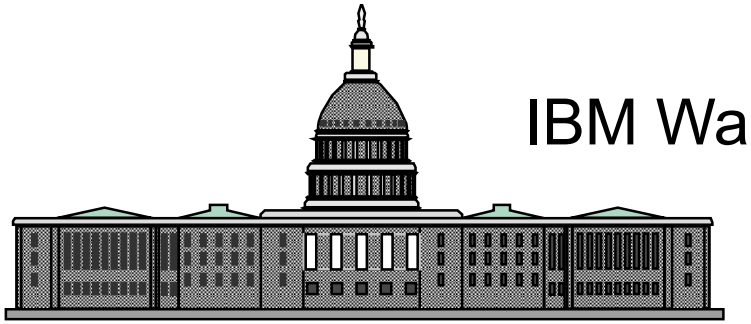
z/OS Communications Server use of OSA (Open System Adapter)

Linda Harrison
lharriso@us.ibm.com

IBM Washington Systems Center



Trademarks and Limitations

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

IBM
z/OS

The following are trademarks or registered trademarks of other companies.

Microsoft is a registered trademark of Microsoft Corporation in the United States and other countries.

•All other products may be trademarks or registered trademarks of their respective companies.
Refer to www.ibm.com/legal/us for further legal information.

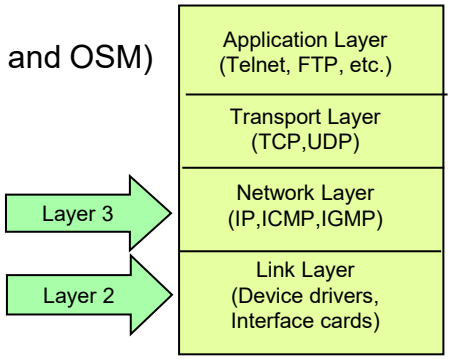
Agenda

- OSA Card Feature Codes and CHPID Types
- TCP/IP Gratuitous ARP Fail-Over Support
- TCP/IP IPCONFIG Multipath Support
- OSA Definition
- OSA TCP/IP Device/Link/Interface Parameters
- OSA Read Storage Usage
- OSA Offload Support
- OSA Interface Isolation
- Primary Router and Layer 3 Virtual MAC (VMAC)
- OSA VLAN Support
- MTU and Jumbo Frames
- Latency and Workload Queuing
- QDIO Accelerator
- HiperSockets Integration with IEDN
- OSAINFO Command
- OSA Error Support
- Synchronization of OSA Diagnostics Data (QDIOSYNC)
- OSA-Express Network Traffic Analyzer (OSAENTA)
- SNMP Support
- OSA TCP/IP Considerations
- Source IP Address
- Policy Based Routing
- Commands
- More Information
- Summary of Features and Other Operating Systems
- Appendix: Syntax

OSA Card Feature Codes and CHPID Types

Open Systems Adapter (OSA) Evolution

- 1995 OSA, 1996 OSA-2, 1999 OSA-Express, 2004 OSA-Express2, 2008 OSA-Express3, 2011 OSA-Express4, 2015 OSA-Express5, 2016 OSA-Express-6, 2017 OSA-Express7
- OSA Modes (There are many OSA modes not covered in this presentations)
  - non-QDIO mode (CHPID type OSE) – IBM z16 will be the last hardware release to support 1000BaseT and OSA in OSE mode
    - Requires OSA/SF (unless only used for TCP/IP and not shared between LPARs)
      - Every IP address, including VIPA, must be manually defined.
    - Supports LCS protocol (Passthru in OAT)
      - TCP/IP traffic to z/OS
      - Both TCP/IP and SNA traffic to Linux (SNA: Communication Controller for Linux and Communications Server for Linux)
      - Maximum of 8 IP addresses per OAT entry
      - Maximum of 2048 IP addresses per port
    - Supports LSA protocol (SNA in OAT)
      - SNA traffic to z/OS
  - QDIO mode (Layer3) (CHPID type OSD, OSX, and OSM)
    - Supports TCP/IP traffic only
  - QDIO Layer2 mode (not supported by z/OS)
    - Only supported by zLinux and z/VM
    - Supports all traffic protocols
- Enterprise Extender (EE)
  - Encapsulates SNA in IP
  - Provided by z/OS Communications Server
    - Requires endpoint on both sides
- System z Hardware Networking web page
  - http://www.ibm.com/systems/z/hardware/networking/index.html



z/OS QDIO Connectivity
- OSD provides OSA connectivity to external LAN (outside of zEnterprise)
- OSX provides OSA connectivity to IEDN (Intraensemble data network inside zEnterprise)
- OSM provides OSA connectivity to INMN (Intraensemble management network inside zEnterprise)

CHPID Type

- OSA CHPID type OSD
  - QDIO mode connection to external Ethernet LAN
- OSA CHPID type OSC
  - Console mode connection to external Ethernet LAN for console, TN3270, and telnet access to LPARs.
- OSA CHPID type OSE
  - Non-QDIO mode connection to external Ethernet LAN
  - **IBM z16 will be the last hardware release to support 1000BaseT and OSA in OSE mode.**
- OSA CHPID type OSX
  - QDIO mode connection to Intraensemble Data Network (IEDN)
  - **IBM zBX and OSA in OSX mode no longer supported.**
- OSA CHPID type OSM
  - QDIO mode connection to Intranode Management Network (INMN)
  - **IBM zBX and OSA in OSX mode no longer supported.**
- OSA CHPID type OSN
  - OSA for NCP mode connects NCP running in LPARs with Communication Controller for Linux on System z (CCL) to LPARs running TPF and VTAM in z/OS, z/VM, and z/VSE.
  - **IBM Communication Controller and OSA in OSN mode no longer supported.**

OSA CHPID vs. OSA Port

- Prior to OSA-Express3:
  - OSA cards have one or two ports per card
  - All OSA cards have one port per CHPID.
- OSA-Express3 and later except 10 GbE:
  - OSA cards have two or four ports per card
  - All OSA cards have two ports per CHPID.
- OSA-Express3 and later 10 GbE:
  - OSA cards have one or two ports per card
  - All OSA cards have one port per CHPID.
- Each OSA CHPID is configured completely independently from the other OSA CHPID on the same OSA card.
- Each OSA CHPID is configurable in only one OSA mode.
  - When an OSA CHPID is configured in a particular OSA mode then all ports on that CHPID are configured in that OSA mode.
  - If an OSA CHPID is configured in OSN mode then the CHPID is dedicated to OSN usage and both ports on that CHPID are disabled.
  - If an OSA CHPID is configured in OSM mode then port 0 is usable for INMN connection and port 1 is disabled.
- Not all OSA modes are supported on all OSA cards.
- See the OSA documentation and the IBM System z Connectivity Handbook Redbook, SG24-5444, for supported OSA cards and corresponding OSA modes.
- Throughout this presentation where OSA is mentioned it implies OSA port.

Queued Direct I/O (QDIO)

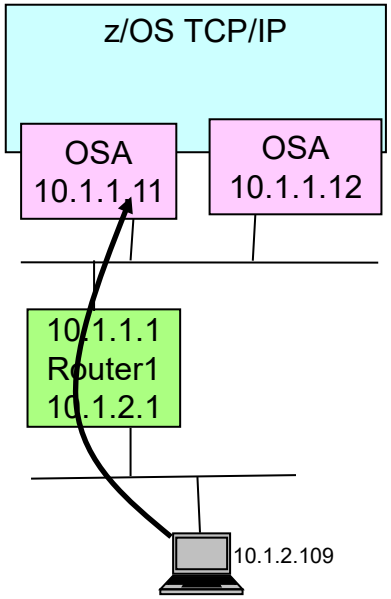
- Supports high-speed LPAR-to-LPAR communication
  - OSA microprocessor communicates directly with System z using data queues in memory
- Continuous direct data exchanges
  - Communications remain active
- Utilizes Direct Memory Access (DMA) protocol
  - Reduced I/O interrupts
  - Reduced Latency
- Dynamically maintains the OSA Address Table (OAT).
  - Does not require OSA/SF.
    - OSA/SF is required for non-QDIO except when only port 0, OSA is not shared, and SNA is not used.
  - All addresses are dynamically downloaded to the OSA.
  - Any VIPA movement/changes are dynamically downloaded to the OSA from TCP/IP.
- Layer 3
  - 4 outbound QoS (Quality of Service) queues for priority queuing of data
  - IP Only (use Enterprise Extender for QDIO advantages with SNA traffic)
  - IP-Assist to handle MAC addressing, ARP processing, some filtering
    - TCP/IP Netstat display and purge of QDIO ARP cache
- Layer 2 (not supported by z/OS)
  - Multiple virtual MAC addresses are supported.
  - HCD (IOCP) definitions are the same as Layer 3.
  - Layer 2 and virtual MAC addresses are configured in the operating system (zLinux or z/VM).

OSA-Express3 and later have an improved QDIO mode using a data router model to eliminate store and forward delays that can occur.

Layer 2 Requires
Hardware: z990+
Software: zLinux kernel 2.6-based distributions or later (SLES9 SP3+ or RHEL U3+), z/VM 5.4+

TCP/IP Gratuitous ARP Fail-Over Support

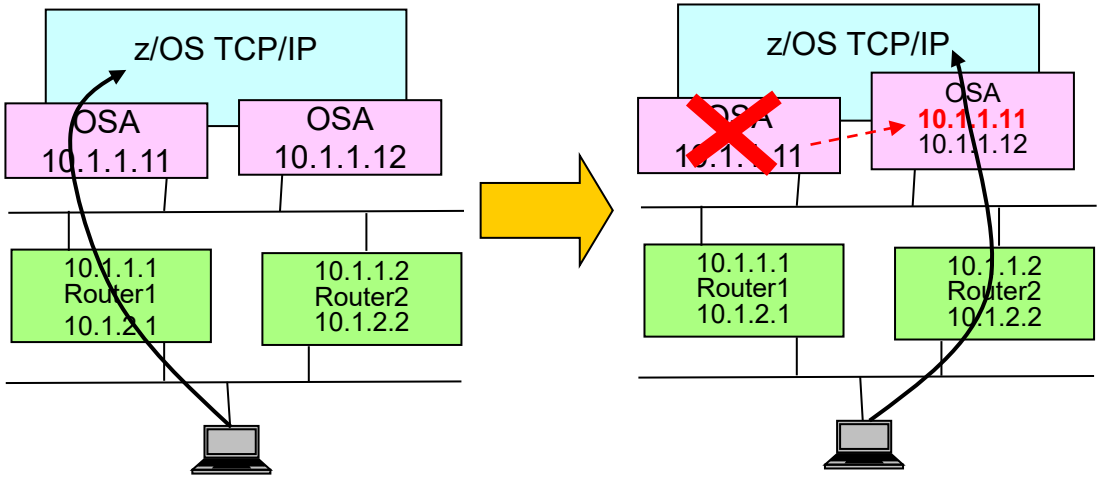
Original OSA with Inbound Connections



OSA is single point of failure.

- Inbound Connection Request
  - Remote Client and z/OS Server
- Remote client sends in a connection request.
  - Source in IP Header = Remote client IP address
  - Destination in IP Header = z/OS server IP address
- Response is sent back to the client.
  - Source in IP Header = z/OS server IP address
  - Destination in IP Header = Remote client IP address
- Destination OSA IP Address
  - Connection Request Source=10.1.2.109 and Destination=10.1.1.11
  - Response Source=10.1.1.11 and Destination=10.1.2.109
  - All Inbound packets are received over the destination OSA (unless OSA failure).
  - An OSA outage may cause a connection drop.
- For a TCP connection, the same client/server IP addresses are used for the life of the connection.
- TCP/IP Routing Table is used to determine which OSA the outbound packets are sent over.
  - Static Routing and OSPF support multiple concurrent parallel routes if IPCONFIG MULTIPATH is configured.
  - Policy Agent and NetAccess can effect the decision.

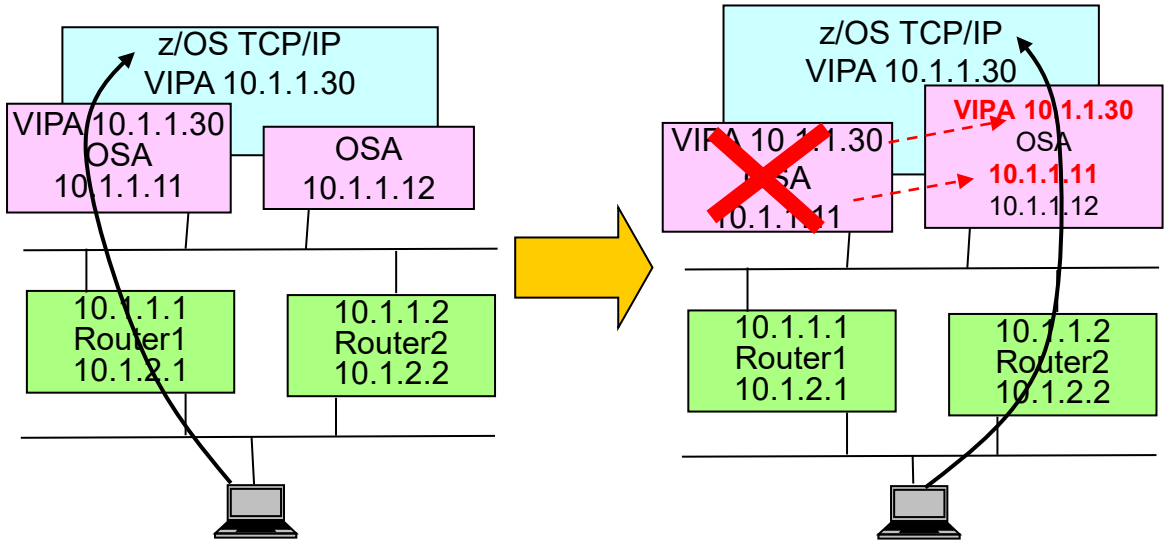
OSA QDIO Gratuitous ARP Fail-over



OSA backup capability. Connections are not dropped.

- When devices are started by the TCP/IP stack the stack determines if there is already a parallel connection to the same network. If at a later time one of the OSA connections goes down the other OSA will "take-over" for it. The other OSA will send out a gratuitous ARP with the IP address of the failed OSA and "own" the IP address of the failed OSA until its recovery.
  - Note that this does require that both OSAs did originally come up so that the stack marked them as parallel to the same network.
- If one of the OSAs fails (any failure that causes the LINK to go down), then OSA QDIO Gratuitous ARP Fail-over occurs.
  - The failed OSA IP address is taken over by one of the working OSAs.
  - A gratuitous ARP will be sent out to associate that IP address with the working OSA's MAC (Media Access Card) address.
- OSA QDIO Gratuitous ARP Fail-over does not require any configuration.
- The same failover support exists for IPv6 where gratuitous neighbor advertisements are sent rather than gratuitous ARPs.

ARP Fail-over with VIPA and Static Routing



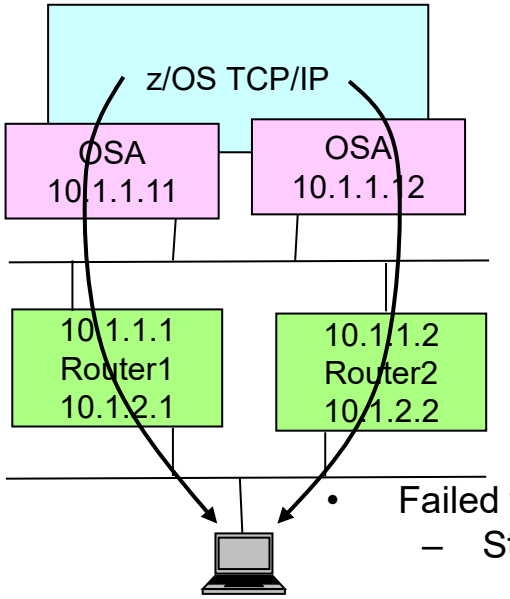
Summary:

- Assign OSA IP addresses and VIPA all in the same subnet when using static routing to take advantage of Gratuitous ARP support.
- Assign OSA IP addresses and VIPA in different subnets when using dynamic routing to avoid Gratuitous ARP packets.

