified the Fabric and Host resiliency to a host-facing module removal/re-insertion when multi-pathing software is running in a host with redundant paths. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. Test-host facing module was removed to verify traffic re-routing to redundant connections. Edge module is re-inserted to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager.

Test Procedure

The procedure used to perform the Host Facing Module Failure (OIR)—EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Remove edge module connecting test-host (windows) and fabric.

Step 4 Verify that the failure is detected and reported to the management applications.

Step 5 Verify traffic flows completely over redundant path and record any traffic loss.

Step 6 Re-insert the module and confirm that it recovers without problems.

Step 7 Verify storage traffic flow recovers over the reconnected link.

Step 8 Verify link recovery is detected and reported by the devices to the management applications.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the failure as multi-pathing software takes care of failing it over to the redundant host link.
- We expect traffic loss to be null or minimal during failure and recovery.
- We expect traffic from non-test-hosts is not affected by the failure or recovery.
- We expect all systems to recover completely from the module removal/re-insertion. Traffic rates and delays return to pre-failure status.
- We expect all the failure and recovery to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Host Facing Module Failure (OIR)—EMC passed.

Host Facing Module Failure (Reload)—EMC

This test verified the Fabric and Host resiliency to a host-facing module reload when multi-pathing software is running in a host with redundant paths. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. Test-host facing module is reloaded to verify traffic re-route to redundant connections. Edge module came back online to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager.

Test Procedure

The procedure used to perform the Host Facing Module Failure (Reload)—EMC test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Reload edge module connecting test-hosts and fabric.

Step 4 Verify that the failure is detected and reported to the management applications.

Step 5 Verify traffic flows completely over redundant path and record any traffic loss.

Step 6 On reload of the module confirm that it recovers without problems.

Step 7 Verify storage traffic flow recovers over the reconnected link.

Step 8 Verify link recovery is detected and reported by the devices to the management applications.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the failure as multi-pathing software takes care of failing it over to the redundant host link.
- We expect traffic loss to be null or minimal during failure and recovery.
- We expect traffic from non-test-hosts is not affected by the failure or recovery.
- We expect all systems to recover completely from the module reload. Traffic rates and delays return to pre-failure status.
- We expect all the failure and recovery to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Host Facing Module Failure (Reload)—EMC passed.

NetApp

NetApp resiliency functionality tests involve failing a path by disabling an edge link, disconnecting an edge link, and reloading and replacing an edge switch module and making sure the fabric design and host multipath software (native MPIO for Linux, Veritas DMP for Windows) transparently route all traffic over the remaining link

The following tests were performed:

- Host Link Failure (Link Pull)—NETAPP, page 4-84
- Host Link Failure (Port Shutdown)—NETAPP, page 4-85
- Host Facing Module Failure (OIR)—NETAPP, page 4-86
- Host Facing Module Failure (Reload)—NETAPP, page 4-87

Host Link Failure (Link Pull)—NETAPP

This test verified the Fabric and Host resiliency to a physical link failure when multi-pathing software is running in a host with redundant paths. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. One of the test-host links was pulled to verify traffic re-route to redundant connections. This link was later re-connected to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager.

Test Procedure

The procedure used to perform the Host Link Failure (Link Pull)—NETAPP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Physically remove link connection between test-host (windows) and fabric.

Step 4 Verify that the failure is detected and reported to the management applications.

Step 5 Verify traffic flows completely over redundant path and record any traffic loss.

Step 6 Reconnect link and confirm that it recovers without problems.

Step 7 Verify storage traffic flow recovers over the reconnected link.

Step 8 Verify link recovery is detected and reported by the devices to the management applications.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the failure as multi-pathing software takes care of failing it over to the redundant host link.
- We expect traffic loss to be null or minimal during failure and recovery.
- We expect traffic from non-test-hosts is not affected by the failure or recovery.
- We expect all systems to recover completely from the link failure. Traffic rates and delays return to pre-failure status.
- We expect all the failure and recovery to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Host Link Failure (Link Pull)—NETAPP passed.

Host Link Failure (Port Shutdown)—NETAPP

This test verified the Fabric and Host resiliency to a port shut down when multi-pathing software is running in a host with redundant paths. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. One of the test-host links was shut down to verify traffic re-route to redundant connections. This link was later re-enabled to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager .

Test Procedure

The procedure used to perform the Host Link Failure (Port Shutdown)—NETAPP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Shutdown a link connection between test-host (windows) and fabric.

Step 4 Verify that the shut down is detected and reported to the management applications.

Step 5 Verify traffic flows completely over redundant path and record any traffic loss.

Step 6 Re-enable the link and confirm that it recovers without problems.

Step 7 Verify storage traffic flow recovers over the re-enabled link.

Step 8 Verify link recovery is detected and reported by the devices to the management applications.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the shut down as multi-pathing software takes care of failing it over to the redundant host link.
- We expect traffic loss to be null or minimal during failure and recovery.
- We expect traffic from non-test-hosts is not affected by the failure or recovery.
- We expect all systems to recover completely from the link shut down. Traffic rates and delays return to pre-failure status.
- We expect the failure and recovery to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Host Link Failure (Port Shutdown)—NETAPP passed.

Host Facing Module Failure (OIR)—NETAPP

This test verified the Fabric and Host resiliency to a host-facing module removal/re-insertion when multi-pathing software is running in a host with redundant paths. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. Test-host facing module was removed to verify traffic re-routing to redundant connections. Edge module is re-inserted to verify full recovery of failed condition. All configurations and verifications were done via with confirmation through Fabric Manager.

Test Procedure

The procedure used to perform the Host Facing Module Failure (OIR)—NETAPP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Remove edge module connecting test-host (windows) and fabric.

Step 4 Verify that the failure is detected and reported to the management applications.

Step 5 Verify traffic flows completely over redundant path and record any traffic loss.

Step 6 Re-insert the module and confirm that it recovers without problems.

Step 7 Verify storage traffic flow recovers over the reconnected link.

Step 8 Verify link recovery is detected and reported by the devices to the management applications.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the failure as multi-pathing software takes care of failing it over to the redundant host link.
- We expect traffic loss to be null or minimal during failure and recovery.
- We expect traffic from non-test-hosts is not affected by the failure or recovery.
- We expect all systems to recover completely from the module removal/re-insertion. Traffic rates and delays return to pre-failure status.
- We expect all the failure and recovery to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Host Facing Module Failure (OIR)—NETAPP passed.

Host Facing Module Failure (Reload)—NETAPP

This test verified the Fabric and Host resiliency to a host-facing module reload when multi-pathing software is running in a host with redundant paths. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. Test-host facing module is reloaded to verify traffic re-route to redundant connections. Edge module came back online to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager.

Test Procedure

The procedure used to perform the Host Facing Module Failure (Reload)—NETAPP test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Reload edge module connecting test-hosts and fabric.

Step 4 Verify that the failure is detected and reported to the management applications.

Step 5 Verify traffic flows completely over redundant path and record any traffic loss.

Step 6 On reload of the module confirm that it recovers without problems.

Step 7 Verify storage traffic flow recovers over the reconnected link.

Step 8 Verify link recovery is detected and reported by the devices to the management applications.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the failure as multi-pathing software takes care of failing it over to the redundant host link.
- We expect traffic loss to be null or minimal during failure and recovery.
- We expect traffic from non-test-hosts is not affected by the failure or recovery.
- We expect all systems to recover completely from the module reload. Traffic rates and delays return to pre-failure status.
- We expect all the failure and recovery to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Host Facing Module Failure (Reload)—NETAPP passed.

MDS

MDS resiliency functionality tests induced failures and resets for every major component type in the host-to-storage connectivity fabrics to make sure the fabric design transparently avoided disruption of host and storage traffic.

The following tests were performed:

- Active Crossbar Fabric Failover (OIR), page 4-89
- Active Supervisor Failover (OIR), page 4-90
- Active Supervisor Failover (Reload), page 4-90
- Active Supervisor Failover (Manual CLI), page 4-91
- Back Fan-Tray Failure (Removal), page 4-92
- Core Facing Module Failure (OIR), page 4-93
- Core Facing Module Failure (Reload), page 4-94
- Front Fan-Tray Failure (Removal), page 4-95
- Node Failure (Power Loss), page 4-96
- Node Failure (Reload), page 4-97
- Power Supply Failure (Cord Removal), page 4-98

- Power Supply Failure (Power Off), page 4-99
- Power Supply Failure (Removal), page 4-100
- Standby Supervisor Failure (OIR), page 4-100
- Standby Supervisor Failure (Reload), page 4-101
- Unused Module Failure (OIR), page 4-102

Active Crossbar Fabric Failover (OIR)

This test verified that a removal/re-insertion of an active crossbar-fabric in an edge node causes NO disruption to active services and storage traffic. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. An active Crossbar-fabric module was removed to verify the non-disruption of traffic as the other module takes over. Crossbar-fabric module was re-inserted back online to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager.

Test Procedure

The procedure used to perform the Active Crossbar Fabric Failover (OIR) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Remove an active crossbar-fabric from the Edge node where storage traffic is entering the fabric. Verify the removal is detected an reported to the management applications.

Step 4 Verify traffic flows are not affected by the crossbar-fabric removal.

Step 5 On re-insertion of the module confirm that it recovers without problems.

Step 6 Verify storage traffic flows without loss or problems.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the removal or re-insertion of the ACTIVE crossbar-fabric as redundant a module is present.
- We expect connectivity from test-hosts to arrays is not affected by the failure or recovery.
- We expect all systems to recover completely from the re-insertion.
- We expect the removal and re-insertion to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Active Crossbar Fabric Failover (OIR) passed.

Active Supervisor Failover (OIR)

This test verified that a removal/re-insertion of the active supervisor in an edge node causes NO disruption to active services and storage traffic. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. Active Supervisor module was removed to verify the non-disruption of traffic as the standby module takes over. Supervisor module was re-inserted back online (in standby) to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager.

Test Procedure

The procedure used to perform the Active Supervisor Failover (OIR) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Remove the ACTIVE supervisor from the Edge node where storage traffic is entering the fabric.

Step 4 Verify that the standby supervisor becomes ACTIVE and the reload is detected and reported to the management applications.

Step 5 Verify traffic flows are not affected by the supervisor removal.

Step 6 On re-insertion of the module confirm that it recovers without problems in STANDBY mode.

Step 7 Verify storage traffic flows without loss or problems.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the removal or re-insertion of the ACTIVE supervisor as redundant supervisor modules are present.
- We expect connectivity from test-hosts to arrays is not affected by the failure or recovery.
- We expect all systems to recover completely from the re-insertion.
- We expect the removal and re-insertion to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Active Supervisor Failover (OIR) passed.

Active Supervisor Failover (Reload)

This test verified that a reload of the active supervisor in an edge node causes NO disruption to active services and storage traffic. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. Active Supervisor module was reloaded to verify the non-disruption of traffic.

Supervisor module came back online (in standby) to verify full recovery of failed condition. All configurations were done through Fabric Manager and verifications were done via CLI with confirmation through Fabric Manager.

Test Procedure

The procedure used to perform the Active Supervisor Failover (Reload) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular switch.

Step 3 Executed a reload of the ACTIVE supervisor to the STANDBY one in an Edge node where storage traffic is entering the fabric.

Step 4 Verify that the reload is detected and reported to the management applications.

Step 5 Verify traffic flows are not affected by the supervisor reload.

Step 6 On reload of the module confirm that it recovers without problems in STANDBY mode.

Step 7 Verify storage traffic flows without loss or problems.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the reload as redundant supervisor modules are present.
- We expect connectivity from test-hosts to arrays is not affected by the failure or recovery.
- We expect all systems to recover completely from the reload.
- We expect the reload to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Active Supervisor Failover (Reload) passed.

Active Supervisor Failover (Manual CLI)

This test verified that a manual failover of the active supervisor in an edge node causes NO disruption to active services and storage traffic. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. Active Supervisor module is failed over to the standby to verify the non-disruption of traffic. Supervisor module came back online (in standby) to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager .

Test Procedure

The procedure used to perform the Active Supervisor Failover (Manual CLI) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular switch.

Step 3 Execute a manual fail-over of the ACTIVE supervisor to the STANDBY one in an Edge node where storage traffic is entering the fabric.

Step 4 Verify that the fail-over is detected and reported to the management applications.

Step 5 Verify traffic flows are not affected by the supervisor fail-over.

Step 6 On reload of the module confirm that it recovers without problems in STANDBY mode.

Step 7 Verify storage traffic flows without loss or problems.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the fail-over as redundant supervisor modules are present.
- We expect connectivity from test-hosts to arrays is not affected by the failure or recovery.
- We expect all systems to recover completely from the fail-over.
- We expect the failover and recovery to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Active Supervisor Failover (Manual CLI) passed.

Back Fan-Tray Failure (Removal)

This test verified that the removal of the back fan-tray in a core node causes NO disruption to active services and storage traffic. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. The back fan-tray unit was removed to verify the non-disruption of traffic. Fan-tray was re-inserted to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager.

Test Procedure

The procedure used to perform the Back Fan-Tray Failure (Removal) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Remove one the back fan-tray unit in a core node where storage traffic is crossing the fabric.

Step 4 Verify that the front fan-tray removal is detected and reported to the management applications.

Step 5 Verify traffic flows are not affected by the removal.

Step 6 Monitor environmental alarms - expect fan-tray and possible temperature alarms.

Step 7 Re-insert the back fan-tray unit. Confirm that it recovers without problems. Confirm via device environmental status report.

Step 8 Verify storage traffic flows without loss or problems.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the back fan-tray removal if replaced within the specified time (5 min).
- We expect connectivity from test-hosts to arrays is not affected by the failure or recovery.
- We expect all systems to recover completely from the removal/re-insertion.
- We expect the back fan-tray removal/re-insertion to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Back Fan-Tray Failure (Removal) passed.

Core Facing Module Failure (OIR)

This test verified the fabric resiliency to a core-facing module removal/re-insertion when port-channeling is running between Edge and Core nodes with multi-module member distribution. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. Core-facing module is removed to verify traffic re-distribution over remaining port-channel members. Core-facing module is re-inserted to verify full recovery of failed condition. All configurations and verifications were done via Fabric Manager with confirmation through CLI.

Test Procedure

The procedure used to perform the Core Facing Module Failure (OIR) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Remove core-facing module connecting half of the port-channel member count between the Edge node and Core node in the fabric.

Step 4 Verify that the failure is detected and reported to the management applications.

Step 5 Verify traffic flows completely over remaining group members and record any traffic loss.

Step 6 On re-insertion of the module confirm that it recovers without problems.

Step 7 Verify storage traffic flow recovers over the reconnected links.

Step 8 Verify link recovery is detected and reported by the devices to the management applications.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the failure as port-channel takes care of distributing it over the remaining group members.
- We expect traffic loss to be null or minimal during failure and recovery.
- We expect connectivity from test-hosts to arrays is not affected by the failure or recovery.
- We expect all systems to recover completely from the module re-insertion. Traffic rates and delays return to pre-failure status.
- We expect all the failure and recovery to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Core Facing Module Failure (OIR) passed.

Core Facing Module Failure (Reload)

This test verified the Fabric resiliency to a core-facing module reload port-channeling is running between Edge and Core nodes with multi-module member's distribution. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. Core-facing module is reloaded to verify traffic re-distribution over remaining port-channel members. Core-facing module came back online to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager .

Test Procedure

The procedure used to perform the Core Facing Module Failure (Reload) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Reload core-facing module connecting half of the port-channel member count between the Edge node and Core node in the fabric.

Step 4 Verify that the failure is detected and reported to the management applications.

Step 5 Verify traffic flows completely over remaining group members and record any traffic loss.

Step 6 On reload of the module confirm that it recovers without problems.

Step 7 Verify storage traffic flow recovers over the reconnected links.

Step 8 Verify link recovery is detected and reported by the devices to the management applications.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the failure as port-channel takes care of distributing it over the remaining group members.
- We expect traffic loss to be null or minimal during failure and recovery.
- We expect connectivity from test-hosts to arrays is not affected by the failure or recovery.
- We expect all systems to recover completely from the module reload. Traffic rates and delays return to pre-failure status.
- We expect all the failure and recovery to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Core Facing Module Failure (Reload) passed.

Front Fan-Tray Failure (Removal)

This test verified that the removal of the front fan-tray in a core node causes NO disruption to active services and storage traffic. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. The front fan-tray unit was removed to verify the non-disruption of traffic. Fan-tray was re-inserted after 5 minute to ensure the system proactively shut down as designed and to verify full recovery of failed condition after the switch came back up. All configurations and verifications were done via CLI with confirmation through Fabric Manager.

Test Procedure

The procedure used to perform the Front Fan-Tray Failure (Removal) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Remove one the front fan-tray unit in a core node where storage traffic is crossing the fabric.

Step 4 Verify that the front fan-tray removal is detected and reported to the management applications.

Step 5 Verify traffic flows are not affected by the removal.

Step 6 Monitor environmental alarms - expect fantray and possible temperature alarms.

Step 7 Re-insert the front fan-tray unit after waiting for more than 5 minutes and bringing the switch back up. Confirm that it shuts down by itself after 5 minutes and recovers without problems. Confirm via device environmental status report.

Step 8 Verify storage traffic flows without loss or problems.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the front fan-tray removal if not replaced within the specified time (5 min).
- We expect connectivity from test-hosts to arrays is not affected by the failure or recovery.
- We expect all systems to recover completely from the removal/re-insertion.
- We expect the front fan-tray removal/re-insertion to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Front Fan-Tray Failure (Removal) passed.

Node Failure (Power Loss)

This test verified that a complete core node failure (power loss) causes NO disruption to active services and storage traffic. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. One of the core nodes was powered-off to verify the non-disruption of traffic. The core node was powered ON to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager.

Test Procedure

The procedure used to perform the Node Failure (Power Loss) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Power down one of the core nodes where storage traffic is crossing the fabric.

Step 4 Verify that the core node loss is detected and reported to the management applications.

Step 5 Verify traffic flows are not affected by the core node loss. Confirm traffic is re-routed around the failed node.

Step 6 Power the node and confirm that it recovers without problems.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the complete loss of a core node as a redundant core node is present.
- We expect connectivity from test-hosts to arrays is not affected by the failure or recovery.
- We expect all systems to recover completely from core node loss/recovery.
- We expect the core node loss to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Node Failure (Power Loss) passed.

Node Failure (Reload)

This test verified that a complete core node failure (reload) causes NO disruption to active services and storage traffic. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. One of the core nodes was reloaded to verify the non-disruption of traffic. The core node was online to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager .

Test Procedure

The procedure used to perform the Node Failure (Reload) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Reload one of the core nodes where storage traffic is crossing the fabric.

Step 4 Verify that the core node loss is detected and reported to the management applications.

Step 5 Verify traffic flows are not affected by the core node loss. Confirm traffic is re-routed around the failed node.

Step 6 Once online confirm that it recovers without problems.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the complete loss of a core node as a redundant core node is present.

- We expect connectivity from test-hosts to arrays is not affected by the failure or recovery.
- We expect all systems to recover completely from core node reload.
- We expect the core node reload to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Node Failure (Reload) passed.

Power Supply Failure (Cord Removal)

This test verified that a loss of a power supply unit due to power-cords removal in a core node causes NO disruption to active services and storage traffic. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. One of the two power supply units lost power cord connection to verify the non-disruption of traffic. Power supply was plugged back ON to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager.

Test Procedure

The procedure used to perform the Power Supply Failure (Cord Removal) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Unplug the power cable off one of the power supply units in a core node where storage traffic is crossing the fabric.

Step 4 Verify that the power supply loss of power is detected and reported to the management applications.

Step 5 Verify traffic flows are not affected by the power loss.

Step 6 Plug the power supply and confirm that it recovers without problems. Confirm via device environmental status report.

Step 7 Verify storage traffic flows without loss or problems.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the power supply loss as a redundant power supply unit is present.
- We expect connectivity from test-hosts to arrays is not affected by the failure or recovery.
- We expect all systems to recover completely from the power loss/recovery.
- We expect the power supply power loss to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).

- We expect no unacceptable impact on the CPU or memory.

Results

Power Supply Failure (Cord Removal) passed.

Power Supply Failure (Power Off)

This test verified that a loss of a power supply unit in a core node causes NO disruption to active services and storage traffic. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. One of the two power supply units was powered-off to verify the non-disruption of traffic. Power supply was powered ON to verify full recovery of failed condition. All configurations and verifications were done via Fabric Manager with confirmation through CLI.

Test Procedure

The procedure used to perform the Power Supply Failure (Power Off) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Power down one of the power supply units in a core node where storage traffic is crossing the fabric.

Step 4 Verify that the power supply shutdown is detected and reported to the management applications.

Step 5 Verify traffic flows are not affected by the power loss.

Step 6 Power the unit and confirm that it recovers without problems. Confirm via device environmental status report.

Step 7 Verify storage traffic flows without loss or problems.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the power supply loss as a redundant power supply unit is present.
- We expect connectivity from test-hosts to arrays is not affected by the failure or recovery.
- We expect all systems to recover completely from the power loss/recovery.
- We expect the power supply shutoff to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Power Supply Failure (Power Off) passed.

Power Supply Failure (Removal)

This test verified that the removal of a power supply unit in a core node causes NO disruption to active services and storage traffic. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. One of the two power supply units was removed to verify the non-disruption of traffic. Power supply was re-inserted to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager.

Test Procedure

The procedure used to perform the Power Supply Failure (Removal) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Remove one of the power supply units in a core node where storage traffic is crossing the fabric.

Step 4 Verify that the power supply removal is detected and reported to the management applications.

Step 5 Verify traffic flows are not affected by the power loss.

Step 6 Re-insert the power supply and power the unit. Confirm that it recovers without problems. Confirm via device environmental status report.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the power supply loss as a redundant power supply unit is present.
- We expect connectivity from test-hosts to arrays is not affected by the failure or recovery.
- We expect all systems to recover completely from the power loss/recovery.
- We expect the power supply removal/re-insertion to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Power Supply Failure (Removal) passed.

Standby Supervisor Failure (OIR)

This test verified that a removal/re-insertion of the STANDBY supervisor in an edge node causes NO disruption to active services and storage traffic. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. STANDBY Supervisor module was removed to verify the non-disruption of traffic. Supervisor module was re-inserted and came up online (in standby) to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager.

Test Procedure

The procedure used to perform the Standby Supervisor Failure (OIR) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Remove the STANDBY supervisor in an Edge node where storage traffic is entering the fabric.

Step 4 Verify that the removal is detected and reported to the management applications.

Step 5 Verify traffic flows are not affected by the STANDBY supervisor removal.

Step 6 On re-insertion of the module confirm that it recovers without problems in STANDBY mode.

Step 7 Stop background scripts to collect final status of network devices and analyze for error.

Step 8 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the removal as the supervisor module is the STANDBY unit.
- We expect connectivity from test-hosts to arrays is not affected by the failure or recovery.
- We expect all systems to recover completely from the removal/re-insertion.
- We expect the removal/re-insertion to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Standby Supervisor Failure (OIR) passed.

Standby Supervisor Failure (Reload)

This test verified that a reload of the STANDBY supervisor in an edge node causes NO disruption to active services and storage traffic. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. STANDBY Supervisor module was reloaded to verify the non-disruption of traffic. Supervisor module came back online (in standby) to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager.

Test Procedure

The procedure used to perform the Standby Supervisor Failure (Reload) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Executed a reload of the STANDBY supervisor in an Edge node where storage traffic is entering the fabric.

Step 4 Verify that the reload is detected and reported to the management applications.

Step 5 Verify traffic flows are not affected by the STANDBY supervisor reload.

Step 6 On reload of the module confirm that it recovers without problems in STANDBY mode.

Step 7 Verify storage traffic flows without loss or problems.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the reload as supervisor module is the STANDBY unit.
- We expect connectivity from test-hosts to arrays is not affected by the failure or recovery.
- We expect all systems to recover completely from the reload.
- We expect the reload to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Standby Supervisor Failure (Reload) passed.

Unused Module Failure (OIR)

This test verified the Fabric resiliency to an unused module (edge node) reload/removal/re-insertion when there is NO storage traffic through it. Storage traffic (IOs) were generated by the test-hosts (Windows and Linux) to the storage array. An unused module in the Edge node was reloaded/removed/re-inserted to verify that there is NO effect on storage traffic or support services. The module was then re-inserted to verify full recovery of failed condition. All configurations and verifications were done via CLI with confirmation through Fabric Manager.

Test Procedure

The procedure used to perform the Unused Module Failure (OIR) test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Generate storage traffic (IOs) from multiple test-hosts (OS: Windows and Linux) to the particular storage array.

Step 3 Remove unused edge node module unrelated to test-hosts or core connections.

Step 4 Verify that the failure is detected and reported to the management applications.

Step 5 Verify traffic continues to flow without interruption.

Step 6 On re-insertion of the module confirm that it recovers without problems.

Step 7 Verify storage traffic flow is not affected by the re-insertion.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that traffic is not stopped by the unrelated failure.
- We expect connectivity from test-hosts to arrays is not affected by the unrelated failure or recovery.
- We expect all systems to recover completely from the module re-insertion.
- We expect all the failure and recovery to be detected and reported by the devices to the management application servers (e.g., Fabric Manager, SYSLOG server, etc.).
- We expect no unacceptable impact on the CPU or memory.

Results

Unused Module Failure (OIR) passed.

Security Functionality

Security functionality tests checked the proper operation of the FC Security Protocol (FC-SP), port security, and TACACS+ access control.

The following tests were performed:

- FC SP Authentication Failure, page 4-103
- Port Security Basic Implementation, page 4-104
- User Access—TACACS Basic Test, page 4-105
- User Access—TACACS Servers Failure, page 4-106

FC SP Authentication Failure

This test verified FC-SP capability to reject an unauthorized node from joining the fabric. All configuration and verification was done via Fabric Manager with confirmation through CLI.

Test Procedure

The procedure used to perform the FC SP Authentication Failure test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Configure a fabric with FS-SP.

Step 3 Verify the configuration and successful authorization of all nodes in the fabric. All members (all fabric nodes) must be active. Testbed must be fully operational.

Step 4 Change an Edge MDS FC-SP configuration to have the wrong key. Reconnect to the fabric.

Step 5 Verify that the Edge MDS is rejected (prohibited to join the fabric).

Step 6 Stop background scripts to collect final status of network devices and analyze for error.

Step 7 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that FC-SP successfully rejects the integration of an edge switch with wrong credentials.
- We expect that detection, reporting, validation, verification was successfully done with Fabric Manager with CLI confirmation.
- We expect no unacceptable impact on the CPU or memory.

Results

FC SP Authentication Failure passed.

Port Security Basic Implementation

The configuration and verification of Port-Security was tested. A single host was allowed per port and then replaced with another non-authorized host. All configuration and verification was done via Fabric Manager with confirmation through CLI.

Test Procedure

The procedure used to perform the Port Security Basic Implementation test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Configure Port-Security for auto-learning in a host facing ports.

Step 3 Verify that all Test-Hosts are learned by Port-Security in all edge ports.

Step 4 Disable Port-Security Auto mode.

Step 5 Generate storage traffic from two(2) SANTester ports. From an edge switch to an core switch.

Step 6 Verify storage traffic is flowing without loss or problems across the fabric.

Step 7 Replace one of the end-devices with a new one (different pwwn) and verify that Port-security rejects the new connection (at flogi as it is not allowed.

Step 8 Verify that Port-Security detects and rejects the wrong host connection and reports it to the management applications.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that Port-Security is configured without problems or issues.
- We expect for Port-Security to automatically learn the address of the test-host and keep it locked after auto-learn is disabled.

- We expect for Port-Security to reject (or not allow) the incorrect non-authorized node to login into the fabric.
- We expect that detection, reporting, validation, verification was successfully done with Fabric Manager with CLI confirmation.

Results

Port Security Basic Implementation passed.

User Access—TACACS Basic Test

This test verified TACACS+ support in the MDS as the authorization/authentication main mechanism in the testbed. Remote (TACACS+) authorization/authentication is validated. All configuration and verification was done via Fabric Manager with confirmation through CLI.

Test Procedure

The procedure used to perform the User Access—TACACS Basic Test test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Remote(TACACS+) authorization/authentication must be configured and services active/available.

Step 3 Access target node and login using the Remote(TACACS+) Username/Password configured in the TACACS+ Server configuration.

Step 4 Verify that access and administrator authorization is granted then logout.

Step 5 Access target node and login using the Local Username/Password configured in the nodes configuration. Verify that access is NOT granted (i.e. Access fails with Local Username/Password combination).

Step 6 Access target node and login using wrong Username/Password combinations. Verify that access is NOT granted.

Step 7 Verify that all rejections are reported to the management applications.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect successful access and Remote Authorization/Authentication.
- We expect for access to be denied with wrong Username/Password combination.
- We expect that detection, reporting, validation, verification was successfully done with Fabric Manager with CLI confirmation.
- We expect no unacceptable impact on the CPU or memory.

Results

User Access—TACACS Basic Test passed.

User Access—TACACS Servers Failure

This test verified TACACS+ support for redundant servers and local authentication as last resort in the MDS. All configuration and verification was done via Fabric Manager with confirmation through CLI.

Test Procedure

The procedure used to perform the User Access—TACACS Servers Failure test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Remote(TACACS+) authorization/authentication must be configured and services active/available.

Step 3 Access target node and login using the Remote(TACACS+) Username/Password configured in the TACACS+ Server configuration.

Step 4 Verify that access and administrator authorization is granted then logout.

Step 5 Take Primary TACACS+ server offline and attempt to login again with the predefined Username/Password.

Step 6 Verify that access and administrator authorization is granted using the second TACACS+ server. Confirm via CLI and FM that the Primary TACACS+ server is offline. Logout.

Step 7 Take Secondary TACACS+ server offline and attempt to login again with the predefined Username/Password. Verify that access failed using TACACS+ defined Username/Password.

Step 8 Attempt to login using Local Authentication Username/Password. Once logged in verify that both TACACS+ servers are down.

Step 9 Bring both TACACS+ servers online and attempt login thru them. Verify full connectivity from the target MDS to the TACACS+ servers.

Step 10 Stop background scripts to collect final status of network devices and analyze for error.

Step 11 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect for the MDS to successfully access the TACACS+ secondary servers if communication to the primary is lost.
- We expect for local authorization to be used as last resort when all TACACS+ servers are down.
- We expect that detection, reporting, validation, verification was successfully done with Fabric Manager with CLI confirmation.
- We expect no unacceptable impact on the CPU or memory.

Results

User Access—TACACS Servers Failure passed.

End-to-End Functionality

End-to-end functionality tests tied the DCAP LAN and SAN topologies together by verifying that transactions initiated by test client hosts were properly routed by L4-L7 services and serviced by test servers accessing replicated storage on the SAN.

