

# Sun™ Secure Application Switch - N1000 Series: SSL and Layer 7 performance test

*Test report prepared under contract from Sun Microsystems*

## Executive summary

Sun Microsystems commissioned VeriTest, a division of Lionbridge Technologies, to conduct a performance audit to measure the SSL and Layer 7 performance of their Sun™ Secure Application Switch – N1000 Series. The specific configuration under test was the N1400. Sun supplied all test equipment and a N1400 switch running version V3\_0A51629 firmware and configured with four small form factor Gigabit Ethernet ports.

Sun requested that VeriTest audit performance tests, executed by Sun engineers, to ensure the following performance metrics on the Sun N1400 switch:

- Maximum SSL connections/second with no SSL session reuse
- Maximum SSL bulk cryptographic throughput with no SSL session reuse
- Maximum Layer 7 connections/second
- Maximum Layer 7 throughput

For each of the performance tests, Sun engineers configured and tested the Sun N1400 switch as a load balancer and TCP connection termination device positioned between clients generating HTTP requests and Web servers responding to requests. To provide Layer 2 aggregation to the N1400, Sun chose to use a 12-port switch as a port aggregator

Sun used a series of Spirent Avalanche 2500 and Reflector 2500 systems to generate the web client and server test loads. Sun performed three separate runs of each test to verify consistent results and rebooted the Sun N1400 switch before each run. Please refer to the Test Methodology section for a detailed description of the test methodology used for each test.

The Spirent Avalanche 2500™ and Reflector™ 2500 are load testing appliances that generate high levels of realistic network and user traffic. The Avalanche 2500 is able to generate extremely high-levels of multi-protocol traffic. For additional information on the Spirent Avalanche 2500 and Reflector 2500, see [http://www.spirentcom.com/analysis/product\\_line.cfm?wt=2&az-c=pl&PL=32](http://www.spirentcom.com/analysis/product_line.cfm?wt=2&az-c=pl&PL=32).

## SSL connections/second test results

For the SSL connections/second test, Sun created an Avalanche 2500 test suite that sent HTTP 1.0 SSL requests for a 1KB static object from the Avalanche 2500 clients to virtual IP addresses configured on the Sun N1400 switch. For each request, the Sun N1400 switch terminated the TCP connection from the client, negotiated an SSL session with the client, performed SSL decryption, and load balanced the request across a pool of Web servers, defined by the series of Spirent Reflector 2500 systems. The Reflector 2500 systems sent

### Key findings

- ❑ The Sun N1400 switch reached a maximum of 12,019 SSL connections/second in the test configuration.
- ❑ The Sun N1400 switch reached a maximum of 1.5423 Gigabits/second of encrypted SSL throughput in the test configuration.
- ❑ The Sun N1400 switch reached a maximum of 53,728 Layer 7 connections/second in the test configuration.
- ❑ The Sun N1400 switch reached a maximum of 1.9483 Gigabits/second of Layer 7 throughput in the test configuration.

their response in clear text to the Sun N1400 switch, which encrypted the data and returned the response to the client.

Figure 1 shows the results of each test run for the SSL connections/second test.

The Sun N1400 switch reached a maximum of 12,019 SSL connections/second in the test configuration.

Note that Sun engineers disabled SSL session reuse in the test suite, which required the Sun N1400 switch to perform a full SSL session negotiation for each incoming request. This increased the amount of SSL processing required by the Sun N1400 switch.

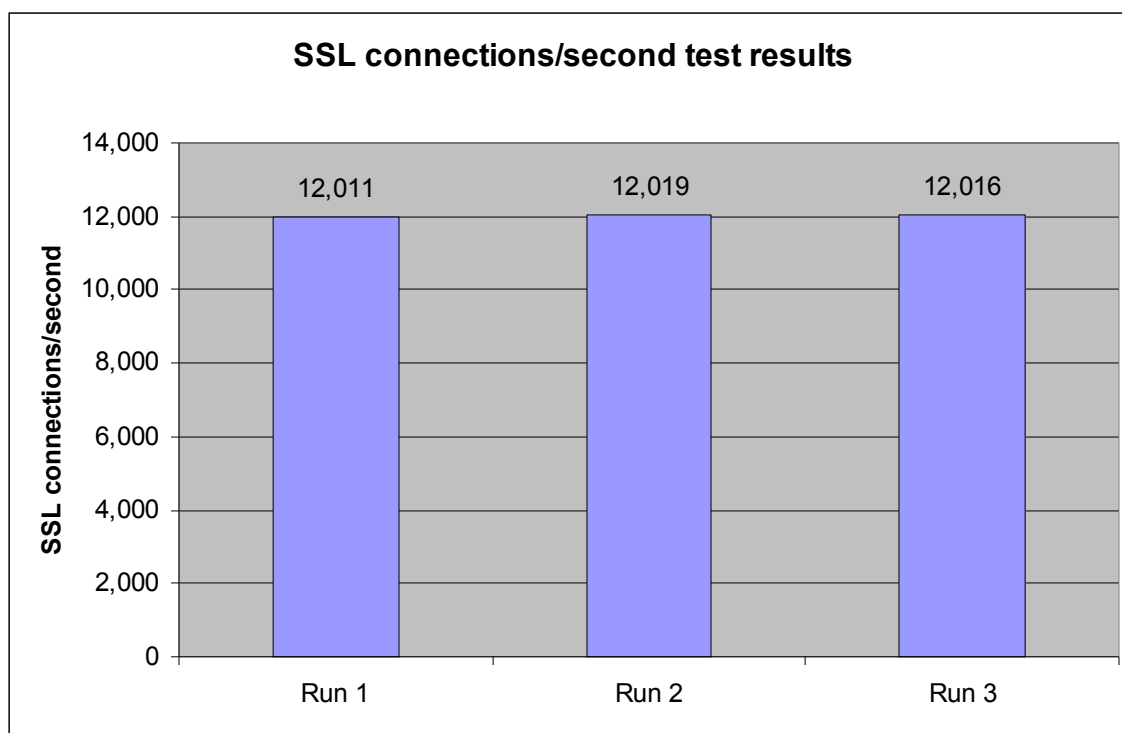
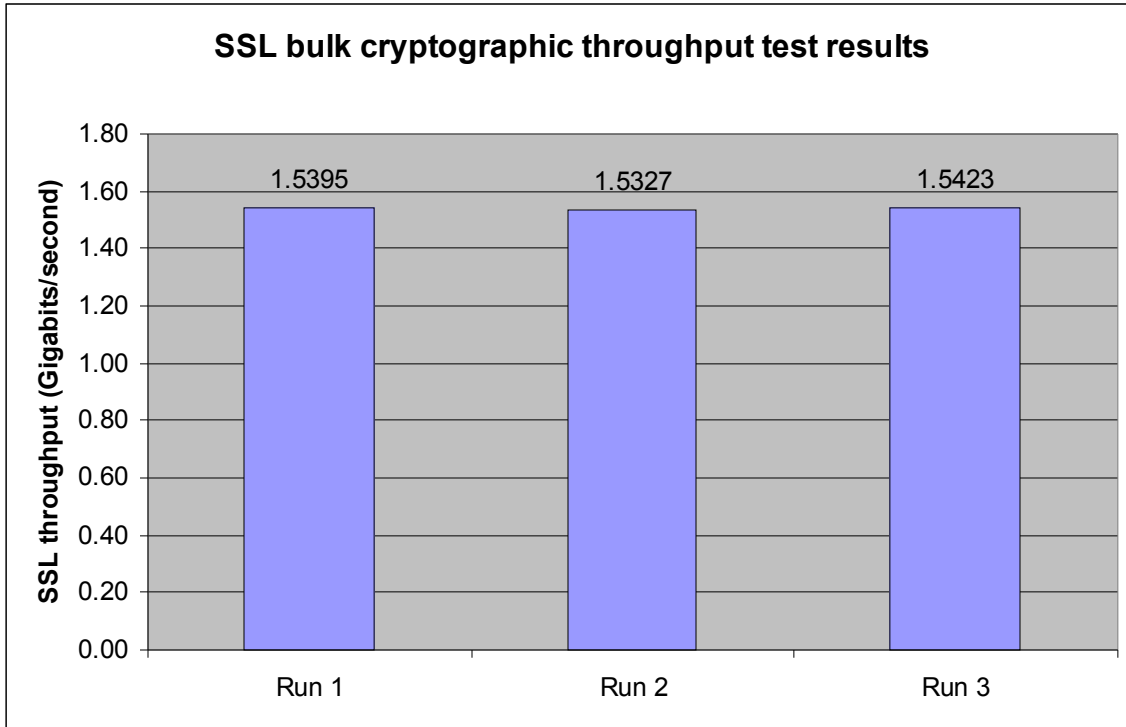


Figure 1: Sun N1400 SSL connections/second test results

### ***SSL bulk cryptographic throughput test results***

Figure 2 shows the results for the SSL bulk cryptographic test. Sun created an Avalanche 2500 test suite that sent HTTP 1.0 SSL requests for a 1MB static object from the Avalanche 2500 clients to a virtual IP address configured on the Sun N1400 switch. For each request, the Sun N1400 switch terminated the TCP connection from the client, negotiated an SSL session with the client, performed SSL decryption, and load balanced the request across a pool of Web servers. The Web servers sent responses in clear text to the Sun N1400 switch, which encrypted the data and returned the response to the client.

As shown in Figure 2, the Sun N1400 switch reached a maximum of 1.5423 Gbps SSL throughput in the test configuration.



