IBM @server zSeries

OS/390 and z/OS TCP/IP in the Parallel Sysplex Environment - Blurring the Boundaries

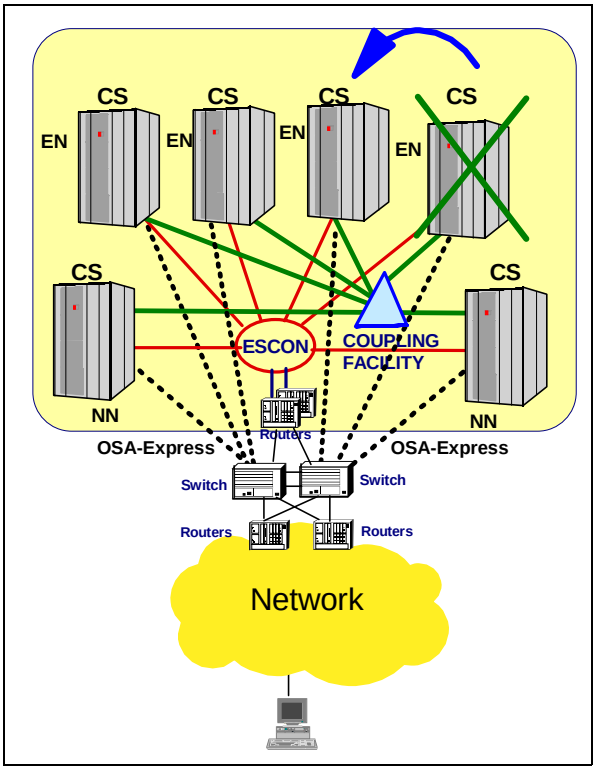
October 2001

Author: Jay Aiken
jaaiken@us.ibm.com

Table of Contents

Blurring the Boundaries

With OS/390® V2R5, TCP/IP for MVS™ became an integral part of OS/390, IBM's flagship operating system for the S/390® Parallel Sysplex® implementation, and the premier clustering solution in the industry. At that time, OS/390 TCP/IP capabilities and functions for the most part were analogous to TCP/IP functions on other platforms. There was one difference, though: OS/390 TCP/IP runs on System/390®, the IBM® server platform for Parallel Sysplex clustering that delivers industry-leading availability and scalability. This availability and scalability continues to be enhanced in z/OS™ TCP/IP.



One of the aspects of both availability and scalability of a clustered solution such as Parallel Sysplex cluster is that clients and the IP network should know as little as possible about the internal make-up of the cluster in terms of individual nodes and application instances. On the other hand, the IP way of assigning IP addresses to specific physical adapter links to some extent exposes the internal composition of the cluster to the network and clients. OS/390 TCP/IP, made available in March 1998, provided much improved scalability on a single multiprocessing system, but in other respects was similar to TCP/IP on other platforms. Since that time, IBM has made continuing improvements in TCP/IP to exploit Parallel Sysplex technology, evolving toward a true cluster IP server and distributed application platform.

The Basics

At the time of the OS/390 V2R5 TCP/IP availability, three technologies were available for use with S/390 Parallel Sysplex technology in support of clustered IP applications: Virtual IP Addresses, Workload Manager enabled Domain Name Server, and Network Dispatcher. The first two were delivered with OS/390 TCP/IP, while the last one executes on an outboard system (AIX® or 2216).

Virtual IP Addresses, or VIPAs, are IP addresses that are independent of any particular network interface. To an external router or other TCP/IP stack, a VIPA appears as simply an address (or a subnet) that is reachable via the hosting stack. When an OS/390 TCP/IP receives an IP packet whose destination is a VIPA that it supports, the packet is merely routed up the stack to the upper-layer protocol (TCP, UDP, or raw socket) as with any other IP packet to an address of a physical link hosted by the stack. VIPAs are advertised via static routes or dynamic routing protocols just as with any other IP address. The benefit of using a VIPA as an application address on an OS/390 TCP/IP with multiple physical links, however, is that a failure of any one link will not disrupt connectivity to the application. As long as there is a network path from a remote client to the TCP/IP hosting the VIPA and the server application, the client and the server application can interact uninterrupted. A VIPA thus provides independence from any particular adapter, but is still for the most part tied to a particular OS/390 TCP/IP stack.

DNS/WLM, the Workload Manager (WLM) enabled Domain Name Server, provides standard name resolution services (converting a name into an IP address) for IP applications in the Parallel Sysplex cluster. OS/390 TCP/IP stacks register their IP addresses with WLM. Application instances register themselves under a generic application name (similar to SNA Generic Resources). The DNS/WLM associates the IP addresses with application names as appropriate. When Workload Manager is being run in goal mode in the OS/390s in the Parallel Sysplex cluster, and a resolution request for an application name arrives, DNS/WLM consults WLM information to determine to which TCP/IP the request should be routed to balance

the workload appropriately. If a client resolves the same name again, a different IP address may be returned if relative available capacity among the server instances has changed. As long as clients in the network resolve the name each time, and do not cache the IP address from a previous request, the workload will be balanced among the available server applications according to relative available capacity. Work may thus be spread across the cluster, Parallel Sysplex cluster presents the appearance of a single platform to well-behaved client applications.