The following test feature was conducted:

- HTTP Functionality, page 4-107

HTTP Functionality

HTTP functionality tests checked whether HTTP PUTs and GETs were properly load balanced by CSM and that all data changes were properly replicated from one data center to the other via the SAN transit fabrics.

The following tests were performed:

- End-to-End Get, page 4-107
- End-to-End Put, page 4-109

End-to-End Get

This test verified that client devices could retrieve files from LUNs configured in the various arrays, and mapped to the real servers.

Each of the three vendor arrays (EMC, NetApp, and HP) have configured on them 6 LUNs. Two of the LUNs (Local_IO_1 and Local_IO_2) are for local storage only (not replicated to arrays in DC-B). Two of the LUNs (Sync_IO_1 and Sync_IO_2) are synchronously replicated to the sister array in DC-B. Two of the LUNs (Async_IO_1 and Async_IO_2) are asynchronously replicated to the sister array in DC-B. There are three serverfarms in DC-A that point to the three arrays, and three Vservers that point to the three serverfarms.

A script is used to generate client HTTP GET requests for a file on each of the LUNs on each of the arrays.

Test Procedure

The procedure used to perform the End-to-End Get test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify that each of the three Vservers (E2E-EMC-VIP, E2E-NETAPP-VIP and E2E-HP-VIP) on the active CSM in DC-A is configured and **OPERATIONAL** using the **show module csm 2 vserver name** *vserver* **detail** command.

For each of the three Vservers, the state should be **OPERATIONAL**.

Step 3 Verify that each of the three serverfarms (E2E-EMC, E2E-NETAPP and E2E-HP) on the active CSM in DC-A is configured and each of the five real servers are **OPERATIONAL** using the **show module csm 2 serverfarm name** *serverfarm* **detail** command.

Each of the five real servers in each serverfarm, should display as being **OPERATIONAL**.

Step 4 On dca-penguin-1, verify that the following files are in place:

/Local_IO_1/100k-EMC-L-1.html

/Local_IO_2/100k-EMC-L-2.html

/Sync_IO_1/100k-EMC-S-1.html

/Sync_IO_2/100k-EMC-S-2.html

/Async_IO_1/100k-EMC-A-1.html

/Async_IO_2/100k-EMC-A-2.html

Step 5 On dca-penguin-2, verify that the following files are in place:

/Local_IO_1/100k-NA-L-1.html

/Local_IO_2/100k-NA-L-2.html

/Sync_IO_1/100k-NA-S-1.html

/Sync_IO_2/100k-NA-S-2.html

/Async_IO_1/100k-NA-A-1.html

/Async_IO_2/100k-NA-A-2.html

Step 6 On dca-penguin-4, verify that the following files are in place:

/Local_IO_1/100k-HP-L-1.html

/Local_IO_2/100k-HP-L-2.html

/Sync_IO_1/100k-HP-S-1.html

/Sync_IO_2/100k-HP-S-2.html

/Async_IO_1/100k-HP-A-1.html

/Async_IO_2/100k-HP-A-2.html

Step 7 Use the **clear module csm 2 counters** command on the active CSM to clear the vserver and serverfarm statistics.

Step 8 Start the traffic script on dca-penguin-11 that will create multiple HTTP GET requests from several hundred client IP addresses for those files that were verified above.

Step 9 Verify that the **Client pkts** and **Server pkts** count has increased for each of the vservers using the **show module csm 2 vserver name** *vserver* **detail** command.

The **Client pkts** and **Server pkts** fields should show a large positive value.

Step 10 Verify that the real servers in each of the three serverfarms has processed a connection using the **show module csm 2 serverfarm name** *serverfarm* **detail** command.

The **total conns established** counter should be a large value for each of the real servers in each serverfarm.

Step 11 Verify that the script returned all 200 return codes, indicating that there were no retrieval failures.

Step 12 Stop background scripts to collect final status of network devices and analyze for error.

Step 13 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that all of the HTTP GET requests will succeed.

Results

End-to-End Get passed.

End-to-End Put

This test verified that client devices could write files into LUNs configured in the various arrays, and mapped to the real servers.

Each of the three vendor arrays (EMC, NetApp, and HP) have configured on them 6 LUNs. Two of the LUNs (Local_IO_1 and Local_IO_2) are for local storage only (not replicated to arrays in DC-B). Two of the LUNs (Sync_IO_1 and Sync_IO_2) are synchronously replicated to the sister array in DC-B. Two of the LUNs (Async_IO_1 and Async_IO_2) are asynchronously replicated to the sister array in DC-B. There are three serverfarms in DC-A that point to the three arrays, and three Vservers that point to the three serverfarms.

A script is used to generate client FTP PUTs to write a file to each of the LUNs on each of the arrays. For those LUNs that are replicated, it was verified that the file was replicated successfully.

Test Procedure

The procedure used to perform the End-to-End Put test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify that each of the three Vservers (E2E-FTP-EMC-VIP, E2E-FTP-NETAPP-VIP and E2E-FTP-HP-VIP) on the active CSM in DC-A is configured and **OPERATIONAL** using the **show module csm 2 vserver name** *vserver* **detail** command.

For each of the three Vservers, the state should be **OPERATIONAL**.

Step 3 Verify that each of the three serverfarms (E2E-EMC, E2E-NETAPP and E2E-HP) on the active CSM in DC-A is configured and each of the five real servers are **OPERATIONAL** using the **show module csm 2 serverfarm name** *serverfarm* **detail** command.

Each of the five real servers in each serverfarm, should display as being **OPERATIONAL**.

On dca-penguin-11, verify that the file random_file.txt is in place.

Step 4 Use the **clear module csm 2 counters** command on the active CSM to clear the vserver and serverfarm statistics.

Step 5 Start the script that will generate the FTP PUTs to write the file to each of the LUNs.

Step 6 When the script completes, verify that it returned all successful return codes.

Step 7 Verify that the **Client pkts** and **Server pkts** count has increased for each of the vservers using the **show module csm 2 vserver name** *vserver* **detail** command.

The **Client pkts** and **Server pkts** fields should show a large positive value.

Verify that the random_file.txt file is now located in the following directories on dca-penguin-1:

/Local_IO_1/user-emc-local-1/

/Local_IO_2/user-emc-local-2/

/Sync_IO_1/user-emc-sync-1/

/Sync_IO_2/user-emc-sync-2/

/Async_IO_1/user-emc-async-1/

/Async_IO_2/user-emc-async-2/

Verify that the random_file.txt file is now located in the following directories on dca-penguin-2:

/Local_IO_1/user-na-local-1/

/Local_IO_2/user-na-local-2/

/Sync_IO_1/user-na-sync-1/

/Sync_IO_2/user-na-sync-2/

/Async_IO_1/user-na-async-1/

/Async_IO_2/user-na-async-2/

Verify that the random_file.txt file is now located in the following directories on dca-penguin-4:

/Local_IO_1/user-hp-local-1/

/Local_IO_2/user-hp-local-2/

/Sync_IO_1/user-hp-sync-1/

/Sync_IO_2/user-hp-sync-2/

/Async_IO_1/user-hp-async-1/

/Async_IO_2/user-hp-async-2/

Step 8 Verify, visually, that the file has been replicated to the appropriate LUNs on the remote arrays.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that all of the FTP PUTs will succeed.
- We expect that the file will be replicated correctly, synchronously and asynchronously.

Results

End-to-End Put passed.

Global Site Selector (GSS)

The Global Site Selector (GSS) leverages the Domain Name System (DNS) to provide clients with reliable and efficient content services. Domain to IP address mapping is performed, with consideration for availability, location and load of content servers. Using the GSS in combination with Cisco's Content Services Switch (CSS) or Cisco's Catalyst 6000 Content Switching Module (CSM) allows users to create Global Server Load Balancing (GSLB) networks.

The GSS may be deployed in a variety of locations in a customer's network to serve DNS requests, providing answers based on availability and preference. The GSS combines basic concepts of DNS with monitoring of answer status and providing users with IP addresses that are most appropriate for their content requests.

The GSS provides configuration and monitoring services through a central configuration manager, the Global Site Selector Manager (GSSM), and through a CLI that is available on each GSS. Configuration for a GSS network is mostly identical on all devices (global config model) and is entered by the user on a single GSS (central configuration model). For standard features the customer may choose to create a network of up to 8 GSSs with global/central configuration. The customer may instead choose to configure and monitor individual devices (local configuration model), in which case the GUI runs independently on each GSS and configuration is not shared.

The GSS receives DNS queries from client DNS proxies (D-Proxy), and matches these requests with a user-defined set of DNS Rules. A match on a DNS rule provides the list of 1st, 2nd and 3rd choice sets of answers that should be considered for the request.

Within a GSS network an answer is a host address which identifies a resource within a network that the GSS can direct a user to to respond to a content request. GSS answers are either a Virtual IP (VIP) Address associated with a server load balancer (SLB), a Name Server which can answer queries that the GSS cannot, or a Content Routing Agent (CRA) that use a resolution process called DNS race to send identical and simultaneous responses back to a user's D-proxy.

The DNS rule also defines the balancing methods that should be applied for choosing from each set of possible answers, and can be combined with advanced features including checking for answers with the closest network proximity to the client's requesting D-proxy, and use of a sticky database.

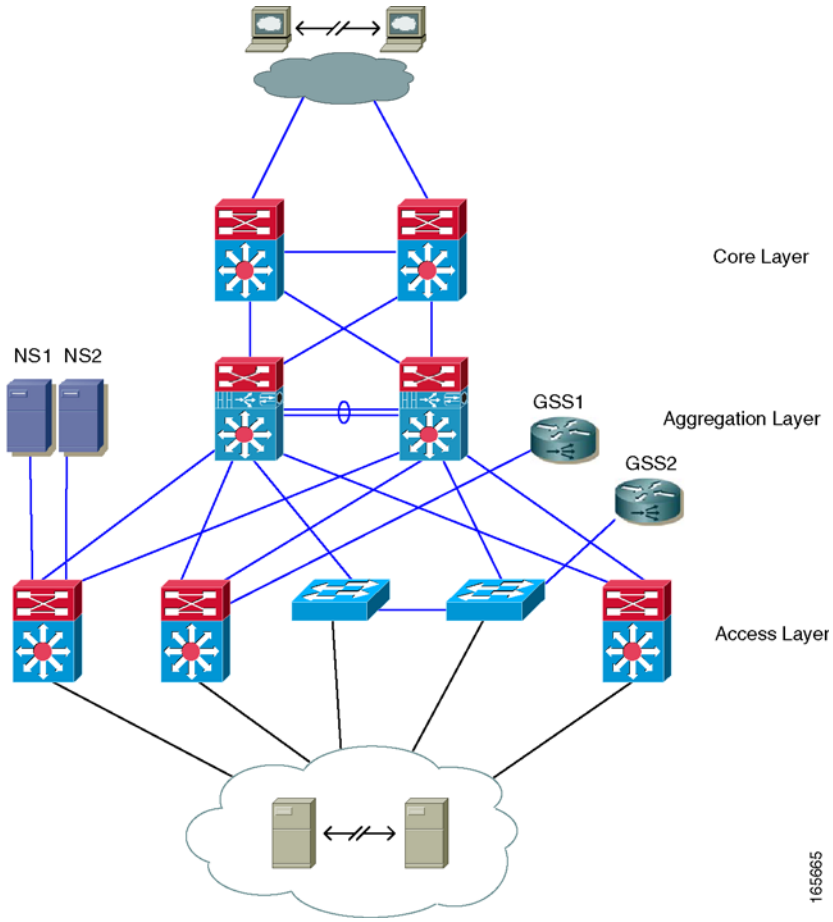
In addition to answering queries directly, the GSS offers the feature of forwarding requests to NS Forwarders, which will return a DNS response packet to the GSS, which in turn returns the exact same response packet to the originally requesting D-Proxy. This can be used for any query type on any domain, and is not limited to the record types supported by the GSS.

The tests in this chapter focus on the fundamental ability of the GSS working together with existing BIND Name Servers to provide global server load-balancing.

GSS Topology

The GSS's are integrated into the existing DCAP 2.0 topology (Figure 5-1) along with BIND Name Servers and tested using various DNS rules configured on the GSS. Throughout the testing, the GSS receives DNS queries sourced from client machines as well as via DNS proxies (D-Proxies). The Name Server zone files on the D-Proxies are configured to nsfoward DNS queries to the GSS to obtain authoritative responses. Time To Live (TTL) values associated with the various DNS resource records are observed and taken into consideration throughout the testing.

Figure 5-1 DCAP GSS Test Topology



Test Results Summary

Table 5-1 summarizes tests executed as part of the Cisco DCAP 2.0 testing initiative. Table 5-1 includes the feature or function tested, the section that describes the feature set the feature or function belongs to, the component tests for each feature or function, and whether the test is new in this phase of DCAP testing.

Note Test results are unique to technologies covered and actual scenarios in which they were tested. DCAP is designed to cover critical path areas and augment ongoing regression and systems testing.

Table 5-1 Cisco DCAP 2.0 GSS Testing Summary

Test Suites	Features/Functions	Tests	Results
GSS Test Cases, page 5-3	GSS Test Cases, page 5-3	1. GSS Backup Restore 2. GSS DNS Request Processing 3. GSS DNS Static Proximity 4. GSS DNS Global Sticky 5. GSS DNS Keepalive Verification 6. GSS Load Balancing Methods	

GSS Test Cases

The Global Site Selector (GSS) leverages DNS's distributed services to provide high availability to existing data center deployments by incorporating features above and beyond today's DNS services.

Functionality critical to global enterprises in Cisco DCAP 2.0 Storage Area Network (SAN) testing is described in the following section. Refer to Cisco Data Center Assurance Program (DCAP) 2.0 Configurations document for test device configurations.

The following tests were performed:

- GSS Backup Restore, page 5-3
- GSS DNS Request Processing, page 5-5
- GSS DNS Static Proximity, page 5-10
- GSS DNS Global Sticky, page 5-11
- GSS DNS Keepalive Verification, page 5-13
- GSS Load Balancing Methods, page 5-14

GSS Backup Restore

This test verified that both the GSS Database and the GSS Network Configuration on the Primary GSSM was successfully backed up and restored. DNS queries were sent from clients to both name servers and forwarded to both GSS's during the restore/backup process on the GSSM to ensure proper nsforwarding to the correct GSS.

Test Procedure

The procedure used to perform the GSS Backup Restore test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 On dcap-gss-1.gslb.com ensure the GSS status is primary by issuing the command **gss status** and on dcap-gss-2.gslb.com ensure the GSS status is standby by issuing the command **gss status**

Step 3 On dcap-gss-1.gslb.com create a full backup of your primary GSSM by issuing the command **gssm backup full gssmbk**

Step 4 On dcap-gss-1.gslb.com make a change to the GSSM database by creating a new source address list called list_1 with includes the source address 1.1.1.1/32.

Step 5 On dcap-gss-1.gslb.com create another full backup of your primary GSSM by issuing the command **gssm backup full gssmbk_new**

Step 6 On dcap-gss-1.gslb.com verify the full backup files were created and note the size of the files by issuing the command **dir**

Step 7 On dcap-gss-1.gslb.com stop the GSS, by issuing the command **gss stop**

Step 8 On both client machines; gss-winxp-1 and gss-linux-1 ensure the primary and secondary name servers are 10.0.5.111 and 10.0.5.102 respectively.

Step 9 While dcap-gss-1.gslb is stopped verify that the clients are still able to resolve DNS via dcap-gss-2.gslb. Verify dcap-gss-2.gslb responds with the expected result. Send a DNS A record query from both clients; gss-winxp-1 and gss-linux-1 to both of the name servers.

From gss-linux-1

dig @10.0.5.111 eng.gslb.com. a +qr

dig @10.0.5.102 eng.gslb.com. a +qr

From gss-winxp-1

nslookup

set d2

server 10.0.5.111

eng.gslb.com.

server 10.0.5.102

eng.gslb.com

Step 10 On both GSS's; dcap-gss-1.gslb.com and dcap-gss-2.gslb.com force a rotation and deletion of logs by issuing the command **rotate-logs** and the command **rotate-logs delete-rotated-logs**.

Step 11 On both GSS's; dcap-gss-1.gslb.com and dcap-gss-2.gslb.com, enable DNS console logging by issuing the command **logging disk enable** and the command **logging disk subsystem dnsserver priority debugging**. Ensure DNS logging is enabled by issuing the command **show logging**

Step 12 On both GSS's; dcap-gss-1.gslb.com and dcap-gss-2.gslb.com, enable real-time logging by issuing the command **show log follow**

Step 13 On GSS dcap-gss-2.gslb.com verify the DNS global statistics by issuing the command **show statistics dns global**

Step 14 On dcap-gss-1.gslb.com restore the full backup file named gssmbk.full by issuing the command **gssm restore gssmbk.full** and follow the prompts to reboot the GSS.

Step 15 When dcap-gss-1.gslb.com is back online verify the source address list list_1 which was added into the backup file gssmbk.full is no longer present in the db configuration on the GSS.

Step 16 Verify both GSS's respond with the expected result. Send a DNS A record query from both clients; gss-winxp-1 and gss-linux-1 to both of the name servers.

From gss-linux-1

dig @10.0.5.111 eng.gslb.com. a +qr

dig @10.0.5.102 eng.gslb.com. a +qr

From gss-winxp-1

nslookup

set d2

server 10.0.5.111

eng.gslb.com.

server 10.0.5.102

eng.gslb.com

Step 17 Stop background scripts to collect final status of network devices and analyze for error.

Step 18 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that the GSS software performs a full backup of the GSSM network configuration settings as well as the GSSM database which contains the global server load-balancing configuration information.

Results

GSS Backup Restore passed.

GSS DNS Request Processing

This test verified that the GSS responded properly when sent different DNS resource record types. DNS queries were sent from client machines directly to the GSS and from client machines to ns forwarding name servers (d-proxies) to the GSS.

Test Procedure

The procedure used to perform the GSS DNS Request Processing test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 On both client machines; gss-winxp-1 and gss-linux-1 ensure the primary and secondary name servers are 10.0.5.111 and 10.0.5.102 respectively.

Step 3 On both client machines; gss-winxp-1 and gss-linux-1; flush the DNS resolver cache.

Step 4 On both GSS's; dcap-gss-1.gslb.com and dcap-gss-2.gslb.com force a rotation and deletion of logs by issuing the command **rotate-logs** and the command **rotate-logs delete-rotated-logs**.

Step 5 On both GSS's; dcap-gss-1.gslb.com and dcap-gss-2.gslb.com, enable DNS console logging by issuing the command **logging disk enable** and the command **logging disk subsystem dnsserver priority debugging**. Ensure DNS logging is enabled by issuing the command **show logging**.

Step 6 On both GSS's; dcap-gss-1.gslb.com and dcap-gss-2.gslb.com, clear the DNS statistics by issuing the command **clear statistics dns** Ensure the DNS statistics have been cleared by issuing the command **show statistics dns global**.

Step 7 On both GSS's; dcap-gss-1.gslb.com and dcap-gss-2.gslb.com, enable real-time logging by issuing the command **show log follow**

Step 8 Verify both GSS's respond with the expected result. Send a DNS A record query from both clients; gss-winxp-1 and gss-linux-1 to both of the name servers.

From gss-linux-1

dig @10.0.5.111 eng.gslb.com. a +qr

dig @10.0.5.102 eng.gslb.com. a +qr

From gss-winxp-1

nslookup

set d2

server 10.0.5.111

eng.gslb.com.

server 10.0.5.102

eng.gslb.com

Step 9 On both GSS's; dcap-gss-1.gslb.com and dcap-gss-2.gslb.com, verify the DNS global statistics by issuing the command **show statistics dns global**

Step 10 Verify both GSS's respond with the expected result for host names that the GSS's are authoritative for when responding to a DNS query type of AAAA. Send a DNS AAAA record query from both clients; gss-winxp-1 and gss-linux-1 to both of the name servers.

From gss-linux-1

dig @10.0.5.111 eng.gslb.com. aaaa +qr

dig @10.0.5.102 eng.gslb.com. aaaa +qr

From gss-winxp-1

nslookup

set d2

set type=aaaa

server 10.0.5.111

eng.gslb.com.

server 10.0.5.102

eng.gslb.com

Step 11 Verify both GSS's respond with the expected result for host names that the GSS's are not authoritative for when responding to a DNS query type of A.

From gss-linux-1

dig @10.0.5.111 not-here.gslb.com. a +qr

dig @10.0.5.102 not-here.gslb.com. a +qr

From gss-winxp-1

nslookup

set d2

set type=a

server 10.0.5.111

not-here.gslb.com.

server 10.0.5.102

not-here.gslb.com.

Step 12 Verify the GSS responds with the correct/valid response when sending DNS A queries for which the GSS is not authoritative for by asking the GSS directly.

dig @101.1.32.11 not-here.gslb.com. a +qr

dig @101.1.32.12 not-here.gslb.com. a +qr

Step 13 Verify both GSS's respond with the expected result for host names that the GSS's are authoritative for the wildcard domain of .*\.gslb\.com. Ensure all other DNS rules on the GSS are suspended. Ask the GSS directly.

From gss-linux-1

dig @101.1.32.11 wildcard.gslb.com. a +qr

dig @101.1.32.12 wildcard.gslb.com. a +qr

From gss-winxp-1

nslookup

set d2

set type=a

server 101.1.32.11

wildcard.gslb.com.

server 101.1.32.12

wildcard.gslb.com.

Step 14 Verify the GSS responds with the correct/valid response when sending valid DNS A queries to the GSS for which the GSS is authoritative for the wildcard domain of .*\.gslb\.com. Ensure all other DNS rules on the GSS are suspended.

dig @10.0.5.111 eng.gslb.com. a

dig @10.0.5.102 eng.gslb.com. a

Step 15 Verify both GSS's respond with the expected result for host names that the GSS's are authoritative for the wildcard domain of .*\.gslb\.com but does not support the resource record type. Ensure all other DNS rules on the GSS are suspended. Send the DNS queries directly to the GSS.

From gss-linux-1

dig @101.1.32.11 wildcard.gslb.com. MX +qr

dig @101.1.32.12 wildcard.gslb.com. MX +qr

From gss-winxp-1

nslookup

set d2

set type=mx

server 101.1.32.11

wildcard.gslb.com.

server 101.1.32.12

wildcard.gslb.com.

Step 16 Verify both GSS's respond with the expected result for host names that the GSS's are authoritative for when responding to a DNS query type of A using TCP as a transport rather than UDP. Send the DNS queries directly to the GSS.

From gss-linux-1

dig @101.1.32.11 eng.gslb.com. a +tcp +qr

dig @101.1.32.12 eng.gslb.com. a +tcp +qr

Step 17 Verify both GSS's respond with the expected result for host names that the GSS's are authoritative for when responding to a DNS query type A queries and setting the UDP message buffer size (EDNS0 bytes). Send the DNS queries directly to the GSS.

From gss-linux-1

dig @101.1.32.11 eng.gslb.com a +bufsiz=1024 +qr

dig @101.1.32.12 eng.gslb.com a +bufsiz=1024 +qr

Step 18 Verify both GSS's respond with the expected result when nsforwarding DNS type A queries. Send the DNS queries directly to the GSS.

From gss-linux-1

dig @101.1.32.11 send-me-away.gslb.com +qr

dig @101.1.32.12 send-me-away.gslb.com +qr

From gss-winxp-1

nslookup

set d2

set type=a

server 101.1.32.11

send-me-away.gslb.com.

server 101.1.32.12

send-me-away.gslb.com.

Step 19 Verify both GSS's respond with the expected result when nsforwarding DNS type a queries using TCP rather than UDP. Send the DNS queries directly to the GSS.

From gss-linux-1

dig @101.1.32.11 send-me-away.gslb.com +tcp +qr

dig @101.1.32.12 send-me-away.gslb.com +tcp +qr

Step 20 Verify both GSS's respond with the expected result when nsforwarding DNS type MX queries. Send the DNS queries directly to the GSS.

From gss-linux-1

dig @101.1.32.11 mail.gslb.com mx +qr

dig @101.1.32.12 mail.gslb.com mx +qr

From gss-winxp-1

nslookup

set d2

set type=mx

server 101.1.32.11

mail.gslb.com.

server 101.1.32.12

mail.gslb.com.

Step 21 Verify both GSS's respond with the expected result for host names that the GSS's are authoritative for when responding to a DNS query type of MX. Send the DNS queries directly to the GSS.

From gss-linux-1

dig @101.1.32.11 eng.gslb.com. mx +qr

dig @101.1.32.12 eng.gslb.com. mx +qr

From gss-winxp-1

nslookup

set d2

set type=mx

server 101.1.32.11

eng.gslb.com.

server 101.1.32.12

eng.gslb.com

Step 22 Verify both GSS's respond with the expected result for host names that the GSS's are authoritative for when responding to a DNS query type of ANY. Send the DNS queries directly to the GSS.

From gss-linux-1

dig @101.1.32.11 eng.gslb.com. any +qr

dig @101.1.32.11 eng.gslb.com. any +qr

From gss-winxp-1

nslookup

set d2

set type=any

server 101.1.32.11

eng.gslb.com.

server 101.1.32.11

eng.gslb.com.

Step 23 Stop background scripts to collect final status of network devices and analyze for error.

Step 24 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the GSS to respond to various well formed, RFC based DNS queries in the proper manor.

Results

GSS DNS Request Processing passed.

GSS DNS Static Proximity

This test verified that the GSS responded with the correct answer(s) based on the source address of the d-proxy.

Test Procedure

The procedure used to perform the GSS DNS Static Proximity test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 On client's gss-winxp-1 and gss-linux-1, ensure the primary and secondary name server is 10.0.5.111 and 10.0.5.101, respectively.

Step 3 Flush the DNS resolver cache on client's gss-winxp-1 and gss-linux-1.

Step 4 On the GSS, force a rotation and deletion of logs by issuing the commands **rotate-logs** and **rotate-logs delete-rotated-logs**.

Step 5 On the GSS, enable DNS console logging by issuing the commands **logging disk enable** and **logging disk subsystem dnsserver priority debugging**. Ensure DNS logging is enabled by issuing the command **show logging**.

Step 6 On the GSS, clear the DNS statistics by issuing the command **clear statistics dns** . Ensure the DNS statistics have been cleared by issuing the command **show statistics dns global**.

Step 7 On the GSS, enable real-time logging by issuing the command **show log follow**.

Step 8 On the GSS, configure the DNS rule lb_dca_and_dcb with the source address list of via_penguin_111, and verify both GSS's respond with the expected result when using both d-proxy name servers.

dig @10.0.5.111 eng.gslb.com. a +qr

dig @10.0.5.102 eng.gslb.com. a +qr

Step 9 On the GSS, configure the DNS rule lb_dca_and_dcb with the source address list of via_penguin_102, and verify both GSS's respond with the expected result when using both d-proxy name servers.

dig @10.0.5.111 eng.gslb.com. a +qr

dig @10.0.5.102 eng.gslb.com. a +qr

Step 10 On the GSS, verify the DNS global statistics by issuing the command **show statistics dns global**, and verify the source address lists statistics by issuing the command **show statistics dns source-address**.

Step 11 On the GSS, configure the DNS rule lb_dca_and_dcb with the source address list of via_client_net, and verify both GSS's respond with the expected result when using both d-proxy name servers.

dig @10.0.5.111 eng.gslb.com. a +qr

dig @10.0.5.102 eng.gslb.com. a +qr

Step 12 On the GSS, configure the DNS rule lb_dca_and_dcb with the source address list of via_client_net, and verify both GSS's respond with the expected result when using both d-proxy name servers.

dig @101.1.32.11 eng.gslb.com. a +qr

dig @101.1.32.12 eng.gslb.com. a +qr

Step 13 On the GSS, configure the DNS rule dr_perfer_dca_via_nameserver with the source address list of via_penguin_111, and configure the DNS rule dr_perfer_dcb_via_clients with the source address list of via_client_net. Configure the domain list all_domains for both DNS rules. Verify both GSS's respond with the expected result when using both d-proxy name servers, as well as by asking each GSS directly.

dig @10.0.5.111 eng.gslb.com. a +qr

dig @10.0.5.102 eng.gslb.com. a +qr

dig @101.1.32.11 eng.gslb.com. a +qr

dig @101.1.32.12 eng.gslb.com. a +qr

Step 14 Stop background scripts to collect final status of network devices and analyze for error.

Step 15 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the GSS to respond with the correct answer(s) based on the source address of the d-proxy.

Results

GSS DNS Static Proximity passed.

GSS DNS Global Sticky

This test verified that the GSS properly replicated DNS responses to it's peer GSS while maintaining affinity based on the source of the d-proxy. VIP's were taken offline in order to ensure the proper answer was provided by the GSS and replicated to its peer.

Test Procedure

The procedure used to perform the GSS DNS Global Sticky test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 On client's gss-winxp-1 and gss-linux-1 ensure the primary and secondary name server is 10.0.5.111 and 10.0.5.101 respectively.

Step 3 Flush the DNS resolver cache on client's gss-winxp-1 and gss-linux-1.

Step 4 On the GSS, enable DNS console logging by issuing the command **logging disk enable** and the command **logging disk subsystem sticky priority debugging**. Ensure DNS logging is enabled by ensuring the command **show logging**

Step 5 On the GSS, force a rotation and deletion of logs by issuing the command **rotate-logs** and the command **rotate-logs delete-rotated-logs**.

Step 6 On the GSS, clear the DNS statistics by issuing the command **clear statistics sticky** Ensure the sticky statistics have been cleared by issuing the command **show statistics sticky global**

Step 7 On both GSS's dcap-gss-1.gslb.com and dcap-gss-2.gslb.com verify ntp is enabled by issuing the command **show ntp**

Step 8 Verify the GSS responds with the correct/valid response when sending a valid DNS A query to the name server in order to validate global sticky table entry.

dig @10.0.5.111 eng.gslb.com. a +qr

Step 9 On GSS dcap-gss-2.gslb.com, verify the sticky entry was replicated from dcap-gss-1.gslb.com to dcap-gss-2.gslb.com by issuing the command **show sticky database all**

Step 10 Verify the GSS responds with the correct/valid response when sending valid DNS A query's to the name server in order to validate the client receives the same answer in the sticky database from both GSS's.

dig @10.0.5.111 eng.gslb.com. a +qr

dig @10.0.5.102 eng.gslb.com. a +qr

Step 11 On both GSS's dcap-gss-1.gslb.com and dcap-gss-2.gslb.com inspect the sticky databases by issuing the command **sticky database dump file-name format xml** and **type file-name** on both GSS's and validate.