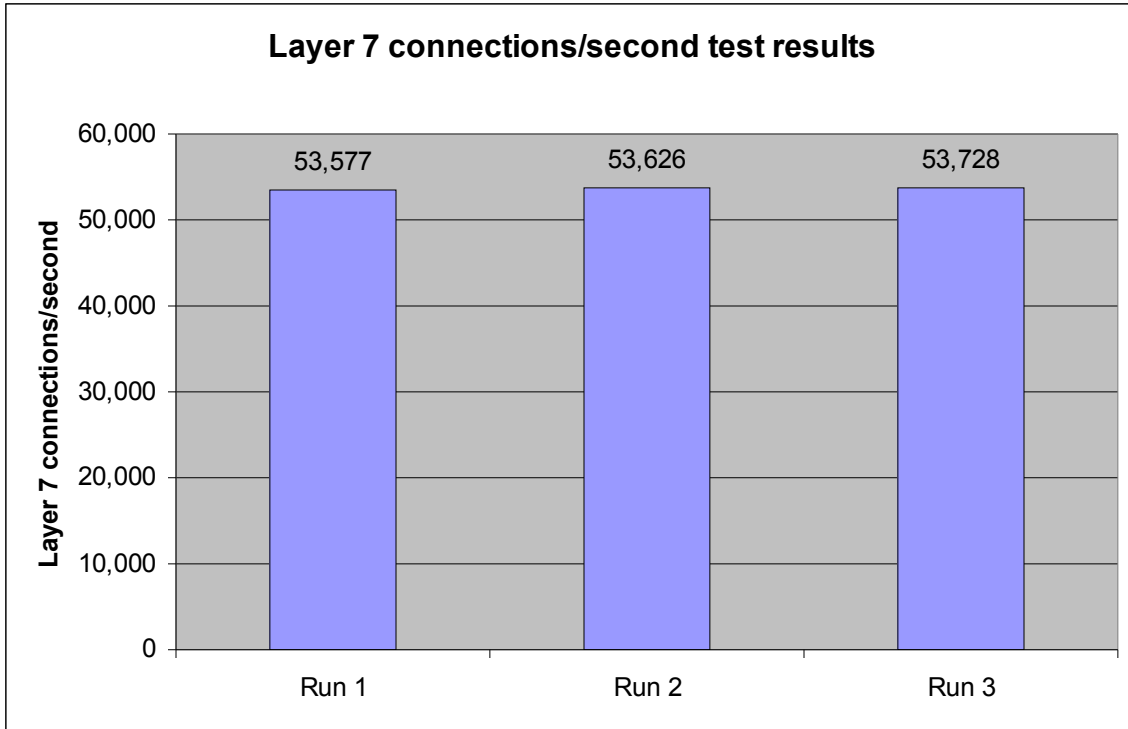
**Figure 2: Sun N1400 SSL bulk cryptographic throughput test results**

The Sun N1400 switch reached 1.5423 Gigabits/second of encrypted SSL throughput during our two test runs. As in the SSL connections/second test, Sun disabled SSL session reuse on the Avalanche 2500 clients requiring the switch to negotiate a new SSL session for each request.

### ***Layer 7 connections/second test results***

Figure 3 shows the results of each test run for the Layer 7 connections/second test. Sun created an Avalanche test suite that sent HTTP 1.0 requests for a 1KB static object from the Avalanche 2500 clients to a virtual IP address configured on the Sun N1400 switch. For each request, the switch terminated the TCP connection from the client, inspected the Layer 7 contents of the request, and load balanced the request based on the Layer 7 content across a pool of Web servers. The Web servers sent their responses to the Sun N1400 switch, which returned the data to the client.

The Sun N1400 switch reached a maximum of 53,728 Layer 7 connections/second in the test configuration.



**Figure 3: Sun N1400 Layer 7 connections/second test results**

### ***Layer 7 throughput test results***

Figure 4 shows the results for the Layer 7 throughput test. Sun created an Avalanche test suite that sent HTTP 1.0 requests for a 1MB static object from the Avalanche 2500 clients to a virtual IP address configured on the Sun N1400 switch. For each request, the switch terminated the TCP connection from the client, inspected the Layer 7 contents of the request, and load balanced the request based on the Layer 7 content across a pool of Web servers. The Web servers sent their responses to the Sun N1400 switch, which returned the data to the client.

The Sun N1400 switch reached a maximum of 1.9483 Gbps of Layer 7 throughput during our two test runs.

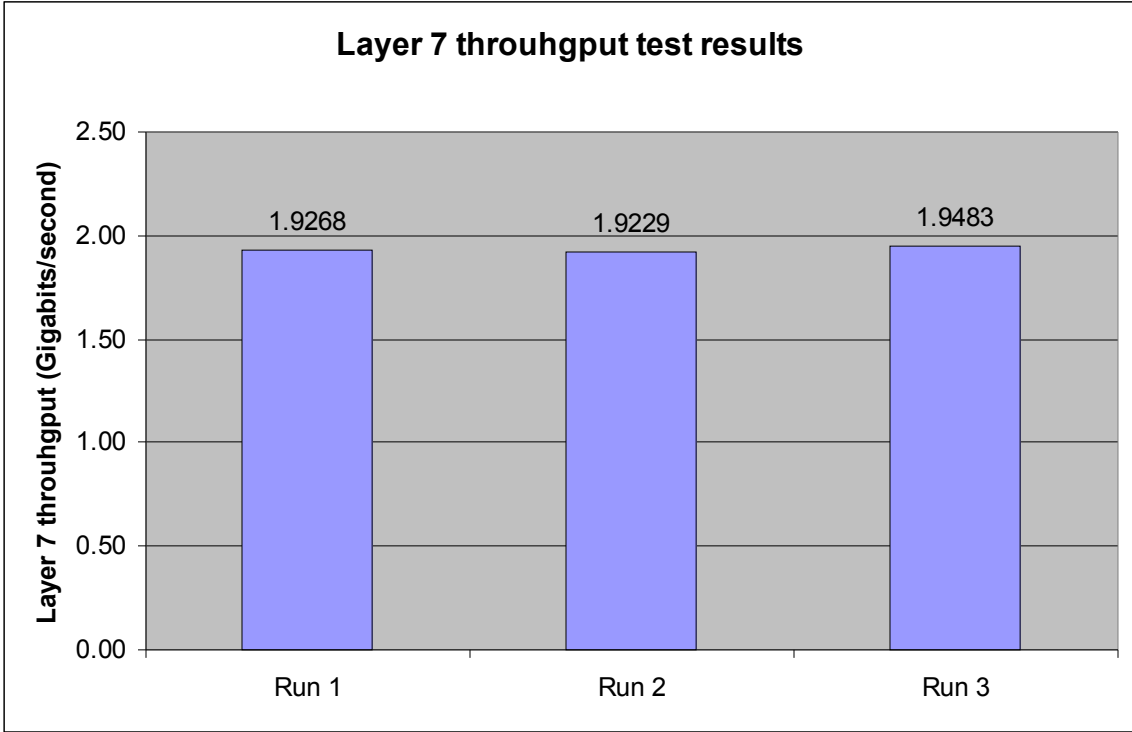


Figure 4: Sun N1400 Layer 7 throughput test results

## Test methodology

Sun Microsystems commissioned VeriTest, a division of Lionbridge Technologies, to conduct a performance audit to measure the SSL and Layer 7 performance of their Sun™ Secure Application Switch N1400. Sun supplied all test equipment and a N1400 switch running version V3\_0A51629 firmware and configured with four small form factor Gigabit Ethernet ports.

Sun requested that VeriTest audit performance tests, executed by Sun engineers, to ensure the following performance metrics on the Sun N1400 switch:

- Maximum SSL connections/second with no SSL session reuse
- Maximum SSL bulk cryptographic throughput with no SSL session reuse
- Maximum Layer 7 connections/second
- Maximum Layer 7 throughput

For each of the performance tests, the Sun N1400 switch was configured as a load balancer and TCP connection termination device positioned between clients generating HTTP requests and Web servers responding to requests. For the SSL tests, the Sun N1400 switch terminated the connection request from the client, decrypted the incoming request, load balanced the request across the pool of Web servers, encrypted the response data from the Web server, and returned the response to the client. For the Layer 7 tests, the Sun N1400 switch terminated the connection request from the client, inspected the HTTP request contents, load balanced the request across the pool of Web servers, and returned the response to the client.

To generate the necessary load to stress the Sun N1400 switch, Sun used a 12-port switch to provide port aggregation. Figures 5 and 6 illustrate how Sun incorporated this switch into the testbed.

Sun used a series of Spirent Avalanche 2500 and Reflector 2500 systems to generate the web client and server test loads. Sun performed three separate runs of each test to verify consistent results and rebooted the Sun N1400 switch before each run.

The Spirent Avalanche 2500™ and Reflector™ 2500 are load testing appliances that generate high levels of realistic network and user traffic. The Avalanche 2500 is able to generate extremely high-levels of multi-protocol traffic. For additional information on the Spirent Avalanche 2500 and Reflector 2500, see [http://www.spirentcom.com/analysis/product\\_line.cfm?wt=2&az-c=pl&PL=32](http://www.spirentcom.com/analysis/product_line.cfm?wt=2&az-c=pl&PL=32).

For the following tests,

- Maximum SSL connections/second with no SSL session reuse
- Maximum SSL bulk cryptographic throughput with no SSL session reuse
- Maximum Layer 7 throughput

