HiperSockets Usage by z/OS

Session 17294

Tuesday 8/11 Europe 1 at 1:45pm

Linda Harrison

lharriso@us.ibm.com

#SHAREorg

SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

AIX*	IBM i*	POWER7*	PureSystems	Tivoli*
BladeCenter*	IBM logo*	Power Systems	Storwize*	WebSphere*
DB2*	Informix*	PowerVM	System Storage*	zEnterprise*
Easy TIER*	PartnerWorld*	PureApplication	System x*	
IBM*	Power*	PureFlex	System z*	

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Windows Server and the Windows logo are trademarks of the Microsoft group of countries.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

* Other product and service names might be trademarks of IBM or other companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

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 product or services available in your area.

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

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Abstract

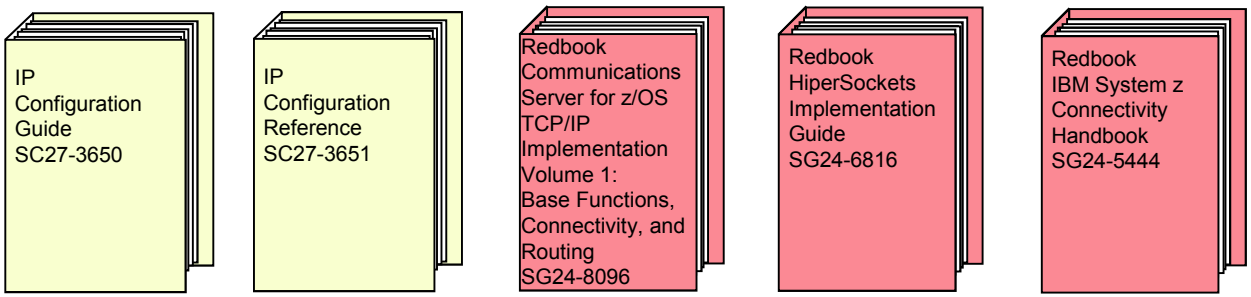
HiperSockets are virtual LANs provided by the z Systems platform without any additional fee. Because they are internal and virtual LANs on the z System there is no exposed cable or wire and therefore provide a secure connection between LPARs in the same z Systems machine. This session will detail what z Systems HiperSockets are and how they can be used between LPARs. HiperSockets implementation on z/OS will be explained. Some Linux on System z implementations are resistant to use dynamic routing. Without dynamic routing HiperSockets and OSA connections between z/OS and Linux on System z will need static routing definitions. HiperSockets routing options between z/OS and Linux on System z will be discussed.

This document details the HiperSockets in z/OS running standalone on a System z processor LPAR. For the additional tasks required for configuring HiperSockets in z/OS running as a guest under z/VM, or HiperSockets in z/OS running as a guest under z/VM using Guest LAN(s) support, or for the tasks required for HiperSockets in z/Linux or in z/VM, please consult the “HiperSockets Implementation Guide” Redbook, SG24-6816

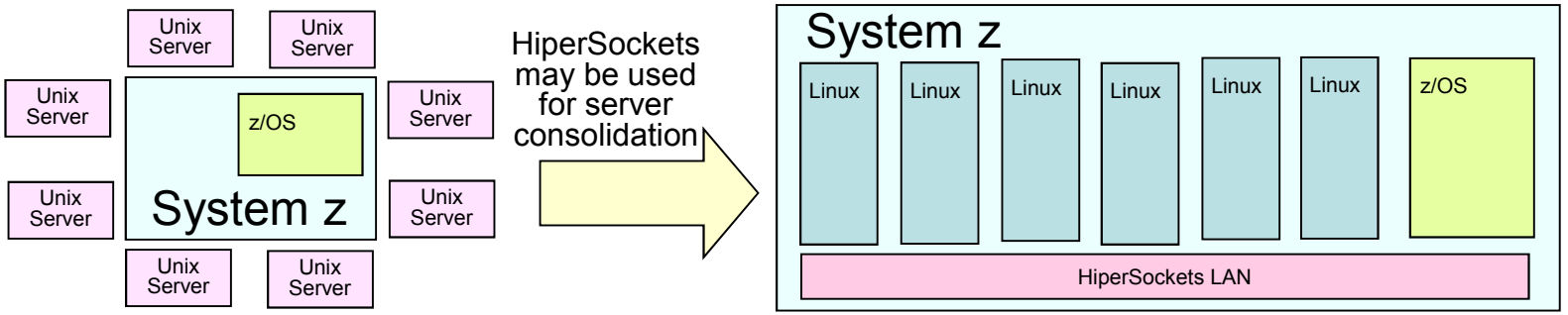
Agenda

- HiperSockets Overview
- OSA vs. HiperSockets vs. RoCE
- Routing Basics
- Sending and Receiving Data
- ARP and IP Routing
- HiperSockets Between z/OS and Linux
- More Information
- Appendicies
  - More Configuration Details and Commands
  - DynamicXCF Details

HiperSockets Overview

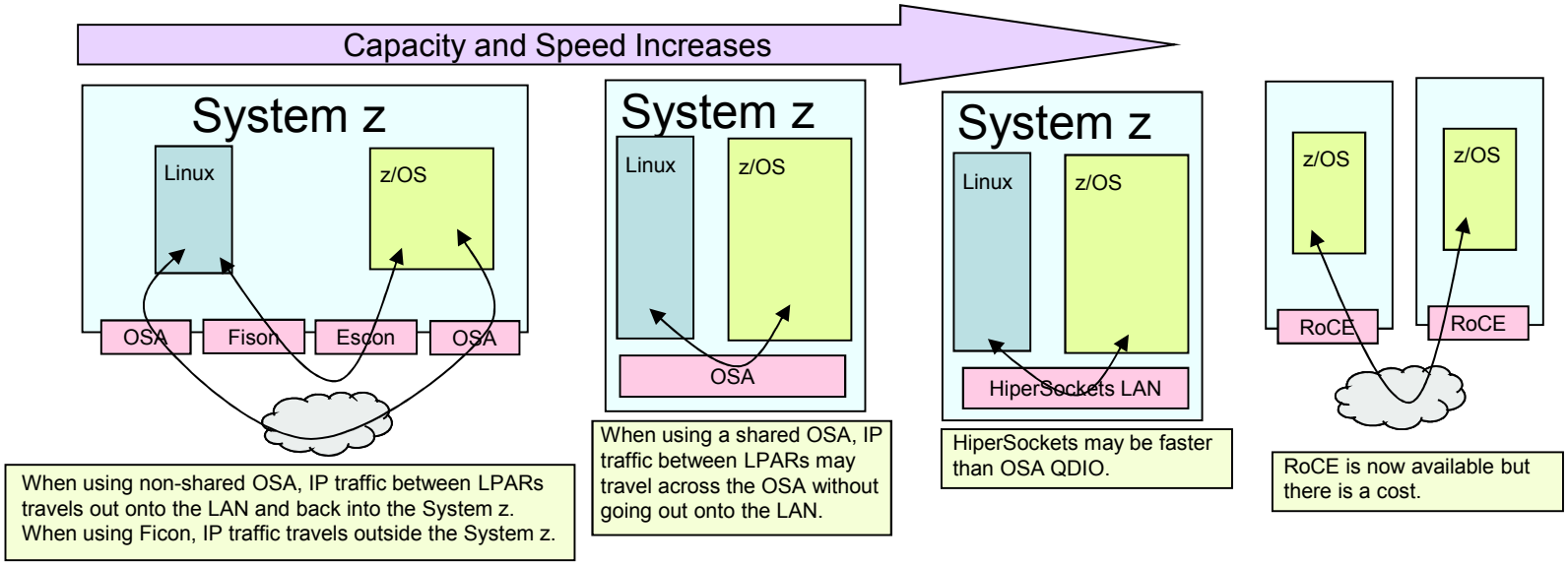


HiperSockets (iQDIO)



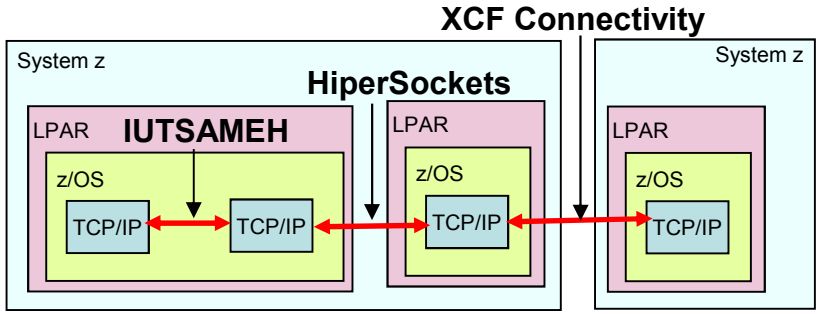
- HiperSockets = Internal Queued Direct Input Output (iQDIO)
  - Was developed from the OSA QDIO architecture
    - Is limited to TCP/IP protocol only to/from z/OS (z/VM and Linux on System z support Layer 2)
  - Also known as HiperSockets device or System z internal virtual LAN or HiperSockets LAN
  - LPAR to LPAR communication via shared memory
    - High speed, low latency, similar to cross-address-space memory move using memory bus
    - Provides better performance than channel protocols for network access.
  - Multiple HiperSockets may be configured as internal LANs on the System z box.
  - A HiperSockets LAN may be configured to be part of TCP/IP DynamicXCF.
    - A TCP/IP stack may only define a single HiperSockets LAN for DynamicXCF.
      - Some TCP/IP stacks may use one HiperSockets LAN for DynamicXCF connectivity while other TCP/IP stacks use a different HiperSockets LAN for DynamicXCF connectivity.
    - Not recommended for some LPARs to define a HiperSockets LAN for DynamicXCF and other LPARs to manually define the same HiperSockets LAN.
  - Common Lookup Table is stored in the Hardware System Area, the same as QDIO.
    - HiperSockets LAN, IP address, TCP/IP stack

TCP/IP LPAR to LPAR Communication Path



- The System z does a direct move from one LPAR's memory to another LPAR's memory.
- HiperSockets is usually the fastest connection possible between LPARs
  - Now RoCE offers an additional option but has a cost associated with it.
- HiperSockets Maximums
  - 32 HiperSockets LANs on zEnterprise z196 and later
  - 16 HiperSockets LANs on System z9 and z10
  - 32 HiperSockets LANs on z196 and later
  - Current limits are documented in Redbook "IBM System z Connectivity Handbook", SG24-5444
- HiperSockets CHPIDs may span across multiple Logical Channel SubSystems (LCSS)