Step 12 Verify the GSS responds with the correct/valid response when sending valid DNS A queries to the name server in order to validate global sticky based on domain.

dig @10.0.5.111 eng.gslb.com. a +qr

dig @10.0.5.102 eng.gslb.com. a +qr

dig @10.0.5.111 fin.gslb.com. a +qr

dig @10.0.5.102 fin.gslb.com. a +qr

Step 13 On the CSM, suspend the VIP that resides in the sticky database on the GSS, by issuing the command **no inservice** for vserver RED-100K-VIP. Verify the VIP/Answer is offline on both GSS's by issuing the command **show statistics keepalive tcp list**

Step 14 Verify the GSS responds with the correct/valid response when sending valid DNS A query's to dcap-gss-1.gslb.com in order to validate a new VIP is issued by the GSS and the sticky database is updated.

dig @10.0.5.111 eng.gslb.com. a +qr

Step 15 Verify the GSS responds with the correct/valid response when sending valid DNS A query's to the GSS in order to validate global sticky based on domain list. Verify the same answer is returned to the client.

dig @10.0.5.111 eng.gslb.com. a +qr

dig @10.0.5.102 eng.gslb.com. a +qr

dig @10.0.5.111 fin.gslb.com. a +qr

dig @10.0.5.102 fin.gslb.com. a +qr

Step 16 Stop background scripts to collect final status of network devices and analyze for error.

Step 17 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that the GSS will track the DNS response returned for a client d-proxy and will return the same answer when the same client d-proxy makes a subsequent DNS request.
- We expect that the GSS will not return an A record to a client for which the VIP on the GSS is deemed offline.

- We expect that the GSS will replicate the DNS response to its peer GSS.

Results

GSS DNS Global Sticky passed.

GSS DNS Keepalive Verification

This test verified that the GSS properly returned the expected resource record containing the VIP of the CSM based on the keepalive properties.

Test Procedure

The procedure used to perform the GSS DNS Keepalive Verification test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 On client's gss-winxp-1 and gss-linux-1 ensure the primary and secondary name server is 10.0.5.111 and 10.0.5.101 respectively.

Step 3 Flush the DNS resolver cache on client's gss-winxp-1 and gss-linux-1.

Step 4 On dcap-gss-1.gslb.com and dcap-gss-2.gslb.com, force a rotation and deletion of logs by issuing the command **rotate-logs** and the command **rotate-logs delete-rotated-logs**.

Step 5 On dcap-gss-1.gslb.com and dcap-gss-2.gslb.com, enable DNS console logging by issuing the command **logging disk enable** and the command **logging disk subsystem keepalive priority debugging**. Ensure DNS logging is enabled by issuing the command **show logging**

Step 6 On dcap-gss-1.gslb.com and dcap-gss-2.gslb.com, clear the keepalive statistics by issuing the command **clear statistics keepalive all**

Step 7 On dcap-gss-1.gslb.com and dcap-gss-2.gslb.com, enable real-time logging by issuing the command **show log follow**

Step 8 On dcap-gss-1.gslb.com and dcap-gss-2.gslb.com verify all CSM VIP's are online by issuing the command **show statistics keepalive answer type vip list**

Step 9 On the CSM, suspend a VIP issuing the command **no inservice** for vserver vserver_name.

Step 10 On dcap-gss-1.gslb.com and dcap-gss-2.gslb.com, verify the GSS has taken the VIP 101.40.40.245 offline by issuing the command **show statistics keepalive answer type vip list** On dcap-gss-1.gslb.com and dcap-gss-2.gslb.com

Step 11 Verify the GSS responds with the correct IP's when sending valid DNS A query's to the GSS.

From gss-linux-1

dig @10.0.5.111 eng.gslb.com. a +qr

dig @10.0.5.111 fin.gslb.com. a +qr

dig @10.0.5.102 hr.gslb.com. a +qr

dig @10.0.5.102 market.gslb.com. a +qr

From gss-winxp-1

nslookup

set d2

set type=a

server 10.0.5.111

eng.gslb.com.

hr.gslb.com

server 10.0.5.102

fin.gslb.com

market.gslb.com

Step 12 On dcap-gss-1.gslb.com and dcap-gss-2.gslb.com, verify the proper rule is being matched by issuing the command **show statistics dns answer-group**

Step 13 Cause all the answers in all groups on the GSS to go offline by issuing the command **no inservice** for all vservers on the CSM. Verify all answers OFFLINE on both GSS's buy issuing the command **show statistics keepalive answer type vip list**

Step 14 Verify the GSS responds with the correct answers when sending valid DNS A query's to the GSS while the answers are OFFLINE.

From gss-linux-1

dig @10.0.5.111 eng.gslb.com. a +qr

dig @10.0.5.102 hr.gslb.com. a +qr

Step 15 Stop background scripts to collect final status of network devices and analyze for error.

Step 16 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the GSS keepalives to fail when the CSM VIP's are taken out of service (offline).
- We expect the proper answer to be responded to via the GSS based on the keepalive status of the answer.

Results

GSS DNS Keepalive Verification passed.

GSS Load Balancing Methods

We expect that the GSS will respond to a well formed DNS query properly using the load balancing method defined in the DNS rule on the GSS.

Test Procedure

The procedure used to perform the GSS Load Balancing Methods test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 On client's gss-winxp-1 and gss-linux-1 ensure the primary and secondary name server is 10.0.5.111 and 10.0.5.101 respectively.

Step 3 Flush the DNS resolver cache on client's gss-winxp-1 and gss-linux-1.

Step 4 On the GSS, force a rotation and deletion of logs by issuing the command **rotate-logs** and the command **rotate-logs delete-rotated-logs**.

Step 5 On the GSS, enable DNS console logging by issuing the command **logging disk enable** and the command **logging disk subsystem dnsserver priority debugging**. Ensure DNS logging is enabled by issuing the command **show logging**

Step 6 On the GSS, clear the DNS statistics by issuing the command **clear statistics dns** Ensure the DNS statistics have been cleared by issuing the command **show statistics dns global**

Step 7 On the GSS, enable real-time logging by issuing the command **show log follow**

Step 8 On the GSS, configure the rule lb_dca_and_dcb DNS for round robin load balancing.

Step 9 Verify the GSS responds with the correct/valid response when sending DNS A queries for which the GSS is authoritative for by asking the GSS directly.

dig @10.0.5.111 eng.gslb.com. a +qr

Step 10 On the GSS, configure the rule lb_dca_and_dcb DNS for weighted round robin load balancing.

Step 11 On the GSS, configure the answer dcb-vip3 101.40.29.254 with a weight of 5 and verify the GSS responds with the correct/valid response when sending DNS A queries for which the GSS is authoritative.

From gss-linux-1

dig @10.0.5.111 eng.gslb.com. a +qr

dig @10.0.5.102 eng.gslb.com. a +qr

From gss-winxp-1

nslookup

set d2

set type=a

server 10.0.5.111

eng.gslb.com.

server 10.0.5.102

eng.gslb.com

Step 12 On the GSS, configure the rule lb_dca_and_dcb DNS for a return record count of 8 and re test dig and nslookup but this time ask the GSS directly.

From gss-linux-1

dig @101.1.32.11 eng.gslb.com. a +qr

dig @101.1.32.12 eng.gslb.com. a +qr

From gss-winxp-1

nslookup

set d2

set type=a

server 101.1.32.11

eng.gslb.com.

server 101.1.32.12

eng.gslb.com

Step 13 On the GSS, configure the rule lb_dca_and_dcb DNS for ordered list load balancing.

Step 14 On the GSS, configure the answer group dca_dcb_vips in chronological order from 1 to 8, starting with 101.40.29.254 and ending with 101.40.40.248 and verify the GSS responds with the correct/valid response in the correct order when sending DNS A queries for which the GSS is authoritative. Ask the GSS directly.

From gss-linux-1

dig @101.1.32.11 eng.gslb.com. a +qr

dig @101.1.32.12 eng.gslb.com. a +qr

From gss-winxp-1

nslookup

set d2

set type=a

server 101.1.32.11

eng.gslb.com.

server 101.1.32.12

eng.gslb.com

Step 15 On the GSS, configure the lb_dca_and_dcb DNS rule for hashed load balancing and select hashed based on domain name.

Step 16 Verify the GSS responds with the correct/valid response when sending DNS A queries for which the GSS is authoritative. Send requests to each of the four sub domains multiple times, in order to verify affinity for each sub domain.

From gss-linux-1

dig @10.0.5.111 eng.gslb.com. a +qr

dig @10.0.5.111 hr.gslb.com. a +qr

dig @10.0.5.111 fin.gslb.com. a +qr

dig @10.0.5.111 market.gslb.com. a +qr

dig @10.0.5.102 eng.gslb.com. a +qr

dig @10.0.5.102 hr.gslb.com. a +qr

dig @10.0.5.102 fin.gslb.com. a +qr

dig @10.0.5.102 market.gslb.com. a +qr

From gss-winxp-1

nslookup

set d2

set type=a

server 10.0.5.111

eng.gslb.com.

hr.gslb.com.

fin.gslb.com.

market.gslb.com.

server 10.0.5.102

eng.gslb.com.

hr.gslb.com.

fin.gslb.com.

market.gslb.com.

Step 17 On the GSS, configure the lb_dca_and_dcb DNS rule for hashed load balancing and select hashed based on source address.

Step 18 Verify the GSS responds with the correct/valid response when sending DNS A queries for which the GSS is authoritative. Send requests to each of the four sub domains multiple times, to both name servers in order to verify affinity for each of the two name servers.

dig @10.0.5.111 eng.gslb.com. a +qr

dig @10.0.5.111 hr.gslb.com. a +qr

dig @10.0.5.111 fin.gslb.com. a +qr

dig @10.0.5.111 market.gslb.com. a +qr

dig @10.0.5.102 eng.gslb.com. a +qr

dig @10.0.5.102 hr.gslb.com. a +qr

dig @10.0.5.102 fin.gslb.com. a +qr

dig @10.0.5.102 market.gslb.com. a +qr

Step 19 Stop background scripts to collect final status of network devices and analyze for error.

Step 20 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the correct load balancing method is chosen and implemented by the GSS based on the load balancing algorithm defined in the DNS rule.

Results

GSS Load Balancing Methods passed.

CHAPTER 6

Wide Area Application Services (WAAS)

Cisco Wide Area Application Services 4.0 (WAAS) is a powerful application acceleration and WAN optimization solution for the branch office that improves the performance of any TCP-based application operating in a Wide Area Network (WAN) environment. The WAAS software is built on the WAFS framework and still provides WAFS functionality as well as some added optimization features.

With Cisco WAAS, enterprises can consolidate costly branch office servers and storage into centrally managed data centers, while still offering LAN-like service levels for remote users.

The solution offers a significantly lower total cost of ownership (TCO), greater application performance, more efficient WAN usage, and transparent integration with the network with secure, centralized manageability and control in an easy-to-implement package. Cisco WAAS provides the technologies necessary to enable consolidation of infrastructure into the data center while also providing application acceleration and Wide Area Network (WAN) optimization capabilities that achieve application delivery performance similar to that of a Local Area Network (LAN).

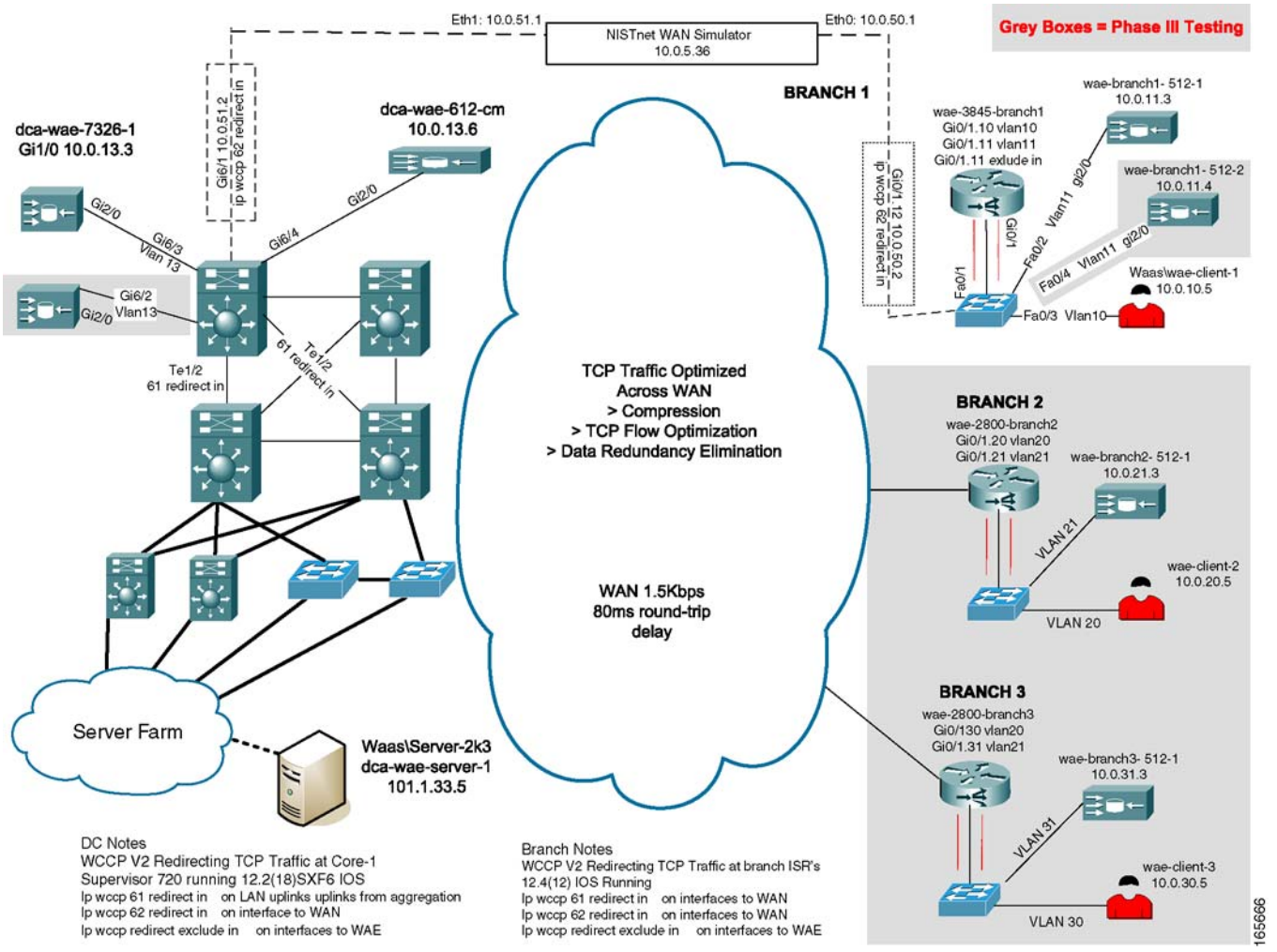
The solution provides the LAN-like performance across the WAN through a combination of technologies, including:

- Application acceleration—Mitigate latency and bandwidth through advanced protocol optimizations, including read-ahead, message prediction, and caching.
- Throughput optimization—Improve behavior of transport protocols to make them more efficient in WAN environments.
- Bandwidth optimization—Minimize the transmission of redundant data patterns through data redundancy elimination (DRE) and compression.

WAAS Topology

Cisco WAAS software running on Cisco Wide Area Application Engine (WAE) platforms is deployed in the data center and remote office locations as appliances attached to the LAN or as network modules (NME-WAE) integrated with the branch router. Cisco WAAS employs the Web Cache Communication Protocol (WCCP) v2 or Policy-Based Routing (PBR) to intercept traffic and transparently forward it to the local Cisco WAE on both sides of the network (Figure 6-1).

Figure 6-1 DCAP WAAS Test Topology



WAAS Test Results Summary

Table 6-1 summarizes tests executed as part of the Cisco DCAP 2.0 testing initiative. Table 6-1 includes the feature or function tested, the section that describes the feature set the feature or function belongs to, the component tests for each feature or function, and whether the test is new in this phase of DCAP testing.

Note Test results are unique to technologies covered and actual scenarios in which they were tested. DCAP is designed to cover critical path areas and augment ongoing regression and systems testing.

Table 6-1 Cisco DCAP 2.0 WAAS Testing Summary

Test Suites	Features/Functions	Tests	Results
Baseline, page 6-5	Device Management, page 6-5	1. SNMP MIB Walk WAE512 2. SNMP MIB Walk WAE7326 3. SNMP MIB Walk-WAE612	
	Reliability, page 6-7	1. Core Reload WAE7326 2. WAAS Edge Reload WAE512	
	CLI Functionality, page 6-9	1. Parser via Telnet 3845	CSCsh30404
	WCCP, page 6-10	1. WCCPv2 on Core WAE7326 2. WCCPv2 on EDGE-WAE 512 3. WCCPv2 on Edge-3845 4. WCCPv2 on CORE-Sup720	CSCsd12760 CSCsd12760
	NTP, page 6-15	1. NTP Configuration and Functionality	
	Authentication (Kerberos), page 6-17	1. Core Kerberos Authentication WAE7326 2. Edge Kerberos Authentication WAE512	
Optimization (DRE/TFO/LZ), page 6-20	WAFS, page 6-20	1. WAFS Configuration Verification	
	CIFS/Performance, page 6-22	1. CIFS Windows Acceleration Verification WAE512 2. CIFS Cache Hit Benchmark 3. CIFS Cache Miss Benchmark 4. CIFS Native WAN Benchmark	

WAAS DDTS Summary

Table 6-2 lists Development Defect Tracking System (DDTS) software bugs with descriptions, and comments filed by the DCAP testing team during Cisco DCAP 2.0 WAAS testing. Table 6-3 lists DDTS with descriptions encountered during Cisco DCAP 2.0 WAAS testing. Table 6-4 lists DDTS with descriptions of interest not encountered during Cisco DCAP 2.0 WAAS testing.

Table 6-2 Summary of DDTS Filed During Cisco DCAP 2.0 WAAS Testing

CSCsh30404	ISR: CPU hogs and crash after issuing show call malloc tcl contents. **Workaround:** Do not issue "show malloc tcl contents" on non voice images.

Table 6-3 Summary of DDTS Encountered During Cisco DCAP 2.0 WAAS Testing

CSCsd12760	Update WCCP CLI with WAAS Terminology

Table 6-4 Summary of DDTS of Interest yet Not Encountered During Cisco DCAP 2.0 WAAS Testing

CSCsg11506	EPM breaks connections if it doesn't intercept both traffic directions. **Workaround:** Ensure that both WCCP service groups 61 and 62 are on the same interface, and that is the WAN interface that connects to another WAE-accelerated site. Alternatively, configure WCCP redirection "in" on all relevant interfaces.
CSCse36646	Filesystem Corruption. **Workaround:** The corrupted partition should be deleted and re-create after reload. Ensure that after partition initialization you wait till the whole RAID is synced (might take more than an hour) before any reload.

WAAS Test Cases

Functionality critical to global enterprises in Cisco DCAP 2.0 Wide Area Application Services (WAAS) testing is described in the following sections. Refer to Cisco Data Center Assurance Program (DCAP) 2.0 Configurations document for test device configurations.

- Baseline, page 6-5
- Optimization (DRE/TFO/LZ), page 6-20

Baseline

The suite of baseline tests was created to verify the basic configuration and functionality of several integral features necessary for the WAAS software to perform correctly. The functionality of the Simple Network Management Protocol (SNMP) on Wide-Area Application Engine (WAE) devices serving as the edge, core, and central manger device was first tested. The reliability of the WAE devices configurations after reload was also tested. The Web Cache Communication Protocol version 2 (WCCPv2), the method used for TCP-based traffic interception and redirection, was then tested on the core and edge WAE devices, as well as, on the core and branch routers. The next protocol tested was the Network Time Protocol (NTP) which plays an important role in the WAAS solution. Each WAE devices clocks must be synchronized in order for the WAAS software to work properly. The NTP protocols functionality was verified on core, edge and central manager WAE devices. The final feature tested was the ability for the core and edge WAE devices to authenticate via Kerberos with a Windows based server.

The following test features were conducted:

- Device Management, page 6-5
- Reliability, page 6-7
- CLI Functionality, page 6-9
- WCCP, page 6-10
- NTP, page 6-15
- Authentication (Kerberos), page 6-17

Device Management

The Simple Network Management Protocol (SNMP) is an application layer protocol that facilitates the exchange of management information between network devices. It is part of the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite. SNMP enables network administrators to manage network performance, find and solve network problems, and plan for network growth. The tests performed verified the functionality and resiliency of the device as a series of SNMP Walks were performed on the WAE devices.

The following tests were performed:

- SNMP MIB Walk WAE512, page 6-6
- SNMP MIB Walk WAE7326, page 6-6
- SNMP MIB Walk-WAE612, page 6-7

SNMP MIB Walk WAE512

Simple Network Management Protocol (SNMP) is ubiquitous as a tool used by network administrators to manage network performance, find and solve network problems, and plan for network growth.

This test verified that 1000 SNMP walk of the MIB tree of a WAE-512 device did not cause any memory loss, tracebacks, or crashes. From a server, 1000 version 1 SNMP walks were performed on the device.

Test Procedure

The procedure used to perform the SNMP MIB Walk WAE512 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify the SNMP configuration of wae-branch1-512-1 using the **show running-config** command.

Step 3 From the Lickskillet server CLI perform 1000 SNMP walks on the DUT using the **snmpwalk** utility.

Step 4 Stop background scripts to collect final status of network devices and analyze for error.

Step 5 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect all SNMP walks will run without error.
- We expect that no tracebacks or crashes to occur on the DUT.

Results

SNMP MIB Walk WAE512 passed.

SNMP MIB Walk WAE7326

Simple Network Management Protocol (SNMP) is ubiquitous as a tool used by network administrators to manage network performance, find and solve network problems, and plan for network growth.

This test verified that 1000 SNMP walk's of the MIB tree of a WAE-7326 device did not cause any memory loss, tracebacks, or crashes. From a server, 1000 version 1 SNMP walks were performed on the device.

Test Procedure

The procedure used to perform the SNMP MIB Walk WAE7326 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify the SNMP configuration of dca-wae-612-1 using the **show running-config** command.

Step 3 From the Lickskillet server CLI perform 1000 SNMP walks on the DUT using the **snmpwalk** utility.

Step 4 Stop background scripts to collect final status of network devices and analyze for error.

Step 5 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect all SNMP walks will run without error.
- We expect that no tracebacks or crashes to occur on the DUT.

Results

SNMP MIB Walk WAE7326 passed.

SNMP MIB Walk-WAE612

Simple Network Management Protocol (SNMP) is ubiquitous as a tool used by network administrators to manage network performance, find and solve network problems, and plan for network growth.

This test verified that 1000 SNMP walk's of the MIB tree of a WAE-612 device did not cause any memory loss, tracebacks, or crashes. From a server, 1000 version 1 SNMP walks were performed on the device.

Test Procedure

The procedure used to perform the SNMP MIB Walk-WAE612 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify the SNMP configuration of dca-wae-612-1 using the **show running-config** command.

Step 3 From the Lickskillet server CLI perform 1000 SNMP walks on the DUT using the **snmpwalk** utility.

Step 4 Stop background scripts to collect final status of network devices and analyze for error.

Step 5 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect all SNMP walks will run without error.
- We expect that no tracebacks or crashes to occur on the DUT.

Results

SNMP MIB Walk-WAE612 passed.

Reliability

Reliability of network devices is essential for keeping the network functional. WAEs reliability testing included reloading the devices and verifying the configuration was restored after boot up.

The following tests were performed:

- Core Reload WAE7326, page 6-8
- WAAS Edge Reload WAE512, page 6-9

Core Reload WAE7326

This test verified that after a reload, the configuration on a Core WAE was the same as before the reload. The running configuration is verified and then copied to the startup configuration. The device is reloaded when it comes back online it is verified that there are no differences between the pre/post reload configuration. Since registration with the domain controller is a manual process, the WAE device is re-registered after the reload.

Test Procedure

The procedure used to perform the Core Reload WAE7326 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify the configuration on the DUT with the **show running-config** command.

Step 3 Copy the running configuration to the startup configuration with the **copy running-config startup-config** command.

Step 4 Verify the startup configuration with the **show startup-config** command.

Step 5 Reload the WAE with the **reload** command.

Step 6 After the reload verify that the WAE is configured the same as before the reload with the **show running-config** command.

Step 7 Through the Central Manger GUI navigate as follows:

Devices -> select the Core wae -> Authentication Methods -> Windows Domain

Enter the Administrator password and click the Register button to re-register the device.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the WAE device to come online after the reload.
- We expect the configuration to be the same after the reload.
- We expect the WAE device to re-register successfully after the reload.

Results

Core Reload WAE7326 passed.

WAAS Edge Reload WAE512

This test verified that after a reload, the configuration on a Edge WAE was the same as before the reload. The running configuration is verified and then copied to the startup configuration. The device is reloaded when it comes back online it is verified that there are no differences between the pre/post reload configuration. Since registration with the domain controller is a manual process, the WAE device is re-registered after the reload.

Test Procedure

The procedure used to perform the WAAS Edge Reload WAE512 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify the configuration on the DUT with the **show running-config** command.

Step 3 Copy the running configuration to the startup configuration with the **copy running-config startup-config** command.

Step 4 Verify the startup configuration with the **show startup-config** command.

Step 5 Reload the WAE with the **reload** command.

Step 6 After the reload verify that the WAE is configured the same as before the reload with the **show running-config** command.

Step 7 Through the Central Manger GUI navigate as follows:

Devices -> select the Core wae -> Authentication Methods -> Windows Domain

Enter the Administrator password and click the Register button to re-register the device.

Step 8 Stop background scripts to collect final status of network devices and analyze for error.

Step 9 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the WAE device to come online after the reload.
- We expect the configuration to be the same after the reload.
- We expect the WAE device to re-register successfully after the reload

Results

WAAS Edge Reload WAE512 passed.

CLI Functionality

The command-line interface (CLI) is used to configure and show network devices. The test performed verified the ability of the 3845 ISR to execute all the show and clear commands, found when doing a "long help" or "show ?" and "clear ?" on the device CLI, without crashing or interfering with the devices functionality.

The following test was performed:

- Parser via Telnet 3845, page 6-10

Parser via Telnet 3845

An automated script was used to test the valid show and clear commands on a 3845 ISR.

The script does a long help on the parser tree and executes both valid and invalid variations of all the show commands. Both the CPU and memory of the router are monitored to ensure that no memory leaks or unexpected CPU utilization is seen.

Several commands("show rmon" and "show parser dump") were not executed due to the long time it takes the to run or due to the fact that the command prompts for input causing the script to eventually time out and crash. "Show call malloc tcl contents" was also excluded due to CSCsh30404 which caused the router to crash when executed.

Test Procedure

The procedure used to perform the Parser via Telnet 3845 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Begin executing the show and clear commands on the device under test.

Step 3 Stop background scripts to collect final status of network devices and analyze for error.

Step 4 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that no tracebacks or crashes will occur.
- We expect no unacceptable impact on the CPU or memory.

Results

Parser via Telnet 3845 passed with exception CSCsh30404.

WCCP

WCCPv2 is a Cisco-developed content-routing technology that enables you to integrate content engines, such as Wide-Area Application Engines, into your network infrastructure. It is used for transparent interception and redirection of application and file traffic in branch offices and data centers to the local WAE. The tests performed verified the configuration and functionality of WCCPv2 in the WAAS/DCAP topology.

The following tests were performed:

- WCCPv2 on Core WAE7326, page 6-11
- WCCPv2 on EDGE-WAE 512, page 6-11
- WCCPv2 on Edge-3845, page 6-12
- WCCPv2 on CORE-Sup720, page 6-14

WCCPv2 on Core WAE7326

The Web Cache Communication Protocol (WCCP) is a Cisco-developed content-routing technology that allows integration of WAE devices, among other content engines, into your network infrastructure. The WAE devices are logically deployed using WCCPv2, providing transparent network interception and redirection of packets, high-availability clustering, and load sharing.

This test verified the basic configuration and functionality of WCCPv2 on a Core WAE device. The Core device is a WAE-7326.

Test Procedure

The procedure used to perform the WCCPv2 on Core WAE7326 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify the WCCP configuration on the core device(s) with the **show running-config** command.

WCCP version 2 should be configured and the IP address of gateway to the WCCP router should be configured as the IP address for router-list 1. TCP promiscuous mode service should also be turned on with router-list 1 assigned.

Step 3 Verify that WCCP is enabled on the device with the **show wccp status** command.

Step 4 Verify that the WCCP router is seen by issuing the **show wccp routers**.

The WCCP router has service 61 and 62 enabled. The *Router ID* is the Loopback address of the WCCP router. The *Sent to* IP address is the IP of the gateway to the WCCP router.

Step 5 Verify the file engine list for services 61 and 62 are seen by the WCCP router by issuing the **show wccp file-engines** command.

Step 6 Stop background scripts to collect final status of network devices and analyze for error.

Step 7 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that WCCP version 2 will be configured.
- We expect that the WCCP router list will be configured with the IP address of the Gateway to the WCCP router.
- We expect that the core device will see the neighboring WCCP router for services 61 and 62.
- We expect no unacceptable impact on the CPU or memory.

Results

WCCPv2 on Core WAE7326 passed with exception CSCsd12760.

WCCPv2 on EDGE-WAE 512

The Web Cache Communication Protocol (WCCP) is a Cisco-developed content-routing technology that allows you to integrate WAE devices into your network infrastructure. The Cisco WAAS Network Module is logically deployed using WCCPv2, providing transparent network interception and

redirection of packets, high-availability clustering, and load sharing. Alternatively, Policy Based Routing (PBR) can be employed to deploy the Cisco WAAS Network Module in a high-availability configuration with failover support.

This test verified the basic configuration and functionality of WCCPv2 on a Edge WAE device. In this case the device is a WAE-512.

Test Procedure

The procedure used to perform the WCCPv2 on EDGE-WAE 512 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify the WCCP configuration on the edge device(s) with the **show running-config** command.

WCCP version 2 should be configured and the gateway to the WCCP router should configured as the IP address for router-list 1. TCP promiscuous mode service should also be enabled on with router-list 1 assigned.

Step 3 Verify that WCCP is enabled on the device with the **show wccp status** command.

Step 4 Verify that the WCCP router is seen by issuing the **show wccp routers** command.

The WCCP router has service 61 and 62 enabled. The "Router ID" is the highest IP address on the WCCP router. The "Sent to" IP address is the IP of the gateway to the WCCP router.

Step 5 Verify the File Engine List for services 61 and 62 are seen by the WCCP router by issuing the **show wccp file-engines** command.

Step 6 Stop background scripts to collect final status of network devices and analyze for error.

Step 7 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that WCCP version 2 will be configured.
- We expect that the WCCP router list will be configured with the IP address of the Gateway to the WCCP router.
- We expect that the core device will see the neighboring WCCP router for services 61 and 62.