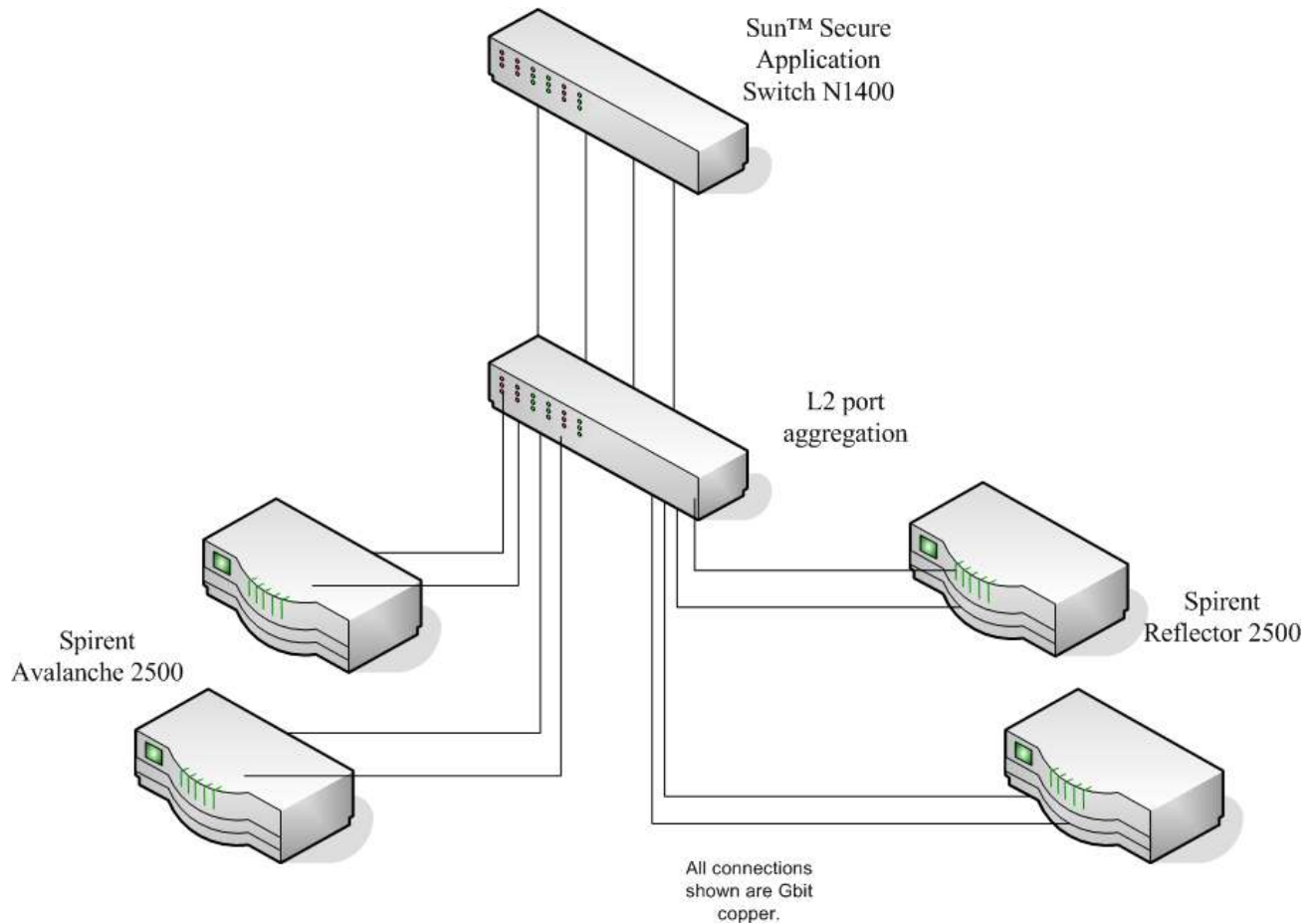
Sun used two Avalanche 2500 systems and two Reflector 2500 systems to generate the test load. The Spirent systems split the load equally across all systems. For the Maximum Layer 7 connections/second test, Sun used three Avalanche 2500 systems and two Reflector 2500 systems. They included the additional Avalanche 2500 system to generate a higher load required for this test.

The following sections describe the test methodology for each test. Note that Sun performed three separate runs of each test to verify consistent results and rebooted all test equipment and devices under test before each test run. Refer to the appendices for detailed information on the hardware and software used during the tests.

### **SSL connections/second test**

For the SSL connections/second test, Sun created a test suite that sent HTTP 1.0 SSL requests for a 1KB static object from the Avalanche 2500 clients to a virtual IP address configured on the Sun N1400 switch. For each request, the Sun N1400 switch terminated the TCP connection from the client, negotiated an SSL session with the client, performed SSL decryption, and load balanced the request across a pool of Web servers. The Web server sent its response in clear text to the Sun N1400 switch, which encrypted the data and returned the response to the client.

Figure 5 illustrates the test bed configuration used for the SSL connections/second and Layer 7 throughput test.



**Figure 5: SSL connections/second test bed**

Figure 6 shows the port connections on the Sun N2120 switch for the SSL connections/second test. All ports on the Sun N1400 switch and N2120 were set to auto negotiate link speed and duplex settings.

Port	Connection
1	Avalanche 2500 #1 – VLAN 100
2	Avalanche 2500 #1 – VLAN 100
3	Avalanche 2500 #2 – VLAN 100
4	Avalanche 2500 #2 – VLAN 100
5	N1400 – VLAN 100/200
6	N1400 – VLAN 100/200
7	N1400 – VLAN 100/200
8	N1400 – VLAN 100/200
9	Reflector 2500 #1 – VLAN 200
10	Reflector 2500 #1 – VLAN 200
11	Reflector 2500 #2 – VLAN 200
12	Reflector 2500 #2 – VLAN 200

**Figure 6: Sun N2120 port connections for the SSL connections/second test**

A Sun engineer configured and performed the testing on the Sun N1400 switch for the test. The Sun N1400 switch configuration specified one service group containing all Web servers. The service group used the round robin load balance type to distribute HTTP requests among the Web servers in the group. The configuration included two request policies each containing a single Layer 7 rule. The request policy matched all incoming requests (URI\_PATH matches “\*”). The configuration included two virtual services.

Figure 7 lists a summary of the Sun N1400 switch configuration for the SSL connections/second test. Refer to Appendix B for a listing of the switch configuration file used for the test.

Switch Setting	Value
LoadBalance->serviceGroup->In Line Health Check	Disabled
LoadBalance->requestPolicy->Optimize Last Response	Enabled
LoadBalance->requestPolicy->First Object Switching	Enabled
Resource->serviceBandwidth->Service Percent	100

**Figure 7: Sun N1400 switch configuration changes for the SSL connections/second test**

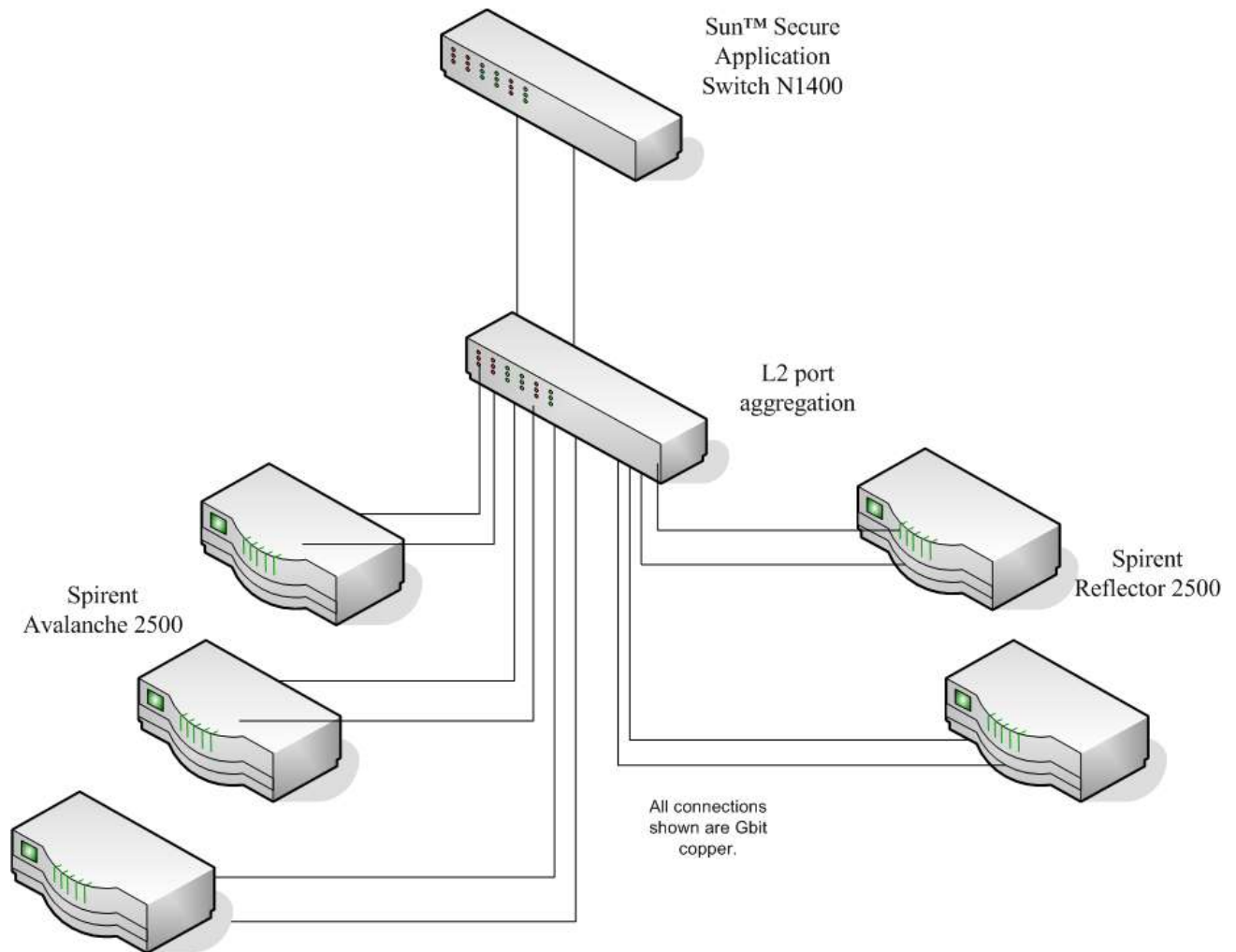
Sun created a custom test suite that generated HTTP SSL requests for a 1KB static object located on each Web server. All of the clients sent their requests to the virtual IP address configured on the Sun N1400 switch. The test suite contained one mix that included all clients in the test. Sun set the ramp up time and the delay time to 60 seconds. This staggered the start time of the clients for the first 60 seconds of the test. Sun configured an overall test length of ten minutes. Sun set the thread inactivity timeout to 120 seconds, modified the workload file to specify the URL of the 1KB file, and configured 100% percent of the requests to use SSL. Finally, Sun set the min and max values for the SSL reuse session identifiers to zero. This required the clients to renegotiate a new SSL session for each request and increased the amount of load on the Sun N1400 switch.

### ***SSL bulk cryptographic throughput test***

For the SSL bulk cryptographic throughput test, Sun created a custom test suite that sent HTTP 1.0 SSL requests for a 1MB static object from the clients to a virtual IP address configured on the Sun N1400 switch. For each request, the Sun N1400 switch terminated the TCP connection from the client, negotiated an SSL session with the client, performed SSL decryption, and load balanced the request across a pool of Web servers. The Web server sent its response in clear text to the Sun N1400 switch, which encrypted the data and returned the response to the client.

