Best Practices for IP Workload Distribution in an IBM zSeries Server Environment

-

An IBM and Cisco Interoperability Study of z/OS Sysplex Distributor with Cisco Multi Node Load Balancing (MNLB), Cisco Content Services Switch (CSS), Cisco Content Switching Module (CSM), and the z/OS Load Balancing Advisor

Document Version 2.0

Authors:

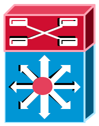
Jay Aiken, IBM – jaaiken@us.ibm.com
Dyan Gray Collins, Cisco – dpgray@cisco.com
Derek Huckaby, Cisco – dhuckaby@cisco.com
Scott Hodgdon, Cisco – hodgdon@cisco.com
Stefano Testa, Cisco – testas@cisco.com
Mike Law, IBM – mslaw@us.ibm.com
David Rainey, IBM – darainey@us.ibm.com
Alfred B Christensen, IBM – alfredch@us.ibm.com
Gus Kassimis , IBM – kassimis @us.ibm.com

Table of Contents

Terms of Use (Glossary):

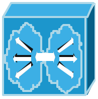
ASR	Adaptive Session Redundancy
AWM	IBM Application Workload Modeler
CF	Coupling Facility
CSM	Content Switching Module
CSS	Content Services Switch
DVIPA	Dynamic Virtual IP Address
FTP	File Transfer Protocol
GBIC	Gigabit Interface Converter
HSRP	Hot Standby Routing Protocol
HTTP	HyperText Transfer Protocol
ICF	Integrated Coupling Facility
IOM	Interface Module
LPARs	Logical Partitions
MNLB	Multi Node Load Balancing
MSFC2	Multilayer Switching Feature Card 2
MSFC3	Multilayer Switching Feature Card 3
NAT	Network Address Translation
OSA	Open Systems Adapter
PAT	Port Address Translation
PBR	Policy Based Routing
PFC	Policy Feature Card
SASP	Server/Application State Protocol
SD	Sysplex Distributor
SCM	Switch Control Module
SSLM	SSL Services Module
SSL	Secure Sockets Layer
SUP720	Multilayer Switching Feature Card 3
TLS	Transport Layer Security
VIPA	Virtual IP Address
VIP	Virtual IP Address
WLM	Workload Manager
XCF	Cross Coupling Facility

Networking Symbols Used in Diagrams



Cisco Catalyst 6500

Router or L3 switch (both provide IP routing features)



Content Switching Hardware

zSeries Server

1 Introduction

1.1 Customer Environment

Customers require highly available server systems to provide mission-critical e-business application support for their businesses and their clients. Supporting availability and scalability requirements generally means exploiting clusters of server hosts, as well as a highly available network infrastructure. IBM's zSeries® servers and Parallel Sysplex provide industry-leading clustering for mixed-workload e-business applications and Cisco System® content switching products provide unparalleled availability and network scalability. IBM and Cisco are working together to meet our customers' availability and scalability requirements.

Clustering requires distribution of work to available servers and automatic reaction to changing conditions such as the addition of new servers, flash crowd traffic, server failures, or network outages. Both IBM and Cisco provide workload distribution solutions – Sysplex Distributor and Multi-Node Load Balancing from IBM, Cisco Content Services Switch (CSS), and the Cisco Catalyst 6500 Content Switching Module (CSM) from Cisco. IBM and Cisco also provide for a joint load balancing solution based on the Server/Application State Protocol (SASP) that is supported by the CSM and the new z/OS® Load Balancing Advisor. Customers have asked IBM and Cisco for recommendations on workload balancing for various zSeries application workloads. IBM and Cisco have performed testing of a selected set of solutions in a mixed workload zSeries clustering and content switching environment intended for high availability, to demonstrate both that high availability can be attained and that the solutions can work together for mixed workloads. In addition, the architecture tested is highly scalable and can easily meet future growth demands. It should be noted, however, that scaled configurations were not tested and aren't covered as part of this paper.

1.2 This White Paper

This white paper documents the results of an interoperability testing initiative. It should be noted that the implementation of any component illustrated in this paper neither constitutes nor represents the respective component's only possible implementation.

It is beyond the scope of this paper to exhaustively document each component's full spectrum of supported features, implementation configurations, options, and recommended deployment scenarios. Therefore, it is the readers' responsibility to fully understand the technologies described within this document and to determine this paper's applicability to their particular environment prior to applying or implementing any recommendations cited within this document.

The information contained in this document has not been through any formal IBM or Cisco test and is distributed "AS IS". While IBM and Cisco have reviewed each item for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. The use of this information or the implementation of any techniques described herein is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. Customers attempting to adapt these techniques to their own environments do so at their own risk. Neither IBM nor Cisco makes any claims of the content of this document's applicability to any specific environment, either similar or different from those described within this paper.

Note also that the focus of the testing was high availability and interoperability of Sysplex Distributor with MNLB, CSS, and CSM (with and without SASP support) for different application workloads at the same time. Performance and stress testing was not a focus of this effort. Customers considering these solutions will additionally need to evaluate throughput and response time requirements in selecting the approach for a particular enterprise.

1.3 Scope and Rationale

The scope of the document is to:

1. Describe the design principles and best practices implemented to achieve high availability in a zSeries server environment with Cisco networking equipment, without compromising future scalability requirements.
2. Demonstrate the ability to distribute a variety of application workloads to various server programs residing on zSeries, using z/OS® Sysplex Distributor with MNLB, Cisco Content Services Switch (CSS), and Cisco Content Switching Modules (CSM) in a Cisco Catalyst 6500 environment. The CSM configurations also include scenarios where SASP is exploited along with the z/OS® Load Balancing Advisor.
3. Describe a variety of failure scenarios and how the cluster and network design reacts to these failures.

This document focuses on the connectivity between the Cisco Catalyst 6500, CSS, and CSM, and the zSeries server using OSA-Express components.

The first area of interest is the actual hardware and software components used in the tests.

1.4 Hardware and Software Inventory

The following sections describe the Cisco and IBM equipment and software used during the testing.

1.4.1 Testing Network Overview

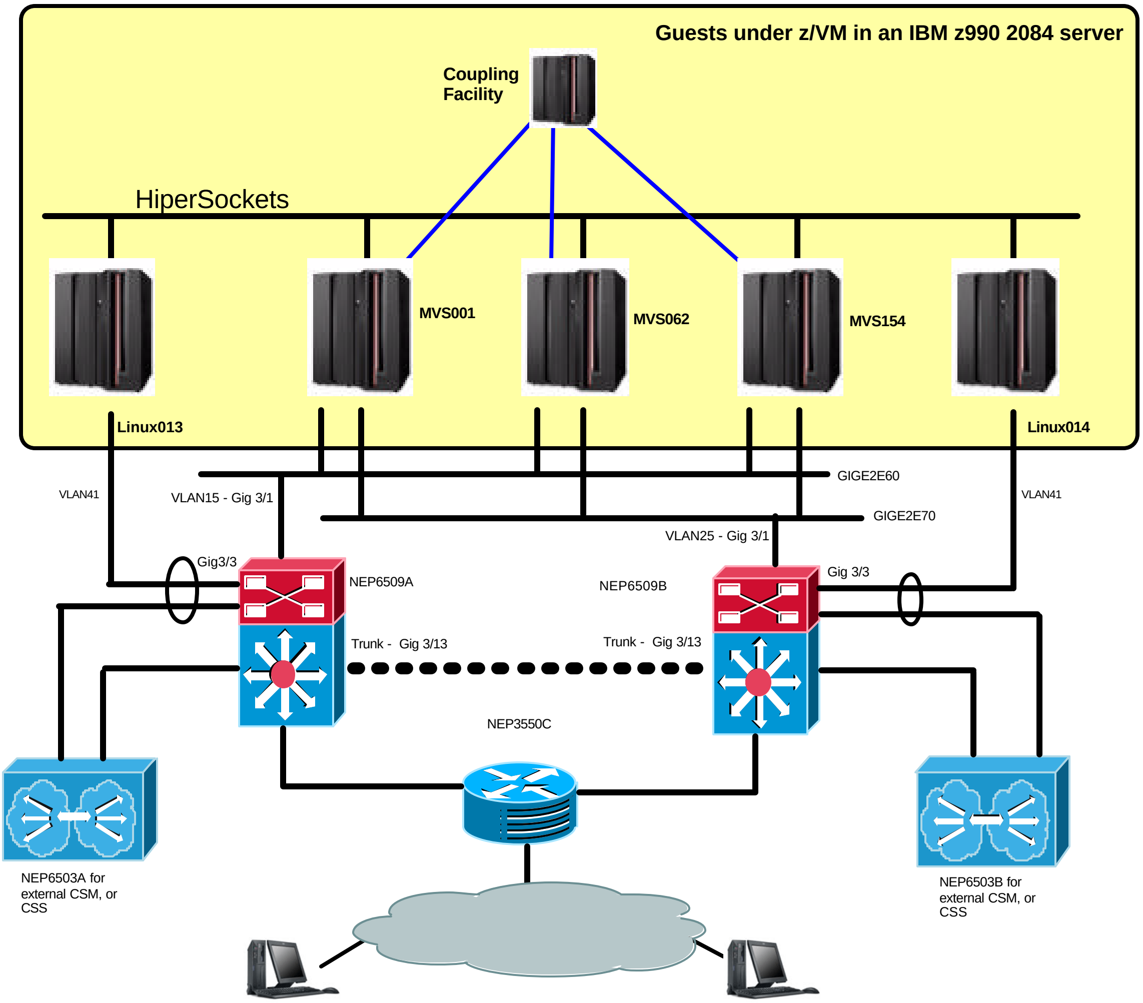


Figure 1 Test network overview

Figure 1 Test network overview shows the overall testing network. Two Cisco Catalyst 6509 switches connect to zSeries servers via Gigabit Ethernet interfaces. The zSeries servers are in this test environment deployed as guest operating systems under z/VM®. Two guests are configured for Linux® on zSeries (HTTP/HTTPS serving only). Three other guests are configured for z/OS (WebSphere® Application Server, Sysplex Distributor, FTP, Telnet and TN3270). A final z/VM guest is configured as an Integrated Coupling Facility (ICF) for the z/OS Parallel Sysplex that consists of the three z/OS operating system images. Traffic between the Linux operating system images and the z/OS operating system images, and among the z/OS systems, use HiperSockets^{TM}.

The CSS/external CSM switches in Figure 1 Test network overview and the CSM blades within the Cisco Catalyst 6509 switches are used to provide load balancing for workload distribution. They are not needed for Sysplex Distributor/MNLB. Also note that the test involving the Cisco

CSM using SASP did not include the Linux® on zSeries systems as the HTTP serving was performed using the IBM HTTP Server on the z/OS systems.

CSS/external CSM switches are not used in conjunction with the CSM blades within the Cisco Catalyst 6509 switches. External switches with CSMs are used only if the environment does not allow for service modules in the switching core. Specific configurations for Sysplex Distributor/MNLB, CSS, CSM, and CSM/SASP configurations tested will be shown and described in more detail in the "Testing Performed" section below.

1.4.2 IBM Corporation – Equipment Inventory

This section provides details of the IBM equipment (hardware and software) used to provide an inter-operable solution for high availability environments with multiple application workloads.

1.4.2.1 zSeries Connectivity and Operating System Images

Testing was performed on a single zSeries system - an IBM z990 2084 Model 332 (32 processors). In a true high-availability solution, at least two zSeries systems would be deployed so the zSeries hardware itself would not be a single point of failure, but a single zSeries system with multiple operating system images was sufficient to test Sysplex Distributor/MNLB, CSS, and CSM functions, compatibility, and coexistence. The z990 was not dedicated to this interoperability test, but was shared with many other users using the zSeries LPAR virtualization technologies. Note that while no explicit testing was performed on z9 systems, it is expected that the results and configurations described in this white paper to also apply to z9 servers.

The interoperability testing was done in a single LPAR on the z990, where z/VM was IPLed with two guests running Linux SuSE SLES 8.0 with the IBM HTTP server for testing Web workloads (Linux013 and Linux014), and three guests running z/OS V1R4 with FTP, TN3270, DB2®, LDAP, and WebSphere Application Server V5.0.3 (MVS001, MVS062, and MVS154).

As mentioned earlier, the configuration for the CSM tests using SASP did not involve the two Linux guests. The IBM HTTP Servers were deployed directly on the z/OS systems (MVS001, MVS062, and MVS154).

Network connectivity to the zSeries operating systems was via a Gigabit Ethernet infrastructure connected to zSeries OSA-Express adapters with current microcode levels, which at the time of testing was level 5.05. The Linux systems each had a dedicated OSA port. The z/OS systems were each configured to use two OSA-E ports for high availability, and the same two ports were shared among all three z/OS systems.

OSPF (Open Shortest Path First) was used as the dynamic routing update protocol by all network components in all test scenarios to achieve network high-availability objectives. On z/OS, OMPROUTE was used as the dynamic routing daemon. On Linux on zSeries, Zebra was used as the dynamic routing daemon.

Sysplex Distributor is provided as part of the z/OS Communications Server. In the testing configuration, MVS001 was chosen as the primary distributing TCP/IP stack (the stack that advertises network ownership of the distributed dynamic VIPA), while the other two z/OS images both were configured as backup distributing stacks

The z/OS Load Balancing Advisor that provides support for SASP is also part of the z/OS Communications Server. It is a new function that is available on z/OS V1R4 via APAR PQ90032 and on z/OS V1R5 and V1R6 with APAR PQ96293. In the testing configuration for CSM using SASP, the Load Balancing Advisor was deployed on MVS001, and systems MVS062 and MVS154 were configured as backup systems where the Load Balancing Advisor could be restarted in cases of a failure.

1.4.2.2 Client Hardware and Software

For testing purposes, the IBM Application Workload Modeler (AWM) V1.1 product was used. AWM provides the ability to configure test scenarios to exercise a wide range of applications with hundreds of thousands of individual application connections each. Between one and six IBM xSeries Model 335 machines (dual 2.4GHz Pentium 4 processors, 2.5 GB of memory) were used during the testing, with Red Hat V7.3 as the operating system.

While AWM will, of course, not be used for production application clients, you may wish to investigate using it for application testing and network modeling.

1.4.3 Cisco Systems – Equipment Inventory

This section provides details about the Cisco equipment that was used in the test scenarios described in this document.

1.4.3.1 Content Services Switch (CSS11503)

The CSS11503 was populated with three modules: Switch Control Module (SCM), I/O Module (IOM), and Secure Sockets Layer Module (SSLM). The SCM contained 288MB of system memory and provided two Gigabit Ethernet interfaces. One interface was used to provide connectivity to the client-side VLAN and the other interface interconnected the server-side VLAN. The IOM used was a 16 port Fast Ethernet Module with 288MB of memory. For redundancy testing, two ports on the IOM were used to provide TCP state replication. In a production environment, it is highly recommended that the replication connection be of equal capacity to the client/server inter connectivity. The SSL Module contains 512MB of memory reserved for SSL decrypting/encrypting only. The CSS was configured with the latest maintenance release of WebNS 7.10.

1.4.3.2 Content Switching Module

For the CSM testing, the Cisco Catalyst 6509 is configured similarly to the CSS, with a Supervisor 2 Module populated with both a PFC2 and MSFC2. The Supervisor, running Supervisor Cisco IOS® Software Version 12.1(19)E1, has two Gigabit Interface Card (GBIC) ports populated with GBIC transceivers. The CSM is running Version 3.2.1 and the SSLM Release 1.2.1. Note that for the CSM tests using SASP, the CSM was upgraded to level 4.1(2.5). The Cisco Catalyst 6509 is populated with a 16 port GBIC module to provide core L3 switching connectivity.
A regression test of the scenarios outlined in this paper was also performed using the Cisco Catalyst 6509, with a SUP720 (MSFC3) running Supervisor Cisco IOS® Software Version 12.2(18)SXD6. The CSM was upgraded to level 4.2(3) for the regression test.

2 Workload Distribution Technologies

This section discusses the IBM and Cisco technologies of interest to the workload distribution testing and recommendations in this paper.

2.1 IBM TCP/IP Technologies

This section introduces the IBM TCP/IP technologies that were used in the test scenarios to provide high availability and scalability.

2.1.1 VIPA and Dynamic VIPA

Traditionally, an IP address is associated with each end of a physical link, or each point of access to a shared-medium LAN. IP addresses are unique across the entire visible network. The majority of IP nodes have a single point of physical attachment to the network, but some nodes, particularly large server nodes, have more than one physical link into the network.

Within the IP routed network, failure of any intermediate link or physical adapter may disrupt end user service unless there is an alternate path through the routing network. Routers can route IP traffic around failures of intermediate links or nodes in such a way that the failures are not visible to the end applications or IP hosts.

The Virtual IP Address (VIPA) concept removes the server network adapter itself as a single point of failure by providing an IP address that is associated with a server node's TCP/IP stack, but without associating it with a specific physical network attachment, such as an OSA-Express feature. Therefore, since a VIPA has no single physical network attachment associated with it, it is always active and never experiences a physical failure as long as the TCP/IP stack is active.

To the routed network, a VIPA appears to be a host destination (a 32-bit network prefix for IPv4) on a multi-homed TCP/IP stack, such as a TCP/IP stack in a z/OS operating system image. When a packet with a VIPA destination reaches the TCP/IP stack that owns that particular VIPA, the IP layer recognizes the address as an address in the TCP/IP stack's HOME list and passes it up to the transport protocol layer in the stack for application processing.

A real IP address on a server may become unreachable if the physical network interface with which the address is associated fails. A virtual IP address is not associated with any specific physical network interface and will remain reachable as long as just a single physical network interface to the server node remains operational. A VIPA may become unavailable if the server node's TCP/IP stack or the server node itself fails, such as the failure of the full z/OS operating system image. For such failure scenarios, a VIPA can be "moved" to a backup z/OS system's TCP/IP stack and the routes to the VIPA can be re-advertised so that clients can transparently reconnect to the backup TCP/IP stack. This process is known as VIPA takeover support.

VIPA takeover has been improved with the introduction of Dynamic Virtual IP Addresses (DVIPA) in OS/390 V2R8 and Distributed Dynamic Virtual IP Addresses (Distributed DVIPA) in OS/390 V2R10. The DVIPA function improves VIPA takeover by providing a systems programmer with the ability to plan for system outage scenarios by identifying backup z/OS systems in a z/OS Sysplex to dynamically take over responsibility for a VIPA without either operator intervention or external automation. Furthermore, as soon as the failed TCP/IP stack has recovered, the "taken over" VIPA will immediately be "taken back" automatically and transparently by the recovered TCP/IP stack.

2.1.2 Sysplex Distributor

Sysplex Distributor distributes TCP connection requests to multiple server nodes within a z/OS Sysplex, without any knowledge of the nature of the requests or the application. All connection requests for a particular application are for the same IP address. Sysplex Distributor selects the application z/OS system to be used for a particular TCP connection request and uses Dynamic XCF (a facility within z/OS TCP/IP) to forward the connection request and subsequent inbound connection data to the target z/OS system's TCP/IP stack. The decision about which z/OS system to forward a new connection request to is based on input from the z/OS Workload Manager (WLM) in such a way that z/OS systems in a Sysplex with the largest amount of displaceable capacity are preferred. z/OS V1R4 implements an option to allow Sysplex Distributor to ignore WLM input and distribute in a round-robin fashion if desired for a particular application where equal distribution of workload across available servers is more important to an installation than distribution based on server capacity.

In addition to WLM input, networking policies can be defined with the z/OS TCP/IP Policy Agent to affect the workload distribution decisions made by the Sysplex Distributor. The Policy Agent policies may be used to modify the workload distribution based on time of day or which client IP address is connecting to the server cluster.

TCP/IP stacks participating in Sysplex Distributor functions all collaborate to provide the service. Basic configuration is done on the Sysplex Distributor stack, the stack that makes the distribution decision. Additional stacks should be configured as backup distributing stacks, but the configuration can be inherited from the primary distributing stack. Sysplex Distributor configuration is communicated by the primary distributing stack to all designated application hosting (target) stacks. These target stacks immediately activate the Distributed DVIPA (cluster address) as a hidden address, somewhat like a loopback address. The Sysplex Distributor stack, all backup distributing stacks, and all target stacks must be in z/OS operating system images configured to be in the same z/OS Sysplex.

When an application instance is started on a target operating system image and the application instance establishes its listening socket for the designated port, the hosting target stack notifies the distributing stack that it is ready to receive work. The distributing stack will only send connection requests to a target stack that has a listening server instance. The mere fact of starting an application instance is adequate, and no explicit registration by the application instance is required. If the application terminates or closes its listening socket, the hosting target stack will likewise notify Sysplex Distributor not to send connection requests to this stack until (or unless) another application instance is started and active.

Should the distributing stack be stopped or suffer an outage, the backup distributing stack will take over distribution responsibility. Connections to all surviving application hosting target stacks and application instances will not be affected by failure of the distributing stack, as long as a backup distributing stack has been configured.

Through z/OS V1R4, all connection requests are distributed independently, and there is no notion that requests from a particular client should go to a particular server (a condition known as "affinity"). Z/OS V1R5 adds affinity capabilities to Sysplex Distributor, where a timer-based affinity can be established between a client IP address and a specific distributed server application.

2.1.3 Sysplex Distributor and MNLB

A potential concern when using Sysplex Distributor is that while traffic from the server to the client takes the most optimal direct route, traffic inbound to the server on the application hosting target node has to flow through the Sysplex Distributor node to the target stack. This introduces an extra hop for inbound traffic, even when the distributing node and the target nodes are all directly connected to the same networking infrastructure. This extra routing hop on the inbound path through the distributing stack uses zSeries CPU cycles for inbound connection routing for all connection data even after the target stack decision has been made. Fortunately, most workload types on z/OS, such as TN3270, HTTP, CICS transaction workload, etc., are characterized by relatively small amounts of inbound data as compared to the amount of outbound data. In general, the cost of routing inbound data through the distributing node is not significantly high, but if the inbound traffic volume is high, the cost of the extra hop can become significant. Furthermore, this extra hop takes place over the Dynamic XCF link, which may be using the same coupling facility links that are used by applications and subsystems to access persistent data in the coupling facility.

As of z/OS V1R2, Sysplex Distributor and Cisco MNLB forwarding agents can interoperate to provide the best of both worlds. Sysplex Distributor can make a high-quality decision on where to send a new connection request, because it consults not only WLM, but also Service Policy Agent, and because Sysplex Distributor knows immediately when server applications come and go. The Cisco forwarding agents can bypass the Sysplex Distributor node after the target stack has been selected, for optimal routing in both directions, inbound and outbound.

Configuration on z/OS TCP/IP and Cisco equipment is not complicated. Appendix A – Example Configuration Files shows the configuration for z/OS TCP/IP and Cisco Catalyst 6509 switches for the test environment.

2.1.4 Understanding SD/MNLB Dispatch Mode Load Balancing and GRE Encapsulation

Sysplex Distributor/MNLB works on the basis of what is generally known as dispatch mode load balancing. This is also referred to as MAC-level forwarding.

The destination IP address in inbound IP packets is not changed by the load balancer, but remains the cluster IP address (the distributed DVIPA of z/OS or a VIP address in the outboard load balancer).

Dispatch mode requires that the cluster IP address is made available in the HOME list of all target stacks to which workload can be balanced. Otherwise, these target stacks would reject connection requests they received for the cluster IP addresses.

In a z/OS Sysplex using Sysplex Distributor, this is done transparently by the TCP/IP stacks in the z/OS Sysplex by exchanging control information among the stacks using XCF signaling – instructing potential target stacks to install the distributed DVIPAs as *hidden* IP addresses in the HOME list of the target stacks (status “I” in a z/OS NETSTAT HOME report).

When Sysplex Distributor is used in conjunction with MNLB, all the distributed DVIPAs are added to the HOME list of all target stacks in this manner, and no further HOME list definitions are needed on the target stacks.

If dispatch mode is used with an outboard load balancer, such as the Cisco CSM, the cluster IP addresses (the virtual server addresses on the CSM) must be manually defined in the HOME list of all the target stacks. The cluster addresses must be defined in a manner where the routing daemon of z/OS (OMPROUTE) understands that it is not to advertise those addresses to the network. This can be done in one of two ways:

1. Define the cluster IP address as an additional LOOPBACK address.

2. Define the cluster IP address as a static VIPA that is associated with a definition in OMPROUTE’s configuration file that defines the static VIPA interface as an interface that does not use dynamic routing (using the OMPROUTE interface definition statement instead of the ospf_interface or rip_interface statements).

The most common technique is to define the cluster address as a LOOPBACK address.

In its initial form, dispatch mode required that the target nodes were exactly one-hop away from the load-balancing node. This was typically achieved by having the load balancer attached to the same shared network media as the target nodes were attached to, for example, a shared Ethernet.

The dispatch mode load balancer knows the target nodes by their IP addresses on the shared network, allowing the load balancer to forward incoming packets to the cluster IP address to the chosen target stack by using the target stack’s IP address on the shared network as the next-hop IP address. The destination IP address in the forwarded packet would remain unchanged –

pointing to the cluster IP address, which is the reason why this cluster IP address must be in the HOME list of the target stacks for them to pass the packet up from their IP layer to their transport layer for local processing, instead of trying to forward the packet further on into the network.