Results

WCCPv2 on EDGE-WAE 512 passed with exception CSCsd12760.

WCCPv2 on Edge-3845

The Web Cache Communication Protocol (WCCP) is a Cisco-developed content-routing technology that allows integration of WAE devices, among other content engines, into your network infrastructure. The WAE devices are logically deployed using WCCPv2, providing transparent network interception and redirection of packets, high-availability clustering, and load sharing. Once the WAE devices have joined the service group with the router, the router will monitor traffic for flows that should be forwarded to the WAE instead of the original destination. With WCCPv2, up to 32 WAEs can join a service group with up to 32 routers.

This test verified the basic configuration and functionality of WCCPv2 on a Branch ISR. The ingress LAN and WAN port configuration are first verified. The ingress ports from the WAE device(s) configuration(s) are then verified. Finally the Assigned Hash info for service 61 and 62 is verified. In this test the branch device is a 3845 ISR.

Test Procedure

The procedure used to perform the WCCPv2 on Edge-3845 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify the WCCP configuration on the core device with the **show running-config** command.

WCCP version 2 is enabled by default. Verify *ip wccp 61* and *ip wccp 62* are configured globally.

Step 3 On the ingress interface from the WAN verify that **ip wccp 62 redirect in** is configured with the **show running config interface *interface*** command.

Step 4 On the ingress interface from the LAN verify that **ip wccp 61 redirect in** is configured with the **show running config interface *interface*** command.

Step 5 On the interface connecting the WAE device verify that **ip wccp redirect exclude in** is configured with the **show running config interface *interface*** command.

Step 6 Verify WCCPv2 is running and that services 61 and 62 are enabled with the **show ip wccp** command.

For services 61 and 62 the *Number of Service Group Clients* is equal to the number of WAE devices connected to the router and the *Number of Service Group Routers* should be 1.

Note If no Loobpack address is configured on the router then the highest IP address will serve as the Router ID.

Step 7 Verify the interface configuration for the DUT by issuing the **show ip wccp interfaces** command.

The *Input services* for the ingress WAN and LAN interfaces should be 1 and *Exclude In* should be FALSE. For the WAE interface all services should be 0 and *Exclude In* should be TRUE.

Step 8 Verify that the WAE device(s) are correctly seen by issuing **show ip wccp *service* detail** command for service 61 and 62.

The *Assigned Has Info* depends on the number of WAE devices seen by the router. Verify that if more than one is present that the hash is correct.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect service 61 redirect-in to be configured on the ingress LAN ports.
- We expect service 62 redirect-in to be configured on the ingress WAN ports.
- We expect exclude-in to be configured on the WAE interface.
- We expect that the Assigned Hash Info for services 61 and 62 will be as expected.

Results

WCCPv2 on Edge-3845 passed.

WCCPv2 on CORE-Sup720

The Web Cache Communication Protocol (WCCP) is a Cisco-developed content-routing technology that allows integration of WAE devices, among other content engines, into your network infrastructure. The WAE devices are logically deployed using WCCPv2, providing transparent network interception and redirection of packets, high-availability clustering, and load sharing. Once the WAE devices have joined the service group with the router, the router will monitor traffic for flows that should be forwarded to the WAE instead of the original destination. With WCCPv2, up to 32 WAEs can join a service group with up to 32 routers.

This test verified the basic configuration and functionality of WCCPv2 on a Supervisor 720. The ingress LAN and WAN port configuration are first verified. The ingress ports from the WAE device(s) configuration(s) are then verified. Finally the Assigned Hash info for service 61 and 62 is verified.

Test Procedure

The procedure used to perform the WCCPv2 on CORE-Sup720 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify that WCCPv2 services 61 and 62 are enabled globally on the core device with the **show running-config** command.

WCCP version 2 is enabled by default. Verify *ip wccp 61* and *ip wccp 62* are configured globally.

Step 3 On the ingress port from the WAN verify that **ip wccp 62 redirect in** is configured with the **show-running config interface *interface*** command.

Step 4 On the ingress ports from the aggregation layer of the Data Center LAN, verify that **ip wccp 62 redirect in** is configured with the **show running-config interface *interface*** command.

Step 5 On the interface to the WAE devices, verify that **ip wccp redirect exclude in** is configured with the **show running-config interface *interface*** command.

Step 6 Verify the routers loopback address with the **show interfaces loopback 0** command.

This will be the WCCP *Router ID* address.

Step 7 Verify WCCPv2 is running and that services 61 and 62 are enabled with the **show ip wccp** command.

In this case the *Router Identifier* is the routers loopback address and the *Protocol Version* is 2.0. For services 61 and 62 the *Number of Cache Engines* is equal to the number of WAE devices connected to the router and the *Number of routers* should be 1.

Step 8 Verify that the WAE device(s) are correctly seen by issuing **show ip wccp *service* detail** command for service 61 and 62.

The *Assigned Has Info* depends on the number of WAE devices seen by the router. Verify that if more than one is present that the hash is correct.

Step 9 Stop background scripts to collect final status of network devices and analyze for error.

Step 10 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect service 61 redirect-in to be configured on the ingress LAN ports.
- We expect service 62 redirect-in to be configured on the ingress WAN ports.
- We expect exclude-in to be configured on the WAE interface.
- We expect that the Assigned Hash Info for services 61 and 62 will be as expected.

Results

WCCPv2 on CORE-Sup720 passed.

NTP

The Network Time Protocol (NTP) allows the synchronization of time and date settings for the different geographical locations of the devices in your WAAS network. The tests performed verified the configuration and functionality of NTP on core, edge and central manager WAE devices.

The following test was performed:

- NTP Configuration and Functionality

NTP Configuration and Functionality

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. In order for WAFS/WAAS to work correctly all the devices clocks must be synchronized using NTP. This is because timestamps play an important role in determining whether or not the data being accessed has been changed since the last time it was accessed.

This test verified the basic configuration and functionality of NTP on the Central Manager, Core, and Edge device(s). Through the Central Manager GUI, The Central Manager was first synchronized to a NTP server that is used throughout our lab networks. Each WAE device was then configured through the GUI to be synchronized to the Central Manager. The configuration and clock for each device was verified via the CLI.

Test Procedure

The procedure used to perform the NTP Configuration and Functionality test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Navigate to the Windows Name Services page in the CM GUI as follows:

Devices -> select the Central Manager (CM) -> General Settings -> Miscellaneous -> Date/Time -> NTP.

Check the enable box and enter the IP address of the global NTP server in the *NTP Server:* field and click the submit button to apply the configuration. Navigate back to the Time Zone page and select the applicable time zone and click submit to apply the configuration.

Step 3 Verify the NTP configuration on the Central Manger by issuing the **show running-config** command via the CLI.

Step 4 Verify the CM clock is in sync with the global NTP server which is located on goel (172.18.177.132). SSH to goel and issue **ps -ef | grep /usr/lib/inet/xntpd** to verify the NTP daemon is running. The issue the **date** command at the prompt to get the time. On the CM issue the **show clock** command and verify that the two timestamps are in sync.

Step 5 The Central Manager is now synchronized to a global NTP server. The rest of the devices can now be synchronized to Central Manager.

Navigate in the GUI as follows:

Click Devices -> Device Groups -> All Device Groups -> General Settings -> Miscellaneous -> Date/Time -> NTP.

Check the enable box and enter the IP address of the CM in the *NTP Server:* field and click the submit button to apply the configuration. Navigate back to the *Time Zone* page and select the applicable time zone and click submit to apply the configuration.

Step 6 To push this configuration out to all the device look to the right of the part of the page near the top that reads "Time Zone Settings for Device Group, AllDevicesGroup."

Hover the mouse pointer over the icons and click the last one that should have a pop up box that reads *Force Settings on all Devices in Group*. You will see a pop up that lists the devices that the configuration will be sent to. Click OK to configure.

Step 7 Verify the configuration on each device by issuing the **show running-config** command via the CLI on each device configured in the *AllDevicesGroup*.

The IP address should be that of the CM(10.0.13.6).

Step 8 Verify that each device is now synchronized to the CM by issuing the **show ntp status** and **show clock** commands.

Each device should have NTP enabled and the CM IP address as the server list. All the times should be in sync.

Step 9 Verify that all clients and servers are synchronized to this NTP server. On Windows clients/servers first open a command prompt by clicking Start -> Run and type cmd followed by OK.

At the command prompt type "w32tm /config /syncfromflags:manual /manualpeerlist:*PeerList*" and press Enter.(Peer list is the IP of the CM) Next type "w32tm /config /update" and press Enter To resync at any time type "w32tm /resync" at the prompt.

Step 10 Stop background scripts to collect final status of network devices and analyze for error.

Step 11 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that the Central Manger will be synchronized with the lab NTP server.
- We expect that all WAE devices in the network will accept the NTP configuration by using the GUI.
- We expect that the WAE devices in the network will synchronize to the Central Manger clock.

Results

NTP Configuration and Functionality passed.

Authentication (Kerberos)

Kerberos is a network authentication protocol to provide strong authentication for client/server applications by using secret-key cryptography. The WAE devices in the WAAS network are configured to authenticate with Windows file servers using this protocol. The tests performed verified the ability of the core and edge WAE devices to properly register with the Windows file server in the data center.

The following tests were performed:

- Core Kerberos Authentication WAE7326
- Edge Kerberos Authentication WAE512

Core Kerberos Authentication WAE7326

Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography. Kerberos was created by MIT as a solution to these network security problems. The Kerberos protocol uses strong cryptography so that a client can prove its identity to a server (and vice versa) across an insecure network connection. After a client and server has used Kerberos to prove their identity, they can also encrypt all of their communications to assure privacy and data integrity as they go about their business. MIT provides Kerberos in source form so that anyone who wishes to use it may look over the code for themselves and assure themselves that the code is trustworthy. In addition, for those who prefer to rely on a professionally supported product, Kerberos is available as a product from many different vendors.

This test verified that the Core WAE device authenticates properly via Kerberos to a server running Windows Server 2003. The configuration and the authentication status of the Core device is verified through both the Central Manager GUI and the CLI.

Test Procedure

The procedure used to perform the Core Kerberos Authentication WAE7326 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Navigate to the Windows Name Services page in the GUI as follows:

Devices -> select the Core wae -> Show Advanced -> General Settings -> Network -> Window Name Services.

Enter in the Workgroup or Domain Name(short name only) and WINS server IP address and click the submit button to apply the configuration.

Step 3 Verify the configuration on the DUT with the **show running-config** command.

Step 4 Navigate to the Windows Name Services page in the GUI as follows:

Devices -> select the Core wae -> Authentication Methods

Check the check box for *Authentication Login Methods* and select "WINDOWS" as the *Primary Login Method* and "local" for the *Secondary Login Method*. Check the check box for *Authorization Methods* and select "WINDOWS" as the *Primary Login Method* and "local" for the Secondary Login Method. Click the submit button to apply the configuration.

Step 5 Verify the authentication method configuration on the DUT by issuing the **show running-config** command.

The *primary authentication login* and *primary configuration login* should be for the windows-domain and the *secondary authentication login* and *secondary configuration login* should be local.

Step 6 Navigate to the Windows Domain page in the GUI as follows:

Devices -> select the Core wae -> Authentication Methods -> Windows Domain

Uncheck the box for *NTLM enabled* and check the box for *Kerberos* enabled. In the Realm field enter the fully qualified name of the domain. In the *Key Distribution Center* field enter the IP address of the Kerberos device to authenticate with followed by port 88. In the *Domain Controller* field enter the name of the Domain Controller. In the *Domain administrator* username field enter the username in the domain\username format. Click the submit button to apply the configuration.

Step 7 Verify the authentication method configuration on the DUT by issuing the **show running-config** command.

Step 8 In the *Domain administrator password* fields enter the administrator password and click the Register button.

Step 9 Enable pop-ups for the site and after two minutes click the *Show Authentication Status* button.

Verify that each test in the pop up has a green check under status. The four checks are SMB Commands executed on the DUT.

Step 10 Verify the registration was successful by issuing these commands via the CLI:

windows-domain diagnostics wbinfo -t

windows-domain diagnostics wbinfo --sequence

windows-domain diagnostics wbinfo --domain-info=WAAS

The Time Skew is checked by comparing the clock time on the WAE device with the clock on the domain controller. For kerberos to work these must be within 5 minutes of each other. To check the time on the WAE issue the **show ntp status** command.

Step 11 Repeat the Kerberos authentication steps(2-10) on the other Core WAE and verify that the device has registered with the domain controller.

Step 12 Stop background scripts to collect final status of network devices and analyze for error.

Step 13 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that the commands configured via the GUI will be applied to the running-configuration.
- We expect that the authentication status on the core WAE device(s) will successfully register with the domain controller.

Results

Core Kerberos Authentication WAE7326 passed.

Edge Kerberos Authentication WAE512

Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography. Kerberos was created by MIT as a solution to these network security problems. The Kerberos protocol uses strong cryptography so that a client can

prove its identity to a server (and vice versa) across an insecure network connection. After a client and server has used Kerberos to prove their identity, they can also encrypt all of their communications to assure privacy and data integrity as they go about their business. MIT provides Kerberos in source form so that anyone who wishes to use it may look over the code for themselves and assure themselves that the code is trustworthy. In addition, for those who prefer to rely on a professionally supported product, Kerberos is available as a product from many different vendors.

This test verified that the Edge WAE device authenticates properly via Kerberos to a server running Windows Server 2003. The configuration and the authentication status of the Edge device is verified through both the Central Manager GUI and the CLI.

Test Procedure

The procedure used to perform the Edge Kerberos Authentication WAE512 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Navigate to the Windows Name Services page in the GUI as follows:

Devices -> select the Edge wae -> Show Advanced -> General Settings -> Network -> Window Name Services.

Enter in the Workgroup or Domain Name(short name only) and WINS server IP address and click the submit button to apply the configuration.

Step 3 Verify the configuration on the DUT with the show **running-config** command.

Step 4 Navigate to the Windows Name Services page in the GUI as follows:

Devices -> select the Edge wae -> Authentication Methods

Check the check box for *Authentication Login Methods* and select "WINDOWS" as the *Primary Login Method* and "local" for the *Secondary Login Method*. Check the check box for *Authorization Methods* and select "WINDOWS" as the *Primary Login Method* and "local" for the Secondary Login Method. Click the submit button to apply the configuration.

Step 5 Verify the authentication method configuration on the DUT by issuing the **show running-config** command.

The *primary authentication login* and *primary configuration login* should be for the windows-domain and the *secondary authentication login* and *secondary configuration login* should be local.

Step 6 Navigate to the Windows Domain page in the GUI as follows:

Devices -> select the Edge wae -> Authentication Methods -> Windows Domain

Uncheck the box for *NTLM enabled* and check the box for *Kerberos* enabled. In the Realm field enter the fully qualified name of the domain. In the *Key Distribution Center* field enter the IP address of the Kerberos device to authenticate with followed by port 88. In the *Domain Controller* field enter the name of the Domain Controller. In the *Domain administrator* username field enter the username in the domain\username format. Click the submit button to apply the configuration.

Step 7 Verify the authentication method configuration on the DUT by issuing the **show running-config** command.

Step 8 In the *Domain administrator password* fields enter the administrator password and click the Register button.

Step 9 Enable pop-ups for the site and after two minutes click the *Show Authentication Status* button.

Verify that each test in the pop up has a green check under status. The four checks are SMB Commands executed on the DUT.

Step 10 Verify the registration was successful by issuing these commands via the CLI:

windows-domain diagnostics wbinfo -t

windows-domain diagnostics wbinfo --sequence

windows-domain diagnostics wbinfo --domain-info=WAAS

The Time Skew is checked by comparing the clock time on the WAE device with the clock on the domain controller. For kerberos to work these must be within 5 minutes of each other. To check the time on the WAE issue the **show ntp status** command.

Step 11 Repeat the Kerberos authentication steps(2-10) on the any other Edge WAE and verify that the device has registered with the domain controller.

Step 12 Stop background scripts to collect final status of network devices and analyze for error.

Step 13 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that the commands configured via the GUI will be applied to the running-configuration.
- We expect that the authentication status on the Edge WAE device(s) will successfully register with the domain controller.

Results

Edge Kerberos Authentication WAE512 passed.

Optimization (DRE/TFO/LZ)

The basic configuration for Wide Area File Services (WAFS) implementation was first verified in this suite. The Common Internet File System (CIFS) functionality through the WAE devices was then verified. Finally, the WAFS Benchmark tool was used to baseline the open, save, and close times for several different sizes ranging from 50k to 2MB of Microsoft Office files from a centralized file share on a Windows 2003 file server across the WAN with, and without, optimization.

The following test features were conducted:

- WAFS, page 6-20
- CIFS/Performance, page 6-22

WAFS

Cisco Wide Area File Services (WAFS) software overcomes WAN latency and bandwidth limitations with proprietary Cisco optimization technologies, offering users at branch offices a LAN-like experience when accessing the centralized files over the WAN. This facilitates the consolidation of all branch-office data into central file servers in your data center. WAAS contains a full implementation of the industry-leading WAFS product. The test performed verified the setup of WAFS on the WAAS software platform.

The following tests were performed:

- WAFS Configuration Verification

WAFS Configuration Verification

Cisco Wide Area File Services (WAFS) software overcomes WAN latency and bandwidth limitations with proprietary Cisco optimization technologies, offering users at branch offices a LAN-like experience when accessing the centralized files over the WAN.

This test verified the WAFS configuration for the WAAS network. The configuration of the appropriate system and services is verified. Then the WAFS policies are verified within the Central manager.

Test Procedure

The procedure used to perform the WAFS Configuration Verification test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Frpm a client, open an Internet Explorer Browser to the Central Manager.

Step 3 Verify that the WAFS Core Cluster is created (CM > Devices > Device Groups > Core Cluster).

Step 4 Verify that the WAFS Core WAE is assigned to this cluster (CM > Devices > Device Groups > Core Devices > Members, also CM > Devices > Core WAE > File Services > Core Configuration).

Step 5 Verify that the WAFS Core service is started on the WAFS Core WAE (WAFS Core Device GUI).

Step 6 Ensure that the file server is configured (CM > Services > File > File Servers > server-2k3.waas.domain.com).

Step 7 Verify that the file server is resolvable from the WAFS Core Cluster (CM > Services > File > File Server > server-2k3.waas.domain.com > Assign Core Clusters > Resolve).

Step 8 Verify that the WAFS Edge WAE has the Edge service enabled and the correct interception and port configuration is applied (CM > Devices > (WAE) > File Services > Edge Configuration).

Step 9 Verify that the connectivity directive is defined (CM > Services > File > Connectivity).

Step 10 Verify that the connectivity directive lists the file server as "exported" (CM > Services > File > Connectivity > WAE-Branch > File Server Settings).

Step 11 Verify that the connectivity directive lists the appropriate edge devices or groups (CM > Services > File > Connectivity > WAE-Branch > Assign Edge Devices).

Verify that each edge WAE has a green check next to it and that the status is "Online."

Step 12 Verify that the WAN parameters in the connectivity directive are accurate (CM > Services > File > Connectivity > WAE-Branch > WAN Utilization).

The WAN Defaults are: Maximum allocated bandwidth: 1544Kbits/sec Minimum roundtrip delay: 80ms

Step 13 Verify that the WAFS Accept policy is applied against a device group (CM > Devices > Device Group > All Devices > Acceleration > Policies > Definition).

Step 14 Verify that the WAFS Accept policy is assigned to the application "WAFS", application classifier is set to "Match All Traffic", and action is set to "Accelerate using CIFS adapter."

Step 15 Verify that the WAFS Accept policy 'enable' box is checked.

Step 16 Verify that the WAFS Transport policy is applied against the same device group (CM > Devices > Device Group > All Devices > Acceleration > Policies > Definition).

Step 17 Verify that the WAFS Transport policy is assigned to the application "WAFS", application classifier is set to "Match All Traffic", and action is set to "Full Optimization."

Step 18 Verify that the WAFS Transport policy 'enable' box is checked.

Step 19 Verify the WAFS map adaptor configuration on the WAE devices by issuing the **show run** command.

Step 20 Verify that the WAFS Edge and WAFS Core WAEs have established optimized connections between them using the **show tfo connection summary** commands. Look for connections using server-port 4050.

For more detailed statistics issue the **show tfo connection server-port 4050** command.

Step 21 Then, verify in the WAE Device GUI for both the Edge and Core that they are connected. Visit WAFS Edge > Monitoring and WAFS Core > Monitoring and validate that connection data is being exchanged. A green checkmark should appear.

Step 22 Stop background scripts to collect final status of network devices and analyze for error.

Step 23 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that a Core cluster will have been created.
- We expect that a Core WAE device is assigned to the core cluster.
- We expect that the Core service is started on the Core WAE.
- We expect that a file server is configured.
- We expect that the file server name is resolvable.
- We expect that the Edge service is started on the Edge WAE.
- We expect that the connectivity directive will list the file server as exported.
- We expect that the connectivity directive lists the edge device(s).
- We expect that the WAFS policies are configured on the Core and Edge device(s).
- We expect that the WAFS Accept/Transport policies are configured and enabled.

Results

WAFS Configuration Verification passed.

CIFS/Performance

Common Internet File System (CIFS) is a protocol that defines a standard for remote file access. With CIFS, users with different platforms and computers can share files without having to install new software. CIFS runs over TCP/IP but uses the SMB (Server Message Block) protocol found in Microsoft Windows for file and printer access; therefore, CIFS will allow all applications, not just Web browsers, to open and share files across the Internet. With CIFS, changes made to a file are simultaneously saved on both the client and server side. Clients in a WAAS network use the CIFS cache service to request file and print services from servers over a network. The tests performed verified the functionality and baseline performance for CIFS with, and without, the WAAS software.

The following tests were performed:

- CIFS Windows Acceleration Verification WAE512
- CIFS Cache Hit Benchmark
- CIFS Cache Miss Benchmark
- CIFS Native WAN Benchmark

CIFS Windows Acceleration Verification WAE512

CIFS enables collaboration on the Internet by defining a remote file-access protocol that is compatible with the way applications already share data on local disks and network file servers. CIFS incorporates the same high-performance, multi-user read and write operations, locking, and file-sharing semantics that are the backbone of today's sophisticated enterprise computer networks. CIFS runs over TCP/IP and utilizes the Internet's global Domain Naming Service (DNS) for scalability, and is optimized to support slower speed dial-up connections common on the Internet.

WAAS has an imbedded flow protection mechanism to ensure that existing CIFS sessions will not be broken when the device is brought online or additional WAE devices join the WCCP service groups.

CIFS sessions that were not established through while the WAEs were fully online and accelerating will not be CIFS accelerated and the redirected CIFS traffic will be returned to the router for native processing (DRE/TFO/LZ may be applied assuming the CIFS-non-WAFs policy is configured accordingly). To ensure CIFS sessions are fully accelerated, the CIFS session needs to be established AFTER the WAEs are online, optimizing, and configured to accelerate CIFS. If the connection was established before the WAE came online, this connection will not be accelerated, but will be a pass through connection ("In Progress").

This test verified that CIFS acceleration was working for a specific Windows client located at a remote branch connected to the data center by a Wide Area Network (WAN). Redundant WAE512's were connected to the branch LAN and it was verified that the WAE device accessed the file server in the data center as expected.

Test Procedure

The procedure used to perform the CIFS Windows Acceleration Verification WAE512 test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify that the bypass list on the Edge WAE (near the user) contains the IP of the configured server; (2 records for each server) by issuing the **show bypass list** command.

Step 3 Use Microsoft Management Console to inspect the Name or IP of the computer that opened the CIFS session. If you see the IP address of the Core WAE, it means that the session is being accelerated by WAAS CIFS acceleration.

If the IP address of the Windows client appears under the Computer section, then it means that the session is connected directly without acceleration . The session would need to be re-established so acceleration can be applied.

Step 4 Inspect the CIFS statistics on the WAE device GUI. Statistics should be incrementing when accessing files or folders (number of open files/ sessions / remote/local request should increment).

To check the statistics open browser to the edge WAE(s) and navigate as follows: WAFS Edge -> Monitoring -> CIFS

Step 5 Stop background scripts to collect final status of network devices and analyze for error.

Step 6 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that the bypass list on the Edge WAE will have entries for the server on ports 139 and 445.
- We expect that on the Domain Controller sessions to the Core WAE will be established.
- We expect that the statistics on the Edge WAE will show an increase in CIFS accesses once files have been transferred.
- We expect that on the Domain Controller no Sessions to the client will be established.

Results

CIFS Windows Acceleration Verification WAE512 passed.

CIFS Cache Hit Benchmark

The Cisco WAFS Benchmark Tool copies the workload files to the file server, and begins the test. You see the progress in the progress box. Each files is opened, modified, saved, and closed, and times are recorded for each operation.

Cache Hit Benchmark, also known as warm cache, tests how quickly a file server can be subsequently accessed through the WAFS cache (with data already in the cache).

The Cache Miss Benchmark test populates the WAE cache with all the files that are run in this test. The WAFS Benchmark tool is then run and the performance results for a Cache hit are verified.

Test Procedure

The procedure used to perform the CIFS Cache Hit Benchmark test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify that WCCPv2 is running on wae-3845-branch1 and dca-core-1 with the **show ip wccp** command.

The output of the command should show a value of at least 1 for the "Number of Cache Engines:" field for service 61 and 62. If you have redundant WAEs this number will reflect the number of WAEs that are configured.

Step 3 Since existing CIFS connections are not broken verify on the server that no CIFS session to the client exist. The only sessions open should be to the WAE devices.

Close any session to the client by navigating as follows:

Start -> All Programs -> Administrative Tools -> Computer Management -> System Tools -> Shared Folders -> Sessions

Step 4 Clear the cache on the Edge WAE(s).

Navigate to the WAE GUI, stop the Edge service, clear the WAFS cache, and then restart the Edge service.

Step 5 On the Client, launch the Cisco WAFS Benchmark Tool.

Step 6 Check the Prepare benchmark checkbox.

Step 7 Set the Copy workload files to drive value to W: and click GO!.

This is the target directory where the files are copied so that tests can be run on them.

Step 8 Uncheck the Prepare benchmark and check the Run benchmark checkbox's.

Step 9 Set the Use workload files from drive value to W:

This is the drive used for the testing.

Step 10 Set the Delay before file save (seconds): value to 60.

Step 11 Click Go!

Step 12 Click Save results to file to save the results, or click Open results folder to view the results.

View the results and verify that the time taken to open, save, and close each file has improved over the Native WAN results by a factor of at least 2x.

Step 13 Stop background scripts to collect final status of network devices and analyze for error.

Step 14 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect that the WAE devices will provide optimization.
- We expect that files will open and save considerably faster than they do across a WAN connection with no optimization.

Results

CIFS Cache Hit Benchmark passed.

CIFS Cache Miss Benchmark

The Cisco WAFS Benchmark Tool copies the workload files to the file server, and begins the test. You see the progress in the progress box. Each files is opened, modified, saved, and closed, and times are recorded for each operation.

Cache Miss, Also known as cold cache, this benchmark tests how quickly a file server can be accessed through the WAAS cache for the first time (with no data in the cache).

In this test we run the WAFS Benchmark tool while WCCPv2 redirection to the WAE device in the CORE and EDGE is enabled. The WAE Edge cache is first cleared and then the Benchmark tool is started. This provides performance results for files that are not cached.

Test Procedure

The procedure used to perform the CIFS Cache Miss Benchmark test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Verify that WCCPv2 is running on wae-3845-branch1 and dca-core-1 with the **show ip wccp** command.

The output of the command should show a value of at least 1 for the "Number of Cache Engines:" field for service 61 and 62. If you have redundant WAEs this number will reflect the number of WAEs that are configured.

Step 3 Since existing CIFS connections are not broken verify on the server that no CIFS session to the client exist. The only sessions open should be to the WAE devices.

Close any session to the client by navigating as follows:

Start -> All Programs -> Administrative Tools -> Computer Management -> System Tools -> Shared Folders -> Sessions

Step 4 Clear the cache on the Edge WAE(s).

Navigate to the WAE GUI, stop the Edge service, clear the WAFS cache, and then restart the Edge service.

Step 5 On the Client, launch the Cisco WAFS Benchmark Tool.

Step 6 Check the Prepare benchmark checkbox.

Step 7 Set the Copy workload files to drive value to W: and click GO!.

This is the target directory where the files are copied so that tests can be run on them.

Step 8 Uncheck the Prepare benchmark and check the Run benchmark checkbox's.

Step 9 Set the Use workload files from drive value to W:

This is the drive used for the testing.

Step 10 Set the Delay before file save (seconds): value to 60.

Step 11 Click Go!

Step 12 Click Save results to file to save the results, or click Open results folder to view the results.

View the results and verify that the time taken to open, save, and close each file has improved over the Native WAN results by a factor of at least 2x.

Step 13 Stop background scripts to collect final status of network devices and analyze for error.

Step 14 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the WAE devices to provide optimization.
- We expect files to open and save considerably faster than they do across a WAN connection with no optimization.

Results

CIFS Cache Miss Benchmark passed.

CIFS Native WAN Benchmark

The Cisco WAFS Benchmark Tool copies the workload files to the file server, and begins the test. You see the progress in the progress box. Each files is opened, modified, saved, and closed, and times are recorded for each operation.

This benchmark tests how quickly a file server can be accessed directly over the WAN. For this test WCCP was unconfigured on the Core and Branch routers so that no acceleration took place. The results express the performance what would be seen normally over a WAN connection with 80ms round trip delay.

Test Procedure

The procedure used to perform the CIFS Cache Hit Benchmark test follows:

Step 1 Begin background scripts to collect initial status of test network devices. Monitor memory and CPU utilization of those network devices.

Step 2 Unconfigure WCCPv2 on wae-3845-branch1 and dca-core-1 with the **no ip wccp 61** and **no ip wccp 62** commands.

This will cause all WAN traffic to not hit the WAE devices.

Step 3 Launch the benchmark tool.

Step 4 Check the Prepare benchmark checkbox.

Step 5 Set the Copy workload files to drive value to W:\.

This is the target directory on the file server where the files are copied so that tests can be run on them.

Step 6 Check the Run benchmark checkbox.

Step 7 Set the Use workload files from drive value to W:

This is the drive used for the testing.

Step 8 Set the Delay before file save (seconds): value to 60.

Step 9 Click Go!

Step 10 Click Save results to file to save the results, or click Open results folder to view the results.

View the results.

Step 11 Click Exit to close the Cisco WAFS Benchmark Tool.

Step 12 Reconfigure WCCPv2 on wae-3845-branch1 and dca-core-1 with the **ip wccp 61** and **ip wccp 62** commands.

Step 13 Stop background scripts to collect final status of network devices and analyze for error.

Step 14 Verify that memory and the CPU did not suffer severe, sustained, or unacceptable impact.