TCP/IP DynamicXCF Transport Choices



- TCP/IP DynamicXCF is capable of dynamically creating multiple device, link, and interfaces all with the same IPv4 or IPv6 address:
  - Same host: device IUSAMEH, link EZASAMEMVS (IPv4), interface EZ6SAMEMVS (IPv6)
  - Hipersockets: device IUTIQDIO, link IQDLNK*nnnnnnnn* (IPv4), interface IQDIOINTF6 (IPv6)
    - where *nnnnnnnn* is the hexadecimal representation of the IP address specified on the IPCONFIG DYNAMICXCF statement
  - XCF connectivity: device CPName, link EZAXCF*nn* (IPv4), interface EZ6XCF*nn* (IPv6)
    - where *nn* is the 2-character &SYSCLONE value
- TCP/IP DynamicXCF automatically chooses the fastest path:
  - Same host is used between TCP/IP stacks inside the same LPAR
  - HiperSockets is used between TCP/IP stacks inside same System z CEC (when configured)
  - XCF connectivity is used between TCP/IP stacks outside the CEC
- VTAM start-options required for TCP/IP DynamicXCF use of HiperSockets:
  - IQDCHPID=nn (where nn is the HiperSockets LAN CHPID, ie. FA)
  - XCFINIT=YES (required for TCP/IP DynamicXCF with or without HiperSockets)
    - Generates XCF device ISTLSXCF in VTAM
    - Requires prerequisite Start Options like HPR=RTP, which requires minimal APPN enablement.
- TCP/IP Profile options required to define DynamicXCF:
  - IPCONFIG DYNAMICXCF 10.1.2.101 ...
  - IPCONFIG6 DYNAMICXCF 2001:0DB8:1:0:50C9:C2D4:0:1 ...
- When HiperSockets DynamicXCF is configured:
  - HiperSockets DEVICE/LINK/INTERFACE/HOME and VTAM TRLE are all dynamically built

Defining HiperSockets (iQDIO)

- Define HiperSockets LAN in HCD/IOCP
  - CHPID TYPE=IQD
    - CHPARM determines MTU
- VTAM Start Options
  - IQDIOSTG – optionally modify the amount of Read Storage
  - IQDCHIP – Do not take the default!
    - IQDCHIP=NONE – if you do not want a HiperSockets LAN used for DynamicXCF traffic.
    - IQDCHIP=chpid – if you do want a HiperSockets LAN used for DynamicXCF traffic.
- PROFILE.TCPIP
  - INTERFACE IPAQIDIO (or IPAQIDIO6) and START
  - or DEVICE MPCIPA, LINK IPAQIDIO, HOME, and START
  - BEGINROUTES ROUTE – define HiperSockets LAN to static routing
  - GLOBALCONFIG
    - AUTOIQDX – enables HiperSockets Integration with IEDN
    - IQDMULTIWRITE – enables HiperSockets Multi-Write
    - IQDVLANID – to define a DynamicXCF HiperSockets VLAN ID
    - ZIIP IQDIOMULTIWRITE – offloads HiperSockets processing to zIIP engine(s) (requires IQDMULTIWRITE)
  - IPCONFIG
    - QDIOACCELERATOR – enables QDIO/iQDIO Accelerator
    - IQDIOROUTING – outdated parameter for original HiperSockets Acclerator support
      - *Use QDIOACCELERATOR instead*
- OSPF configuration file
  - Define HiperSockets LAN to dynamic routing

Dynamically Created

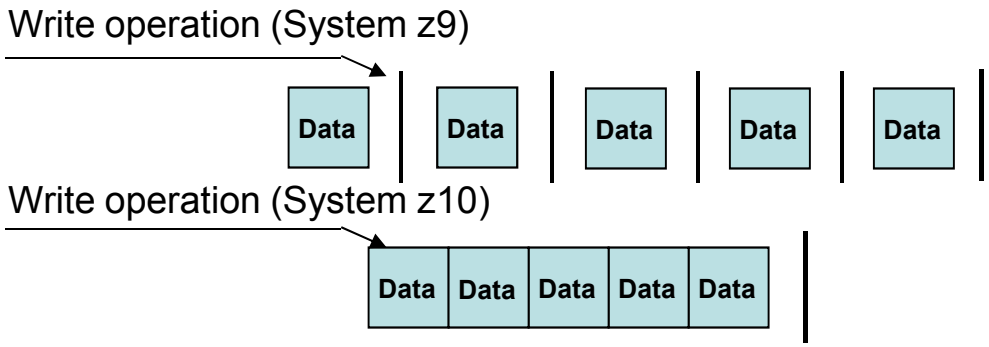
- TRLEs in ISTTRL Major Node (where xx = CHPID)
  - Manual HiperSockets
    - Device/Link MPCIPA/IPAQIDIO or Interface IPAQIDIO6 TRLE = **IUTIQDxx**
    - Interface IPAQIDIO TRLE = **IUTIQ4xx**
  - IPv4 DynamicXCF HiperSockets TRLE = **IUTIQDIO**
  - IPv6 DynamicXCF HiperSockrets TRLE = **IQDIOINTF6**
  - IPv4 HiperSockets Integration with IEDN (IQDX) TRLE = **IUTIQXxx**
  - IPv6 HiperSockets Integration with IEDN (IQDX) TRLE = **IUTIQ6xx**
- Device, Link, and Home (where where nnnnnnnn is hexadecimal representation of the IP address)
  - IPv4 DynamicXCF
    - DEVICE = **IUTIQDIO**
    - LINK = **IQDIOLNKnnnnnnnn**
    - HOME for IQDIOLNKnnnnnnnn
- Interface (where xx = CHPID)
  - IPv6 DynamicXCF INTERFACE = **IQDIOINTF6**
  - HiperSockets Integration with IEDN (IQDX)
    - IPv4 Interface IPAQIQDX = **EZAIQX*xx***
    - IPv6 Interface IPAQIQDX6 = **EZ6IQXxx**

Multiple Write Facility and zIIP Offload

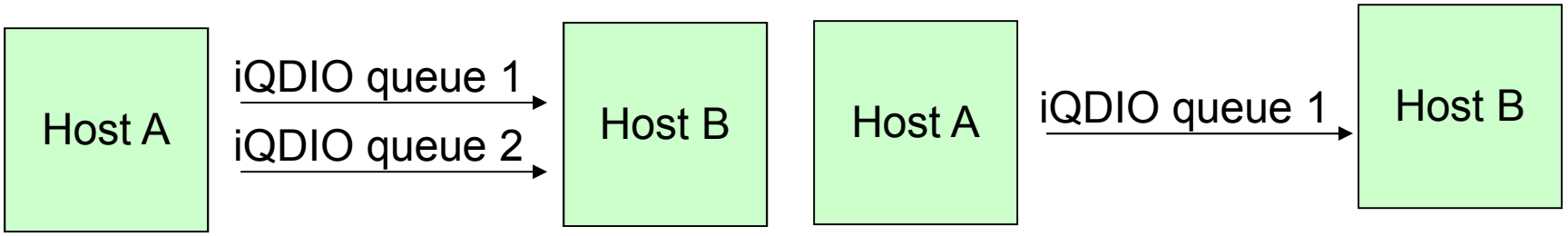
- HiperSockets can move multiple output data buffers in one write operation
  - Reduces CPU utilization

GLOBALCONFIG IQDMULTIWRITE ZIIP IQDIOMULTIWRITE

- Multiwrite operation can be offloaded to a zIIP
  - Only for TCP traffic that originates in this host
    - **Only large TCP outbound messages (32KB+)**



HiperSockets Completion Queue



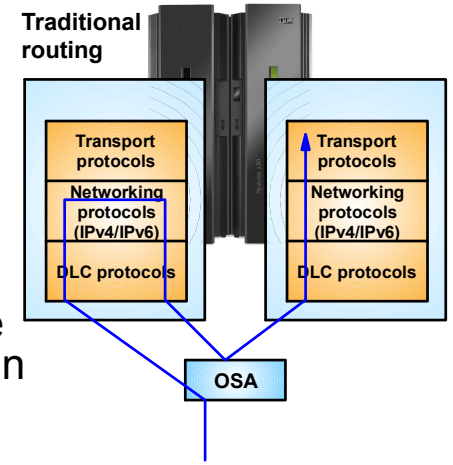
When the remote side can receive data volume then data is sent synchronously.

When the remote side cannot receive data volume then data is sent asynchronously.

- Automatic capability in z/OS V1.13 or later.
- HiperSockets transfers data synchronously if possible and asynchronously if necessary.
- Ultra-low latency with more tolerance for traffic peeks.
- Requires zEnterprise z196 or later.

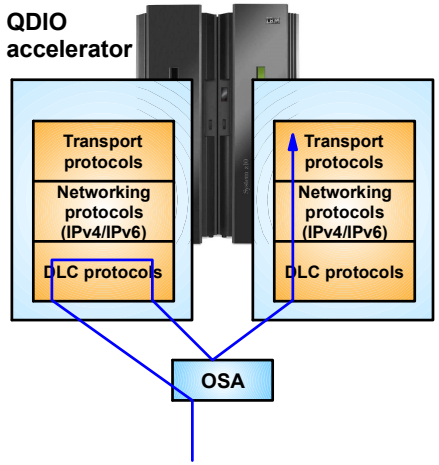
QDIO Accelerator

- Accelerator support includes all combinations of QDIO and iQDIO traffic
  - When traffic is routed through z/OS.
  - Inbound over OSA or HiperSockets and Outbound over OSA or HiperSockets
- The first packet will travel up thru QDIO to the Accelerator stack and down thru iQDIO device drivers to reach the backend LPAR IP address. After that first packet, all the rest of the packets flow via the accelerated path through the DLC layer, thus bypassing the IP layer in z/OS and reducing path length and improving performance.



	Outbound QDIO	Outbound iQDIO
Inbound QDIO	Yes	Yes
Inbound iQDIO	Yes	Yes

Accelerator requires IP Forwarding to be enabled for non-Sysplex Distributor acceleration.
No Acceleration for:
•IPv6
•Traffic which requires fragmentation in order to be forwarded
•Incoming fragments for a Sysplex Distributor connection
•OSA port interfaces using Optimized Latency Mode (OLM)



- Supports Sysplex Distributor (SD) IPCONFIG QDIOACCELERATOR
  - When traffic to target stack is sent over HiperSockets Dynamic XCF or QDIO as a result of VIPAROUTE definition.

MTU is Configured in HCD/IOCP

CHPID Parameter	MFS	MTU
CHPARM=00	16K	8K
CHPARM=40	24K	16K
CHPARM=80	40K	32K
CHPARM=C0	64K	56K

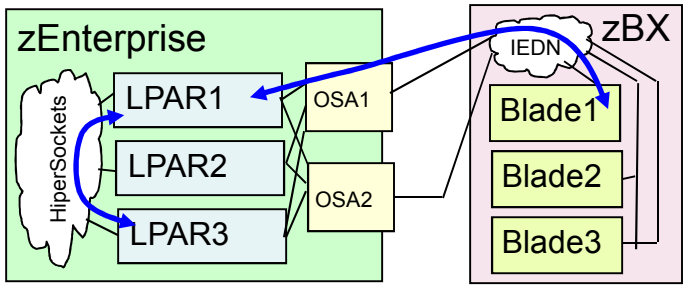
Default

"CHPARM" parameter was originally "OS" parameter.

On z196 and later processor the CHPID is also used to identify usage.
First character still indicates frame size:
0x, 4x, 8x, and Cx (as documented on the left)
The second character indicates usage:
x0 indicates Normal HiperSockets
x2 indicates HiperSockets for IEDN
x4 indicates HiperSockets for z/VM External Bridge
Where all "x" characters are wild cards.

- Each CHPID has configurable frame size (16K, 24K, 40K, 64K)
  - Allows optimization per HiperSockets LAN for small packets versus large streams
  - Affects MTU size of 8K, 16K, 32K, 56K
- HiperSockets LANs
  - Each HiperSockets LAN has its own CHPID, type IQD
    - IBM recommends starting from x"FF" and working your way backwards through the CHPID numbers, picking addresses from the high range to avoid addressing conflicts
    - May be shared by all defined LPARs
    - Delivered as object code only (OCO)
      - HiperSockets CHPIDs do not reside physically in the hardware but these CHPIDs cannot be used by other devices.
      - No physical media constraint, so no priority queuing or cabling required .
    - Each Operating System image configures its own usage of available HiperSockets CHPIDs.