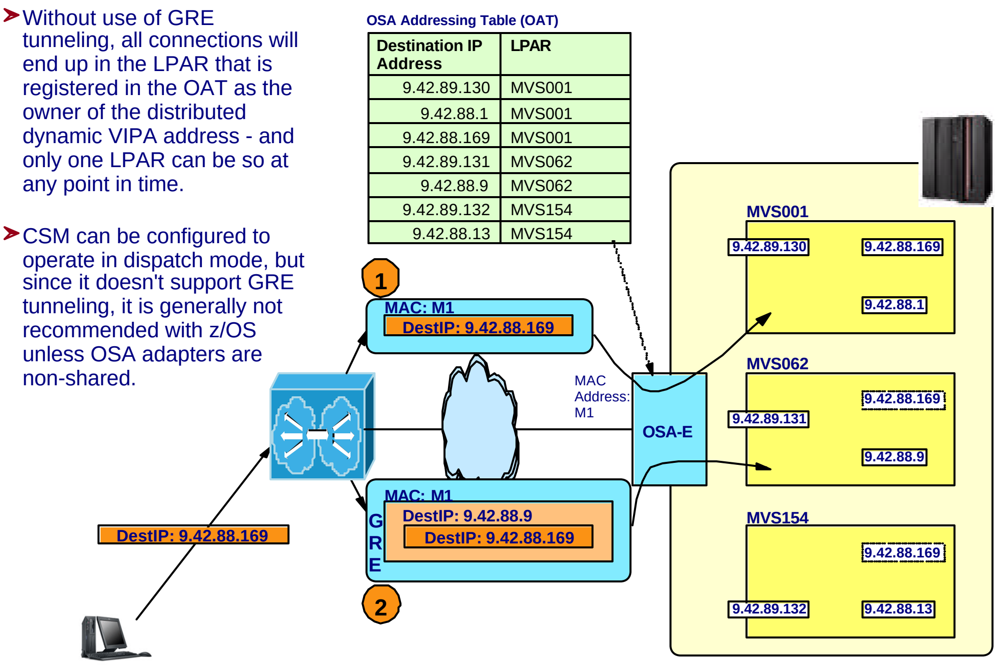
There were two main issues with this initial dispatch mode implementation:

1. As mentioned above, the target nodes had to be attached to a network that was exactly one-hop away from the load balancer. There could not be any intermediate routing in-between.

2. When OSA adapters are shared between multiple operating system images, the OSA adapter microcode uses the destination IP address in incoming IP packets to determine which operating system image to hand the packet to. If more of the operating system images were potential targets for a dispatch mode load balancer, then packets to all the target stacks would arrive with one and the same destination IP address: that of the cluster. Since the OSA adapter can have only one operating system image registered for a specific destination IP address, all those packets would be handed to one single operating system image and the other operating system images in the cluster would never receive any of the clustered workload.

Both these two issues can be addressed with a technology that is known as Generic Routing Encapsulation (GRE). Please refer to Figure 2 Generic Routing Encapsulation (GRE) for an overview.

A load balancer (including a Cisco MNLB forwarding agent) that is configured to use GRE and is to forward an IP packet to a chosen target stack will wrap an envelope around the IP packet with a new envelope IP header that sends the GRE packet to a destination IP address on the target stack coming from one of the router's IP addresses. It doesn't matter which destination IP address is used on the target stack, as long as it isn't the cluster IP address. When the GRE packet arrives at the target stack, the target stack recognizes that it is the endpoint of the GRE path, unwraps the GRE envelope, and processes the original IP packet that was destined for the cluster IP address on this target stack. In the Sysplex Distributor / MNLB test scenario, we used static VIPAs as GRE destination IP addresses.



- Without use of GRE tunneling, all connections will end up in the LPAR that is registered in the OAT as the owner of the distributed dynamic VIPA address - and only one LPAR can be so at any point in time.
- CSM can be configured to operate in dispatch mode, but since it doesn't support GRE tunneling, it is generally not recommended with z/OS unless OSA adapters are non-shared.

OSA Addressing Table (OAT)

Destination IP Address	LPAR
9.42.89.130	MVS001
9.42.88.1	MVS001
9.42.88.169	MVS001
9.42.89.131	MVS062
9.42.88.9	MVS062
9.42.89.132	MVS154
9.42.88.13	MVS154

Figure 2 Generic Routing Encapsulation (GRE)

The advantage of this technology is that the target stack may be any number of router hops away from the load balancer, and that a shared OSA adapter will never receive an IP packet for the cluster IP address, but will always receive GRE packets to individual IP addresses of the target stacks.

Please refer to Figure 3 Sysplex Distributor / MNLB Basic principles for an overview of the basic principles in this type of forwarding.

If the z/OS operating system images are attached to the same shared network as the Cisco forwarding agents and the z/OS operating system images use dedicated OSA adapters, then you do not need GRE. Otherwise, the Cisco forwarding agents must be configured to use GRE when they are load balancing to z/OS operating system images.

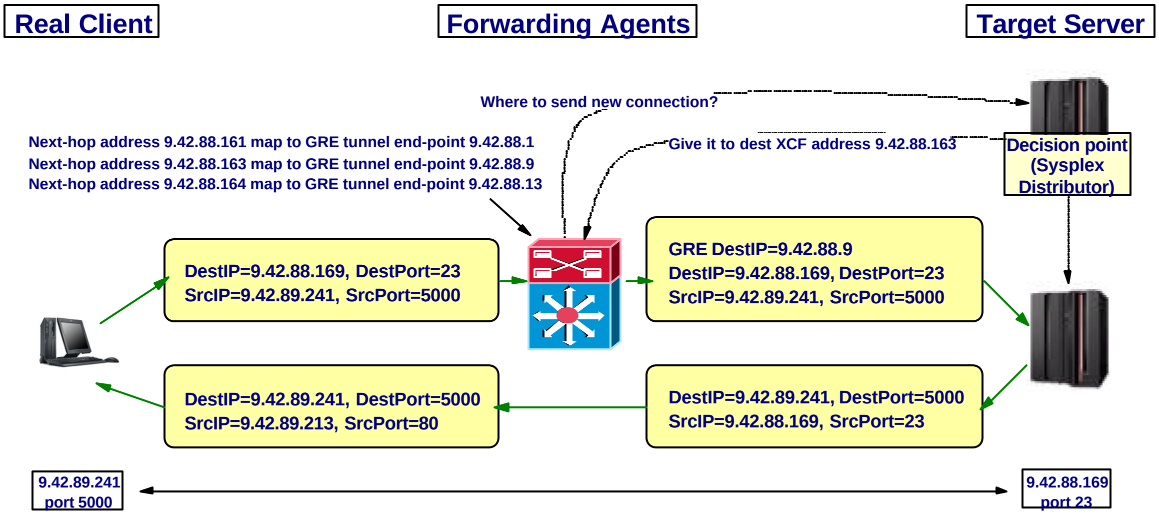

Real Client
Forwarding Agents
Target Server
Where to send new connection?
Give it to dest XCF address 9.42.88.163
Decision point (Sysplex Distributor)
Next-hop address 9.42.88.161 map to GRE tunnel end-point 9.42.88.1
Next-hop address 9.42.88.163 map to GRE tunnel end-point 9.42.88.9
Next-hop address 9.42.88.164 map to GRE tunnel end-point 9.42.88.13
DestIP=9.42.88.169, DestPort=23
SrcIP=9.42.89.241, SrcPort=5000
GRE DestIP=9.42.88.9
DestIP=9.42.88.169, DestPort=23
SrcIP=9.42.89.241, SrcPort=5000
DestIP=9.42.89.241, DestPort=5000
SrcIP=9.42.89.213, SrcPort=80
DestIP=9.42.89.241, DestPort=5000
SrcIP=9.42.88.169, SrcPort=23
9.42.89.241 port 5000
9.42.88.169 port 23

Figure 3 Sysplex Distributor / MNLB Basic principles

Sysplex Distributor and the MNLB forwarding agents use a protocol that is known as the Cisco Appliance Services Architecture (ASA) protocol to exchange control information. When the Sysplex Distributor stack starts up, it multicasts information to all the MNLB forwarding agents on a pre configured multicast IP address and port (such as 224.0.1.2 and port 1637) about which IP addresses and port numbers it is able to balance traffic for. These are known as so-called “wild-card affinities”, where only the destination IP address and port number are identified, while the source IP address and port number are wild-carded (specified as zero). When a connection request arrives in one of the MNLB forwarding agents for which Sysplex Distributor handles load balancing, the forwarding agent will send the connection request to the Sysplex Distributor stack and Sysplex Distributor will then determine which target stack this new connection should go to. Sysplex Distributor will forward the connection request packet to the chosen target stack over the dynamic XCF link, but it will also send a Cisco ASA reply back to the forwarding agent with information about which target stack it chose for this specific connection as identified via the client IP address, client port number, server IP address, and server port number – a so-called “fixed affinity”. With this information stored in the MNLB forwarding agent’s cache, it is able to route further incoming packets for this particular connection directly to the chosen target stack.

Since Sysplex Distributor identifies target stacks via their dynamic XCF addresses, the MNLB forwarding agents need to know how to reach those dynamic XCF addresses. That is normally not an issue since the z/OS routing daemon advertises the dynamic XCF addresses as host routes, but as we mentioned above, if the OSA adapter ports are shared among multiple z/OS operating system images, you need to define GRE tunnels that instruct the Cisco Catalyst switch to forward packets for those dynamic XCF destination IP addresses to specific unique IP addresses on the operating system images in mind. Our testing shows that the best choice for such unique addresses is static VIPA addresses, so a tunnel definition will basically instruct the switch that for packets where the next-hop IP address is a dynamic XCF address, the packet must be encapsulated into a GRE envelope and a GRE destination IP address must be chosen that matches a static VIPA on the stack that owns the dynamic XCF address in question.

Neither the Cisco CSS nor the CSM supports GRE tunneling. Dispatch mode load balancing should be used by the CSM only when the target zSeries operating systems use dedicated OSA ports.

2.1.5 z/OS Load Balancing Advisor and SASP

The z/OS Load Balancing Advisor, a part of the z/OS Communications Server, provides an interface that allows external IP load balancing solutions, such as the Cisco CSM, to obtain detailed information and recommendations that can be used to optimize load balancing decisions for a z/OS Sysplex environment. The recommendations provided by the Advisor are dynamic in nature; they may change as the conditions of the applications and the systems in the sysplex change. These recommendations are comprised of the following components:

- State of the target applications and systems
  This includes an indicator on whether the target application and/or target system is currently active. This allows the load balancer to exclude systems that are not active or do not have the desired application running. This can help eliminate the need for having the external load balancer produce application-level health probes in order to determine the status of a target application.
- z/OS WLM recommendations
  WLM recommendations provide a relative measure of a target system's ability to handle new workload as compared to other target systems in the sysplex. For example, if a system in the sysplex in underutilized, the weights returned for that system will be higher than the weights of the other systems. Even in scenarios where all systems in the sysplex are highly utilized, differentiation is still possible – WLM will favor systems that have a larger proportion of lower importance work (as designated by the user's WLM policy) since that work can be displaced by higher importance workload.
- Server application health, from a TCP/IP perspective
  TCP/IP statistics for target applications are monitored to determine if specific server applications are encountering problems keeping up with the current workload. For example, is a target TCP server application keeping up with TCP connection requests? Or are requests being rejected because the TCP backlog queue is full? When these type problems are detected, recommendations are adjusted appropriately so that the load balancer can direct less connections to any applications experiencing these problems. Note this type of logic is also applied to server applications using UDP. For UDP applications, the depth of UDP receive queue along with statistics relating to UDP datagrams received and dropped are analyzed to determine the health of the application.

The Server/Application State Protocol (SASP) defines the interface that is used between the external load balancers and the z/OS Load Balancing Advisor. SASP was introduced recently as part of the IBM's Enterprise Workload Manager (EWLM) solution that is part of the IBM Virtualization Engine and is available on several platforms. For more details on EWLM refer to: http://www-1.ibm.com/servers/eserver/about/virtualization/suiteforservers/

As a result, load balancing solutions that exploit SASP, can work with both the EWLM and the z/OS Load Balancing Advisor solution. SASP is an efficient protocol that is quite flexible, it provides support for both TCP and UDP applications, and also supports both IPv4 and IPv6 network configurations.

z/OS Load Balancing Advisor

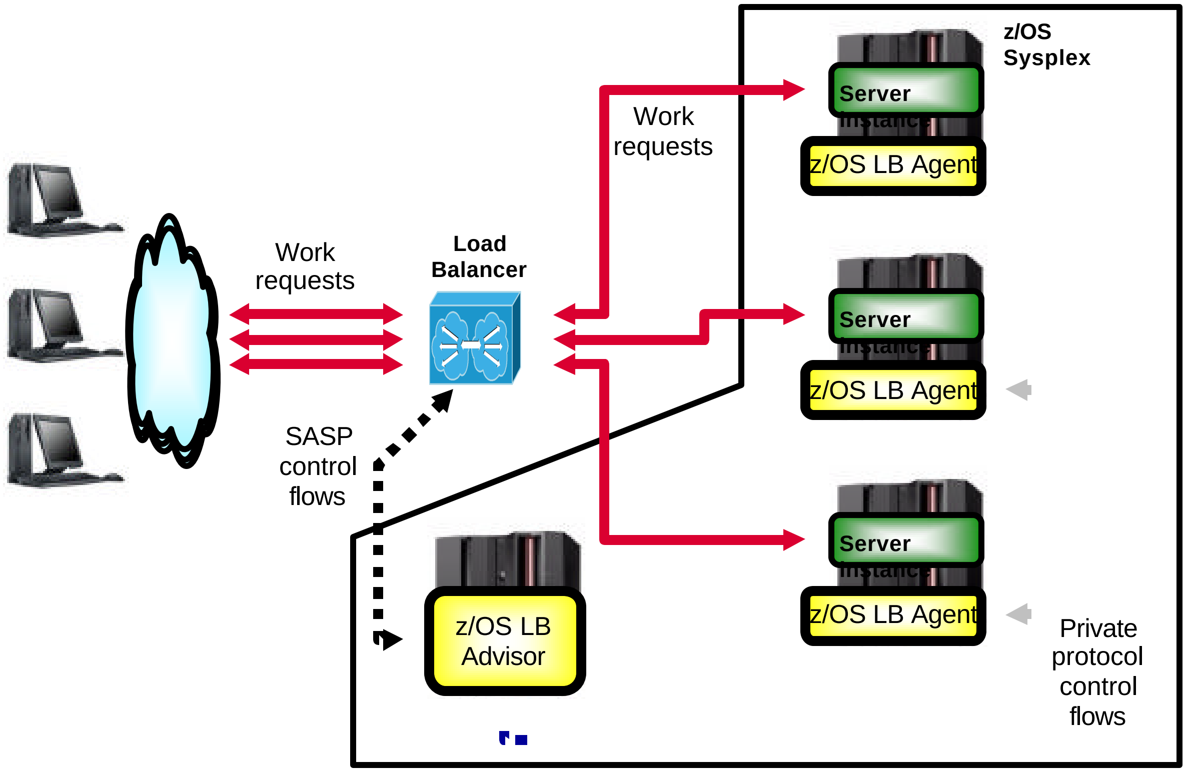


Figure 4: Caption: z/OS Load Balancing Advisor solution overview

The graphic above depicts the high level structure of the z/OS Load Balancing Advisor. A single Advisor, which executes as a started task, must be active within one of the systems in the sysplex. Using SASP, the external load balancer(s) contact the Advisor to register their interest in specific TCP and UDP applications executing within the z/OS sysplex. The Advisor communicates this information to Load Balancing Agents that are executing on every system in the sysplex that is considered a target for load balancing. The Agents, also executing as started tasks, dynamically determine what applications are active on the local system, derive the load balancing recommendations, and then communicate those back to the Advisor who provides them to the load balancer(s).

The load balancing recommendations returned by the Advisor are in the form of a relative weight for each application that is part of the load balancing group (or a server farm in Cisco CSM terms). The external load balancer can then use these weights, to make its load balancing decisions and deliver new work requests to the systems and applications that can best handle them.

All connections to the Advisor (i.e. from both agents and external load balancers) are long lived TCP connections. Using SASP, the external load balancer can decide on whether it will poll (i.e. a “pull model”) the Advisor periodically for updates in recommendations, or whether the Advisor should periodically send updates (i.e. a “push model”) to the external load balancer. The interval that the Advisor uses to compute and send recommendations is configurable on z/OS, and defaults to 60 seconds. This interval influences how often the Agents will poll TCP/IP for determining application status and health, and also on how often the Agents consult with the z/OS WLM to obtain the latest recommendations. Reducing this interval allows the Advisor and Agents to be more responsive in detecting application status and health. Obviously, there is a tradeoff to be made here; a lower interval will also increase the CPU and network overhead of the solution.

2.1.5.1 High Availability Considerations

As mentioned earlier, the z/OS Load Balancing Advisor can detect failures in applications and systems and will reflect these changes in status to any connected external load balancers. This allows the load balancers to avoid routing new requests to inactive application and system instances. However, what about other failures in the load balancing infrastructure?

For example, even though the z/OS Load Balancing Advisor is a single instance, several steps can be taken to ensure it is not a single point of failure:

- Define a unique Application Instance DVIPA (i.e. a Dynamic VIPA that is specifically associated with a single application). This allows the Advisor to be restarted anywhere in the sysplex and have the DVIPA move along with it. External load balancers and z/OS Load Balancing Agents can then connect to the new instance of the Advisor without any configuration changes.
- Ensure that for any failures in key components that the Advisor is dependent on, an automated process is in place to restart or move the Advisor. The following are some of the failure scenarios that should be considered:
    - The system that the Advisor is executing on fails. An automated process should exist to move the Advisor to another system in the sysplex. This can be accomplished using an automation software package or by using the z/OS Automatic Restart Manager (ARM). It is worth noting that the Advisor does not maintain any persistent data. Upon a restart, the Agents reconnect with the advisor dynamically. When the external load balancers reconnect to the Advisor it quickly re-establishes all of its topology information and recommendations by consulting with the Agents.
    - The TCP/IP stack on the system that the Advisor is executing on fails, causing the Advisor to also terminate. One way of dealing with this type of failure is by restarting the application on another system that has an active TCP/IP stack. Another alternative is to place the Advisor in the Autolog list of the TCP/IP stack, so that when the TCP/IP stack is restarted (by automation), the advisor is restarted as well.
    - The Advisor itself fails. In this scenario, automation to restart the Advisor (on the same or different system) is needed. This can be accomplished using ARM or an installation's automation software package.

Similar planning should be performed for the Load Balancing Agents. If an Agent is not active, then the Advisor will assume that all registered applications that are executing on that system are not active as well. It is therefore important to ensure that a process exists to automatically restart the Agent in the case of a failure. Note, that the Agent does not terminate when the local TCP/IP stack terminates. It will continue to execute and dynamically reconnect to the Advisor once the TCP/IP stack is once again operational. Therefore, the key failure that needs to be protected against is a failure in the Agent itself. In this scenario, provisions for an automated restart should be in place. This automation can also be accomplished using the z/OS ARM or an installation's automation software.

Consideration should also be given to failures in the external load balancers. Any redundant load balancers should be configured so that they can either maintain duplicate connections to the z/OS Load Balancing Advisor or to initiate a connection to the Advisor if a failure in the primary load balancer is encountered.

2.1.5.2 Detecting network connectivity failures

Possible failure scenarios that occur outside of the immediate z/OS sysplex environment should also be examined. For example, what happens if the external load balancer can not reach the Advisor because of a network connectivity issue? The external load balancer should have provisions that will allow it to detect a prolonged connectivity failure with Advisor and to provide for a fallback mechanism for performing the load balancing (e.g. such as reverting back to round robin distribution or other load balancing mechanism).

Also consider the scenario where no connectivity exists between the external load balancer and a target system in the sysplex. In this scenario, the Advisor may still indicate that the applications on that system are active as long as it can communicate with the Agent on that system. In other words, it may not be aware of down stream network failures. Furthermore, it may include a relative high recommendation for that target system as that system may be underutilized given that the load balancer is not routing work requests to it. This is a scenario where IP level health probes (e.g. pinging the IP addresses of the target system) on the external load balancer can be very useful in detecting these types of failures.

2.1.5.3 Availability of the z/OS Load Balancing Advisor

The z/OS Load Balancing Advisor function is available on z/OS release V1R4 via APAR PQ90032. Support for z/OS releases V1R5 and V1R6 is also available via APAR PQ96293. This function is a standard feature of the z/OS Communications Server in z/OS V1R7 and future releases. For more details related to this new function, refer to the APARs mentioned above and the following web site:
http://www-1.ibm.com/support/docview.wss?uid=swg27005585

2.2 Cisco Workload Distribution Technologies

This section describes the Cisco technologies that were used in the test scenarios to provide highly available and scalable solutions.

2.2.1 Cisco Catalyst 6500 Series Switch

To enable high availability within the network, Hot Standby Routing Protocol (HSRP) was implemented on the client and server VLANs on the MSFCs. This allows for a router virtual IP address to be shared between both Cisco Catalyst 6509 MSFCs, which provide constant connectivity to MSFC for both client and server packet routing. In order to provide high availability in the event of switch failure or network connectivity outage, both Cisco Catalyst 6509 chassis are connected to a client-side router. This would be an Internet facing router(s) in a production environment. The switches are also both connected to the redundant VLANs providing access to the operating system image OSA-Express adapters. In addition, the switches themselves are interconnected via Gigabit Ethernet using two channeled Gigabit interfaces carrying the client and server VLANs in an 802.1q trunk. Production environments should implement channeled trunks using line cards as opposed to the supervisor interfaces. This provides a more stable environment during code upgrades in dual supervisor configurations. OSPF is used throughout the network to maintain dynamic route availability in the event of connectivity failure.

2.2.2 Content Services Switch (CSS)

The CSS11503s are deployed as a failover pair. Each CSS is configured for VIP and Interface redundancy, which allows the CSSes to share a virtual interface similar to that of HSRP, except using Virtual Router Redundancy Protocol (VRRP). The CSSes are configured with Adaptive Session Redundancy (ASR) to allow stateful replication of the TCP state and connection tables over the Inter Switch Communication (ISC) cables. Critical services are applied to the CSSes to ensure an active/standby high availability design. A critical service is a CSS service configuration to monitor a device that is critical to the CSS's health. Critical services are typically used to monitor next hop routers. In most configurations the CSS cannot pass traffic to clients or servers without next hop routers. Thus if one of these devices cannot be reached, the CSS should fail over to allow the other CSS to become active. This allows one CSS to be the active Master handling client connections, server responses, and server health checking. In the event the Master CSS fails, or loses network connectivity to the Cisco Catalyst HSRP interfaces, the Master CSS will fail over to the Backup CSS.

2.2.3 Content Switching Module (CSM)

The CSMs are configured in redundant mode for fault tolerance, allowing the active and standby CSMs to share state information about user sessions and provide connection redundancy. If the active Cisco CSM fails, open connections are handled by the standby CSM without interruption, and users experience transparent failover. Similar to the CSS, the CSMs are also configured with shared VIPs and management (interface) IPs. A dedicated VLAN (Fault Tolerant VLAN) is carried across the two Cisco Catalyst switches hosting the CSMs, allowing for direct communication, monitoring and state information exchange between the active and standby CSMs. The active and standby CSMs must be Layer-2 attached; the fault tolerant VLAN cannot be routed.

The CSMs also provide support for the SASP protocol which can be enabled in environments where SASP is supported, such as environments that include the IBM Enterprise Workload Manager and z/OS Sysplex environments using the z/OS Load Balancing Advisor. When enabled for SASP in a z/OS Sysplex environment, the CSMs register all members of the specified server farms to the z/OS Load Balancing Advisor. The z/OS Load Balancing Advisor then periodically provides weight recommendations and status information for each registered member. The CSMs use this information to influence how new TCP connection requests are distributed among the available servers.

2.2.4 SSL Services Module (SSLM)

The Cisco Catalyst SSLM provides high-performance SSL termination capabilities. Traffic is directed to the SSLM for termination using the CSM. Once the traffic is SSL terminated, it is decrypted and sent back to the CSM for content-aware load balancing. Redundancy at the SSL level is provided by the CSM, which uses health checks to monitor the availability of multiple (usually 2 or 4) SSL modules hosted on the two Cisco Catalyst switches. When the CSM detects that an SSL module has been powered down or has failed, it will stop sending new connections to it and will start utilizing only the remaining SSL module/s. This is referred to as active-active stateless redundancy, since all the SSL modules are active at the same time and in case of a SSL module failure, the TCP connections and SSL sessions are torn down (the client will have to reconnect and complete a new SSL handshake).

2.2.5 Understanding Directed Mode Load Balancing

It is not common to configure an external load balancer to use dispatch mode in traditional server load balancing environments. They will normally be configured to use what is generally known as directed mode, also sometimes referred to as NATed mode or proxied mode. Dispatch mode load balancing is supported by the CSM, but it can be used with zSeries servers only if OSA ports are dedicated. Since the majority of zSeries installations tend to use shared OSA ports, this document focuses on directed mode load balancing and will not include any samples of using CSM in dispatch mode.

Directed mode load balancing works on the basis of changing the destination IP address of the inbound IP packets to a unique IP address of the chosen target stack.

One of the advantages of directed mode is that there can be any number of router hops in-between the load balancer and the target stacks.

One of the challenges of directed mode is that outbound packets must be directed back via the load balancer in order for the load balancer to change the source IP address of outbound IP packets from the unique IP address of the target stack to the cluster IP address. Otherwise, the remote client would reject the response since it would come from an IP address other than the one it sent a request to.

There are two ways to ensure that this outbound routing back via the load balancer happens:

1. The load balancer can be configured to change only the destination IP address in inbound IP packets and not the source IP address (known as server NAT). If that is the

case, then the target stack will respond with IP packets back towards the client IP address directly. These packets must be directed to the load balancer for it to change the source IP address in the outbound packets. This can be achieved in one of two ways:

- **a.** By having all outbound packets routed via the load balancer (via a default route definition on the target stacks), or
- **b.** By having the routing infrastructure in-between the load balancer and the target node implement Policy-Based Routing (PBR) where the routers recognize IP packets from the clustered servers (based on source IP address and/or source port number); in this setup the router sends only those packets back via the load balancer, while outbound packets for workload that were not balanced are routed directly back to the clients.