Expected Results

- We expect the files opened to take considerably longer than when WAFS/CIFS acceleration is configured.

Results

CIFS Native WAN Benchmark passed.

SAN Considerations

The following emphasizes each storage array vendor differences, including a description of the multipath and replication mechanisms and basic host and storage array configuration information.

- EMC, page A-1
- Network Appliance, page A-6
- Network Appliance FAS6070 Device Information, page A-8
- Hewlett Packard, page A-11
- HP XP10000 Device Information, page A-13

EMC

This appendix has general and detailed information about the EMC DMX3 frames used in the DCAP SAN topology. Following a brief general summary of results in Table A-1, Table A-2 shows software and firmware information, Table A-3 shows hardware information, and at the end is representative host device information showing redundant host paths and replicated devices.

No issues were found in host connectivity or replication. The load balancing policy for port channels was SID/DID/OXID. Table A-1 summarizes the iometer and iorate results from representative base connectivity and synchronous and asynchronous replication tests.

Note These results are only valid to compare results with various MDS optimizations applied; they are not a reflection on the performance capabilities or quality of the storage arrays themselves.

Table A-1 EMC DMX3 Iometer/Iorate Results (Average Per Device

Traffic Type	I/O type	Host	I/O / sec	MB / sec
Base connectivity	8 KB sequential writes	Linux Windows	2594 3273	20.2 25.6
Synch replication		Linux Windows	293 291	2.3 2.3
Synchs replication with FC write acceleration		Linux Windows	451 462	3.5 3.6
Asynch replication		Linux Windows	2310 2620	18.0 20.5
Asynch replication with FCIP write acceleration		Linux Windows	2272 3023	17.8 23.6
Asynch replication with FCIP compression		Linux Windows	2310 2843	18.0 22.2
Asynch replication with FCIP encryption		Linux Windows	2281 2828	17.8 22.1
Asynch replication with FCIP write acceleration, compression, & encryption		Linux Windows	2313 2826	18.1 22.1

Table A-2 EMC DMX3 Software/Firmware Information

Software Component	Function	Location	Version
Symmetrix CLI (SYMCLI)	configuration & monitoring, replication control	Linux host	V6.3.1.1 (Edit Level: 782)
Symmetrix CLI (SYMCLI)	configuration & monitoring, replication control	Windows host	V6.3.0.0 (Edit Level: 771)
Enginuity Microcode	operating system	frame	5771 (168B0000), Patch date 10.26.2006, Patch level 92
PowerPath	multipath	Linux host	4.5.1 (build 22)
PowerPath	multipath	Windows host	4.6.1 (build 5)

Table A-3 EMC DMX3 Hardware Information

Hardware Component	Quantity	Comments
Frame	2	Serials 000190300320, 000190300321
Cache	16384 MB	per frame
Disk	60 @ 146 GB	per frame
Disk director (DA)	2 @ 2 ports	per frame
Fiber director (FA)	2 @ 2 ports	per frame
Replication director (RA)	2 @ 1 port	per frame

EMC DMX3 Host Device Information

The following device configuration information is available.

- Linux host dcap-xchng-a1:, page A-3
- Windows host dcap-xchng-a2:, page A-4

Linux host dcap-xchng-a1:

```
[root@dcap-xchng-a1 ~]# /bin/df -k
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/emcpowerf1       69591480   5301280  60755100   9% /Local_IO_1
/dev/emcpowere1       69591480   1102876  64953504   2% /Local_IO_2
/dev/emcpowerd1       69591480   1102876  64953504   2% /Sync_IO_1
/dev/emcpowerc1       69591480   1102876  64953504   2% /Sync_IO_2
/dev/emcpowera1       69591480   2185280  63871100   4% /Async_IO_2
/dev/emcpowerb1       69591480   2185280  63871100   4% /Async_IO_1

[root@dcap-xchng-a1 ~]# /sbin/powermt display dev=all
Pseudo name=emcpowerf
Symmetrix ID=000190300320
Logical device ID=0033
state=alive; policy=SymmOpt; priority=0; queued-IOs=0
==============================================================================
---------------- Host ---------------   - Stor -   -- I/O Path -  -- Stats ---
### HW Path                I/O Paths    Interf. Mode    State  Q-IOs Errors
==============================================================================
   0 qla2xxx                   sda       FA  1cA   active  alive      0      0
   1 qla2xxx                   sdg       FA 16cB   active  alive      0      0

Pseudo name=emcpowere
Symmetrix ID=000190300320
Logical device ID=0037
state=alive; policy=SymmOpt; priority=0; queued-IOs=0
==============================================================================
---------------- Host ---------------   - Stor -   -- I/O Path -  -- Stats ---
### HW Path                I/O Paths    Interf. Mode    State  Q-IOs Errors
==============================================================================
   0 qla2xxx                   sdb       FA  1cA   active  alive      0      0
   1 qla2xxx                   sdh       FA 16cB   active  alive      0      0

Pseudo name=emcpowerd
Symmetrix ID=000190300320
Logical device ID=003B
state=alive; policy=SymmOpt; priority=0; queued-IOs=0
==============================================================================
---------------- Host ---------------   - Stor -   -- I/O Path -  -- Stats ---
### HW Path                I/O Paths    Interf. Mode    State  Q-IOs Errors
==============================================================================
   0 qla2xxx                   sdc       FA  1cA   active  alive      0      0
   1 qla2xxx                   sdi       FA 16cB   active  alive      0      0

Pseudo name=emcpowerc
Symmetrix ID=000190300320
Logical device ID=003F
state=alive; policy=SymmOpt; priority=0; queued-IOs=0
==============================================================================
---------------- Host ---------------   - Stor -   -- I/O Path -  -- Stats ---
### HW Path                I/O Paths    Interf. Mode    State  Q-IOs Errors
==============================================================================
   0 qla2xxx                   sdd       FA  1cA   active  alive      0      0
   1 qla2xxx                   sdj       FA 16cB   active  alive      0      0
```

```
Pseudo name=emcpowerb
Symmetrix ID=000190300320
Logical device ID=0043
state=alive; policy=SymmOpt; priority=0; queued-IOs=0
==============================================================================
--------------- Host ---------------   - Stor -   -- I/O Path -  -- Stats ---
### HW Path                I/O Paths    Interf. Mode    State  Q-IOs Errors
==============================================================================
   0 qla2xxx                   sde       FA  1cA   active  alive      0      0
   1 qla2xxx                   sdk       FA 16cB   active  alive      0      0

Pseudo name=emcpowera
Symmetrix ID=000190300320
Logical device ID=0047
state=alive; policy=SymmOpt; priority=0; queued-IOs=0
==============================================================================
--------------- Host ---------------   - Stor -   -- I/O Path -  -- Stats ---
### HW Path                I/O Paths    Interf. Mode    State  Q-IOs Errors
==============================================================================
   0 qla2xxx                   sdf       FA  1cA   active  alive      0      0
   1 qla2xxx                   sdl       FA 16cB   active  alive      0      0

[DCAP-SAN-MAIN-S] C:\Program Files\EMC\SYMCLI\bin>symrdf -g AB_A_GRP query


Device Group (DG) Name             : AB_A_GRP
DG's Type                          : RDF1
DG's Symmetrix ID                  : 000190300320


       Source (R1) View                Target (R2) View     MODES
--------------------------------    ------------------------ ----- ------------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks   Tracks  S Dev   E  Tracks   Tracks MDA   STATE
-------------------------------- -- ------------------------ ----- ------------

DEV001  0043 RW       0        0 RW 006B WD       0        0 A.. Consistent
DEV002  0047 RW       0        0 RW 006F WD       0        0 A.. Consistent
DEV003  005B RW       0        0 RW 007B WD       0        0 A.. Consistent
DEV004  005F RW       0        0 RW 007F WD       0        0 A.. Consistent

Total          -------- --------           -------- --------
  Track(s)            0        0                  0        0
  MB(s)             0.0      0.0                0.0      0.0

Legend for MODES:

 M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)           : X = Enabled, . = Disabled
 A(daptive Copy)    : D = Disk Mode, W = WP Mode, . = ACp off
```

Windows host dcap-xchng-a2:

```
E:   Local_IO_1
F:   Local_IO_2
G:   Sync_IO_1
H:   Sync_IO_2
I:   Async_IO_1
J:   Async_IO_2

[DCAP-XCHNG-A2] C:\Program Files\EMC\PowerPath>powermt display dev=all
```

```
Pseudo name=harddisk2
Symmetrix ID=000190300320
Logical device ID=004B
state=alive; policy=SymmOpt; priority=0; queued-IOs=0
==============================================================================
---------------- Host ---------------   - Stor -   -- I/O Path -  -- Stats ---
### HW Path                I/O Paths    Interf. Mode    State  Q-IOs Errors
==============================================================================
   2 port2\path0\tgt0\lun7     c2t0d7    FA 16cA   active  alive      0      0
   3 port3\path0\tgt0\lun7     c3t0d7    FA  1cB   active  alive      0      0

Pseudo name=harddisk3
Symmetrix ID=000190300320
Logical device ID=004F
state=alive; policy=SymmOpt; priority=0; queued-IOs=0
==============================================================================
---------------- Host ---------------   - Stor -   -- I/O Path -  -- Stats ---
### HW Path                I/O Paths    Interf. Mode    State  Q-IOs
Errors==========================================================================
   2 port2\path0\tgt0\lun8     c2t0d8    FA 16cA   active  alive      0      0
   3 port3\path0\tgt0\lun8     c3t0d8    FA  1cB   active  alive      0      0

Pseudo name=harddisk4
Symmetrix ID=000190300320
Logical device ID=0053
state=alive; policy=SymmOpt; priority=0; queued-IOs=0
==============================================================================
---------------- Host ---------------   - Stor -   -- I/O Path -  -- Stats ---
### HW Path                I/O Paths    Interf. Mode    State  Q-IOs Errors
==============================================================================
   2 port2\path0\tgt0\lun9     c2t0d9    FA 16cA   active  alive      0      0
   3 port3\path0\tgt0\lun9     c3t0d9    FA  1cB   active  alive      0      0

Pseudo name=harddisk5
Symmetrix ID=000190300320
Logical device ID=0057
state=alive; policy=SymmOpt; priority=0; queued-IOs=0
==============================================================================
---------------- Host ---------------   - Stor -   -- I/O Path -  -- Stats ---
### HW Path                I/O Paths    Interf. Mode    State  Q-IOs Errors
==============================================================================
   2 port2\path0\tgt0\lun10    c2t0d10   FA 16cA   active  alive      0      0
   3 port3\path0\tgt0\lun10    c3t0d10   FA  1cB   active  alive      0      0

Pseudo name=harddisk6
Symmetrix ID=000190300320
Logical device ID=005B
state=alive; policy=SymmOpt; priority=0; queued-IOs=0
==============================================================================
---------------- Host ---------------   - Stor -   -- I/O Path -  -- Stats ---
### HW Path                I/O Paths    Interf. Mode    State  Q-IOs Errors
==============================================================================
   2 port2\path0\tgt0\lun11    c2t0d11   FA 16cA   active  alive      0      0
   3 port3\path0\tgt0\lun11    c3t0d11   FA  1cB   active  alive      0      0

Pseudo name=harddisk7
Symmetrix ID=000190300320
Logical device ID=005F
state=alive; policy=SymmOpt; priority=0; queued-IOs=0
==============================================================================
---------------- Host ---------------   - Stor -   -- I/O Path -  -- Stats ---
### HW Path                I/O Paths    Interf. Mode    State  Q-IOs Errors
==============================================================================
   2 port2\path0\tgt0\lun12    c2t0d12   FA 16cA   active  alive      0      0
```

```
   3 port3\path0\tgt0\lun12    c3t0d12    FA  1cB   active  alive      0      0

[DCAP-SAN-MAIN-S] C:\Program Files\EMC\SYMCLI\bin>symrdf -g AB_S_GRP query


Device Group (DG) Name             : AB_S_GRP
DG's Type                          : RDF1
DG's Symmetrix ID                  : 000190300320


       Source (R1) View                Target (R2) View     MODES
--------------------------------    ------------------------ ----- ------------
             ST                  LI      ST
Standard      A                   N       A
Logical       T  R1 Inv   R2 Inv  K       T  R1 Inv   R2 Inv       RDF Pair
Device  Dev   E  Tracks    Tracks  S Dev   E  Tracks    Tracks MDA   STATE
--------------------------------- -- ------------------------ ----- ------------

DEV001  003B RW       0         0 RW 0063 WD       0         0 S.. Synchronized
DEV002  003F RW       0         0 RW 0067 WD       0         0 S.. Synchronized
DEV003  0053 RW       0         0 RW 0073 WD       0         0 S.. Synchronized
DEV004  0057 RW       0         0 RW 0077 WD       0         0 S.. Synchronized

Total          -------- --------           -------- --------
  Track(s)            0        0                  0        0
  MB(s)             0.0      0.0                0.0      0.0

Legend for MODES:

 M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
 D(omino)           : X = Enabled, . = Disabled
 A(daptive Copy)    : D = Disk Mode, W = WP Mode, . = ACp off
```

Network Appliance

This appendix has general and detailed information about the Network Appliance FAS6070 frames used in the DCAP SAN topology. Following a brief general summary of results in Table A-4, Table A-5 has software and firmware information, Table A-6 has hardware information, and at the end is representative host device information showing redundant host paths and replicated devices

No issues were found in host connectivity, but some minor issues were found with replication. Synchronous throughput was rather low (almost 10 times slower than base connectivity), and asynchronous replication had to be throttled using the following configuration in the snapmirror.conf file, or else link flaps occurred in the FCVI (replication) HBA ports and occasionally even frame crashes occurred:

```
fcip:async1 dcap-netapp-A1:async1 kbs=10000 * * * *
```

Note The "kbs=10000" limits the bandwidth utilization to 10,000 KB/sec. Testing revealed that a single replication stream could go up to about 60,000 KB/sec before link flaps occurred. With multiple replication streams the limit was about 15,000 KB/sec per stream. Network Appliance is aware of this issue and research as to root cause and permanent fix are still ongoing.

The load balancing policy for port channels was SID/DID. Table A-4 summarizes the iometer and iorate results from representative base connectivity and synchronous and asynchronous replication tests.

Note These results are only valid to compare results with various MDS optimizations applied; they are not a reflection on the performance capabilities or quality of the storage arrays themselves.

Note Some tests were not performed in Phase 2 due to lack of hardware.

Table A-4 Network Appliance FAS6070 Iometer/Iorate Results (Average Per Device)

Traffic Type	I/O type	Host	I/O / sec	MB / sec
Base connectivity	8 KB sequential writes	Linux Windows	1105 2020	8.6 15.8
Synch replication		Linux Windows	283	2.2
Synchs replication with FC write acceleration		Linux Windows	289	2.3
Asynch replication		Linux Windows	1919 2248	15.0 17.6
Asynch replication with FCIP write acceleration		Linux Windows	1958 2511	15.3 19.6
Asynch replication with FCIP compression		Linux Windows	1892 2455	14.8 19.2
Asynch replication with FCIP encryption		Linux Windows	1915 2442	15.0 19.1
Asynch replication with FCIP write acceleration, compression, & encryption		Linux Windows	1891 2436	14.8 19.0

Table A-5 Network Appliance FAS6070 Software/Firmware Information

Software Component	Function	Location	Version
ONTAP	operating system	frame	7.2
MPIO (device-mapper-mulitpath)	multipath	Linux host	v0.4.5 (16/06, 2005)
Veritas DMP	multipath	Windows host	4.3

Table A-6 Network Appliance FAS6070 Hardware Information

Hardware Component	Quantity	Comments
Frame	4	Serials 073252 and 073251 in DCA, 073250 & 073249 in DCB.
Memory	32768 MB	per frame
Disk	28 @ 133 GB	2 shelves @ 14 disks per frame
FC HBAs	2 @ 2 ports	per frame, for base connectivity
FCVI HBAs	2 @ 2 ports	per frame, for replication

Network Appliance FAS6070 Device Information

Linux host dcap-san-hst-04

```
[root@dcap-san-hst-04 ~]# /bin/df -k
Filesystem           1K-blocks      Used Available Use% Mounted on
/view/dev/mapper/vg_netapp_local_04_01-h04_netapp_local_0
                      50574012   4284276  43720728   9% /E
/dev/mapper/vg_netapp_local_04_01-h04_netapp_local_1
                      50574012   4284276  43720728   9% /F
/dev/mapper/vg_netapp_async_04_01-h04_netapp_async_4
                      51602044   4284276  44696536   9% /J
/dev/mapper/vg_netapp_async_04_01-h04_netapp_async_5
                      51602044   4284276  44696536   9% /K
/dev/mapper/vg_netapp_sync_04_01-h04_netapp_sync_2
                      51602044   4284276  44696536   9% /G
/dev/mapper/vg_netapp_sync_04_01-h04_netapp_sync_3
                      51602044   4284276  44696536   9% /H

[root@dcap-san-hst-04 ~]# /usr/sbin/vgs @dcap-san-hst-04 -o +devices
  VG                    #PV #LV #SN Attr   VSize  VFree Devices
  vg_netapp_async_04_01   2   2   0 wz--n- 99.99G    0
/dev/mapper/360a98000486e5366535a395850456b38(0)
  vg_netapp_async_04_01   2   2   0 wz--n- 99.99G    0
/dev/mapper/360a98000486e5366535a395850467432(0)
  vg_netapp_local_04_01   2   2   0 wz--n- 99.99G 1.99G
/dev/mapper/360a98000486e5365574a387949547930(0)
  vg_netapp_local_04_01   2   2   0 wz--n- 99.99G 1.99G
/dev/mapper/360a98000486e5365574a387949557749(0)
  vg_netapp_local_04_01   2   2   0 wz--n- 99.99G 1.99G /dev/mapper/360a98000486
e5365574a387949557749(0)
  vg_netapp_sync_04_01    2   2   0 wz--n- 99.99G    0  /dev/mapper/360a98000486
e5366535a395a70327941(0)
  vg_netapp_sync_04_01    2   2   0 wz--n- 99.99G    0  /dev/mapper/360a98000486
e5366535a395a70336b48(0)

[root@dcap-san-hst-04 ~]# /sbin/multipath -l
360a98000486e5365574a394948453574[size=50 GB][features="1
queue_if_no_path"][hwhandler="0"]
\_ round-robin 0 [active]
  \_ 5:0:0:6     sdaj 66:48  [active]
  \_ 4:0:1:6     sdad 65:208 [active]
  \_ 5:0:1:6     sdl  8:176  [active]
  \_ 4:0:0:6     sdx  65:112 [active]

360a98000486e5365574a387949557749
[size=50 GB][features="1 queue_if_no_path"][hwhandler="0"]
\_ round-robin 0 [active]
 \_ 5:0:0:1     sdag 66:0   [active]
 \_ 4:0:1:1     sdaa 65:160 [active]
 \_ 5:0:1:1     sdi  8:128  [active]
 \_ 4:0:0:1     sdu  65:64  [active]

360a98000486e5365574a387949547930
[size=50 GB][features="1 queue_if_no_path"][hwhandler="0"]
\_ round-robin 0 [active]
 \_ 5:0:0:0     sdaf 65:240 [active]
 \_ 5:0:1:0     sdh  8:112  [active]
 \_ 4:0:0:0     sdt  65:48  [active]
 \_ 4:0:1:0     sdz  65:144 [active]
```

```
360a98000486e5366535a395850456b38
[size=50 GB][features="1 queue_if_no_path"][hwhandler="0"]
\_ round-robin 0 [active]
 \_ 5:0:0:4     sdah 66:16  [active]
 \_ 4:0:1:4     sdab 65:176 [active]
 \_ 4:0:0:4     sdv  65:80  [active]

360a98000486e5366535a395a70336b48
[size=50 GB][features="1 queue_if_no_path"][hwhandler="0"]
\_ round-robin 0 [active]
 \_ 5:0:1:3     sdas 66:192 [active]
 \_ 4:0:0:3     sdam 66:96  [active]
 \_ 4:0:1:3     sdao 66:128 [active]
 \_ 5:0:0:3     sdaq 66:160 [active]

360a98000486e5366535a395850467432
[size=50 GB][features="1 queue_if_no_path"][hwhandler="0"]
\_ round-robin 0 [active]
 \_ 5:0:0:5     sdai 66:32  [active]
 \_ 4:0:1:5     sdac 65:192 [active]
 \_ 5:0:1:5     sdk  8:160  [active]
 \_ 4:0:0:5     sdw  65:96  [active]

360a98000486e5366535a395a70327941
[size=50 GB][features="1 queue_if_no_path"][hwhandler="0"]
\_ round-robin 0 [active]
 \_ 5:0:0:2     sdap 66:144 [active]
 \_ 4:0:0:2     sdal 66:80  [active]
 \_ 4:0:1:2     sdan 66:112 [active]
 \_ 5:0:1:2     sdar 66:176 [active]

dcap-netapp-A1> snapmirror status
Snapmirror is on.
Source                  Destination            State          Lag        Status
dcap-netapp-A1:async2  dcap-netapp-B1:async2  Source         00:00:55   Idle
dcap-netapp-A1:sync2   dcap-netapp-B1:sync2   Source         -          In-sync
```

Windows host dcap-san-hst-03

```
E: h03-l-net-0 (harddisk1)
F: h03-l-net-1 (harddisk2)
G: h03-s-net-2 (harddisk3)
H: h03-s-net-3 (harddisk4)

[DCAP-SAN-HST-03] C:\Documents and Settings\plowden>vxdmpadm pathinfo Harddisk1
================================================================================
Device information
    Name        :   Harddisk1
    Media Name  :
    No. of Paths:   2
    LoadBalancePolicy       :   Round Robin (Active/Active)
Path information ...
Path 6-0-0:
    State       :   Healthy
    Primary     :   NO
    SCSI-3 Persistent Reservation: YES
    SCSI-3 Reserved: NO
    Port        :   6
    Channel     :   0
    Target      :   0
    LUN         :   0
Path 7-0-0:
```

```
    State       :   Healthy
    Primary     :   NO
    SCSI-3 Persistent Reservation: YES
    SCSI-3 Reserved: NO
    Port        :   7
    Channel     :   0
    Target      :   0
    LUN         :   0

[DCAP-SAN-HST-03] C:\Documents and Settings\plowden>vxdmpadm pathinfo Harddisk2
================================================================================
Device information
    Name        :   Harddisk2
    Media Name  :
    No. of Paths:   2
    LoadBalancePolicy       :   Round Robin (Active/Active)
Path information ...
Path 6-0-0:
    State       :   Healthy
    Primary     :   NO
    SCSI-3 Persistent Reservation: YES
    SCSI-3 Reserved: NO
    Port        :   6
    Channel     :   0
    Target      :   0
    LUN         :   1
Path 7-0-0:
    State       :   Healthy
    Primary     :   NO
    SCSI-3 Persistent Reservation: YES
    SCSI-3 Reserved: NO
    Port        :   7
    Channel     :   0
    Target      :   0
    LUN         :   1

[DCAP-SAN-HST-03] C:\Documents and Settings\plowden>vxdmpadm pathinfo Harddisk3
================================================================================
Device information
    Name        :   Harddisk3
    Media Name  :
    No. of Paths:   2
    LoadBalancePolicy       :   Round Robin (Active/Active)
Path information ...
Path 6-0-0:
    State       :   Healthy
    Primary     :   NO
    SCSI-3 Persistent Reservation: YES
    SCSI-3 Reserved: NO
    Port        :   6
    Channel     :   0
    Target      :   0
    LUN         :   2
Path 7-0-0:
    State       :   Healthy
    Primary     :   NO
    SCSI-3 Persistent Reservation: YES
    SCSI-3 Reserved: NO
    Port        :   7
    Channel     :   0
    Target      :   0
    LUN         :   2

[DCAP-SAN-HST-03] C:\Documents and Settings\plowden>vxdmpadm pathinfo Harddisk4
```

```
================================================================================
Device information
    Name         :   Harddisk4
    Media Name   :
    No. of Paths:   2
    LoadBalancePolicy       :   Round Robin (Active/Active)
Path information ...
Path 6-0-0:
    State        :   Healthy
    Primary      :   NO
    SCSI-3 Persistent Reservation: YES
    SCSI-3 Reserved: NO
    Port         :   6
    Channel      :   0
    Target       :   0
    LUN          :   3
Path 7-0-0:
    State        :   Healthy
    Primary      :   NO
    SCSI-3 Persistent Reservation: YES
    SCSI-3 Reserved: NO
    Port         :   7
    Channel      :   0
    Target       :   0
    LUN          :   3

dcap-netapp-A1> snapmirror status
Snapmirror is on.
Source                  Destination             State          Lag        Status
dcap-netapp-A1:sync1    dcap-netapp-B1:sync1    Source         -          In-sync
```

Hewlett Packard

This appendix has general and detailed information about the Hewlett Packard XP10000 frames used in the DCAP SAN topology. Following a brief general summary of results in Table A-7, Table A-8 has software and firmware information, Table A-9 has hardware information, and at the end is representative host device information showing redundant host paths and replicated devices.

During the test, one of the HP frames had hardware errors that possibly affected test results. For example, base connectivity I/O rates were not nearly as high as what could be expected. Otherwise no issues were found in host connectivity or replication. The load balancing policy for port channels was SID/DID. Table A-7 summarizes the iometer and iorate results from representative base connectivity and synchronous and asynchronous replication tests.

Note These results are only valid to compare results with various MDS optimizations applied; they are not a reflection on the performance capabilities or quality of the storage arrays them.

Note Some tests were not performed in Phase 2 due to lack of hardware.

Table A-7 P XP10000 Iometer/Iorate Results (average per device)

Traffic Type	I/O type	Host	I/O / sec	MB / sec
Base connectivity	8 KB sequential writes	Linux Windows	379 2914	3.0 22.8
Synch replication		Linux Windows	283	2.2
Synchs replication with FC write acceleration		Linux Windows	428	3.3
Asynch replication		Linux Windows	293	2.3
Asynch replication with FCIP write acceleration		Linux Windows	276	2.2
Asynch replication with FCIP compression		Linux Windows	785	6.1
Asynch replication with FCIP encryption		Linux Windows	595	4.6
Asynch replication with FCIP write acceleration, compression, & encryption		Linux Windows	914	7.1

Table A-8 HP XP10000 Software/Firmware Information

Software Component	Function	Location	Version
Microcode	operating system	frame	50-08-05
Command View XP AE Device Manager	configuration & monitoring, replication control	Windows host, frame service processor	5.1.0-00
service processor	operating system	frame	50-08-05/00
RMI Server	remote management	frame	04_08_00
MPIO (device-mapper-mulitpath)	multipath	Linux host	v0.4.5 (16/06, 2005)
Veritas DMP	multipath	Windows host	4.3

Table A-9 HP XP10000 Hardware Information

Hardware Component	Quantity	Comments
Frame	2	Serials 82836, 42042
Cache	20 GB	per frame
Disk	60 @ 146 GB	per frame
Fiber host ports	4 ports	per frame
Fiber replication ports	4 ports	per frame
Fiber unused ports	8 ports	per frame

HP XP10000 Device Information

Linux host dcap-san-hst-02

re quick

what did you Windows host dcap-san-hst-03:

```
R: h03-1-hp-0 (harddisk 7)
S: h03-1-hp-1 (harddisk 8)
T: h03-s-hp-2 (harddisk 9)
U: h03-s-hp-3 (harddisk 10)

[DCAP-SAN-HST-03] C:\Documents and Settings\plowden>vxdmpadm pathinfo Harddisk7
======================================================================
Device information
    Name        :   Harddisk7
    Media Name  :
    No. of Paths:   2
    LoadBalancePolicy       :   Round Robin (Active/Active)
Path information ...
Path 6-0-1:
    State       :   Healthy
    Primary     :   NO
    SCSI-3 Persistent Reservation: YES
    SCSI-3 Reserved: NO
    Port        :   6
    Channel     :   0
    Target      :   1
    LUN         :   0
Path 7-0-1:
    State       :   Healthy
    Primary     :   NO
    SCSI-3 Persistent Reservation: YES
    SCSI-3 Reserved: NO
    Port        :   7
    Channel     :   0
    Target      :   1
    LUN         :   0

[DCAP-SAN-HST-03] C:\Documents and Settings\plowden>vxdmpadm pathinfo Harddisk8
======================================================================
Device information
    Name        :   Harddisk8
    Media Name  :
    No. of Paths:   2
    LoadBalancePolicy       :   Round Robin (Active/Active)
Path information ...
Path 6-0-1:
    State       :   Healthy
    Primary     :   NO
    SCSI-3 Persistent Reservation: YES
    SCSI-3 Reserved: NO
    Port        :   6
    Channel     :   0
    Target      :   1
    LUN         :   1
Path 7-0-1:
    State       :   Healthy
    Primary     :   NO
    SCSI-3 Persistent Reservation: YES
    SCSI-3 Reserved: NO
    Port        :   7
```

```
    Channel     :   0
    Target      :   1
    LUN         :   1

[DCAP-SAN-HST-03] C:\Documents and Settings\plowden>vxdmpadm pathinfo Harddisk9
================================================================================
Device information
    Name        :   Harddisk9
    Media Name  :
    No. of Paths:   2
    LoadBalancePolicy       :   Round Robin (Active/Active)
Path information ...
Path 6-0-1:
    State       :   Healthy
    Primary     :   NO
    SCSI-3 Persistent Reservation: YES
    SCSI-3 Reserved: NO
    Port        :   6
    Channel     :   0
    Target      :   1
    LUN         :   2
Path 7-0-1:
    State       :   Healthy
    Primary     :   NO
    SCSI-3 Persistent Reservation: YES
    SCSI-3 Reserved: NO
    Port        :   7
    Channel     :   0
    Target      :   1
    LUN         :   2

[DCAP-SAN-HST-03] C:\Documents and Settings\plowden>vxdmpadm pathinfo Harddisk10
================================================================================
Device information
    Name        :   Harddisk10
    Media Name  :
    No. of Paths:   2
    LoadBalancePolicy       :   Round Robin (Active/Active)
Path information ...
Path 6-0-1:
    State       :   Healthy
    Primary     :   NO
    SCSI-3 Persistent Reservation: YES
    SCSI-3 Reserved: NO
    Port        :   6
    Channel     :   0
    Target      :   1
    LUN         :   3
Path 7-0-1:
    State       :   Healthy
    Primary     :   NO
    SCSI-3 Persistent Reservation: YES
    SCSI-3 Reserved: NO
    Port        :   7
    Channel     :   0
    Target      :   1
    LUN         :   3
```

Cisco GSS Deployment

The Global Site Selector (GSS) is a database of IP address's, domain names, and resource records. The only sensitive data that resides on the GSS is the username and password used to access and make configuration changes. Content security is not a concern on the GSS. However, as an administrator, one should be concerned with a few entities in regards to the DNS/Security aspect. A DNS solution should not be vulnerable to a DDos attack or allow unauthorized users access to the GSS.