Sun used the same Web server configuration described in SSL connection/second test for the SSL bulk cryptographic throughput test. The only change was to copy a 1M byte static file to the document root on each Web server.

Figure 8 illustrates the test bed configuration used for the SSL bulk cryptographic throughput test.



**Figure 8: SSL bulk cryptographic connections/second testbed**

Figure 9 shows the port connections on the Sun N1400 switch for the SSL bulk cryptographic throughput test.

Port	Connection
1	Avalanche 2500 #1 – VLAN 100
2	Avalanche 2500 #1 – VLAN 100
3	Avalanche 2500 #2 – VLAN 100
4	Avalanche 2500 #2 – VLAN 100
5	Avalanche 2500 #3 – VLAN 100
6	Avalanche 2500 #3 – VLAN 100
7	N1400 – VLAN 100/200
8	N1400 – VLAN 100/200
9	Reflector 2500 #1 – VLAN 200
10	Reflector 2500 #1 – VLAN 200
11	Reflector 2500 #2 – VLAN 200
12	Reflector 2500 #2 – VLAN 200

**Figure 9: Sun N2120 port connections for the SSL bulk cryptographic test**

Figure 10 lists a summary of the Sun N1400 switch configuration for the SSL bulk cryptographic throughput test. Refer to Appendix B for a listing of the switch configuration file used for the test.

Switch Setting	Value
LoadBalance->realService->Advanced->Receive Window Size	65535
LoadBalance->realService->Advanced->SMM Stream Limit	4xRcvWindow
LoadBalance->virtualService->Advanced->Receive Window Size (HTTP)	65535
LoadBalance->virtualService->Advanced->SMM Stream Limit (HTTP)	4xRcvWindow
LoadBalance->virtualService->Advanced->Receive Window Size (HTTPS)	65535
LoadBalance->virtualService->Advanced->SMM Stream Limit (HTTPS)	4xRcvWindow
LoadBalance->serviceGroup->In Line Health Check	Disabled
LoadBalance->requestPolicy->Optimize Last Response	Enabled
LoadBalance->requestPolicy->First Object Switching	Enabled
System->Switch Services->tideRunner->initKeys->SMM Page Size	8
Resource->serviceBandwidth->Service Percent	100

**Figure 10: Sun N1400 switch configuration changes for the SSL bulk cryptographic throughput test**

Sun created a custom test suite that generated HTTP SSL requests for a 1M byte static object located on each Web server. All of the clients sent their requests to the virtual IP address configured on the Sun N1400 switch. The test suite contained one mix that included all clients in the test. Sun set the ramp up time and the delay time to 60 seconds. This staggered the start time of the clients for the first 60 seconds of the test. Sun configured an overall test length of 10 minutes. Finally, Sun set the min and max values for the SSL reuse session identifiers to zero. This required the clients to renegotiate a new SSL session for each request and increased the amount of load on the Sun N1400 switch.

### ***Layer 7 throughput test***

For the Layer 7 throughput test, Sun created a custom test suite that sent HTTP 1.0 requests for a 1M byte static object from the clients to a virtual IP address configured on the Sun N1400 switch. For each request, the switch terminated the TCP connection from the client, inspected the Layer 7 contents of the request, and load balanced the request based on the Layer 7 content across a pool of Web servers. The Web server sent its response to the Sun N1400 switch, which returned the data to the client.

Sun used the same test bed configuration and Sun N1400 switch settings defined in the SSL bulk cryptographic throughput test for the Layer 7 throughput test.

Sun created a custom test suite that generated HTTP 1.0 requests for a 1M byte static object located on each Web server. All of the clients sent their requests to the virtual IP address configured on the Sun N1400 switch. The test suite contained one mix that included all clients in the test. Sun set the ramp up time and the delay time to 60 seconds. This staggered the start time of the clients for the first 60 seconds of the test. Sun configured an overall test length of 10 minutes.

### ***Layer 7 connections/second test***

For the Layer 7 connections/second test, Sun created a custom test suite that sent HTTP 1.0 requests for a 1KB static object from the clients to a virtual IP address configured on the Sun N1400 switch. For each request, the switch terminated the TCP connection from the client, inspected the Layer 7 contents of the request, and load balanced the request based on the Layer 7 content across a pool of Web servers. The Web server sent its response to the Sun N1400 switch, which returned the data to the client.

Sun used the same test bed configuration defined in the SSL bulk cryptographic throughput test for the Layer 7 connections/second test. Sun used the same Sun N1400 switch settings defined in the SSL connections/second test for the Layer 7 connections/second test.

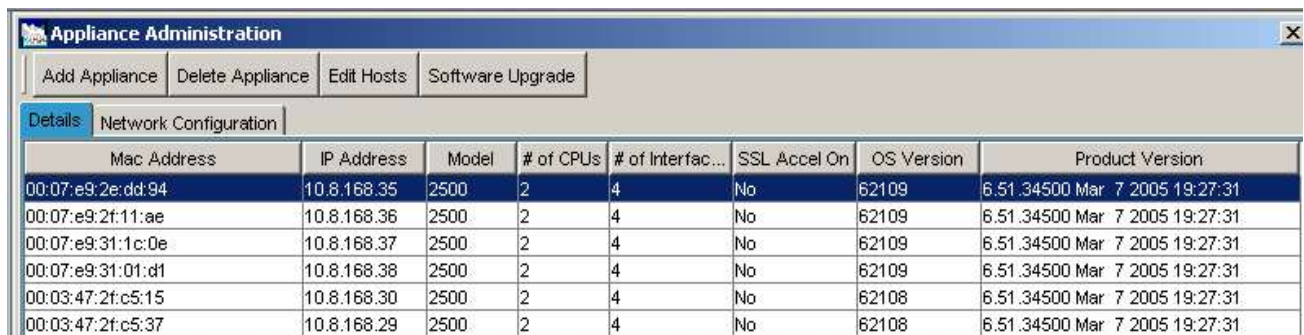
Sun created a custom test suite that generated HTTP 1.0 requests for a 1KB static object located on each Web server. All of the clients sent their requests to the virtual IP address configured on the Sun N1400 switch. The test suite contained one mix that included all clients in the test. Sun set the ramp up time and the delay time to 60 seconds. This staggered the start time of the clients for the first 60 seconds of the test. Sun configured an overall test length of 10 minutes.

## Appendix

### A. Test Hardware and Software

Sun N1400 switch	
Model	N1400
Software version	V3_0A51629
Number of Gigabit ports	4 small form factor fiber Gigabit ports

Figure A1: Sun N1400 switch



The screenshot shows the 'Appliance Administration' window with a 'Network Configuration' tab selected. Below the navigation buttons, there is a table listing appliance details.

Mac Address	IP Address	Model	# of CPUs	# of Interfac...	SSL Accel On	OS Version	Product Version
00:07:e9:2e:dd:94	10.8.168.35	2500	2	4	No	62109	6.51.34500 Mar 7 2005 19:27:31
00:07:e9:2f:11:ae	10.8.168.36	2500	2	4	No	62109	6.51.34500 Mar 7 2005 19:27:31
00:07:e9:31:1c:0e	10.8.168.37	2500	2	4	No	62109	6.51.34500 Mar 7 2005 19:27:31
00:07:e9:31:01:d1	10.8.168.38	2500	2	4	No	62109	6.51.34500 Mar 7 2005 19:27:31
00:03:47:2f:c5:15	10.8.168.30	2500	2	4	No	62108	6.51.34500 Mar 7 2005 19:27:31
00:03:47:2f:c5:37	10.8.168.29	2500	2	4	No	62108	6.51.34500 Mar 7 2005 19:27:31

Figure A2: Avalanche Appliance Administration Settings

## B. Sun N1400 switch configuration files

### B.1 SSL and Layer 7 connections/second test configuration file

```
#####  
# System Name : Sun Secure Application Switch  
# Date : Wed Sep 14 13:59:27 2005  
# Serial No : 00521000121  
# Software Version : V3_0A51629  
#####  
commandModeEntry on  
enable  
configure  
  
#  
# Event configuration and statistics  
#  
event  
#  
# Profile rules for event filters  
#  
filterProfile defaultFile {default filter for saving to file}  
filterProfile defaultFile  
#  
# Profile rules to cause event filtering  
#  
rule 130 drop logLevel warning  
rule 200 send true  
exit;  
  
filterProfile defaultLog {default log filter}  
filterProfile defaultLog  
#  
# Profile rules to cause event filtering  
#  
rule 90 drop logLevel debug  
rule 100 send true  
exit;  
  
filterProfile defaultSyslog {default syslog filter}  
filterProfile defaultSyslog  
#  
# Profile rules to cause event filtering  
#  
rule 100 send true  
exit;  
  
filterProfile defaultTrapd {default trapd filter}  
filterProfile defaultTrapd  
#  
# Profile rules to cause event filtering  
#  
rule 90 drop logLevel warning  
rule 100 send true  
exit;  
  
exit;  
  
#  
# LAG configuration  
#  
lag spirent  
lag spirent linkUpDownTrap disabled  
lag spirent  
#  
# LAG interface configuration  
#  
interface eth.1.1 4 1000  
interface eth.1.1 linkUpDownTrap disabled
```

```

interface eth.1.2 4 1000
interface eth.1.2 linkUpDownTrap disabled
interface eth.1.3 4 1000
interface eth.1.3 linkUpDownTrap disabled
interface eth.1.4 4 1000
interface eth.1.4 linkUpDownTrap disabled
exit;

#
# NMON state and status
#
nmon
exit;

#
# Port configuration
#
port eth.1.1 normal phyDuplex fullDuplex jumboFrames disabled advSpeed 1000M \
  advDuplex fullDuplex defVlan discard
port eth.1.1
exit;

port eth.1.2 normal phyDuplex fullDuplex jumboFrames disabled advSpeed 1000M \
  advDuplex fullDuplex defVlan discard
port eth.1.2
exit;

port eth.1.3 normal phyDuplex fullDuplex jumboFrames disabled advSpeed 1000M \
  advDuplex fullDuplex defVlan discard
port eth.1.3
exit;

port eth.1.4 normal phyDuplex fullDuplex jumboFrames disabled advSpeed 1000M \
  advDuplex fullDuplex defVlan discard
port eth.1.4
exit;

#
# Software key
#
switchServices software key {}

#
# vSwitch configuration
#
vSwitch scale
#
# Host configuration
#
loadBalance host hostWA0 34.6.6.1 vRouter scale:default
loadBalance host hostWA1 34.6.6.11 vRouter scale:default
loadBalance host hostWA2 34.6.6.21 vRouter scale:default
loadBalance host hostWA3 34.6.6.31 vRouter scale:default

#
# Expressions used to classify the application data stream
#
loadBalance objectRule or1 {URI_SUFFIX matches "1"}
loadBalance objectRule orDefault {URI matches ""}

#
# Real service parameters
#
loadBalance realService rsWA0 hostWA0 certType Literal
loadBalance realService rsWA0
exit; exit;

loadBalance realService rsWA1 hostWA1 certType Literal
loadBalance realService rsWA1
exit; exit;

loadBalance realService rsWA2 hostWA2 certType Literal
loadBalance realService rsWA2