HiperSockets Integration with OSA for IEDN



CHPARM=x2 See previous discussion in this presentation.

GlobalConfig AUTOIQDX ALLTRAFFIC
or GlobalConfig AUTOIQDX NOLARGEDATA
or GlobalConfig NOAUTOIQDX

Dynamically created TRLE is IUTIQXxx or IUTIQ6xx and dynamically created interface is EZAIQXxx or EZ6IQXxx where xx is the OSX CHPID.

- A single HiperSockets LAN may be defined such that it is automatically used when the destination is an LPAR on the same CEC belonging to the same OSX/HiperSockets LAN.
  - The OSA OSX devices are assigned IP Addresses.
  - The HiperSockets LAN (IQDX) is not assigned an IP Address.
  - Requires zEnterprise z196 or later processor
  - Requires z/OS V1.13
- Background
  - With VIPA and Dynamic Routing
    - The application may bind to the VIPA.
    - Dynamic routing causes traffic between LPARs to be routed over HiperSockets.
    - Dynamic routing causes traffic to remote partners (outside the CEC) to be routed over OSA.
  - Without VIPA and Dynamic Routing
    - It is a challenge to cause same CEC traffic to flow over HiperSockets at the same time that remote traffic flows over OSA.
    - Static Host routes may be used.
      - The application binds to the OSA IP address.
      - A static route is used on each LPAR such that when the other LPAR OSA address is the destination then the HiperSockets LAN is used to route the traffic.
      - With a large number of LPARs the administration of these static host routes is onerous.

HCD (IOCP)
Define 10 subchannel addresses for each IQDX CHPID that is in use for IPv4.
Define 10 subchannel addresses for each IQDX CHPID that is in use for IPv6.
Multiple VLAN does not affect the required number of subchannel addresses.

HiperSockets Features

HiperSockets Supported Features	z/OS	z/VM	Linux on System z	z/VSE
IPv4 Support	Yes	Yes	Yes	Yes
IPv6 Support	Yes	Yes	Yes	Yes
VLAN Support	Yes	Yes	Yes	Yes
Network Concentrator	No	No	Yes	No
Layer 2 Support	No	Yes	Yes	No
Multiple Write Facility	Yes	No	No	No
zIIP Assisted Multiple Write Facility	Yes	No	No	No
HiperSockets NTA (Network Traffic Analyzer)	No	No	Yes	No
Integration with IEDN (IQDX)	Yes	No*	Yes	No
Virtual Switch Bridge Support	No	Yes	No	No
Fast Path to Linux (LFP) Support / IUCV over HiperSockets	No	No	Yes	Yes
Completion Queue	Yes	No	Yes	Yes
* Depends upon the z/VM release				

Non-z/OS Support

- z/VM Virtual HiperSockets Support
  - In addition to CEC HiperSockets that are available to all LPARs, z/VM is able to support virtual HiperSockets available to all guests running on that z/VM image.
- z/VM HiperSockets Virtual Switch Bridge Support (also referred to as External Bridge)
  - A single HiperSockets LAN may be defined such that it is automatically used when the destination is an LPAR on the same CEC belonging to the same OSD/HiperSockets or OSX/HiperSockets Virtual Switch.
  - The OSA OSD or OSX devices are assigned IP Addresses.
  - The HiperSockets LAN is not assigned an IP Address.
- zLinux HiperSockets Network Concentrator
  - zLinux with HiperSockets and OSD devices is able to bridge traffic without routing overhead providing increased performance.
- zLinux HiperSockets Network Traffic Analyzer
  - Allows Linux on System z to control tracing of the internal virtual LAN.
  - Requires System z10 or later hardware.
  - Requires Linux for System z software.
- z/VM and zLinux Layer 2 Support
  - Both z/VM and zLinux support HiperSockets Layer 2 as well as Layer 3
  - z/OS only supports Layer 3
- z/VSE Fast Path to Linux (LFP) Support
  - Allows communications of z/VSE TCP/IP applications to Linux without a TCP/IP stack on z/VSE.
  - Requires:
    - z196 or later
    - z/VSE V5.1.1
    - LFP in an LPAR
    - HiperSockets Completion Queue

OSA vs. HiperSockets vs. RoCE

Cross CEC, OS, and Firewall

- OSA, HiperSockets, and RoCE are all used for data transfer between hosts.
- Cross CEC traffic
  - OSA and RoCE can be used to send traffic between CECs.
  - HiperSockets only supports traffic between LPARs on a single CEC.
- Different Operating Systems
  - OSA and HiperSockets are supported by multiple Operating Systems (ie. z/OS, z/VM, Linux on System z, etc.)
  - RoCE is only supported by z/OS.
- Traffic outside of a single CEC (and might therefore require additional security measures)
  - OSA traffic can go over the card if shared between LPARs on a single CEC.
  - OSA traffic goes over a network if used to a non-shared-OSA partner.
  - RoCE traffic goes over a 10GbE Layer 2 LAN.
    - Security exposure is limited if contained in a secure location.
  - HiperSockets traffic never goes outside a single CEC.
- Firewall with stateful packet inspection (a PCI (Payment Card Industry) requirement for some traffic)
  - OSA traffic can be sent over a LAN to a Firewall with stateful packet inspection support.
  - HiperSockets traffic cannot be sent over a Firewall with stateful packet inspection support (unless routed traffic).
  - RoCE traffic cannot be sent over a Firewall with stateful packet inspection support (does not support routed traffic).

Protocol, HW, and Overhead

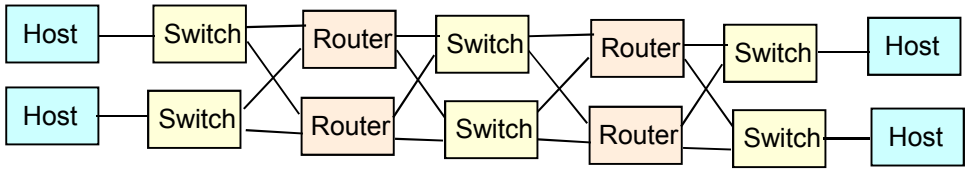
- Protocol Support
  - OSA supports all TCP/IP protocols and even supports SNA protocol (natively in OSE mode, or when sent with UDP (Enterprise Extender (EE))).
  - HiperSockets only supports IP traffic, it does not support native SNA (so EE must be used to send SNA).
  - RoCE only supports TCP traffic (except IPsec), it does not support native SNA or UDP (EE).
- Required hardware feature
  - OSA feature is required.
  - HiperSockets is part of z System Firmware so it does not require any additional hardware / adapter card purchase.
  - RoCE feature is required. A minimum of 2 per LPAR is recommended.
- CP Overhead
  - OSA provides many different types of offload to the adapter that reduces CP overhead (ARP, Check Sum, Segmentation, etc.)
  - HiperSockets supports zIIP offload to reduce associated cost.
  - RoCE reduces TCP/IP overhead by using RDMA protocol.
- Storage Usage
  - OSA and HiperSockets use CSM fixed storage backed by 64-bit real for data buffers and use Hardware System Area (HAS) memory for routing table.
  - RoCE uses pinned Fixed Memory (mostly 64-bit Common), not CSM managed memory. The maximum amount of memory available for SMC-R with RoCE is definable. When the maximum is reached, new connections will not be SMC-R RoCE eligible.

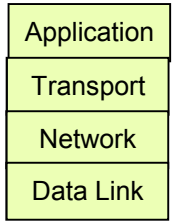
Routing and VLAN

- IP Routing
  - OSA requires IP routing. OSA does provide dynamic backup with multiple OSAs to the same subnet and Multipath. When multiple OSAs are attached to different subnets or OSA and other attachements, like HiperSockets, are used there is no dynamic backup without a Dynamic Routing protocol (ie. OSPF).
  - HiperSockets requires IP routing so there is no dynamic backup without a Dynamic Routing protocol.
    - When HiperSockets is defined as part of a DynamicXCF network routing is handled automatically.
    - When HiperSockets is defined with Integration with IEDN (IQDX) routing is handled automatically.
    - Backup might not be a requirement because if HiperSockets fails there is probably major CEC problems occurring.
  - Traffic is automatically / transparently switched from OSA to RoCE. If RoCE connection is not available traffic is sent over OSA. RoCE has automatic / transparent backup to a different RoCE path if one exists.
    - Active RoCE sessions are dropped if the RoCE connection fails and there is no alternate RoCE path, but new sessions will flow using OSA.
- IP Routed Traffic
  - OSA supports routed traffic. Routed traffic is optimized on z/OS with QDIO Accelerator.
  - HiperSockets supports routed traffic (ie. traffic may come in over OSA and then be routed over HiperSockets). Routed traffic is optimized on z/OS with QDIO Accelerator.
  - RoCE does not support routed traffic. All connections are over a Layer 2 network.
- VLAN Support
  - OSA supports VLAN tagging.
  - HiperSockets supports VLAN tagging.
  - RoCE inherits VLAN IDs from all associated OSAs and supports VLAN tagging.

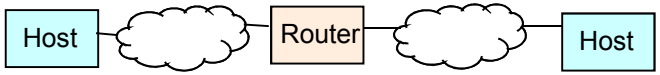
Routing Basics

Physical and Logical Network



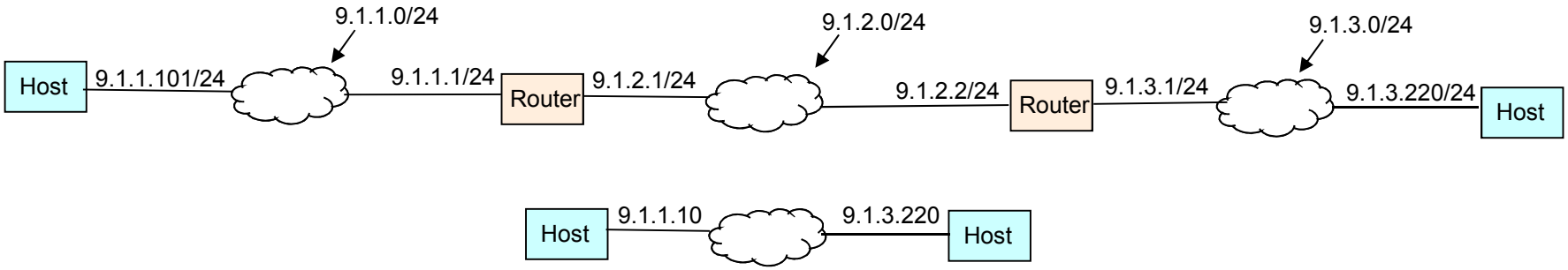


- There is some physical network.
- It can be shared between TCP/IP and SNA, and even other protocols concurrently.
- Devices are connected with cables.
- Each device network connection has a Media Access Control (MAC) address that is used for sending and receiving messages.