- If static routing is defined between the z/OS system and the first hop routers, then a VIPA in the same subnet as the OSA attachments is "owned" by one of the OSAs at any given time (randomly/unpredictable). "Owned" means that the OSA will respond to ARP for that VIPA. If the OSA that "owns" the VIPA goes down then one of the other OSAs will not only send out a gratuitous ARP for the failed OSA IP address but also for the VIPA.
  - Note that this does require that both (or multiple) OSAs originally come up so that the stack marked them as parallel to the same network.
  - Outbound traffic could be "load balanced" using IPCONFIG MULTIPATH but inbound traffic with the VIPA destination would all be sent to the VIPA "owning" OSA.
- The same failover support exists for IPv6 where gratuitous neighbor advertisements are sent rather than gratuitous ARPs.
- Prior to z/OS V1.10, the stack updates OSA to perform ARP processing for all VIPAs.
  - This causes many unnecessary gratuitous ARPs which can cause confusion in routers and sniffer traces.
- z/OS V1.10 with the IPv4 INTERFACE and /num_bits, causes the stack to only update OSA for a VIPA if it the VIPA is in the same subnet as the OSA.
  - This eliminates superfluous gratuitous ARPs.

TCP/IP IPCONFIG Multipath Support

IPCONFIG MULTIPATH

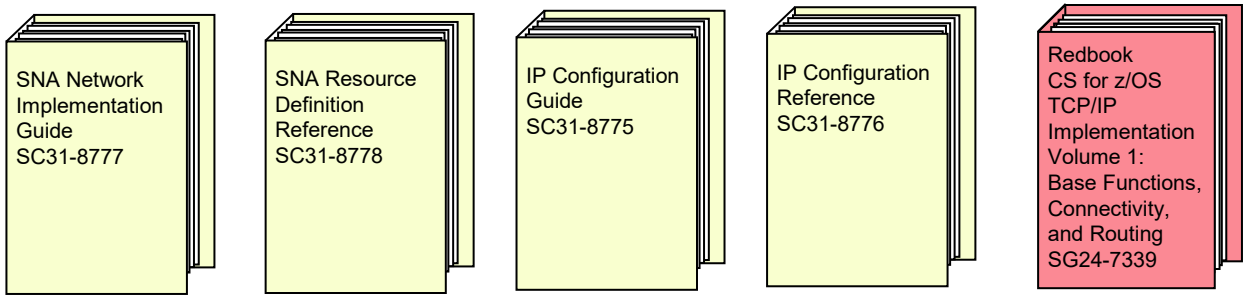



```
BEGINROUTES
 ROUTE   10.1.1.0/24    =       OSALNK11   MTU   1492
 ROUTE   10.1.1.0/24    =       OSALNK12   MTU   1492
 ROUTE   DEFAULT   10.1.1.1    OSALNK11   MTU   1492
 ROUTE   DEFAULT   10.1.1.2    OSALNK11   MTU   1492
 ROUTE   DEFAULT   10.1.1.1    OSALNK12   MTU   1492
 ROUTE   DEFAULT   10.1.1.2    OSALNK12   MTU   1492
ENDROUTES
```

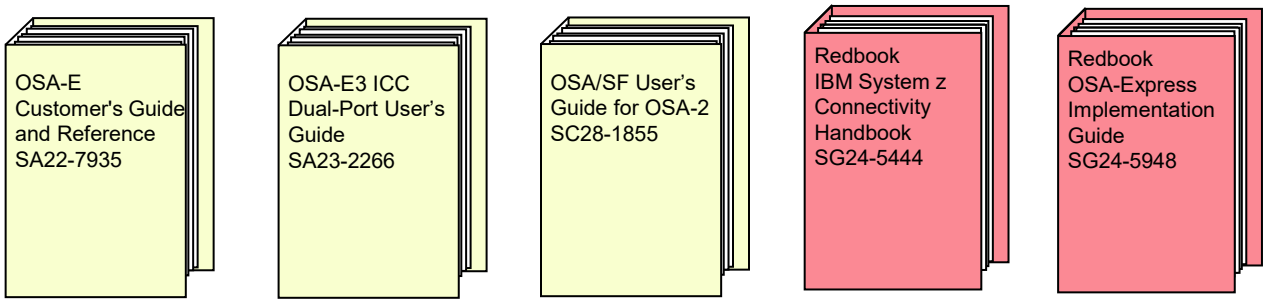
- IPCONFIG MULTIPATH "load balances" outbound packets
  - Static Routing and OMPROUTE OSPF support IPCONFIG MULTIPATH
  - Default Multipath routing is per connection as opposed to per packet
- Failed first hop router
  - Static Routing Dead Gateway Detection
    - TCP connection will eventually timeout (3 to 10 minutes) and TCP will redrive the route selection algorithm and hopefully get a successful connection
    - UDP and RAW packets are lost
    - With Static Routing HSRP/VRRP should be used between first hop routers
      - See http://cisco.com for HSRP and VRRP details.
  - Dynamic Routing detects failures
    - OSPF defaults to 40 seconds and RIP takes up to 3 minutes

Inbound Load Balance

Inbound load balance is really determined by the first hop router.
If the first hop router is capable of load balancing traffic across multiple OSAs when the destination is a VIPA address, then inbound traffic will be truly load balanced.
Routers have static and OSPF load balancing capability similar to z/OS outbound Multipath. See
http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094820.shtml



OSA Definition

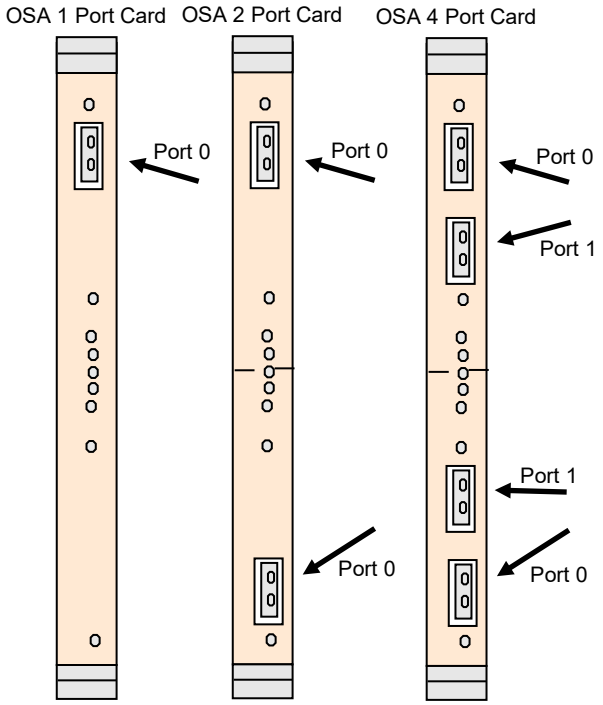


OSE and OSD Customization Requirements

- non-QDIO OSA Customization
  - OSA/SF Required for shared OSA and/or SNA support.
  - HCD (IOCP) CHPID type OSE
    - TCP/IP requires 2 devices
      - The first device is defined in the OAT (OSA Address Table) as “passthru”
    - SNA requires 1 device
      - The device is defined in the OAT as “SNA”
  - VTAM Customization
    - Only required for SNA protocol over the OSA
    - Define VTAMLST External Communication Adapter (XCA)
  - TCP/IP Customization
    - Only required for TCP/IP protocol over the OSA
    - TCP/IP Profile
      - Define DEVICE, LINK, HOME
      - Configure BeginRoutes or OMPROUTE definitions
      - Define START statement
- QDIO OSA Customization (TCP/IP only)
  - HCD (IOCP) CHPID type OSD and 3 devices
  - VTAM Customization
    - Define TRL
  - TCP/IP Customization
    - TCP/IP Profile
      - Define DEVICE, LINK, HOME or INTERFACE
      - Configure BeginRoutes or OMPROUTE definitions
      - Define START statement

•Prior to OSA-Express3 each OSA provides a single port per CHPID.
•OSA four port cards provide two ports per CHPID.
•No difference between HCD for OSA with single port per CHPID and two ports per CHPID.

IBM z16 is the last hardware release to support OSA-E 1000BaseT and OSE mode.

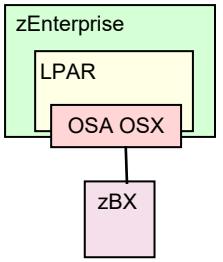


OSA/SF device is defined with IODEVICE UNITADD=FE and UNIT=OSAD.
All other devices are defined with IODEVICE UNIT=OSA.
OSA/SF is not required for OSD but was most useful for OSD prior to the OSAINFO command support for retrieving OSA information and status.
OSA/SF not supported for OSX and OSM.

OSX Configuration Requirements

- OSX definition (minimum of two OSX for availability)
  - HMC Customization
    - Define Ensemble
    - Define VLAN ID
    - Add LPAR and CHPID to VLAN
  - HCD (IOCP) Customization
    - Define CHPIDTYPE OSX with minimum of 3 devices.
  - VTAM Customization
    - Define VTAM start option ENSEMBLE=YES in SYS1.VTAMLST(ATCSTRxx).
    - Define TRL or have it dynamically created by defining CHPID on TCP/IP INTERFACE.
  - TCP/IP Profile Customization
    - Define INTERFACE IPAQENET or IPAQENET6 with parameters CHPIDTYPE OSX and VLANID.
      - Optionally, include the CHPID parameter to have VTAM dynamically create the TRLE (transport resource list element).
        - » TRLE name is IUTXT0*xx* (where *xx* is the configured CHPID parameter)
        - » Port name is IUTXP0*xx* (where *xx* is the configured CHPID parameter)
        - » If the values that are chosen do not suit your needs (for example, you need more than 17 DATAPATH devices), you can define your own QDIO TRLE with PORTNAME=IUTXP0xx.
      - If defining IPv6 an external router may be used for stateless autoconfiguration, otherwise define IPv6 address and prefixes.
    - Configure BeginRoutes or OMPROUTE definitions
    - Define START statement
  - OSX for IEDN enabled HiperSockets (IQDX) defined the same way.

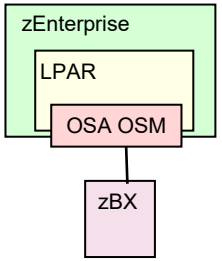
IBM zBX and OSA in OSX mode are no longer supported.



OSM Configuration Requirements

- OSM definition (two OSM ports for availability)(IPv6 only)(Port 0 only)
  - HMC Customization
    - Define Ensemble
    - Define VLAN ID
    - Add LPAR and CHPID to VLAN
  - HCD (IOCP) Customization
    - Define CHPIDTYPE OSM with minimum of 3 devices.
  - VTAM Customization
    - Define VTAM start option ENSEMBLE=YES in SYS1.VTAMLST(ATCSTRxx)
    - VTAM TRL is automatically created.
      - TRLE name is IUTMT0*xx* (where *xx* is the OSM CHPID)
      - Port name is IUTMP0*xx (*where *xx* is the OSM CHPID)
  - RACF (SAF) SERVAUTH class resource controls access to OSM.
    - EZB.OSM.sysname.tcpname
  - TCP/IP Customization
    - Enable IPv6 by adding AF_INET6 in PARMLIB member BPXPRMxx after NETWORK.
    - OSM interfaces are not configured in the TCP/IP profile, but are automatically generated.
      - Only link-local IPv6 addresses
    - Reserve UDP port that platform management appl will listen on for multicast traffic.
  - All traffic to and from the INMN over OSM interfaces is exempt from network access control, and is instead subject to OSM access control.
    - Only multicast addresses to which a platform management application binds are subject to network access control.
  - If you enable IPsec for IPv6, you can configure a security class for IP filtering that applies to all OSM interfaces.
    - OSMSECCLASS parameter on the IPCONFIG6 statement

IBM zBX and OSA in OSM mode are no longer supported.



OSA HCD

- non-QDIO OSA HCD (IOCP)
  - Channel path type=OSE
  - CNTLUNIT UNIT=OSA
  - Device type=OSA
    - 1 Device for LSA (Link State Architecture) protocol (z/OS SNA)
      - 1 device per VTAM
    - 2 Devices for LCS (LAN Channel Station) protocol (z/OS TCP/IP, Linux TCP/IP or SNA)
      - 1 Read and 1 Write per IP stack
  - OSA/SF Required for shared OSA and/or SNA support.

IBM z16 is the last hardware release to support OSA-E 1000BaseT and OSE mode.

HCD (IOCP) parameter CHPARM=00 enables Priority Queuing.

OSA mode OSM does not support Priority Queuing so it is automatically set to CHPARM=02 for OSM.

CHPARM=00 for OSA mode OSC is used for TN3270E protocol.
CHPARM=40 for OSA mode OSC is used for 3215 protocol to TPF.

- QDIO OSA HCD (IOCP)
  - Channel path type=OSD,OSX,OSM
  - CNTLUNIT UNIT=OSA
  - Device type=OSA
    - Minimum of 3 Devices per z/OS
      - 1 Device for Read processing
      - 1 Device for Write processing
      - 1 Device for the Data Path per TCP/IP DEVICE or INTERFACE
    - i.e.. z/OS with CINET environment with 2 TCP/IP stacks would require 4 devices. (1 Read + 1 Write + 2 Data Path = 4)

Example of an OSA-Express2 CHPID or two ports on one CHPID of an OSA-Express3:

```
CHPID PATH=(CSS(0.1),02),SHARED, PARTITION=((CSS(1),(A12),(=))),
    PCHPID=1C0,TYPE=OSD
CNTLUNIT CUNUMBR=2980,PATH=((CSS(0),02),(CSS(1),02)),UNIT=OSA
IODEVICE ADDRESS=(2980,015),UNITADD=00,CUNUMBR=(2980),UNIT=OSA
IODEVICE ADDRESS=298F,UNITADD=FE,CUNUMBR=(2980),UNIT=OSAD
```

Another example of two ports on one CHPID of an OSA-Express3:

```
CHPID PATH=(CSS(0.1),02),SHARED, PARTITION=((CSS(1),(A12),(=))),
    PCHPID=1C0,TYPE=OSD
CNTLUNIT CUNUMBR=3980,PATH=((CSS(0),02),(CSS(1),02)),UNIT=OSA
IODEVICE ADDRESS=(3980,015),UNITADD=00,CUNUMBR=(3980),UNIT=OSA
IODEVICE ADDRESS=398F,UNITADD=FE,CUNUMBR=(3980),UNIT=OSAD
IODEVICE ADDRESS=(4980,015),UNITADD=20,CUNUMBR=(3980),UNIT=OSA
```

VTAM XCA and TRL

- non-QDIO OSA VTAM XCA major node member

```
xcaname  VBUILD TYPE=XCA
portname PORT   ADAPNO=0,
                CUADDR=aaaa,
                MEDIUM=CSMACD,
                SAPADDR=bb
groupname GROUP DIAL=NO,
                ISTATUS=INACTIVE
linename  LINE  USER=SNA
puname    PU    MACADDR=cccccccccccc,
                PUTYPE=4,SUBAREA=ddd,TGN=e,
                SAPADDR=ff
```

On OSA two-port/CHPID cards code ADAPNO=0 for port 0 and ADAPNO=1 for port 1.

IBM z16 is the last hardware release to support OSA-E 1000BaseT and OSE mode.