Network Dispatcher is also an approach to workload distribution across a set of application instances in a cluster, but it works with a "cluster IP address", rather than with name resolution. Network Dispatcher (and the similar function in the Cisco Multi-Node Load Balancer, or MNLB) is an external entity adjacent to the Parallel Sysplex cluster, running on either a 2216 or AIX. An agent in the sysplex communicates workload capacities of the nodes hosting the application to the Network Dispatcher node. The Network Dispatcher advertises ownership of the cluster IP address (application IP address) to the routing network. OS/390 and z/OS TCP/IP stacks hosting the application define the same address as a loop-back address, which is not advertised to the routing network. The Network Dispatcher must have a direct link (single IP hop) to each TCP/IP stack hosting the IP application. When a new TCP connection request arrives, Network Dispatcher consults the most recent capacity information received from the WLM agents, selects the appropriate TCP/IP for this request, and forwards the request over the direct link to the selected stack. For selected applications such as Web serving, an application advisor function in the Network Dispatcher will periodically query the application to be sure it is available and responding, to ensure that requests are routed only to functioning server applications. Subsequent traffic from the client to the server is routed through Network Dispatcher to the same TCP/IP, though return traffic from the server application to the client need not flow through the Network Dispatcher. Clients thus see a network presence represented by a single IP address, and the servers that make up the processing that backs up the cluster address are hidden from the clients.

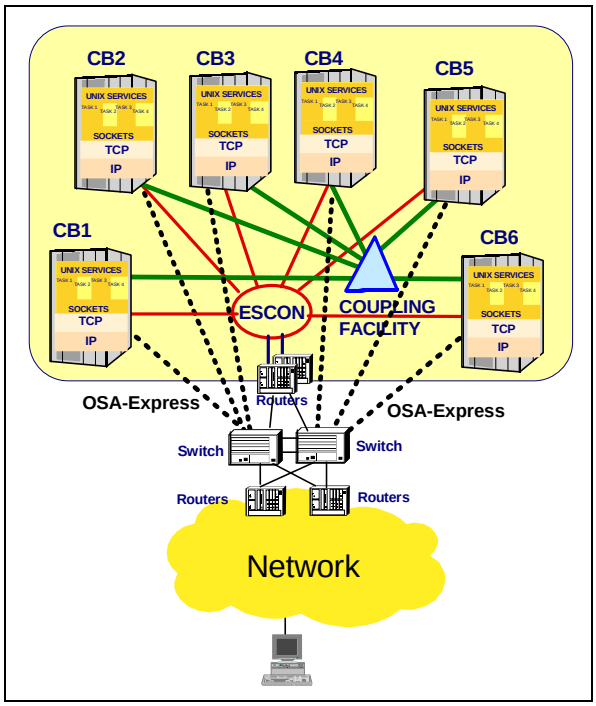
TCP/IP for OS/390 V2R7 — Sysplex Awareness

In OS/390 V2R7, TCP/IP began to exploit Parallel Sysplex functionality to become aware of other TCP/IPs in the sysplex, and to exchange information with these other stacks via MVS XCF Messaging, a basic service of OS/390 that is independent of network protocols. Functions provided by OS/390 TCP as a part of this initial sysplex awareness include MVS System Symbols, Sysplex Sockets, and Dynamic XCF IP connectivity.

MVS System Symbols allows a single configuration profile to be used for more than one TCP/IP. MVS System Symbols used within PROFILE.TCPIP are automatically resolved at TCP/IP initialization or during VARY OBEY processing. Items that are unique to each TCP/IP, such as device and link definitions, may be segregated into separate included files. This reduces the administrative workload for managing multiple TCP/IPs in a sysplex. Even other configuration files such as TCP.DATA that are read and used by applications may now also be managed as a single entity. A "source" file is maintained for the sysplex, and changes are made only to that file. A supplied utility may be executed on each OS/390 (with unique definitions for the included symbols) that reads the "source" file and produces an output file with symbols resolved appropriately for the OS/390 image. While this is more cumbersome than with PROFILE.TCPIP, the process can be automated similar to program compilation. Other than being an EBCDIC text file, there are no particular requirements on the "source" file, and both source file and the image-specific output files may reside in the HFS, an MVS data set, or a member of a partitioned data set.

Sysplex Sockets provides a way for collaborating applications in an OS/390 Parallel Sysplex cluster to determine that both partners reside in the sysplex, to allow for programmed optimizations that might not be possible when the partner application resides elsewhere in the network. For example, information exchanged need not be converted to and from a common format, but may be exchanged in native S/390 format. Similarly, when the traffic flows over a link that is protected with the same physical security as the data center containing the sysplex, such as ESCON®, CTC, or XCF IP links, additional security such as SSL or other encryption may not be necessary, and the overhead may be avoided. TCP/IP stacks in the sysplex use MVS XCF Messaging to exchange their supported IP addresses, so each TCP/IP in the sysplex is aware of all IP addresses active in the sysplex, and can therefore tell when the TCP partner application is using an IP address supported by one of the OS/390 TCP/IP stacks in the sysplex. As new

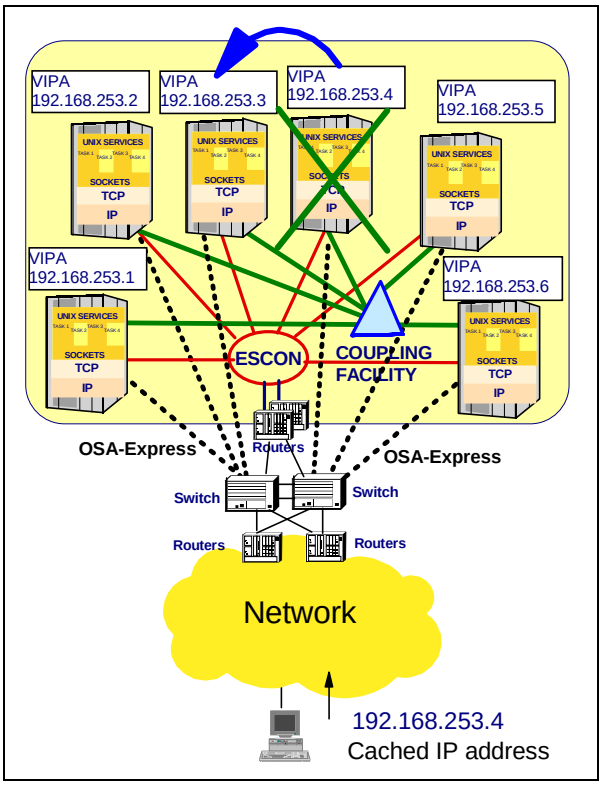
TCP/IP stacks are added with new OS/390 images (or in existing images), information is exchanged with all other TCP/IPs. The information is updated whenever an IP address is deleted from or added to a TCP/IP in the sysplex via VARY OBEY, and the sysplex TCP/IPs are also notified when a TCP/IP is halted or suffers an outage, so the respective IP addresses may be removed from their tables of Sysplex IP addresses.