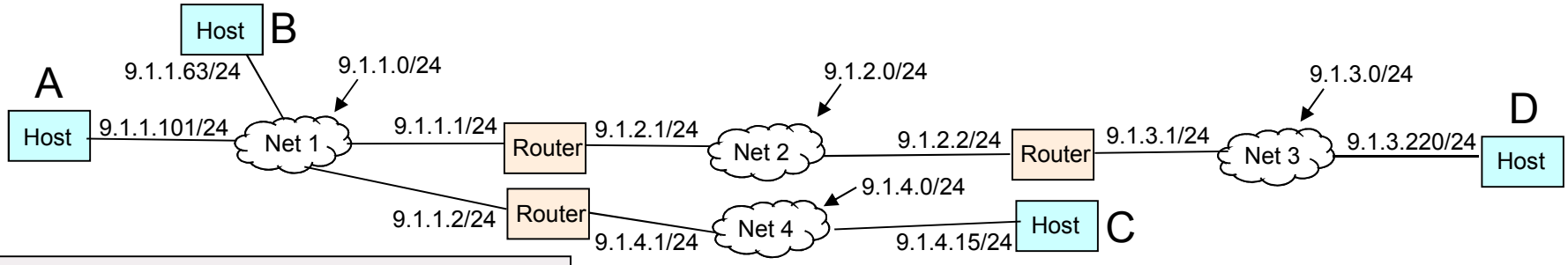
- There are logical network views that use the underlying physical network.
  - The level of detail depends upon the discussion.
- Each device network connection has a Logical address that is used for sending and receiving messages.

TCP/IP Network



- IPv4 Address has 32 bits 1111 1111 1111 1111 1111 1111 1111 1111
- Hex calculation 9.1.1.101 = 0000 1001 0000 0001 0000 0001 0110 0101
- Subnet Mask 24 indicates that the first 24 bits identify the subnetwork and only the last 8 bits identify the host address.
- 24 = 1111 1111 1111 1111 1111 1111 0000 0000
  - Also referred to as 255.255.255.0

TCP/IP Routing



Host A:
Direct Route to 9.1.1.0/24
Indirect Route to 9.1.4.0/24 via 9.1.1.2
Host Route to 9.1.4.15 via 9.1.1.2
Default Route to 9.1.1.1

```
ROUTE 9.1.1.0/24  =  OSA472 MTU 1492
ROUTE 9.1.4.0/24 9.1.1.2 OSA472 MTU 1492
ROUTE 9.1.4.15 HOST 9.1.1.2 OSA472 MTU 1492
ROUTE DEFAULT  9.1.1.1 OSA472 MTU 1492
```

- Direct Routes
  - Subnetworks that this host connects to.
- Indirect Routes
  - Subnetworks that this host can reach by routing through a router.
  - Host route is a special type of Indirect Route (see next page).
  - When the destination is not on a directly attached subnet but a host on a directly attached subnet does have a route to the destination.
- Default Routes
  - Where to send messages when there is no explicit (direct or indirect) route.

z/OS Routing Table

Subnet mask of 32 indicates Host Routes, the most specific type of route possible. More specific routes always take precedence over less specific routes.
Subnet or Network routes, in this case subnet route with mask of 24, are less specific than host routes but more specific than default routes.
Default routes are the least specific routes possible and are therefore only used when no other route matches the destination.

Host A:
Direct Route to 9.1.1.0/24
Indirect Route to 9.1.4.0/24 via 9.1.1.2
Host Route to 9.1.4.15 via 9.1.1.2
Default Route to 9.1.1.1

```
ROUTE 9.1.1.0/24  =  OSA472 MTU 1492
ROUTE 9.1.4.0/24 9.1.1.2 OSA472 MTU 1492
ROUTE 9.1.4.15 HOST 9.1.1.2 OSA472 MTU 1492
ROUTE DEFAULT  9.1.1.1 OSA472 MTU 1492
```

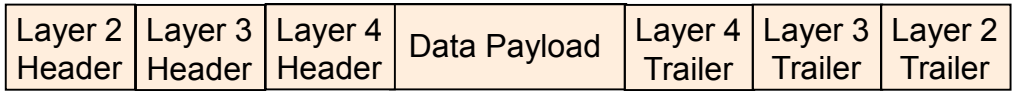
- The destination IP Address, IP subnet, or IP network is in the first column above.
  - DEFAULT is a special destination keyword indicating where to send packets when the destination does not match any other route in the table.
- Gateway IP Address is in the second column above. An equal sign, =, indicates there is no next hop gateway.
  - A route without a next hop gateway is known as a direct route.
  - A route with a next hop gateway is known as an indirect route.
- The interface name or link name is in the third column above to indicate which interface the packet is to be sent over.
- Each route in the table may be read as “If this destination, then send packet to this gateway over this interface”.

Sending and Receiving Data

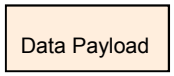
Source and Destination IP Addresses

- TCP/IP Application Session
  - One partner initiates the connection and sends a connection request packet. For TCP connections the host that sends the connection request is the client.
    - What destination IP address to connect to?
      - *If hostname is given then IP address is resolved by DNS or Local Host Name file (IPNodes).*
    - What source IP address to use in the initiate packet?
      - *Hierarchy of source IP address is detailed in z/OS Communication Server (CS) IP Configuration Guide, SC31-8775.*
        - » *Sendmsg( ) using the IPV6_PKTINFO ancillary option specifying a nonzero source address (RAW and UDP sockets only)*
        - » *Setsockopt( ) IPV6_PKTINFO option specifying a nonzero source address (RAW and UDP sockets only)*
        - » *Explicit bind to a specific local IP address*
        - » *bind2addrsel socket function (AF_INET6 sockets only)*
        - » *PORT profile statement with the BIND parameter*
        - » *SRCIP profile statement (TCP connections only)*
        - » *TCPSTACKSOURCEVIPA parameter on the IPCONFIG or IPCONFIG6 profile statement (TCP connections only)*
        - » *SOURCEVIPA: Static VIPA address from the HOME list or from the SOURCEVIPAINTERFACE parameter*
        - » *HOME IP address of the link over which the packet is sent*
  - The other partner receives the connection request packet and responds. For TCP connections the host that receives the connection request is the server.
    - The host that receives the connection request takes the source IP address from the connection packet and uses that for the destination IP address in the packet that it sends back.
    - The host that receives the connection request takes the destination IP address from the connection packet and uses that for the source IP address in the packet that it sends back.
    - These source and destination assignments are usually used for the life of the connection.

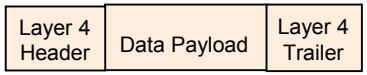
Sending a Packet



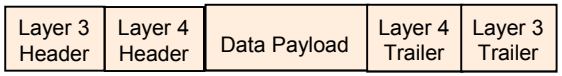
- Application Layer 5 sends some data



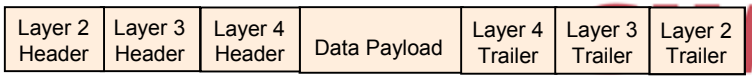
- TCP/IP stack Transport Layer 4
  - It puts some header information in front of the data.
  - It might put some trailer information after the data.

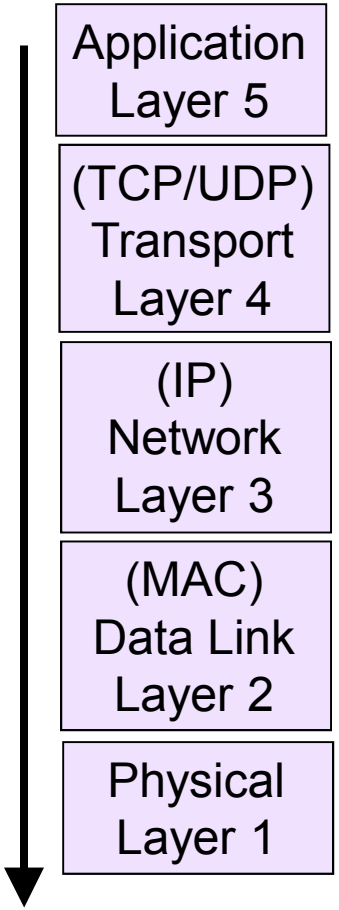


- TCP/IP stack Network Layer 3
  - It puts some header information in front of the data.
  - It might put some trailer information after the data.

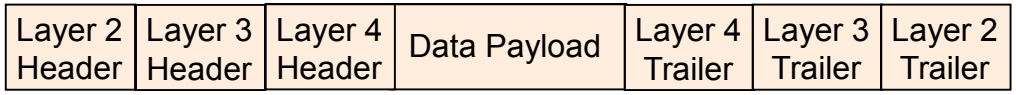


- OSA Transport Layer 2
  - It puts some header information in front of the data.
  - It might put some trailer information after the data.

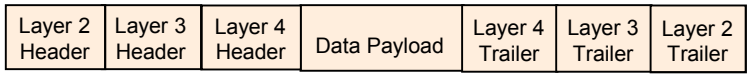




Receiving a Packet

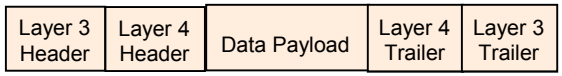


- OSA Transport Layer 2



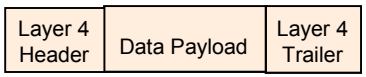
  - It strips off header and option trailer and passes packet to TCP/IP Stack Layer 3.

- TCP/IP stack Network Layer 3



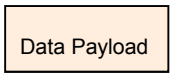
  - It strips off header and option trailer and passes packet to TCP/IP Stack Layer 4.

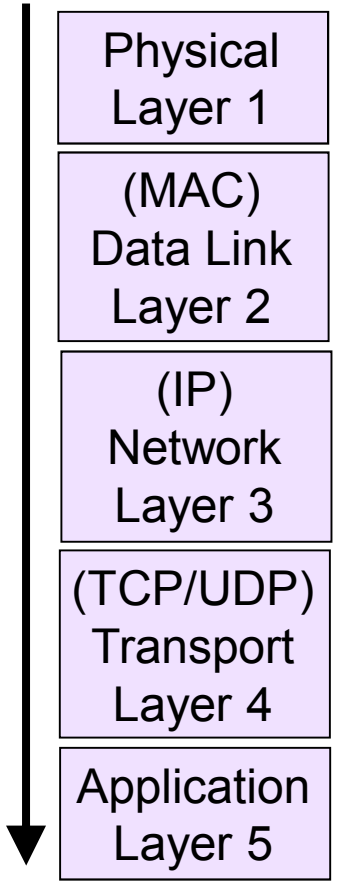
- TCP/IP stack Transport Layer 4



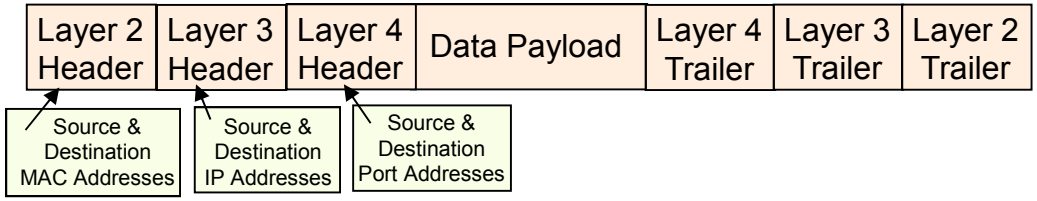
  - It strips off header and option trailer and passes data to Application Layer 5.