- QDIO OSA VTAM TRL major node member

```
TRL VBUILD TYPE=TRL
   trl_name TRLE LNCTL=MPC,
     READ=(xxx),
     MPCLEVEL=QDIO,
     WRITE=(yyy),
     DATAPATH=(zzz),
     PORTNAME=device_name,
     PORTNUM=1
```

All z/OS systems that share a port must define the port with the same PORTNAME.
PORTNAME relief for z/VM and zLinux APAR PQ73878

PORTNUM is always 0 for single port per CHPID OSA.
–When PORTNUM=1 is defined for single port per CHPID OSA it is ignored.
On OSA two-port/CHPID cards code PORTNUM=0 for port 0 and PORTNUM=1 for port 1.

Note: z/OS PORTNAME must be unique. There is a subtle difference in PORTNAME support between OSA-E2 and OSA-E3. Two OSA ports may use the same PORTNAME if they are on the same OSA card but on different CHPIDs and not both defined to the same VTAM. With OSA-E2 the ports are always on different CHPIDs but with OSA-E3 two ports could be on the same CHPID.

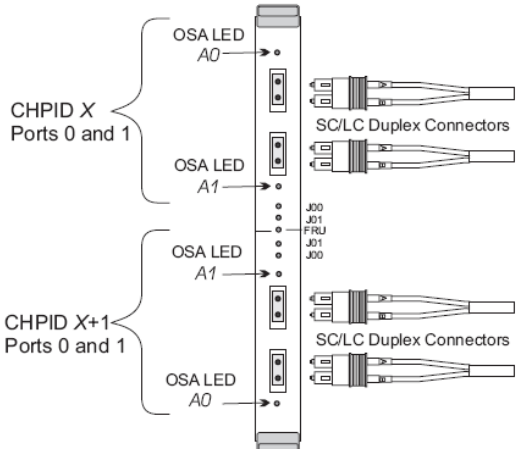
OSA TCP/IP Device and Interface

IBM z16 is the last hardware release to support OSA-E 1000BaseT and OSE mode.

- non-QDIO TCP/IP Profile DEVICE (IPv4 only)
  - DEVICE device_name LCS device_number…
  - LINK link_name link_type link_number device_name…
  - HOME ipaddr link_name
  - START device_name
  - link_number must match the OSA port number on the OSA card, 0 or 1.
- QDIO TCP/IP Profile DEVICE (IPv4 only)
  - DEVICE device_name MPCIPA…
  - LINK link_name link_type device_name…
  - HOME ipaddr link_name
  - START device_name
- QDIO TCP/IP Profile INTERFACE (IPv4)
  - INTERFace intf_name intf_action IPAQENET…
  - START interface_name
- QDIO TCP/IP Profile INTERFACE (IPv6)
  - INTERFace intf_name intf_action IPAQENET6…
  - START interface_name
- Port name and device name must match:
  - non-QDIO
    - DEVICE device name = OSA/SF port name
  - QDIO
    - TRLE port name = DEVICE device name = INTERFACE port name
- TRLE port name must match in all z/OS TCP/IP stacks for a shared OSA port.
- Per TCP/IP Stack:
  - Only one TRLE per port.
  - Only one LINK per DEVICE.
  - Only one DEVICE and/or INTERFACE per TRLE without VMAC.

On OSA two-port/CHPID cards code link_number of 0 for port 0 and link_number of 1 for port 1.

OSA-Express3



- If the OSA is configured for both IPv4 (using DEVICE and LINK) and IPv6 for a stack, then you can define the same VMAC for both the INTERFACE statement and the LINK statement, or a stack can use one VMAC on the LINK statement for IPv4 usage and a different VMAC on the INTERFACE statement for IPv6 usage.
- If the OSA is configured for both IPv4 (using the INTERFACE statement) and IPv6 for a stack, then the stack must use one VMAC on the INTERFACE statement for IPv4 usage, and a different VMAC on the INTERFACE statement for IPv6 usage.

OSA TCP/IP Device/Link/Interface Parameters

non-QDIO LCS and QDIO MPCIPA Link Types

- non-QDIO LCS Link Types
  - ETHERNET
    - Indicates that only Ethernet protocol is used.
  - 802.3
    - Indicates that only 802.3 protocol is used.
    - 802.3 is the recommended ethernet protocol definition unless you know there are older devices using Ethernet II DIX protocol that you know z/OS must communicate with.
  - ETHEROR802.3
    - Indicates that both Ethernet and 802.3 protocols are used.
- QDIO MPCIPA Link Types
  - IPAQENET
    - Ethernet QDIO OSA
    - QDIO always uses ETHEROR802.3.

IBM z16 is the last hardware release to support OSA-E 1000BaseT and OSE mode.

Non-QDIO SYS1.OSA.SIOASAMP(IOAFENET) second position of field identifies port:

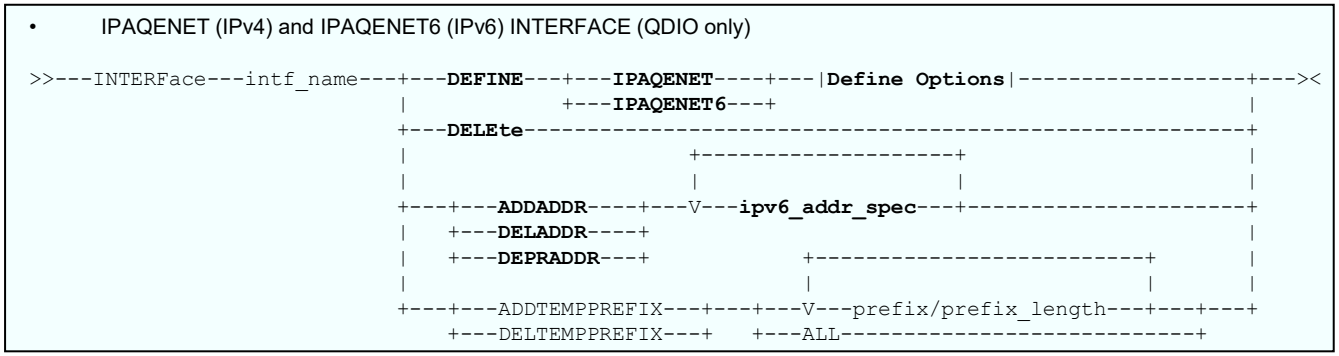
```
fenet.0.1 = GBGCHPD1
fenet.0.2 = OSA for test net 1 access
fenet.0.3 = ENETD1
fenet.0.4 = 020000001000
fenet.0.5 = auto
fenet.1.1 = GBGCHPD2
fenet.1.2 = OSA for test net 1 access
fenet.1.3 = ENETD2
fenet.1.4 = 020000002000
fenet.1.5 = auto
sna.0.1 = connport1
sna.0.2 = 90.00
sna.0.3 = 10.00
sna.0.4 = 1.04
sna.0.5 = 4
sna.0.6 = 8
sna.1.1 = connport2
sna.1.2 = 90.00
sna.1.3 = 10.00
sna.1.4 = 1.04
sna.1.5 = 4
sna.1.6 = 8
```

Non-QDIO SYS1.OSA.SIOASAMP(IOAOSHRA) PORT field identifies port:

```
00(D050)* passthru 00 No 000.000.000.008 SIU ALL
                        172.030.020.71
02(D052)  SNA      00
03(D053)  SNA      00
00(E060)* passthru 01 No 000.000.000.008 SIU ALL
                        172.030.020.72
02(E062)  SNA      01
03(E053)  SNA      01
```

Interface Action Types

- IPAQENET (IPv4) and IPAQENET6 (IPv6) INTERFACE (QDIO only)



- IPAQENET (in z/OS V1.10+) and IPAQENET6 intf_action types:
  - DEFINE
    - Adds the Interface to the list of defined adapters.
  - DELEte
    - Removes the Interface from the list of defined adapters.
    - DELETE does not have any parameters associated with it.
- Additional IPAQENET6 only intf_action types:
  - ADDADDR
    - Adds the address to the defined Interface definition.
  - DELADDR
    - Removes the address from the defined Interface definition.
  - DEPRADDR
    - Deprecates the address in the Interface definition. This makes the address less preferred. See the “Default address selection” section of the “IPv6 Network and Design Guide, SC31-8885”.

Device, Link, and Interface Parameters

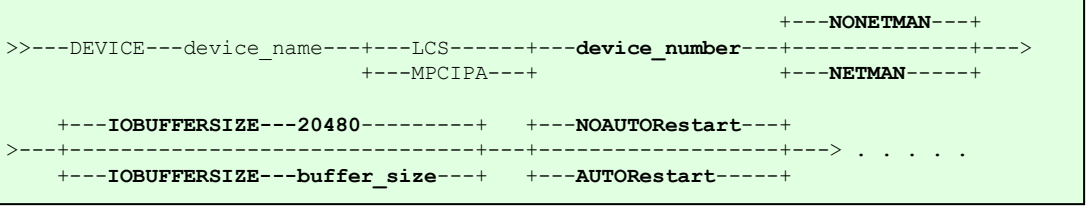
Parameter	non-QDIO LCS DEVICE or LINK	QDIO MPCIPA DEVICE or LINK	QDIO IPAQENET Interface	QDIO IPAQENET6 Interface
AUTORESTART	DEVICE	DEVICE		
CHPID			DEFINE only	DEFINE only
device_number	DEVICE			
DUPADDRDET & TEMPPREFIX				DEFINE only
DYNVLANREG, VLANID & VMAC		IPAQENET LINK only	DEFINE only	DEFINE only
IFSPEED	All LINK types	All LINK types		
INBPERF		All LINK types	DEFINE only	DEFINE only
INTFID				DEFINE only
IOBUFFERSIZE	DEVICE			
IPADDR ipaddr_spec			DEFINE only	
ipaddr_spec				All interface actions
IPBCAST		All LINK types	DEFINE only	
ISOLATE			DEFINE only	DEFINE only
link_number	All LINK types			
MONSYSPLEX	All LINK types	All LINK types	DEFINE only	DEFINE only
MTU			DEFINE only	DEFINE only
NETMAN	DEVICE			
OLM			DEFINE only	DEFINE only
PORTNAME			DEFINE only	DEFINE only
PRIROUTER		DEVICE	DEFINE only	DEFINE only
READSTORAGE		All LINK types	DEFINE only	DEFINE only
SECCLASS	All LINK types	All LINK types	DEFINE only	DEFINE only
SOURCEVIPAINTERFACE			DEFINE only	DEFINE only

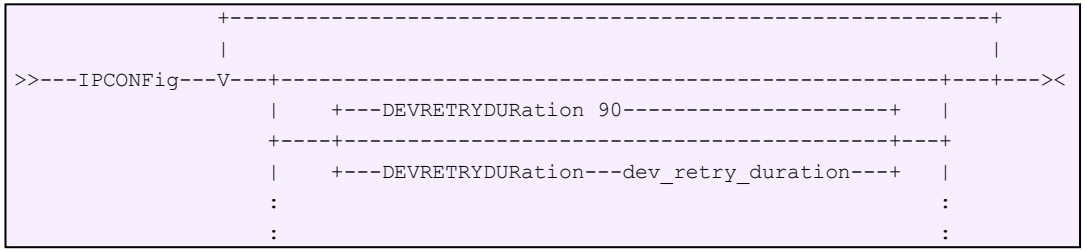
Auto Restart, Device Number, I/O Buffer Size, and Network Monitor

DEVICE only

- AUTORestart/NOAUTORestart
  - Specifying AUTORESTART causes TCP/IP to attempt reactivation following most device-failure indications.
  - Reactivation attempts are done every 30 seconds.
  - The total amount of time reactivation attempts are done is determined by IPCONFIG DEVRETRYDURATION.
  - NOAUTORESTART is the default.
- device_number
  - Specifies the hexadecimal device number.
  - Only the first of the two device numbers is defined.
    - The READ address is defined.
    - The WRITE address is READ+1.
- IOBUFFERSIZE buffer_size (not used by the OSA)
  - Specifies the I/O buffer size.
  - IOBUFFERSIZE must be the default size of 20,480 for OSA.
- NETMAN/NONETMAN (not used by the OSA)
  - Specifies the device is a 3172 that supports IBM Enterprise-specific MIB variables for the 3172.

- LCS (non-QDIO OSA OSE mode) and MPCIPA (QDIO OSA OSD mode) DEVICE

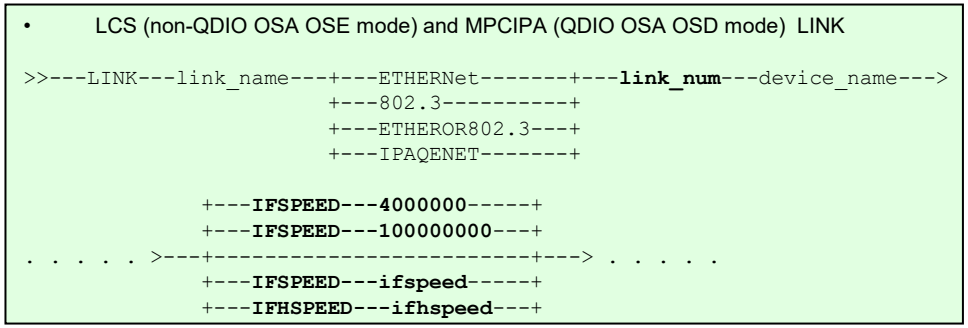




Device and Link will no longer be supported in a future z/OS release.

Interface Speed and Link Number

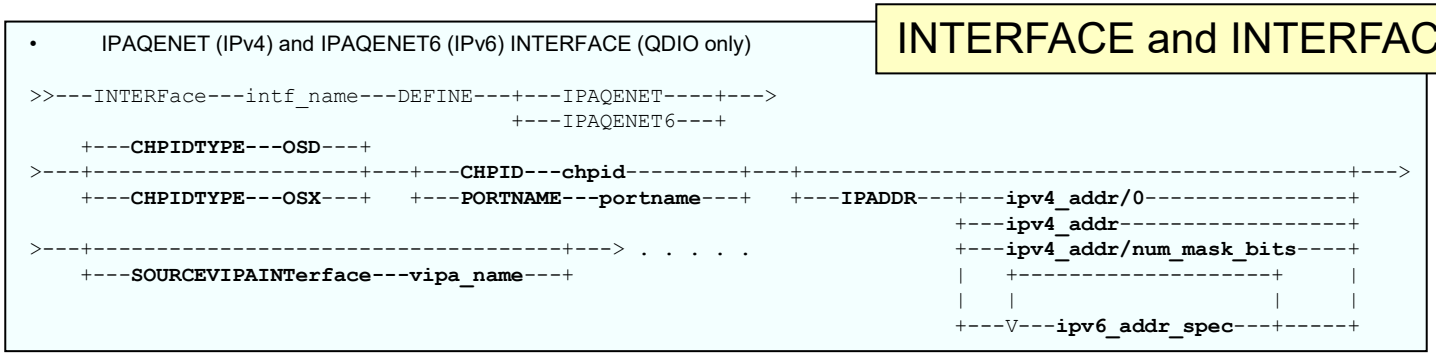
LINK only



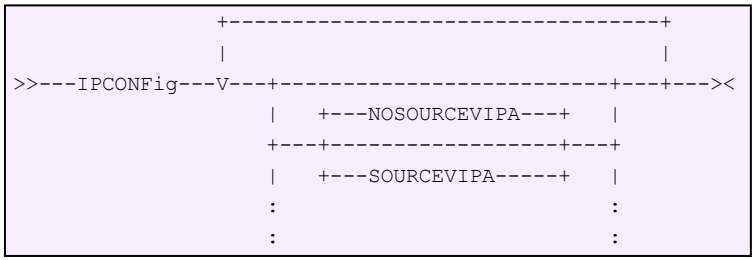
- IFSPEED/IFHSPEED speed_num
  - Estimate of the interface's current bandwidth.
    - IFSPEED in bits per second.
    - IFHSPEED in one million bits per second.
  - Until the interface is successfully started, this value is used by SNMP as the value of the ifSpeed MIB object. After the interface is successfully started, SNMP uses the actual speed reported by the interface as the value of the ifSpeed MIB object. The value of this parameter has no effect on the operation of the device.
- link_num
  - Indicates the port number on the OSA.
  - Ignored unless OSA-Express3 or later.