XCF has been available as an IP transport between TCP/IP stacks in the sysplex since OS/390 V2R5 (and earlier in TCP/IP V3R2). Dynamic XCF builds on both the automatic XCF connectivity of VTAM® and the use of MVS XCF Messaging by TCP/IP to define XCF IP links between TCP/IPs automatically, removing the need for manual definition and updates as new TCP/IPs are added to the sysplex. A single configuration statement (IPCONFIG DYNAMICXCF) on each TCP/IP is sufficient to enable this function. Whenever a new OS/390 or z/OS TCP/IP is added to the sysplex, information exchanged via MVS XCF Messaging allows the existing TCP/IPs to discover this new stack, and appropriate DEVICE, LINK, HOME, and BSDROUTINGPARMS statements are created and activated automatically—with no additional manual definition to the existing TCP/IPs. If dynamic routing protocols such as RIP or OSPF are in use in the sysplex and the attached routing network, a new application host may be added to the sysplex without external physical connectivity (other than the Coupling Facility links or ESCON links necessary for sysplex MVS XCF Messaging), and IP applications may be reached from clients via DNS name resolution to the Dynamic XCF IP address on the new TCP/IP stack. Seamless horizontal growth of the sysplex is thus made much easier for IP applications, especially when combined with DNS/WLM as already described or with technologies soon to follow....

TCP/IP for OS/390 V2R8 — Dynamic Virtual IP Addresses

Since VIPAs are not associated with any physical link, there is no particular reason that a VIPA should be associated with one and only one TCP/IP in the sysplex, other than the normal IP requirement that only one stack may advertise ownership of a given IP address to the attached routing network. When a TCP/IP or its hosting operating system suffers an outage (e.g., power failure), access via the physical links and their associated IP addresses is of course lost. However, a VIPA may be moved to another TCP/IP manually or via automation, and dynamic routing protocols may be used to notify the routing network of the new location of the VIPA without additional manual configuration change in the routers. In OS/390 V2R8, the concept of a Dynamic VIPA was introduced. Simplified configuration definitions allow Dynamic VIPAs to be activated either continuously or on demand from an application. Other TCP/IPs in the sysplex may also be configured as backup for a continuously active Dynamic VIPA, such that the Dynamic VIPA is automatically activated on a backup stack whenever the normally owning TCP/IP suffers an outage. This automatic Dynamic VIPA backup is called VIPA Takeover. VIPA Takeover is applicable when multiple application instances exist, each of which can respond appropriately to any client request. However, some applications need to associate a particular IP address (or VIPA) with a particular application instance, such that client requests to that IP address always go to the same application instance, OS/390 V2R8 supports this as well, with application-initiated Dynamic VIPAs.



VIPA Takeover allows multiple identical application instances to be deployed in different sysplex nodes, each with a unique IP address, while preserving the appearance of almost continuous availability to the clients in the event of an outage on one of the nodes. Web serving is an example of such an application, where multiple instances of a Web server are distributed among sysplex nodes for availability and scalability. Each instance serves the same set of static and dynamic Web pages, so it really does not matter to the client which server handles a particular request. (Client requests may be distributed among the server instances via DNS/WLM, for example.) Each server instance accepts requests to any local IP address (binding to INADDR_ANY, for the technically-inclined). Each TCP/IP serving such an application is configured as the primary owner of a Dynamic VIPA with a simple configuration statement (VIPADEFINE). This single statement causes activation of a virtual device and link, and adds the IP address to the HOME list for the stack. Other stacks are configured as backup for that VIPA with a VIPABACKUP configuration statement, in addition to having their own respective Dynamic VIPAs activated via VIPADEFINE. When an outage occurs at the server side, one of the backup stacks will automatically activate (take over) the Dynamic VIPA from the stack that suffered the outage. The active client connections will be disrupted, but the normal client recovery action is to attempt to reconnect to the same IP address, so the client will very quickly be able to establish a new connection to the backup stack.

As an added benefit, if the backup stack receives client traffic for a connection to the no longer available stack, the backup stack will immediately notify the client that the connection has been terminated, and the client will not have to wait for normal TCP time-outs to discover the connection outage. (Performance studies have shown up to 60% faster TN3270 failure detection and session re-establishment with this mechanism.)

Application-Initiated Dynamic VIPAs were provided in OS/390 V2R8 to address a different set of applications. Some applications define a relationship between client and server that requires the client to reconnect to the same server instance, which means that the IP address of the server instance must remain constant, and be different from the IP address of other server instances. When such an application suffers a failure, or the underlying OS/390 suffers an outage, automation such as OS/390 Automatic Restart Manager may in fact restart the application on a different OS/390 image, depending on available capacity and the restart policy defined for the application. In such a scenario, the IP address must move with the application, and this can only be accomplished with a VIPA. TCP/IP supports this by allowing the application to bind to an IP address that is not currently active on the TCP/IP, or on any other TCP/IP in the sysplex. When the application binds to a nonexistent IP address, the stack will automatically activate a Dynamic VIPA with that IP address. A configuration statement (VIPARANGE) is provided to ensure that a miss-configured application will not be able to activate an IP address that is not appropriate for the installation. As long as all stacks where the application may be started have an appropriate VIPARANGE configuration statement, the application may be restarted on any of the TCP/IPs in the event of a failure, and the Dynamic VIPA will be activated when the application is initialized and issues the bind.