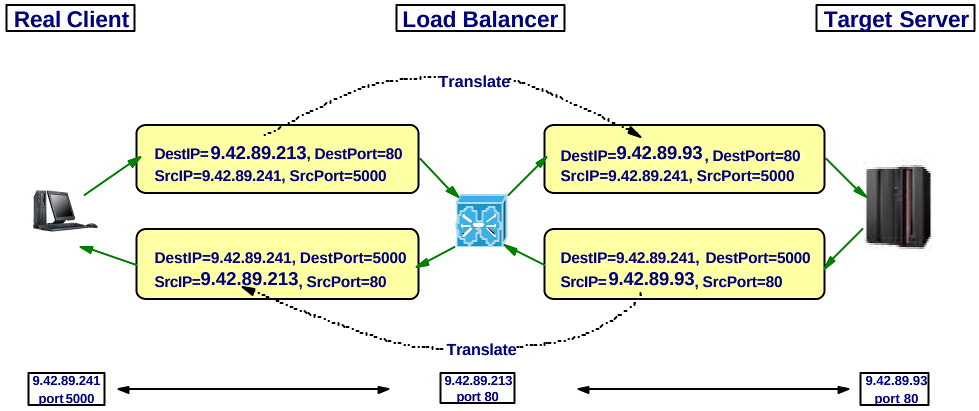

Real Client
Load Balancer
Target Server
Translate
DestIP=9.42.89.213, DestPort=80
SrcIP=9.42.89.241, SrcPort=5000
DestIP=9.42.89.93, DestPort=80
SrcIP=9.42.89.241, SrcPort=5000
DestIP=9.42.89.241, DestPort=5000
SrcIP=9.42.89.213, SrcPort=80
DestIP=9.42.89.241, DestPort=5000
SrcIP=9.42.89.93, SrcPort=80
Translate
9.42.89.241
port 5000
9.42.89.213
port 80
9.42.89.93
port 80

Figure 4 Server NAT with Policy Based Routing

2. The load balancer may also be configured to change both the destination IP address and the source IP address in the balanced inbound IP packets (known as server NAT and client NAT). The advantage of this is that outbound IP packets from the target stacks will be sent out with a destination IP address of the load balancer and because of that be routed to the load balancer, where both source IP address and destination IP address of the outbound packets are then changed to match the original cluster IP address as source and client IP address as destination.

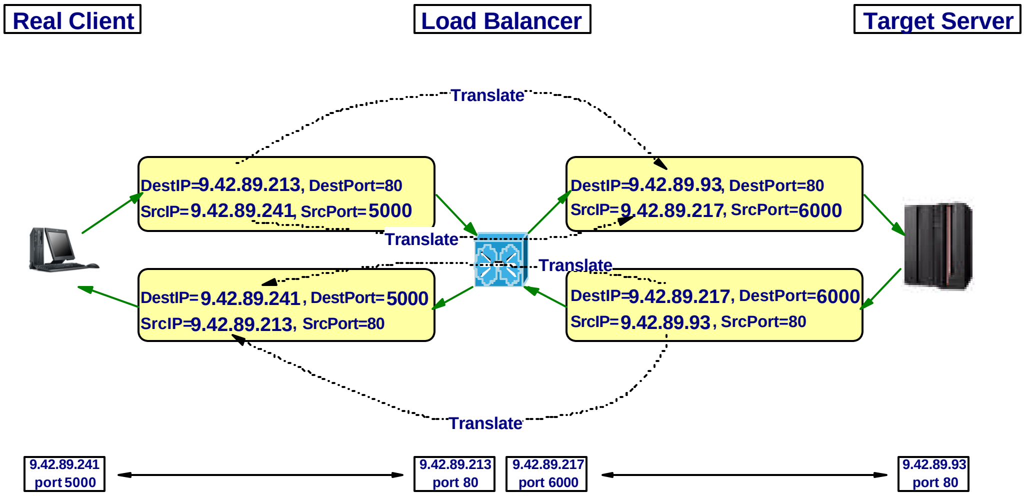

Real Client
Load Balancer
Target Server
Translate
DestIP=9.42.89.213, DestPort=80
SrcIP=9.42.89.241, SrcPort=5000
DestIP=9.42.89.93, DestPort=80
SrcIP=9.42.89.217, SrcPort=6000
Translate
Translate
DestIP=9.42.89.241, DestPort=5000
SrcIP=9.42.89.213, SrcPort=80
DestIP=9.42.89.217, DestPort=6000
SrcIP=9.42.89.93, SrcPort=80
Translate
9.42.89.241
port 5000
9.42.89.213
port 80
9.42.89.217
port 6000
9.42.89.93
port 80

Figure 5 Server NAT with Client NAT

> The client NATing will be done by NATing the real client IP address to a load balancer IP "client" address and be combined with client source port address translation (PAT). From a target server point of view, it will look like all connections come from one and the same client source IP address (that of the load balancer), but each client connection comes from a different source port number on that load balancer. One of the implications of this behavior is that it isn't possible on the server nodes to see what the real client IP address is, which among other things may complicate diagnosing network related problems.
>
> Client NATing may have an impact on z/OS networking policy functions. Networking policies on z/OS can be used to apply network QoS differentiated services, selection of Intrusion Detection Service (IDS) actions, and Traffic Regulation (TR) limitations. z/OS networking policies are specified via policy conditions and actions. The conditions can be defined in various ways, one of which is to use the client source IP address. One example is to apply high-priority outbound treatment to traffic that is destined for a specific user community, such as all IP addresses that belong to the customer query department. If client NATing is used by the load balancer, all inbound packets will come from one IP address (that of the load balancer) and networking policy conditions that were defined based on the real client IP addresses will not be applied to that traffic.
>
> Another z/OS function that is impacted by client NATing is the NETACCESS rules that can be used to authorize individual users on z/OS to send or receive data from or to selected sections of a network, as identified via network prefix definitions on the NETACCESS policies in the z/OS TCP/IP configuration files. NETACCESS rules may also be used to assign Multi Level Security (MLS) labels to inbound IP packets in an MLS-enabled security environment.
>
> If z/OS networking policies are defined based on client IP addresses or if NETACCESS rules are in use, client NATing should be chosen with care, since it may disrupt the operation of those functions.
>
> To complete this discussion, one also needs to use client NATing with care if connections are balanced to TN3270 servers that use the client source IP address or hostname to choose TN3270 connection options, such as selecting an LU name or a primary SNA application for a given connection.

Having introduced the hardware and software components of this project and discussed the basic principles of the forwarding technologies used, attention now turns to the workloads that were distributed.

3 Description of the Workloads and Client Simulation

This section describes the workloads and how the client simulation was performed during the testing.

3.1 FTP

This workload used at least one client workstation, but not more than six client workstations, to transfer a set of ten 10MB (megabyte) files to a zSeries Server using File Transfer Protocol (FTP).

3.2 TN3270

This workload had 15 clients, each with 50 connections, and the test returned 8KB of data to the client in total. The workload was a typical one of requests being a couple of hundred bytes, and responses being 1-2 KB.

3.3 Web/HTTP

20 client sessions were configured to retrieve files of 64 KB and 128 KB sizes.

3.4 Application Workload Modeler (AWM)

AWM is an IBM product that is capable of simulating several real-world client applications. The following workloads can be simulated by AWM.

3.4.1 AWM Web Client

The AWM Web client is designed to simulate a real user browsing a Web site. It supports both HTTP and HTTPS requests.

For each Web page that the AWM Web client requests, a TCP connection is created, a request for a page from a Web site is made, and the TCP connection is closed once it has received the Web page. The AWM Web client is configured to use a list of Web pages for its requests and will continuously cycle through the list until the test case run has ended.

One AWM Web client test tool is capable of simultaneously simulating multiple real-world client workstation sessions.

3.4.2 AWM TN3270

The AWM TN3270 client is designed to simulate a real interactive Telnet 3270 user connected to the zSeries TN3270 server. The AWM TN3270 client workload supports SSL/TLS connections to the z/OS TN3270 server.

3.4.3 AWM FTP

The AWM FTP client is designed to simulate a real interactive FTP client user connected to the zSeries FTP server. The AWM FTP client workload uses non-SSL/TLS sessions and active mode data connections. The tests with SSL/TLS, passive mode, and extended passive mode data connections were performed manually.

4 Testing Performed

This section of the document describes each of the three main test scenarios that were implemented during the creation of this white paper.

4.1 Sysplex Distributor/MNLB

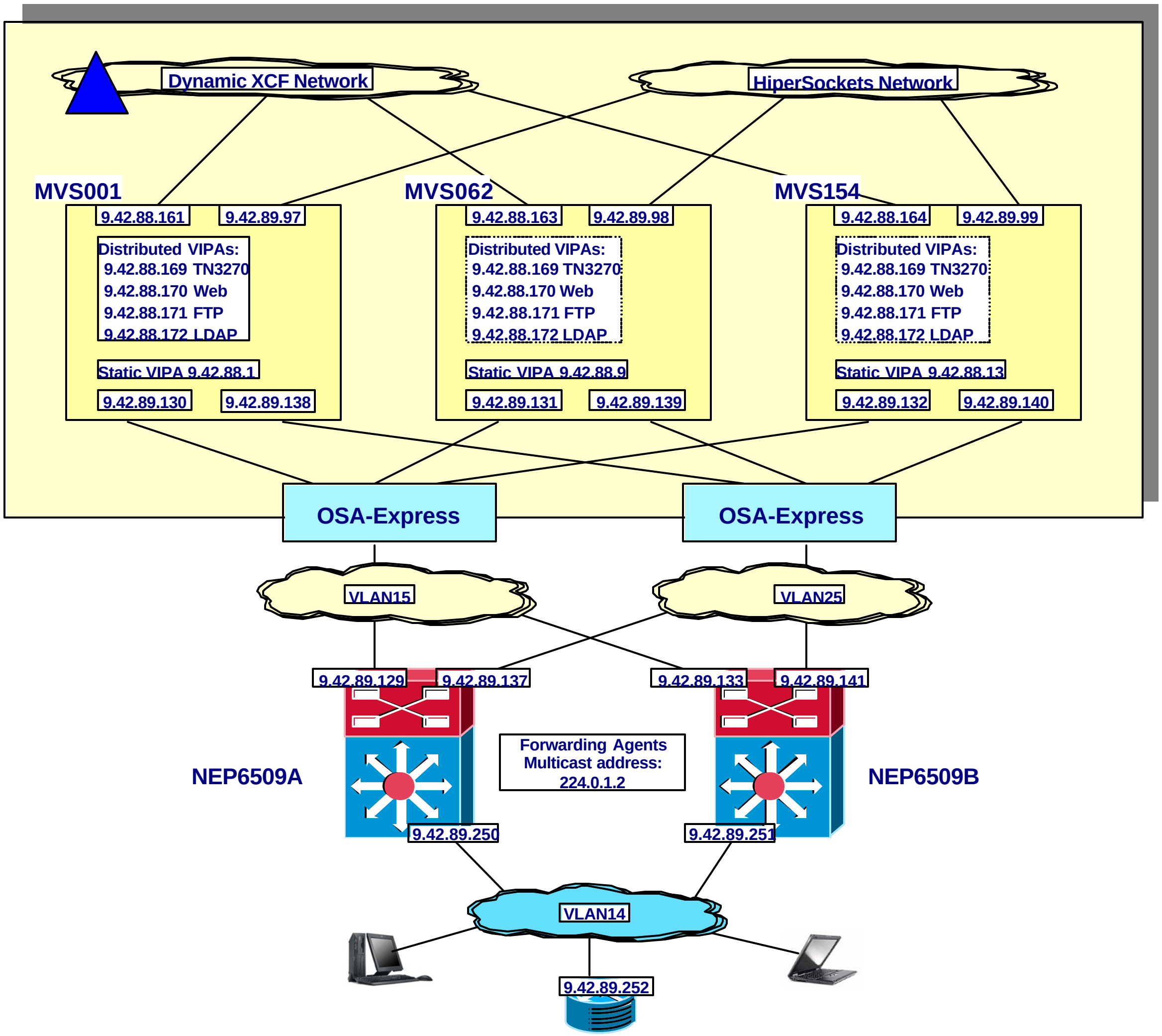


Figure 6 Sysplex Distributor/MNLB test configuration

Figure 6 Sysplex Distributor/MNLB test configuration shows the portions of the test network that directly apply to Sysplex Distributor and MNLB. Since Sysplex Distributor applies to a z/OS Sysplex only, the Linux operating system images do not participate in this test scenario. It is worth noting that the SD/MNLB workloads were tested simultaneously with the Web workloads being distributed by CSS and CSM and both inboard and outboard configurations (see below) to verify good coexistence. Since the workloads are assigned unique and separate IP addresses, the above configuration is all that is required simply to understand Sysplex Distributor and MNLB.

MVS001 was designated as the primary Sysplex Distributor node, and MVS062 and MVS154 were application hosts for TN3270 and FTP and backup distributing nodes. Sysplex Distributor does allow workload to be distributed to the same operating system where the Sysplex Distributor function resides (in this case on MVS001). MNLB forwarding agents were configured on both Cisco Catalyst 6509 switches.

The following tests were performed:

1. **Basic distribution testing.** Numerous client requests were sent to Sysplex Distributor on MVS001 for FTP and TN3270. FTP and TN3270 requests were properly distributed between all three z/OS images. Once the connection was established, connection traffic was observed to flow straight from the Cisco Catalyst 6509-hosted MNLB forwarding agents to the respective target stacks.

2. **Basic network failure testing.** Links into and out of the Cisco Catalyst 6509 switches were pulled, and the Cisco Catalyst 6509 switches themselves were shut down. As long as at least one Cisco Catalyst 6509 was operational, and there was a network path from the clients to the z/OS operating system images, application traffic continued uninterrupted, and new connections were distributed to application instances.

3. **Application failure testing.** Application instances were stopped. Client connections to the stopped application were of course lost but new connection requests were distributed to the remaining active server instance(s). When the stopped application instance was restarted, new connection requests were once again distributed to that application instance and all other active application instances.

4. **Target stack failure testing.** The stacks on MVS062 and MVS154 were stopped. Existing connections terminating in the stopped stacks were lost, as is always the case with losing a connection endpoint stack. The lost connections were immediately reset by the distributing stack on MVS001. This so-called fast connection reset prevents end users from having to wait through the normal TCP recovery time period (up to three minutes) before being able to reconnect to the server. The fast connection reset means that the client end users were informed immediately that their connection was lost and they could then reconnect to one of the remaining server instances in the Sysplex. All new connection requests were distributed only to an active server instance on an active target stack. When the stopped target stack and server instance were once more functional, new connection requests were distributed to the restored stack and application instances once again.

5. **Sysplex Distributor stack failure testing.** The stack on MVS001 was halted. Client TN3270 connections to the TN3270 server on MVS001 were lost, but the distributing responsibility was taken over by one of the other stacks, and client connections to the other two stacks were not interrupted. When the stack on MVS001 was restored, it took back Sysplex Distributor responsibility transparently (without interruptions) to all then-existing connections.

Since SD/MNLB is a dispatch mode technology, it will work for both SSL/TLS connections and non-SSL/TLS connections. All the above test cases work for non-SSL/TLS connections as well as for SSL/TLS connections.

Since the OSA ports were shared in this test scenario, the Cisco Catalyst 6509 switches were configured to use GRE tunneling.

4.2 Cisco Content Services Switch (CSS)

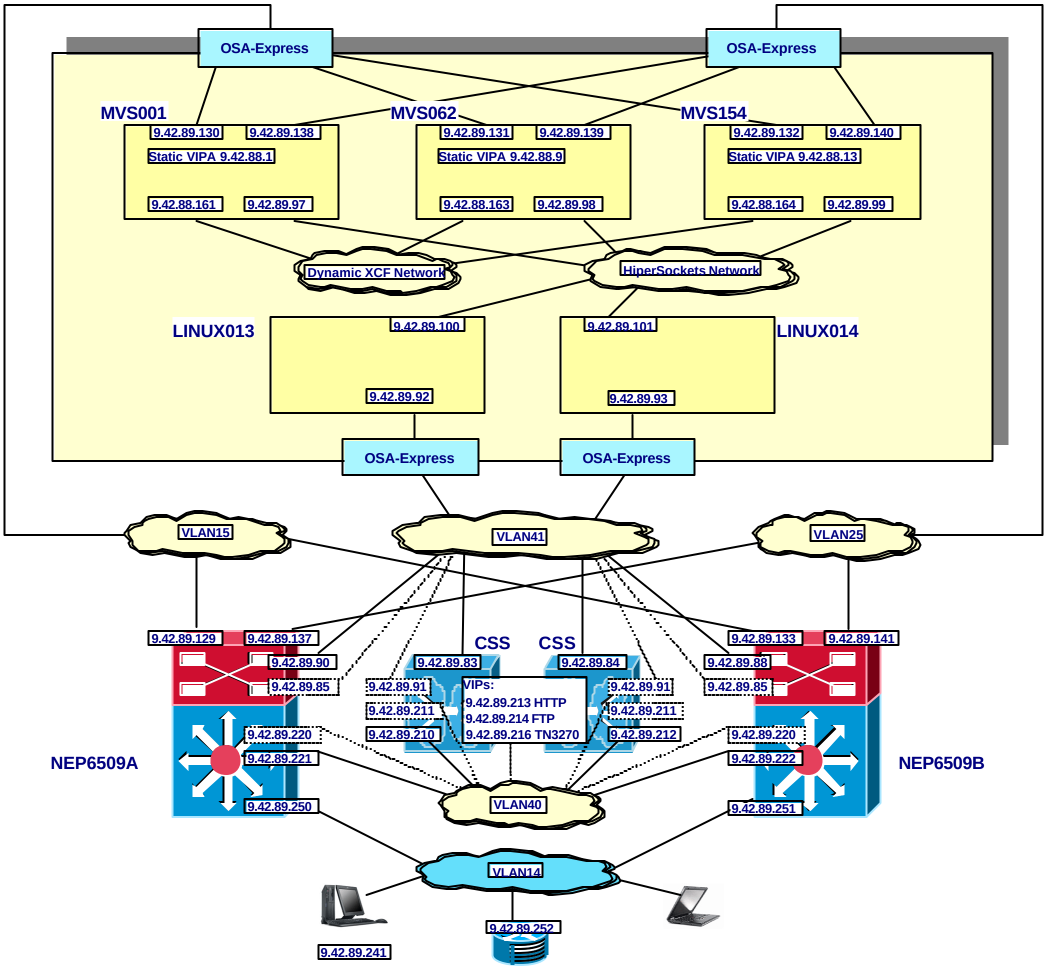


Figure 7 Content Services Switch test configuration

Two CSSs were used for CSS testing, attached to each of the Cisco Catalyst 6509 switches. The HSRP address for the client-side VLAN on the Cisco Catalyst 6509 switches was made the default route for the CSS. The CSS's other interface was Layer-2 adjacent to the Linux on zSeries servers.

The following tests were performed:

1. **Basic Web HTTP distribution testing.** Web requests were distributed to both of the Linux operating system images and HTTP server instances. Non-Web application traffic to z/OS was also run during basic HTTP distribution testing without any affects (all application traffic was distributed among all available instances of the respective application servers).

2. **FTP distribution testing.** The FTP traffic was load balanced between the three z/OS hosts.

3. **TN3270 distribution testing.** The TN3270 traffic was load balanced between all three z/OS hosts.

4. **Stop the OSA interface on a z/OS host.** Generally traffic had an additional route to the host. In cases where another route was not available (such as to the Linux on zSeries operating system images in this sample configuration), the CSS used keepalives (application-specific probe packets) to detect when the server was no longer available. The CSS would then balance traffic between the remaining available servers. The initial connection to the failing server would be lost, but new connections would from then on be directed to the operational server.

5. **Stop the HiperSockets interface on a z/Linux host (HTTP traffic only).** Traffic was redirected out the Ethernet interface to the WebSphere Application Server cluster and continued uninterrupted.

6. **Media failure testing: pulled connections from CSSs.** The CSS was connected to the Cisco Catalyst 6509 via two 1GB Ethernet links. VLAN40 connected the CSS to the client-side of the configuration and VLAN41 connected the CSS to the server-side. When either connection was pulled, the Backup CSS became the Master and a stateful (no established connections lost) failover was achieved. The CSSs were interconnected via two ISC ports for redundancy. When either connection went down, the CSSs were able to communicate over the remaining link.

7. **Power failure testing: shut down single Cisco Catalyst 6509 or CSS.** When the Master CSS was shut down, the Backup CSS became the Master. Traffic continued to flow and was balanced across the servers. When the Cisco Catalyst 6509 was shut down, traffic was rerouted through the remaining Cisco Catalyst 6509. If required, the Backup CSS would become the Master. Traffic would continue flowing unless there was no longer an alternate path to the server.

8. **System reset/reload.** Same results were achieved as for power failure testing.

9. **Tier 1 failures: shut down HTTP, FTP, TN3270 services on hosts.** The application-specific probes on the CSS detected when a service was not available. Existing connections to the server were lost, but new connections were routed to available servers. Once the failing service was revived, the subsequent incoming connections were distributed across all available servers once again.

10. **Tier 2 failures: shut down individual/all WebSphere Application server cluster nodes (HTTP traffic only).** When individual cluster nodes were shut down, traffic was balanced between the remaining nodes. When the entire cluster went down, the actual Web content could not be retrieved. Complete cluster failures could also be handled by configuring a backup server farm that would redirect connections to a remote or alternate location.

4.3 Cisco Content Switching Module (CSM)

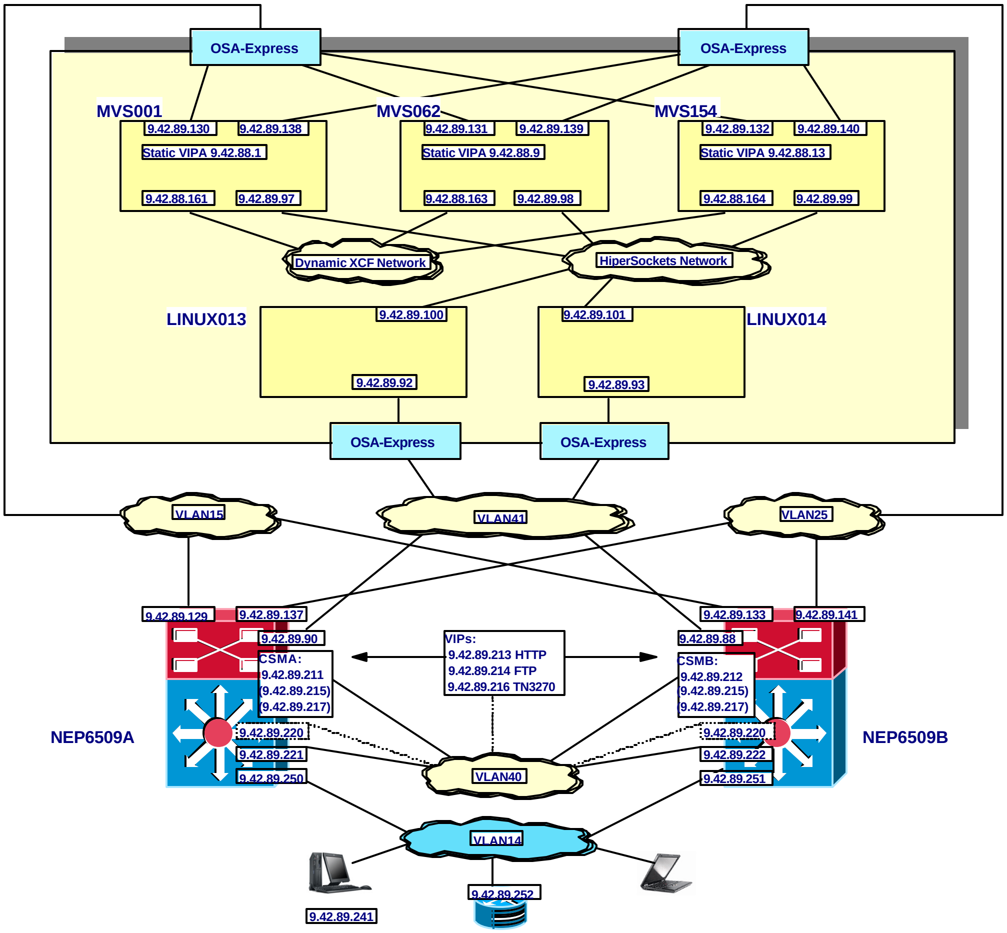


Figure 8 Content Switch Module test configuration

The CSM and SSL Modules were deployed directly in the Cisco Catalyst 6509 switches. The testing was performed for two alternatives: Server NAT with Policy-Based Routing (PBR) and server NAT with client (source) NAT. Figure 8 Content Switch Module test configuration shows the CSM configuration.

The following CSM tests were performed:

1. **Basic Web HTTP distribution testing.** Web requests were distributed to both the Linux operating system images and HTTP server instances. Non-Web application traffic to z/OS was also run during basic HTTP distribution testing without any affects (all application traffic was distributed among all available instances of the respective application servers).

2. **FTP distribution testing.** The FTP traffic was load balanced between the three z/OS hosts.

3. **TN3270 distribution testing.** The TN3270 traffic was load balanced between the three z/OS hosts.

4. **Stop the OSA interface on a z/OS host.** In most cases the traffic had an additional route to the host, but where another route was not available, the CSM health probe

successfully detected that the server was no longer available. The CSM would then balance traffic between the remaining servers in the server farm. New connections would be directed to the operational server.

5. **Stop HiperSockets interface on a Linux on zSeries host (HTTP Traffic only).** Traffic was redirected out the Ethernet interface to the WebSphere Application server cluster.

6. **Media failure testing: pulled connections from Cisco Catalyst 6509 switches.** Because of the redundancy built into the channel port, the channel port remained up when individual connections were pulled and replaced on the Cisco Catalyst 6509s.

7. **Power failure testing: shut down single Cisco Catalyst 6509.** Traffic was rerouted through the remaining Cisco Catalyst 6509 and CSM. Traffic continued flowing unless there was no longer an alternate path to the server.

8. **System reset/reload.** The tests were run the same as for power failure testing.

9. **Tier 1 failures: shutdown HTTP, FTP, TN3270 services on hosts.** The application probes on the CSM detected when a service was not available. Existing connections to the server were lost, but new connections were routed to available servers. Once the failing service was revived, the subsequent incoming connections were distributed across all available servers once again.

10. **Tier 2 failures: shut down individual/all WebSphere Application Server cluster nodes (HTTP traffic only).** When individual cluster nodes were shut down, traffic was balanced between the remaining nodes. When the entire cluster went down, the actual Web content could not be retrieved.

The CSM may alternatively be deployed in a separate Cisco Catalyst 6500 chassis in the form of an "appliance" hanging off the main Cisco Catalyst 6500 Switch. Some customers prefer to keep advanced services separate from the main switching capabilities but still require the high performance of the CSM versus the CSS. In this case, the CSM can be installed in a separate Cisco Catalyst 6500 chassis such as the Cisco Catalyst 6503 that was tested for this white paper. In addition, the CSM today is not a fabric-enabled module, meaning it does not have a connection to the switch fabric but rather communicates solely over the shared bus. Customers with switch fabric modules (SFM) or Supervisor 720 (with integrated fabric) modules deployed should analyze the performance impacts of deploying a non-fabric enabled card in these chassis. If the switch performance impacts are not acceptable, then deploying CSM in a separate chassis should be considered.