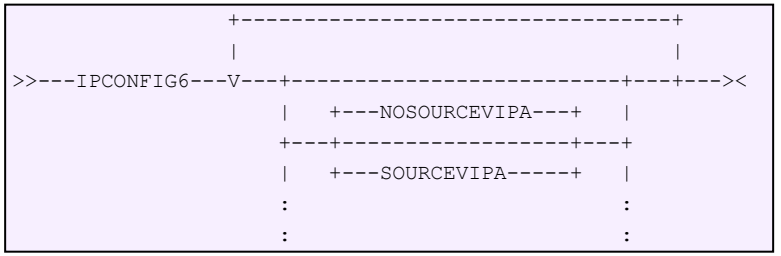
CHPID, IP Address, Port Name, and Source VIPA

INTERFACE and INTERFACE6 only



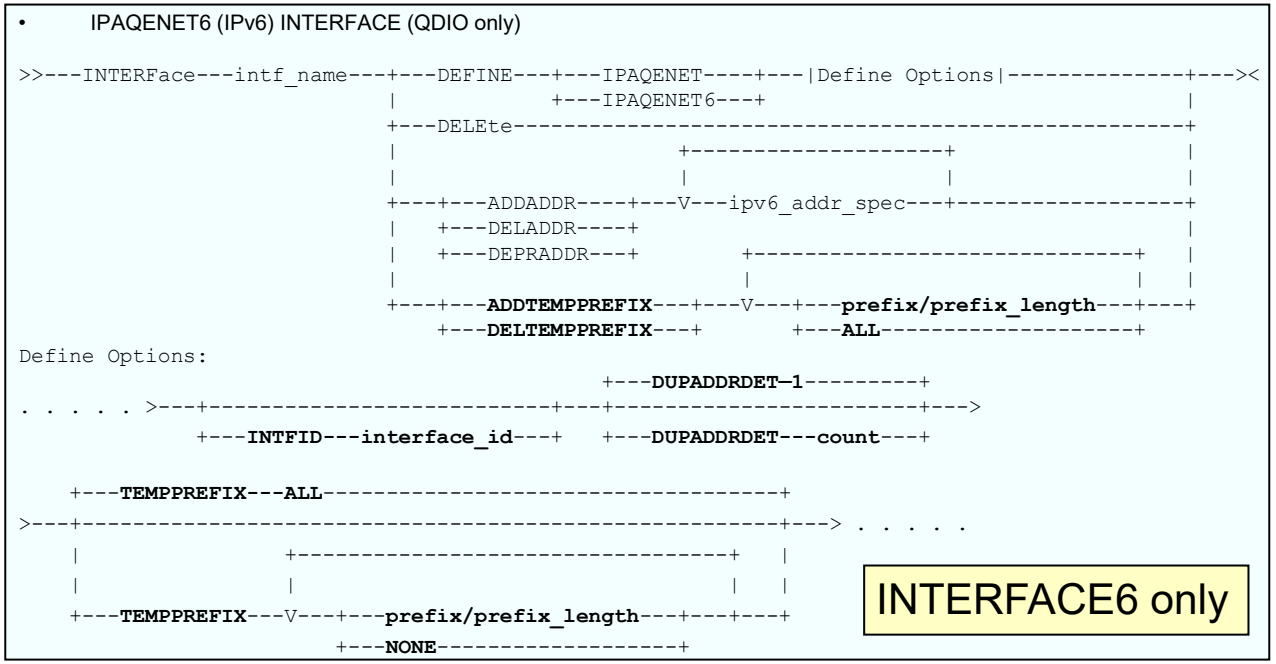
- CHPIDTYPE
  - Indicates either OSD or OSX mode.
- CHPID chpid
  - Identifies the CHPID for the interface. A 2-character hexadecimal value (00 - FF).
- ipaddr_spec
  - Specifies the ipv4_addr, ipv4_addr/mask, ipv6_addr, or prefix/prefix_length.
- PORTNAME port_name
  - Port name and device name must match between TRLE, DEVICE, and INTERFACE.
  - Interface and Portname must be different.





- SOURCEVIPAINTerface vipa_name
  - Specifies which static VIPA interface is to be used for SOURCEVIPA.
  - Requires IPCONFIG or IPCONFIG6 SOURCEVIPA.

Duplicate Address Detection Count, Interface ID, and Temporary Address Prefix

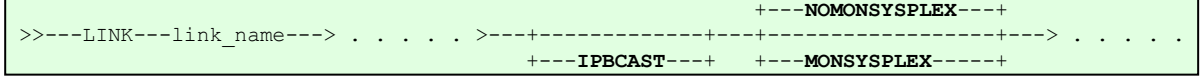


- DUPADDRDET count_num
  - Number of times duplicate address detection is done.
- INTFID interface_id
  - 64-bit interface identifier in colon-hexadecimal format.
  - Interface ID is either manually defined by this parameter or TCP/IP builds the Interface ID using information from the OSA.
  - Interface ID is used to form the link-local address for the interface, and is also appended to any prefixes for the interface (either manually configured (by IPADDR or ADDADDR), or learned over the interface by router advertisements) to form complete IPv6 addresses.

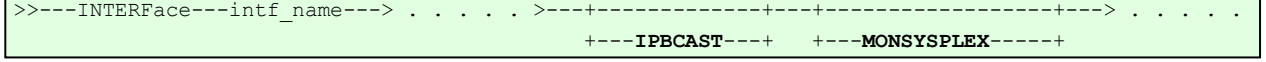
- TEMPPREFIX ALL/NONE/prefix/prefix_length (Default is ALL)
  - ALL causes temporary addresses to be generated for all prefixes learned over this interface by router advertisements.
  - NONE causes no IPv6 temporary addresses to be generated.
  - prefix/prefix_length specifies the set of prefixes for which temporary IPv6 addresses will be generated.
  - Temporary addresses are generated only on an interface that is enabled for stateless address autoconfiguration.
    - Stateless address autoconfiguration is enabled for an interface if no address or prefix is specified with the IPADDR keyword.
  - Temporary addresses are generated only when the TEMPADDRS keyword is specified on the IPCONFIG6 statement.
  - The job name of an application must be in the SRCIP statement block with a value of TEMPADDRS to cause a temporary IPv6 address to be preferred over a public IPv6 address as the source IP address for the application; otherwise, the default source address selection algorithm prefers public IPv6 addresses over temporary addresses.
  - Temporary prefixes may be added using ADDTEMPPREFIX and deleted using DELTEMPPREFIX.

Broadcasts, and Sysplex Monitor

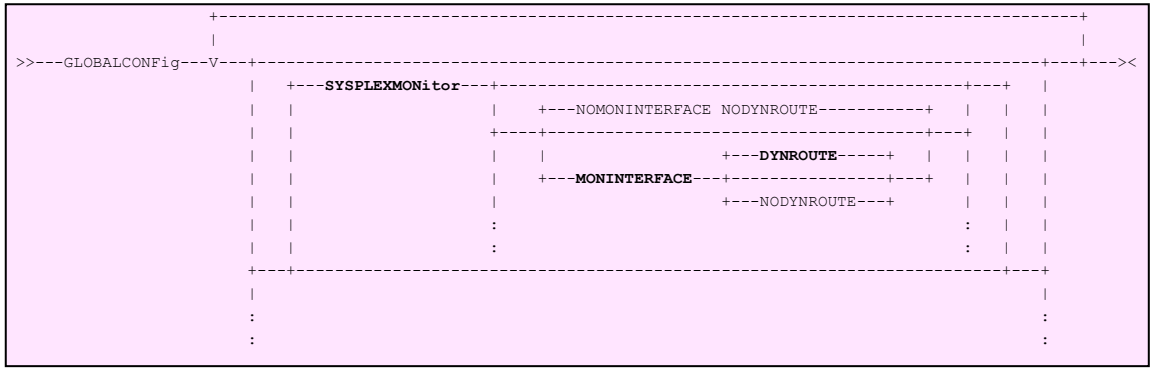
- LCS (non-QDIO OSA OSE mode) and MPCIPA (QDIO OSA OSD mode) LINK



- IPAQENET (IPv4) and IPAQENET6 (IPv6) INTERFACE (QDIO only)



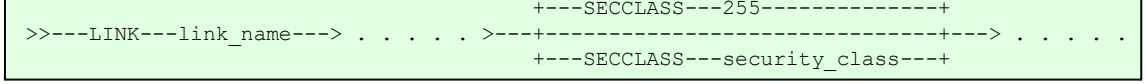
- IPBCAST
  - Enables IP broadcasts over this link. Without IPBCAST no IP broadcast will be passed over this link.



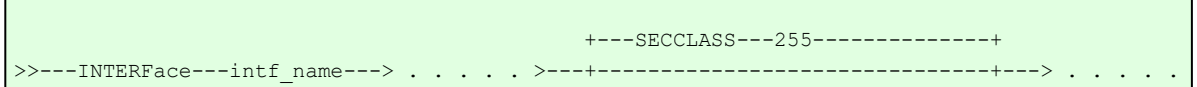
- MONSYSPLEX
  - Specifies whether or not sysplex autonomics should monitor the link's status.
    - MONINTERFACE is required on GLOBALCONFIG SYSPLEXMONITOR statement.
  - Dynamic routes over this link may be monitored.
    - Requires MONSYSPLEX and DYNROUTE on the GLOBALCONFIG SYSPLEXMONITOR statement.
  - NOMONSYSPLEX is the default.
  - See VIPA presentation for more information about Sysplex Autonomics:
    - http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS789

Security Class Parameter

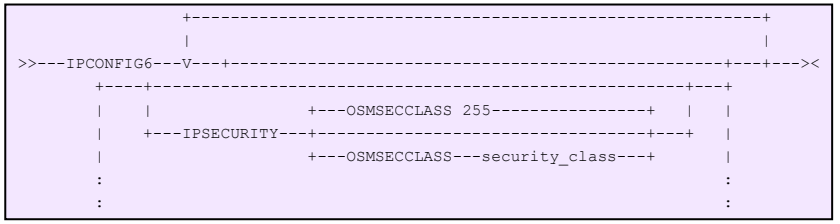
- LCS (non-QDIO OSA OSE mode) and MPCIPA (QDIO OSA OSD mode) LINK



- IPAQENET (IPv4) and IPAQENET6 (IPv6) INTERFACE (QDIO only)



If you have a zBX and you enable IPsec for IPv6, you can configure a security class for IP filtering that applies to all OSM interfaces using the OSMSECCLASS parameter on the IPCONFIG6 statement.

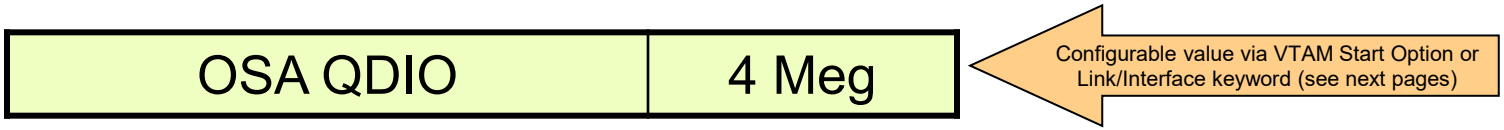


- SECCLASS security_class
  - Used for Multi-Level Security.
  - Security class for IP filtering with this interface.
  - The matching policy action is applied when the SECCLASS parameter matches the SecurityClass parameter defined on the policy IPsec condition IpService statement.
  - TCP/IP stack ignores this value if IPSECURITY is not specified on the IPCONFIG or IPCONFIG6 statement.

OSA Read Storage Usage

QDIO Read Storage

- Amount of storage for read processing:



- 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

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

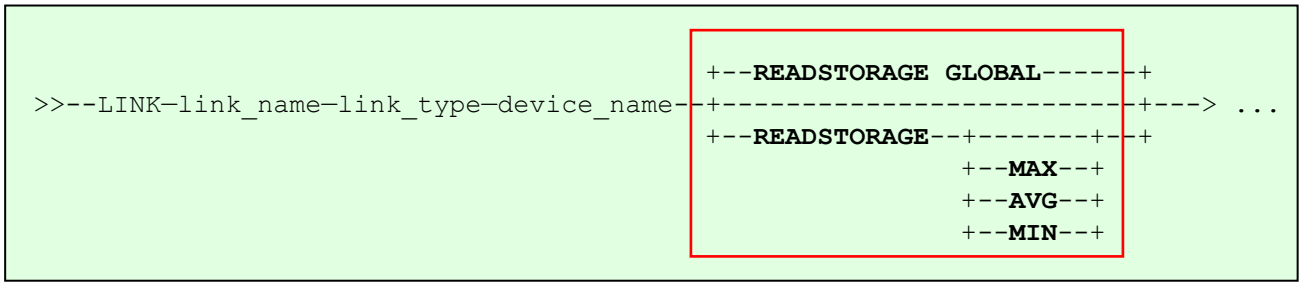
```
        +--QDIOSTG=-MAX---------+
>>------+-----------------------+---><
        +--QDIOSTG=--+--MAX--+--+
                     +--AVG--+
                     +--MIN--+
                     +--nnn--+
```

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

Default

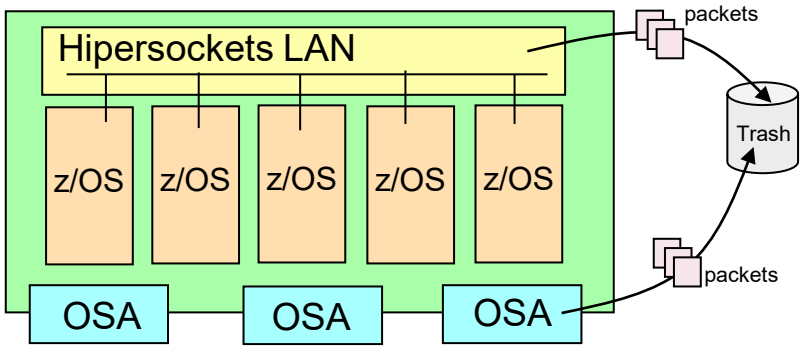
- Storage units are defined in terms of QDIO SBALs (QDIO read buffers)
  - nnn is the exact number of SBALs in the range 8-126
  - MAX allows for the best performance (for example, throughput), but requires more storage.
  - MIN may be used for devices with lighter workloads or where system storage might 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.

Read Storage Parameter



- READSTORAGE
  - Defines the amount fixed storage for read processing.
  - READSTORAGE must match between LINK and INTERFACE for the same OSA.
- Overrides VTAM Start option QDIOSTG for a specific QDIO device.
- Global causes the QDIOSTG 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.

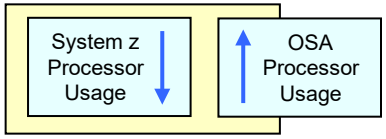
Storage Shortage



- New in z/OS V1.11
  - Amounts of ECSA storage buffered on HiperSockets and OSA-Express inbound data paths are monitored.
  - When ECSA storage usage is excessive or when ECSA limits become constrained or critical, packets might be intentionally discarded to minimize impact to system performance and system integrity.