In some cases of this class of application, particularly with purchased, off-the-shelf products, the application cannot be configured to bind to a specific IP address. OS/390 V2R8 provides a utility, which may be added to the JCL, or OMVS shell script that starts the application. The utility issues a command to the stack to activate the Dynamic VIPA (subject to the same VIPARANGE considerations). As long as the application is always started with the same JCL package or shell script, even in automated restart scenarios, the VIPA will be activated on the TCP/IP that will be hosting the application instance.

In terms of presenting a single-node appearance of a clustered IP platform to the network, Dynamic VIPAs are a step forward, since the application address is no longer tied to a single processing node (OS/390 operating system image) in the sysplex. However, when multiple application server instances are deployed for availability and scalability, each instance still has its own IP address, which means the composition of the cluster still shows through to the client population to some degree. Also, moving an active application server instance from one OS/390 node to another within the sysplex is still disruptive, in that either client access to a server must first be quiesced, or active client connections to the server at the time of the move will be terminated.

Because Dynamic VIPAs are considered application addresses, rather than stack addresses, they are not automatically reported to DNS/WLM. If they were, then DNS/WLM might return a Dynamic VIPA to clients for other applications such as Telnet or FTP, which would greatly complicate the problem of relocating the Dynamic VIPA automatically, requiring coordinated movement of a disparate server population. Applications assigned to use Dynamic VIPAs should be configured statically in DNS/WLM.

TCP/IP for OS/390 V2R10 and z/OS — Nondisruptive VIPA Movement, Sysplex Distributor, and Server Bind Control

One of the limitations of VIPA Takeover in OS/390 V2R8 is that when the TCP/IP that normally hosts the VIPA is restored after a failure, the VIPA cannot be moved back to the normal stack immediately without disruption to application connections to that VIPA on the backup stack. Since new client connections may be established to the backup stack while existing connections are being serviced, it may be a very long time before there are no active connections to the VIPA on the backup stack—if ever. OS/390 V2R10 and z/OS provide the ability to move a Dynamic VIPA back to the normal hosting TCP/IP immediately, without disrupting connections to applications on the backup stack. New connection requests are serviced by applications on the normal hosting stack, while IP traffic for existing backup stack connections continues to be routed to the backup stack.

z900 and S/390 provide the ability to run multiple OS/390 and z/OS images in a single processor complex via the Logical Partitioning function, or LPAR. High-speed connectivity functions like the Open Systems Adapter-Express may be shared between TCP/IPs in different LPARs, and the OSA performs a routing function on incoming IP traffic based on destination IP address. This can cause a problem for Network Dispatcher and other outboard solutions, because two such TCPs cannot both be using the same cluster IP address—the OSA would not be able to determine to which TCP/IP to route the incoming traffic. Also, the cost of direct physical connectivity from Network Dispatcher to all IP application hosts, and the difficulties of configuring the external Network Dispatcher node for new application hosts, make the external distribution function undesirable for some customers. The Sysplex Distributor function in OS/390 V2R10 and z/OS address all of these problems, by bringing the distribution function into OS/390 and z/OS TCP/IP.

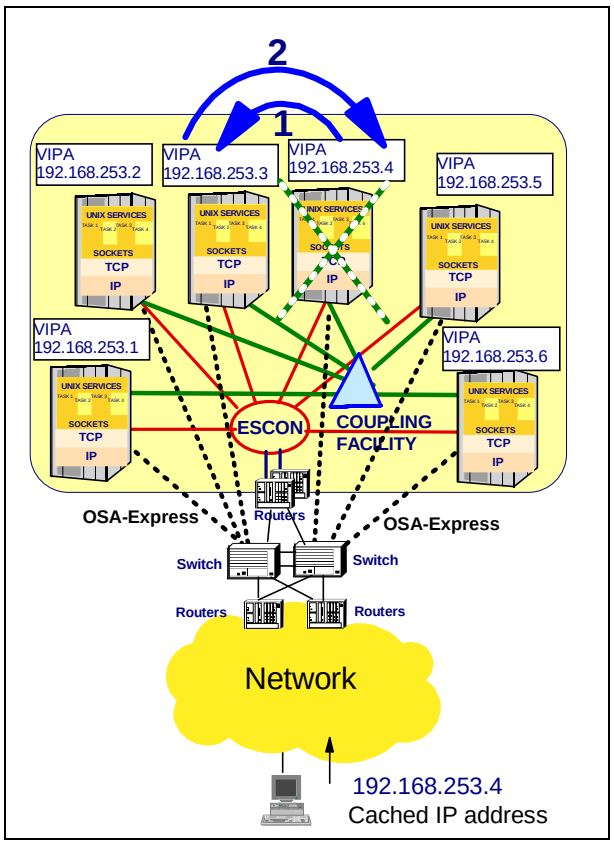
Note: Such OSA-related restrictions will also be addressed via Generic Resource Encapsulation (GRE), an industry standard mechanism for encapsulating packets in other IP packets with a different destination IP address. GRE requires that the outboard workload balancing solutions also support GRE, as well as the receiving TCP/IP stack. GRE was made available via PTF on OS/390 V2R10 TCP/IP when it became generally available in September 2000.