Note The GSS does not run a distribution of BIND so it does not suffer from the many vulnerabilities that such open source code implementations are susceptible to.

A good understanding of the GSS's role in augmenting DNS and the requirements for accomplishing this task helps position the GSS properly in the network. Having an in-depth understanding of DNS and specifically how the GSS supports given sub-domains allows for effective positioning into existing network architectures.

WAAS Implementation

A single WAE-7326 running Cisco Wide Area Application Service (WAAS) version 4.0.3 was deployed as a core Wide-Area Application Engine (WAE) and connected to data center core Supervisor 720 device running IOS version 12.2(18)SXF6. The core catalyst device was then connected to a 3845 branch ISR running IOS version 12.4(12). A NISTnet server was placed in the between the data center and branch to simulate a WAN connection with 80ms round-trip delay, T1 bandwidth and 0.5% packet loss. At the branch site a single WAE-512 running WAAS v4.0.3 was deployed as an edge WAE device. A single WAE-612, connected in the data center, running WAAS v4.0.3 was used as the central manager. The central manager GUI was used as the main method of configuration for the WAAS deployment.

The Web Cache Communication Protocol version 2 (WCCPv2) was used as the method of traffic interception and redirection on the data center and branch routers.

A server running Windows Server 2003 SP4 was configured with Active directory and connected to the access layer of the data center. The server was placed on a VLAN that was not handled by the CSM, FWSM, and SSLSM. At the branch site two client PCs were connected and configured with Windows XP SP2.

The following sections are discussed:

- Initial Configuration of the WAAS Central Manager, page C-2
- Initial Core WAE Configuration in the Data Center, page C-2
- Initial Edge WAE Configuration at Remote Branch, page C-3
- WAN Connection, page C-3
- Basic Server/Client Configuration Overview, page C-4
- WAAS Network Configuration Via the Central Manager, page C-4
- Core cluster Settings, page C-5
- Configure WAE Devices for Domain Name System (DNS), page C-6
- Configure WAE Devices for Windows Name Services (WINS), page C-6
- Configure NTP on the Central Manager, page C-6
- Configure NTP on Core and Edge WAE Devices, page C-7
- Defining the Core WAE, page C-7
- Defining the Edge WAE, page C-7
- Configure WAE Authentication Methods, page C-8
- Configure a File Server, page C-8
- Create a New Connection, page C-9

- WCCPv2 Overview, page C-9
- WCCPv2 Implementation, page C-9
- Testing Concept, page C-10

Initial Configuration of the WAAS Central Manager

A WAE-612 device was connected into the data center to one of the Core devices.

Note For Phase III this connection will be moved to the access layer.

Once powered up and connected via console, the first-time setup utility was used to enter network settings to give the WAE basic access to the network. (Note: you can re-enter the setup mode by simply typing setup on the CLI if necessary) With the device pingable and the primary-interface specified, the device mode was then configured as central-manager via the CLI. The configuration was saved and then a reload was initiated for the new configuration to take effect.

```
dca-wae-612-cm#configure terminal
dca-wae-612-cm(config)#
dca-wae-612-cm(config)#primary-interface gigabitEthernet 2/0
dca-wae-612-cm(config)#device mode central-manager
dca-wae-612-cm(config)#exit
dca-wae-612-cm#copy run start
dca-wae-612-cm#reload
Proceed with reload?[confirm] y
Shutting down all services, will Reload requested by CLI@ttyS0.
Restarting system.
```

Note To ensure the configuration is saved, log into the device(s) configuration was pushed out to and issue the “copy run start” command.

Once rebooted the device was then specified as the primary WAAS central manager and the management database was created and initialized.

```
dca-wae-612-cm#configure terminal
dca-wae-612-cm(config)#central-manager role primary
dca-wae-612-cm(config)#cms enable
dca-wae-612-cm(config)#exit
dca-wae-612-cm#copy run start
```

Initial Core WAE Configuration in the Data Center

A single WAE-7326 was directly connected to a Catalyst 6500 with a Supervisor 720 running IOS version 12.2(18)SXF6 in the core of the data center. Similar to the central manager, the first-time setup utility was used to configure basic network settings to provide the core device IP connectivity to the data center network. The interface on the WAE connected to the Catalyst device was configured as the primary interface and the device mode was set to application-accelerator mode. The IP address of the central manager was also specified and the core WAE device was then registered to it.

```
dca-wae-7326-1#configure terminal
dca-wae-7326-1(config)#primary-interface gigabitEthernet 2/0
dca-wae-7326-1(config)#device mode application-accelerator
dca-wae-7326-1(config)#central manager address 10.0.13.6
```

```
dca-wae-7326-1(config)#cms enable
Registering WAAS Application Engine...
Sending device registration request to Central Manager with address 10.0.13.6
Please wait, initializing CMS tables
copy^H^HSuccessfully initialized CMS tables
Registration complete.
Please preserve running configuration using 'copy running-config startup-config'.
Otherwise management service will not be started on reload and node will be shown
'offline' in WAAS Central Manager UI.
management services enabled
dca-wae-7326-1(config)#end
dca-wae-7326-1#copy run start
dca-wae-7326-1#
```

Note IP connectivity to the central-manager must first be established before registering.

Initial Edge WAE Configuration at Remote Branch

A single WAE-512 was connected to a 3845 Integrated Services Router (ISR) running 12.4(12) at a remote Branch. The edge WAE device was brought online and the first-time setup utility was used to configure basic network settings to provide the edge device IP connectivity to the branch LAN. Once complete the interface on the WAE connecting to the ISR was configured as the primary interface and the device mode was set to application-accelerator mode. The IP address of the central manager was also specified and the edge WAE device was registered with it.

```
wae-branch1-512-1#configure terminal
wae-branch1-512-1(config)#primary-interface gigabitEthernet 2/0
wae-branch1-512-1(config)#device mode application-accelerator
wae-branch1-512-1(config)#central manager address 10.0.13.6
wae-branch1-512-1(config)#cms enable
Registering WAAS Application Engine...
Sending device registration request to Central Manager with address 10.0.13.6
Please wait, initializing CMS tables
Successfully initialized CMS tables
Registration complete.
Please preserve running configuration using 'copy running-config startup-config'.
Otherwise management service will not be started on reload and node will be shown
'offline' in WAAS Central Manager UI.
management services enabled
wae-branch1-512-1(config)# end
wae-branch1-512-1#copy run start
```

Note The device must be put on a different subnet than that of the clients at the site and IP connectivity to the central manager must first be established before registering.

WAN Connection

For Phase II, a single port on the data center core device, dca-core-1, was connected to a server running NISTnet WAN simulator software. The WAN simulator was then connected to a 3845 branch router, dca-wae-branch1. The software introduced a WAN network connection with 80ms round-trip delay, T1 bandwidth, and 0.5% traffic loss.

Basic Server/Client Configuration Overview

A server was installed with Windows Server 2003 SP4. Active directory, DNS, WINS and IIS were configured on the server and the domain used was waas.domain.com. Two client PC's were used at the branch location, each running Windows XP with the latest service pack.

Note Cisco WAAS is not able to apply the full breadth of CIFS acceleration when a feature known as "digital signatures" are enabled on the file server (default for Windows 2000 SP4 and Windows 2003 SP1 and above). If digital signatures are enabled, Cisco WAAS can still provide WAN optimization to CIFS flows, including DRE, LZ, and TFO. This, however, will not provide the same levels of performance that can be found when Cisco WAAS is also providing CIFS acceleration.

To disable digital signatures, login to a domain controller and open the "Default Domain Controller Security Settings" MMC snap-in, generally found under Start > Programs > Administrative Tools. From the left pane, expand to Security Settings > Local Policies > Security Options, and locate a policy called:

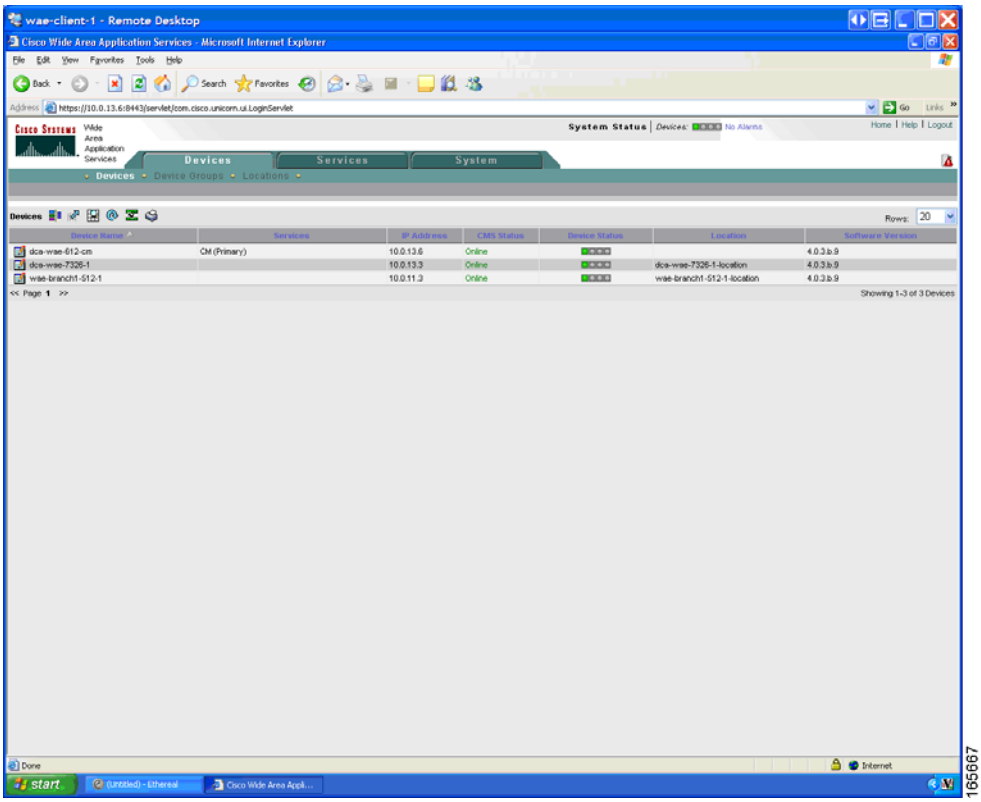
- Domain member: Digitally encrypt or sign secure channel data (always) - double-click this item, ensure that "Define this policy setting" is checked, and that the radio button next to "disabled" is selected.
- Microsoft network server: Digitally sign communications (always) - double-click this item, ensure that "Define this policy setting" is checked, and that the radio button next to "disabled" is selected.
- Microsoft network server: Digitally sign communications (if client agrees) - double-click this item, ensure that "Define this policy setting" is checked, and that the radio button next to "disabled" is selected.

Reboot server, or execute the **gpupdate /force** command from a DOS window to refresh policy settings.

WAAS Network Configuration Via the Central Manager

The central manager GUI (Figure C-1Central Management GUI) was then reachable by opening a web browser (IE is the only browser supported). It was verified in this central manager GUI that the registration actions were in fact successful. The CMS status for each device was then verified to be "online."

Figure C-1 Central Management GUI



Both the CLI and GUI can be used for configuring the WAAS network. The GUI was used in this deployment and the following instructions use the GUI as the main method for configuration.

Core cluster Settings

A core cluster was first defined which consisted of the single file-server, running Windows Server 2003, in the data center used for centralized file storage. The cluster was defined by navigating/configuring through the GUI as follows:

```
Devices -> Device Groups -> Create New Device Group
    Type: WAFS Core Cluster
    Name: Core Cluster
```

Once defined the Core Server settings were specified by navigating/configuring as follows:

```
Devices -> Device Groups -> Core Server Settings
    File server access username/password
```

Note To ensure the configuration is saved, log into the device(s) configuration was pushed out to and issue the "copy run start" command.

Configure WAE Devices for Domain Name System (DNS)

The DNS device used for testing was a Windows Server 2003 server, Server-2k3, located in the data center. Once entries for each WAE devices were specified in DNS table, all WAE devices were configured to use the Server-2k3 DNS database for domain name lookups. This was done by navigating/configuring through the GUI as follows:

```
Devices -> Device Groups -> AllDevicesGroup -> General Settings -> Network -> DNS
Local Domain Name: waas.domain.com
List of DNS Servers: 101.1.33.5(server-2k3.waas.domain.com)
Submit and force setting on all devices in group
```

Note To ensure the configuration is saved, log into the device(s) configuration was pushed out to and issue the "copy run start" command.

Configure WAE Devices for Windows Name Services (WINS)

WINS can be used instead of DNS in some circumstance like when no DNS is available. WINS, as well as DNS, was configured on Server-2k3 in the data center. The WINS settings were configured to WAE devices by navigating/configuring through the GUI as follows:

```
Devices -> Device Groups -> AllDevicesGroup -> General Settings -> Windows Name Services
Workgroup or Domain Name: WAAS
WINS Server: 101.1.33.5
```

Configure (or verify) NETBIOS names for the WAE devices by navigating/configuring through the GUI as follows:

```
Devices -> Devices -> WAE Device -> Activation
    NetBIOS Name: <name>
```

Note To ensure the configuration is saved, log into the device(s) configuration was pushed out to and issue the "copy run start" command.

Configure NTP on the Central Manager

The lab NTP server used is running on goel.cisco.com (172.18.177.132). This server was used to sync the central manager with the rest of the data-center devices. The configuration was done via the GUI by navigating/configuring as follows:

```
    Devices -> Central Manager (primary) -> Show Advanced -> General Settings ->
Miscellaneous -> Date/Time -> NTP/Time Zone
    Time Zone: EST
    NTP: Check Enable Box
NTP Server: 172.18.177.132
```

Note To ensure the configuration is saved, log into the device(s) configuration was pushed out to and issue the "copy run start" command.

Configure NTP on Core and Edge WAE Devices

The Core and Edge devices were then synchronized to the central manager by navigating/configuring through the GUI as follows:

```
    Devices -> Device Groups -> AllDevicesGroup -> General Settings -> Miscellaneous ->
Date/Time -> NTP/Time Zone
    Time Zone: EST
    NTP: Check Enable Box
NTP Server: 10.0.13.6(CM address)
```

Note **T**o ensure the configuration is saved, log into the device(s) configuration was pushed out to and issue the "copy run start" command.

Defining the Core WAE

The core WAE-7326 device was defined via the GUI as a **Core** device by navigating as follows:

```
    Devices -> Devices -> Select device to define as Core WAE
    File Services -> Core Configuration
        Check enable "Core Server"
        Select predefined Core Cluster
```

Note To ensure the configuration is saved, log into the device(s) configuration was pushed out to and issue the "copy run start" command.

The device was reloaded via the GUI as instructed and when it came back online it was verified that the Core device was online by navigating through the GUI as follows:

```
    Devices -> Device Groups -> Core Cluster -> Members
Devices -> Devices (Device is now "Core" and CMS status should be "online")
```

Defining the Edge WAE

The edge WAE-512 device was defined via the GUI as an **Edge** device by navigating as follows:

```
    Devices -> Devices -> Select device to define as Edge WAE
File Services -> Edge Configuration
    Enable Edge Server and leave all boxes checked.
```

Note To ensure the configuration is saved, log into the device(s) configuration was pushed out to and issue the "copy run start" command.

The device was reloaded via the GUI as instructed and when it came back online it was verified that the Edge device was online by navigating through the GUI as follows:

```
Devices -> Devices (Device is now "Edge" and CMS status should be "online")
```

Configure WAE Authentication Methods

Each WAE must be authenticated with the Windows File Server before CIFS optimizations will occur. The WAE devices were configured for Windows authentication by navigating/configuring through the GUI as follows:

```
Devices -> Device Groups -> AllDevicesGroup -> General Settings -> Authentication ->
Authentication Methods
Check "Authentication Login Methods" and set WINDOWS to the primary and "Local" to the
secondary
Check "Authorization Methods" and set WINDOWS to the primary and "Local" to the secondary
```

The specific login information for registrations with the file server was then configured on each WAE. Kerberos authentication was the method used was configured by navigating through the GUI as follows:

```
Devices -> Device Groups -> AllDevicesGroup -> General Settings -> Authentication ->
Authentication Methods
Check Kerberos enabled
Realm: WAAS.DOMAIN.COM
Key Distribution Center: 101.1.33.5:88 (port 88 is WINS port)
Domain Controller: SERVER-2K3
```

After submitting the Realm, Key Distribution Center, and Domain Controller settings the fields for username/password became available and the proper credentials were entered and then submitted.

```
    Domain administrator username: WAAS\Administrator
Domain administrator password: password
Confirm password: password
Click the "Show authentication Status" button to see the registration status.
```

Note The commands actually executed on each WAE to verify the registration are these SMB command:

```
windows-domain diagnostics wbinfo -t
windows-domain diagnostics wbinfo --sequence
windows-domain diagnostics wbinfo --domain-info=WAAS
Time skew is verified
```

You can send the register from the CLI w/ this command:

```
windows-domain diagnostics net "ads join -U Username%password -W domain(short name)"
```

Configure a File Server

The Windows server was set up as a file server by navigating/configuring through the GUI as follows:

```
Services -> File -> File Servers -> Definition
File Server Name: server-2k3.waas.domain.com
Submit
Services -> File -> File Servers
Assign Core Clusters
  Click the blue X then Submit
```

Verify green check is assigned in the *A* column and that the IP is resolved in the comments column.

Create a New Connection

A connection was then established between core and edge WAE devices by navigating/configuring through the GUI as follows:

```
Services -> File -> Connectivity -> Definition
Name: WAE
Assign Core Cluster and submit
Services -> File -> Connectivity -> Definition -> Assign Edge Devices
    Click the blue X then Submit Edge WAE
```

WCCPv2 Overview

Cisco WAAS utilizes a WCCPv2 service group called “TCP Promiscuous”. The TCP promiscuous service group is actually a combination of two service groups: 61 and 62.

While both service groups look for the same traffic, only one service group should be in the path for each direction of traffic flow. If service group 61 is configured in such a way that it intercepts traffic going from the client to the server, service group 62 should be configured to intercept traffic returning to the client from the server. Given that the service groups distribute load based on the source IP or destination IP of the packets, this ensures that the packets will be routed to the same WAE regardless of the direction of traffic flow.

Service group 61 redirects TCP packets to a WAE device, distributes load based on the source IP address of the flow.

Service group 62 redirects TCP packets to a WAE device, distributes load based on the destination IP address of the flow.

When the TCP promiscuous mode service is used, the CIFS caching service is not required. (WCCP Version 2 service 89).

WCCPv2 Implementation

The data center core router and branch ISR were each configured to use WCCPv2 as the method for traffic interception and redirection. On the data center core Supervisor 720 router, all interfaces from the aggregation switches were configured to redirect ingress traffic using WCCPv2 service group 62. For DCAP Phase II a single Gigabit Ethernet port was simply used as a WAN interface. The WAN port was configured to redirect ingress traffic using WCCPv2 service group 61. The branch ISR was configured with service group 61 on the ingress LAN port and service group 61 on its ingress WAN port. The service group deployment scenario used is the most commonly deployed scenario.

Note The service group deployment at the branch and data center are opposite, however, they are functionally equivalent.

Testing Concept

DCAP Phase II WAAS testing focused on the basic network deployment, functionality and configuration for the WAE devices. The WAFS Benchmark tool was used to verify that both optimization and file services were successful. DCAP Phase III will go beyond basic functionality testing and provide testing centered on performance and high availability scenarios.

The Voodoo Solution

The DCAP test topology is highly scaled, in terms of the number of servers that are present across all serverfarms. In DC-A, the CSM is configured with 2000+ real servers across 30+ serverfarms. Obviously, it is not realistic or cost-effective to deploy 2000 physical servers in a test environment like DCAP.

Emulating 2000 Servers in DCAP

One of the goals of the DCAP project is to stress the Catalyst 6500 and Catalyst 4900 products in their roles as access switches by fully populating one of each chassis with Fast Ethernet-connected servers. For a modular chassis such as the Catalyst 6509, this meant 6 48-port linecards, or 288 connected servers. This does not scale, either.

What is Voodoo?

The solution in DCAP 2.0 testing used ten physical servers to emulate the 2000 that were configured in the LAN topology of DC-A. These servers, combined with a Catalyst 6513 chassis that is separate from the DC-A LAN infrastructure, provided the magic, or "Voodoo" that made this solution happen.

Why the Need for Voodoo?

So the problem was how to scale the number of real servers that the load-balancer could send traffic to as well as scale the functional connectivity of the access switches. On top of this, the solution must be transparent to the rest of the network. In other words, the infrastructure devices themselves, such as the access or aggregation switches must not have any special configuration to make it work.

What are the Necessary Components?

In reality, the solution could have been provided with less than the ten physical servers that were used. Ten were used so that traffic to them could be scaled to some degree (a single physical server emulating 2000 real servers and handling file requests for each of them would quickly be overwhelmed.) The ten servers that were used matched the following specs:

- Dual AMD Opteron processors @ 2600 MHz, w/1 GB onboard cache
- 4 GB DDR RAM

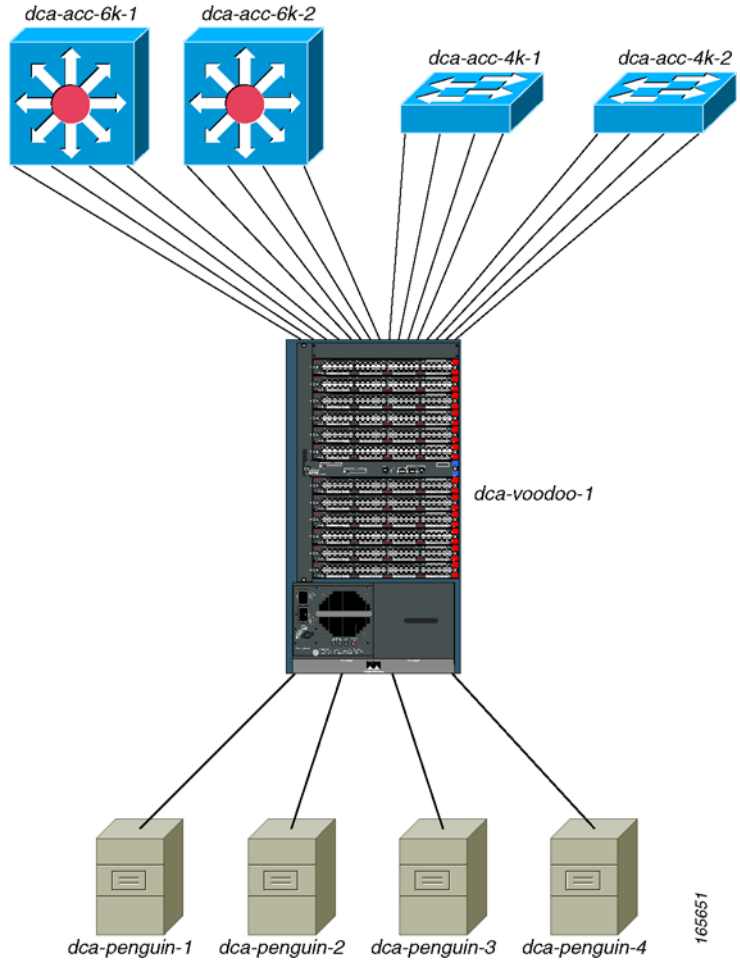
- Red Hat Enterprise Linux

The Catalyst 6513 switch used in the solution SAT between the access switches and the ten servers. It uses a Supervisor 720 as the switching engine and is fully populated with a variety of 48-port line cards (WS-X6148A-GE-TX, WS-X6348-RJ-45 and WS-X6548-RJ-45). The 12 line cards provided a total of 576 ports of Fast Ethernet or Gigabit Ethernet density. In addition to this Catalyst 6513, a second Catalyst 6500 was used to provide connectivity for some of the ten servers (this will be discussed in more detail below).

The Catalyst 6513 that is used to supply connections to the access switches is deployed in the manner described below.

In Figure D-1, the dca-voodoo-1 is fully populated with 48-port linecards, giving it 576 Fast Ethernet and Gigabit Ethernet ports. Four of these ports are used to connect to the four Linux servers via 802.1q trunks. That leaves 572 ports to connect to the Access Layer switches. This is divided nicely among each of the four Linux servers so that each Linux server emulates 143 servers. The Access Layer switch dca-acc-6k-1, a Catalyst 6509 with six WS-X6748-GE-TX linecards, has each of its 288 ports connected into dca-voodoo-1. The top-of-rack Access Layer switch dca-acc-4k-1, a Catalyst 4948, has 47 of its ports connected to dca-voodoo-1 (one port is reserved for management). The remainder of the connections available from dca-voodoo-1 is distributed proportionally between dca-acc-6k-2 and dca-acc-4k-2.

Figure D-1 Catalyst 6513 Used in Voodoo Solution

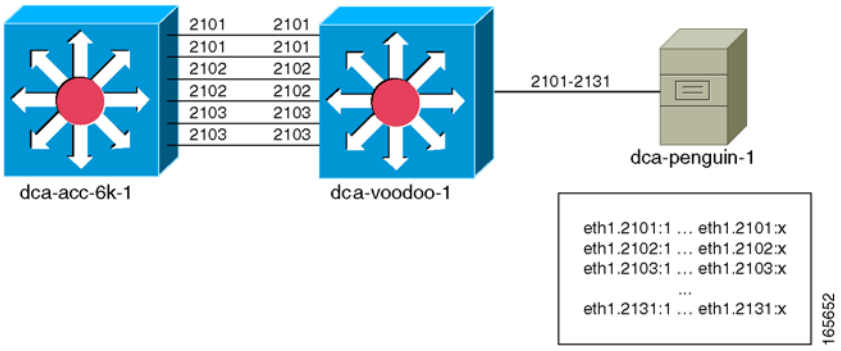


What Features are Used to Make Voodoo Work?

In the DCAP topology, there are only 31 VLANs configured in the Layer 2 domain. These are VLANs 2101-2131. The challenge of using a single physical server to emulate 143 individual hosts cannot be solved by using the same VLAN subinterfaces on the physical servers. In other words, simply configuring eth1.2101 through eth1.2131 on each of the Linux servers would not work for several reasons. First, using only 802.1q subinterfaces would only allow for a maximum of 31 emulated hosts per Linux box. Second, even if virtual interfaces were configured per 802.1q subinterface (eth1.2101:1, eth1.2101:2 and so on) to allow for more than one host per VLAN, nothing is gained towards the goal of having traffic pass across each of the physical links between the Voodoo device and the access switches.

Figure D-2 illustrates this problem. In this diagram, the access switch dca-acc-6k-1 has multiple links on the same VLAN, which reflects the real-world scenario of multiple server hosts being on the same VLAN. The Linux server is configured with 802.1q subinterfaces which, in turn, have multiple virtual interfaces. While traffic could be exchanged between a client and any of the servers emulated by the virtual interfaces on dca-penguin-1, there is no telling what path that traffic will take. An HTTP GET request could pass out of the top link on the access switch on its way to the server, but there's no way to guarantee that the response from the server to the client will use the same path.

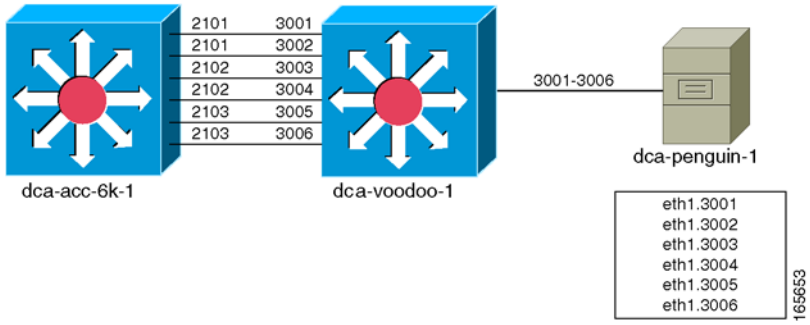
Figure D-2 Limitation of Voodoo with Virtual Interfaces



One other problem with this method is that it does not allow for unique MAC addresses to be configured on a per-virtual interface basis. In the Red Hat Enterprise Linux distribution, unique MAC addresses can be configured for each 802.1q subinterface, but that same MAC is shared with any virtual interfaces configured on that subinterface. Having the ability to configure unique MAC addresses for each of the emulated hosts helps in reflecting the real-world traffic flows.

The Voodoo solution solves the above problems by using a different set of VLANs on the Linux server than are in the DCAP topology Layer 2 domain. On the dca-voodoo-1 side, as illustrated in Figure D-3, each port connecting to an access switch belongs to a unique VLAN. On the Linux server, an 802.1q subinterface is configured for each of the VLANs on the Voodoo device. The important point here is that these Voodoo-only VLANs are only known to the Voodoo device and the Linux server; the actual topology switching infrastructure still only knows about the VLANs in its Layer 2 domain.

Figure D-3 Voodoo Solution Using a Dedicated 802.1q Subinterface for Each Emulated Server



The 802.1q subinterfaces on the Linux server may belong to similar subnets, depending on the DCAP topology VLAN the dca-voodoo-1 port maps to. For example, in Figure D-3, both VLANs 3001 and 3002 on dca-voodoo-1 map to VLAN 2101 on dca-acc-6k-1 and, therefore, are configured with IP addresses in the same subnet. The same holds true for VLANs 3003 and 3004 on the Voodoo device, which both map to VLAN 2102 on the access switch, and for VLANs 3005 and 3006, which map to VLAN 2103.

Thus, there is an allowance for unique MAC addresses to be assigned to each "individual host" emulated by the Linux server. The problem of non-deterministic return paths for emulated hosts belonging to the same subnet has also apparently been solved. Unfortunately, another roadblock is sprung, again stemming from the fact that multiple subinterfaces are sharing a common IP subnet on the Linux server.