```

```

exit; exit;

loadBalance realService rsWA3 hostWA3 certType Literal
loadBalance realService rsWA3
exit; exit;

#
# Request Policies
#
loadBalance requestPolicy rqp1 sorry or1 sorry
loadBalance requestPolicy rqp1
exit; exit;

loadBalance requestPolicy rqpDefault forward orDefault sgWA 2 \
  optimizeLastResponse enabled
loadBalance requestPolicy rqpDefault
exit; exit;

#
# Service group configuration
#
loadBalance serviceGroup sgWA roundRobin {rsWA0; rsWA1; rsWA2; rsWA3} \
  inlineHealthCheck disabled responsePolicyList {}
loadBalance serviceGroup sgWA
exit; exit;

#
# Sorry Data configuration
#
loadBalance sorryData sorry reset
loadBalance sorryData sorry1 reset

#
# Virtual Service configuration
#
loadBalance virtualService vsWA HTTP 11.11.11.11 {rqpDefault; rqp1}
loadBalance virtualService vsWA
  #
  # Virtual service advanced settings
  #
  advanced initParseWithData true
exit; exit;

#
# Port Bandwidth configuration
#
resource portBandwidth eth.1.1 100 100 65534 65535
resource portBandwidth eth.1.2 100 100 65534 65535
resource portBandwidth eth.1.3 100 100 65534 65535
resource portBandwidth eth.1.4 100 100 65534 65535

#
# Service Engine Bandwidth Configuration
#
resource serviceBandwidth functionCard1 20 100

#
# vRouter configuration
#
vRouter default {Default vRouter}
vRouter default
  #
  # VLAN configuration parameters
  #
  vlan 200 200
  vlan 200 linkUpDownTrap disabled
  vlan 200
  #
  # VLAN interface configuration
  #
  interface lag.spirent tagging enabled
  interface lag.spirent linkUpDownTrap disabled

```

```

#
# VLAN STP Interface Configuration
#
interface spanningTree lag.spirent pathCost 3

exit;

#
# Display IP layer configuration
#
ip forwarding enabled
ip
#
# IP Interfaces
#
interface vlan.200
interface vlan.200 linkUpDownTrap disabled

#
# IP Interface Address
#
address vlan.200 34.6.6.101 255.255.0.0

#
# ICMP configuration
#
icmp true true false true

exit;

#
# Interfaces
#
interfaces sock.scale:default linkUpDownTrap disabled
interfaces sock.scale:default/ip.scale:default linkUpDownTrap disabled
interfaces ip.scale:default linkUpDownTrap disabled
interfaces ip.scale:default/vlan.200 linkUpDownTrap disabled
interfaces vlan.200 linkUpDownTrap disabled
interfaces vlan.200/lag.spirent linkUpDownTrap disabled
interfaces loopback linkUpDownTrap disabled

#
# VRRP configuration
#
vrrp
exit;

exit;

vSwitch system {System vSwitch}
vSwitch system
#
# Port Bandwidth configuration
#
resource portBandwidth eth.1.1 100 100 65534 65535
resource portBandwidth eth.1.2 100 100 65534 65535
resource portBandwidth eth.1.3 100 100 65534 65535
resource portBandwidth eth.1.4 100 100 65534 65535

#
# vRouter configuration
#
vRouter management {System Management vRouter}
vRouter management
#
# Display IP layer configuration
#
ip forwarding disabled
ip
#
# IP Interfaces

```

```

#
interface ethMgmt.1
interface ethMgmt.1 linkUpDownTrap disabled

#
# IP Interface Address
#
address ethMgmt.1 10.8.169.140 255.0.0.0

#
# ICMP configuration
#
icmp true true false true

#
# Static route configuration
#
route static 0.0.0.0 0.0.0.0 10.8.169.254 unspecified
exit;

#
# Interfaces
#
interfaces sock.system:management linkUpDownTrap disabled
interfaces sock.system:management/ip.system:management linkUpDownTrap \
disabled
interfaces ip.system:management linkUpDownTrap disabled
interfaces ip.system:management/ethMgmt.1 linkUpDownTrap disabled
interfaces lag.spirent linkUpDownTrap disabled
interfaces lag.spirent/eth.1.1 linkUpDownTrap disabled
interfaces lag.spirent/eth.1.2 linkUpDownTrap disabled
interfaces lag.spirent/eth.1.3 linkUpDownTrap disabled
interfaces lag.spirent/eth.1.4 linkUpDownTrap disabled
interfaces ethMgmt.1 linkUpDownTrap disabled
interfaces loopback linkUpDownTrap disabled

#
# VRRP configuration
#
vrrp
exit;

exit;

vRouter shared {Shared vRouter}
vRouter shared
#
# VLAN configuration parameters
#
vlan 100 100
vlan 100 linkUpDownTrap disabled
vlan 100
#
# VLAN interface configuration
#
interface lag.spirent tagging enabled
interface lag.spirent linkUpDownTrap disabled

#
# VLAN STP Interface Configuration
#
interface spanningTree lag.spirent pathCost 3

exit;

#
# Display IP layer configuration
#
ip forwarding enabled
ip
#
# IP Interfaces
#

```

```

interface vlan.100
interface vlan.100 linkUpDownTrap disabled

#
# IP Interface Address
#
address vlan.100 11.11.11.1 255.0.0.0

#
# ICMP configuration
#
icmp true true false true

exit;

#
# Interfaces
#
interfaces sock.system:shared linkUpDownTrap disabled
interfaces sock.system:shared/ip.system:shared linkUpDownTrap disabled
interfaces ip.system:shared linkUpDownTrap disabled
interfaces ip.system:shared/vlan.100 linkUpDownTrap disabled
interfaces vlan.100 linkUpDownTrap disabled
interfaces vlan.100/lag.spirent linkUpDownTrap disabled
interfaces loopback linkUpDownTrap disabled

#
# VRRP configuration
#
vrrp
exit;

exit;

#
# CLI configuration
#
switchServices cli defaultHistoryLength 200 defaultAutoLogoutTimeout 0

#
# HTTP configuration and status
#
switchServices httpd enabled
switchServices httpd
exit; exit;

#
# Global NTP configuration parameters
#
switchServices ntp
#
# Global NTP configuration parameters (advanced)
#
advanced
exit;

exit; exit;

#
# SNMP configuration
#
switchServices snmp
exit; exit;

#
# SSHd configuration and operation
#
switchServices sshd confEncryption {des3Cbc; blowfishCbc; des} confHmac \
{md5; sha1; md5b96; sha1b96} userAuthentication {publicKey; password}
switchServices sshd
#

```

```
# SSHd configuration and operation (advanced)
#
advanced
exit;

exit; exit;

#
# Telnetd configuration and current status
#
switchServices telnetd enabled
switchServices telnetd
exit; exit;

#
# TFTPd configuration and session statistics
#
switchServices tftpd
exit; exit;

#
# TideRunner Configuration
#
switchServices tideRunner initkeys functionCard1 20000 statPollPeriod 5 \
  smmPageSize 2 dleMaxHdrLen 8192

#
# Configuration and status for the trap process
#
switchServices trap
exit; exit;
```