OSA Offload Support

ARP, Checksum, and Segmentation Offload



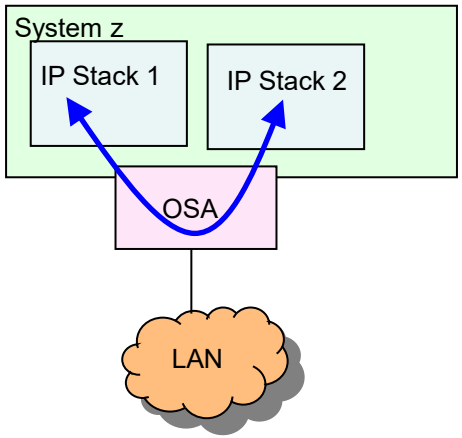

```
                +-----------------------------------------+
                |                                         |
>>---IPCONFig---V---+-----------------------------------+---+---><
                    |    +---CHECKSUMOFFLoad-----+       |
                    +----+-----------------------+-------+
                    |    +---NOCHECKSUMOFFLoad---+       |
                    |    +---NOSEGMENTATIONOFFLoad---+   |
                    +----+---------------------------+---+
                    |    +---SEGMENTATIONOFFLoad-----+   |
                    :                                    :
                    :                                    :
```

```
                 +----------------------------------------+
                 |                                        |
>>---IPCONFIG6---V---+-----------------------------------+---+---><
                     |   +---CHECKSUMOFFLoad-----+        |
                     +---+-----------------------+--------+
                     |   +---NOCHECKSUMOFFLoad---+        |
                     |    +---NOSEGMENTATIONOFFLoad---+   |
                     +----+---------------------------+---+
                     |    +---SEGMENTATIONOFFLoad-----+   |
                     :                                    :
                     :                                    :
```

- The following OSA Offload Support avoids CPU processing.
- IPv4 ARP Offload
  - OSA automatically offloads ARP processing.
  - Use the NETSTAT command to display the OSA ARP cache.
  - Use the vary command to flush the ARP cache on the OSA.
    - **Vary TCPIP,proc_name,PURGECache,intf_name**
- Checksum Offload
  - A checksum of the packet data is calculated and sent with the packet to provide integrity of the data.
  - OSA Checksum Offload processing is controlled by the IPConfig and IPConfig6 statements.
- Segmentation Offload
  - Also known as Large Send for TCP/IP traffic, Segmentation Offload allows larger amounts of data to be sent by the TCP/IP stack because the OSA provides the segmentation of that data.
  - OSA Segmentation Offload processing is controlled by the IPConfig and IPConfig6 statements.
  - Always check for the latest PSP bucket and OSA driver levels.

OSA Interface Isolation

Routing for Shared OSA

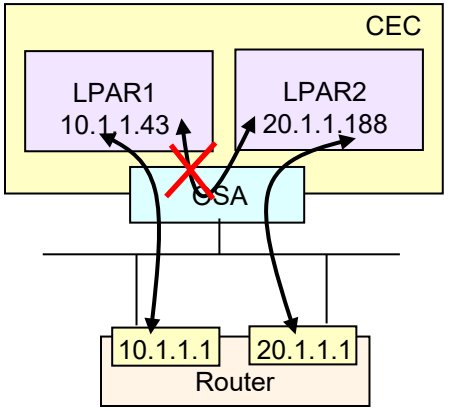


- All IP addresses in HOME list are added to OSA Address Table (OAT)
- When a packet is sent from one of the systems sharing the OSA and the destination is an IP address in the OAT, the packet is sent directly to the destination without going out onto the LAN.

Disable LPAR to LPAR Traffic via Shared OSA

```
•       IPAQENET (IPv4) and IPAQENET6 (IPv6) INTERFACE (QDIO only)

                                                          +---NOISOLATE---+
>>---INTERFace---intf_name---DEFINE---+---IPAQENET----+---+---------------+---> . . . . .
                                      +---IPAQENET6---+   +---ISOLATE-----+
```

- z/OS V1.11 and later
  - ISOLATE/NOISOLATE option on QDIO network interface definition.
  - Only OSA local routing, without flowing out onto the LAN, is disabled.
  - LPAR to LPAR traffic may still flow over the OSA if it is sent out onto the LAN to a router and then back in over the same OSA.
- OSA local routing can in some scenarios be seen as a security exposure.
- Requires proper OSA MCL.



Be careful using ISOLATE if using OSPF and share a subnet between TCP/IP stacks that share the OSA.

Primary Router and Layer 3 Virtual MAC (VMAC)

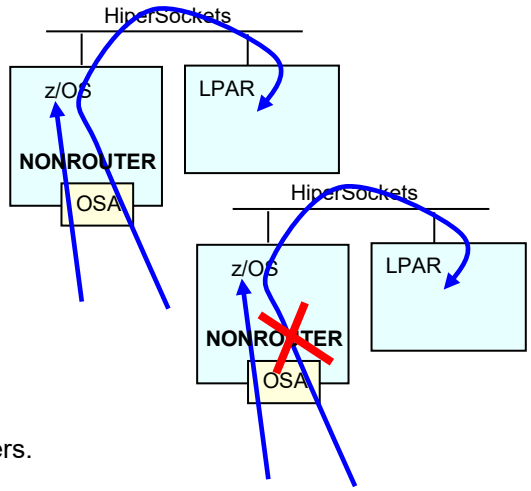
PRIRouter, SECRouter, NONRouter

- LCS (non-QDIO OSA OSE mode) and MPCIPA (QDIO OSA OSD mode) DEVICE

```
                                                          +---NONRouter-----+
>>---DEVICE---device_name---+---LCS------+---device_number---+-----------------+---> . . . . .
                            +---MPCIPA---+                   +---PRIRouter-----+
                                                             +---SECRouter-----+
```

- IPAQENET (IPv4) and IPAQENET6 (IPv6) INTERFACE (QDIO only)

```
                                                       +---NONRouter---+
>>---INTERFace---intf_name---DEFINE---+---IPAQENET----+---+---------------+---> . . . . .
                                      +---IPAQENET6---+   +---PRIRouter---+
                                                          +---SECRouter---+
```



- All IP addresses in a TCP/IP HOME list are registered (dynamically downloaded) with the QDIO adapters.
  - HOME changes are automatically sent to QDIO adapters.
- If the OSA receives any packets with its MAC as the destination and a destination IP address that is "unknown" (meaning not an IP address in the HOME list), then OSA does the following:
  - If PRIRouter is defined (assuming the OSA is started to that stack) then all "unknown" packets are sent to the PRIRouter stack.
  - If PRIRouter is not defined (or the OSA is not started to any stack with PRIRouter)(could be that PRIRouter is coded but that OSA connection is down due to failure or other outage) then if SECRouter is coded all "unknown" packets are sent to the SECRouter stack. If multiple SECRouters then a random (unpredictable) stack with SECRouter coded will be sent the "unknown" packets.
    - There is no way to set the order of precedence for the secondary routers.
  - If only NONRouter is defined (or any PRIRouter and SECRouter connection are down due to failure or other outage) then all "unknown" packets are discarded by the OSA. NONROUTER is the default.
- Non-QDIO OSA (OSE mode) may define PRIROUTER and SECROUTER via OSA/SF.
- IPCONFIG DATAGRAMFWD
  - PRIRouter is used when traffic is routed through the stack to another stack. Keep in mind that if one stack is used to route to other stacks IPCONFIG DATAGRAMFWD is required.
- PriRouter, SecRouter, NonRouter definition is ignored when VMAC parameter is defined.
  - **Recommendation: Use VMAC for shared OSA ports rather than PRIROUTER/SECROUTER.**

VMAC (Virtual MAC)

- LCS (non-QDIO OSA OSE mode) and MPCIPA (QDIO OSA OSD mode) LINK

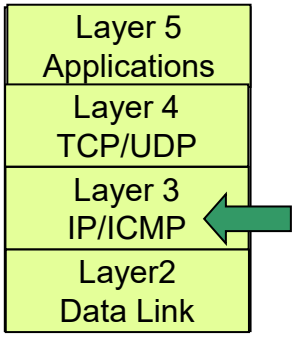
```
>>---LINK---link_name---> . . . . . >---+-----------------------------------------------+---> . . . . .
                                        |                          +---ROUTEALL---+     |
                                        +---VMAC---+-------------+---+--------------+---+
                                                   +---macaddr---+   +---ROUTELCL---+
```

VMAC Requires
Hardware: System z9 or later
Software: z/OS V1.8 (PK36947) or later

- IPAQENET (IPv4) and IPAQENET6 (IPv6) INTERFACE (QDIO only)

```
                                                  +---VMAC---ROUTEALL-----------------------------+
>>---INTERFace---intf_name---DEFINE---+---IPAQENET----+---+-----------------------------------------------+---> . . . . .
                                     +---IPAQENET6---+   |                          +---ROUTEALL---+     |
                                                         +---VMAC---+-------------+---+--------------+---+
                                                                    +---macaddr---+   +---ROUTELCL---+
```

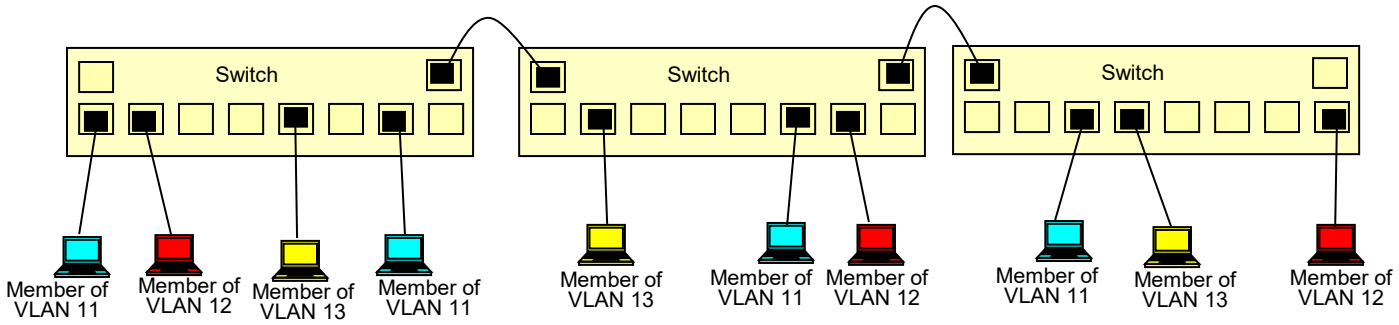
- VMAC mac_addr ROUTEALL/ROUTELCL
  - Indicates virtual MAC address. OSA uses this address rather than the physical MAC address for all IP packets (Layer 3) to and from this TCP/IP stack.
  - If mac_addr is not coded, then the OSA generates a virtual MAC address.
    - Unless the virtual MAC address must remain the same even after TCP/IP restart, configure VMAC without mac_addr.
  - NONROUTER, PRIROUTER, and SECROUTER are ignored for an OSA if the VMAC parameter is configured.
    - Recommendation: Use VMAC for shared OSA ports rather than PRIROUTER/SECROUTER.
  - ROUTEALL causes all IP traffic destined to the virtual MAC to be forwarded to the TCP/IP stack. The default.
  - ROUTELCL causes only the traffic destined to the virtual MAC and whose destination IP address is registered to the OSA (all active IP addresses on the TCP/IP stack), to be forwarded to the TCP/IP stack.
  - VMAC is required to define multiple VLAN IDs for IPv4 or IPv6, from a single stack for a given OSA port.



OSA Layer 2 is supported by z/VM and zLinux but is not supported by z/OS.

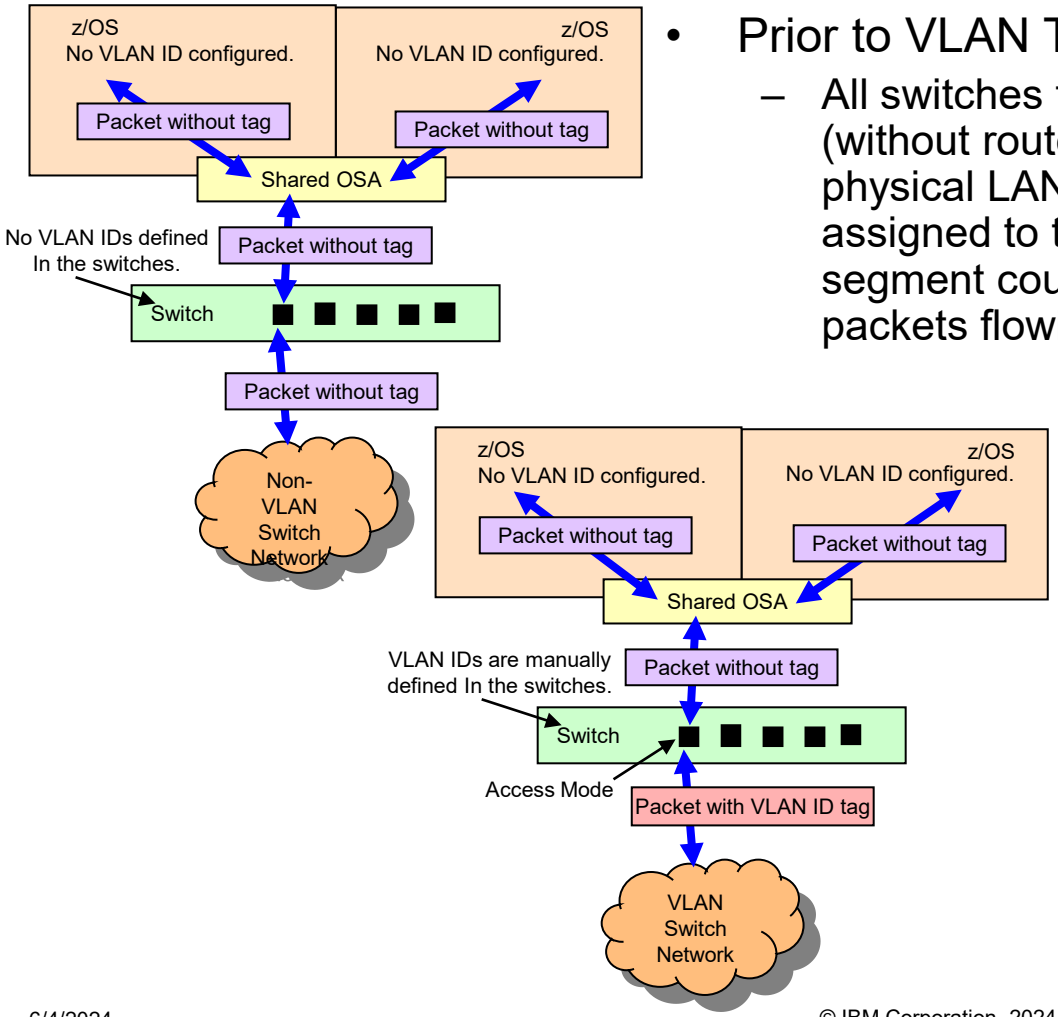
OSA VLAN Support

What is a VLAN?