It is recommended, when configuring the CSM in a separate Cisco Catalyst 6500, that a port channel interface be used to aggregate multiple Gigabit interfaces. This feature will allow for additional bandwidth as well as failover capabilities in the configuration. The port channel is then configured as a trunk, and VLAN interfaces on both the external Cisco Catalyst 6500 and the core Cisco Catalyst 6500 switches are common. It is recommended, when using multiple Gigabit interfaces for port channeling, that the interfaces be split over multiple processor cards. This increases the failover protection if the entire module is lost.

4.4 Content Switching Module (CSM) using SASP and the z/OS Load Balancing Advisor

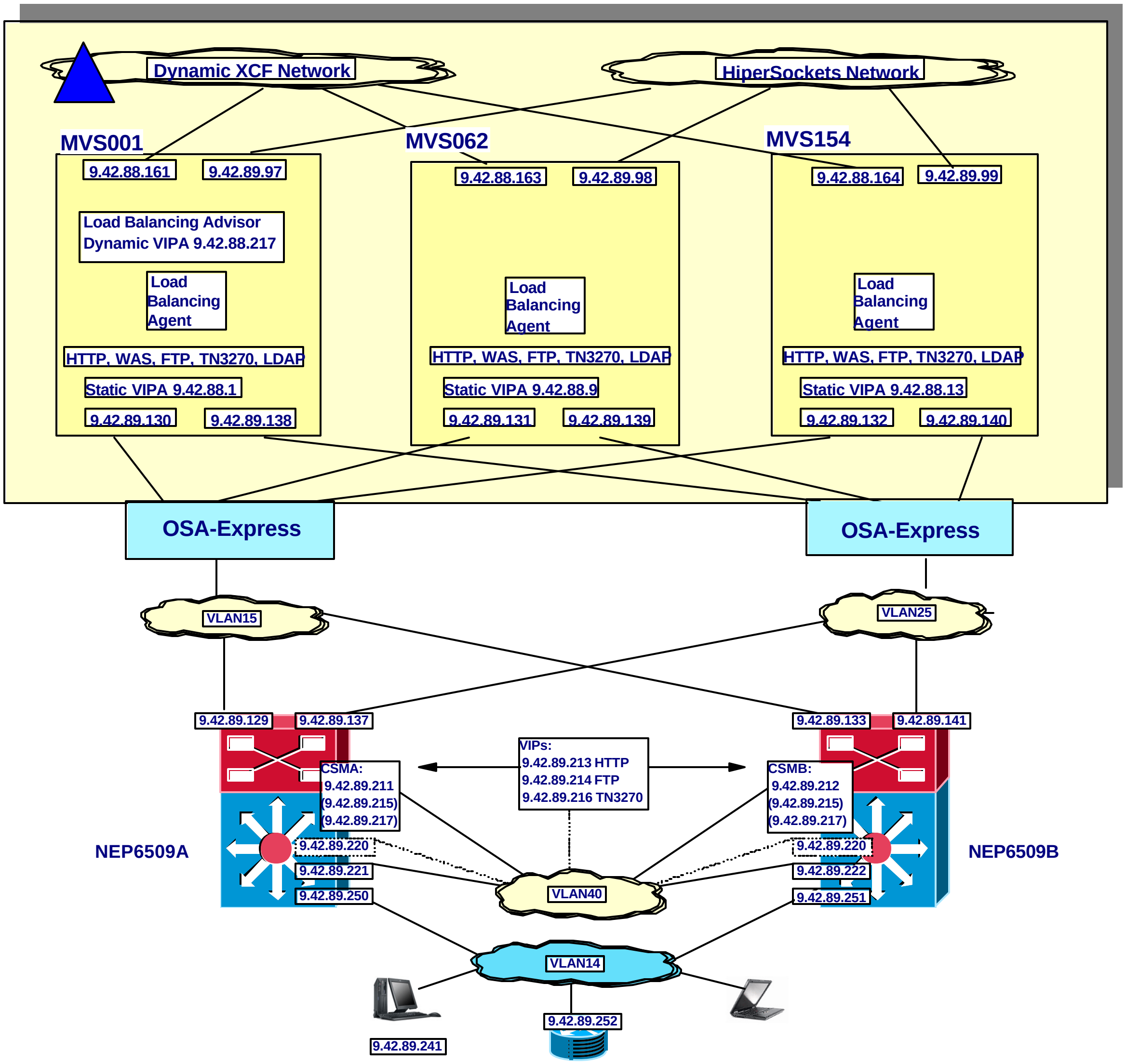


Figure 9 Content Switch Module test configuration with SASP and the z/OS Load Balancing Advisor

The CSMs were deployed directly in the Cisco Catalyst 6509 switches. The testing was performed for two alternatives: Server NAT with Policy-Based Routing (PBR) and server NAT with client (source) NAT. Figure 9 Content Switch Module test configuration with SASP and the z/OS Load Balancing Advisor shows the CSM configuration.

The testing performed in this configuration was very similar to the testing performed in section 4.3 Cisco Content Switching Module (CSM) with the following key differences:

- The HTTP servers were deployed on z/OS systems (using the IBM HTTP Server). This allowed the z/OS Load Balancing Advisor to provide recommendations using SASP for these servers as well.
- The Load Balancing Advisor was deployed on z/OS system MVS001 and Load Balancing Agents were deployed on all z/OS systems.
- Both CSMs were enabled for SASP communications. Both NEP6509A and NEP6509B were configured to maintain parallel SASP connections to the Load Balancing Advisor. This configuration allowed for fault tolerance in the case of CSM failure. It should be noted that this configuration requires that each CSM is configured to use a unique Identifier when connecting to the Load Balancing Advisor.

.

The following CSM tests were performed:

1. **Basic Web HTTP distribution testing.** Web requests were distributed HTTP servers running on the three z/OS hosts. Non-Web application traffic to z/OS was also run during basic HTTP distribution testing without any affects (all application traffic was distributed among all available instances of the respective application servers).

2. **FTP distribution testing.** The FTP traffic was load balanced between the three z/OS hosts.

3. **TN3270 distribution testing.** The TN3270 traffic was load balanced between the three z/OS hosts.

4. **Media failure testing: pulled connections from Cisco Catalyst 6509 switches.** Because of the redundancy built into the channel port, the channel port remained up when individual connections were pulled and replaced on the Cisco Catalyst 6509s.

5. **Power failure testing: shut down single Cisco Catalyst 6509.** Traffic was rerouted through the remaining Cisco Catalyst 6509 and CSM. The remaining CSM continued to obtain weight recommendations from the Load Balancing Advisor over its own SASP connection. The traffic continued flowing unless there was no longer an alternate path to the server.

6. **Cisco Catalyst 6509 and CSM System reset/reload.** The tests were run the same as for power failure testing.

7. **Target application failures: shutdown HTTP, FTP, TN3270 services on hosts.** Within the configured update interval, the Load Balancing Advisor detected when a service was not available and informed the CSM using SASP. Existing connections to the server were lost, but new connections were routed to available servers. Once the failing service was revived, the Advisor notified the CSMs and subsequent incoming connections were distributed across all available servers once again. Application level CSM health probes were not configured and not necessary to detect these failures.

8. **Stop the OSA interface on a target z/OS host.** In most cases the traffic had an additional route to the host, but where another route was not available, the CSM health probe successfully detected that the server was no longer available. The CSM would then balance traffic between the remaining servers in the server farm. New connections would be directed to the operational server.

9. **Stop the OSA interface on the host that Load Balancing Advisor was executing on.** In most cases the traffic had an additional route to the host, but where another route was not available, the CSM did detect that the Load Balancing Advisor was not responding and reverted to using Round Robin distribution for new work requests.

10. **Advisor z/OS system reset.** System z/OS MVS001 was reset. The Agents and the CSMs lost their connections to the Advisor. The Advisor was restarted by the z/OS ARM on MVS062 and the load balancers and agents established new connections to the load balancer automatically. Existing and new TCP connection requests were not impacted.

11. **Target z/OS system reset.** The Advisor lost its connection to the Agent on that system and notified the CSMs that all applications on that system were no longer active. Existing connections on that system were terminated but new connections were load balanced across the remaining servers.

12. **Application overload/system overload.** Scenarios where particular systems and applications were overloaded were also executed. In these scenarios, the Advisor and Agents returned recommendations that reflected the need to reduce the amount of new workload requests that were forwarded to these servers.

5 Best Practices and Design Principles

Based on the results of the test scenarios, this section describes the best practices and design guidelines that were learned during this white paper project.

5.1 General Considerations

5.1.1 High Availability

For high availability, there should be no single point of failure. During the testing, networking equipment and all zSeries software functions were configured to have at least two of everything of interest. In a real customer situation, the zSeries server itself should be replicated, with both Linux and z/OS operating system images distributed among the zSeries server boxes, so that no single zSeries server outage can take down all instances of Linux, z/OS, or any of the applications or middleware.

Similarly, when using Sysplex Distributor, a backup distributing TCP/IP stack or stacks should always be configured, in addition to the primary designated distributing stack. Sysplex Distributor functions may be performed on designated network operating system images (generally known as NET390 LPARs), or on the TCP/IP stacks that also host the application instances, according to customer installation policy and network design. The key point is that there should always be at least one backup distributing stack, and more than one backup stack is even better, since there is no performance penalty for being a backup distributing stack until the stack takes over actual distributing responsibility from the (formerly) active Sysplex Distributor stack.

Sysplex Distributor with or without MNLB forwarding agents can take Sysplex-related information into consideration when making load balancing decisions, information such as WLM displaceable capacity per operating system image, real-time server instance availability, and server instance-specific network performance.

5.1.2 Content Switching

Cisco Systems offers a complete line of content switching and SSL products ranging from switch integrated solutions to a complete line of modular appliances at differing price/performance levels. This white paper offers a best practice approach for both a Cisco Catalyst 6500 integrated content switching and SSL solution and a stand-alone appliance content switching and SSL solution. Both options offer the same level of redundancy and feature richness, therefore giving customers the opportunity to select the platform that offers the form factor and performance level that is appropriate for their network configuration.

The Content Switching Module (CSM) and SSL Services Module (SSLM) solution documented in this paper is Cisco's highest performance solution that offers integration with the Cisco Catalyst 6500 Series Switch. This combination offers industry-leading L2/L3 functionality with high-performance, feature-rich L4-7 switching and SSL. The Content Services Switch (CSS) 11503 with integrated SSL is a midrange performance solution for customers that do not require the high performance levels of the integrated solution and/or who prefer to deploy their advanced L4-7 services in an appliance form factor. Offering the same feature-rich L4-7 capabilities as the CSM and SSLM, the CSS also provides a scalable, modular platform that can hang off of any switching infrastructure.

5.2 Web Traffic

5.2.1 HTTP Serving

z/OS and WebSphere Application Server is the solution of choice for Web application serving, along with business data middleware such as CICS, IMS, MQ, and DB2. z/OS can also host the HTTP serving, given intelligent workload distribution in front of the HTTP servers, so that all processing can be local to a single z/OS image once the work arrives.

An alternative that can yield performance and cost efficiencies, however, is to host the HTTP serving and static content caching in a Linux on zSeries operating system image. Content-based routing is used for requests to the Linux HTTP servers, and the HTTP servers then use high-speed IP connectivity (Gigabit Ethernet or HiperSockets) to communicate with the WebSphere Application Servers on z/OS.

5.2.2 Sysplex Distributor/Multi-Node Load Balancer

Given the capabilities of a content-based load distribution solution such as CSS or CSM, Sysplex Distributor would not normally be the choice for distributing HTTP requests, because Sysplex Distributor at present has no support for content-based balancing decisions or knowledge of or support for HTTP session affinity – routing parallel or subsequent connections from the same client to the same HTTP server instance. However, there is one aspect of Web application workloads in which Sysplex Distributor is a recommended approach.

If a front-end tier of HTTP servers is used, such as the Linux HTTP servers tested for this paper, work between the Linux HTTP servers and the z/OS WebSphere Application Server instances also must be distributed. WebSphere Application Server (V4.0.1 and all later versions) provides a plug-in to most popular HTTP servers, which is aware of application affinity, and routes work to the appropriate application server during a client transaction. When the client is just starting a transaction, however, it makes no difference to which application server the request is directed.

In its basic mode, when there is no affinity required (single requests or at the start of a transaction), the plug-in will distribute requests in a round-robin fashion (evenly) among the available WebSphere Application Server instances. However, the plug-in can also be configured to send such requests to a “cluster address”. It is recommend that this cluster address be a Sysplex Distributor DVIPA, because Sysplex Distributor can take into account capacity on the application server operating system images and additional service policies defined by the installation (such as preferring certain server operating system images at certain times of the day), and because Sysplex Distributor is instantly aware when application server instances are stopped or started. This cluster address is a relatively simple configuration addition that is simply replicated on all HTTP server plug-ins.

5.2.3 CSS

For Web traffic, the CSSes sit in front of the Linux on zSeries servers and round-robin connections between the HTTP servers to provide an equal distribution of client HTTP Requests. HTTP responses are sent from the Linux on zSeries host to their default gateway, the Cisco Catalyst 6509 MSFC. The MSFC is configured with Policy Based Routing (PBR) to redirect HTTP response traffic to the server-side virtual interface on the active CSS to allow the CSS to map the connection back to the client. CSSs were also tested using source NAT.

For applications, it is critical that content switch failover occurs without disrupting existing client to server connections. The content switches are configured with ASR (Application Session Redundancy) to maintain session state information on both primary and secondary switches. In the event of a failure, the redundant switch will forward packets associated with the existing connection between client and server as soon as the underlying network reconverges.

The CSS monitors the health of the HTTP server by sending periodic HTTP connections to the servers. These connections are called “keepalives”. The CSS uses these keepalives to ensure the server is capable of responding to HTTP requests and the content has not become mangled or in any way changed. If an HTTP server cannot respond, all new client connections will be directed to the remaining HTTP services.

5.2.4 CSM

Similar to the CSS, the CSM is connected to the upstream router (MSFC) via a single VLAN. The CSM is positioned between the MSFC and the zSeries servers. The CSM can use round robin, or any load balancing algorithm (predictor) of the user’s choice, to provide the desired distribution of connections among the zSeries servers. The CSM can take advantage of either Policy-Based

Routing (PBR), a feature of the Cisco Catalyst Supervisor 2, or Source NAT to efficiently receive zSeries return traffic back and maintain proper bi-directional flows.

The CSMs are deployed in a redundant configuration with stateful connection failover, with one CSM as the active and the other as the standby. The active and standby CSMs are deployed in separate Cisco Catalyst chassis to protect also against a chassis failure or power shutdown. In an active and standby configuration each CSM has a unique VLAN management IP address, but the two CSMs share the same alias IP (similar to HSRP virtual IPs), virtual servers, server farms, and real server information. From the client-side and server-side networks, each CSM is configured identically (except for the management IP addresses) and the network sees the fault-tolerant configuration as a single CSM. The CSMs in this configuration will synchronize the sticky database and TCP connection state to provide redundancy. The CSM monitors the health of the HTTP server by sending HTTP probes to the Linux on zSeries HTTP servers. If an HTTP server cannot respond, all new client connections will be directed to the remaining HTTP services.

In scenarios where the target systems are inside a z/OS Sysplex, the z/OS Load Balancing Advisor can be deployed with the SASP support enabled on the CSM. This allows the CSM to obtain detailed sysplex load balancing recommendations that can help optimize load balancing. In environments where the target systems are not on z/OS, the CSM SASP support can still be enabled if the target systems are part of an EWLM environment where SASP support is present. To support redundancy for failover scenarios, both the active and standby CSMs can be configured to use SASP concurrently.

5.3 Non-Web Workloads

5.3.1 Sysplex Distributor/Multi-Node Load Balancer

.

Sysplex Distributor may be used by itself, or in conjunction with MNLB. The decision to use MNLB forwarding agents with Sysplex Distributor is based on inbound traffic volume, since MNLB forwarding agents bypass the Sysplex Distributor node to go straight to the application host, except for the initial connection request. The decision may be different for different applications, as long as each application has its own IP address (Distributed Dynamic VIPA).

Applications such as Telnet (TN3270) are characterized by relatively small inbound traffic volumes, on the order of a hundred bytes (as opposed to traffic back to the client, on the order of a thousand). It may be simpler for network management and problem determination for such applications only to use Sysplex Distributor, rather than Sysplex Distributor and MNLB together. On the other hand, applications such as FTP may be characterized by large inbound traffic volumes, and such applications should definitely consider using Sysplex Distributor with MNLB.

Note that initial deployment may use Sysplex Distributor alone. When the initial deployment is successful, and inbound volumes are seen to be significant, then MNLB exploitation may be added for that Distributed DVIPA at any time by a configuration change to the Sysplex Distributor stack.

The Cisco Catalyst 6500 switches should be configured to enable the MNLB forwarding agents from the beginning. There is no penalty for enabling the function, and it does not come into play until Sysplex Distributor is configured to communicate with the forwarding agents for specific applications. The Cisco equipment configuration may also need to be configured for Generic Routing encapsulation in a shared OSA environment, because GRE tunnels must be configured for each application hosting stack (though not for each application using the hosting stacks). Again, these definitions can be configured in anticipation of use, and need not be modified when new Sysplex Distributor-supported applications are deployed and deemed suitable for MNLB. When using GRE, the configuration must be modified when new z/OS operating system images are added as application hosting stacks. The configuration need not be modified for new z/OS TCP/IP stacks if OSAs are not shared and GRE is not being used.

5.3.2 CSS

For non-Web traffic such as FTP and TN3270 the CSS will use any available load balancing algorithm to balance connections between the z/OS servers to provide an equal distribution of client connections just like the HTTP requests. The CSS in the test scenario used a static VIPA address as the destination IP address for the client requests. Static 32-bit routes were added to the CSS to force traffic out the server-side interface to the server-side HSRP address on the Cisco Catalyst 6509 switches. This placed the routing decision upon the Cisco Catalyst MSFC, which is the best device to make this decision. This allows the CSS to handle server load balancing and the MSFC to handle network routing decisions using OSPF. This provides CSS to VIPA connectivity over multiple redundant LANs without disseminating the routing complications to the CSS. For non-balanced workload, protocols that are not load balanced on the CSS, response traffic for non-balanced requests is sent from the z/OS host to the default gateway, the Cisco Catalyst 6509 MSFC. The MSFC is configured with Policy Based Routing (PBR) to redirect non-balanced traffic to the server-side virtual interface on the active CSS to allow the CSS to map the connection back to the client. Any traffic not load balanced by the CSS is routed directly to the client. This maximizes the overall network bandwidth by using the MSFC to push CSS load balanced traffic through the CSS and all other traffic is directly routed to the clients.

For applications, it is critical for content switch failover to occur without disrupting existing client to server connections. The content switches are configured with ASR to maintain session state information on both primary and secondary switches. In the event of a failure, the redundant switch will forward packets associated with the existing connection between client and server as soon as the underlying network re-converges.

5.3.3 CSM

Positioned between the upstream router and the zSeries servers, the CSM can use any of the available load balancing algorithms (predictors) to provide the desired distribution of connections to the servers' applications. The CSM can take advantage of either Policy-Based Routing (PBR), a feature of the Cisco Catalyst Supervisor 2, or Source NAT to efficiently receive zSeries return traffic back and maintain proper bi-directional flow.

The CSMs are deployed in separate Cisco Catalyst 6500 chassis, to maximize the resiliency of the solution. They are configured in redundant active-standby mode and a dedicated VLAN (Fault Tolerant VLAN) is used to carry synchronization information between them, allowing for stateful failover. With stateful failover, stickiness state or full connection table's state can be exchanged. In a redundant configuration, the two CSMs share the same configuration, except for their VLAN management IP addresses (virtual server and server farms are exactly the same). Similar to an HSRP configuration, the CSMs share an alias IP address, which can be used by adjacent devices as a default gateway or next-hop. This configuration makes a pair of CSMs appear as a single load balancer to the rest of the network. The CSM monitors the health of the servers by sending application probes to the Linux on zSeries HTTP and z/OS servers. If a TN3270 or FTP server cannot respond, all new client connections will be directed to the remaining TN3270 or FTP services.

By default, the CSM keeps existing connections open when a server fails the probes (this can be changed by using the vserver option "failaction purge") or when a server is manually taken out of service (connections in this case can be manually torn down, if desired).

The CSMs can also be configured to obtain load balancing and state information via SASP using the z/OS Load Balancing Advisor. This allows the CSMs to take into account detailed sysplex information when performing load balancing. It also may reduce or eliminate the need to configure application probes on the CSM, as the z/OS Load Balancing Advisor will notify the CSMs of application state changes using SASP.

5.4 Workload-Specific Considerations

It isn't possible to cover all potential application aspects of workload balancing in this white paper. We have chosen three application protocols for detailed discussion: TN3270, HTTP, and FTP.

For all three application protocols we will discuss how they operate with the three major categories of load-balancing technologies:

- Sysplex Distributor/MNLB
- Directed mode – server NATing only combined with Policy Based Routing (PBR)
- Directed mode – server NATing and client NATing

We will discuss operational characteristics for the three chosen application protocols with or without SSL/TLS session security. For the SSL/TLS variations, we will cover the implications of offloading the SSL/TLS endpoint processing to the SSL module in the Cisco Catalyst 6500 or the CSS.

5.4.1 SSL/TLS offloading

Secure Sockets Layer (SSL) and Transport Layer Security (TLS) are in the context of this discussion assumed to be variations of the same underlying technology: securing individual TCP connections. SSL/TLS is applied to individual TCP connections. The use of SSL/TLS is generally application-specific: applications have to be programmed to support SSL/TLS. SSL/TLS applies to TCP connections only, not UDP applications.

The reasons for using SSL/TLS are, in general:

- Ability to encrypt data that is transported over the TCP connection so it never appears in clear text outside the client and server application programs.
- Ability for the client to authenticate who the server it connects to is. This is achieved by the server presenting its digital certificate to the client during the SSL/TLS handshake. In order to use SSL/TLS, a server certificate is required.
- Optionally allow for the server to authenticate who the client is that connects to it. This is achieved by the server requesting a digital certificate from the client during the SSL/TLS handshake. The use of client authentication is generally governed by configuration options on the server. On z/OS, a client certificate can be passed over the Security Access Facility (SAF) interface. If the certificate has been registered in the security database and associated with a known user ID, that user ID can be returned to the server program for user identification purposes without the user actually sending the user ID to the server. The user ID can also be checked against server-specific access profiles to authorize this user's access to this specific server – before any user data is transported over the connection. The TN3270 Express Logon feature requires use of client authentication.

The CPU overhead of doing SSL/TLS can be significant – especially for the SSL/TLS handshake that takes place when a new SSL/TLS TCP connection is being set up. Offloading the SSL/TLS processing to a specialized processor may therefore be an attractive technology. Another reason for offloading the SSL/TLS processing is for the load balancer to inspect clear-text traffic and perform content-based load balancing, which typically is used for Web traffic. In order for the load balancer to inspect the HTTP protocol details in an HTTPS request, the load balancer obviously needs to decrypt the HTTPS flows before it can inspect the details of the HTTP request. The way to achieve that is to have the HTTPS connection terminate at the load balancer and then proxy the request as a normal HTTP connection to the chosen target server.

When the SSL/TLS processing is offloaded, the SSL/TLS connection is established between the client and the SSL/TLS module in the Cisco Catalyst 6500 or CSS switch. The switch then establishes another connection between it and the real server. For the purpose of fully offloading the SSL/TLS processing, this connection between the switch and the real server typically is a non-SSL/TLS connection. The offloaded SSL/TLS processing includes authentication based on digital certificates. The client will be able to authenticate that it connected to a proper server, which just happens to be the SSL/TLS module in a CSS or Cisco Catalyst 6500 switch and not the real target server.

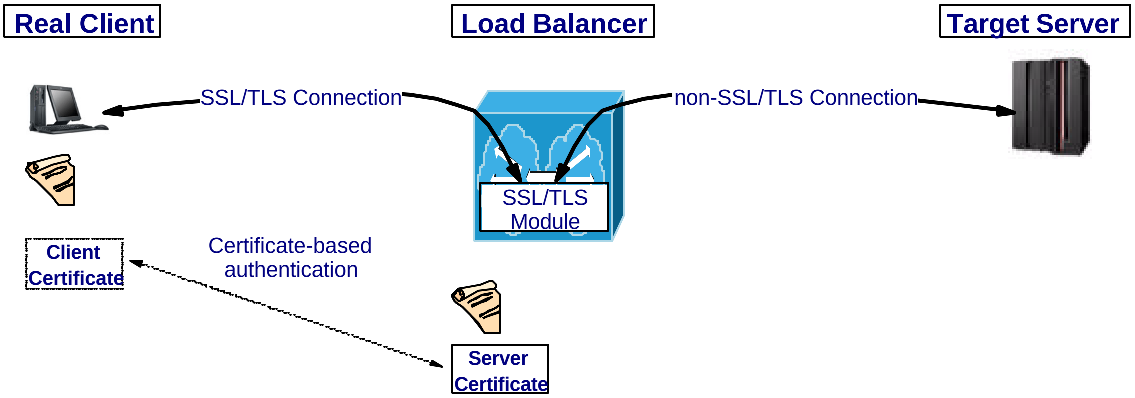


Figure 10 SSL/TLS offload basic principles

Obviously, if client authentication is required by the z/OS server application, the SSL/TLS session endpoints must be on the real client and on z/OS, and the SSL/TLS processing cannot be offloaded to an SSL/TLS module in the CSS or Cisco Catalyst 6500 switch. CSS and Cisco Catalyst 6500 are adding support to insert additional HTTP headers in the data stream on the proxied HTTP(S) connection to pass along the credentials from the original client's certificate. The connection between the load balancer and the target server may optionally be a separate SSL/TLS connection, but keep in mind that the client certificate that will be presented to the real server is not the certificate of the real client, but a certificate that resides on the load balancer, meaning that the real server will see one and the same client user for all connections that are established to it.

SSL/TLS for a given TCP connection is chosen using one of two general methods:

1. Two server port numbers are configured: one that doesn't use SSL/TLS and another that is configured so that all connections being established to that server port number enter SSL/TLS negotiation immediately after the connection has been established. All connections with such a port number are SSL/TLS enabled. This is also known as implicit SSL/TLS.