- Application Layer 5 sends some data

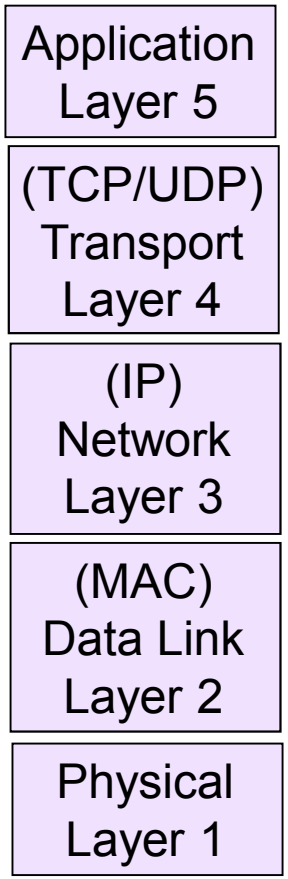




What is Layer 2

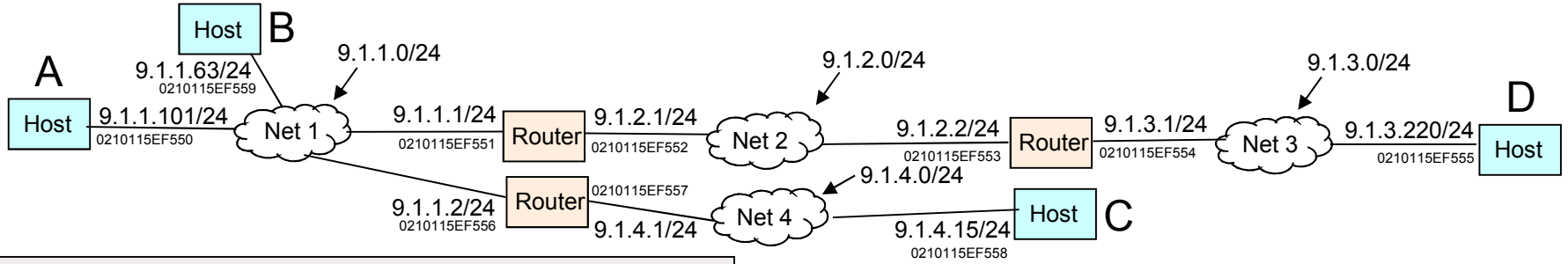


- z/OS Operating System only supports Layer 3 protocol transmission. The data sent/received is either TCP/IP data (OSD or OSE) or SNA (OSA OSE). The Layer 2 Header and Trailer have been stripped off the packet before it is passed to z/OS.
- Linux on System z supports Layer 2 protocol transmission. Data received by OSA is passed to Linux with the Layer 2 Header and Trailer in place.
  - Layer 2 is also referred to as protocol agnostic since the Layer 3 protocol type does not matter (it could be IP, SNA, Apple Talk, anything).
- Can z/OS communicate to Linux even though z/OS is using Layer 3 LAN attachment and Linux is using Layer 2 attachment? YES
- A Layer 2 LAN can refer to a LAN between devices that does not have an IP router (also referred to as a Layer 3 router) in the communication path, all devices are in the same IP subnet.
  - z/OS can attach to a Layer 2 LAN.



ARP & IP Routing

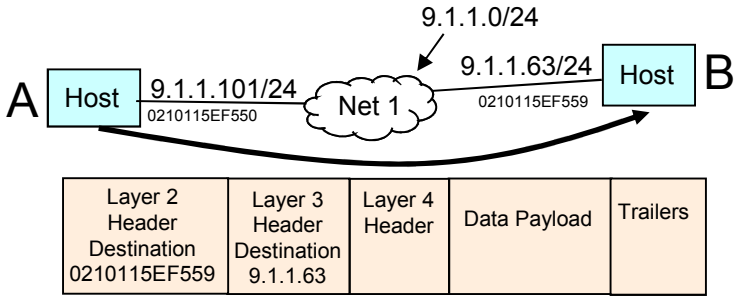
Address Resolution Protocol (ARP)



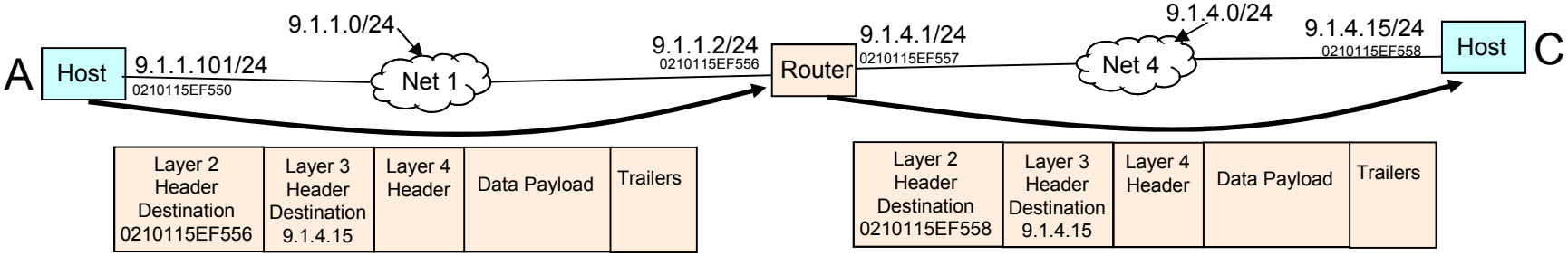

```
ROUTE 9.1.1.0/24  =  OSA472 MTU 1492
ROUTE 9.1.4.0/24 9.1.1.2 OSA472 MTU 1492
ROUTE 9.1.4.15 HOST 9.1.1.2 OSA472 MTU 1492
ROUTE DEFAULT  9.1.1.1 OSA472 MTU 1492
```

- Address Resolution Protocol (ARP) is used to discover the MAC address associated with an IP address.
  - Source and Destination MAC address are in the Layer 2 Header of the packet.
  - Source and Destination IP address are in the Layer 3 Header of the packet.
- For Host A to send data to Host B, 9.1.1.63, on the subnet that it is attached to (Direct Route) it uses ARP to learn the MAC address, 0210115EF559. The Layer 2 header is created with the destination 0210115EF559.
- For Host A to send data to Host C, 9.1.4.15, on remote subnet Net 4, the IP routing table indicates to send the packet to 9.1.4.15 (Indirect Route). ARP is used to learn the MAC address, 0210115EF557. It is put in the Layer 2 header as the destination.
- For Host A to send data to Host D, 9.1.3.220, on remote subnet Net 3, the IP routing table indicates to send the packet to 9.1.1.1 (Default Route). ARP is used to learn the MAC address, 0210115EF551. It is put in the Layer 2 header as the destination.

Direct and Indirect Routes

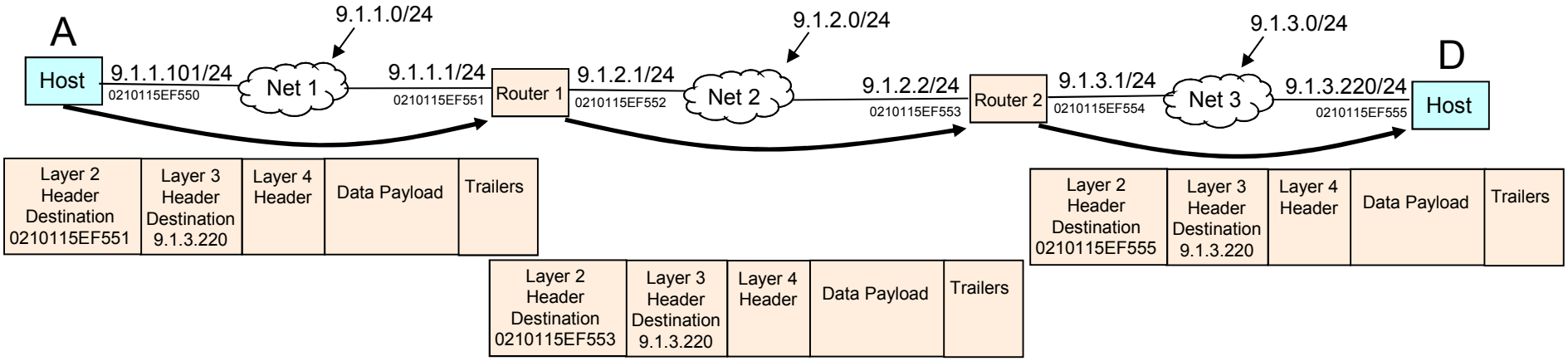


- For Host A to send data to Host B, 9.1.1.63, on the subnet that it is attached to (Direct Route) it uses ARP to learn the MAC address, 0210115EF559. The Layer 2 header is created with the destination 0210115EF559.



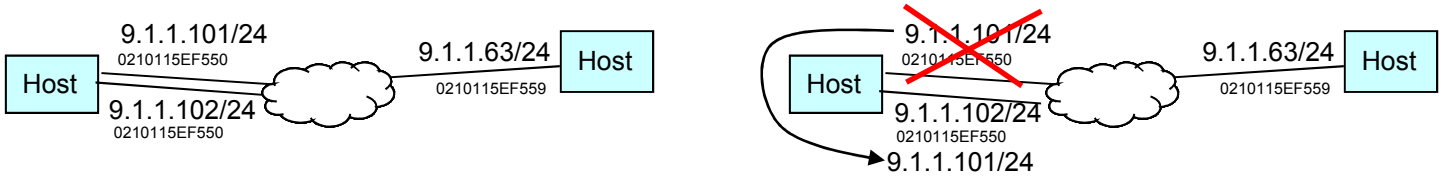
- For Host A to send data to Host C, 9.1.4.15, on remote subnet Net 4, the IP routing table indicates to send the packet to 9.1.4.15 (Indirect Route). ARP is used to learn the MAC address, 0210115EF557. It is put in the Layer 2 header as the destination.
- When the Router receives the packet it strips off the Layer 2 header, at Layer 3 the Router IP routing table determines the packet destination 9.1.4.15 is on the direct attached network. ARP is used to learn the MAC address, 0210115EF558. It is put in the new Layer 2 header as the destination.

Default Route

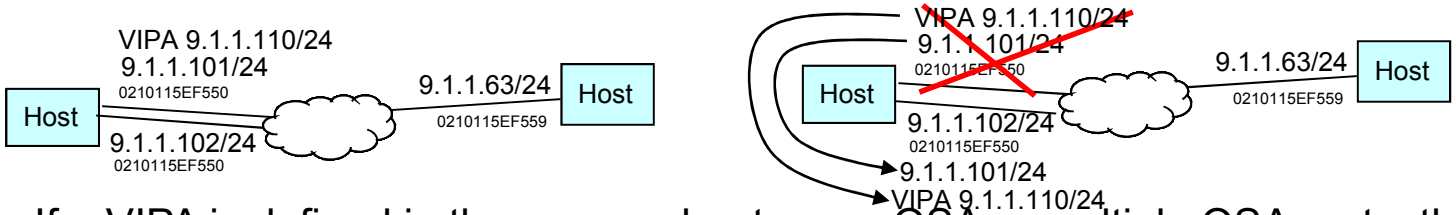


- For Host A to send data to Host D, 9.1.3.220, on remote subnet Net 3, the IP routing table indicates to send the packet to 9.1.1.1 (Default Route). ARP is used to learn the MAC address, 0210115EF551. It is put in the Layer 2 header as the destination.
- When the Router1 receives the packet it strips off the Layer 2 header, at Layer 3 the Router IP routing table determines the packet destination 9.1.3.220 should be sent to 9.1.2.2. ARP is used to learn the MAC address, 0210115EF553. It is put in the new Layer 2 header as the destination.
- When the Router2 receives the packet it strips off the Layer 2 header, at Layer 3 the Router IP routing table determines the packet destination 9.1.3.220 is on the direct attached network. ARP is used to learn the MAC address, 0210115EF555. It is put in the new Layer 2 header as the destination.