## B.2 SSL bulk cryptographic and Layer 7 throughput test configuration file

```
#####
# System Name      : Sun Secure Application Switch
# Date            : Thu Sep 15 09:56:44 2005
# Serial No       : 00521000121
# Software Version : V3_OA51629
#####
commandModeEntry on
enable
configure

#
# Ethernet management port
#
ethMgmt phySpeed auto phyDuplex halfDuplex adminMac 00:00:00:00:00:00

#
# Event configuration and statistics
#
event fileLogSize 65536 fileLogFilter defaultFile fileLogName eventlog.txt \
  logFilter defaultLog
event
#
# Profile rules for event filters
#
filterProfile name defaultFile description \
  {default filter for saving to file}
filterProfile name defaultFile
#
# Profile rules to cause event filtering
#
rule position 130 action drop all false vSwitchAndVRouter <All> \
  logLevel warning
rule position 200 action send all true vSwitchAndVRouter <All> \
  logLevel All
exit;

filterProfile name defaultLog description {default log filter}
filterProfile name defaultLog
#
# Profile rules to cause event filtering
#
rule position 90 action drop all false vSwitchAndVRouter <All> \
  logLevel debug
rule position 100 action send all true vSwitchAndVRouter <All> \
  logLevel All
exit;

filterProfile name defaultSyslog description {default syslog filter}
filterProfile name defaultSyslog
#
# Profile rules to cause event filtering
#
rule position 100 action send all true vSwitchAndVRouter <All> \
  logLevel All
exit;

filterProfile name defaultTrapd description {default trapd filter}
filterProfile name defaultTrapd
#
# Profile rules to cause event filtering
#
rule position 90 action drop all false vSwitchAndVRouter <All> \
  logLevel warning
rule position 100 action send all true vSwitchAndVRouter <All> \
  logLevel All
exit;

#
# Syslog host configuration
```

```

#
syslog host 10.8.168.191 port 10002 filter defaultSyslog facility local0
exit;

#
# LAG configuration
#
lag lagName spirent jumboFrames disabled defVlan discard
lag lagName spirent adminState enabled eventFilter informational \
  linkUpDownTrap disabled packetTrace disabled description {}
lag lagName spirent
#
# LAG interface configuration
#
interface ifName eth.1.1 floodPref 4 weight 1000
interface ifName eth.1.1 adminState enabled eventFilter informational \
  linkUpDownTrap disabled mtu 1500 packetTrace disabled description {}
interface ifName eth.1.2 floodPref 4 weight 1000
interface ifName eth.1.2 adminState enabled eventFilter informational \
  linkUpDownTrap disabled mtu 1500 packetTrace disabled description {}
interface ifName eth.1.3 floodPref 4 weight 1000
interface ifName eth.1.3 adminState enabled eventFilter informational \
  linkUpDownTrap disabled mtu 1500 packetTrace disabled description {}
interface ifName eth.1.4 floodPref 4 weight 1000
interface ifName eth.1.4 adminState enabled eventFilter informational \
  linkUpDownTrap disabled mtu 1500 packetTrace disabled description {}
exit;

#
# NMON state and status
#
nmon adminState disabled
nmon
exit;

#
# Port configuration
#
port ifName eth.1.1 phyMode normal phyDuplex fullDuplex jumboFrames disabled \
  advSpeed 1000M advDuplex fullDuplex defVlan discard
port ifName eth.1.1
exit;

port ifName eth.1.2 phyMode normal phyDuplex fullDuplex jumboFrames disabled \
  advSpeed 1000M advDuplex fullDuplex defVlan discard
port ifName eth.1.2
exit;

port ifName eth.1.3 phyMode normal phyDuplex fullDuplex jumboFrames disabled \
  advSpeed 1000M advDuplex fullDuplex defVlan discard
port ifName eth.1.3
exit;

port ifName eth.1.4 phyMode normal phyDuplex fullDuplex jumboFrames disabled \
  advSpeed 1000M advDuplex fullDuplex defVlan discard
port ifName eth.1.4
exit;

#
# Software key
#
switchServices software key softwareKey {}

#
# vSwitch configuration
#
vSwitch vSwitchName scale adminState enabled
vSwitch vSwitchName scale
#
# Host configuration
#
loadBalance host name hostWA0 ipAddress 34.6.6.1 adminState enabled \

```

```

vRouter scale:default
loadBalance host name hostWA1 ipAddress 34.6.6.11 adminState enabled \
vRouter scale:default
loadBalance host name hostWA2 ipAddress 34.6.6.21 adminState enabled \
vRouter scale:default
loadBalance host name hostWA3 ipAddress 34.6.6.31 adminState enabled \
vRouter scale:default

#
# Expressions used to classify the application data stream
#
loadBalance objectRule name or1 predicate {URI_SUFFIX matches "1"}
loadBalance objectRule name orDefault predicate {URI matches "*"}

#
# Real service parameters
#
loadBalance realService name rsWA0 hostName hostWA0 protocol TCP port 80 \
weight 1 adminState enabled disableDelay 0 ilSHCFailureRateThreshold 1 \
clientAddressTranslationMask 255.255.255.255 bridgeMode disabled \
encryption unencrypted certType Literal sslProto {SSLv3; TLSv1} \
sslCiphers \
{RSA_WITH_AES_256_CBC_SHA; RSA_WITH_RC4_128_MD5; RSA_WITH_RC4_128_SHA; RSA_WITH_AES_128_CBC_SHA;
RSA_WITH_3DES_EDE_CBC_SHA} \
reneg true resume true
loadBalance realService name rsWA0
#
# Real service advanced settings
#
advanced tcbTemplateKey 0 ipTos Normal xmtRetryLimit 4 estRetryLimit 4 \
shortRxTimer 16_seconds longRxTimer 64_seconds rcvWnd 65535 xmtRTT \
750_msec smmStreamLimit 4xRcvWnd estShortTimeout ExpRetr \
rcvWndDisabled false rcvMss 1460 xmtMss 1460 enableHttpMode false \
httpGetPiggyBack true rxUseLongTime true
exit; exit;

loadBalance realService name rsWA1 hostName hostWA1 protocol TCP port 80 \
weight 1 adminState enabled disableDelay 0 ilSHCFailureRateThreshold 1 \
clientAddressTranslationMask 255.255.255.255 bridgeMode disabled \
encryption unencrypted certType Literal sslProto {SSLv3; TLSv1} \
sslCiphers \
{RSA_WITH_AES_256_CBC_SHA; RSA_WITH_RC4_128_MD5; RSA_WITH_RC4_128_SHA; RSA_WITH_AES_128_CBC_SHA;
RSA_WITH_3DES_EDE_CBC_SHA} \
reneg true resume true
loadBalance realService name rsWA1
#
# Real service advanced settings
#
advanced tcbTemplateKey 0 ipTos Normal xmtRetryLimit 4 estRetryLimit 4 \
shortRxTimer 16_seconds longRxTimer 64_seconds rcvWnd 65535 xmtRTT \
750_msec smmStreamLimit 4xRcvWnd estShortTimeout ExpRetr \
rcvWndDisabled false rcvMss 1460 xmtMss 1460 enableHttpMode false \
httpGetPiggyBack true rxUseLongTime true
exit; exit;

loadBalance realService name rsWA2 hostName hostWA2 protocol TCP port 80 \
weight 1 adminState enabled disableDelay 0 ilSHCFailureRateThreshold 1 \
clientAddressTranslationMask 255.255.255.255 bridgeMode disabled \
encryption unencrypted certType Literal sslProto {SSLv3; TLSv1} \
sslCiphers \
{RSA_WITH_AES_256_CBC_SHA; RSA_WITH_RC4_128_MD5; RSA_WITH_RC4_128_SHA; RSA_WITH_AES_128_CBC_SHA;
RSA_WITH_3DES_EDE_CBC_SHA} \
reneg true resume true
loadBalance realService name rsWA2
#
# Real service advanced settings
#
advanced tcbTemplateKey 0 ipTos Normal xmtRetryLimit 4 estRetryLimit 4 \
shortRxTimer 16_seconds longRxTimer 64_seconds rcvWnd 65535 xmtRTT \
750_msec smmStreamLimit 4xRcvWnd estShortTimeout ExpRetr \
rcvWndDisabled false rcvMss 1460 xmtMss 1460 enableHttpMode false \
httpGetPiggyBack true rxUseLongTime true

```

```

exit; exit;

loadBalance realService name rsWA3 hostName hostWA3 protocol TCP port 80 \
  weight 1 adminState enabled disableDelay 0 ilSHCFailureRateThreshold 1 \
  clientAddressTranslationMask 255.255.255.255 bridgeMode disabled \
  encryption unencrypted certType Literal sslProto {SSLv3; TLSv1} \
  sslCiphers \
  {RSA_WITH_AES_256_CBC_SHA; RSA_WITH_RC4_128_MD5; RSA_WITH_RC4_128_SHA; RSA_WITH_AES_128_CBC_SHA;
RSA_WITH_3DES_EDE_CBC_SHA} \
  renegotiate true resume true
loadBalance realService name rsWA3
#
# Real service advanced settings
#
advanced tcbTemplateKey 0 ipTos Normal xmtRetryLimit 4 estRetryLimit 4 \
  shortRxTimer 16_seconds longRxTimer 64_seconds rcvWnd 65535 xmtRTT \
  750_msec smmStreamLimit 4xRcvWnd estShortTimeout ExpRetr \
  rcvWndDisabled false rcvMss 1460 xmtMss 1460 enableHttpMode false \
  httpGetPiggyBack true rxUseLongTime true
exit; exit;

#
# Request Policies
#
loadBalance requestPolicy name rqpl action sorry objectRule orl sorryData \
  sorry precedence 1
loadBalance requestPolicy name rqpl
exit; exit;

loadBalance requestPolicy name rqpDefault action forward objectRule \
  orDefault serviceGroupName sgWA precedence 2 persistType none \
  srcAddressMask 255.255.255.255 optimizeLastResponse enabled \
  usePooledConnections disabled firstObjectSwitching enabled
loadBalance requestPolicy name rqpDefault
exit; exit;

#
# Service group configuration
#
loadBalance serviceGroup name sgWA loadBalanceType roundRobin \
  cfgRealServices {rsWA0; rsWA1; rsWA2; rsWA3} standbyRSActivation \
  asNeeded adminState enabled healthName {} inlineHealthCheck disabled \
  retryCount 1 flashCrowdThreshold none
loadBalance serviceGroup name sgWA
exit; exit;

#
# Sorry Data configuration
#
loadBalance sorryData name sorry action reset

#
# Virtual Service configuration
#
loadBalance virtualService name vsWA appServiceType HTTPS ipAddress \
  11.11.11.11 requestPolicyList {rqpDefault; rqpl} adminState enabled \
  disableDelay 0 port 443 vRouter system:shared clientSrcIPRange \
  0.0.0.0-255.255.255.255 synRateLimit unlimited ckmKeyName angrykey \
  sslProto {SSLv3; TLSv1} ieExportCiphersSupport disabled sslCiphers \
  {RSA_WITH_AES_256_CBC_SHA; RSA_WITH_RC4_128_MD5; RSA_WITH_RC4_128_SHA; RSA_WITH_AES_128_CBC_SHA;
RSA_WITH_3DES_EDE_CBC_SHA} \
  renegotiate true sgcSupport false resume false
loadBalance virtualService name vsWA
#
# Virtual service advanced settings
#
advanced tcbTemplateKey 0 ipTos Normal xmtRetryLimit 4 estRetryLimit 4 \
  shortRxTimer 32_seconds longRxTimer 64_seconds rcvWnd 65535 xmtRTT \
  1500_msec smmStreamLimit 4xRcvWnd estShortTimeout ExpRetr \
  rcvWndDisabled false rcvMss 1460 xmtMss 1460 enableHttpMode false \
  initParseWithData true rxUseLongTime false disableSynCookies true \
  clientFirstProtocol false