2. There is a single server port number to which clients connect and initially start out using non-SSL/TLS. As part of the early application protocol exchange, application protocol-specific options are exchanged that either negotiate use of SSL/TLS or not. If the partners agree on using SSL/TLS, then the SSL/TLS negotiation phase is entered and that connection from then on becomes an SSL/TLS-enabled connection. Otherwise, the connection remains a non-SSL/TLS connection. So some of the connections with the server port number are SSL/TLS enabled and others are not. This is also known as negotiated SSL/TLS.

TN3270 can use both of these two ways to establish a secure connection. A specific server port can be configured so all connections established with it immediately enter SSL/TLS negotiation. For such a port number, the SSL/TLS processing can be offloaded to the Cisco Catalyst 6500 or CSS SSL module. The TN3270 protocol also supports application protocol-specific negotiation so that a single server port number can be configured to support both SSL/TLS and non-SSL/TLS connections. For such a port number, the SSL/TLS processing cannot be offloaded to an outboard SSL module.

Web traffic always uses a separate port for HTTPS requests – the standard default is port 443 as opposed to the non-SSL/TLS port, which is port 80. Port 443 can be offloaded to the SSL module in the CSS or Cisco Catalyst 6500 and must be so if the load balancer is to perform content-based switching for HTTPS requests. Otherwise, the CSM or CSS cannot analyze the HTTP GET request content.

FTP is normally implemented using negotiated SSL/TLS where the FTP protocol recently has been extended with specific security negotiation commands and replies, primarily the AUTH command that is used by the client to request SSL/TLS or Kerberos-based security. When SSL/TLS is negotiated for an FTP session, the control connection is assumed to always be encrypted. The protection of the data connection is negotiated separately and may be either unprotected or protected. The use of implicit SSL/TLS for FTP was originally defined as part of SSL/TLS-enabling FTP and a separate FTP control connection port number reserved, but it is generally not used. Even with implicit SSL/TLS for the FTP control connection, it would be quite difficult for an offload SSL/TLS module to identify the FTP data connections. Offloading SSL/TLS FTP processing in general isn't possible.

5.4.2 Web traffic (HTTP and HTTPS)

A load balancer can typically balance this type of workload using either a connection balancing (layer-4) approach or an application layer (layer-5 and above) approach – also called content-based balancing because the load balancer analyzes the individual HTTP(S) requests (URLs) and their HTTP headers to determine which server instance is the best server to serve this specific request.

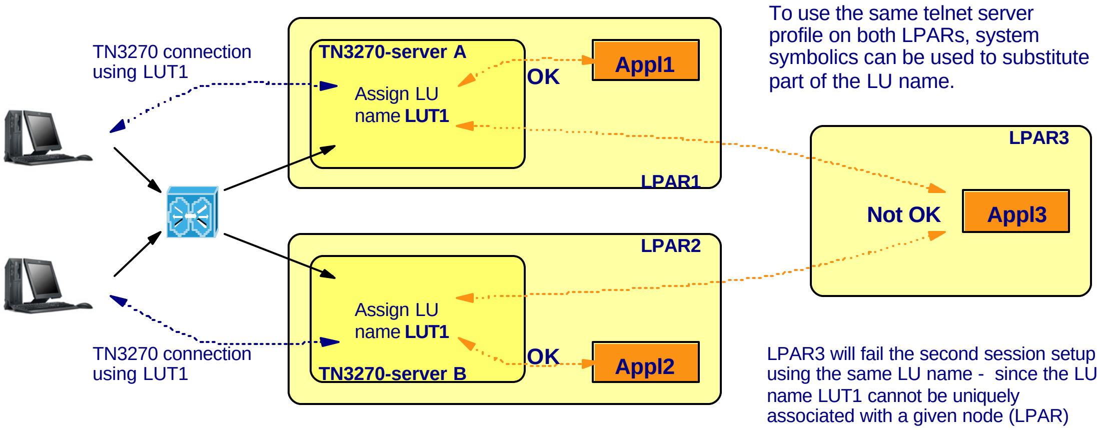
When the load balancer does content-based balancing, it completes the full TCP connection setup sequence with the client and reads the HTTP(S) request before it turns around and establishes a TCP connection with the real server and then proxies the HTTP(S) request to the chosen real server.

5.4.3 Telnet 3270 (TN3270)

TN3270 connections can in most cases be load balanced over multiple TN3270 server instances, but there are a few functions that require analysis before doing so:

- LU assignment – also known as LU nailing.
- Printer association requests.
- Server reconnect processing.

The main objective of assigning LU names in a TN3270 cluster environment is that you have to find a way to make sure that the same LU name is not being used by more than one TN3270 server at any point in time.



To use the same telnet server profile on both LPARs, system symbolics can be used to substitute part of the LU name.

LPAR3 will fail the second session setup using the same LU name - since the LU name LUT1 cannot be uniquely associated with a given node (LPAR)

Figure 11 TN3270 Server LU name assignment

If the SNA applications have no requirements on specific LU names being used by specific end users, then each target TN3270 server instance can be configured with a unique generic pool of LU names and LUs can be assigned freely out of those pools by each TN3270 server instance.

In many cases, there are requirements to have specific end user SNA sessions tied to specific LU names that in some way or another link the client endpoint of the SNA session to a specific physical location or a specific group of users. Terminal-based (LU-name based) security is still quite common.

If a pool of LU names are available for a given user community to be assigned and that pool is large enough to be divided among the TN3270 server instances, then define an LU group in each TN3270 server instance (the group can have the same name in all TN3270 server instances – such as TNGRP1) – and specify unique LU names in those groups in each TN3270 server. Client connections can be mapped to those groups using generic mapping or specific mapping where the client TN3270 emulator is configured to request an LU name in the LU group name (TNGRP1). Since the group name exists on all TN3270 instances, the request can be serviced by all of the TN3270 servers.

If there is a one-to-one relationship between a given client (client IP address, hostname, or user ID if SSL/TLS with client authentication is used), then load balancing should be used with care since two TN3270 server instances have no way of sharing information about which LU names are in use and which are not. One way to limit the risk of assigning the same LU name by two TN3270 servers would be to configure the load balancer to assign "stickiness" between the client IP address and the TN3270 VIP address and port number so that only the first connection from a given client IP address to the TN3270 cluster is load balanced and succeeding connections to the TN3270 VIP address and port are forwarded to the same TN3270 server instance.

Printer association is a function where the TN3270 emulator client first starts a terminal session (an LU type 2 session). The TN3270 server assigns an LU name to that LU2 session and informs the TN3270 emulator of the chosen LU name. A little later, the TN3270 emulator client starts a printer session (an LU Type 1 or 3 session) and passes the LU name of the LU2 session on to the TN3270 server requesting a printer LU name that in some way or another is associated with that LU2 name. Obviously, if the printer request is sent to a different TN3270 server instance than the instance that set up the LU2 session, that TN3270 server will not know anything about the associated LU2 LU name and the printer association request will fail. The way to avoid this situation from occurring is again to use stickiness in the load balancer so that all connection requests from a given client IP address to the TN3270 server VIP address and port number will be forwarded to one and the same TN3270 server instance.

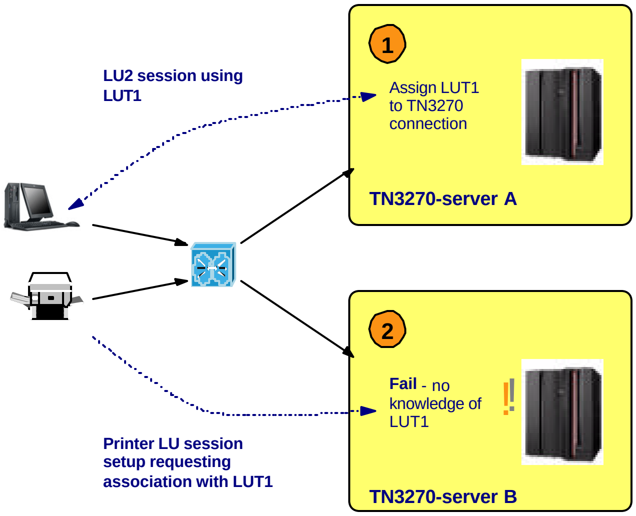

1
LU2 session using LUT1
Assign LUT1 to TN3270 connection
TN3270-server A
2
Fail - no knowledge of LUT1
!
Printer LU session setup requesting association with LUT1
TN3270-server B

Figure 12 TN3270 Server printer association

To use the TN3270 server reconnect functions, the same approach as for printer association needs to be applied: stickiness so that all connection requests from the same client IP address go to one and the same TN3270 server instance.

Both the CSS and the CSM support stickiness, and so does Sysplex Distributor in z/OS V1R5.

As we mentioned earlier in this paper, the TN3270 server may be configured to use the client IP address to map various TN3270 server objects to the TN3270 server connection, such as an LU name, a primary SNA application name, a USS table name, etc.

If such client IP addresses are used to perform this mapping, the load balancer must be configured so it presents the real client IP address to the TN3270 server, meaning that it cannot be configured to use directed mode – server NAT and client NAT. Directed mode server NAT combined with Policy Based Routing will work fine.

Also as mentioned earlier, if a specific TN3270 server port is meant for implicit SSL/TLS, client authentication is not needed on the TN3270 server, and client IP addresses are not used for TN3270 server mapping purposes, then SSL/TLS processing can be offloaded to an SSL/TLS module in the CSM or CSS. Otherwise SSL/TLS processing should be done on the TN3270 server node.

5.4.4 File Transfer Protocol (FTP)

FTP is one of the more complex application protocols to handle in a load balancing scenario.

An FTP session uses two TCP connections: an FTP control connection that stays connected for the duration of the FTP session, and an FTP data connection for transferring a file. FTP generally sets up a data connection for each file that is to be transferred during the FTP session, so an FTP session may use as many data connections as the number of files that are transferred during that FTP session.

The FTP control connection is established to the FTP server control port number, which is port 21. The FTP client sends FTP commands to the FTP server over the control connection and the FTP server responds with FTP replies over the control connection.

When the client and the server are ready to transfer a file, the FTP client sends a command to the FTP server instructing it how the client wants the data connection to be established. There are two basic ways the data connection can be established: active mode and passive mode.

Active mode means that the data connection is initiated from the FTP server back to the FTP client. The client starts the process by sending a PORT command to the FTP server and in that PORT command is the IP address and port number the FTP server is to establish the data connection to. Normally the IP address in the PORT command will be the client IP address, but the FTP protocol does allow for it to be an IP address that is different from the one the FTP server sees as the client IP address on the control connection. The FTP server will establish the data connection from port 20 on the server node.

Passive mode means that the data connection is initiated from the FTP client to the FTP server – the same direction as the control connection. This is sometimes referred to as firewall-friendly FTP since both control connection and data connection are established in the same direction: from the client to the server. Firewall-friendly FTP is more acceptable from a firewall point of view when the client resides in an intranet, while the server resides on the Internet (you generally accept connections outbound from the intranet, but you are very careful accepting connections coming from the Internet into your intranet). The FTP client starts the process by sending a PASV command to the FTP server, and the server replies back with the IP address and port number the client is to establish the data connection to. Normally the IP address in the PASV reply will be the same server IP address to which the client established the control connection, but again the FTP protocol does allow it to be a different IP address.

The main issue for FTP and load balancing is that we have to make sure the data connection ends up with the same FTP server as the one for which the control connection was established. The way to achieve this is to load balance the FTP control connection request only, and from then on implement technologies that will ensure the data connection ends up with the same FTP server instance as the one that was chosen for the control connection.

In the Sysplex Distributor / MNLB case, this is handled by the Sysplex Distributor logic. For both active mode and passive mode FTP data connections, Sysplex Distributor is informed by the target stacks in the Sysplex about preparation for a new FTP data connection so that Sysplex Distributor can record the affinity for a new data connection before it is established, and multicast it to the forwarding agents to make sure the data connection packets are routed correctly to the proper target stack. To handle passive mode FTP data connections, the Sysplex Distributor DVIPA must be defined with the SYSPLEXPORTS option. SD/MNLB works for both SSL/TLS and non-SSL/TLS FTP sessions.

In the CSM case, we chose an example based on both server NATing and client NATing to illustrate one of the more complex scenarios. In general, CSM load balancing does not get involved with the FTP data connections at all, but allows those connections to be established directly between the real FTP client and the target FTP server using the target FTP server's real server IP address, and not the CSM VIP FTP service IP address.

In the example shown here, the CSM load balancer uses the 9.42.89.216 IP address for the FTP workload. The FTP client will send a connection setup request to 9.42.89.216 port 21. The CSM will turn around and use its 9.42.89.215 IP address as the client source and forward the connection setup request to the chosen server's IP address – in this example 9.42.88.9 and port 21. This connection represents the FTP control connection. The client believes it communicates with 9.42.89.216, and the server believes it communicates with 9.42.89.215. Neither the client nor the server knows the real IP addresses of the other part.

For active mode data connections, the client will send a PORT command to the server with its real IP address and a port number for the server to send a data connection setup request to. The CSM load balancer does not intercept this exchange on the control connection, but lets it pass unchanged, meaning that the server is now instructed to set up the data connection to another IP address (9.27.18.206) than what it knows as the client's IP address (9.42.89.215). The FTP protocol allows this to happen – it is known as three-way FTP proxy. The z/OS FTP server has some configuration options to control whether the z/OS FTP server should allow this to take place. The reason for those options is that three-way FTP proxy is known as a function that can be misused for certain malicious purposes and some installations have chosen to disable its use on z/OS. In order to load balance FTP workload through a load balancer that is based on directed mode (either server NAT with Policy Based Routing or server NAT plus client NAT), those options must be set on the z/OS FTP server to not disallow three-way proxy. You must configure the FTP server with PORTCOMMANDIPADDR UNRESTRICTED, which is the default setting of this option.

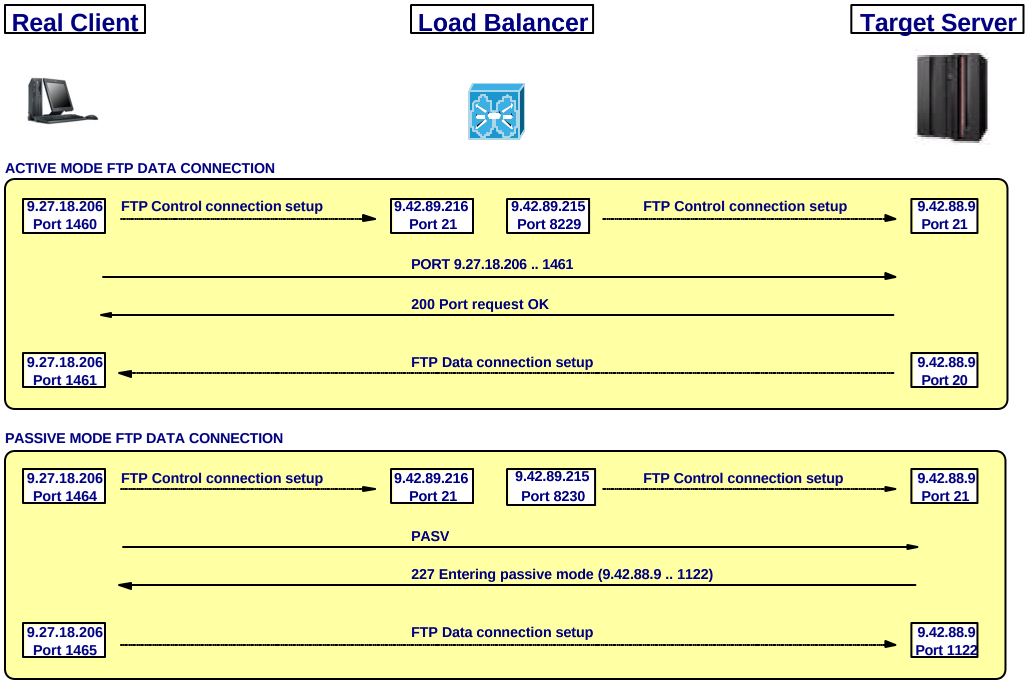


Figure 13 FTP data connections

For passive mode data connections, the client will send a PASV command to the server. The server will reply with the IP address and port number to which the client is to send the data connection setup request. Again, the CSM load balancer will not intercept these exchanges, so the client will now be instructed to send the data connection setup request to an IP address that is different from the IP address to which the control connection is established. This is again valid according to the FTP protocol support of three-way FTP proxy. The FTP server believes the client resides on IP address 9.42.89.215, so when the client sends the data connection setup request the server will see the source IP address of the data connection as an IP address that is different from the source IP address of the control connection. Again, the z/OS FTP server has configuration options to disallow that from happening, and you need to make sure those options are not enabled on your z/OS FTP server by specifying PASSIVEDATACONN UNRESTRICTED, which is the default setting of this option.

If you configure the CSM load balancer to operate in server NAT mode only (combined with Policy Based Routing), then you can set both the PORTCOMMANDIPADDR and the PASSIVEDATACONN options as your security policies require. When server NATing only is used, the IP address on the PORT command for active mode data connections will be the same IP address as the one the server knows for the control connection – and for passive mode data connections, the data connection will come from the same client IP address as the one the server knows for the control connection.

Be aware that some FTP clients will not allow the data connection to be established between IP addresses other than the addresses that are in use for the control connection. A general recommendation is to test FTP workload through any type of load balancer solution thoroughly before moving such a solution into production. Remember that testing must include both active mode and passive mode data connections, and if use of SSL/TLS FTP connections is needed, then testing must also include active mode, passive mode, and extended passive mode with SSL/TLS FTP connections before deciding on how such a solution is to be supported.

Since the CSM load balancer doesn't get involved at all with the FTP data connection setup and since it doesn't rely on the ability to investigate the data that is exchanged on the FTP control connection, it isn't sensitive to use of FTP session encryption technologies. It is fully possible to load-balance SSL/TLS enabled FTP sessions through a CSM load balancer that operates in directed mode. There is no requirement to use extended passive mode as there normally is in order to set up SSL/TLS sessions through NATing firewalls.

In fact, with directed mode load balancing, extended passive mode data connections are not supported at all. This is the case for both the server NAT/client NAT case and the server NAT only case. Extended passive mode is a recent enhancement to the FTP protocol that is intended to ease the setup of FTP data connections through NATing firewalls and for that purpose it works very well, especially in allowing SSL/TLS-enabled FTP sessions through such NATing firewalls. However, it doesn't work well with load balancers that operate in directed mode since a load balancer doesn't do static NATing, but is able to NAT a given VIP to one of the additional target server IP addresses. There is not a one-to-one relationship between the IP address the client connects to and the real server IP address. The extended passive mode command (EPSV) is sent from the FTP client to the server in much the same way as a normal passive mode command (PASV). The main difference is that the extended passive mode reply does not include an IP address – the requirement for using extended passive mode is that the data connection must be established by the client to the same IP address to which the control connection was established. Since that IP address is the load balancer VIP address and the load balancer doesn't know which FTP server target to forward that data connection setup request to, the extended passive mode data connection setup fails.

Extended passive mode FTP works without problems with SD/MNLB.

5.5 Positioning

The following are some of the considerations to be made in deciding which load balancing solution(s) to deploy. Note that as long as each workload is assigned a separate IP address (as visible to the clients), the workload balancing decision may be made for each application, based on its characteristics, independent of other applications, and it is not necessary to deploy the same workload balancing solution for all applications.

5.5.1 Where and How Application Server Instances Are Deployed

Sysplex Distributor has as its scope TCP connections to a single z/OS Sysplex. This means that if application server instances are deployed on other platforms (such as Linux on zSeries), or workload is to be balanced across more than one z/OS Sysplex, then Sysplex Distributor cannot be used for balancing work to those server instances, and an external solution such as CSM or CSS must be used.

The z/OS Load Balancing Advisor technology supports all transport protocols, but has as its scope also a single z/OS Sysplex. This means that CSM in combination with the z/OS Load Balancing Advisor can balance workload across z/OS systems in a single z/OS Sysplex. The CSM can support other platforms, but it cannot in this scenario balance incoming workload across platforms outside the z/OS Sysplex in which the z/OS Load Balancing Advisor executes.

5.5.2 Server Application Transport Protocol

Sysplex Distributor balances work carried in TCP connections only. Applications that use UDP cannot be balanced with Sysplex Distributor, and another solution must be used.

5.5.3 Content-Based HTTP Workload Balancing

When the content of the request should determine or influence where the request should go, an external load balancing solution such as CSM or CSS must be used. Note that if Secure HTTP (HTTPS, or SSL/TLS) is to be included, the load balancing node must be accompanied by an SSL/TLS offload mechanism.

The outboard workload balancing solution can also be accompanied by a caching appliance for improved performance in serving static content.

5.5.4 Administrative control

Sysplex Distributor is configured and managed on the z/OS platform as part of the normal CS for z/OS TCP/IP configuration, typically by the mainframe network administrators.

If Sysplex Distributor is used in combination with the Cisco MNLB forwarding agents, the configuration of the MNLB forwarding agents is a static configuration on the switch that doesn't change based on which application services are to be load balanced. Configuration of which applications services are to be load balanced is done in Sysplex Distributor on z/OS.

CSS and CSM are configured and managed on the Cisco equipment as part of normal network equipment configuration, typically by network administrators.

If CSM is used in combination with the z/OS Load Balancing Advisor, the configuration of the z/OS Load Balancing Advisor and the z/OS Load Balancing Agents are static configurations that do not change based on which applications services are to be load balanced. Configuration of which applications services are to be load balanced is done on the CSM equipment itself.

If balancing workload to a z/OS Sysplex is under control of the mainframe network administrators, a Sysplex Distributor based solution allows for that to happen without having to extend the mainframe network administrators' scope of control to include network equipment. If balancing workload to a z/OS Sysplex is seen as balancing workload to just another operating system platform and is to be under control of the network administrators, a CSS or CSM implementation model allows for that to happen without extending the network administrators' scope of control to the z/OS network configuration.

5.5.5 Workload Volume

Sysplex Distributor with MNLB scales well in a multi-application environment. Any z/OS TCP/IP stack may be used to distribute workload for an application, and where multiple applications are being balanced, the distribution responsibility may be spread across multiple z/OS TCP/IP stacks.

Both CSS and CSM have the capability to balance large workloads. The CSM may be preferred for higher-performance workloads.

5.5.6 Quality of the load balancing decision

When the workload balanced application uses TCP connections and all server instances are hosted within the same z/OS Sysplex, Sysplex Distributor can make use of Sysplex information, such as capacity from z/OS Workload Manager (WLM), real-time server instance availability, and service policies created by the installation in the Service Policy Agent, to balance work from clients across the server population according to available displaceable capacity and such information as the client IP address or time of day

CSM and CSS support both TCP and UDP application workload. In addition they are both able to perform content inspection for HTTP or HTTPS workload. Both are external load balancers (the load balancing decision point is located outside the z/OS Sysplex) and the load balancers in general do not have access to real-time system capacity or server instance availability. The workload balancing decision is typically made based on static weights for the server instances combined with availability information that is obtained using various types of probing or polling mechanisms.

CSM in combination with the z/OS Load Balancing Advisor provides support for the same types of application workload as CSM and CSS in general for server instances residing in a single z/OS Sysplex. CSM in this environment is still an external load balancer (the load balancing decision point is located outside the Sysplex). The load balancer now has near real-time information available through the SASP protocol about system capacity as reported by the z/OS Workload Manager (WLM), network interface and server instance availability, along with server instance health (how well does the server instance cope with the workload that is directed towards it). CSM in combination with the z/OS Load Balancing Advisor is able to make better load balancing decisions for workload entering a z/OS Sysplex than an external load balancer operating without use of the z/OS Load Balancing Advisor.

5.5.7 Positioning sum up

The following section is a quick guide to determine which load balancing technology could be used for your specific z/OS Sysplex workload:

1. Where HTTP workload is to be balanced based on content of the HTTP requests, an external load balancer capable of doing content inspection, such as both the CSM and the CSS, should be used.

    a. If HTTPS workload is to be included, the external load balancing node must be accompanied by an SSL/TLS offload technology.
    b. A content switch that load balances HTTP(S) workload can be combined with a cache appliance for improved overall performance.

2. UDP workload balancing must be done using an external load balancer, such as the CSM or CSS.

3. Remaining TCP-based workload into a z/OS Sysplex can be deployed using either Sysplex Distributor or an external load balancer. Main decision criteria are:

    a) Quality of decision: Sysplex Distributor uses real-time information, CSM in combination with the z/OS Load balancing Advisor uses near real-time information, external load balancers in general uses information that is obtained through probing and polling.
    b) Administrative control: Sysplex Distributor provides for the z/OS systems programmer to maintain all dynamic configuration information about which services to load balance in the z/OS TCP/IP configuration. External load balancers, including CSM in combination with the z/OS Load Balancing Advisor, allow the network engineers to maintain all dynamic configuration information about which services to load balance in the external load balancers local configuration.

See the following table for detailed guidelines on which technology to use.