Gratuitous ARP Failover

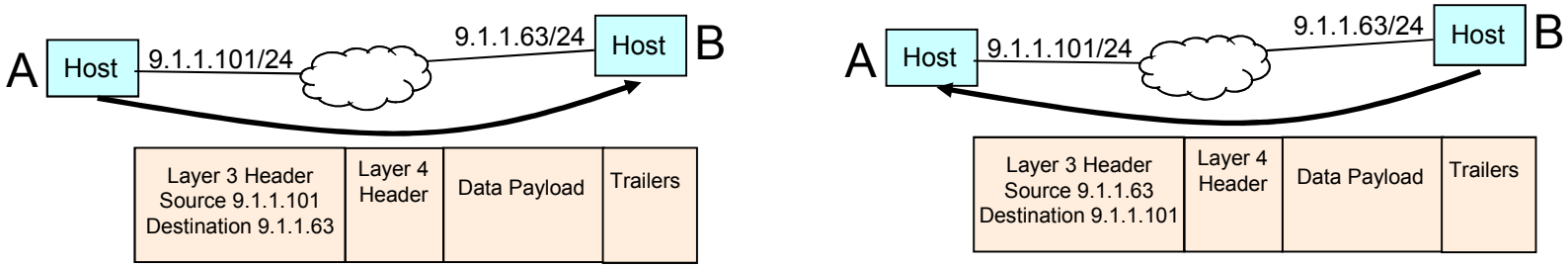


- z/OS TCP/IP stack detects when multiple OSA ports activate to the same subnet. If one OSA fails the working OSA sends out a Gratuitous ARP (unsolicited ARP message) so that the failed OSA IP Address is associated with the working OSA MAC Address. In that case a single OSA owns two IP addresses associated with the single MAC Address. If the failed OSA recovers the IP Addresses moves back.
  - Dynamic Backup between OSA without Dynamic Routing (OSPF)



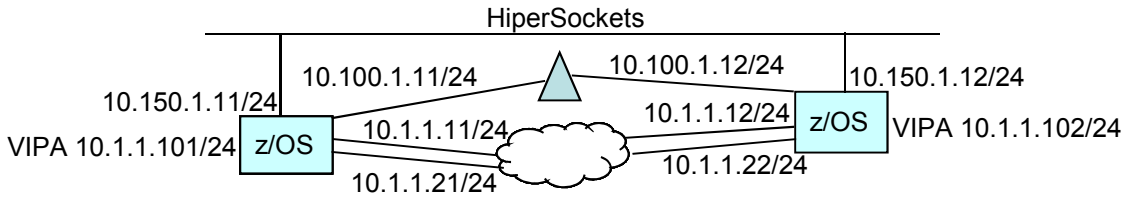
- If a VIPA is defined in the same subnet as an OSA or multiple OSA ports, the first OSA that activates sends out a Gratuitous ARP so that the VIPA is associated with it. The OSA owns two IP addresses associated with the single MAC Address. If the OSA fails, both the OSA IP address and VIPA move to another OSA. Now all three IP addresses are associated with a single MAC Address. If the original OSA recovers the OSA port's IP address moves back but the VIPA remains on the second OSA port.

Packet Source and Destination IP Addr



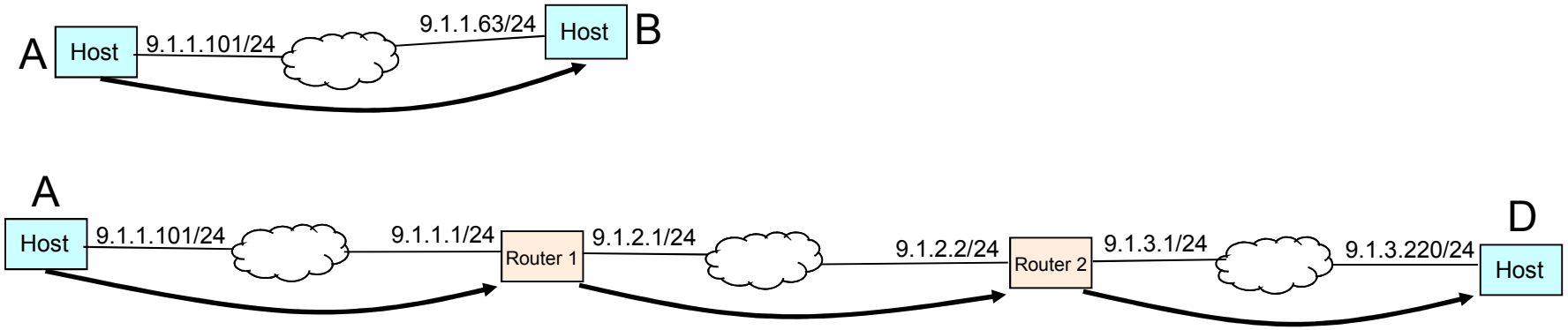
- The Source and Destination IP Addresses in the initial packet of a TCP session are determined as detailed on a previous page.
- Typically the server reverses those addresses for the response. The Source in the initial packet is the Destination in the response packet. The Destination in the initial packet is the Source in the response packet.
- Typically those IP Addresses remain the same for the duration of the connection.

Multiple Network Connections



- A Workstation, Laptop, and Linux on System z typically only have a single network connection defined to them. z/OS usually has multiple network connections:
  - Multiple OSA ports (10.1.1.0/24)
  - VIPA (10.1.1.0/24)
  - DynamicXCF (10.100.1.0/24)
  - HiperSockets (10.150.1.0/24)
- All networks (physical and virtual) should be a unique subnet.
  - The only exception to this rule is VIPA
    - When the VIPA address is in the same subnet as OSA.
    - When multiple z/OS systems have VIPAs they can all be in the same subnet.

Routing Overhead



- There is an overhead associated with routing. All things being equal, performance is typically better between two hosts on the same subnet (sometimes referred to as a Layer 2 connection (or Flat network)), rather than two hosts on different subnets.
  - One of the benefits of the IBM zEnterprise zBX.

HiperSockets Between z/OS and Linux

Routing Goals

1. Dynamic Backup/Failover for multiple OSA ports.
2. Dynamic Backup/Failover between different LAN types, ie. between HiperSockets and OSA.
3. Ability to have a server Bound to a single IP Address concurrently reachable by HiperSockets for same CEC partners and by OSA for non-same CEC partners.

Solutions

1. Dynamic Backup/Failover for multiple OSA ports.
   - z/OS uses VIPA and either OSA Gratuitous ARP support or Dynamic Routing
   - Linux uses VSWITCH
2. Dynamic Backup/Failover between different LAN types, ie. between HiperSockets and OSA.
   - z/OS uses VIPA and Dynamic Routing
   - Linux doesn't have this capability. But keep in mind that a HiperSockets LAN is virtual running in zEnterprise microcode and if there is a problem with your HiperSockets LAN you have a whole CEC problem so the HiperSockets LAN is the least of your worries.
3. Ability to have a server Bound to a single IP Address concurrently reachable by HiperSockets for same CEC partners and by OSA for non-same CEC partners.
   - z/OS uses VIPA and Dynamic Routing
   - Linux can use DNS or Host Routes

More Information

Web Pages

- URLs for Publications
  - http://www.ibm.com/systems/z/os/zos/bkserv/index.html
  - http://www.redbooks.ibm.com
- URLs for z/OS Communications Server and GUI Download
  - http://www.ibm.com/software/network/commserver/zos/support/
  - http://www.ibm.com/support/docview.wss?rs=852&context=SSSN3L&dc=DB530&dc=D430&dc=D410&dc=D420&dc=DB510&dc=DB550&q1=Configuration+Assistant&uid=swg24013160&loc=en_US&cs=utf-8&lang=en
  - or better:
    - http://www.ibm.com/software/network/commserver/zos/support/
    - then select "IBM Configuration Assistant for z/OS Communications Server" in "Download" section of the page
- Main Security Web Pages:
  - https://www.pcisecuritystandards.org/
  - http://www.iss.net/
  - http://www.ibm.com/servers/eserver/zseries/zos/security
  - http://www.ibm.com/systems/z/security/
- IBM Mainframe Servers
  - http://www.ibm.com/servers/eserver/zseries
- IBM zEnterprise Servers Network Technologies
  - http://www.ibm.com/servers/eserver/zseries/networking/technology.html
- z/OS Communications Server
  - http://www.ibm.com/software/network/commserver/zos/

Web Pages (cont.)

- Communications Server for Linux on System z
  - http://www.ibm.com/software/network/commserver
  - http://www.ibm.com/software/network/commserver/library
- Communication Controller for Linux on System z
  - http://www.ibm.com/software/network/ccl
- PKI Services web site:
  - http://www.ibm.com/servers/eserver/zseries/zos/pki
- PKI Services Red Book:
  - http://www.redbooks.ibm.com/abstracts/sg246968.html
- ITSO Redbooks
  - http://www.redbooks.ibm.com
- RACF web site:
  - http://www.ibm.com/servers/eserver/zseries/zos/racf
- IBM Washington Systems Center Technical Sales Support
  - http://www.ibm.com/support/techdocs/
- Request for Comment (RFC)
  - http://www.rfc-editor.org/rfcsearch.html
  - http://www.rfc-editor.org/
- Online Courses (search for cryptography)
  - http://coursera.org

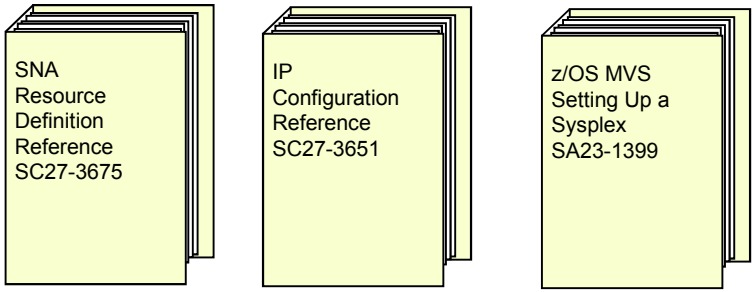
IBM Manuals

- IBM z/OS and z/OS Communications Server Manuals
  - z/OS Communications Server IP Configuration Guide (z/OS V1.13 SC31-8775, z/OS V2.1+ SC27-3650)
  - z/OS Communications Server IP Configuration Reference (z/OS V1.13 SC31-8776, z/OS V2.1+ SC27-3651)
  - z/OS IP Diagnosis Guide (z/OS V1.13 GC31-8782, z/OS V2.1+ GC27-3652)
  - z/OS IP System Administrator Commands (z/OS V1.13 SC31-8781, z/OS V2.1+ SC27-3661)
  - z/OS Four Volumes of IP Messages (z/OS V1.13 SC31-8783, SC31-8784, SC31-8785, SC31-8786, z/OS V2.1+ SC27-3654, SC27-3655, SC27-3656, SC27-3657)
  - z/OS Migration Manual (z/OS V1.13 GA22-7499, z/OS V2.1+ GA32-0889)
  - z/OS System SSL Programming Guide (z/OS V1.13 SC24-5901, z/OS V2.1+ SC14-7495)
  - z/OS Integrated Cryptographic Services (ICSF) System Programmer Guide (z/OS V1.13 SA22-7520, z/OS V2.1+ SC14-7507)
  - z/OS Cryptographic Services PKI Services Guide and Reference (z/OS V1.13 SA22-7693, z/OS V2.1+ SA23-2286)
  - z/OS Security Server RACF Security Administrator's Guide (z/OS V1.13 SA22-7683, z/OS V2.1+ SA23-2289)
  - z/OS Security Server RACF Command Language Reference (z/OS V1.13 SA22-7687, z/OS V2.1+ SA23-2292)
  - z/OS UNIX System Services Planning (z/OS V1.13 GA22-7800, z/OS V2.1+ GA32-0884)
  - z/OS UNIX System Services User's Guide (z/OS V1.13 SA22-7801, z/OS V2.1+ SA23-2279)
  - z/OS UNIX System Services Command Reference (z/OS V1.13 SA22-7802)
- RACF Command Samples for TCP/IP on z/OS
  - SYS1.TCPIP.SEZAINST(EZARACF)

IBM Manuals (cont.)