Both nondisruptive movement of Dynamic VIPAs and Sysplex Distributor are oriented towards TCP connections. They do not apply to UDP applications, where the relationship between client and server over time, if any, is maintained solely at the application level if at all. The combination of the fact that most server applications allow requests to any IP address (binding to INADDR_ANY in socket terms) and the prevalent usage of SOURCEVIPA means that many UDP server applications may not be able to use Dynamic VIPAs unaided, since the source address of a response may not match the destination address of the corresponding request, a situation that many UDP clients will not

tolerate. Server Bind Control, introduced in OS/390 V2R10 and made available via PTF on V2R8, allows OS/390 TCP/IP to restrict such servers to a single IP address through TCP/IP configuration, without server program modification, thus allowing such servers (e.g., DNS/WLM) to share in the benefits of Dynamic VIPAs.

Nondisruptive VIPA Movement in OS/390 V2R10 and z/OS is useful for both of the Dynamic VIPA scenarios. For VIPA Takeover, when the normally hosting TCP/IP is restored, the Dynamic VIPA is immediately activated on the restored stack. The backup stack notifies the restored stack of all active TCP connections, so traffic for those connections will continue to be routed to the backup stack as long as the connections are active. The normal stack, not the backup stack, handles new client connection requests for the Dynamic VIPA. Assuming normal connection duration of minutes or hours (or even days), eventually all connections to the backup stack for the VIPA will end normally, and the VIPA can be deactivated.

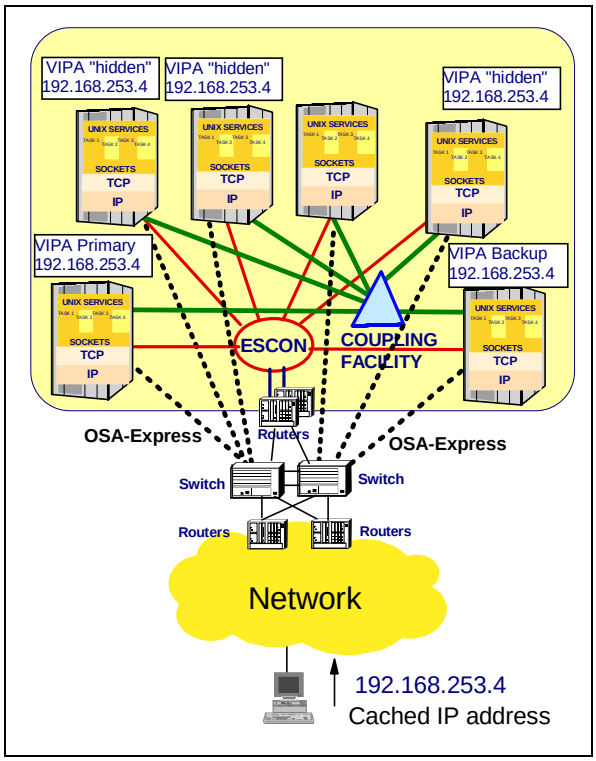
As described above, application-initiated Dynamic VIPAs are tied to a specific application instance. In addition to short-term fluctuations, application traffic patterns and workloads may vary over time, such that it may be desirable to move an application instance to a different OS/390 or z/OS in the sysplex to provide better long-term balance. Adding new application host processors to the sysplex is another case where moving existing application instances may be desirable. Nondisruptive VIPA Movement in V2R10 and z/OS allow such planned application movement to take place nondisruptively, as far as the clients are concerned, simply by starting another copy of the same application instance in another OS/390. The Dynamic VIPA is activated by the application (or JCL or OMVS shell script) on the new TCP/IP, and advertised via dynamic routing protocols to the routing network, but TCP traffic to the existing instance continues to be routed to the former stack until all connections there end naturally. The former application instance may then be shut down, and the new instance now is servicing all the client requests.



Sysplex Distributor was made available in OS/390 V2R10 and in z/OS V1.1 to provide additional customer network configuration flexibility, particularly as related to network attachment costs and sharing of OSAs among LPARs. In addition, Sysplex Distributor provides more real-time consultation with Workload Manager for cluster node capacities, as well as consulting the Service Policy Agent to allow the server selection to be influenced by policies governing Service Level Agreements.

Sysplex Distributor builds on Dynamic VIPAs from V2R8. A particular TCP/IP is configured as a routing stack with a Dynamic VIPA, and one or more other stacks may be configured as backup routing stacks for that Dynamic VIPA. An additional configuration statement (VIPADISTRIBUTE) on the primary stack designates the identifying server port number(s) and which TCP/IPs (target stacks) in the sysplex will be hosting server applications. This configuration information is distributed automatically to all the target stacks via MVS XCF Messaging, so only the routing stack needs the additional VIPADISTRIBUTE statement. The target stacks each activate the same VIPA address, but do not advertise it to the routing networks. When a server application binds to the designated application port on one of the target stacks,

the target stack notifies the routing stack via MVS XCF Messaging that the application is available for work.



When the routing stack receives a connection request from the client for a Distributed VIPA, the routing stack first determines which target stacks have active server applications listening on the port. The routing stack then consults Workload Manager information on relative capacities, and adjusts those capacity values with information from the Service Policy Agent (for Network Quality of Service considerations), before selecting the target stack. The connection request is then routed via the appropriate Dynamic XCF IP link to the target stack and the server application. The routing stack keeps track of active connections, so that future client IP traffic for a connection is routed to the same target stack and application. The target stack notifies the routing stack when the connection ends, to allow the routing table entry to be deleted.

If the routing stack should suffer an outage, normal VIPA Takeover mechanisms will move the Dynamic VIPA to a backup routing stack. The remaining target stacks notify the backup routing stack of active connections, so that clients connected to those target stacks will not see an outage at all (other than possible TCP retransmission's during the takeover process). When the primary routing stack is restored, nondisruptive VIPA movement is used to restore the routing function to the restored stack, and the backup stack no longer needs to serve as the routing stack, again with no visible disruption to the clients.