Features or Considerations	Sysplex Distributor	External Load Balancers	External Load Balancers with SASP
How is the solution administered/configured.	Initial setup may require some interactions with the network (dynamic routing protocols, DNS updates for Dynamic VIPAs, etc.). Ongoing administration (adding/removing target server applications and or systems) typically confined within z/OS systems.	Initial setup/configuration on load balancer, some configuration on z/OS may be required. Ongoing administration should be mostly confined to the load balancer (although z/OS configuration may be necessary when adding new target systems, etc.)	Initial setup/configuration on load balancer and on z/OS. Ongoing administration may need to be performed on both the load balancer and the z/OS systems.
When is the server instance decision made?	Connection Setup (in line Syn segment)	Connection Setup (in line Syn segment)	Connection Setup (in line Syn segment)
Support for TCP and UDP applications	TCP only	Depends on the load balancer implementation	Depends on the load balancer implementation (SASP supports both TCP and UDP)
Extra Network Flows	Yes for inbound traffic. Inbound traffic must traverse the Sysplex Distributor node. If Sysplex Distributor is configured as Service Manager for Cisco routers then the inbound traffic can flow directly to the target application. No for outbound traffic	Depends on the load balancer implementation (can be avoided if the load balancer is implemented as part of a router/switch)	Depends on the load balancer implementation (can be avoided if the load balancer is implemented as part of router/switch)
Support for affinities between TCP connection requests based on data content	No, support does however exist for timer based affinities	Depends on implementation, some support affinities for HTTP/HTTPS requests by inspecting data content (correlating cookies, jsessionid)	Depends on implementation, some support affinities for HTTP/HTTPS requests by inspecting data content (correlating cookies, jsessionid)
Network Address Translation	Not needed (client and server IP addresses are not modified)	May be required by some implementations (client and/or server IP addresses may be translated)	May be required by some implementations (client and/or server IP addresses may be translated)
Support for IPv6	Yes	Depends on the load balancer implementation	Depends on the load balancer implementation (SASP supports both IPv4 and IPv6)
z/OS WLM recommendations	Yes (System level and Server level WLM recommendations are available)	Depends on the load balancer implementation	Yes (System level and Server level WLM recommendations)
z/OS Network QoS recommendations	Yes (based on z/OS QoS policy)	No	No
z/OS TCP/IP server health information	Yes	No	Yes
Detection of target application and/or target system state changes (active or inactive).	Yes, application and system state changes are detected in near real-time fashion.	Depends on the load balancer implementation	Yes, the z/OS load balancing Advisor and Agents detect application and system state changes within a configurable time period (60 seconds by default). How quickly

			external load balancers become aware of these changes depends on several factors: 1) if the load balancer is using a push model with SASP, the load balancing Advisor will send a notification of a state change as soon as it detects it. 2) if the load balancer is using a poll model with SASP, it will depend on the load balancer's polling interval the load balancer may also have additional mechanisms for detecting application/system state changes , which may provide for faster detection of these changes.
High availability solution (load balancing continues even if the primary load balancing component becomes unavailable)	Yes, one or more backups can be configured to allow for dynamic take over in cases where the TCP/IP stack or system that is acting as the distributor fails.	For failures to the load balancer, it depends on the load balancer implementation. Some solutions provide for backup load balancers that can dynamically take over load balancing responsibilities in cases of failures.	For failures to the load balancer, it depends on the load balancer implementation. Some solutions provide for backup load balancers that can dynamically take over load balancing responsibilities in cases of failures. The z/OS Load Balancing Advisor and Agents can be configured for high availability to minimize the impact of an Advisor, Agent or system failure).

6 Conclusions

This white paper is intended to provide an insight into designing and integrating Cisco Systems' content switching and SSL technology with IBM's Sysplex Distributor in zSeries OSA-Express environments for distributing various types of workloads.

Both HTTP and non-HTTP workloads were tested in combination, to ensure interoperability and coexistence. Sysplex Distributor and MNLB may be used for some workloads at the same time that CSS or CSM is used for other workloads. CSS and CSM have the capability to inspect the content of HTTP requests, and are preferred when content-based workload balancing can result in better service to the clients. CSS and CSM can also balance work among servers that are not deployed on z/OS. When all application servers are in the same z/OS Sysplex, CSM with SASP and the z/OS Load Balancing Advisor or Sysplex Distributor can make use of Sysplex information such as server node capacity and server availability to tune the balancing of work among available servers.

As with all network design, various approaches are applicable. One should bear this in mind when applying these principles to production networks.

7 Bibliography

The following is a list of useful document references that may be consulted for further details:

- OSPF Design and Interoperability Recommendations for Cisco Catalyst 6500 and OSA-Express Environments, a white paper available at both Cisco and IBM Web sites. The IBM URL is: http://www-1.ibm.com/servers/eserver/zseries/networking/pdf/ospf_design.pdf
- Leveraging z/OS TCP/IP Dynamic VIPAs and Sysplex Distributor for Higher Availability: *http://www-1.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130165.pdf*
- z/OS Communication Server IP Configuration Reference, Version 1 Release 7, IBM publication number SC31-8776-08: http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/download/F1A1B450.pdf
- z/OS Communication Server IP Configuration Guide, Version 1 Release 7, IBM publication number SC31-8775-07: http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/download/F1A1B350.pdf
- Datacenter Networking: Infrastructure Architecture: http://www.cisco.com/application/pdf/en/us/guest/netsol/ns304/c649/ccmigration_09186a008014f2c6.pdf
- Datacenter Networking: Optimizing Server and Application Environments: http://www.cisco.com/application/pdf/en/us/guest/netsol/ns304/c649/ccmigration_09186a008014edf2.pdf
- Datacenter Networking: Integrating Security, Load Balancing, and SSL Services using Services Modules: http://www.cisco.com/application/pdf/en/us/guest/netsol/ns304/c649/ccmigration_09186a008014efaf.pdf
- Installation and Configuration Guides for the Cisco Catalyst 6500 Series Switches: http://www.cisco.com/en/US/partner/products/hw/switches/ps708/products_installation_and_configuration_guides_list.html
- Technical Documentation for the Cisco CSS 11500 Series Content Services Switches: http://www.cisco.com/en/US/partner/products/hw/contnetw/ps792/prod_technical_documentation.html
- HTTP header insert documentation – SSLM: http://www.cisco.com/en/US/partner/products/hw/switches/ps708/products_module_configuration_guide_chapter09186a00801f33c4.html#1241063

8 Appendix A – Example Configuration Files

Please note that only the configuration files from the Cisco equipment and those portions of the z/OS TCP/IP configuration relating to Sysplex Distributor, are provided here. Additional configuration will naturally be required for any specific installation. (For example, OSPF was used as the dynamic routing protocol, and must be configured in z/OS and Linux TCP/IP, as well as in the Cisco equipment, but other routing protocols may be used if desired, so OSPF configuration was not shown here.)

8.1 Sysplex Distributor/MNLB

This section includes the configurations for the Sysplex Distributor/MNLB test scenarios.

8.1.1 z/OS TCP/IP Configuration

Activating Sysplex Distributor on z/OS TCP/IP requires several additional configuration statements in the profile for the TCP/IP instance selected to be the normal routing stack for the application, as well as an additional statement on each of the TCP/IP stacks selected as backup routing stacks. No additional configuration is required for Sysplex Distributor on z/OS TCP/IP stacks that host the application instances (unless these are also the designated routing or backup routing stacks).

The z/OS TCP Dynamic VIPA configuration statements for the Sysplex Distributor routing stack (MVS001 in the figures) are as follows:

Certain IPCONFIG statements are required or recommended for all TCP/IP stacks participating in Sysplex Distributor, whether routing stack, backup routing stack, application hosting (target) stack, or a combination of these roles:

```
;
;*******************************************************************
;* IP Config Statements and Dynmaic XCF
;*******************************************************************
;
IPCONFIG DYNAMICXCF 9.42.88.161 255.255.255.248 2
IPCONFIG DATAGRAMFWD
IPCONFIG SOURCEVIPA
IPCONFIG SYSPLEXROUTING
;
```

The first statement in the VIPADYNAMIC block for the routing stack is the VIPASMPARMS statement, which defines the multicast group and port with which the stack will communicate with the Cisco Catalyst 6509 forwarding agents. Note the IP address (224.0.1.2) and port number (1637) – they will also appear in the Cisco configuration…

```
;
;*****************************************************************
; Dynamic and Distributed VIPA Statements
;*****************************************************************
;
VIPADYNAMIC
;
  VIPASMPARMS SMMCAST 224.0.1.2 SMPORT 1637
;
```

The following Dynamic VIPA configuration statements define Distributed DVIPAs for TN3270 and the other applications. The "SERVICEMGR" keyword on the VIPADEFINE statement is the only addition for integrating MNLB. For reference, the Dynamic XCF addresses of the target stacks on MVS062 and MVS154 are 9.42.88.163 and 9.42.88.164.

```
;
;---------------------------------------------------------------
; DISTRIBUTED VIPAS FOR TN3270 (SECURE AND NON-SECURE)
;---------------------------------------------------------------
;
  VIPADEFINE MOVEABLE IMMED SERVICEMGR 255.255.255.248 9.42.88.169
  VIPADIST   DEFINE  9.42.88.169 PORT 23 523 1923
  DESTIP 9.42.88.163
         9.42.88.164
         9.42.88.161
;
;---------------------------------------------------------------
; DISTRIBUTED VIPAS FOR FTPS
;---------------------------------------------------------------
;
  VIPADEFINE MOVEABLE IMMED SERVICEMGR 255.255.255.248 9.42.88.171
  VIPADIST DEFINE SYSPLEXPORTS 9.42.88.171 PORT 20 21
  DESTIP 9.42.88.163
         9.42.88.164
         9.42.88.161

;
;---------------------------------------------------------------
; DISTRIBUTED VIPAS FOR LDAP
;---------------------------------------------------------------
;
  VIPADEFINE MOVEABLE IMMED SERVICEMGR 255.255.255.248 9.42.88.172
  VIPADIST DEFINE 9.42.88.172 PORT 389
  DESTIP 9.42.88.163
         9.42.88.164
;
ENDVIPADYNAMIC
;
```

Once again, only the VIPASMPARMS statement and the SERVICEMGR keyword on the VIPADEFINE statements are unique to using Cisco MNLB forwarding agents. The other configuration comprises basic stack and Sysplex Distributor definitions.

The Sysplex Distributor routing stack function is also backed up on the other z/OS TCP stacks. This is done with the addition of a single configuration statement for each Distributed DVIPA within the VIPADYNAMIC/ENDVIPADYNAMIC block in the profile:

```
;
;****************************************************************
; Dynamic and Distributed VIPA Statements
;****************************************************************
;
VIPADYNAMIC
;
;---------------------------------------------------------------
; BACKUPDISTRIBUTED VIPAS
;---------------------------------------------------------------
;
  VIPABACKUP 50 9.42.88.169 ; TN3270
  VIPABACKUP 50 9.42.88.171 ; FTP
  VIPABACKUP 50 9.42.88.172 ; LDAP
;
. . . <any other Dynamic VIPA configuration for the stack>
;
```

```
ENDVIPADYNAMIC
;
```

Note that the '50' on the VIPABACKUP statements is a "rank", or order of backing up the Dynamic VIPA. If there is more than one backup stack for a particular Dynamic VIPA, then the VIPABACKUP statements for each DVIPA should have different ranks on different backup stacks. The active backup stack with the highest rank will take over the DVIPA in case of failure of the TCP or operating system image where the DVIPA is currently active.

8.1.2 Cisco Catalyst 6509 Switches

The following Cisco Catalyst 6509 configuration (NEP6509A in the figures) was used for Sysplex Distributor/MNLB testing. The configuration related to the forwarding agents is annotated within the configuration, as <text within angle brackets>.

```
hostname NEP6509A
!
no ip igmp snooping
!
```

<The following statements set up the multicast address and port for the forwarding agents. Sysplex Distributor will multicast its Distributed DVIPAs to this address and port, so the switches are aware of where to go for connection routing information.>

```
ip multicast-routing
ip casa 1.1.1.1 224.0.1.2
 forwarding-agent 1637
```

<End of basic forwarding agent setup>

```
!
interface Port-channel1
 description Trunk Connection to 6509B - Gig 1/1 and Gig 3/13
 no ip address
 no logging event link-status
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport nonegotiate
!
```

<Because the Open Systems Adapter (OSA) ports on the zSeries are shared among the z/OS images, a Generic Routing Encapsulation tunnel is required to each z/OS image. The tunnel destination address is a Virtual IP Address (VIPA) configured on each of the z/OS TCP/IP stacks. See the above z/OS TCP/IP definitions for reference. >

```
interface Tunnel1
 description GRE tunnel to MVS001
 ip address 4.4.4.1 255.255.255.252
 no ip mroute-cache
 no logging event link-status
 tunnel source 9.42.89.129
 tunnel destination 9.42.88.1
!
interface Tunnel62
 description GRE tunnel to MVS062
 ip address 5.5.5.1 255.255.255.252
 no logging event link-status
 tunnel source 9.42.89.129
 tunnel destination 9.42.88.9
!
interface Tunnel154
 description GRE tunnel to MVS154
 ip address 6.6.6.1 255.255.255.252
 no logging event link-status
 tunnel source 9.42.89.129
 tunnel destination 9.42.88.13
!
```

<End of GRE tunnel definitions>

```
interface GigabitEthernet1/1
 description First Fiber for Port Channel
 no ip address
 no logging event link-status
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport nonegotiate
 channel-group 1 mode active
 channel-protocol lacp
!
interface GigabitEthernet1/2
 description VLAN14 - Connection to Backbone Network
 no ip address
 no logging event link-status
 switchport
 switchport access vlan 14
 switchport mode access
 no cdp enable
!
interface GigabitEthernet3/1
 description VLAN15 - GIGE2E60 to NIVT Sysplex
 no ip address
 no logging event link-status
 switchport
 switchport access vlan 15
 switchport mode access
 no cdp enable
!
interface GigabitEthernet3/13
 description Second Fiber for Port Channel
 no ip address
 no logging event link-status
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport nonegotiate
 channel-group 1 mode active
 channel-protocol lacp
!
interface GigabitEthernet3/15
 description VLAN14 - Connection to AWM Clients
 no ip address
 no logging event link-status
 switchport
 switchport access vlan 14
 switchport mode access
 no cdp enable
!
interface Vlan14
 description VLAN14 - Connection to Site and AWM Clients
 ip address 9.42.89.250 255.255.255.240
 no logging event link-status
!
interface Vlan15
 description VLAN15 - Gigabit Ethernet 2E60 to Sysplex - Real
 ip address 9.42.89.129 255.255.255.248
 no ip redirects
 ip pim dense-mode
 ip igmp join-group 224.0.1.2
 no ip mroute-cache
 ip ospf cost 1
 no logging event link-status
!
interface Vlan25
 description VLAN25 - Gigabit Ethernet 2E70 to Sysplex - Virtual
 ip address 9.42.89.137 255.255.255.248
 no ip redirects
 ip pim dense-mode
 ip igmp join-group 224.0.1.2
 no ip mroute-cache
 ip ospf cost 1
 no logging event link-status
!
router ospf 1
 router-id 9.42.89.250
 log-adjacency-changes
 area 1.1.1.1 stub no-summary
 redistribute connected subnets
 redistribute static subnets
 network 1.1.1.1 0.0.0.0 area 0.0.0.0
 network 9.42.89.129 0.0.0.0 area 1.1.1.1
 network 9.42.89.137 0.0.0.0 area 1.1.1.1
 network 9.42.89.250 0.0.0.0 area 0.0.0.0
 maximum-paths 5
!
```

<When Sysplex Distributor picks a target hosting stack for an incoming connection for an application it manages, it notifies the forwarding agent of the destination by specifying the target stack's Dynamic XCF address. The following three routes to the Dynamic XCF addresses for the three z/OS images (MVS001, MVS062, and MVS154) identify the respective tunnel to be used.>

```
ip classless
ip route 9.42.88.161 255.255.255.255 Tunnel1
ip route 9.42.88.163 255.255.255.255 Tunnel62
ip route 9.42.88.164 255.255.255.255 Tunnel154
```

8.2 CSM

8.2.1 CSM Configurations

The CSMs were tested with Policy-Based Routing (PBR) and Network Address Translation (NAT). Both of the respective configurations are shown here for NEB6509A. These configurations are for the case when the CSM was distributing HTTP, FTP, and TN3270 traffic. (See above for the forwarding agent and GRE tunnel configuration when Sysplex Distributor is handling the latter two applications.)

8.2.1.1 CSM PBR

```
hostname NEP6509A
!
module ContentSwitchingModule 5
!
 vlan 40 server
  description VLAN40 - Client side VLAN
  ip address 9.42.89.211 255.255.255.240
  gateway 9.42.89.220
  alias 9.42.89.215 255.255.255.240
!
 probe ICMP icmp
  interval 5
  failed 60
  receive 2
!
```

<The following statements define the server farms (collections of servers) for FTP/TN3270 (the z/OS images, defined by static VIPAs), and HTTP (the Linux on zSeries servers).>

```
 serverfarm FTP-TN3270
  nat server
  no nat client
  real 9.42.88.1
   inservice
  real 9.42.88.9
   inservice
  real 9.42.88.13
   inservice
  probe ICMP
!
 serverfarm ZLINUX
  nat server
  no nat client
  real 9.42.89.92
   inservice
  real 9.42.89.93
   inservice
  probe ICMP
!
```

<The vserver definitions define individual services – FTP, Web, and TN3270.>

```
 vserver FTP
  virtual 9.42.89.214 tcp ftp service ftp
  no unidirectional
  serverfarm FTP-TN3270-1
  replicate csrp connection
  persistent rebalance
  inservice
!
 vserver HTTP
  virtual 9.42.89.213 tcp www
  no unidirectional
  serverfarm ZLINUX
  replicate csrp connection
  persistent rebalance
  inservice
```

```
!
 vserver TN3270
  virtual 9.42.89.216 tcp 1923
  no unidirectional
  serverfarm FTP-TN3270
  replicate csrp connection
  persistent rebalance
  inservice
!
 ft group 1 vlan 50
  priority 10
!
interface Port-channel1
 description Trunk Connection to 6509B - Gig 1/1 and Gig 3/13
 no ip address
 no logging event link-status
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport nonegotiate
!
interface GigabitEthernet1/1
 description First Fiber for Port Channel
 no ip address
 no logging event link-status
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport nonegotiate
 no cdp enable
 channel-group 1 mode active
 channel-protocol lacp
!
interface GigabitEthernet1/2
 description VLAN14 - Connection to Backbone Network
 no ip address
 no logging event link-status
 switchport
 switchport access vlan 14
 switchport mode access
 no cdp enable
!
interface GigabitEthernet3/1
 description VLAN15 - GIGE2E60 to NIVT Sysplex
 no ip address
 no logging event link-status
 switchport
 switchport access vlan 15
 switchport mode access
 no cdp enable
!
interface GigabitEthernet3/3
 description VLAN41 - Connection to zLinux Host LINUX013
 no ip address
 no logging event link-status
 switchport
 switchport access vlan 41
 switchport mode access
 no cdp enable
!
```

```
interface GigabitEthernet3/13
 description Second Fiber for Port Channel
 no ip address
 no logging event link-status
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport nonegotiate
 channel-group 1 mode active
 channel-protocol lacp
!
interface GigabitEthernet3/15
 description VLAN14 - Connection to AWM Clients
 no ip address
 no logging event link-status
 switchport
 switchport access vlan 14
 switchport mode access
 no cdp enable
!
interface Vlan14
 description VLAN14 - Connection to Site and AWM Clients
 ip address 9.42.89.250 255.255.255.240
 no logging event link-status
!
interface Vlan15
 description VLAN15 - Gigabit Ethernet 2E60 to Sysplex - Real
 ip address 9.42.89.129 255.255.255.248
 no ip redirects
 ip pim dense-mode
 ip igmp join-group 224.0.1.2
 no ip mroute-cache
 ip ospf cost 1
 ip policy route-map Tn3270
 no logging event link-status
!
interface Vlan25
 description VLAN25 - Gigabit Ethernet 2E70 to Sysplex - Virtual
 ip address 9.42.89.137 255.255.255.248
 no ip redirects
 ip pim dense-mode
 ip igmp join-group 224.0.1.2
 no ip mroute-cache
 ip ospf cost 1
 ip policy route-map Tn3270-FTP-to-zOS
 no logging event link-status
!
interface Vlan40
 description VLAN40 - Client side VLAN
 ip address 9.42.89.221 255.255.255.240
 no ip redirects
 ip ospf cost 5
 no logging event link-status
 standby 40 ip 9.42.89.220
 standby 40 priority 200
 standby 40 preempt
!
interface Vlan41
 description VLAN41 - Server Side VLAN
 ip address 9.42.89.90 255.255.255.240
 ip ospf cost 5
 ip policy route-map HTTP
 no logging event link-status
!
router ospf 1
 router-id 9.42.89.250
 log-adjacency-changes
 area 1.1.1.1 stub no-summary
 redistribute connected subnets
 redistribute static subnets
 network 1.1.1.1 0.0.0.0 area 0.0.0.0
 network 9.42.89.90 0.0.0.0 area 1.1.1.1
 network 9.42.89.129 0.0.0.0 area 1.1.1.1
 network 9.42.89.137 0.0.0.0 area 1.1.1.1
 network 9.42.89.221 0.0.0.0 area 1.1.1.1
 network 9.42.89.250 0.0.0.0 area 0.0.0.0
 maximum-paths 5
!
```

<The access-list definitions check data coming in or being returned. The number in the access list correlates with the "match-ip address" statement in one of the following route-map statements.>

```
access-list 100 permit tcp host 9.42.88.1 eq 1923 any
access-list 100 permit tcp host 9.42.88.9 eq 1923 any
access-list 100 permit tcp host 9.42.88.13 eq 1923 any
access-list 100 permit tcp host 9.42.88.1 eq ftp any
access-list 100 permit tcp host 9.42.88.9 eq ftp any
access-list 100 permit tcp host 9.42.88.13 eq ftp any
access-list 100 permit tcp host 9.42.88.1 eq ftp-data any
access-list 100 permit tcp host 9.42.88.9 eq ftp-data any
access-list 100 permit tcp host 9.42.88.13 eq ftp-data any
access-list 101 permit tcp host 9.42.89.92 eq www any
access-list 101 permit tcp host 9.42.89.93 eq www any
!
route-map HTTP permit 101
 match ip address 101
 set ip next-hop 9.42.89.215
!
route-map Tn3270-FTP-to-zOS permit 100
 match ip address 100
 set ip next-hop 9.42.89.215
```

8.2.1.2 CSM NAT

```
hostname NEP6509A
!
module ContentSwitchingModule 5
!
 vlan 40 server
  description VLAN40 - Client Side VLAN
  ip address 9.42.89.211 255.255.255.240
  gateway 9.42.89.220
!
 natpool ZOS 9.42.89.215 9.42.89.215 netmask 255.255.255.240
 natpool ZLINUX 9.42.89.217 9.42.89.217 netmask 255.255.255.240
!
 probe ICMP icmp
  interval 5
  failed 60
  receive 2
!
 serverfarm FTP-TN3270
  nat server
  nat client ZOS
  real 9.42.88.1
   inservice
  real 9.42.88.9
   inservice
  real 9.42.88.13
   inservice
  probe ICMP
!
 serverfarm ZLINUX
  nat server
  nat client ZLINUX
  real 9.42.89.92
   inservice
  real 9.42.89.93
   inservice
  probe ICMP
!
 vserver FTP
  virtual 9.42.89.214 tcp ftp
  no unidirectional
  serverfarm FTP-TN3270
  replicate csrp connection
  persistent rebalance
  inservice
!
 vserver HTTP
  virtual 9.42.89.213 tcp www
  no unidirectional
  serverfarm ZLINUX
  replicate csrp connection
  persistent rebalance
  inservice
!
 vserver TN3270
  virtual 9.42.89.216 tcp 1923
  no unidirectional
  serverfarm FTP-TN3270
  replicate csrp connection
  persistent rebalance
  inservice
!
 ft group 1 vlan 50
  priority 10
!
interface Port-channel1
 description Trunk Connection to 6509B - Gig 1/1 and Gig 3/13
 no ip address
 no logging event link-status
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport nonegotiate
!
interface GigabitEthernet1/1
 description First Fiber for Port Channel
 no ip address
 no logging event link-status
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport nonegotiate
 no cdp enable
 channel-group 1 mode active
 channel-protocol lacp
!
interface GigabitEthernet1/2
 description VLAN14 - Connection to Backbone Network
 no ip address
 no logging event link-status
 switchport
 switchport access vlan 14
 switchport mode access
 no cdp enable
!
interface GigabitEthernet3/1
 description VLAN15 - GIGE2E60 to NIVT Sysplex
 no ip address
 no logging event link-status
 switchport
 switchport access vlan 15
 switchport mode access
 no cdp enable
!
interface GigabitEthernet3/3
 description VLAN41 - Connection to zLinux Host LINUX013
 no ip address
 no logging event link-status
 switchport
 switchport access vlan 41
 switchport mode access
 no cdp enable
!
interface GigabitEthernet3/13
 description Second Fiber for Port Channel
 no ip address
 no logging event link-status
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport nonegotiate
 channel-group 1 mode active
 channel-protocol lacp
!
interface GigabitEthernet3/15
 description VLAN14 - Connection to AWM Clients
 no ip address
 no logging event link-status
 switchport
 switchport access vlan 14
 switchport mode access
 no cdp enable
!
interface Vlan14
 description VLAN14 - Connection to Site and AWM Clients
 ip address 9.42.89.250 255.255.255.240
 no logging event link-status
!
interface Vlan15
 description VLAN15 - Gigabit Ethernet 2E60 to Sysplex - Real
 ip address 9.42.89.129 255.255.255.248
 no ip redirects
 ip pim dense-mode
 ip igmp join-group 224.0.1.2
 no ip mroute-cache
 ip ospf cost 1
 no logging event link-status
!
interface Vlan25
 description VLAN25 - Gigabit Ethernet 2E70 to Sysplex - Virtual
 ip address 9.42.89.137 255.255.255.248
 no ip redirects
 ip pim dense-mode
 ip igmp join-group 224.0.1.2
 no ip mroute-cache
 ip ospf cost 1
 no logging event link-status
!
interface Vlan40
 description VLAN40 - Client Side VLAN
 ip address 9.42.89.221 255.255.255.240
 no ip redirects
 ip ospf cost 5
 no logging event link-status
 standby 40 ip 9.42.89.220
 standby 40 priority 200
 standby 40 preempt
!
interface Vlan41
 description VLAN41 - Server Side VLAN
 ip address 9.42.89.90 255.255.255.240
 ip ospf cost 5
 no logging event link-status
!
router ospf 1
 router-id 9.42.89.250
 log-adjacency-changes
 area 1.1.1.1 stub no-summary
 redistribute connected subnets
 redistribute static subnets
 network 1.1.1.1 0.0.0.0 area 0.0.0.0
 network 9.42.89.90 0.0.0.0 area 1.1.1.1
 network 9.42.89.129 0.0.0.0 area 1.1.1.1
 network 9.42.89.137 0.0.0.0 area 1.1.1.1
 network 9.42.89.221 0.0.0.0 area 1.1.1.1
 network 9.42.89.250 0.0.0.0 area 0.0.0.0
 maximum-paths 5
```

8.3 CSS

As with the CSMs, the CSSs were tested with HTTP only (FTP and TN3270 handled by Sysplex Distributor/MNLB) handling the traffic for all three applications. The following configuration is for CSS11503A (CSS11503B is similar), and shows the configuration for all three traffic types. The configuration for FTP and TN3270 would be omitted for configurations where those applications were not present or were distributed via Sysplex Distributor/MNLB…