- IBM RedBooks
  - Communications Server for z/OS V1R11 TCP/IP Implementation
    - Volume I: Base Functions, Connectivity, and Routing (SG24-7798)
    - Volume II: Standard Applications (SG24-7799)
    - Volume III: High Availability, Scalability, and Performance (SG24-7800)
    - Volume IV: Security and Policy-based Networking (SG24-7801)
  - Communications Server for z/OS V1R12 TCP/IP Implementation
    - Volume I: Base Functions, Connectivity, and Routing (SG24-7896)
    - Volume II: Standard Applications (SG24-7897)
    - Volume III: High Availability, Scalability, and Performance (SG24-7898)
    - Volume IV: Security and Policy-based Networking (SG24-7899)
  - Communications Server for z/OS V1R13 TCP/IP Implementation
    - Volume I: Base Functions, Connectivity, and Routing (SG24-7996)
    - Volume II: Standard Applications (SG24-7997)
    - Volume III: High Availability, Scalability, and Performance (SG24-7998)
    - Volume IV: Security and Policy-based Networking (SG24-7999)
  - Communications Server for z/OS V2R1 TCP/IP Implementation
    - Volume I: Base Functions, Connectivity, and Routing (SG24-8096)
    - Volume II: Standard Applications (SG24-8097)
    - Volume III: High Availability, Scalability, and Performance (SG24-8098)
    - Volume IV: Security and Policy-based Networking (SG24-8099)

Appendix:
More Configuration Details and Commands



Sample IOCP for HiperSockets

```
************************************************************
* CHPARM values are '00'=16K,'40'=24K,'80'=40K and 'C0'=64K.  *
*                                                          *
* Need at least 3 addresses per z/OS, maximum of 10:       *
*   - 2 addresses for control                              *
*   - 1 address for data for each TCP stack (between 1 and 8) *
************************************************************
CHPID PATH=(FA),SHARED,                                    *
      PARTITION=((LPAR1,LPAR2,LPAR3),(LPAR1,LPAR2,LPAR3)),    *
      TYPE=IQD,CHPARM=00
CNTLUNIT CUNUMBR=FD00,PATH=(FA),UNIT=IQD
IODEVICE ADDRESS=(FD00,010),CUNUMBR=(FD00),UNIT=IQD
CHPID PATH=(FB),SHARED,                                    *
      PARTITION=((LPAR1,LPAR2,LPAR3),(LPAR1,LPAR2,LPAR3)),    *
      TYPE=IQD,CHPARM=40
CNTLUNIT CUNUMBR=FD10,PATH=(FB),UNIT=IQD
IODEVICE ADDRESS=(FD10,010),CUNUMBR=(FD10),UNIT=IQD
CHPID PATH=(FC),SHARED,                                    *
      PARTITION=((LPAR1,LPAR2,LPAR3),(LPAR1,LPAR2,LPAR3)),    *
      TYPE=IQD,CHPARM=80
CNTLUNIT CUNUMBR=FD20,PATH=(FC),UNIT=IQD
IODEVICE ADDRESS=(FD20,010),CUNUMBR=(FD20),UNIT=IQD
CHPID PATH=(FD),SHARED,                                    *
      PARTITION=((LPAR1,LPAR2,LPAR3),(LPAR1,LPAR2,LPAR3)),    *
      TYPE=IQD,CHPARM=C0
CNTLUNIT CUNUMBR=FD30,PATH=(FD),UNIT=IQD
IODEVICE ADDRESS=(FD30,010),CUNUMBR=(FD30),UNIT=IQD
```

CHPID	MFS	MTU
CHPARM=00	16K	8K
CHPARM=40	24K	16K
CHPARM=80	40K	32K
CHPARM=C0	64K	56K

- Hardware Configuration Definition (HCD) or I/O Configuration Program (IOCP) must be used to create an IOCDS with the HiperSockets CHPID and subchannel (I/O device) definitions. Because HiperSockets are shared among LPARs, the CHPIDs must be defined as shared in the hardware definitions. A minimum of three subchannel addresses must be configured. One read control device, one write control device, and one data device. Each TCP/IP stack (max of 8) on a single LPAR requires a single data device. The read device must be an even number. The write device must be the read device number plus 1. The data devices can be any device numbers; it does not need to be the next sequential number after the read and write device numbers.
- The Maximum Frame Size (MFS) to be used on a HiperSockets network is set by specifying the “CHPARM=" parameter.
- TCP/IP coding for Maximum Transmission Unit (MTU), will need to correspond to this setting.

QDIO/iQDIO Read Storage

- Amount of storage for read processing:

OSA QDIO	4 Meg
HiperSockets MFS=16K	2 Meg
HiperSockets MFS=24K	3 Meg
HiperSockets MFS=40K	5 Meg
HiperSockets MFS=64K	8 Meg

Configurable value via VTAM Start Option or Link/Interface keyword (see OSA doc)

Configurable value via VTAM Start Option or Link/Interface keyword (see next pages)

- The storage used for read processing is allocated from the CSM data space 4K pool, and is fixed storage backed by 64-bit real. (CSM fixed storage defined in PARMLIB member IVTPRMxx)
- OSA QDIO
  - 64 SBALs (storage block address lists) x 64K = 4M
- HiperSockets
  - 126 SBALs x 16K = 2M
  - 126 SBALs x 24K = 3M
  - 126 SBALs x 40K = 5M
  - 126 SBALs x 64K = 8M

VTAM Start Options to Define Storage

- OSA QDIO Read Storage VTAM Start Option QDIOSTG
  - Defines how much storage VTAM keeps available for read processing for all OSA QDIO devices

```
        +--QDIOSTG=-MAX---------+
>>------+-----------------------+--->
        +--QDIOSTG=--+--MAX--+--+
                     +--AVG--+
                     +--MIN--+
                     +--nnn--+
```

MAX	64 SBALs x 64K = 4M
AVG	32 SBALs x 64K = 2M
MIN	16 SBALs x 64K = 1M

Default

- HiperSockets Read Storage VTAM Start Option IQDIOSTG
  - Defines how much storage VTAM keeps available for read processing for all HiperSockets devices that use an MFS of 64K

```
        +--IQDIOSTG=-MAX---------+
>>------+-----------------------+--->
        +--IQDIOSTG=--+--MAX--+--+
                      +--AVG--+
                      +--MIN--+
                      +--nnn--+
```

MAX	126 SBALs x 64K = 8M
AVG	96 SBALs x 64K = 6M
MIN	64 SBALs x 64K = 4M

Default

- Storage units are defined in terms of QDIO SBALs (QDIO read buffers)
  - nnn is the exact number of SBALs in the range 8-126
  - MAX allows for the best performance (for example, throughput), but requires more storage.
  - MIN may be used for devices with lighter workloads or where system storage might be constrained.
  - The amount of storage used is times the number of active QDIO data devices.
- Start Option defaults are appropriate for most environments
  - Review CSM specifications in PARMLIB member IVTPRMxx and increase, if appropriate
  - Use the D NET,CSM to display CSM usage
  - Modify storage settings using Start Options, as appropriate
  - Use VTAM tuning stats to evaluate needs and usage. Under a typical workload, the NOREADS counter should remain low (close to 0). If this count does not remain low you may need to consider a higher setting for QDIOSTG/IQDIOSTG.
  - RMF records send failures can be an indication that the HiperSockets target LP (logical partition) does not have enough storage.

Interface/Link Keyword to Define Storage

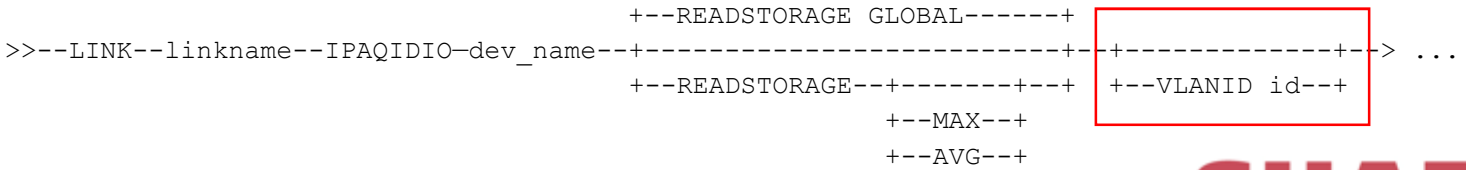
```
                                                  +--READSTORAGE GLOBAL------+
>>--INTERFACE—intf_name—DEFINE IPAQIDIO—CHPID-chpid--…--+--------------------------+--+-------------+--> ...
                                                  +--READSTORAGE--+-------+--+  +--VLANID id--+
                                                                  +--MAX--+
                                                                  +--AVG--+
                                                                  +--MIN--+

                                                 +--READSTORAGE GLOBAL------+
>>--LINK--linkname--IPAQIDIO—dev_name--+--------------------------+--+-------------+--> ...
                                       +--READSTORAGE--+-------+--+  +--VLANID id--+
                                                       +--MAX--+
                                                       +--AVG--+
                                                       +--MIN--+
```

- Keyword READSTORAGE on LINK and INTERFACE
  - Link statement for IPAQENET (IPv4 OSA Eth), IPAQTR (IPv4 OSA TR), and IPAQIDIO (IPv4 HiperSockets)
  - Interface statement for IPAQENET6 (IPv6 OSA), and IPAQIDIO6 (IPv6 HiperSockets)
- Override VTAM Start option QDIOSTG or IQDIOSTG for a specific QDIO or iQDIO device.
- Global causes the QDIOSTG or IQDIOSTG VTAM start option values to be used.
  - This is the default.
- MAX, AVG, and MIN
  - Causes the MAX, AVG, or MIN VTAM Start option MAX, AVG, or MIN values to be used.

HiperSockets VLAN ID Summary

- HiperSockets LAN may be divided into separate Virtual LANs (VLANs)
  - DynamicXCF HiperSockets VLANs are required for different TCP/IP stacks in an LPAR to belong to different TCP/IP Subplexes.
    - TCP/IP Profile GLOBALCONFIG IQDVLANid nnnn
  - Manual HiperSockets VLANs may be defined with VLANID on LINK and INTERFACE to divide a HiperSockets LAN into multiple subsets just as OSA QDIO VLAN support.