The benefits of Sysplex Distributor over external IP workload balancing solutions are thus as follows:

- Only the primary routing stack and the backup routing stack(s) need to be connected to the routing network. Target stacks may be connected to the routing network, and traffic from server to client will take the least-cost route, but application hosts need not have physical connectivity to the external routing network.
- Since only one routing stack receives inbound IP traffic from the external network for the Distributed VIPA, and distributes traffic to target stacks using Dynamic XCF IP links, there are no problems with OSA adapters shared among two or more TCP/IP stacks. The OSA always routes inbound traffic to a single (routing) stack.
- Target stacks notify the routing stack automatically when a server application binds to the designated application port, and also when the server closes the listening socket, so the routing stack is always aware of which target stacks actually have applications listening. All application types are thus covered without the need for application-unique advisor programming at the routing stack.
- In addition to capacity information from Workload Manager, Sysplex Distributor uses policy and Quality of Service information from the Service Policy Agent to determine how to distribute the work. For example, if an individual server is not meeting its Service Level Agreement, new work is directed at other server instances that are meeting their SLA instead, rather than making the lagging server's problem worse by directing additional work to it

Server Bind Control actually addresses two different requirements. The first requirement concerns different and incompatible server application instances identified by the same well-known port number, when no such server instance can be configured to use a specific IP address. One example would be OTELNET and TN3270, both of which use well-known port 23, both of which accept requests to any IP address (binding to INADDR_ANY in socket terms), and which use different application protocols. If the servers use the same application protocol, normal port sharing functions would allow the server instances to coexist on the same TCP/IP stack. However, the stack balances connections between server instances sharing the same port, and has no way to know that a client needs to use one particular server over another. Before Server Bind Control, this requirement was addressed through the use of different TCP/IP stack instances, associating one server instance type with one stack, and the other server instance type with the other stack, such that DNS/WLM only returns

addresses appropriate for the name of the particular server type. Note that if the server instances themselves could bind to different IP addresses, there would be no problem running both server instances on a single stack, since the destination IP address would uniquely identify the type of server, and the DNS could be configured to supply the correct IP address for each server type.

Server Bind Control allows the OS/390 TCP/IP stack to be configured to address this. A modification to the PORT configuration statement allows an IP address to be specified and associated with a particular job name. If the server job binds its listening socket to a particular IP address, nothing new occurs. However, if the server job binds its listening socket to INADDR_ANY, OS/390 TCP/IP converts the bind to use only the specified IP address. In the example above, OTELNET would be configured for one IP address, and TN3270 would be configured for a different address, with an appropriate DNS name to IP address configuration. Both servers can now run on the same TCP/IP stack using the same well-known port, and clients are automatically directed to the desired server instance.

Note that this also addresses the concern identified above with some UDP server applications. If such a server application is configured to the TCP/IP stack to be bound to a specific IP address, the problem of SOURCEVIPA causing a response to use a different source address than was specified on the corresponding previous request goes away, because the socket is bound solely to the designated address.

The specified address may also be a Dynamic VIPA, and need not already be active on the stack at the time that the application establishes its listening socket. In other words, applications that bind to INADDR_ANY can now use application-initiated Dynamic VIPAs, with all the benefits of such use, including the ability to restart the application on another OS/390 image (with an appropriate TCP/IP configuration already in place) and have the Dynamic VIPA move with the application. This configuration is available for both TCP and UDP applications, since the PORT statement differentiates between the two.

TCP/IP for z/OS V1R2 — Fast Connection Reset, HiperSockets, and HiperSockets Accelerator

IBM's @server family has undergone significant changes, and some of these changes are reflected in the name change of S/390 to zSeries and z900, and OS/390 to z/OS. TCP/IP for z/OS includes all of the functionality of OS/390 V2R10 TCP/IP, with significant additional enhancements. TCP/IP continues its annual functional enhancement cycle, so the first set of additions appear in z/OS V1R2.

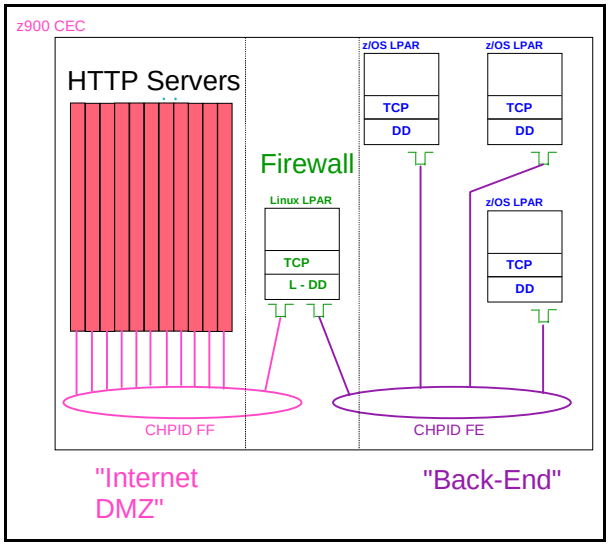
Fast Connection Reset after System Failure addresses availability for applications supported by Sysplex Distributor. Though rare, failures of an entire operating system image or a TCP/IP stack (such as through power outages) do occur, and this enhancement is designed to minimize the duration of such outages where possible. Clients connected to a server application will normally attempt an immediate reconnection when the client TCP/IP detects a failure of a TCP connection and notifies the client application. However, TCP/IP is designed to be very tolerant of failures in the intermediate routing network, so the client stack will not detect a failure unless it does not receive an acknowledgment of transmitted data—or unless it is told by the partner TCP/IP that the connection has ended. VIPA Takeover provided quicker notification when the client is sending data, by returning an immediate TCP Reset when connection data arrives at a stack for a connection that the stack does not own, bypassing the normal TCP retransmission time-outs that may last for minutes. However, with client/server applications such as are typical in e-Commerce, the client is most often waiting for information to be sent back from the server, and in this scenario, the client could wait forever. (This is true of any client/server application, not just those associated with e-Commerce, of course...)