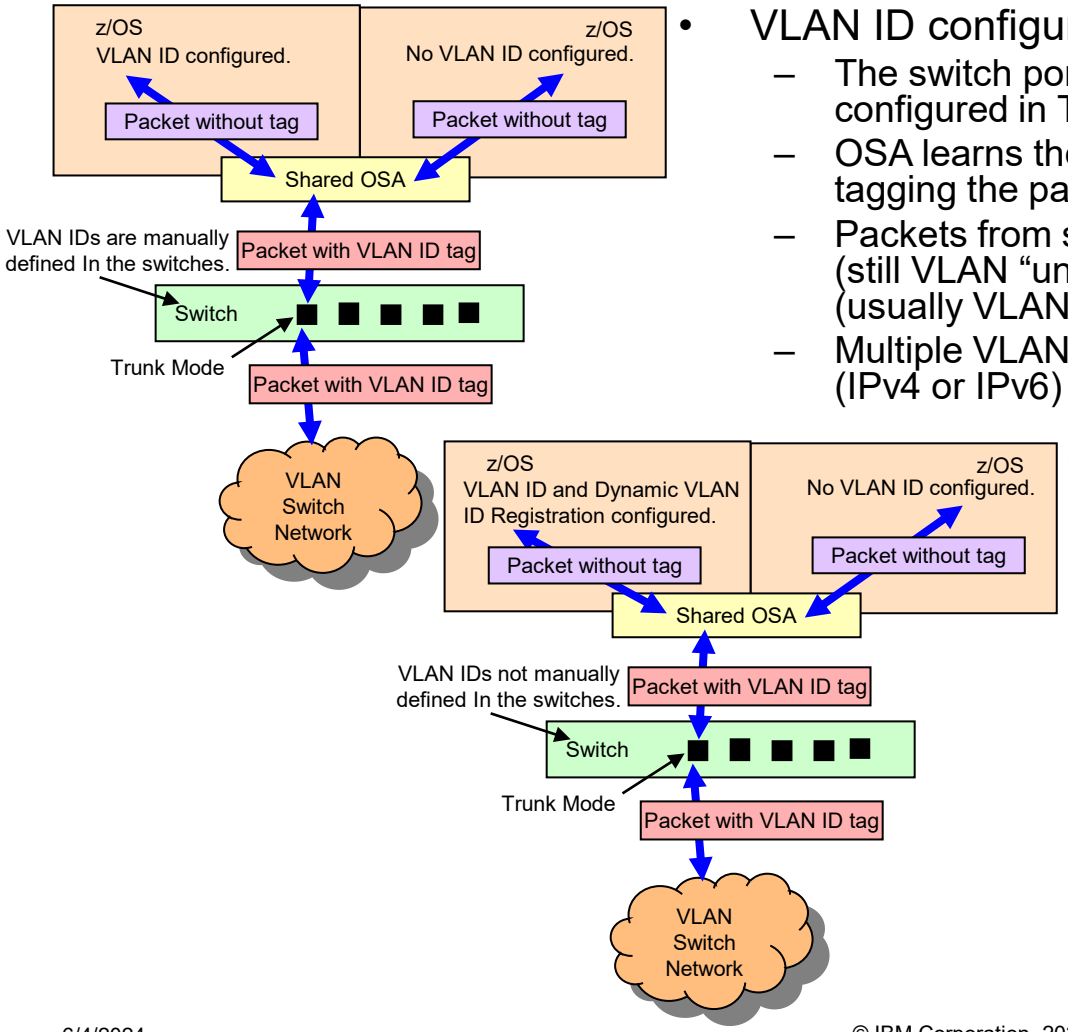
- A VLAN is a switched network that is logically segmented on an organizational basis, by functions, project teams, or applications rather than on a physical or geographical basis.
- Reconfiguration of the network can be done through software rather than by physically unplugging and moving devices or wires.
- A VLAN can be thought of as a broadcast domain that exists within a defined set of switches.
- A VLAN consists of a number of end systems, either hosts or network equipment (such as bridges and routers), connected by a single bridging domain.
- VLANs are created to provide the segmentation services traditionally provided by routers in LAN configurations.
- None of the switches within the defined group will bridge any frames, not even broadcast frames, between two VLANs.
  - IP Router is needed to communicate between VLANs.

When z/OS is VLAN “un-aware”



- Prior to VLAN Technology
  - All switches that were attached together (without routers in between them) formed one physical LAN segment with one IP subnet assigned to them. All devices on a LAN segment could potentially have access to all the packets flowing on the segment.
- VLANs Defined on Switches
  - z/OS may still be VLAN “un-aware”.
  - The switch port that OSA attaches to should be configured in Access Mode with a certain VLAN ID assigned.
  - The switch itself manages the VLAN ID tagging of the packets.

When z/OS is VLAN “aware”



- VLAN ID configured on LINK/INTERFACE
  - The switch port that OSA attaches to must be configured in Trunk Mode.
  - OSA learns the VLAN ID from the stack and manages tagging the packets with the appropriate VLAN ID.
  - Packets from stacks that do not configure a VLAN ID (still VLAN “un-aware”) are part of the default VLAN ID (usually VLAN ID 1).
  - Multiple VLAN IDs per stack/OSA port per IP version (IPv4 or IPv6) requires z/OS V1.10 and VMACs.
- VLAN ID and Dynamic VLAN ID Registration Defined on LINK/INTERFACE
  - Rather than manually configure the supported VLAN IDs per switch port, the switch learns the VLAN IDs for the port from the OSA.

QDIO VLAN Support

- LCS (non-QDIO OSA OSE mode) and MPCIPA (QDIO OSA OSD mode) LINK

```
                                                    +---NODYNVLANREG---+
>>---LINK---link_name---> . . . . . >---+-----------------+---+------------------+---> . . . . .
                                        +---VLANID---id---+   +---DYNVLANREG-----+
```

DYNVLANREG Requires Software: z/OS V1.8 or later

- IPAQENET (IPv4) and IPAQENET6 (IPv6) INTERFACE (QDIO only)

```
                                                        +---NODYNVLANREG---+
>>---INTERFace---intf_name---> . . . . . >---+-----------------+---+------------------+---> . . . . .
                                             +---VLANID---id---+   +---DYNVLANREG-----+
```

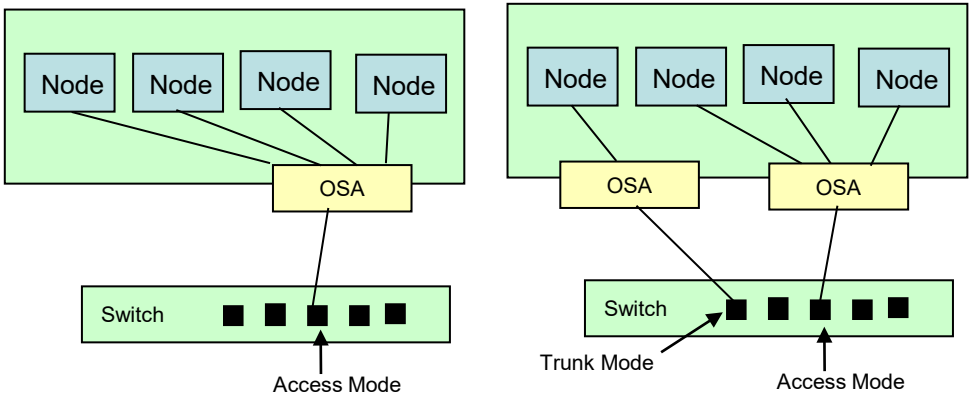
VLANID Requires Software: z/OS V1.5 or later.
Prior to z/OS V1.10:
Limited to one VLAN ID per IPv4 or IPv6 per stack/OSA port.
Other stacks may define different VLANIDs for same port.
z/OS V1.10 and later:
Multiple VLAN IDs per IP version per stack/OSA port (Interface Only – not supported on Link)
-Maximum of 8 VLAN IDs per IP version (IPv4 or IPv6) per OSA port per stack.
-Different VMACs are required.

- VLANID id_number
  - Specifies the VLAN ID tag for this link.
- DYNVLANREG/NODYNVLANREG
  - Dynamic registration of VLAN ID (GVRP).
    - Dynamic registration of VLAN IDs is handled by OSA and switch. Both must be at a level with the hardware support for dynamic VLAN ID registration.
  - DYNVLANREG specifies that if a VLAN ID is configured for this link, it is dynamically registered with physical switches on corresponding LAN.
    - This parameter is only applicable if a VLAN ID is specified.
  - NODYNVLANREG specifies that if VLAN ID is configured, it must be manually registered with switches on corresponding LAN. This is the default.
  - DYNVLANREG must match between LINK and INTERFACE for the same OSA.
- **VMAC is required to define multiple VLAN IDs for IPv4 or IPv6, from a single stack for a given OSA port.**

z/OS Support of VLAN IDs

- z/OS TCP/IP supports configuring the VLAN ID to be used on OSA connections.
  - z/OS may configure the VLAN ID but it is OSA that adds/removes the VLAN ID tag to the packets.
  - Conforms to the IEEE 802.1Q standard
- A Switch may configure a port in Trunk mode or Access mode.
  - Trunk mode
    - VLAN ID is defined by the end device, either configured on z/OS or defaulted by the OSA.
    - Requires VLAN ID tagged packets.
  - Access mode
    - VLAN ID is controlled by the switch rather than the end device. Any VLAN ID configured by z/OS is ignored.
- z/OS VLAN Rules:
  1. An OSA should either be:
     - Attached to a switch port in trunk mode if any of the stacks that share the OSA have a VLAN ID configured, or
     - Attached to a switch port in access mode and each stack that shares the OSA should not have a VLAN ID configured.
  2. As with any IP network, separate VLANs should be treated like separate physical networks and have separate subnets assigned.
  3. Some switch vendors use VLAN ID 1 as the default value when a VLAN ID value is not explicitly configured. It is recommended that you avoid the value of 1 when configuring a VLAN ID value.
  4. When a TCP/IP stack has access to multiple OSA ports that are on the same physical LAN, and a VLAN ID is configured on any of the OSA ports, it is recommended that this stack configure a VLAN ID for all OSA ports on the same physical LAN. Do not mix VLAN and no-VLAN on the same physical network accessed by a single stack through multiple OSA ports.
  5. When multiple INTERFACE statements are defined on a single stack for a single OSA port and a single IP version (IPv4 or IPv6), the VLAN IDs must be unique, and the INTERFACE definition will be rejected if the VLAN ID is omitted.
     - The VLAN ID, VMAC, and IP subnet values must be unique per IP version (IPv4 or IPv6) for multiple INTERFACE statements for a single OSA port defined on a single TCP/IP stack.
     - For parallel interfaces into the same IP subnet/VLAN ID from a single TCP/IP stack, multiple OSA ports are required.
  6. The requirement for a unique VLAN ID per INTERFACE statement rule only applies within a single stack. Each stack on a shared OSA port is completely independent of other stacks sharing the OSA port. Multiple stacks may define the same VLAN ID or different VLAN IDs for the same shared OSA port.

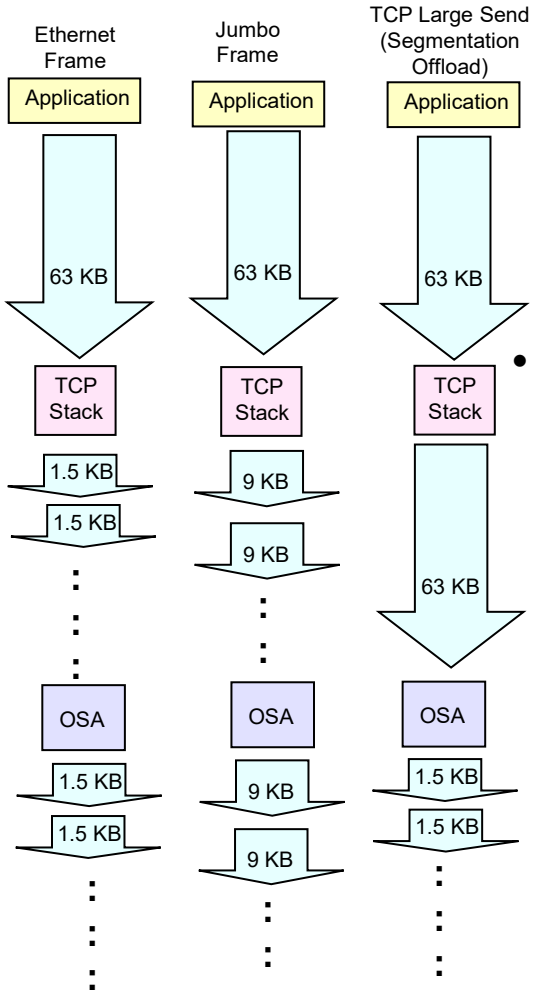
VLAN Migration



- Migration of z/OS VLAN “unaware” to z/OS VLAN “aware”
- Switch port defined in Access Mode
  - Operating Systems should define OSA without VLAN (VLAN “unaware”)
- Switch port defined in Trunk Mode
  - Operating Systems should define OSA with VLAN (VLAN “aware”)
- An OSA port attaches to a Switch in either Access Mode or Trunk Mode. If multiple LPARs share an OSA port attached to a switch in Access Mode and one of those LPARs wants to start to use VLAN configuration either:
  - Use a second OSA port in Trunk Mode.
  - Or change the Switch port from Access Mode to Trunk Mode. All traffic to/from the LPARs that do not define VLAN will be sent using the Default VLAN ID.
    - If the Default VLAN ID uses a different subnet, then all the LPARs IP addresses will have to change.

MTU and Jumbo Frames

MTU, Jumbo Frames, and MTU Discovery

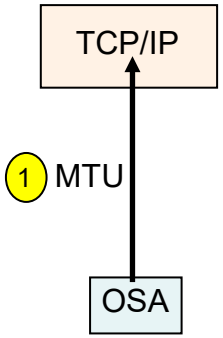



```
>>---INTERFace---intf_name---DEFINE---+---IPAQENET----+---+-------------------+---> . . . . .
                                      +---IPAQENET6---+   +---MTU---mtu_num---+
```

```
                +-------------------------------------------------------------------+
                |                                                                   |
>>---IPCONFig---V---+-------------------------------------------------------------+---+--->< 
                    |   +---NOPATHMTUDISCovery---+                                |
                    +---+----------------------+---------------------------------+|
                    |   +---PATHMTUDISCovery-----+                                |
                    :                                                             :
                    :                                                             :
```

- MTU mtu_num
  - Specifies the maximum transmission unit (MTU) in bytes.
  - IPCONFig PATHMTUDISCovery may be defined to dynamically discover the path MTU (PMTU), which is the smallest MTU of all the hops in the path. Use this parameter to prevent fragmentation of datagrams.
    - Uses ICMP “fragmentation-needed” errors to detect the PMTU for a path. ICMP errors must be permitted to flow at all hosts along the path of a connection. PATHMTUDISCOVERY does not function if a firewall blocks ICMP errors.
  - When defining Jumbo frames, MTU 8992, **PathMTUDiscovery is recommended**.
  - It is recommended to use the same MTU size on all hosts on the same subnet because there is no router in the path to fragment packets.

Lowest MTU



2
```
>>---INTERFace---intf_name---DEFINE---+---IPAQENET----+---+-------------------+---> . . . . .
                                      +---IPAQENET6---+   +---MTU---mtu_num---+
```

3
```
>>---OSPF_Interface--- . . . . . ---MTU---mtu_num---> . . . . .

>>---RIP_Interface--- . . . . . ---MTU---mtu_num---> . . . . .

>>---Interface--- . . . . . ---MTU---mtu_num---> . . . . .
```

4
```
>>---ROUTE--- . . . . . ---MTU---mtu_num---> . . . . .
```

Lowest MTU is used!

1. The OSA hardware will notify the TCP/IP stack of the hardware MTU.
2. MTU may be defined on the Interface statement if a lower MTU than the hardware MTU is desired.
   - If MTU is not defined on the Interface, then 2 = 1
3. MTU may be defined on the OMPROUTE definition for the device.
   - If the MTU is not defined on the OMPROUTE definition for the device, the default MTU of 576 is used.
     - Note: Always define all devices in OMPROUTE or use the IGNORE_UNDEFINED_INTERFACE to prevent the default MTU of 576 to be used.
   - The lowest of the three values, 1, 2, or 3, will be used as the MTU by OMPROUTE.
4. MTU may be defined on the ROUTE definition in the BeginRoutes block.
   - If the MTU is not defined on the ROUTE definition, the default MTU of 576 is used.
   - The lowest of the three values, 1, 2, or 4 will be used as the MTU by the routing table.