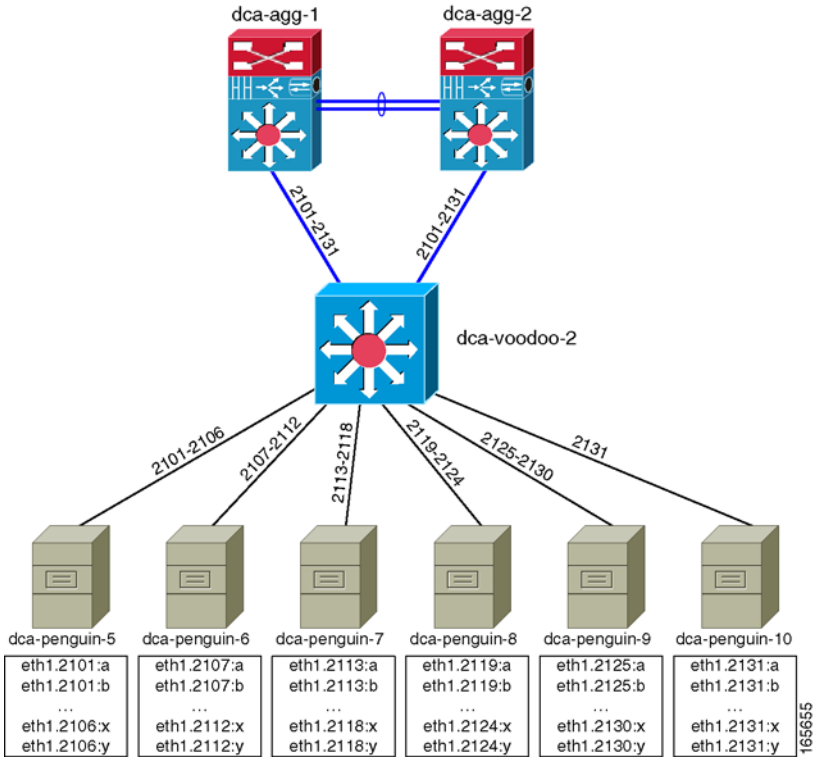
The new problem arises with the usage of the ARP protocol to resolve the MAC address of the default gateway for the emulated servers. It is a given that the 802.1q subinterfaces that share a similar IP subnet also share a common default gateway. So when the Linux box ARPs to resolve the gateway's IP address, dca-voodoo-1 does not know which port to send the ARP request out. The two ports belonging to VLAN 3001 and 3002 are both set up to carry traffic on the same IP subnet, so when dca-voodoo-1 receives that ARP request, it could choose either of the two ports. (In testing, a single port was chosen for each of the IP subnets.) When the access switch, dca-acc-6k-1, receives the ARP request, it populates it's MAC table with the MAC address of the Linux subinterface, mapping it to whatever port the ARP request was received on. When traffic flows between client and server, dca-acc-6k-1 sends all downstream traffic through a single port.

To get around this final obstacle, the source routing feature was employed on the Linux server. Using source routing, the Linux box now looks at the source IP address of the packet and sends it out the appropriate 802.1q subinterface. So even though the subinterfaces eth1.3001 and eth1.3002 share a common IP subnet, because the response is coming from one or the other, the proper path will be followed through the Voodoo device. Since the proper path is followed through the Voodoo device, the access switch's MAC table is populated appropriately. Finally, traffic can deterministically traverse each link in the access layer, making possible a close-to-real-world simulation of multiple server hosts in the datacenter using a single Linux server.

The Voodoo Solution in Full Scale

All that is left to do is increase the scale so that a Catalyst 6509 and Catalyst 4948 can be fully populated, and then some. Figure D-4 shows how that is done in the DCAP topology.

Figure D-4 The full Scale of the Voodoo Solution in the DCAP Test Topology



Again, a Catalyst 6513 fully populated with 48-port line cards is used as the Voodoo device, which yields 572 ports for physical links in the access layer. As will be noted below, each Linux server has a limitation on the number of source routes that can be configured (252), so at least 3 Linux servers were needed to fully utilize the capacity of the Catalyst 6513. The total number of Linux servers that were used to emulate these 572 hosts was taken to 4, both for the even divisibility and allocation of subinterfaces, and for the benefits in regards to scaling (4 servers can handle more load than 3).

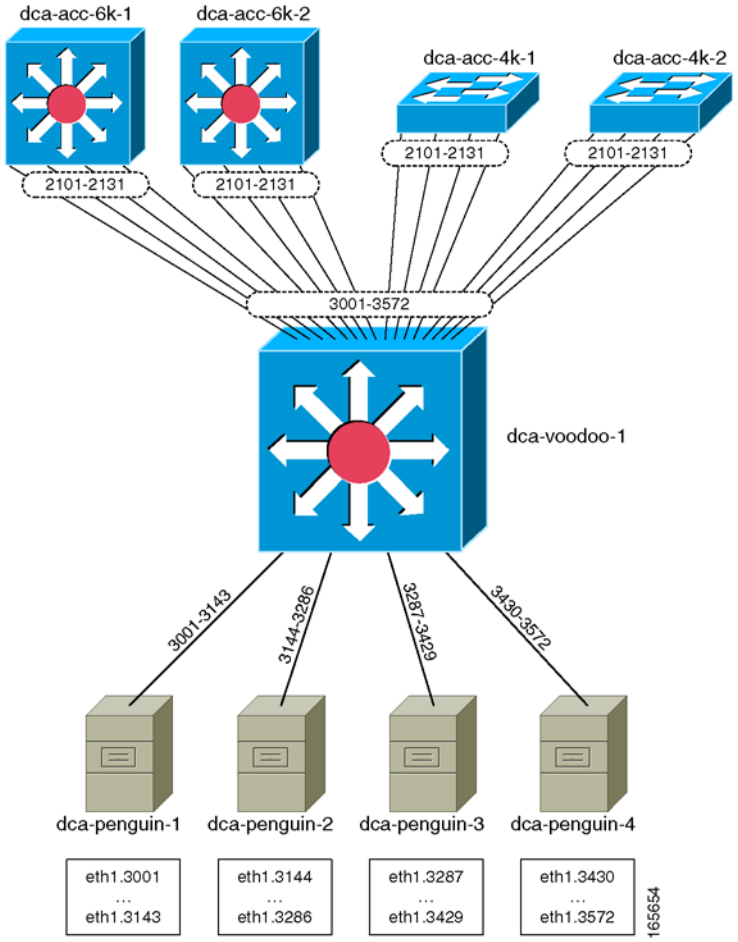
Each of the Linux servers was configured with 143 802.1q subinterfaces spanning all of the 31 IP subnets used in the DCAP test topology (VLANs 2101-2131). This allowed for each of the four access switches to carry traffic for all subnets as well.

Before details of the configuration of this solution are revealed, what about the other 1428 real servers that were going to be functional in the DCAP topology? Having one fully populated access switch from both the Catalyst 6500 and Catalyst 4900 families was enough, from a testing perspective. While it would have been feasible to scale the Voodoo solution to 2000 real servers, the effort would have been superfluous. One area that was still in need of improved coverage, though, was the scaling of the Aggregation Layer, with regards to 10-Gigabit Ethernet density from the Aggregation Layer to the Access Layer.

While, in DCAP 2.0 testing, it was not possible to fully populate an Aggregation Layer device with 10-GE links to the Access Layer (it will be done in DCAP 3.0), the scale was increased incrementally as a side effect of adding the remaining 1472 real servers to the topology.

Figure D-5 shows how the balance of 1472 real servers was emulated in the DCAP topology. It was through the use of six additional Linux boxes, all connected to the Aggregation Layer through a single Catalyst 6500.

Figure D-5 The 1472 Remaining Servers are Emulated Using Six Additional Linux Hosts



For these remaining servers, virtual interfaces on the Linux hosts were used. Also, unlike the actual Voodoo solution described earlier, each of the Linux hosts here were only configured with a subset of the possible IP subnets, using 802.1q subinterfaces that mapped directly to the VLANs in the Layer 2 domain. Since all of the emulated servers would communicate through a single Catalyst 6500 (dca-voodoo-2), and only one link into the Aggregation Layer would be used at any given time, it is not necessary to use a Voodoo-type setup to force the traffic paths. (The naming of this single Catalyst 6500 dca-voodoo-2 is coincidental; the actual Voodoo solution is not used here.)

Configuration Details

The ports connecting the Voodoo device dca-voodoo-1 to the Access Layer switches were configured as access ports, and assigned to the appropriate VLAN. For example:

```
!
interface GigabitEthernet1/3
 switchport
 switchport access vlan 3001
 switchport mode access
 no ip address
 no cdp enable
 spanning-tree bpdufilter enable
!
```

The ports connecting dca-voodoo-1 to the four Linux servers were configured as 802.1q trunks, carrying the appropriate VLANs for the respective Linux server.

```
!
interface GigabitEthernet1/1
 description Penguin-1 Eth1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 3001-3143,3600
 switchport mode trunk
 no ip address
 no cdp enable
!
interface GigabitEthernet1/2
 description Penguin-2 Eth1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 3144-3286
 switchport mode trunk
 no ip address
 no cdp enable
!
interface GigabitEthernet3/1
 description Penguin-3 Eth1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 3287-3429
 switchport mode trunk
 no ip address
 no cdp enable
!
interface GigabitEthernet3/2
 description Penguin-4 Eth1
 switchport
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 3430-3572
 switchport mode trunk
 no ip address
 no cdp enable
end
```

Other than that, there is nothing special about the dca-voodoo-1 configuration; it is a simple Layer 2 device.

There were several steps necessary for configuring the 802.1q subinterfaces on each of the Linux servers.

```
# Enable 802.1q trunking
dca-penguin-1$ modprobe 8021q
```

```
# Configure the 802.1q subinterfaces on device eth1
dca-penguin-1$ vconfig add eth1 3001
dca-penguin-1$ vconfig add eth1 3002
dca-penguin-1$ vconfig add eth1 3003
...
dca-penguin-1$ vconfig add eth1 3143
# Configure each of the 802.1q subinterfaces with IP and MAC addresses
ifconfig eth1.3001 hw ether 00:00:01:01:30:01 101.1.1.11/24
ifconfig eth1.3002 hw ether 00:00:01:01:30:02 101.1.1.12/24
ifconfig eth1.3003 hw ether 00:00:01:01:30:03 101.1.1.13/24
...
ifconfig eth1.3143 hw ether 00:00:01:01:31:43 101.1.31.14/24
```

Enabling the source routing was also a multi-step process.

The very first thing to do in order to use source routing is to delete the default route entries on the Linux server. Be sure to have an explicit route defined for the management access before doing this.

```
dca-penguin-1$ route del default
```

Each routing entry must first be defined with a name in the file /etc/iproute2/rt_tables. Each entry name is indexed with a line number. The only valid values for line numbers are 1-252.

```
dca-penguin-1$ more /etc/iproute2/rt_tables
#
# reserved values
#
#255    local
#254    main
#253    default
#0      unspec
#
# local
#
#1      inr.ruhep
101     VL3001
102     VL3002
103     VL3003
...
242     VL3142
243     VL3143
```

Next, an IP rule must be added indicating that packets sourced from a particular IP address must use a specific table to be routed.

```
dca-penguin-1$ ip rule add from 101.1.1.11 table VL3001
dca-penguin-1$ ip rule add from 101.1.1.12 table VL3002
dca-penguin-1$ ip rule add from 101.1.1.13 table VL3003
...
dca-penguin-1$ ip rule add from 101.1.31.14 table VL3143
```

The only thing that remains is to tell the Linux box to send any traffic using a certain table out a specified interface.

```
dca-penguin-1$ ip route add default via 101.1.1.1 dev eth1.3001 table VL3001
dca-penguin-1$ ip route add default via 101.1.1.1 dev eth1.3002 table VL3002
dca-penguin-1$ ip route add default via 101.1.1.1 dev eth1.3003 table VL3003
...
dca-penguin-1$ ip route add default via 101.1.1.1 dev eth1.3143 table VL3143
```

DCAP Defects

A detailed explanation of defects encountered or discovered in Cisco Data Center Assurance Program (DCAP) 2.0 testing is provided below. Refer to Cisco Data Center Assurance Program (DCAP) 2.0 for respective testing information.

The following DCAP DDTS's are detailed.

- CSCsh30404, page E-1
- CSCsg11506, page E-2
- CSCsg36646, page E-2
- CSCec74017, page E-3
- CSCsg22134, page E-3
- CSCsh25315, page E-4
- CSCek26222, page E-4
- CSCsf23499, page E-5
- CSCsg96362, page E-5
- CSCsc81109, page E-6
- CSCsf05420, page E-6
- CSCsd00731, page E-7
- CSCsh02565, page E-7
- CSCsg11203, page E-8
- CSCsg12760, page E-8

CSCsh30404

Sever

ISR: CPU hogs and crash after issuing show call.

Affected Systems

ISR's running 12.4(12) crypto images.

Summary

After issuing “show call malloc tcl contents” a series of CPU Hogs occur which ultimately cause the router to crash.

Impact

Customers normally don't run this command, but it could hamper effectively troubleshooting by TAC.

Workaround

Do not issue the command on crypto images.

Fixed in Future Release?

Yes, 12.4(12.15)T

CSCsg11506

Sever

EPM breaks connections if it doesn't intercept both traffic directions.

Affected Systems

All WAE devices running WAAS software.

Summary

EPM(End Point Mapper)-based applications are unavailable in asymmetric routing scenarios.

Impact

It is possible for EPM-based application traffic to not be optimized in circumstances where asymmetric routing occurs.

Workaround

Ensure that both WCCP service groups 61 and 62 are on the same interface, and that is the WAN interface that connects to another WAE-accelerated site. Alternatively, configure WCCP redirection "in" on all relevant interfaces.

Fixed in Future Release?

This bug will be fixed in the Alameda code.

CSCsg36646

Sever

Filesystem Corruption.

Affected Systems

All WAE devices running WAAS software.

Summary

When RAID is being rebuild (use "show disk details" to check it) and the system being rebooted during the sync process. The rebuild usually done after initial installation (rescue CD or manufacturing) or initiating a drive.

Impact

File system corruption will, in most cases, cause the need for the partitions to be rebuilt and the WAAS software to be reinstalled.

Workaround

The corrupted partition should be deleted and re-create after reload. Ensure that after partition initialization you wait till the whole RAID is synced (might take more than an hour) before any reload

CSCec74017

Severe

STE:URL rewrite does not handle reloca string in multiple TCP pkts

Affected Systems

Secure Socket Layer Module (SSLM) running 3.1(1).

Summary

An SSLM can be configured as a server proxy and used to do URL rewrites. If the HTTP header that specifies the relocation string spans more than one TCP segment, the SSLM will send the response to the client without doing the URL rewrite. This issue is seen with long URLs.

Impact

Customers who want to employ the URL rewrite feature of the SSLM will find that it does not work with longer URLs coupled with small MTU and/or MSS on the SSLM.

Workaround

There is no workaround.

Fixed in Future Release?

There is no indication of an intention to fix this defect.

CSCsg22134

Sever

FMS startup dialog repeatedly shown when ipaddress set in svr.hostname.

Affected Systems

MDS9500 director switches running SAN-OS 3.0(3).

Summary

When the server.hostname parameter is set to an IP address in the server.properties file, the Fabric Manager Server restart message repeatedly displays in the Fabric Manager Client open dialog box.

Impact

Fabric Manager will not start up.

Workaround

Use host name in server.properties.

Fixed in Future Release?

Yes, 3.0(3a), 3.1(1).

CSCsh25315

Moderate

Standby CSM crashed while issuing vserver inservice command.

Affected Systems

Content Switching Module (CSM) running 4.1(6).

Summary

When issuing the **inservice** command in vserver configuration mode on a CSM running 4.1(6), the CSM suffered a system crash.

Impact

Customers issuing the **inservice** command for a vserver on a CSM may experience a crash of the line card.

Workaround

There is no workaround.

Fixed in Future Release?

This issue has been irreproducible in further testing.

CSCek26222

Moderate

Moderate – mem leak at tm_dbg_msg_queue_init.

Affected Systems

A Supervisor 720 system running 12.2(18)SXF4 with WS-X67xx series linecards and DFC daughtercards.

Summary

In the DCAP testing, it was discovered that each time a WS-X6704-10GE linecard with DFC daughtercard is rebooted, 60 Kilobytes of memory is lost on the SP. The culprit process holding the memory is “SP Logger Proces”. This has not been tested on other WS-X67xx linecards.

Impact

Customers with the above-described system who experience a linecard reset can expect to lose 60 Kilobytes of SP memory each time the susceptible linecard is reset.

Workaround

There is no workaround.

Fixed in Future Release?

This will be fixed in 12.2(18)SXF9.

CSCsf23499

Moderate

VLAN int goes DOWN when etherchannel trunk comes up

Affected System

Supervisor 720 system running 12.2(18)SXF4 which has an etherchannel trunk interface.

Summary

The VLAN SVI interfaces on a Catalyst 6500 router are kept in the up/up state by the autostate feature. With autostate, the SVI will only be in up/up state so long as there is at least one port in the STP Forwarding state for that VLAN. When the device connected to the affected system is rebooted and comes back online, the etherchannel interface on the affected system goes down then comes back up. When it comes back up, there may be a period of time when (apparently) all VLANs in the L2 database flap and the VLAN SVIs go do down/down state. This lasts no more than a few seconds and does not appear to interfere with traffic. This errant behavior was observed in ~6 out of 10 cases.

Impact

A customer with the above described scenario may observe a momentary flapping of their Layer 3 VLAN interfaces when the neighbor device, and the local etherchannel interface, comes back online.

Workaround

While there is no workaround, this may be related to scale, as it was seen in a system with ~240 VLANs.

Fixed in Future Release?

There are no plans as of yet to address this issue in future releases.

CSCsg96362

Moderate

FM java.lang.Nullpointer when editing port-channel in topology map

Affected Systems

MDS9500 director switches running SAN-OS 3.0(3).

Summary

Fabric Manager sometimes return a "java.lang.Nullpointer" Java error when right-clicking on a port-channel link in the topology map and choosing "Edit" from the popup menu.

Impact

Creating, deleting, and changing a port-channel doesn't work.

Workaround

Restart the Fabric Manager server, or use CLI.

Fixed in Future Release?

Yes, 3.1(1).

CSCsc81109

Minor

show crypto pki server cli command causes Traceback

Affected Systems

Supervisor 2/MSFC 2 or Supervisor 720 systems running 12.2(18)SXF4.

Summary

This is a harmless traceback message which, when decoded, basically informs the user that an invalid command was executed. That invalid command (which should be valid) is show crypto pki server cli.

Impact

When a customer executes the show crypto pki server cli command, this traceback can be expected.

Workaround

There is no workaround.

Fixed in Future Release?

This defect is Closed. The Certificate Server feature is not supported in 12.2SX and this is a benign traceback, so it won’t be fixed.

CSCsf05420

Minor

show hw-module all boot cmd broken

Affected Systems

Supervisor 2/MSFC 2 or Supervisor 720 systems running 12.2(18)SXF4.

Summary

This is a simple defect in which the parser is broken for a certain command, in this case show hw-module all boot. Instead of returning the correct information, an invalid-command error is returned.

Impact

Customers will see this error with this command.

Workaround

While there is no way to execute this command, the information can still be gathered, module-by-module.

Fixed in Future Release?

There are no plans as of yet to address this in future releases.

CSCsd00731

Minor

show mod csm x cons displays ftp data port as 0.

Affected Systems

Content Switching Module (CSM) running 4.1(5) or later code.

Summary

The CSM connection table shows the incorrect data port for active and passive FTP flows. The FTP data sessions are dynamically installed by slowpath to match the corresponding FTP control session. Slowpath has no knowledge of the actual data port, since it never touches data traffic.

Impact

This does not impact traffic and is essentially a cosmetic issue.

Workaround

There is no workaround

Fixed in Future Release?

This defect is Closed as it's a documented design limitation.

CSCsh02565

Minor

Misleeding error message for slowstart feature.

Affected Systems

Content Switching Module (CSM) running 4.1(6) or later code.

Summary

When configuring the slowstart feature on the CSM, an error message is produced giving the impression that the feature is not supported: Slowstart feature is not supported by card, config accepted. In fact, the feature is supported and all that's needed is an environment variable.

Impact

This has no operational impact.

Workaround

Configure the appropriate environment variable for the slowstart feature.

Fixed in Future Release?

There are no plans to address this issue in future releases.

CSCsg11203

Minor

MDS9500 show hardware internal debug-info has invalid packet-flow syntax.

Affected Systems

MDS9500 director switches running SAN-OS 3.0(3).

Summary

The linecard diagnostic command, "show hardware internal debug-info interface X/Y," fails with a command parse error.

Impact

Customers normally don't run this command, but it could hamper effectively troubleshooting by TAC.

Workaround

There is no workaround.

Fixed in Future Release?

Yes, 3.1(3).

CSCsg12760

Minor

Update WCCP CLI with WAAS Terminology.

Affected Systems

All WAE devices running WAAS software.

Summary

The outuput of “show wccp file-engines” refers to the WAE hardware as a “file engine” rather than an “application accelerator.”

Impact

This has no operational impact.

Workaround

There is no workaround.

Fixed in Future Release?

This bug is not yet fixed in any release.

CSCsg12760

Safe Harbor Technology Releases

The DCAP testing effort often relies on testing performed by other teams, particularly the Safe Harbor team. The determination of which software to run in the various systems in the DCAP topology is made based on Safe Harbor software recommendations. Many of the tests executed in regular Safe Harbor testing are applicable to the DCAP topology and are leveraged for the final DCAP product. While those test results are considered in the final result, they are not reported in this document. Refer to the appropriate technolgy release for the latest technology release testing details.

Table F-1 lists the EDCS document numbers so that the reader can locate and review the results of relevant Safe Harbor testing. A comprehensive list of the test cases executed in these other projects is provided in the Appendix to this document.

Table F-1 Safe Harbor Technology Releases in EDCS

Platform	Software Version	EDCS Doc. No.
Supervisor 720	12.2(18)SXF6	566444
Firewall Services Module	2.3(3.2)	523606
Content Switching Module	4.1(6)	514342
Secure Socket Layer Module	3.1(1)	504160
Catalyst 4948-10GE	12.2(31)SGA	N/A
MDS9500	3.0(3)	N/A

The following summary tables list tests conducted in the latest Safe Harbor technology releases.

- Native (Classic) IOS 12.2(18)SXF6, page F-2
- Firewall Services Module (FWSM) 2.3.3.2, page F-14
  - Single-Routed Firewall Services Module (FWSM) 2.3.3.2, page F-14
  - Single-Transparent Firewall Services Module (FWSM) 2.3.3.2, page F-17
  - Multi-Routed Firewall Services Module (FWSM) 2.3.3.2, page F-19
  - Multi-Transparent Firewall Services Module (FWSM) 2.3.3.2, page F-21
- Content Switching Module (CSM) 4.1.6, page F-24
- Secure Socket Layer Module (SSLM) 3.1.1, page F-26

Native (Classic) IOS 12.2(18)SXF6

Table F-2 summarizes tests executed as part of the Cisco IOS Release 12.2(18)SXF6 initiative. Table F-2 includes the feature or function tested, the section that describes the feature set the feature or function belongs to, and the component tests for each feature or function.

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

Table F-2 Safe Harbor Cisco IOS Release 12.2(18)SXF6 Testing Summary

Test Suites	Features/Functions	Tests
Housekeeping, page 24	Baseline, page 24	1. Network Steady State 2. Network Validation
	System Upgrade, page 26	1. 12.2(18)SXB11 to 12.2(18)SXF6—Sup 22 2. 12.2(18)SXB11 to 12.2(18)SXF6—Sup 720 3. 12.2(18)SXE5 to 12.2(18)SXF6—Sup 720 4. 12.2(18)SXF5 to 12.2(18)SXF6—Sup 22 5. 12.2(18)SXF5 to 12.2(18)SXF6—Sup 720 6. Fast System 12.2(18)SXB11 to 12.2(18)SXF6—Sup 22 7. Fast System 12.2(18)SXB11 to 12.2(18)SXF6—Sup 720 8. Fast System 12.2(18)SXE5 to 12.2(18)SXF6—Sup 720 9. Fast System 12.2(18)SXF5 to 12.2(18)SXF6—Sup 22 10. Fast System 12.2(18)SXF5 to 12.2(18)SXF6—Sup 720
Memory Leak	Memory Leak, page 34	1. Remove and Restore—Sup22 2. Remove and Restore—Sup720 3. Repeated Telnet—Sup22 4. Repeated Telnet—Sup720 5. Repeated SSHv1—Sup22 6. Repeated SSHv1—Sup720 7. Repeated SSHv2—Sup22 8. Repeated SSHv2—Sup720 9. Multicast Flap Memory

Table F-2 Safe Harbor Cisco IOS Release 12.2(18)SXF6 Testing Summary (continued)

Test Suites	Features/Functions	Tests
Hardware Integrity	Hardware Redundancy, page 40	1. Power Supply Failure 2500W—Sup 22 2. Power Supply Failure 2500W—Sup 720 3. Power Supply Failure 6000W—Sup 22 4. Power Supply Failure 6000W—Sup 720 5. Supervisor Hot Insert—Sup 22 6. Supervisor Hot Insert—Sup 720 7. Change Switch Modes Basic Function—Sup 22 8. Change Switch Modes Basic Function—Sup 720 9. Change Switch Modes RPR+ Failover—Sup 22 10. Change Switch Modes RPR+ Failover—Sup 720 11. Change Switch Modes SSO Failover—Sup 22 12. Change Switch Modes SSO Failover—Sup 720 13. Failover RPR+ —Sup 22 14. Failover RPR+ —Sup 720 15. Failover SSO NSF—Sup 22 16. Failover SSO NSF—Sup 720 17. Compact Mode Standby Sup Reset—Sup 22 18. Compact Mode Standby Sup Reset—Sup 720 19. Compact Mode RPR+ Failover—Sup 22 20. Compact Mode RPR+ Failover—Sup 720 21. Compact Mode SSO Failover—Sup 22 22. Compact Mode SSO Failover—Sup 720
Hardware Integrity	Hardware Redundancy, page 40	1. Truncated Mode Standby Sup Reset—Sup 22 2. Truncated Mode Standby Sup Reset—Sup 720 3. Truncated Mode RPR+ Failover—Sup 22 4. Truncated Mode RPR+ Failover—Sup 720 5. Truncated Mode SSO Failover—Sup 22 6. Truncated Mode SSO Failover—Sup 720
	Hardware Reliability, page 64	1. LC in Second Sup Slot Reset 2. LC DFC Reset after Write Mem—Sup 720 3. Power Cycle 4. Rapid Link Flap

Table F-2 Safe Harbor Cisco IOS Release 12.2(18)SXF6 Testing Summary (continued)

Test Suites	Features/Functions	Tests
Layer 2 Features	Unidirectional Link Detection-Aggressive Mode, page 67	1. UDLD on Layer 2 Link—Sup 22 2. UDLD on Layer 2 Link—Sup 720 3. UDLD on Layer 3 Link—Sup 22 4. UDLD on Layer 3 Link—Sup 720
	Port Aggregation Protocol (Channeling), page 70	1. Basic L2 Channeling Configuration—Sup 22 2. Basic L2 Channeling Configuration—Sup 720 3. Basic L3 Channeling Configuration—Sup 22 4. Basic L3 Channeling Configuration—Sup 720 5. Basic L2 LACP and Negative IOS to CatOS—Sup 22 6. Basic L2 LACP and Negative IOS to CatOS—Sup 720 7. Basic L2 PAgP and Negative IOS to CatOS—Sup 22 8. Basic L2 PAgP and Negative IOS to CatOS—Sup 720 9. L2 EtherChannel Failure/Recovery—Sup 22 10. L2 EtherChannel Failure/Recovery—Sup 720 11. L3 EtherChannel Failure/Recovery—Sup 22 12. L3 EtherChannel Failure/Recovery—Sup 720 13. L2 EtherChannel Load Balancing 14. L3 EtherChannel Load Balancing 15. Gigabit Ethernet Module Reset—Sup 22 16. Gigabit Ethernet Module Reset—Sup 720 17. L3 10-Gigabit Ethernet Module Reset—Sup 720 18. L3 10-Gigabit Ethernet Load Balancing 19. L3 10-Gigabit EtherChannel Fail/Recover 20. L2 DEC Flooding Due to Periodic Purge—Sup 22

Table F-2 Safe Harbor Cisco IOS Release 12.2(18)SXF6 Testing Summary (continued)

Test Suites	Features/Functions	Tests
Layer 2 Features	Port Aggregation Protocol (Channeling), page 70	1. L2 DEC Lost PI E Due To Timeout—Sup 22 2. L2 DEC Lost PI E Due To Timeout—Sup 720 3. L2 DEC Shut on DEC Port Unicast Flood—Sup 22 4. L2 DEC Shut on DEC Port Unicast Flood—Sup 720 5. L2 DEC Traffic No Flood to Linecards—Sup 22 6. L2 DEC Traffic No Flood to Linecards—Sup 720 7. L2 DEC MN Race Conditions—Sup 22 8. L2 DEC MN Race Conditions—Sup 720 9. L2 DEC to DEC Switching—Sup 22 10. L2 DEC to DEC Switching—Sup 720 11. L2 DEC Spanning Tree Mode Change—Sup 720 12. L2 DEC MAC OOB Stress—Sup 720 13. L2 DEC MAC OOB Sync Feature RPR+—Sup 720 14. L2 DEC MAC OOB Sync Feature SSO—Sup 720
	Trunking, page 120	1. Configuration and Failure Recovery—Sup 22 2. Configuration and Failure Recovery—Sup 720 3. VLAN Functionality—Sup 22 4. VLAN Functionality—Sup 720

Table F-2 Safe Harbor Cisco IOS Release 12.2(18)SXF6 Testing Summary (continued)

Test Suites	Features/Functions	Tests
Layer 3 Routing Features	IP Multicast, page 126	1. First Hop Router Functionality—Sup 22 2. First Hop Router Functionality—Sup 720 3. Last-Hop Router Functionality—Sup 22 4. Last-Hop Router Functionality—Sup 720 5. Static RP Functionality—Sup 22 6. Static RP Functionality—Sup 720 7. Static RP Failover First Hop Router Impact—Sup 22 8. Static RP Failover First Hop Router Impact—Sup 720 9. Static RP Failover Impact—Sup 22 10. Static RP Failover Impact—Sup 720 11. Static RP Failover Traffic Impact—Sup 22 12. Static RP Failover Traffic Impact—Sup 720 13. Auto RP Functionality—Sup 720 14. Auto RP Failover First Hop Router Impact—Sup 22 15. Auto RP Failover First Hop Router Impact—Sup 720 16. Auto RP Failover RP Impact—Sup 720 17. Auto RP Failover Traffic Impact—Sup 720 18. PIM DR Failover—Sup 22 19. PIM DR Failover—Sup 720 20. GEM Failover First/Last-Hop Router—Sup 22 21. GEM Failover First/Last-Hop Router—Sup 720 22. GEM Failover Layer 3 Interface—Sup 22 23. GEM Failover Layer 3 Interface—Sup 720 24. IGMP Functionality—Sup 22 25. IGMP Functionality—Sup 720 26. IGMP Join/Prune Stress—Sup 22 27. IGMP Join/Prune Stress—Sup 720 28. IGMP Join/Leave Functionality—Sup 22 29. IGMP Join/Leave Functionality—Sup 720 30. MSDP SA Delay—22 31. MSDP SA Delay—720 32. Bidirectional PIM DF Election—Sup 720 33. Bidirectional PIM RP Failover RP Impact—Sup 720 34. Bidirectional PIM RP Failover Traffic Impact—Sup 720