With Fast Connection Reset, if a stack hosting applications served by Sysplex Distributor suffers an outage, the distributing TCP/IP will be notified as usual, and the distributing TCP/IP will now send a TCP Reset to all clients having active connections with the application host that is no longer available. Clients which are waiting for data from the server will immediately become aware of the failure, and may initiate a new connection to another server that is still available. While there is still a disruption of the client connection, the time to get back into session with a functioning server may be greatly reduced, and thus the much shorter duration of the outage leads to improved availability.

Prior to **HiperSockets™**, OSA-Express supported the LPAR-to-LPAR communication function. When two TCP/IP stacks shared an OSA-Express using QDIO, and one stack sent data to the other stack, OSA-Express was aware that both ends of the relationship were directly attached. The OSA-Express copied the data to internal adapter storage as usual, but instead of sending the data out onto the LAN medium, the OSA-Express copied the data directly to the input queue of the receiving TCP/IP. This is much faster than with other connectivity such as XCF or ESCON. Note that this LPAR-to-LPAR communication could take place regardless of whether the operating system images were part of an OS/390 sysplex or not.

HiperSockets does the same thing—except in processor hardware and microcode. HiperSockets requires IBM **@server** zSeries 900 EC level J10607 or J10608. With this level, the functionality of an OSA-Express adapter is incorporated into the processor itself. There is no external connectivity with HiperSockets, but any TCP/IPs in the same z900 server which have access to the same HiperSockets "LAN" may exchange IP data via the HiperSockets device. The difference in performance comes from this: instead of moving the data across the high-speed I/O bus to adapter memory, and then back across the I/O bus to processor memory accessible to the receiving stack, HiperSockets performs a single copy of the data directly from the sending buffers to the receiving buffers — with performance equivalent to a "Move Character Long" instruction, after the initial setup and locating the target receiving buffers. Further, since any processor may perform the HiperSockets function, adding processors simply improves the parallelism of HiperSockets, and thus increases the throughput for many parallel data transfers between different pairs of TCP/IP stacks.

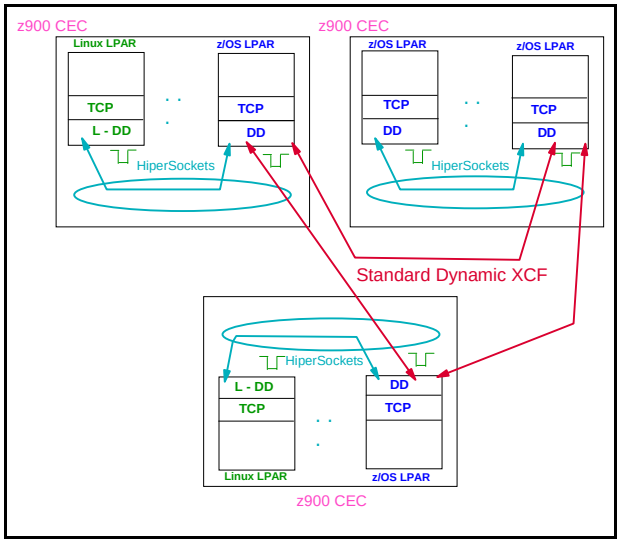
To allow for isolation among different groups of operating system images, up to four different HiperSockets devices may be configured in hardware, and made accessible to the logical partitions (LPARs) and Virtual Servers on an individual basis. A TCP/IP stack in an operating system image, whether Linux®, z/VM™, or z/OS, may thus be able to make use of anywhere between zero and four HiperSockets devices, depending on how the z900 hardware has been configured for that LPAR, and how the stack itself is configured. Because the processor is performing the transfers, with all the associated memory access protection, data traversing one HiperSockets device is in no way visible to any stack not connected to that particular device. This could be useful in server consolidation scenarios, where multiple front-end servers are consolidated onto a zSeries platform and still protected via firewall from the back-end z/OS-based corporate data:



From the point of view of TCP/IP for z/OS, a HiperSockets device may have two manifestations. First, the device may be configured as a normal device, except that the basic device driver definitions (Transport Resource List Entry, or TRLE, in VTAM-speak) will be built automatically and dynamically, so that it is only necessary to add normal DEVICE, LINK, and HOME statements to the TCP/IP profile, and to configure OMPROUTE if dynamic routing protocols are being used. Note that the z/OS TCP/IP stacks configured in this way gain access to the HiperSockets device without respect to membership in any sysplex or Parallel Sysplex cluster.

The second manifestation builds on Dynamic XCF, and simply substitutes the HiperSockets device for a dynamically-built XCF link between two LPARs that both have connectivity to the same HiperSockets device being used for Dynamic XCF in the same sysplex or Parallel Sysplex cluster. (A Coupling Facility is not required, as long as the two z/OS LPARs are configured to be part of the same sysplex.) If only one HiperSockets device is configured to an LPAR, no additional definitions are required in z/OS TCP/IP to make this substitution of a HiperSockets link for an XCF link under Dynamic XCF. If more than one HiperSockets device is configured to be accessible to the LPAR, then the device driver (VTAM) must be told which one to use for Dynamic XCF (a single VTAM option), and the other devices may be used via the normal DEVICE/LINK/HOME statements described above.