Latency and Workload Queuing

Inbound Blocking and Inbound Workload Queuing

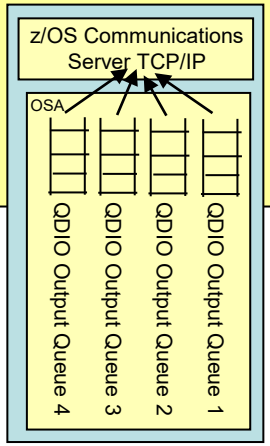
```
•      LCS (non-QDIO OSA OSE mode) and MPCIPA (QDIO OSA OSD mode)  LINK

>>---LINK---link_name---+---ETHERNet-------+---link_num---device_name--->
                        +---802.3----------+
                        +---ETHEROR802.3---+
                        +---IPAQENET-------+

    +---INBPERF---BALANCED-------------+
>---+------------------------------------+---> . . . . .
    +---INBPERF---+---DYNAMIC------+---+
                  +---MINCPU-------+
                  +---MINLATENCY---+
```

Inbound Workload Queuing (IWQ) INBPERF DYNAMIC WORKLOADQ

IWQ automatically provides unique input queues for:
- -Sysplex Distributor traffic
- -Bulk data (streaming) traffic
- -Enterprise Extender (EE) traffic
- -Default (Interactive)

Requires z196+ and z/OS V1.13+

Prevents inbound and outbound out of order packets, and the overhead that goes with it.




```
                                                +---INBPERF---BALANCED-----------------+
                                                +---INBPERF---DYNAMIC---WORKLOADQ------+
>>---INTERFace---intf_name---DEFINE---+---IPAQENET----+---+-------------------------------------+---> . . . . .
                                     +---IPAQENET6---+    +---INBPERF---DYNAMIC---NOWORKLOADQ---+
                                                +---INBPERF---MINCPU-------------------+
                                                +---INBPERF---MINLATENCY---------------+
```

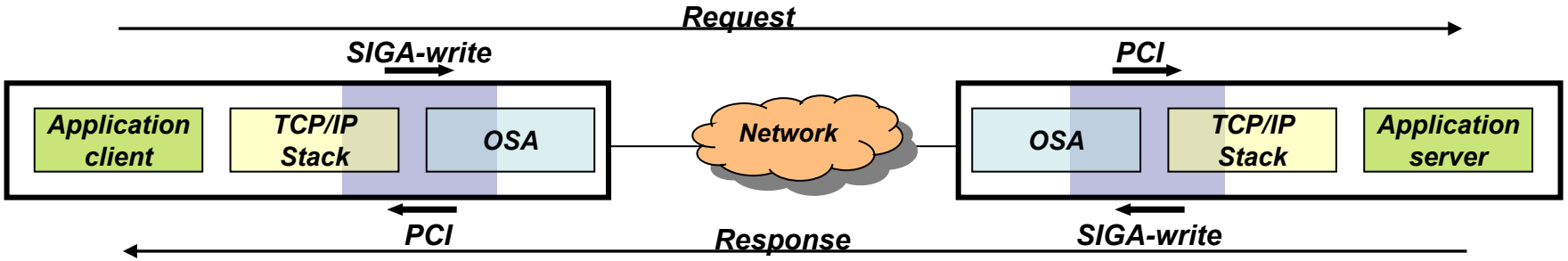
- INBPERF
  - Indicates how frequently the adapter should interrupt the host for inbound traffic.
  - 3 Static Settings
    - MINCPU minimizes host interrupts without regard to throughput.
    - MINLATENCY minimizes delay, by more quickly passing packets to the host.
    - BALANCED achieves high throughput and low CPU consumption.
  - 1 Dynamic Setting (z/OS V1.9+, PTFed back to V1.8)
    - DYNAMIC reacts to changes in inbound traffic patterns and sets interrupt-timing values to where throughput is maximized.
    - **DYNAMIC should outperform the other settings for most workload combinations.**
    - See 2098DEVICE Preventive Service Planning (PSP) buckets for hardware support.
    - DYNAMIC WORKLOADQ provides different queues for inbound traffic.
  - INBPERF must match between LINK and INTERFACE for the same OSA.

Dynamic LAN Idle Support

LINK . . . INBPERF DYNAMIC

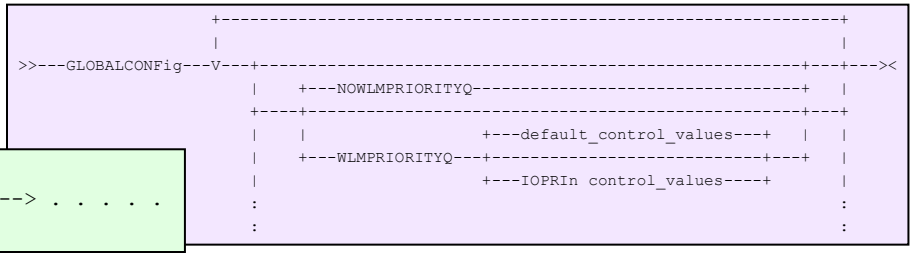
or INTERFACE . . . INBPERF DYNAMIC NOWORKLOADQ

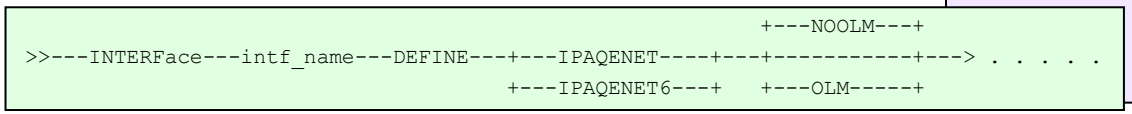
Optimized Latency Mode (OLM)



- Requires z/OS V1.11+ and OSA-E3+
- Define OLM when,
  - Latency is the most critical factor (ie. More important than CPU overhead).
  - Traffic is not streaming bulk data (ie. FTP).
- Inbound
  - OSA signals the host if data is “on its way” (“Early Interrupt”).
  - Host looks more frequently for data from OSA.
- Outbound
  - OSA does not wait for SIGA to look for outbound data (“SIGA reduction”).
- PTFs for z/OS V1.11
  - PK90205 (PTF UK49041) and OA29634 (UA49172)

OLM Configuration

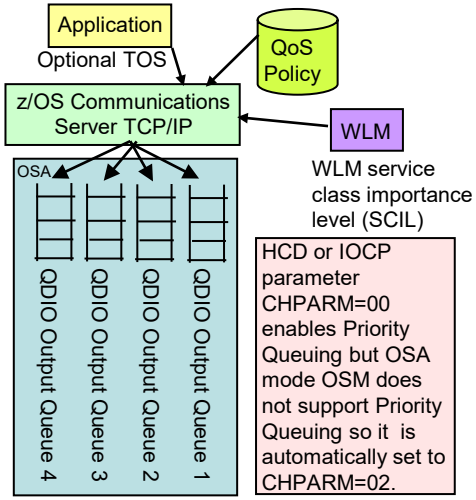




- OLM is specified on the Interface statement.
  - GLOBALCONFIG WLMPRIORITYQ and QoS configuration statement SETSUBNETPRIOTOSMASK may be necessary to benefit from OLM.
    - OLM will not change traffic patterns if all the traffic is being sent to the fourth queue.
- **Restrictions:**
  - **OLM is rarely desired because it is only recommended when Latency is the most critical factor (ie. More important than CPU overhead, etc.)**
  - **When OLM is defined QDIO Accelerator (or HiperSockets Accelerator) will not accelerate the traffic.**
  - **When OLM is specified INBPERF is automatically set to DYNAMIC.**
  - **Interfaces sharing an OSA port using OLM is limited to four.**
    - Each Interface statement counts toward the 4 Interface limit:
      - LPAR TCP/IP stack using the OSA port
      - VLAN defined for this OSA port
      - Protocol (IPv4 or IPv6) interface defined for this OSA port
      - TCP/IP stack on the same LPAR using the OSA port
      - TCP/IP stack activating the OSA-E Network Traffic Analyzer (OSAENTA)

Outbound Priority Routing Queues

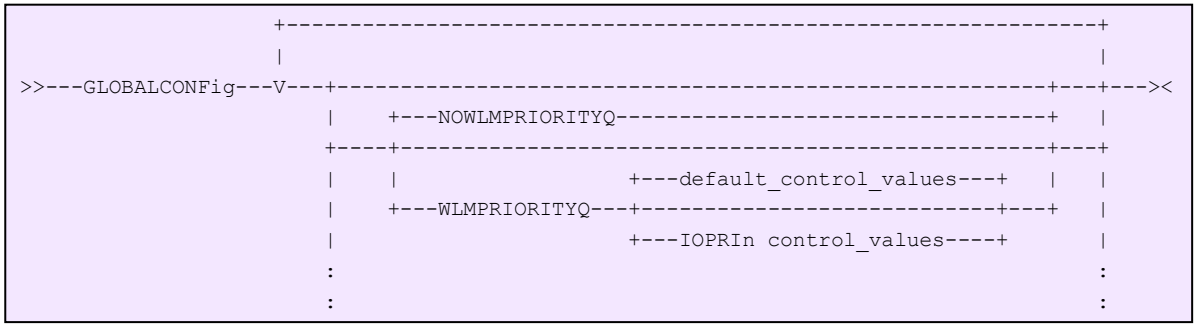
- The Type of Service (ToS) byte in the IP header may be used by routers in the IP network to prioritize traffic (forward some types of traffic before others).
  - The most benefit is realized when the routers are all configured for this support.
- TCP/IP uses the first three bits of the ToS byte in the IP header to determine the outbound priority value for a given datagram.
  - Optionally an application can specify the TOS for its traffic.
- z/OS CS TCP/IP supports four priority values in the range 1–4 for outbound QDIO traffic (with 1 being the highest priority).
  - TCP/IP will send packets using these four queues whether or not any routers in the network are configured to use the ToS settings.
- z/OS CS TCP/IP Policy Agent Quality of Service (QoS) may be used to override the default mapping of ToS values to priorities.
  - This may be used for devices without VLANs.
    - SetSubnetPrioTosMask statement
  - This may be used for devices with VLANs.
    - PriorityTosMapping parameter on the SetSubnetPrioTosMask statement may define VLAN priority-tagging.
- Enterprise Extender (EE) (SNA encapsulation over IP) automatically configures IP ToS.



Default mapping of ToS values to priorities:

ToS	Priority
000	4
001	4
010	3
011	2
100	1
101	1
110	1
111	1

WLM Service Class



- WLM IO Priority Enhancement (z/OS V1.11+)
  - When the GLOBALCONFIG WLMPRIORITYQ parameter is specified and a packet with a ToS or traffic class value 0 is sent over QDIO OSA port, TCP/IP sets the QDIO write priority of the packet based on the priority value provided by the WLM service class.

WLM Service classes	TCP/IP assigned control value	Default QDIO queue mapping
SYSTEM	n/a	Always queue 1
SYSSTC	0	Queue 1
User-defined with IL 1	1	Queue 2
User-defined with IL 2	2	Queue 3
User-defined with IL 3	3	Queue 3
User-defined with IL 4	4	Queue 4
User-defined with IL 5	5	Queue 4
User-defined with discretionary goal	6	Queue 4

Which QDIO Queues are Used?

From Display tcpip,,n,devlinks:

```
DEVNAME: NSQDIO1              DEVTYPE: MPCIPA
   DEVSTATUS: READY
   LNKNAME: LNSQDIO1          LNKTYPE: IPAQENET   LNKSTATUS: READY
     SPEED: 0000001000
```

Match TCP/IP DEVNAME with PORTNAME in your TRLE VTAM definitions

From VTAMLST MACLIB:

```
NSQDIO11 TRLE  LNCTL=MPC,                                        *
              MPCLEVEL=QDIO,                                     *
              READ=(0E28),                                       *
              WRITE=(0E29),                                      *
              DATAPATH=(0E2A,0E2B),                              *
              PORTNAME=(NSQDIO1,0)
```

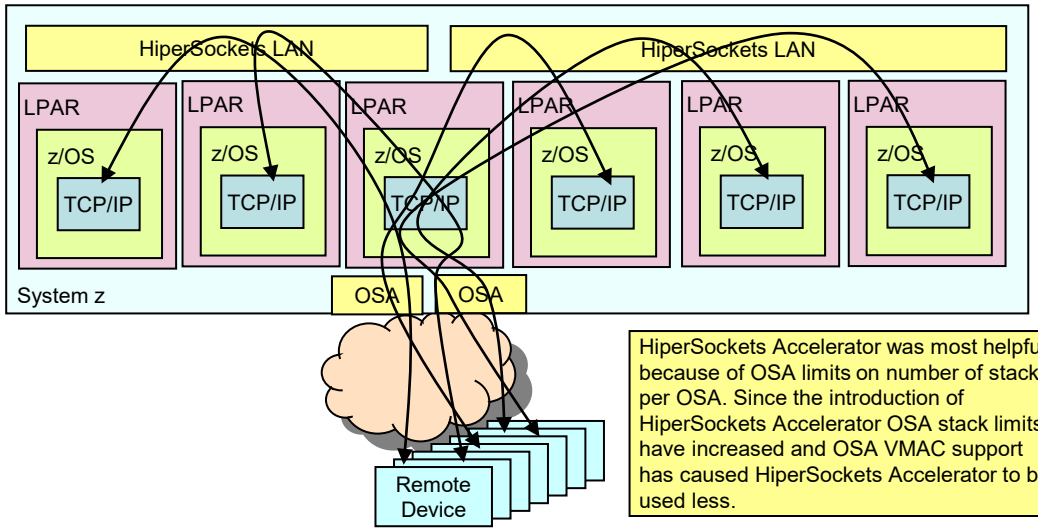
This is your TRLE name

```
d net,trl,trle=NSQDIO11
  .
 IST1802I P1 CURRENT = 25 AVERAGE = 51 MAXIMUM = 116
 IST1802I P2 CURRENT = 0 AVERAGE = 0 MAXIMUM = 0
 IST1802I P3 CURRENT = 0 AVERAGE = 0 MAXIMUM = 0
 IST1802I P4 CURRENT = 0 AVERAGE = 0 MAXIMUM = 0
```

VTAM TNSTATS details the four queues as well.

QDIO Accelerator

HiperSockets Accelerator



HiperSockets Accelerator was most helpful because of OSA limits on number of stacks per OSA. Since the introduction of HiperSockets Accelerator OSA stack limits have increased and OSA VMAC support has caused HiperSockets Accelerator to be used less.

- Accelerator stack TCP/IP Profile coding:
  - IPCONFIG IQDIORouting QDIOPriority n
    - IQDIORouting enables HiperSockets Accelerator
    - QDIOPriority may be coded to define an OSA QDIO outbound queue between 1 and 4.
      - The default is 1, which is the high priority queue.
  - IPCONFIG PATHMTUDiscovery
    - May be desired because IP Fragmentation is not supported (VTAM Device Drivers do not fragment)
  - IPCONFIG DATAGRAMFWD
    - Required because the first packet will use the IP routing.
  - PRIROUTER or VMAC
    - PRIROUTER on the OSA DEVICE or INTERFACE statement(s) so that packets to unknown destinations may be forwarded to the backend IP addresses.
    - Recommendation: Use VMAC on the OSA LINK or INTERFACE statement(s) for shared OSA ports rather than PRIROUTER.
  - Do not code IPCONFIG IPSECURITY
    - HiperSockets Accelerator and IPSec are mutually exclusive.

- A single z/OS stack may be used as a “Router” (Accelerator) between remote devices and other LPARs.
  - The Accelerator stack is the only one that directly connects to the network via QDIO OSAs.
  - The Accelerator stack is also connected to the other LPARs via HiperSockets LANs.
  - The first packet, from a remote device to a backend LPAR, or from a backend LPAR to a remote device, will travel up the Accelerator stack to the IP layer. 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.
    - If a specific HiperSockets Acclerator entry is not used for 60 seconds, it is deleted from the iQDIORouting Table.
    - Each entry in the iQDIORouting table contains: the backend LPAR IP address, first hop IP address beyond OSA, and OSA link name.
  - IP time to live (TTL) Processing bypassed (considered no hop).
  - HiperSockets accelerator is for unicast IPv4 packets only.
  - There are no packet trace functions available to HiperSockets-accelerator-forwarded packets.
    - OSAENTA provides similar information for the packets through the OSA.
- There may be multiple HiperSockets Acclerators on a single CEC supporting different backend LPARs.
- HiperSockets Accelerator is designed to route traffic into one OSA priority queue and does not honor TOS/DSCP setting in the IP header.
  - This was done for performance reasons to block as many packets as possible in one write operation out to the OSA adapter.
- Note: Some documentation refers to HiperSockets Acceleration as HSA. This may causes confusion because it also stands for Hardware System Area.