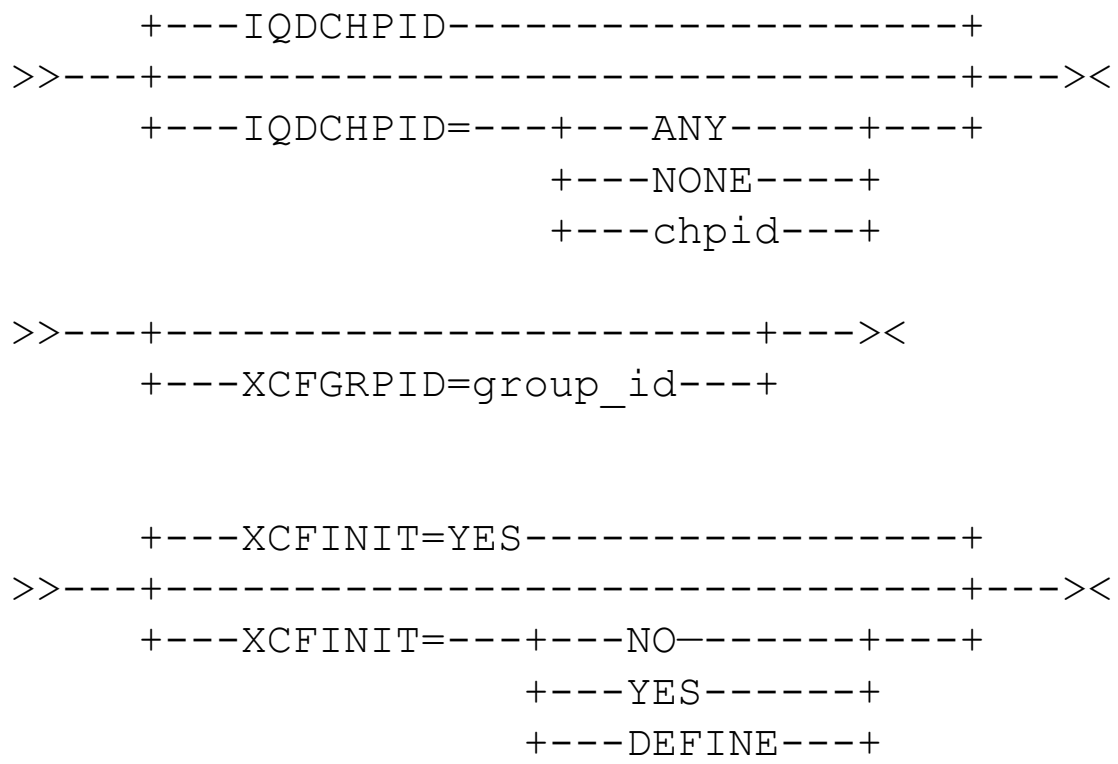
ParmLib

- SYS1.PARMLIB(COUPLExx)

```
PATHIN  DEVICE(dev_num_in1,dev_num_in2,dev_num_in3)
PATHOUT DEVICE(dev_num_out1,dev_num_out2,dev_num_out3)
```

VTAM Start Options

- SYS1.VTAMLST(ATCSTRxx)



VTAM Display

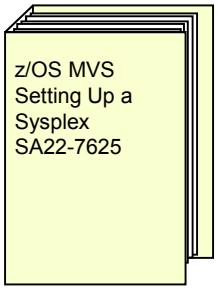
- Display Commands

```
DISPLAY NET,TRL
DISPLAY NET,TRL,TRLMN=trl_maj_node
DISPLAY NET,TRL,TRLMN=ISTTRL
DISPLAY NET,TRL,TRLE=trl_entry_name
DISPLAY NET,TRL,TRLE=IUTSAMEH
DISPLAY NET,TRL,TRLE=IUTIQDIO
DISPLAY NET,TRL,TRLE=IUTIQDxx (xx=CHPID)
DISPLAY NET,TRL,TRLE=ISTTlsrs (ls=local
  &SYSCLONE,rs=remote &SYSCLONE)
DISPLAY NET,TRL,XCFCP=cp_name
DISPLAY NET,VTAMOPTS
```

HiperSockets Defined in OSPF

- Subnet mask and metric on DynamicXCF PROFILE statement is for ORouteD use only.
  - OSPF_Interface in OMPRoute config required for OSPF advertisements to be sent over the XCF network.
- Define DynamicXCF links in OMPROUTE with wildcard IP addresses
  - Avoids having to determine what the names of the various flavors of dynamic XCF links will be.
  - The syntax checker does not require that the "Name" be specified.
- Point-to-Multipoint Networks
  - MPC, XCF, IUTSAMEH
  - Unicast to Each Interface: Hello (Type 1)
  - Does not require DR election
- Broadcast Multiaccess Network
  - Token Ring, Ethernet, FDDI, LANE, **HiperSockets**
  - Multicast to 224.0.0.5: Hello (Type 1)
  - Requires DR election
  - OSPF_INTERFACE NON_BROADCAST=YES should not be defined.
- The HELLO protocol determines who the Designated Router (DR) will be.
- Role of the DR:
  - It is adjacent to all other routers on the network.
  - It generates and floods the network link advertisements on behalf of the network.
    - Reduces amount of router protocol traffic, as only the DR is responsible for flooding the network with the information.
  - It is responsible for maintaining the network topology database.
- Router with highest Router_Priority becomes DR on a broadcast multiaccess network.
  - If there is a tie, the router with the higher Router_ID becomes the DR.
  - If the Router ID is not specified, the IP address of one of the OSPF interfaces will be used as Router ID.
    - Define IP address of static VIPA or physical interface for RouterID to avoid selection of a Dynamic VIPA which could move.
- If your z/OS system is not to be used primarily for routing, consider setting Router_Priority to 0 for all non-HiperSockets interfaces so that the system is ineligible to become the DR.

Appendix: DynamicXCF Details



VTAM & TCP/IP Sysplex Support Background

- Each VTAM in the Sysplex:
  - Joins ISTXCF and ISTCFS01 Sysplex groups
  - Establishes DynamicXCF Connectivity (XCFINIT start option)
  - Can access Generic Resource and Multinode Persistent Session (MNPS) structures (STRGR & STRMNPS start options)
- Each TCP/IP stack in the Sysplex:
  - Joins EZBTCPCS Sysplex group
  - Exchanges IP address information
  - Coordinates DVIPA movement
  - Can be a Sysplex Distributor target
  - Can access Sysplex-Wide Security Associations (SWSA) and Sysplexports structures
  - Sets up TCP/IP DynamicXCF connectivity
    - Dynamic Same Host - if stacks are in the same LPAR
    - Dynamic HiperSockets - if stacks are on the same CEC
    - VTAM's DynamicXCF Connectivity
- DynamicXCF requires cross-system coupling facility (XCF) messaging support. The system parameter PLEXCFG (in the IEASYSxx Member of Parmlib) defines XCF support (XCF messaging and signaling).
- Some APPN enablement is required for TCP/IP DynamicXCF. If APPN searching and routing are not desired, the following VTAM Start Options should be researched:
  - NODETYPE=EN
  - CONNTYPE=LEN
  - CPCP=NO
  - HPR=RTP (default)
  - XCFINIT=YES (default)
  - The above Start Options should allow enough APPN function to be present in VTAM to support DYNAMICXCF but allow searching, session initiation, and routing to remain subarea only).

Sysplex and Subplex Background

	Base Sysplex	Parallel Sysplex
XCF signaling or messaging	Yes	Yes
Coupling Facility installed	No	Yes
Structures defined	No	Yes
DVIPA	Yes	Yes

- Base Sysplex
  - In Base Sysplex XCF signaling or messaging flows over ESCON CTCs, or ICP (Internal Coupling Peer Channel) in CEC (COUPLExx).
  - Coupling Facility is not required for Base Sysplex.
  - Structures are not defined in a Base Sysplex.
- Parallel Sysplex
  - In Parallel Sysplex XCF signaling or messaging flows over Coupling Facility (COUPLExx).
  - Coupling Facility is required for Parallel Sysplex.
  - Structures are defined in COUPLExx and stored in Coupling Facility in a Parallel Sysplex.

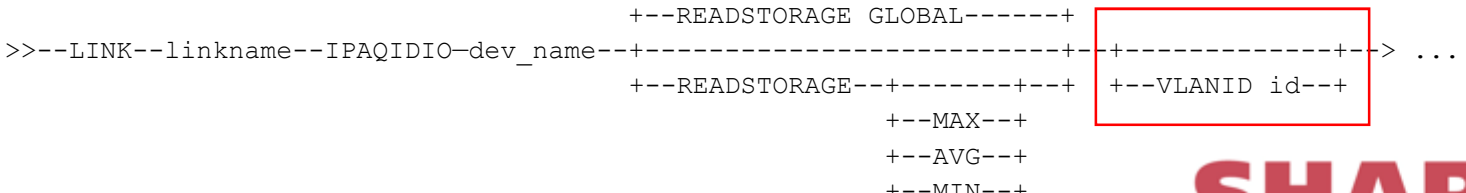
- Partitioning the Sysplex into Subplexes restricts use of Sysplex functions to multiple security areas.
- A Subplex is a subset of VTAM nodes or TCP stacks in the Sysplex.
  - All the members of a subplex can establish dynamic connectivity through the Sysplex with other members of the subplex.
  - Members of the subplex cannot establish dynamic connectivity through the Sysplex with non-members of the subplex.
- Functions restricted to within a Subplex (Subplex Scope):
  - VTAM Generic Resources (GR) & Multi-Node Persistent Session (MNPS) resources
  - Automatic connectivity - IP connectivity (DynamicXCF) and VTAM connectivity over XCF
  - TCP/IP stack IP address awareness and visibility (including DVIPA)
  - DVIPA movement candidates
  - Sysplex Distributor target candidates

Subplex Support

- Each VTAM can belong to only one subplex
  - The subplex is defined by the name of the Sysplex groups that VTAM joins
  - One VTAM Subplex per LPAR
- Each TCP/IP stack can belong to only one subplex
  - The subplex is defined by the name of the Sysplex group that TCP/IP joins
  - A TCP/IP Subplex cannot span multiple VTAM Subplexes
  - Different TCP/IP stacks in an LPAR may belong to different TCP/IP Subplexes
- VTAM Subplex Support
  - Start Option XCFGRPID vv
    - where vv is a number between 2 and 31
  - VTAM joins ISTXCFvv and ISTCFSvv Subplex groups
  - STRGR and STRMNPS CF structure names are suffixed with vv
- TCP/IP Subplex Support
  - Profile GLOBALCONFIG statement
    - XCFGRPID tt to subplex TCP/IP
      - where tt is a numeric value between 2 and 31
    - **IQDVLANID nn to subplex HiperSockets for DynamicXCF connectivity**
      - where nn is a numeric value between 1 and 4094
  - TCP/IP will join Sysplex group EZBTvvtt
    - where vv is the VTAM subplex number
  - SWSA and Sysplexports structure names will be suffixed by vvtt
    - EZBDVIPAvvtt and EZBEPORTvvtt
  - **Profile keyword on LINK and INTERFACE statements for HiperSockets**
    - VLANID nn
      - where nn is numeric value between 1 and 4094

HiperSockets VLAN ID Summary

- HiperSockets LAN may be divided into separate Virtual LANs (VLANs)
  - DynamicXCF HiperSockets VLANs are required for different TCP/IP stacks in an LPAR to belong to different TCP/IP Subplexes.
    - TCP/IP Profile GLOBALCONFIG IQDVLANid nnnn
  - Manual HiperSockets VLANs may be defined with VLANID on LINK and INTERFACE to divide a HiperSockets LAN into multiple subsets just as OSA QDIO VLAN support.



The End

The End