8.3.1 CSS Configurations

8.3.1.1 CSS PBR

```
!Generated on 11/06/2003 15:31:07
!Active version: sg0710206a

configure


!************************** GLOBAL **************************
  global-portmap base-port 4000 range 30000

  app
  app session 9.42.89.212

  ip route 0.0.0.0 0.0.0.0 9.42.89.220 1
  ip route 9.42.88.1 255.255.255.255 9.42.89.85 1
  ip route 9.42.88.9 255.255.255.255 9.42.89.85 1
  ip route 9.42.88.13 255.255.255.255 9.42.89.85 1

!************************* INTERFACE ************************
interface  1/1
  bridge vlan 40
  description "Client Side from NEP6509A 3/5"

interface  1/2
  bridge vlan 41
  description "Server Side from NEP6509A 3/7"

interface  2/1
  isc-port-one
  description "Inter Switch Communications 1"

interface  2/2
  isc-port-two
  description "Inter Switch Communications 2"

!************************** CIRCUIT *************************
circuit VLAN40

  ip address 9.42.89.210 255.255.255.240
    ip virtual-router 1
    ip redundant-interface 1 9.42.89.211
    ip redundant-vip 1 9.42.89.213
    ip redundant-vip 1 9.42.89.214
    ip redundant-vip 1 9.42.89.216
    ip critical-service 1 phy-check
    ip critical-service 1 pinglist

circuit VLAN41

  ip address 9.42.89.83 255.255.255.240
    ip virtual-router 2
    ip redundant-interface 2 9.42.89.91
    ip critical-service 2 pinglist
```

```
!************************* SERVICE *************************

<The following service definitions would be omitted if CSS is not
managing FTP and TN3270…>

service mvs001-ftp
  ip address 9.42.88.1
  port 21
  redundant-index 1
  active

service mvs001-tn3270
  ip address 9.42.88.1
  keepalive type tcp
  port 1923
  redundant-index 2
  active

service mvs062-ftp
  ip address 9.42.88.9
  port 21
  redundant-index 3
  active

service mvs062-tn3270
  ip address 9.42.88.9
  keepalive type tcp
  port 1923
  redundant-index 4
  active

service mvs154-ftp
  ip address 9.42.88.13
  port 21
  redundant-index 5
  active

service mvs154-tn3270
  ip address 9.42.88.13
  keepalive type tcp
  port 1923
  redundant-index 6
  active

<End of TN3270 and FTP definition>

service phy-check
  ip address 1.1.1.1
  keepalive frequency 2
  keepalive maxfailure 1
  keepalive retryperiod 2
  keepalive type script ap-kal-phy-check "1/1 1/2" use-output
  active

service pinglist
  keepalive type script ap-kal-pinglist "9.42.89.220 9.42.89.85" use-
output
  keepalive frequency 2
  keepalive maxfailure 2
  keepalive retryperiod 2
  ip address 1.1.1.1
  active

service zLinux013
  ip address 9.42.89.92
  port 80
  keepalive type tcp
  redundant-index 7
  active

service zLinux014
  ip address 9.42.89.93
  port 80
  keepalive type tcp
  redundant-index 8
  active

!*************************** OWNER ***************************
owner ibm

  content http
    vip address 9.42.89.213
    protocol tcp
    port 80
    add service zLinux013
    add service zLinux014
    redundant-index 10
    active
```

<The following definitions would be omitted if the FTP and TN3270 applications were not being managed by CSS>

```
  content FTP
    protocol tcp
    vip address 9.42.89.214
    port 21
    add service mvs001-ftp
    add service mvs062-ftp
    add service mvs154-ftp
    redundant-index 9
    application ftp-control
    active

  content tn3270
    protocol tcp
    vip address 9.42.89.216
    port 1923
    add service mvs001-tn3270
    add service mvs062-tn3270
    add service mvs154-tn3270
    redundant-index 11
    active
```

8.3.1.2 CSS NAT

```
!Generated on 11/21/2003 09:52:59
!Active version: sg0710206a

configure
```

```
!*************************** GLOBAL ***************************
  global-portmap base-port 4000 range 30000

  app
  app session 9.42.89.212

  ip route 0.0.0.0 0.0.0.0 9.42.89.220 1
  ip route 9.42.88.1 255.255.255.255 9.42.89.85 1
  ip route 9.42.88.9 255.255.255.255 9.42.89.85 1
  ip route 9.42.88.13 255.255.255.255 9.42.89.85 1

  ftp-record FTP1 9.42.88.187 root des-password m0703j

!************************* INTERFACE *************************
interface  1/1
  bridge vlan 40
  description "Client Side from NEP6509A 3/5"

interface  1/2
  bridge vlan 41
  description "Server Side from NEP6509A 3/7"

interface  2/1
  isc-port-one
  description "Inter Switch Communications 1"

interface  2/2
  isc-port-two
  description "Inter Switch Communications 2"

!************************** CIRCUIT **************************
circuit VLAN40

  ip address 9.42.89.210 255.255.255.240
    ip virtual-router 1
    ip redundant-interface 1 9.42.89.211
    ip redundant-vip 1 9.42.89.213
    ip redundant-vip 1 9.42.89.214
    ip redundant-vip 1 9.42.89.216
    ip critical-service 1 pinglist
    ip critical-service 1 phy-check

circuit VLAN41

  ip address 9.42.89.83 255.255.255.240
    ip virtual-router 2
    ip redundant-interface 2 9.42.89.91
    ip redundant-vip 2 9.42.89.81
    ip redundant-vip 2 9.42.89.82
    ip critical-service 2 pinglist

!************************** SERVICE **************************
service mvs001-ftp
  ip address 9.42.88.1
  port 21
  redundant-index 1
  keepalive type tcp
  keepalive port 21
  active
```

```
service mvs001-tn3270
  ip address 9.42.88.1
  keepalive type tcp
  port 1923
  redundant-index 2
  keepalive port 1923
  active

service mvs062-ftp
  ip address 9.42.88.9
  port 21
  redundant-index 3
  keepalive type tcp
  keepalive port 21
  active

service mvs062-tn3270
  ip address 9.42.88.9
  keepalive type tcp
  port 1923
  redundant-index 4
  keepalive port 1923
  active

service mvs154-ftp
  ip address 9.42.88.13
  port 21
  redundant-index 5
  keepalive type tcp
  keepalive port 21
  active

service mvs154-tn3270
  ip address 9.42.88.13
  keepalive type tcp
  port 1923
  redundant-index 6
  keepalive port 1923
  active

service phy-check
  ip address 1.1.1.1
  keepalive frequency 2
  keepalive maxfailure 1
  keepalive retryperiod 2
  keepalive type script ap-kal-phy-check "1/1 1/2" use-output
  active

service pinglist
  keepalive type script ap-kal-pinglist "9.42.89.220 9.42.89.85" use-
output
  keepalive frequency 2
  keepalive maxfailure 2
  keepalive retryperiod 2
  ip address 1.1.1.1
  active

service zLinux013
  ip address 9.42.89.92
  port 80
  redundant-index 7
  keepalive type http
  keepalive uri "/test/file1k.html"
  active

service zLinux014
  ip address 9.42.89.93
  port 80
  redundant-index 8
  keepalive type http
  keepalive uri "/test/file1k.html"
  active

!*************************** OWNER ***************************
owner ibm

  content FTP
    vip address 9.42.89.214
    protocol tcp
    port 21
    redundant-index 9
    application ftp-control
    add service mvs062-ftp
    add service mvs154-ftp
    active

  content http
    vip address 9.42.89.213
    protocol tcp
    port 80
    add service zLinux013
    add service zLinux014
    redundant-index 10
    active

  content tn3270
    vip address 9.42.89.216
    port 1923
    protocol tcp
    add service mvs001-tn3270
    add service mvs062-tn3270
    add service mvs154-tn3270
    redundant-index 11
    active

!*************************** GROUP ***************************
group zLinux
  redundant-index 12
  add destination service zLinux013
  add destination service zLinux014
  vip address 9.42.89.82
  active

group zOS
  redundant-index 13
  add destination service mvs001-ftp
  add destination service mvs062-ftp
  add destination service mvs154-ftp
  add destination service mvs001-tn3270
  add destination service mvs062-tn3270
  add destination service mvs154-tn3270
  vip address 9.42.89.81
  active
```

8.3.1.3 Cisco Catalyst 6509 Definitions with CSS

The following configuration for NEP6509A was used with the above CSS configuration.

```
hostname NEP6509A
!
interface Port-channel1
 no ip address
 no logging event link-status
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport nonegotiate
!
interface GigabitEthernet1/1
 description Trunk Connection to NEP6509B
 no ip address
 no logging event link-status
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport nonegotiate
 channel-group 1 mode active
 channel-protocol lacp
!
interface GigabitEthernet1/2
 description Connection to NIVT6509 - Site
 no ip address
 no logging event link-status
 switchport
 switchport access vlan 14
 switchport mode access
 no cdp enable
!
interface GigabitEthernet3/1
 description Connection to NIVT Sysplex
 no ip address
 no logging event link-status
 switchport
 switchport access vlan 15
 switchport mode access
 no cdp enable
!
interface GigabitEthernet3/3
 description Connection to MVS014 - zLinux
 no ip address
 no logging event link-status
 switchport
 switchport access vlan 41
 switchport mode access
 no cdp enable
!
interface GigabitEthernet3/5
 description Inbound Connection from NEP6509A
 no ip address
 no logging event link-status
 switchport
 switchport access vlan 40
 switchport mode access
 no cdp enable
!
interface GigabitEthernet3/7
 description Outbound Connection from NEP6509A
 no ip address
 no logging event link-status
 switchport
 switchport access vlan 41
 switchport mode access
 no cdp enable
!
interface GigabitEthernet3/13
 description Trunk Connection to NEP6509B
 no ip address
 no logging event link-status
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport nonegotiate
 channel-group 1 mode active
 channel-protocol lacp
!
interface GigabitEthernet3/15
 description Connection to Z52 (site) and AWM Clients
 no ip address
 no logging event link-status
 switchport
 switchport access vlan 14
 switchport mode access
 no cdp enable
!
interface Vlan14
 description Connection to NIVT6500 and Site
 ip address 9.42.89.250 255.255.255.240
 no logging event link-status
!
interface Vlan15
 description Connection to GIGE 2E70 on RALVM2
 ip address 9.42.89.129 255.255.255.248
 no ip redirects
 ip pim dense-mode
 ip igmp join-group 224.0.1.2
 no ip mroute-cache
 ip ospf cost 1
 ip policy route-map Tn3270-FTP-to-zOS
 no logging event link-status
!
interface Vlan25
 ip address 9.42.89.137 255.255.255.248
 no ip redirects
 ip pim dense-mode
 ip igmp join-group 224.0.1.2
 no ip mroute-cache
 ip ospf cost 5
 ip policy route-map Tn3270-FTP-to-zOS
 no logging event link-status
```

```
!
interface Vlan40
 description Client Side VLAN
 ip address 9.42.89.221 255.255.255.240
 no ip redirects
 ip ospf cost 5
 no logging event link-status
 standby 40 ip 9.42.89.220
 standby 40 priority 200
 standby 40 preempt
!
interface Vlan41
 description Server Side VLAN
 ip address 9.42.89.90 255.255.255.240
 no ip redirects
 ip ospf cost 5
 ip policy route-map HTTP-to-Linux
 no logging event link-status
 standby 41 ip 9.42.89.85
 standby 41 priority 200
 standby 41 preempt
!
router ospf 1
 router-id 9.42.89.250
 log-adjacency-changes
 area 1.1.1.1 stub no-summary
 redistribute connected subnets
 redistribute static subnets
 network 1.1.1.1 0.0.0.0 area 0.0.0.0
 network 9.42.89.90 0.0.0.0 area 1.1.1.1
 network 9.42.89.129 0.0.0.0 area 1.1.1.1
 network 9.42.89.137 0.0.0.0 area 1.1.1.1
 network 9.42.89.221 0.0.0.0 area 1.1.1.1
 network 9.42.89.250 0.0.0.0 area 0.0.0.0
 maximum-paths 5
!
access-list 100 permit tcp host 9.42.88.1 eq 1923 any
access-list 100 permit tcp host 9.42.88.9 eq 1923 any
access-list 100 permit tcp host 9.42.88.13 eq 1923 any
access-list 100 permit tcp host 9.42.88.1 eq ftp any
access-list 100 permit tcp host 9.42.88.9 eq ftp any
access-list 100 permit tcp host 9.42.88.13 eq ftp any
access-list 100 permit tcp host 9.42.88.1 eq ftp-data any
access-list 100 permit tcp host 9.42.88.9 eq ftp-data any
access-list 100 permit tcp host 9.42.88.13 eq ftp-data any
access-list 101 permit tcp host 9.42.89.92 eq www any
access-list 101 permit tcp host 9.42.89.93 eq www any
!
route-map Tn3270-FTP-to-zOS permit 100
 match ip address 100
 set ip next-hop 9.42.89.91
!
route-map HTTP-to-Linux permit 101
 match ip address 101
 set ip next-hop 9.42.89.91
```

8.4 CSM Configuration using SASP and the z/OS Load Balancing Advisor

The CSM tests using SASP were very similar to the CSM tests described earlier, with two key differences:

1. The HTTP Servers were located on the z/OS systems, the Linux systems were not used.
2. The CSMs were configured to obtain workload balancing recommendations from the z/OS Load Balancing Advisor using SASP.

8.4.1 z/OS Configuration

The z/OS Load Balancing Advisor was deployed on MVS001 and configured so that it could be restarted on system MVS154 and MVS062 in the case of an MVS001 system failure. A instance of the z/OS Load Balancing Agent was deployed on each system (MVS001, MVS154 and MVS062).

8.4.1.1 z/OS Load Balancing Advisor Configuration

The following is the configuration file for the Load Balancing Advisor. Note, that the same configuration file can be used regardless of which system the advisor is deployed on.

```
#
#
#       LBADVCNF  (Load Balancing Advisor configuration)
#
#  This file contains sample configuration statements for the Load
#  Balancing Advisor.
#


debug_level              7
update_interval          60

# The port that the Advisor will listen to for connections from the
# Agents

agent_connection_port    8100

#-----------------------------
#
# The agent_id_list statement specifies a list of Agent IP addresses
# and ports that are allowed to connect to the Advisor
#

agent_id_list
{
   9.42.88.1..8000
   9.42.88.9..8000
   9.42.88.13..8000
}

 #-----------------------------
 #
 # The IP address and port that the Advisor will listen to for
 # connections from the CSMs
 #

lb_connection_v4         9.42.88.217..3860

#
# List of valid CSMs that can connect to the Load Balancing Advisor
#
```

```
lb_id_list
{
  9.42.89.211
  9.42.89.212
}
```

Note that the Load Balancing Advisor is configured to use a unique Application-instance DVIPA (`9.42.88.217)` on its listening sockets. This allows the CSM and the Load Balancing Agents to be able to reconnect to the Load Balancing Advisor after a failure, even if the Advisor is restarted on another system in the sysplex.

8.4.1.2 z/OS Load Balancing Agent Configuration

The following are the configuration files used for each of the Load Balancing agents.

```
#  This file contains sample configuration statements for the Load
#  Balancing Agent.
#


debug_level      7

#------------------------------
#
# The advisor_id statement specifies the IP address and port of the
# Advisor to which the Agent will connect
#

advisor_id           9.42.88.217..8100

#------------------------------
#
# The host_connection statement specifies the local IP address and port
# the Agent will use in connecting to the
#

host_connection          9.42.88.1..8000
```

Note that only difference in the agent configuration files for each system is the *host_connection* statement that specifies the source IP address and port that each Agent will use in connecting to the Advisor. A static VIPA was used for the source IP address to ensure simplify the configuration of the Advisor (i.e. only a single IP address needs to be specified in the Advisor configuration file for each Agent).

8.4.1.3 Additional z/OS configuration information

While setting up the Load Balancing Advisor and Agents is not difficult, it does require several configuration steps. The list of configuration files and steps in this document is not all inclusive, rather it focuses on the key aspects of the configuration required for the tests described in this document.

For high availability and fault tolerance of the Advisor/Agents, the following steps were taken:

- The Advisor was placed in the AUTOLOG list in the TCPIP profile. This allowed the Advisor to be automatically started on MVS001 when TCP/IP was initialized (or when TCP/IP was restarted). Note while the Advisor was started via the AUTOLOG list, NOAUTOLOG was specified in the Advisor PORT reservation statement (i.e. the Advisor was not monitored for restart by the TCP/IP stack).
- A z/OS ARM policy was defined that allowed restarts of the Advisor on the same system (if the Advisor failed) or on any other system in the sysplex (if the system failed). For more details on

enabling and configuring ARM, refer to *z/OS V1R4.0 MVS Setting Up a Sysplex* (**Document Number:** SA22-7625). The ARM policy included the following the following statements for the Advisor:

```
RESTART_GROUP(LBADV)
        TARGET_SYSTEM(*)
        FREE_CSA(100,500)
        ELEMENT(EZBLBADV)
          RESTART_ATTEMPTS(3,30)
          RESTART_TIMEOUT(15)
          READY_TIMEOUT(30)
          TERMTYPE(ALLTERM)
                 RESTART_METHOD(BOTH,STC,'S LBADV')
```

- z/OS ARM policy was also used to provide for automatic restart of the Agent on the same system. The following are the definitions that were used:

```
RESTART_GROUP(LBAGENT)
        ELEMENT(LBAGENT)
          RESTART_ATTEMPTS(3,30)
          RESTART_TIMEOUT(15)
          READY_TIMEOUT(30)
          TERMTYPE(ELEMTERM)
                 RESTART_METHOD(BOTH,STC,'S LBAGENT')
```

- The following is the definition of the Dynamic VIPA that was associated with the Advisor. Note that the DVIPA definition was placed in the TCP/IP profiles (inside the VIPADYNAMIC block) of all systems (for recovery purposes):

```
;
;------------------------------------------------------------
; VIPA RANGE FOR SASP LBADV
;------------------------------------------------------------
;
  VIPARANGE DEFINE MOVE NONDISRUPTIVE 255.255.255.248 9.42.88.217
;
;------------------------------------------------------------
```

8.4.1.4 CSM configuration for SASP

The configuration of the Cisco CSM for this test was very similar to the CSM configuration described in section 8.2.1.2, CSM NAT. Only the portion of the configuration that required changes for enabling the tests with SASP are included here. Key statement related to the SASP support are italicized.

```
module ContentSwitchingModule 5
variable ROUTE_UNKNOWN_FLOW_PKTS 1
variable SASP_CSM_UNIQUE_ID Cisco-CSM-6509A
!
 ft group 1 vlan 50
   priority 10
!
 vlan 40 server
   ip address 9.42.89.211 255.255.255.240
    gateway 9.42.89.220
!
 natpool ZOS 9.42.89.215 9.42.89.215 netmask 255.255.255.240
!
 probe ICMP icmp
   interval 5
   failed 60
     receive 2
   !
 serverfarm FTP
   nat server
   nat client ZOS
   bindid 65520
   real 9.42.88.9
     inservice
   real 9.42.88.1
    inservice
   real 9.42.88.13
    inservice
   !
 serverfarm HTTP
   nat server
   nat client ZOS
   bindid 65520
   real 9.42.88.1
     inservice
   real 9.42.88.9
     inservice
   real 9.42.88.13
     inservice
    probe ICMP
!
 serverfarm TN3270
   nat server
   nat client ZOS
   bindid 65520
   real 9.42.88.9
     inservice
   real 9.42.88.13
     inservice
   real 9.42.88.1
     inservice
!
 vserver FTP
   virtual 9.42.89.214 tcp ftp
   no unidirectional
   serverfarm FTP
   replicate csrp connection
   persistent rebalance
   inservice
!
 vserver HTTP
  virtual 9.42.89.213 tcp www
  no unidirectional
  serverfarm HTTP
  replicate csrp connection
  persistent rebalance
  inservice
!
 vserver TN3270
  virtual 9.42.89.216 tcp 1923
  no unidirectional
  serverfarm TN3270
  replicate csrp connection
  persistent rebalance
  inservice
!
 dfp
  agent 9.42.88.217 3860 65520
```

Enabling SASP on an existing CSM configuration is a simple operation, the following changes were required:

- Each load balancer connecting to a z/OS Load Balancing Advisor must have a unique ID. By default, the CSM will have the same ID, therefore, if multiple CSMs are deployed using the same z/OS Load Balancing Advisor, a unique ID needs to be configured for each (see ***variable SASP_CSM_UNIQUE_ID)***
- The ***BINDID*** associates each Serverfarm with the configured DFP Agent. Each *vserver* must utilize separate *serverfarms* in order to register application-specific members, otherwise the CSM will only register system members. This why the original serverfarm FTP-TN3270 was split into two serverfarms: FTP and TN3270.
- The ***dfp agent*** is configured with the IP Address and Listening Port of the z/OS Load Balancing Advisor along with the ***BINDID***.

9 Appendix B – Explanation of Workload Distribution Flows

9.1 Sysplex Distributor/MNLB

This section describes the IP subnet allocation and the packet flows in the Sysplex Distributor/MNLB test scenario.

9.1.1 Network Addresses for the Sysplex Distributor/MNLB Test Case

The following table lists all the IP subnets that are used in the Sysplex Distributor/MNLB test case.

Network address	Lowest address	Highest address	Network	Comments
9.42.89.96/29	9.42.89.97	9.42.89.102	HiperSockets	HiperSockets between z/OS operating system images
9.42.89.128/29	9.42.89.129	9.42.89.134	VLAN15	OSA-E to NEP6509A
9.42.89.136/29	9.42.89.137	9.42.89.142	VLAN25	OSA-E to NEP6509B
9.42.89.240/28	9.42.89.241	9.42.89.254	VLAN14	Client network
9.42.88.0/30	9.42.88.1	9.42.88.2	MVS001 Static VIPA	For EE and GRE tunnel endpoint
9.42.88.8/30	9.42.88.9	9.42.88.10	MVS062 Static VIPA	For EE and GRE tunnel endpoint
9.42.88.12/30	9.42.88.13	9.42.88.14	MVS154 Static VIPA	For EE and GRE tunnel endpoint
9.42.88.160/29	9.42.88.161	9.42.88.166	Dynamic XCF	Dynamic XCF network between z/OS operating system images
9.42.88.168/29	9.42.88.169	9.42.88.174	Distributed Dynamic VIPAs	9.42.88.169: TN3270 9.42.88.170: Web 9.42.88.171: FTP 9.42.88.172: LDAP

9.1.2 Sysplex Distributor/MNLB Test Scenario – Flow Explanations

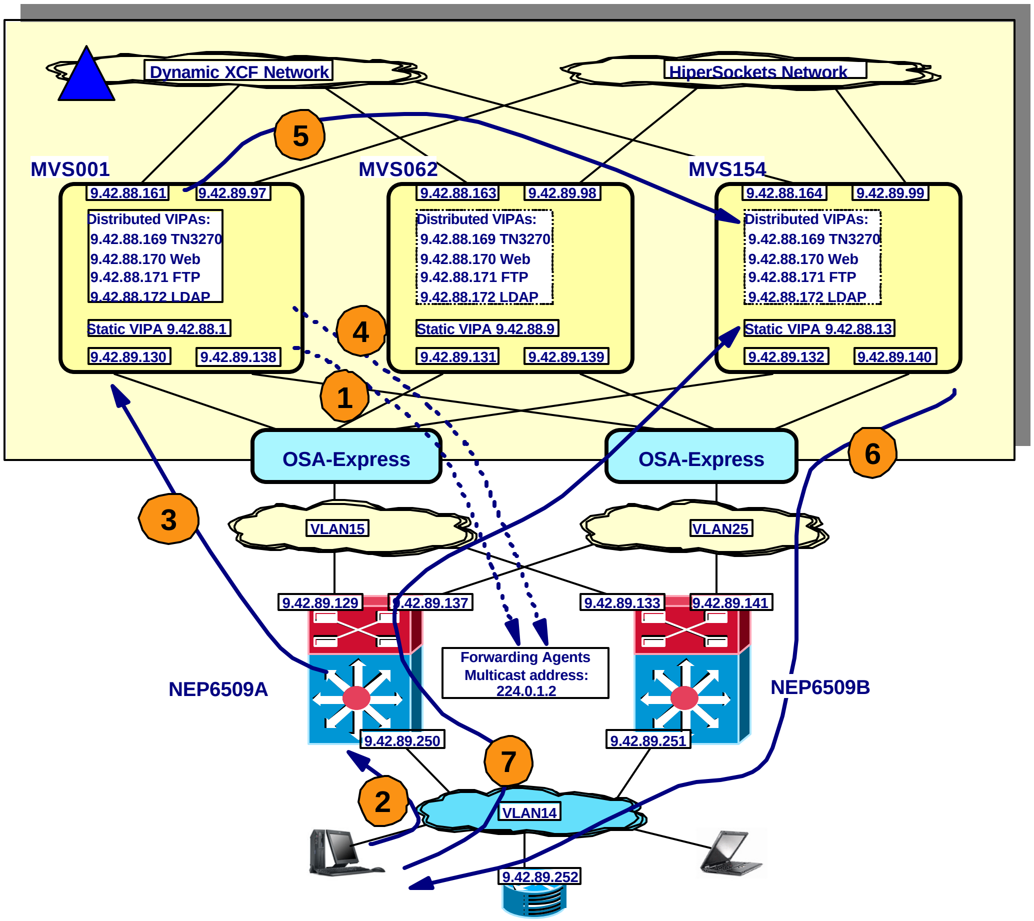


Figure 14 Sysplex Distributor/MNLB packet flow details

Flow explanations:

1. When the stack that has the Sysplex Distributing responsibility initializes, it sends a multicast to the Cisco IOS Software forwarding agents via the configured multicast address. In this scenario, the address is 224.0.1.2. The distributing stack informs the forwarding agents of which distributed VIPA addresses and ports (a so-called wildcard affinity) it manages TCP connection balancing for and which IP address to use as next-hop IP address for packets being routed to those distributed VIPA addresses. This address will be the Dynamic XCF address of the distributing stack. In this scenario, the address is 9.42.88.161.

2. Assume a client on client IP address 9.42.89.241 wants to establish a connection with the TN3270 server cluster. The client will send an IP packet with [DestIP=9.42.88.169, DestPort=23, SrcIP=9.42.89.241, SrcPort=ep1]. The forwarding agent that resides on the path between the client and the z/OS Sysplex looks into its connection affinity table to decide if it has this TCP connection (4-tuple) registered. Assuming this is a new connection request (a TCP SYN segment), the forwarding agent doesn’t know about the connection yet. It therefore takes the IP packet and forwards it to the z/OS TCP/IP stack that had registered the distributed VIPA address and port for the Sysplex Distributor to make a decision about which target stack is the best choice for a new connection.