Pictorially, the Dynamic XCF usage of HiperSockets and XCF within a multi-z900 Sysplex could look roughly as follows:



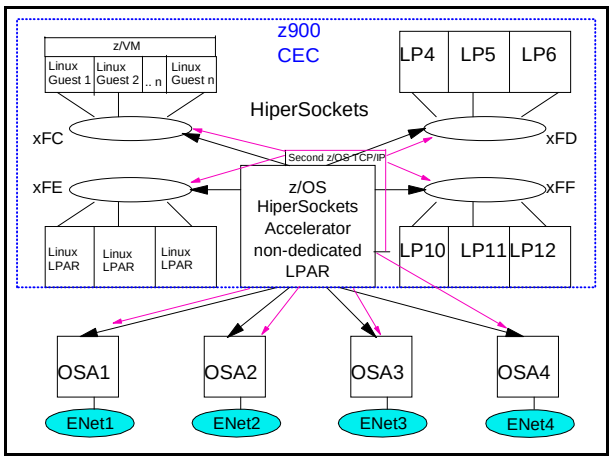
Note also from the diagram that Linux and/or z/VM may share the same HiperSockets device with z/OS LPARs.

For additional detail on HiperSockets hardware and supporting operating system environments, refer to **ibm.com**/eserver/zseries/networking/hipersockets.html

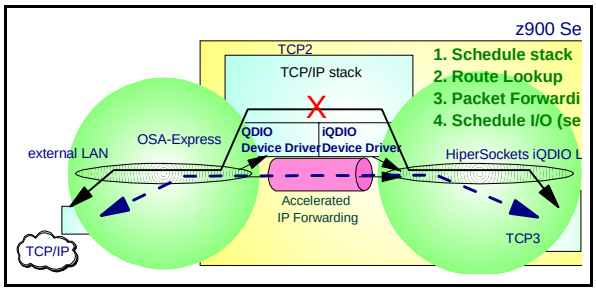
HiperSockets Accelerator allows very fast transfer of IP packets between an OSA-Express device and a HiperSockets device, both inbound and outbound.

With the increase in the number of operating system images on a z900 processor complex, there is an increase in the scale of network connectivity. OSA-Express may be shared among many operating system images, whether z/OS, z/VM, or Linux in any combination, but there is a limit of how many partitions may share a single adapter (15), and a maximum number of TCP/IP stacks per adapter (80). If the TCP/IP stacks are not individually heavily loaded, providing direct OSA-Express connectivity to each partition and TCP/IP stack may mean that the individual OSA-Express adapters are underutilized.

When the HiperSockets Accelerator function is configured, a z/OS TCP/IP stack may function as "owning" the OSA-Express adapter or adapters, and will perform very efficient routing functions via HiperSockets devices to other operating system images and TCP/IP stacks (z/OS, z/VM, or Linux), as shown in the following diagram. A second z/OS image may also be configured to support HiperSockets Accelerator to the same HiperSockets device(s) to provide a redundant path in case one of the z/OS HiperSockets Accelerator images suffers an outage.



Functionally, HiperSockets Accelerator works by maintaining an awareness of when an IP packet is routed between an OSA-Express adapter and a HiperSockets device. The first such IP packet follows the normal routing path through the IP layer of the z/OS TCP/IP stack. At that time, routing information is passed to the device driver layer, so that subsequent packets arriving from the same device and destined for the same IP address will be routed more efficiently, completely bypassing the IP layer of the TCP/IP stack. This process is illustrated in the following diagram, where the upper "path" is taken for the first IP packet, and the lower "path" is taken for all subsequent IP packets from the same device to the same destination IP address.



Parallel Sysplex TCP/IP through z/OS V1R2 — In Summary

TCP/IP for z/OS has made great strides since OS/390 V2R5. It is no longer a single-node stack in the traditional sense. TCP/IP for z/OS uses Parallel Sysplex facilities to maintain awareness of the existence of all other TCP/IPs in the sysplex, and exchanges information to reduce the problem of configuring multiple stacks in a Parallel Sysplex cluster. TCP/IP can automatically establish IP connectivity to other TCP/IPs in the sysplex via Dynamic XCF, and has enhanced this with higher-speed HiperSockets. Use of MVS System Symbols in configuration files allows common profiles to be maintained with less effort. Configuration of new Dynamic and Distributed VIPAs has been simplified, often to a single profile statement, and the need for coordinated definition changes has been reduced greatly through Dynamic XCF and Sysplex Distributor. Sysplex Distributor offers additional IP workload distribution function and flexibility, and now provides additional reduction of apparent outage duration at the client through Fast Connection Reset. Fast Connection Reset ensures that clients of server applications using Dynamic VIPAs and Sysplex Distributor are notified quickly of a connection failure. Server Bind Control removes restrictions that in the past resulted in deploying multiple TCP/IP stacks in the same OS/390 image, and extends the benefits of Dynamic VIPAs to additional UDP-based servers.

Future efforts such as Server-Controlled Affinity and Content-Based Routing may allow new application workloads to be distributed for availability and scalability while maintaining proper server-client relationships with reduced or minimal cost. Connection Recovery may some day provide true continuous availability to clients even in the face of application and endpoint TCP outages.

OS/390 TCP/IP and z/OS TCP/IP are well on the way to providing a single-system image to clients for clustered IP server applications.

IBM Corporation
Marketing Communications, Server Group
Route 100
Somers, NY 10589

Printed in the United States of America, 10/01

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 **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.

IBM, the IBM Logo, the e-business logo, AIX, ESCON, HiperSockets, MVS, OS/390, Parallel Sysplex, S/390, System/390, VTAM, zSeries and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries or both.

Linux is a registered trademark of Linus Torvalds.

Lotus Notes is a trademark owned by Lotus Development Corporation.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Windows NT is a registered trademark of Microsoft Corporation.

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

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

GM13-0026-00

ZSW03329-USEN-00