```

```

exit; exit;

#
# Port Bandwidth configuration
#
resource portBandwidth ifName eth.1.1 bandwidthAllocation 100 \
  bandwidthMaximum 100 burstSize 65534 burstSizeMaximum 65535
resource portBandwidth ifName eth.1.2 bandwidthAllocation 100 \
  bandwidthMaximum 100 burstSize 65534 burstSizeMaximum 65535
resource portBandwidth ifName eth.1.3 bandwidthAllocation 100 \
  bandwidthMaximum 100 burstSize 65534 burstSizeMaximum 65535
resource portBandwidth ifName eth.1.4 bandwidthAllocation 100 \
  bandwidthMaximum 100 burstSize 65534 burstSizeMaximum 65535

#
# Service Engine Bandwidth Configuration
#
resource serviceBandwidth card functionCard1 guaranteedMinPercent 20 \
  absoluteMaxPercent 100

#
# vRouter configuration
#
vRouter name default description {Default vRouter} adminState enabled
vRouter name default
#
# VLAN configuration parameters
#
vlan vlanName 200 vlanId 200 learning enabled bridgeModeLoadBalancing \
  disabled
vlan vlanName 200 adminState enabled linkUpDownTrap disabled \
  eventFilter informational packetTrace disabled description {}
vlan vlanName 200
#
# VLAN interface configuration
#
interface ifName lag.spirent tagging enabled
interface ifName lag.spirent adminState enabled mtu 1500 \
  packetTrace disabled eventFilter informational linkUpDownTrap \
  disabled description {}

#
# VLAN STP Interface Configuration
#
interface spanningTree ifName lag.spirent priority 128 adminState \
  enabled portfast disabled pathCost 3 rootGuard disabled \
  bpduGuard disabled

#
# VLAN STP configuration
#
spanningTree adminState disabled priority 32768 bridgeMaxAge 20 \
  bridgeHelloTime 2 bridgeForwardDelay 15
exit;

#
# Display IP layer configuration
#
ip ttl 64 forwarding enabled adminState enabled eventFilter \
  informational linkUpDownTrap enabled packetTrace disabled \
  description {}
ip
#
# IP Interfaces
#
interface IfName vlan.200
interface IfName vlan.200 adminState enabled eventFilter \
  informational linkUpDownTrap disabled mtu 1500 packetTrace \
  disabled description {}

#
# ARP configuration

```

```

#
arp settings agingTimeout 300 retryCount 2 requestTimeout 2 \
  checkForDuplicateMacs enabled

#
# IP Interface Address
#
address IfName vlan.200 ipAddr 34.6.6.101 netMask 255.255.0.0 \
  vsrpRedirect disabled managedVRouter {}

#
# Display IP Echo Responder configuration and statistics
#
echoResponder tcpEchoResponderAdminState disabled \
  tcpEchoResponderReceivePort 7 udpEchoResponderAdminState \
  disabled udpEchoResponderReceivePort 7

#
# ICMP configuration
#
icmp replyToEchos true sendTimeExceeds true sendParamProbs false \
  replyToMasks true

exit;

#
# Interfaces
#
interfaces connectionName sock.scale:default adminState enabled \
  linkUpDownTrap disabled eventFilter informational packetTrace \
  disabled description {} mtu 1500
interfaces connectionName sock.scale:default/ip.scale:default \
  adminState enabled linkUpDownTrap disabled eventFilter informational \
  packetTrace disabled description {} mtu 1500
interfaces connectionName ip.scale:default adminState enabled \
  linkUpDownTrap disabled eventFilter informational packetTrace \
  disabled description {} mtu 1500
interfaces connectionName ip.scale:default/vlan.200 adminState enabled \
  linkUpDownTrap disabled eventFilter informational packetTrace \
  disabled description {} mtu 1500
interfaces connectionName vlan.200 adminState enabled linkUpDownTrap \
  disabled eventFilter informational packetTrace disabled description \
  {} mtu 1500
interfaces connectionName vlan.200/lag.spirent adminState enabled \
  linkUpDownTrap disabled eventFilter informational packetTrace \
  disabled description {} mtu 1500
interfaces connectionName loopback adminState enabled linkUpDownTrap \
  disabled eventFilter informational packetTrace disabled description \
  {} mtu 1500

#
# OSPF advertise ASE routes
#
ospf advertise-ase directRt enabled staticRt disabled ripRt disabled \
  directMetric 1 staticMetric 1 ripMetric 1 directType type1 \
  staticType type1 ripType type1

#
# OSPF advertise NSSA routes
#
ospf advertise-nssa directRt enabled staticRt disabled ripRt disabled \
  directMetric 1 staticMetric 1 ripMetric 1 directType type1 \
  staticType type1 ripType type1

#
# OSPF global settings
#
ospf globalSettings adminState enabled routerId 0.0.0.0 \
  rfc1583Compatibility enabled

#
# RIP Advertise

```

```

#
rip advertise direct enabled staticRt disabled ospf disabled ospfAse \
  disabled directMetric 1 staticRtMetric 1 ospfMetric 1 ospfAseMetric \
  1

#
# RIP global settings
#
rip globalSettings adminState enabled expireTime 180 updateTime 30

#
# VRRP configuration
#
vrrp traps disabled vsrpPreference 0
vrrp
exit;

exit;

#
# VSRP vSwitch configuration and status
#
redundancy vsrp electionPreference 100
exit;

vSwitch vSwitchName system description {System vSwitch} adminState enabled
vSwitch vSwitchName system
#
# Port Bandwidth configuration
#
resource portBandwidth ifName eth.1.1 bandwidthAllocation 100 \
  bandwidthMaximum 100 burstSize 65534 burstSizeMaximum 65535
resource portBandwidth ifName eth.1.2 bandwidthAllocation 100 \
  bandwidthMaximum 100 burstSize 65534 burstSizeMaximum 65535
resource portBandwidth ifName eth.1.3 bandwidthAllocation 100 \
  bandwidthMaximum 100 burstSize 65534 burstSizeMaximum 65535
resource portBandwidth ifName eth.1.4 bandwidthAllocation 100 \
  bandwidthMaximum 100 burstSize 65534 burstSizeMaximum 65535

#
# vRouter configuration
#
vRouter name management description {System Management vRouter} adminState \
  enabled
vRouter name management
#
# Display IP layer configuration
#
ip ttl 64 forwarding disabled adminState enabled eventFilter \
  informational linkUpDownTrap enabled packetTrace disabled \
  description {}
ip
#
# IP Interfaces
#
interface IfName ethMgmt.1
interface IfName ethMgmt.1 adminState enabled eventFilter \
  informational linkUpDownTrap disabled mtu 1500 packetTrace \
  disabled description {}

#
# ARP configuration
#
arp settings agingTimeout 300 retryCount 2 requestTimeout 2 \
  checkForDuplicateMacs enabled

#
# IP Interface Address
#
address IfName ethMgmt.1 ipAddr 10.8.169.140 netMask 255.0.0.0 \
  vsrpRedirect disabled managedVRouter {}

```

```

#
# Display IP Echo Responder configuration and statistics
#
echoResponder tcpEchoResponderAdminState disabled \
  tcpEchoResponderReceivePort 7 udpEchoResponderAdminState \
  disabled udpEchoResponderReceivePort 7

#
# ICMP configuration
#
icmp replyToEchos true sendTimeExceeds true sendParamProbs false \
  replyToMasks true

#
# Static route configuration
#
route static destAddr 0.0.0.0 mask 0.0.0.0 nextHop 10.8.169.254 \
  ifName unspecified preference low metric 1
exit;

#
# Interfaces
#
interfaces connectionName sock.system:management adminState enabled \
  linkUpDownTrap disabled eventFilter informational packetTrace \
  disabled description {} mtu 1500
interfaces connectionName sock.system:management/ip.system:management \
  adminState enabled linkUpDownTrap disabled eventFilter informational \
  packetTrace disabled description {} mtu 1500
interfaces connectionName ip.system:management adminState enabled \
  linkUpDownTrap disabled eventFilter informational packetTrace \
  disabled description {} mtu 1500
interfaces connectionName ip.system:management/ethMgmt.1 adminState \
  enabled linkUpDownTrap disabled eventFilter informational \
  packetTrace disabled description {} mtu 1500
interfaces connectionName lag.spirent adminState enabled \
  linkUpDownTrap disabled eventFilter informational packetTrace \
  disabled description {} mtu 1500
interfaces connectionName lag.spirent/eth.1.1 adminState enabled \
  linkUpDownTrap disabled eventFilter informational packetTrace \
  disabled description {} mtu 1500
interfaces connectionName lag.spirent/eth.1.2 adminState enabled \
  linkUpDownTrap disabled eventFilter informational packetTrace \
  disabled description {} mtu 1500
interfaces connectionName lag.spirent/eth.1.3 adminState enabled \
  linkUpDownTrap disabled eventFilter informational packetTrace \
  disabled description {} mtu 1500
interfaces connectionName lag.spirent/eth.1.4 adminState enabled \
  linkUpDownTrap disabled eventFilter informational packetTrace \
  disabled description {} mtu 1500
interfaces connectionName ethMgmt.1 adminState enabled linkUpDownTrap \
  disabled eventFilter informational packetTrace disabled description \
  {} mtu 1500
interfaces connectionName eth.1.1 adminState enabled linkUpDownTrap \
  enabled eventFilter informational packetTrace disabled description \
  {} mtu 1500
interfaces connectionName eth.1.2 adminState enabled linkUpDownTrap \
  enabled eventFilter informational packetTrace disabled description \
  {} mtu 1500
interfaces connectionName eth.1.3 adminState enabled linkUpDownTrap \
  enabled eventFilter informational packetTrace disabled description \
  {} mtu 1500
interfaces connectionName eth.1.4 adminState enabled linkUpDownTrap \
  enabled eventFilter informational packetTrace disabled description \
  {} mtu 1500
interfaces connectionName loopback adminState enabled linkUpDownTrap \
  disabled eventFilter informational packetTrace disabled description \
  {} mtu 1500

#
# OSPF advertise ASE routes
#