3. The packet is then forwarded to the next-hop IP address of the distributing stack (its dynamic XCF IP address): 9.42.88.161. This address is matched by a route definition on the Cisco Catalyst 6509 that routes the traffic via a GRE tunnel definition (tunnel1) towards the static VIPA address of MVS001: 9.42.88.1. The packet traveling from the Cisco Catalyst 6509 will be a GRE encapsulated packet: [GRE_DestIP=9.42.88.1, GRE_SrcIP=9.42.89.129, RealDestIP=9.42.88.169, RealPort=23, RealSrcIP=9.42.89.241, RealSrcPort=ep1].

4. The connection request will now be processed by Sysplex Distributor logic to determine which of the target stacks to send this new connection request to. When that decision has been made, the distributing stack sends a multicast to the forwarding agents on their multicast IP address: 224.0.1.2 about the established affinity for this new connection. Let's assume the connection gets distributed to MVS154. The forwarding agents add an entry to their affinity tables for [DestIP=9.42.88.169, DestPort=23, SrcIP=9.42.89.241, SrcPort=ep1 – affinity with Dynamic XCF IP address=9.42.88.164].

5. The distributing stack will then forward the TCP SYN segment over dynamic XCF to MVS154 for further connection setup processing. Please note that this is the only packet in a distributed TCP connection that is forwarded via Dynamic XCF. From here on, all packets from the client will be sent directly to the chosen target stack without involving the distributing stack or the Dynamic XCF network itself.

6. MVS154 will send a TCP SYN+ACK segment back to the client. This IP packet will be sent using the most optimal route that exists between MVS154 and the client. [DestIP=9.42.89.241, DestPort=ep1, SrcIP=9.42.88.169, SrcPort=23]

7. The client will then continue into the normal TCP connection setup sequence and send an ACK segment (the last of the three-way TCP handshake exchanges). This packet will be intercepted by the forwarding agent in the Cisco Catalyst 6509 matching it to the affinity entry that was added previously. This affinity entry will dictate that to send this packet to the chosen target stack it should be forwarded to the dynamic XCF IP address of MVS154 – the 9.42.88.164 address. This address is matched by a route definition on the Cisco Catalyst 6509 that routes the traffic via a GRE tunnel definition (tunnel154) towards the static VIPA address of MVS154: 9.42.88.13. The packet traveling from the Cisco Catalyst 6509 will again be a GRE encapsulated packet: [GRE_DestIP=9.42.88.13, GRE_SrcIP=9.42.89.129, RealDestIP=9.42.88.169, RealPort=23, RealSrcIP=9.42.89.241, RealSrcPort= ep1].

From here on the client and the target stack continue to communicate with inbound packets being GRE encapsulated from the Cisco Catalyst 6509 towards the target stack, and outbound packets flowing directly back towards the client location.

9.2 CSS

This section describes the IP subnet allocation and the packets flows in the CSS test scenario.

9.2.1 Network addresses for the CSS test case

The following table lists all the IP subnets that are used in the CSS test case.

Network address	Lowest address	Highest address	Network	Comments
9.42.89.80/28	9.42.89.81	9.42.89.94	VLAN41	6509, CSS, OSA-E to Linux
9.42.89.96/29	9.42.89.97	9.42.89.102	HiperSockets	HiperSockets between z/OS and Linux systems
9.42.89.128/29	9.42.89.129	9.42.89.134	VLAN15	OSA-E z/OS to NEP6509A
9.42.89.136/29	9.42.89.137	9.42.89.142	VLAN25	OSA-E z/OS to NEP6509B
9.42.89.208/28	9.42.89.209	9.42.89.222	VLAN40	CSM-CSS interconnect
9.42.89.240/28	9.42.89.241	9.42.89.254	VLAN14	Client network
9.42.88.0/30	9.42.88.1	9.42.88.2	MVS001 Static VIPA	Target for CSS balancing
9.42.88.8/30	9.42.88.9	9.42.88.10	MVS062 Static VIPA	Target for CSS balancing
9.42.88.12/30	9.42.88.13	9.42.88.14	MVS154 Static VIPA	Target for CSS balancing
9.42.88.160/29	9.42.88.161	9.42.88.166	Dynamic XCF	Dynamic XCF network between z/OS operating system images

9.2.2 CSS test case – flow explanations

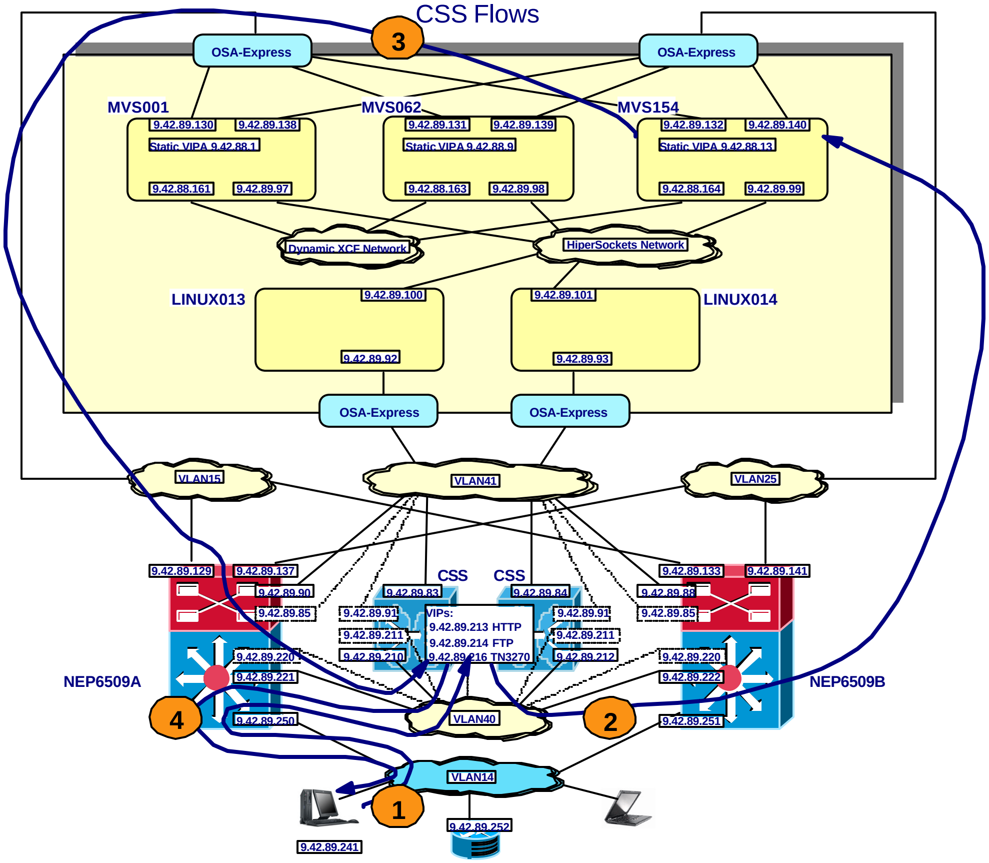


Figure 15 CSS packet flow details

Flow explanations when server NATing in combination with Policy Bases Routing is used:

1. Assume a client on IP address 9.42.89.241 wants to establish a connection with the TN3270 server cluster. The client will send a TCP SYN IP packet with [DestIP=9.42.89.216, DestPort=23, SrcIP=9.42.89.241, SrcPort=ep1]. The Cisco Catalyst 6509 switch will forward that IP packet to the active CSS that currently owns the 9.42.89.216 VIP address.

2. The CSS will select one of the three z/OS systems as the target for this new connection setup, and change the destination IP address in the TCP SYN IP packet to the static VIPA of the chosen z/OS system – in this example assuming MVS154: 9.42.88.13 – [DestIP=9.42.88.13, DestPort=23, SrcIP=9.42.89.241, SrcPort=ep1]. This packet will be routed from the CSS via one of the Cisco Catalyst 6509 switches towards MVS154.

3. MVS154 will process the TCP SYN packet and generate a response TCP SYN+ACK IP packet that will be sent outbound towards the client IP address: [DestIP=9.42.89.241, DestPort=ep1, SrcIP=9.42.88.13, SrcPort=23]. This outbound packet will match Policy Based Routing entries in the Cisco Catalyst 6509 switches and instead of being forwarded directly back to the client IP address, it will be forwarded to the CSS.

4. The CSS will then change the source IP address in the outbound packet to match the IP address the client originally sent the TCP SYN packet to: [DestIP=9.42.89.241, DestPort=ep1, SrcIP=9.42.89.216, SrcPort=23]. This packet will then be routed from the

CSS via the Cisco Catalyst 6509 switch to the client network.

Flow explanations when both server NATing and client NATing is being used in the CSS:

1. Assume a client on IP address 9.42.89.241 wants to establish a connection with the TN3270 server cluster. The client will send a TCP SYN IP packet with [DestIP=9.42.89.216, DestPort=23, SrcIP=9.42.89.241, SrcPort=ep1]. The Cisco Catalyst 6509 will forward that IP packet to the active CSS that currently owns the 9.42.89.216 VIP address.

2. The CSS will select one of the three z/OS systems as the target for this new connection setup, and change the destination IP address in the TCP SYN IP packet to the static VIPA of the chosen z/OS system – in this example assuming MVS154: 9.42.88.13. It will also change the client IP address to the configured client NAT address and choose a free local port number: [DestIP=9.42.88.13, DestPort=23, SrcIP=9.42.89.215, SrcPort=ep2]. This packet will be routed from the CSS via one of the Cisco Catalyst 6509 switches towards MVS154.

3. MVS154 will process the TCP SYN packet and generate a response TCP SYN+ACK IP packet that will be sent outbound towards the NATed client IP address on the CSS: [DestIP=9.42.89.215, DestPort=ep2, SrcIP=9.42.88.13, SrcPort=23]. This outbound packet will be routed to the CSS switch that currently owns the 9.42.89.215 IP address.

4. The CSS will then change the source IP address in the outbound packet to match the IP address the client originally sent the TCP SYN packet to, and it will change the destination IP address and port number to those of the real client: [DestIP=9.42.89.241, DestPort=ep1, SrcIP=9.42.89.216, SrcPort=23]. This packet will then be routed from the CSS via the Cisco Catalyst 6509 switch to the client network.

9.3 CSM

This section describes the IP subnet allocation and the packet flows in the CSM test scenario.

9.3.1 Network addresses for the CSM test case

The following table lists all the IP subnets that are used in the CSM test case.

Network address	Lowest address	Highest address	Network	Comments
9.42.89.80/28	9.42.89.81	9.42.89.94	VLAN41	CSM, OSA-E to Linux
9.42.89.96/29	9.42.89.97	9.42.89.102	HiperSockets	HiperSockets between z/OS and Linux systems
9.42.89.128/29	9.42.89.129	9.42.89.134	VLAN15	OSA-E z/OS to NEP6509A
9.42.89.136/29	9.42.89.137	9.42.89.142	VLAN25	OSA-E z/OS to NEP6509B
9.42.89.208/28	9.42.89.209	9.42.89.222	VLAN40	CSM-interconnect
9.42.89.240/28	9.42.89.241	9.42.89.254	VLAN14	Client network
9.42.88.0/30	9.42.88.1	9.42.88.2	MVS001 Static VIPA	Target for CSM balancing
9.42.88.8/30	9.42.88.9	9.42.88.10	MVS062 Static VIPA	Target for CSM balancing
9.42.88.12/30	9.42.88.13	9.42.88.14	MVS154 Static VIPA	Target for CSM balancing
9.42.88.160/29	9.42.88.161	9.42.88.166	Dynamic XCF	Dynamic XCF network between z/OS operating system images

9.3.2 CSM Test cases – flow explanations

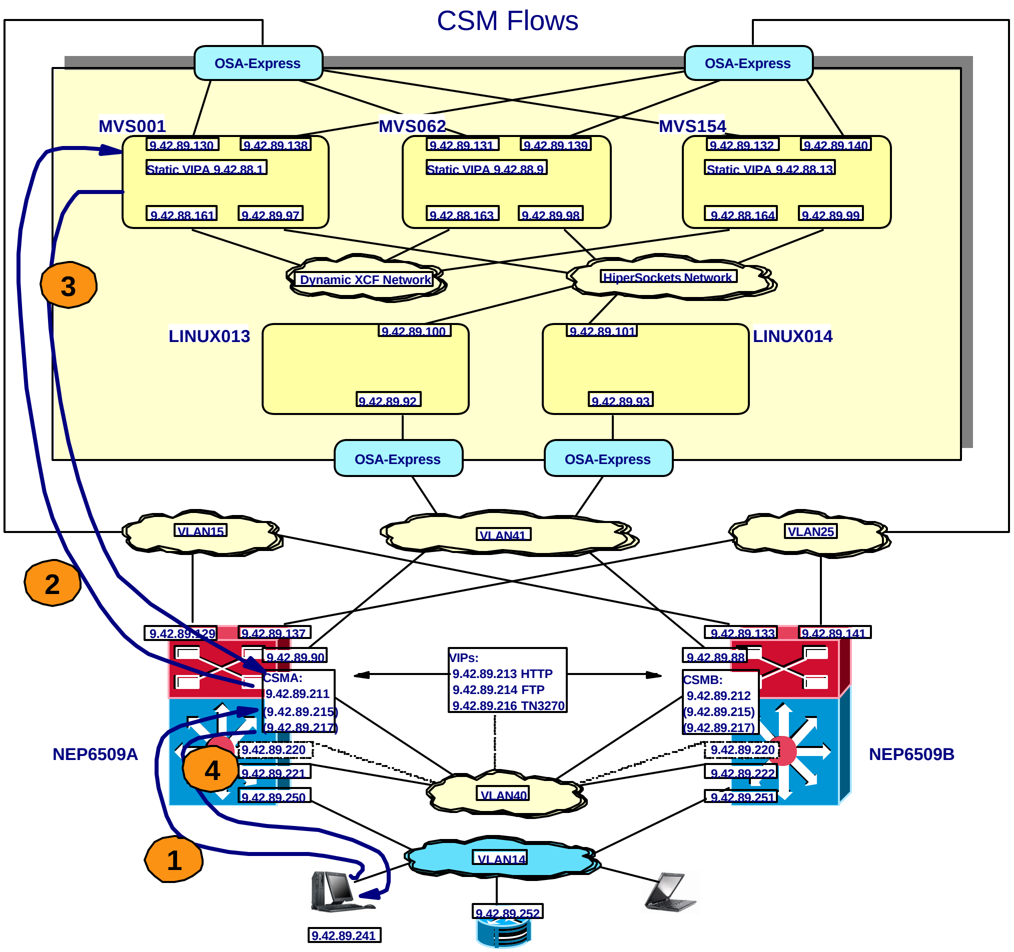


Figure 16 CSM packet flow details

Flow explanations when server NATing in combination with Policy Bases Routing is used:

1. Assume a client on IP address 9.42.89.241 wants to establish a connection with the TN3270 server cluster. The client will send a TCP SYN IP packet with [DestIP=9.42.89.216, DestPort=23, SrcIP=9.42.89.241, SrcPort=ep1]. The Cisco Catalyst 6509 switch will forward that IP packet to the active CSM blade that currently owns the 9.42.89.216 VIP address.

2. The CSM will select one of the three z/OS systems as the target for this new connection setup, and change the destination IP address in the TCP SYN IP packet to the static VIPA of the chosen z/OS system – in this example assuming MVS154: 9.42.88.13 – [DestIP=9.42.88.13, DestPort=23, SrcIP=9.42.89.241, SrcPort=ep1]. This packet will be routed from the CSM via one of the Cisco Catalyst 6509 switches towards MVS154.

3. MVS154 will process the TCP SYN packet and generate a response TCP SYN+ACK IP packet that will be sent outbound towards the client IP address: [DestIP=9.42.89.241, DestPort=ep1, SrcIP=9.42.88.13, SrcPort=23]. This outbound packet will match Policy Based Routing entries in the Cisco Catalyst 6509 switches and instead of being forwarded directly back to the client IP address, it will be forwarded to the CSM blade.

4. The CSM will then change the source IP address in the outbound packet to match the IP address the client originally sent the TCP SYN packet to: [DestIP=9.42.89.241, DestPort=ep1, SrcIP=9.42.89.216, SrcPort=23]. This packet will then be routed from the CSM blade via the Cisco Catalyst 6509 switch to the client network.

Flow explanations when both server NATi ng and client NATing are being used in the CSM:

1. Assume a client on IP address 9.42.89.241 wants to establish a connection with the TN3270 server cluster. The client will send a TCP SYN IP packet with [DestIP=9.42.89.216, DestPort=23, SrcIP=9.42.89.241, SrcPort=ep1]. The Cisco Catalyst 6509 will forward that IP packet to the active CSM that currently owns the 9.42.89.216 VIP address.

2. The CSM will select one of the three z/OS systems as the target for this new connection setup, and change the destination IP address in the TCP SYN IP packet to the static VIPA of the chosen z/OS system – in this example assuming MVS154: 9.42.88.13. It will also change the client IP address to the configured client NAT address and choose a free local port number: [DestIP=9. 42.88.13, DestPort=23, SrcIP=9.42.89.215, SrcPort=ep2]. This packet will be routed from the CSM blade via one of the Cisco Catalyst 6509 switches towards MVS154.

3. MVS154 will process the TCP SYN packet and generate a response TCP SYN+ACK IP packet that will be sent outbound towards the NATed client IP address on the CSM: [DestIP=9.42.89.215, DestPort=ep2, SrcIP=9.42.88.13, SrcPort=23]. This outbound packet will be routed to the CSM blade that currently owns the 9.42.89.215 IP address.

4. The CSM will then change the source IP address in the outbound packet to match the IP address the client originally sent the TCP SYN packet to, and it will change the destination IP address and port number to those of the real client: [DestIP=9.42.89.241, DestPort=ep1, Src IP=9.42.89.216, SrcPort=23]. This packet will then be routed from the CSM blade via the Cisco Catalyst 6509 switch to the client network.

9.4 CSM with SASP and the z/OS Load Balancing Advisor

This section describes the IP subnet allocation and the packet flows in the CSM test scenario that involved SASP and the z/OS Load Balancing Advisor.

9.4.1 Network addresses for the CSM/SASP test case

The following table lists all the IP subnets that are used in the CSM test case with SASP and the z/OS Load Balancing Advisor.

Network address	Lowest address	Highest address	Network	Comments
9.42.89.96/29	9.42.89.97	9.42.89.102	HiperSockets	HiperSockets between z/OS systems
9.42.89.128/29	9.42.89.129	9.42.89.134	VLAN15	OSA-E z/OS to NEP6509A
9.42.89.136/29	9.42.89.137	9.42.89.142	VLAN25	OSA-E z/OS to NEP6509B
9.42.89.208/28	9.42.89.209	9.42.89.222	VLAN40	CSM-interconnect
9.42.89.240/28	9.42.89.241	9.42.89.254	VLAN14	Client network
9.42.88.0/30	9.42.88.1	9.42.88.2	MVS001 Static VIPA	Target for CSM balancing
9.42.88.8/30	9.42.88.9	9.42.88.10	MVS062 Static VIPA	Target for CSM balancing
9.42.88.12/30	9.42.88.13	9.42.88.14	MVS154 Static VIPA	Target for CSM balancing

9.42.88.160/29	9.42.88.161	9.42.88.166	Dynamic XCF	Dynamic XCF network between z/OS operating system images
9.42.88.216/29	9.42.88.217	9.42.88.222	Dynamic VIPA	Dynamic VIPA for z/OS Load Balancing Advisor

9.4.2 CSM with SASP Test cases – flow explanations

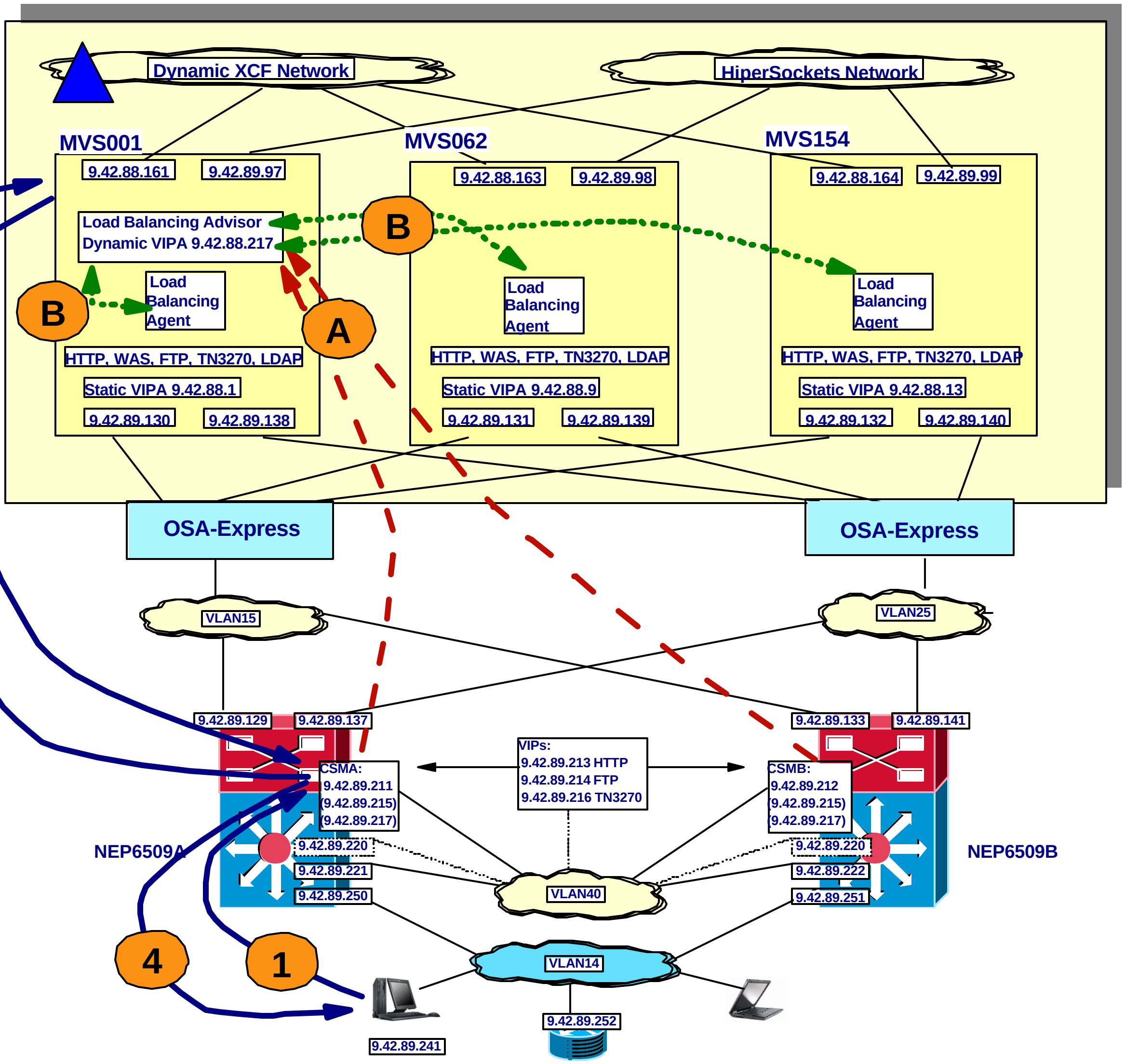


Figure 17 Content Switch Module test configuration with SASP and the z/OS Load Balancing Advisor

The CSMs were deployed directly in the Cisco Catalyst 6509 switches. The testing was performed for two alternatives: Server NAT with Policy-Based Routing (PBR). Note the flows for incoming and outgoing packets (designated by numbers 1,2,3 and 4) are identical to the previous CSM test (see section 9.3.2 CSM Test cases – flow explanations) and will not be not be detailed in this section. Instead, we will review the SASP flows between the CSMs, the Advisor and the Agents in the above diagram:

A. The CSMs, as part of their initialization, establish a TCP connection to the Load Balancing Advisor using the Dynamic VIPA address specified for the Advisor

(2.42.88.217). The CSMs register all the members of the server farms that are configured to use SASP. The Advisor then periodically sends updated recommendations and status to the CSMs. Note that all of processing occurs out of band (i.e. not in the flow of client packets) and does not impact normal traffic flows through the CSM.

B. The Load Balancing Agents connect to the Load Balancing Advisor with a TCP connection during initialization using the Dynamic VIPA address (2.42.88.217). The Advisor and the Agents communicate periodically to exchange registration information (Advisor to Agents) and load balancing status and recommendations (Agents to Advisor).

10 Trademarks and Additional Disclaimers

IBM Corporation
Marketing Communications, Enterprise Systems Group
Route 100
Somers, NY 10589

IBM, IBM logo, OS/390, System/390, S/390, Parallel Sysplex, and ^ log logo are registered trademarks of the International Business Machines Corporation (“IBM”).

Printed in the United States of America. All Rights Reserved

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the products or services available in your area.

You can find additional information via IBM’s World Wide Web server at http://www.ibm.com.

IBM Hardware products are manufactured from new parts or new and serviceable used parts. Regardless, our warranty terms apply. Actual performance and environmental costs will vary depending on individual customer configurations and conditions.

Cisco makes no representations of any kind with respect to any IBM products, or the performance of those products in combination with any Cisco products. All sales of Cisco products, including without limitation any applicable warranty terms, will be exclusively governed by the terms of a Cisco customer agreement.

This publication was produced in the United States. Cisco may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local Cisco business contact for information on the products or services available in your area.

You can find additional information via Cisco’s World Wide Web server at http://www.cisco.com.

Actual performance and environmental costs of Cisco products will vary depending on individual customer configurations and conditions.

Cisco, Cisco Systems, the Cisco Systems logo, Catalyst, and Cisco IOS are registered trademarks or trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries. All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0406R)

IBM makes no representations of any kind with respect to any Cisco Systems, Inc. (Cisco) products, or the performance of those products in combination with any IBM products. All sales of IBM products will be exclusively governed by the terms of an IBM customer agreement.

All other registered trademarks and trademarks are the properties of their respective companies.