Table F-2 Safe Harbor Cisco IOS Release 12.2(18)SXF6 Testing Summary (continued)

Test Suites	Features/Functions	Tests
Layer 3 Routing Features	IP Multicast, page 126	1. Layer 2 GEC Failover—SH3-110 to Dista-2 2. Layer 2 GEC Failover—SH3-108 to Dista-1 3. Layer 2 GEC Failover—SH3-106 to Dista-2 4. Layer 2 GEC Failover—SH3-102 to Dista-1 5. Layer 3 GEC Failover—SH3-104 to SH3-110 6. Layer 3 GEC Failover—SH3-104 to SH3-109 7. Layer 3 GEC Failover—SH3-104 to SH3-108 8. Layer 3 GEC Failover—SH3-104 to SH3-107 9. Layer 3 GEC Failover—SH3-100 to SH3-106 10. Layer 3 GEC Failover—SH3-100 to SH3-105 11. Layer 3 GEC Failover—SH3-100 to SH3-104 12. Layer 3 GEC Failover—SH3-100 to SH3-102 13. Layer 3 GEC Failover—SH3-100 to SH3-101 14. Layer 3 GEC Failover—SH3-100 to SH3-97 15. Multicast Stub and Non-RPF Rate Limiting —Sup 720 16. GEM Failover on Auto RP—Sup 720 17. 10-GEM Failover on Auto RP—Sup 720
Layer 3 Routing Features	Border Gateway Protocol, page 206	1. BGP Functionality 2. Route Scaling 3. BGP Authentication Failure 4. BGP Peer Flapping 5. BGP Route Flap—With Dampening 6. BGP Route Flap—No Dampening 7. Inbound Route Map —Sup 720 8. Outbound Route Map with Peer Group—Sup 720 9. Redistribution OSPF to BGP 10. Redistribution EIGRP to BGP 11. Redistribution OSPF and EIGRP to BGP
	Cisco Express Forwarding, page 216	1. CEF Packet Switching—Sup 22 2. CEF Packet Switching—Sup 720 3. CEF FIB Consistency Verification—Sup 22 4. CEF FIB Consistency Verification—Sup 720 5. CEF Many-to-One Traffic Distribution 6. CEF Many-to-Many Packet Distribution

Table F-2 Safe Harbor Cisco IOS Release 12.2(18)SXF6 Testing Summary (continued)

Test Suites	Features/Functions	Tests
Layer 3 Routing Features	Enhanced Interior Gateway Routing Protocol, page 222	1. EIGRP Functionality 2. EIGRP Authentication Failure 3. EIGRP Neighbor Scaling 4. EIGRP Route Summarization 5. EIGRP Redistribution OSPF to EIGRP
	Hot Standby Routing Protocol, page 227	1. HSRP Failover with Default Timers—Sup 22 2. HSRP Failover with Default Timers—Sup 720 3. HSRP Failover with Fast Timers—Sup 22 4. HSRP Failover with Fast Timers—Sup 720 5. HSRP Traffic Impact on CPU—Sup 22 6. HSRP Traffic Impact on CPU—Sup 720 7. HSRP Recovery from System Failure—Sup 22 8. HSRP Recovery from System Failure—Sup 720 9. HSRP Maximum Group Limit—Sup 22 10. HSRP Maximum Group Limit—Sup 720 11. Distributed GEM Failover—Sup 720
Layer 3 Routing Features	Open Shortest Path First, page 240	1. OSPF Basic Functionality 2. OSPF Autocost 3. OSPF Authentication Failure 4. OSPF Passive Interface 5. OSPF Filtering 6. OSPF Redistribution EIGRP to OSPF 7. OSPF Topology Database Verification
	Software Routing, page 250	1. Baseline Testing
Network Management Features	Simple Network Management Protocol, page 252	1. SNMP Functionality—Sup 22 2. SNMP Functionality—Sup 720 3. Config Copy via SNMP—Sup 22 4. Config Copy via SNMP—Sup 720 5. SNMP Malformed Packet 6. Walk of DUT—Sup 22 7. Walk of DUT—Sup 720 8. Config Synch—Sup 22 9. Config Synch—Sup 720
Miscellaneous Features	TACACS, page 258	1. User Authentication—Sup 22 2. User Authentication—Sup 720

Table F-2 Safe Harbor Cisco IOS Release 12.2(18)SXF6 Testing Summary (continued)

Test Suites	Features/Functions	Tests
Miscellaneous Features	Resiliency Verification, page 260	1. Steady-State Function and Resiliency Verification
	Security, page 261	1. 802.1x Authentication with EAP/MD5—Sup 22 2. 802.1x Authentication with EAP/MD5—Sup 720 3. 802.1x Authentication Negative Tests—Sup 22 4. 802.1x Authentication Negative Tests—Sup 720 5. NMAP Port Scanning—Sup 22 6. NMAP Port Scanning—Sup 720 7. Bad TACACS Login—Sup 22 8. Bad TACACS Login—Sup 720
	Crash Testing, page 269	1. Software Crash with FTP Core File
	Web Cache Communication Protocol (WCCP), page 270	1. WCCP—Version 1 Functionality 2. WCCP—Version 2 Functionality
	Jumbo Frame, page 271	1. Jumbo Frame Support for Unicast Traffic
	Netflow Data Export, page 273	1. NDE Functionality—Sup 22 2. NDE Functionality—Sup 720
	Dynamic Host Control Protocol, page 275	1. DHCP Functionality—Sup 22 2. DHCP Functionality—Sup 720

Table F-2 Safe Harbor Cisco IOS Release 12.2(18)SXF6 Testing Summary (continued)

Test Suites	Features/Functions	Tests
Miscellaneous Features	Switched Port Analyzer, page 277	1. Handling of PIM Packets—Sup 22 2. Handling of PIM Packets—Sup 720 3. Transmit Only Multicast—Sup 22 4. Transmit Only Multicast—Sup 720 5. Receive Only Multicast—Sup 22 6. Receive Only Multicast—Sup 720 7. Transmit/Receive Multicast—Sup 22 8. Transmit/Receive Multicast—Sup 720 9. Transmit Only Unicast—Sup 22 10. Transmit Only Unicast—Sup 720 11. Receive Only Unicast—Sup 22 12. Receive Only Unicast—Sup 720 13. Transmit/Receive Unicast—Sup 22 14. Transmit/Receive Unicast—Sup 720 15. Remote Span Transmit Only Multicast—Sup 22 16. Remote Span Transmit Only Multicast—Sup 720 17. Remote Span Receive Only Multicast—Sup 22 18. Remote Span Receive Only Multicast—Sup 720 19. Remote Span Transmit/Receive Multicast—Sup 22 20. Remote Span Transmit/Receive Multicast—Sup 720 21. Remote Span Transmit Only Unicast—Sup 22 22. Remote Span Transmit Only Unicast—Sup 720 23. Remote Span Receive Only Unicast—Sup 22 24. Remote Span Receive Only Unicast—Sup 720 25. Remote Span Transmit/Receive Unicast—Sup 22 26. Remote Span Transmit/Receive Unicast—Sup 720
	Access Control Lists, page 296	1. ACL 100—Sup 22 2. ACL 100—Sup 720 3. ACL 101—Sup 22 4. ACL 101—Sup 720 5. ACL 110—Sup 22 6. ACL 110—Sup 720 7. ACL 131—Sup 22 8. ACL 131—Sup 720

Table F-2 Safe Harbor Cisco IOS Release 12.2(18)SXF6 Testing Summary (continued)

Test Suites	Features/Functions	Tests
Miscellaneous Features	Parser, page 302	1. Parser via Telnet—Sup 22 2. Parser via Telnet—Sup 720 3. Parser via Telnet BGP—Sup 720 4. Parser via Telnet EIGRP—Sup 22 5. Parser via Telnet EIGRP—Sup 720 6. Parser via Telnet RIP—Sup 22 7. Parser via Telnet RIP—Sup 720 8. Parser via Telnet OSPF—Sup 22 9. Parser via Telnet OSPF—Sup 720 10. Parser via SSH—Sup 22 11. Parser via SSH—Sup 720 12. Parser via SSH BGP—Sup 720 13. Parser via SSH EIGRP—Sup 22 14. Parser via SSH EIGRP—Sup 720 15. Parser via SSH RIP—Sup 22 16. Parser via SSH RIP—Sup 720 17. Parser via SSH OSPF—Sup 22 18. Parser via SSH OSPF—Sup 720

Table F-2 Safe Harbor Cisco IOS Release 12.2(18)SXF6 Testing Summary (continued)

Test Suites	Features/Functions	Tests
Miscellaneous Features	Network Address Translation, page 311	1. NAT Scaling—Sup 22 2. NAT Scaling—Sup 720 3. Static 2 Hosts—Sup 22 4. Static 2 Hosts—Sup 720 5. Static 2 Hosts with Multicast—Sup 22 6. Static 2 Hosts with Multicast—Sup 720 7. Static 2 Hosts with Jumbo Frames—Sup 22 8. Static 2 Hosts with Jumbo Frames—Sup 720 9. Static 2 Hosts with Jumbo UDP Frames—Sup 22 10. Static 2 Hosts with Jumbo UDP Frames—Sup 720 11. Static 40 Hosts—Sup 22 12. Static 40 Hosts—Sup 720 13. Static 40 Hosts Extended—Sup 22 14. Static 40 Hosts Extended—Sup 720 15. Static 40 Hosts Overnight—Sup 22 16. Static 40 Hosts Overnight—Sup 720 17. Static 2 Networks—Sup 22 18. Static 2 Networks—Sup 720 19. Static Inside Dynamic Outside—Sup 22 20. Static Inside Dynamic Outside—Sup 720 21. Dynamic Inside Static Outside—Sup 22 22. Dynamic Inside Static Outside—Sup 720 23. Addition/Removal of NAT Statements—Sup 22 24. Addition/Removal of NAT Statements—Sup 720 25. Increment Inside Random Outside Match Host—Sup 22 26. Increment Inside Random Outside Match Host—Sup 720
	Network Time Protocol, page 331	1. NTP Functionality—Sup 22 2. NTP Functionality—Sup 720 3. NTP Failover—Sup 22 4. NTP Failover—Sup 720
	Syslog, page 334	1. Syslog Functionality—Sup 22 2. Syslog Functionality—Sup 720
	User Datagram Protocol Broadcast Flooding, page 336	1. UDP Broadcast Flooding—Sup 22 2. UDP Broadcast Flooding—Sup 720

Table F-2 Safe Harbor Cisco IOS Release 12.2(18)SXF6 Testing Summary (continued)

Test Suites	Features/Functions	Tests
Miscellaneous Features	Secure Shell, page 339	1. SSH Server Vulnerability—Sup 22 2. SSH Server Vulnerability—Sup 720
	Integrated Data Center Quality of Service, page 340	1. IDC Basic QoS—Sup22 2. IDC Basic QoS—Sup 720 3. QoS Effects on OSPF Functionality—Sup 22 4. QoS Effects on OSPF Functionality—Sup 720 5. QoS Effects on HSRP Functionality—Sup 22 6. QoS Effects on HSRP Functionality—Sup 720 7. QoS Effects on STP Functionality—Sup 22 8. QoS Effects on STP Functionality—Sup 720 9. IDC Overnight Stress—Sup 22 10. IDC Overnight Stress—Sup 720 11. IDC Inband Ping—Sup 22 12. IDC Inband Ping—Sup 720

Firewall Services Module (FWSM) 2.3.3.2

The following Safe Harbor Firewall Services Module (FWSM) 2.3.3.2 release types were tested.

- Single-Routed Firewall Services Module (FWSM) 2.3.3.2, page F-14
- Single-Transparent Firewall Services Module (FWSM) 2.3.3.2, page F-17
- Multi-Routed Firewall Services Module (FWSM) 2.3.3.2, page F-19
- Multi-Transparent Firewall Services Module (FWSM) 2.3.3.2, page F-21

Single-Routed Firewall Services Module (FWSM) 2.3.3.2

Table F-3 summarizes results of all completed testing as part of the Cisco Safe Harbor initiative for this release. Table F-3 includes the technology tested, the feature of function tested, and the component tests for each feature or function.

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

Table F-3 Safe Harbor Single-Routed Firewall Services Module (FWSM) 2.3.3.2 Test Results Summary

Test Suites	Feature/Function	Tests
Access List Control	ACL for IP Traffic, page 8	1. Object Group & Access-List 2. ACL Logging 3. Add Remark to ACL 4. Manual Commit ACL 5. 1K Inbound ACL 6. 1K Outbound ACL 7. 50K ACL Auto Commit Stress 8. 50K ACL Manual Commit Stress
Network Address Translation	High to Low Security, page 16	1. Dynamic PAT 2. Dynamic NAT 3. Dynamic NAT & PAT 4. Bidirectional NAT 5. NAT Outside 6. Dynamic Identify NAT 7. Static Identify NAT 8. NAT Exemption 9. Policy NAT

Table F-3 Safe Harbor Single-Routed Firewall Services Module (FWSM) 2.3.3.2 Test Results Summary (continued)

Test Suites	Feature/Function	Tests
	Low to High Security, page 22	1. Static NAT 2. Static PAT 3. Policy Static PAT
	Between Same Security, page 25	1. No NAT Between Same Security Interfaces 2. No NAT Between, but to Other Security Zone 3. No NAT Between Same Security Interfaces 4. No NAT Between and to Other Security Zone
	Connection Limits—TCP and UDP Interception, page 29	1. Max. TCP and UDP Connection
Network Address Translation	SYN Cookie, page 30	1. TCP SYN Attack 2. HTTP Performance Under SYN Attack 3. Passive FTP Under SYN Attack 4. Active FTP Under SYN Attack
L7 Fixup	ICMP, page 33	1. ICMP Fixup 2. ICMP Error Fixup
	FTP, page 35	1. Fixup Passive FTP 2. Fixup Active FTP 3. FTP Fixup Performance
L7 Fixup	SMTP Fixup, page 38	1. SMTP Fixup 2. SMTP Fixup Performance
	DNS Fixup and DNS Guard, page 40	1. DNS Fixup NAT 2. DNS Fixup—Inspect Length 3. DNS Guard—Inspect Multiple Reply 4. DNS Fixup Performance

Table F-3 Safe Harbor Single-Routed Firewall Services Module (FWSM) 2.3.3.2 Test Results Summary (continued)

Test Suites	Feature/Function	Tests
AAA Authentication, Authorization, and Accounting	AAA for Network Traffic, page 43	1. TACACS+ Authenticate Outside to DMZ2 2. TACACS+ Authenticate Inside to Outside 3. TACACS+ Authenticate Same Security Interface 4. TACACS+ Authorize Outside to DMZ2 5. TACACS+ Authorize Inside to Outside 6. TACACS+ Authorize to Same Security Interface 7. RADIUS Authenticate Outside to DMZ2 8. RADIUS Authenticate Inside to Outside 9. RADIUS Authenticate Same Security Interface 10. 2K AAA ACL Stress 11. Same User from Multiple IP 12. Multiple AAA Servers Stress 13. Authentication Rate
	AAA for Admin Traffic, page 58	1. TACACS+ Authorization 2. RADIUS Authentication 3. TACACs + Authentication Fallback to LOCAL
	AAA HTTPS Proxy, page 61	1. HTTP Cut-Through Proxy 2. HTTPS Cut-Through Proxy
	AAA Virtual Server, page 62	1. Virtual Telnet Server 2. Virtual HTTP Server
Miscellaneous Features	Simple Network Management Protocol (SNMP), page 64	1. SNMP Walks on FWSM Mibs
	OSPF, page 66	1. OSPF Functionality 2. OSPF Interface Parameters 3. OSPF Distributed Rotues
	SYSLOG, page 68	1. Syslog Functionality 2. Syslog Performance 3. Syslog Standby
	Parser, page 71	1. FWSM Single Parser
Hardware Redundancy	Failover/Fallback FWSM, page 73	1. Fail FWSM on Stateless Connection 2. Fail FWSM on Stateful Connection 3. Fail FWSM on Long Lasting Connection Traffic 4. Suspend Config Sync Between Active Standby

Table F-3 Safe Harbor Single-Routed Firewall Services Module (FWSM) 2.3.3.2 Test Results Summary (continued)

Test Suites	Feature/Function	Tests
	Failover/Fallback Switch, page 76	1. Fail Active FWSM
	Failover/Fallback Links, page 77	1. Fail Stateful Links 2. Fail Data Links 3. Fail Links to Upstram Router 4. Fail Links to Access Switch

Single-Transparent Firewall Services Module (FWSM) 2.3.3.2

Table F-4 summarizes results of all completed testing as part of the Cisco Safe Harbor initiative for this release. Table F-4includes the technology tested, the feature of function tested, and the component tests for each feature or function.

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

Table F-4 Safe Harbor Single-Transparent Firewall Services Module (FWSM) 2.3.3.2 Test Results Summary

Test Suites	Feature/Function	Tests
Access List Control	ACL for IP Traffic, page 8	1. Object Group & Access-List 2. ACL Logging 3. Add Remark to ACL 4. Manual Commit ACL 5. 1K Inbound ACL 6. 1K Outbound ACL 7. 50K ACL Auto Commit Stress 8. 50K ACL Manual Commit Stress
	Multicast Traffic from Outside to Inside, page 16	1. Multicast Traffic from Outside to Inside
	Routing Updates, page 16	1. Routing Updates
Network Address Translation	Connection Limits—TCP and UDP Interception, page 17	1. Max. TCP and UDP Connection
	SYN Cookie, page 19	1. TCP SYN Attack 2. HTTP Performance Under SYN Attack 3. Passive FTP Under SYN Attack 4. Active FTP Under SYN Attack
L7 Fixup	ICMP, page 22	1. ICMP Fixup 2. ICMP Error Fixup

Table F-4 Safe Harbor Single-Transparent Firewall Services Module (FWSM) 2.3.3.2 Test Results Summary

Test Suites	Feature/Function	Tests
	FTP, page 24	1. FTP Fixup—Passive FTP 2. FTP Fixup—Active FTP 3. FTP Fixup—Stress
	DNS Fixup and DNS Guard, page 26	1. DNS Fixup—Inspect Length 2. DNS Guard—Inspect Multiple Reply 3. DNS Fixup—Stress
AAA Authentication, Authorization, and Accounting	AAA for Network Traffic, page 29	1. TACACS+ Authentication Inside to Outside 2. RADIUS Authentication Outside to Inside 3. 2K AAA ACL 4. 2K AAA ACL Manual Commit 5. Same User from Multiple IP 6. Multiple AAA Servers Stress 7. Authentication Rate Stress
	AAA for Admin Traffic, page 36	1. TACACS+ Authentication & Authorization 2. RADIUS Authentication 3. TACACs + Authentication Fallback to LOCAL 4. Radius Authentication Fallback to LOCAL
	AAA HTTPS Proxy, page 39	1. HTTPS Cut-Through Proxy
	AAA Virtual Server, page 40	1. Virtual Telnet Server
Miscellaneous Features	Simple Network Management Protocol (SNMP), page 41	1. SNMP Walks on FWSM Mibs
	SYSLOG, page 43	1. Syslog Functionality 2. Syslog Performance 3. Syslog Standby
	Parser, page 45	1. FWSM Single Parser
Hardware Redundancy	Failover/Fallback FWSM, page 48	1. Fail FWSM on Stateless Connection 2. Fail FWSM on Stateful Connection 3. Fail FWSM on Long Lasting Connection Traffic 4. Fail FWSM on Mcast Traffic 5. Suspend Config Sync between Active Standby 6. Suspend Config Sync between Active Standby Manual
	Failover/Fallback Switch, page 52	1. Fail Active FWSM
	Failover/Fallback Links, page 53	1. Fail Stateful Links 2. Fail Data Links 3. Fail Links to Access Switch

Multi-Routed Firewall Services Module (FWSM) 2.3.3.2

Table F-5 summarizes results of all completed testing as part of the Cisco Safe Harbor initiative for this release. Table F-5 includes the technology tested, the feature of function tested, and the component tests for each feature or function.

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

Table F-5 Safe Harbor Multi-Routed Firewall Services Module (FWSM) 2.3.3.2 Test Results Summary

Test Suites	Feature/Function	Tests
Baseline Testing	Baseline Testing, page 10	1. Short Lived HTTP Flows 2. Short Lived FTP Flows 3. Long Lived FTP Flows
Access List Control	ACL for IP Traffic, page 22	1. Object Group Auto Commit 2. Object Group Manual Commit 3. ACL Logging 4. Add Remark to ACL
Resource Usage	Resource Usage, page 28	1. Connection Rate Across Contexts 2. Syslog Rate Across Contexts 3. Fixup Rate Across Contexts 4. Total Connections Across Contexts 5. Total Xlate Across Contexts 6. Total SSH Session Across Contexts
Network Address Translation	High to Low Security, page 56	1. Dynamic NAT 2. Dynamic PAT 3. Dynamic NAT & PAT 4. Bidirectional NAT 5. NAT Outside 6. Dynamic Identify NAT 7. Static Identify NAT 8. NAT Exemption 9. Policy NAT
	Low to High Security, page 67	1. Static NAT 2. Static PAT 3. Static Policy PAT

Table F-5 Safe Harbor Multi-Routed Firewall Services Module (FWSM) 2.3.3.2 Test Results Summary (continued)

Test Suites	Feature/Function	Tests
	Between Same Security, page 72	1. No NAT Between Same Security Interfaces 2. No NAT Between, but to Other Security Zone 3. NAT Between Same Security Interfaces 4. NAT Between, and to Other Security Zone
	Connection Limits—TCP and UDP Interception, page 81	1. Max. TCP and UDP Connection
	SYN Cookie, page 84	1. TCP SYN Attack 2. HTTP Performance Under SYN Attack 3. Passive FTP Under SYN Attack 4. Active FTP Under SYN Attack
L7 Fixup	ICMP, page 90	1. ICMP Fixup 2. ICMP Error Fixup
	FTP, page 93	1. Passive FTP 2. Active FTP 3. FTP Fixup Stress
	DNS Fixup and Guard, page 99	1. DNS Fixup Stress 2. DNS Fixup—Max. Length of DNS Reply 3. DNS NAT Fixup 4. DNS Guard—Inspect Multiple Reply
AAA Authentication, Authorization, and Accounting	AAA for Network Traffic, page 107	1. TACACS+ Authentication 2. TACACS+ Authorization 3. RADIUS Authenticate & Authorize 4. AAA Same User Multiple IP 5. AAA Stress Rate
	AAA for Admin Traffic, page 114	1. TACACS+ Authenticate & Authorize 2. RADIUS Authentication 3. TACACs + Fallback to LOCAL 4. RADIUS Fallback to LOCAL
	AAA HTTPS Proxy, page 119	1. HTTPS Cut-Through Proxy
	AAA Virtual Server, page 120	1. Virtual Telnet Server
Miscellaneous Features	Simple Network Management Protocol (SNMP), page 121	1. SNMP Walks
	SYSLOG, page 123	1. Syslog Functionality 2. Syslog Performance 3. Syslog Standby

Table F-5 Safe Harbor Multi-Routed Firewall Services Module (FWSM) 2.3.3.2 Test Results Summary (continued)

Test Suites	Feature/Function	Tests
	Parser, page 128	1. FW Parser Config Mode 2. FW Parser Enable Mode
Hardware Redundancy	Failover/Fallback FWSM, page 131	1. Fail FWSM on Stateless Connection 2. Fail FWSM on Stateful Connection 3. Fail FWSM on Long Lasting Connection
	Failover/Fallback Switch, page 137	1. Fail Active Sup 720 on Active Switch 2. Fail Active FWSM
	Failover/Fallback Links, page 142	1. Fail Stateful Links 2. Fail Data Links 3. Fail Links to Upstram Router 4. Fail Links to Access Switch

Multi-Transparent Firewall Services Module (FWSM) 2.3.3.2

Table F-6 summarizes results of all completed testing as part of the Cisco Safe Harbor initiative for this release. Table F-6 includes the technology tested, the feature of function tested, and the component tests for each feature or function.

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

Table F-6 Safe Harbor Multi-Transparent Firewall Services Module (FWSM) 2.3.3.2 Test Results Summary

Test Suites	Feature/Function	Tests
Baseline Testing	Baseline Testing, page 9	1. Short Lived HTTPS Flows 2. Long Lived FTP Flows 3. Multicast From Outside to Inside 4. Routing Updates
ACL for IP Traffic	ACL for IP Traffic, page 24	1. Object Group Auto Commit 2. Object Group Manual Commit 3. ACL Logging 4. Inbound ACL Manual Commit 5. 1K Inbound ACL with Traffic 6. 1K Outbound ACL with Traffic 7. 50K ACL Auto Commit Stress

Table F-6 Safe Harbor Multi-Transparent Firewall Services Module (FWSM) 2.3.3.2 Test Results Summary (continued)

Test Suites	Feature/Function	Tests
Resource Usage	Resource Usage, page 34	1. Connection Rate Across Contexts 2. Syslog Rate Across Contexts 3. Fixup Rate Across Contexts 4. Total Connections Across Contexts 5. Total Xlate Across Contexts 6. Total SSH Session Across Contexts
Network Address Translation	Connection Limits—TCP and UDP Interception, page 59	1. Max. Connection Static NAT
	SYN Cookie, page 64	1. TCP SYN Attack 2. HTTP Performance Under SYN Attack 3. Passive FTP Under SYN Attack 4. Active FTP Under SYN Attack
L7 Fixup	ICMP, page 79	1. ICMP Fixup 2. ICMP Error Fixup
L7 Fixup	FTP, page 87	1. FTP Fixup Passive FTP 2. FTP Fixup Active FTP
	DNS Fixup and DNS Guard, page 97	1. DNS Fixup Stress 2. DNS Fixup Max Reply Length 3. DNS Guard
Authentication, Authorization, and Accounting	AAA for Network Traffic, page 110	1. TACACS+ Authentication and Authorization 2. RADIUS Authentication and Authorization 3. Same User with Multiple IP 4. Multiple AAA server 5. AAA Authentication and Authorization Rate
	AAA for Admin Traffic, page 139	1. TACACS Authetication and Authorization for Mgmt Traffic 2. RADIUS AAA for Admin Traffic 3. TACACS+ Authentication and Authorization Fallback for Mgmt 4. RADIUS Authentication Fallback for Mgmt
	AAA HTTPS Proxy, page 156	1. HTTPS Cut-Through Proxy
	AAA Virtual Server, page 160	1. Virtual Telnet Server
Miscellaneous Features	Simple Network Management Protocol (SNMP), page 165	1. SNMP Walks
	SYSLOG, page 167	1. Syslog Functionality 2. Syslog Performance 3. Syslog Standby

Table F-6 Safe Harbor Multi-Transparent Firewall Services Module (FWSM) 2.3.3.2 Test Results Summary (continued)

Test Suites	Feature/Function	Tests
	Parser, page 180	1. Parser for Config Mode 2. Parser for Enable Mode
Hardware Redundancy	Failover/Fallback FWSM, page 183	1. Failover on Stateless Connection 2. Failover on Stateful Connection 3. Failover on Long Lasting Connection Traffic 4. Failover with Mcast Traffic 5. Failover with Config Sync Disabled 6. Failover with Suspend Config Sync and Manual Commit
Hardware Redundancy	Failover/Fallback Switch, page 209	1. Fail Active FWSM
	Failover/Fallback Links, page 214	1. Fail Stateful Links 2. Fail Data Links 3. Fail Link to Upstream Router 4. Fail Link to Access Switch

Content Switching Module (CSM) 4.1.6

Table F-7 and Table F-8 summarizes the tests executed as part of the Cisco Safe Harbor initiative. Table F-7 includes the feature or function tested for router mode functionality, and the component tests for each feature or function. Table F-8 includes the feature or function tested for bridged mode functionality, and the component tests for each feature or function.

Note All tests passed unless noted with a DDTS numbers.

Table F-7 Safe Harbor Router Mode Content Switching Module (CSM) 4.1.6 Test Results Summary

Features/Functions	Tests
Basic Functionality, page 7	1. CLI Parser 2. Lifetime of Idle Connections 3. DEC RSPAN with CSM Load Balance—Sup 720 4. DEC SPAN with CSM Load Balance—Sup 720 5. Addition or Removal of Real Servers 6. Failaction Purge 7. Route Health Injection 8. SNMP MIB Walks
Load Balancing Predictors, page 22	1. IP Address Hashing 2. Least Connection 3. Max Connection 4. Round Robin 5. Server Weighting 6. URL Hashing 7. SASP

Table F-7 Safe Harbor Router Mode Content Switching Module (CSM) 4.1.6 Test Results Summary (continued)

Features/Functions	Tests
Traffic Handling, page 42	1. Redirect Policy 2. URL Maps 3. Cookie Maps 4. Header Maps 5. GSLB 6. Ping Handling 7. URL Lengths 8. Load-Balance of Non-TCP Traffic 9. Persistence Rebalance 10. Netmask Sticky 11. SSL Sticky 12. Cookie Sticky
Health and Redundancy, page 70	1. Sorry Server (Serverfarm Redundancy) 2. Port Reset 3. VLAN Reset 4. Module Reset 5. Intraswitch CSM Redundancy 6. Interswitch CSM Redundancy 7. Health Probes

Table F-8 Safe Harbor Bridged Mode Content Switching Module (CSM) 4.1.6 Test Results Summary

Features/Functions	Tests
Basic Functionality, page 92	1. Lifetime of Idle Connections 2. Addition or Removal of Real Servers 3. Fail Action Purge
Traffic Handling, page 98	1. Redirect Policy 2. Ping Handling 3. Load Balance of Non-TCP Traffic
Health and Redundancy, page 105	1. Port Reset 2. VLAN Reset 3. Intraswitch CSM Blade VIP Redundancy 4. Interswitch CSM Device Redundancy 5. Health Probes

Secure Socket Layer Module (SSLM) 3.1.1

Table F-9 summarizes the tests to be executed as part of the Secure Socket Layer Module (SSLM) 3.1.1 Safe Harbor initiative. Table F-9 includes the component tests for each feature or function.

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

Table F-9 Safe Harbor Secure Socket Layer Module (SSLM) 3.1.1 Test Results Summary

Features/Functions	Tests
Secure Socket Layer (SSL)	1. Upgrade 2. CLI Parser 3. Manual Certificate Signing Request 4. Certificate and Key Importation with PEM Paste 5. Graceful Certificate Rollover 6. URL Rewrite 7. SSL Client Proxy Services 8. Client Authentication 9. HTTP Header Insert Policy Client IP Port 10. HTTP Header Insert Policy Client Cert 11. HTTP Header Insert Custom Header 12. HTTP Header Insert Session 13. HTTP Header Insert Policy ALL 14. SSL Termination 15. SSLSM Configuration Virtualization 16. Export Ciphers 17. Protected Key Storage 18. SSL Session Auto Renegotiation 19. Maximum Connections