```

```

ospf advertise-ase directRt enabled staticRt disabled ripRt disabled \
  directMetric 1 staticMetric 1 ripMetric 1 directType type1 \
  staticType type1 ripType type1

#
# OSPF advertise NSSA routes
#
ospf advertise-nssa directRt enabled staticRt disabled ripRt disabled \
  directMetric 1 staticMetric 1 ripMetric 1 directType type1 \
  staticType type1 ripType type1

#
# OSPF global settings
#
ospf globalSettings adminState enabled routerId 0.0.0.0 \
  rfc1583Compatibility enabled

#
# RIP Advertise
#
rip advertise direct enabled staticRt disabled ospf disabled ospfAse \
  disabled directMetric 1 staticRtMetric 1 ospfMetric 1 ospfAseMetric \
  1

#
# RIP global settings
#
rip globalSettings adminState enabled expireTime 180 updateTime 30

#
# VRRP configuration
#
vrrp traps disabled vsrpPreference 0
vrrp
exit;

exit;

vRouter name shared description {Shared vRouter} adminState enabled
vRouter name shared
#
# VLAN configuration parameters
#
vlan vlanName 100 vlanId 100 learning enabled bridgeModeLoadBalancing \
  disabled
vlan vlanName 100 adminState enabled linkUpDownTrap disabled \
  eventFilter informational packetTrace disabled description {}
vlan vlanName 100
#
# VLAN interface configuration
#
interface ifName lag.spirent tagging enabled
interface ifName lag.spirent adminState enabled mtu 1500 \
  packetTrace disabled eventFilter informational linkUpDownTrap \
  disabled description {}

#
# VLAN STP Interface Configuration
#
interface spanningTree ifName lag.spirent priority 128 adminState \
  enabled portfast disabled pathCost 3 rootGuard disabled \
  bpduGuard disabled

#
# VLAN STP configuration
#
spanningTree adminState disabled priority 32768 bridgeMaxAge 20 \
  bridgeHelloTime 2 bridgeForwardDelay 15
exit;

#
# Display IP layer configuration

```

```

#
ip ttl 64 forwarding enabled adminState enabled eventFilter \
informational linkUpDownTrap enabled packetTrace disabled \
description {}
ip
#
# IP Interfaces
#
interface IfName vlan.100
interface IfName vlan.100 adminState enabled eventFilter \
informational linkUpDownTrap disabled mtu 1500 packetTrace \
disabled description {}

#
# ARP configuration
#
arp settings agingTimeout 300 retryCount 2 requestTimeout 2 \
checkForDuplicateMacs enabled

#
# IP Interface Address
#
address IfName vlan.100 ipAddr 11.11.11.1 netMask 255.0.0.0 \
vsrpRedirect disabled managedVRouter {}

#
# Display IP Echo Responder configuration and statistics
#
echoResponder tcpEchoResponderAdminState disabled \
tcpEchoResponderReceivePort 7 udpEchoResponderAdminState \
disabled udpEchoResponderReceivePort 7

#
# ICMP configuration
#
icmp replyToEchos true sendTimeExceeds true sendParamProbs false \
replyToMasks true

exit;

#
# Interfaces
#
interfaces connectionName sock.system:shared adminState enabled \
linkUpDownTrap disabled eventFilter informational packetTrace \
disabled description {} mtu 1500
interfaces connectionName sock.system:shared/ip.system:shared \
adminState enabled linkUpDownTrap disabled eventFilter informational \
packetTrace disabled description {} mtu 1500
interfaces connectionName ip.system:shared adminState enabled \
linkUpDownTrap disabled eventFilter informational packetTrace \
disabled description {} mtu 1500
interfaces connectionName ip.system:shared/vlan.100 adminState enabled \
linkUpDownTrap disabled eventFilter informational packetTrace \
disabled description {} mtu 1500
interfaces connectionName vlan.100 adminState enabled linkUpDownTrap \
disabled eventFilter informational packetTrace disabled description \
{} mtu 1500
interfaces connectionName vlan.100/lag.spirent adminState enabled \
linkUpDownTrap disabled eventFilter informational packetTrace \
disabled description {} mtu 1500
interfaces connectionName loopback adminState enabled linkUpDownTrap \
disabled eventFilter informational packetTrace disabled description \
{} mtu 1500

#
# OSPF advertise ASE routes
#
ospf advertise-ase directRt enabled staticRt disabled ripRt disabled \
directMetric 1 staticMetric 1 ripMetric 1 directType type1 \
staticType type1 ripType type1

```

```

#
# OSPF advertise NSSA routes
#
ospf advertise-nssa directRt enabled staticRt disabled ripRt disabled \
  directMetric 1 staticMetric 1 ripMetric 1 directType type1 \
  staticType type1 ripType type1

#
# OSPF global settings
#
ospf globalSettings adminState enabled routerId 0.0.0.0 \
  rfc1583Compatibility enabled

#
# RIP Advertise
#
rip advertise direct enabled staticRt disabled ospf disabled ospfAse \
  disabled directMetric 1 staticRtMetric 1 ospfMetric 1 ospfAseMetric \
  1

#
# RIP global settings
#
rip globalSettings adminState enabled expireTime 180 updateTime 30

#
# VRRP configuration
#
vrrp traps disabled vsrpPreference 0
vrrp
exit;

exit;

exit;

#
# Configuration and status of the 'locator LED'
#
switchServices chassis locatorLED lampMode off

#
# CLI configuration
#
switchServices cli prompt sun lineWrap 79 historyLength 50 cliEventLevel none \
  defaultRows 24 defaultEcho disabled defaultPrompt sun defaultHistoryLength \
  200 defaultAutoLogoutTimeout 0 defaultMessageOnAutoLogout enabled \
  defaultCliEventLevel none auditLogging on

#
# HTTP configuration and status
#
switchServices httpd adminState enabled accessMode http httpPort 80 httpsPort \
  443 sessionTimeout 10 auditLogging on
switchServices httpd
exit; exit;

#
# Global NTP configuration parameters
#
switchServices ntp adminState disabled
switchServices ntp
#
# Global NTP configuration parameters (advanced)
#
advanced clockMinStep 200 rtcUpdateInterval 10
advanced
exit;

exit; exit;

#

```

```

# Check Certificate (Date) Parameters
#
switchServices security ckm expiredCerts allow

#
# SNMP configuration
#
switchServices snmp adminState disabled snmpPort 161 auditLogging on
switchServices snmp
#
# SNMP agent configuration
#
systemInfo contact {} name {Sun Secure Application Switch} location {}

exit; exit;

#
# SSHd configuration and operation
#
switchServices sshd adminState disabled maxSessions 4 idleTimeout 600 \
  confEncryption {des3Cbc; blowfishCbc; des} confHmac \
  {md5; sha1; md5b96; shal96} userAuthentication {publicKey; password} \
  hostAuthentication none
switchServices sshd
#
# SSHd configuration and operation (advanced)
#
advanced sshdPort 22 maximumAuthenticationTime 30 globalEventInterval 10 \
  handleErrors enabled executionMonitorMode active clientKeepAliveInterval \
  0 patchVendorIds sshcomFsecureVandyke cliInheritsSshLoginCredentials \
  disabled
advanced
exit;

exit; exit;

#
# Telnetd configuration and current status
#
switchServices telnetd adminState enabled maxSessions 10 telnetdPort 23 \
  rcvBufSize 4000
switchServices telnetd
exit; exit;

#
# TFTPd configuration and session statistics
#
switchServices tftpd adminState disabled maxSessions 10 tftpdPort 69
switchServices tftpd
exit; exit;

#
# TideRunner Configuration
#
switchServices tideRunner initkeys moduleId functionCard1 perfPciHoldoff 20000 \
  tcbTemplateMax 4096 statPollPeriod 5 smmPageSize 8 dleMaxHdrLen 8192

#
# Configuration and status for the trap process
#
switchServices trap adminState disabled systemEvtTraps disabled \
  authenticationFailureTraps disabled
switchServices trap
exit; exit;

```

VeriTest ([www.veritest.com](http://www.veritest.com)), the testing division of Lionbridge Technologies, Inc., provides outsourced testing solutions that maximize revenue and reduce costs for our clients. For companies who use high-tech products as well as those who produce them, smoothly functioning technology is essential to business success. VeriTest helps our clients identify and correct technology problems in their products and in their line of business applications by providing the widest range of testing services available.

VeriTest created the suite of industry-standard benchmark software that includes WebBench, NetBench, Winstone, and WinBench. We've distributed over 20 million copies of these tools, which are in use at every one of the 2001 Fortune 100 companies. Our Internet BenchMark service provides the definitive ratings for Internet Service Providers in the US, Canada, and the UK.

Under our former names of ZD Labs and eTesting Labs, and as part of VeriTest since July of 2002, we have delivered rigorous, objective, independent testing and analysis for over a decade. With the most knowledgeable staff in the business, testing facilities around the world, and almost 1,600 dedicated network PCs, VeriTest offers our clients the expertise and equipment necessary to meet all their testing needs.

**For more information** email us at [info@veritest.com](mailto:info@veritest.com) or call us at 919-380-2800.

#### **Disclaimer of Warranties; Limitation of Liability:**

VERITEST HAS MADE REASONABLE EFFORTS TO ENSURE THE ACCURACY AND VALIDITY OF ITS TESTING, HOWEVER, VERITEST SPECIFICALLY DISCLAIMS ANY WARRANTY, EXPRESSED OR IMPLIED, RELATING TO THE TEST RESULTS AND ANALYSIS, THEIR ACCURACY, COMPLETENESS OR QUALITY, INCLUDING ANY IMPLIED WARRANTY OF FITNESS FOR ANY PARTICULAR PURPOSE. ALL PERSONS OR ENTITIES RELYING ON THE RESULTS OF ANY TESTING DO SO AT THEIR OWN RISK, AND AGREE THAT VERITEST, ITS EMPLOYEES AND ITS SUBCONTRACTORS SHALL HAVE NO LIABILITY WHATSOEVER FROM ANY CLAIM OF LOSS OR DAMAGE ON ACCOUNT OF ANY ALLEGED ERROR OR DEFECT IN ANY TESTING PROCEDURE OR RESULT.

IN NO EVENT SHALL VERITEST BE LIABLE FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH ITS TESTING, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL VERITEST'S LIABILITY, INCLUDING FOR DIRECT DAMAGES, EXCEED THE AMOUNTS PAID IN CONNECTION WITH VERITEST'S TESTING. CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES ARE AS SET FORTH HEREIN.