HiperSockets Accelerator
Renamed QDIO Accelerator

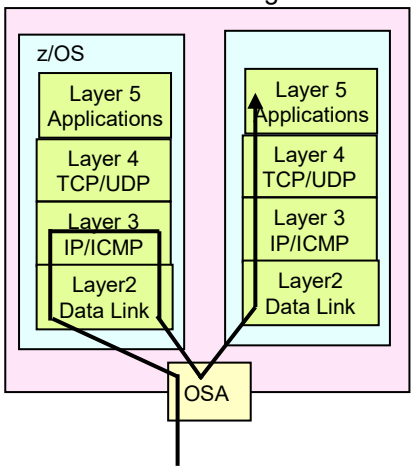
HiperSockets Accelerator

```
                +-------------------------------------------------------------------+
                |                                                                   |
>>---IPCONFig---V---+-------------------------------------------------------------+---+---><
                    |   +---NOIQDIORouting-------------------------------+   |
                    +---+-------------------------------------------------+---+
                    |   |                 +---QDIOPriority 1------------+   |   |
                    |   +---IQDIORouting---+----------------------------+---+   |
                    |                      +---QDIOPriority---priority---+       |
                    :                                                     :
                    :                                                     :
```

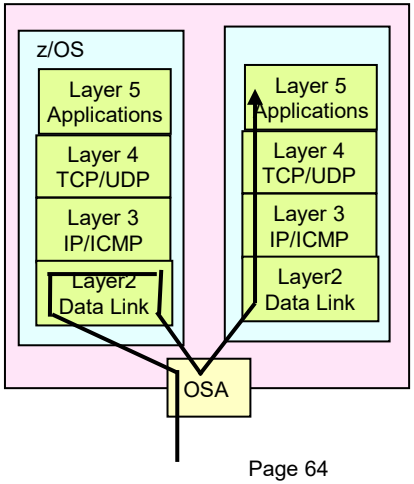
QDIO Accelerator

```
                +-------------------------------------------------------------------+
                |                                                                   |
>>---IPCONFig---V---+-------------------------------------------------------------+---+---><
                    |    +---NOQDIOACCELerator---------------------------+   |
                    +----+-------------------------------------------------+---+
                    |    |                    +---QDIOPriority 1------------+   |   |
                    |    +---QDIOACCELerator---+----------------------------+---+   |
                    |                         +---QDIOPriority---priority---+       |
                    :                                                     :
                    :                                                     :
```

Traditional IP Routing



QDIO Accelerator



- Renamed and Enhanced in z/OS V1.11.
- Accelerator support expanded to include all combinations of QDIO and iQDIO traffic.
- Supports Sysplex Distributor (SD)
  - When traffic to target stack is sent over HiperSockets Dynamic XCF or QDIO as a result of VIPAROUTE definition.

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

Netstat Displays Updated for Accelerator

Netstat VCRT/-V example

```
NETSTAT VCRT DETAIL

MVS TCP/IP NETSTAT CS V1R11       TCPIP Name: TCPCS           14:16:16
Dynamic VIPA Connection Routing Table:
Dest IPaddr      DPort  Src IPaddr       SPort  DestXCF Addr
-----------      -----  ----------       -----  ------------
201.2.10.11      00021  201.1.10.85      01027  201.1.10.10
Intf:   OSAQDIOLINK
    VipaRoute: Yes      Gw: 199.100.1.1
  Accelerator: Yes
```

Netstat ROUTe/-r example

```
NETSTAT ROUTE QDIOACCEL

MVS TCP/IP NETSTAT CS V1R11      TCPIP NAME: TCPCS       09:51:02
Destination      Gateway          Interface
-----------      -------          ---------
9.67.4.1/32      0.0.0.0          OSAQDIO4
9.67.5.2/32      0.0.0.0          OSAQDIO5
9.67.20.3/32     0.0.0.0          HIPERSOCK2
```

VTAM Tuning Statistics Verify Acceleration

- Shows packets/bytes which are:
  - received over an interface and then accelerated (read direction)
  - accelerated over an interface (write direction)

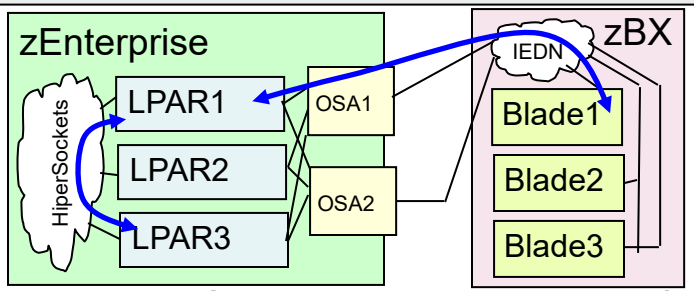
```
IST1233I DEV      = 0E2A       DIR      = READ
…
IST1236I BYTECNTO =          0 BYTECNT  =       9628
IST1810I PKTIQDO  =          0 PKTIQD   =         14
IST1811I BYTIQDO  =          0 BYTIQD   =       9368
IST924I -------------------------------------------------------------
…

IST1233I DEV      = 0E2F       DIR      = WR/1
…
IST1236I BYTECNTO =          0 BYTECNT  =       7424
IST1810I PKTIQDO  =          0 PKTIQD   =         10
IST1811I BYTIQDO  =          0 BYTIQD   =       6840
IST924I -------------------------------------------------------------
```

Acceleration Considerations

- Mutually exclusive with IPSECURITY
- Requires IP Forwarding to be enabled (for non-Sysplex Distributor acceleration)
- No acceleration for:
  - IPv6 traffic
  - Traffic which requires fragmentation in order to be forwarded.
  - Incoming fragments for an SD connection
  - VIPAROUTE over HiperSockets
  - Interfaces using optimized latency mode (OLM)

HiperSockets Integration with IEDN

HiperSockets Integration with OSA for IEDN



CHPARM=x2

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 and 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.

OSA OSX for HiperSockets Integration are defined identically in TCP/IP to OSA OSX defined without Hipersockets Integration defined!

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.

OSAINFO Command

Query and Display OSA Configuration

```
                                                                                                                                       +-----------------------------------------+
                                                                                                                                       |                        +---,MAX=200-----+   |
>---Display---TCPIP,---+--------------+---OSAinfo,---INTFNane=---intf_name---V---+--------------+---+---------------+---+--->< 
                       +---procname,---+                                         +---,BASE-------+   +---,MAX=*-------+
                                                                                 +---,BULKdata---+   +---,MAX=lines---+
                                                                                 +---,EE---------+
                                                                                 +---,REGAddrs---+
                                                                                 +---,SYSDist----+
```

- OSAINFO displays the current OSA configuration.
  - Requires OSA-E3+
  - Displays all sections when no filters are specified.
  - BASE displays physical characteristics and attributes for the interface.
  - BULKDATA displays IWQ routing variables for the BULKDATA ancillary queue.
    - BULKDATA routing variables are source and destination IP addresses, source and destination ports, and protocol.
  - EE displays IWQ routing variables for the Enterprise Extender (EE) ancillary queue.
    - EE routing variables are destination IP addresses, destination ports, and protocol.
  - REGADDRS displays registered Layer 3 unicast and multicast addresses.
  - SYSDIST displays IWQ routing variables for Sysplex Distributor ancillary queue.
    - SYSDIST routing variables are destination IP addresses and protocol.
- Background
  - Prior to OSAINFO, OSA/SF was often used for OSA QDIO to retrieve active information.

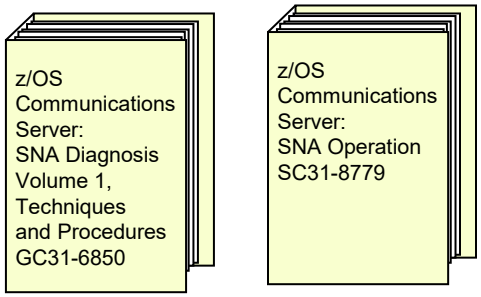
OSA Error Support

OSA Transparent Error Handling

- OSA-Express3+ can invalidate just the packet in error instead of having to invalidate all packets in the read operation
- Sample VTAM Tuning Statistics output
  - F NET,TNSTAT,TRLE=OSA2D80T,CNSL,TIME=5

```
IST1233I DEV      = 2D82        DIR      = READ
IST1719I PCIREALO =          0 PCIREAL  =         22
IST1720I PCIVIRTO =          0 PCIVIRT  =          0
IST1750I PCITHRSO =          0 PCITHRSH =          0
IST1751I PCIUNPRO =          0 PCIUNPRD =          0
IST1752I RPROCDEO =          0 RPROCDEF =          0
IST1753I RREPLDEO =          0 RREPLDEF =          0
IST1754I NOREADSO =          0 NOREADS  =          1
IST1721I SBALCNTO =          0 SBALCNT  =         22
IST1722I PACKCNTO =          0 PACKCNT  =         22
IST2185I FRINVCTO =          0 FRINVCT  =          1
IST1236I BYTECNTO =          0 BYTECNT  =       2796
IST1810I PKTIQDO  =          0 PKTIQD   =          0
IST1811I BYTIQDO  =          0 BYTIQD   =          0
```

Synchronization of OSA Diagnostic Data (QDIOSYNC)



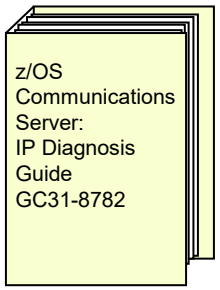
VTAM QDIOSYNC Overview

- VTAM QDIOSYNC trace captures OSA diagnostic data.
  - New in z/OS V1.8
- Instead of or in addition to using the HMC (Hardware Management Console) to manually capture OSA diagnostic data, QDIOSYNC may be used to cause the OSA to automatically capture diagnostic data when:
  - OSA detects an unexpected loss of host connectivity.
    - Unexpected halt signal from host
    - Host unresponsive
  - OSA receives CAPTURE signal from host due to:
    - VTAM-supplied MPF (Message Processing Facility) exit (IUTLLCMP) is driven.
      - Add VTAM-supplied MPF exit module, USEREXIT(IUTLLCMP), to SYS1.PARMLIB(MPFLISTxx).
      - Issue SET MPF=(xx,zz) where xx is the new PARMLIB member and zz is the old.
        - » Activates the new MPFLISTxx member.
      - Set corresponding SLIP trap to initiate a host dump.
      - See z/OS MVS Installation Exits for more information about MPF.
    - VTAM or TCP/IP FRR (Functional Recovery Routine) is driven with ABEND06F.
      - Result of SLIP PER trap that specifies ACTION=RECOVERY.
- After QDIOSYNC trace use the HMC to copy the OSA diagnostic data.

Start Option or Modify Command

- VTAM start options and commands
  - MODIFY TRACE and NOTRACE with TYPE=QDIOSYNC
    - Activate and terminate QDIOSYNC trace.
    - ID=trle_name
      - To activate that trace on a single OSA.
    - ID=* is supported
      - SAVE=NO applies QDIOSYNC trace to all currently active OSA TRLEs.
      - SAVE=YES applies QDIOSYNC trace to all currently active and future active OSA TRLEs.
- QDIOSYNC TRACE OPTION filters
  - ALLIN
    - Collects only inbound diagnostic data for all OSAs.
  - ALLOUT
    - Collects only outbound diagnostic data for all OSAs.
  - ALLINOUT
    - Collects inbound and outbound diagnostic data for all OSAs.
  - IN
    - Collects only inbound diagnostic data only for OSAs defined to this VTAM.
  - OUT
    - Collects only outbound diagnostic data only for OSAs defined to this VTAM.
  - INOUT
    - Collects inbound and outbound diagnostic data only for OSAs defined to this VTAM.

OSA-Express Network Traffic Analyzer (OSAENTA)



OSAENTA Overview

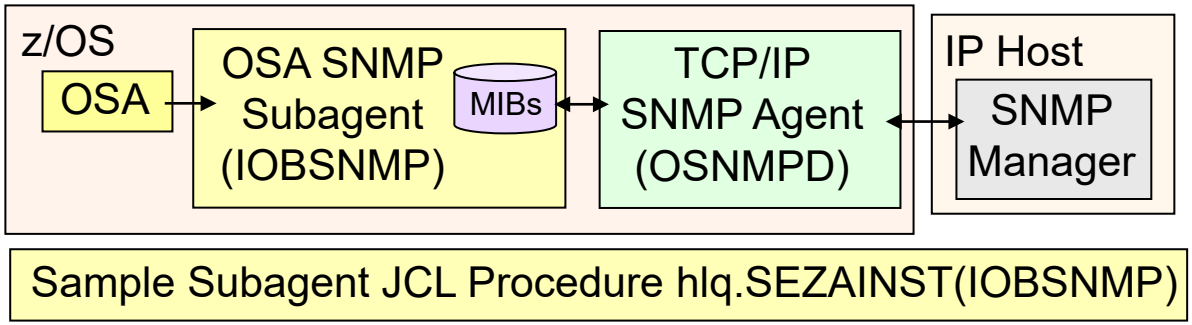
- OSA-Express network traffic analyzer (OSAENTA) traces frames for an OSA in QDIO mode.
  - Requires z/OS V1.8+
  - Also known as OSA Network Traffic Analyzer (OSA NTA)
- OSAENTA is controlled and formatted by z/OS Communications Server (CS), but is collected in the OSA port.
  - Support Element (SE) customization is required to enable or disable OSAENTA and authorize tracing outside the local LPAR.
- OSAENTA has capabilities beyond other z/OS CS tracing:
  - Trace frames discarded by the OSA.
  - Trace the MAC headers for packets.
  - Trace ARP packets
  - Trace packets to and from other users sharing the OSA (other TCP/IP stacks, z/Linux users, and z/VM users)
  - Trace SNA packets

Update Profile or Vary TCPIP

- Control OSAENTA trace using OSAENTA statement in the TCP/IP Profile or the VARY TCPIP,,OSAENTA command.
  - Filter what data is collected
    - IP address
    - Protocol (TCP,UDP,etc.)
    - Port number
    - Frame type
    - There is a limit of only one filter value per OSAENTA statement/command.
    - There is a limit of up to 8 filter values per filter (ie. only 8 port numbers may be defined).
    - Up to 8 IPv4 addresses and up to 8 IPv6 addresses may be specified.
    - All frames that match any IP address and match all other filters are captured.
  - Specify how much data is to be collected.
  - OSAENTA command OPERCMDS resource name is MVS.VARY.TCPIP.OSAENTA.
- Display current OSAENTA trace settings using the Netstat DEvlinks/-d command.
- OSAENTA dynamically creates interface EZANTAxxxxxxxx.
  - xxxxxxxx is the port name in the OSAENTA command and the TRLE.
    - TRLE must exist.
  - Used for receiving trace records.
  - Use VARY TCPIP,,OSAENTA commands ON, OFF, and DEL to start, stop, and delete the OSAENTA interface.
- CTRACE (Component Trace) uses SYSTCPOT to collect the trace records.
  - IPCS CTRACE with component name SYSTCPOT may be used to format the trace.

SNMP Support

OSA SNMP Support



- OSA provides an SNMP Subagent (IOBSNMP)
  - Available for use with TCP/IP SNMP Agent (OSNMPD)

OSA TCP/IP Considerations

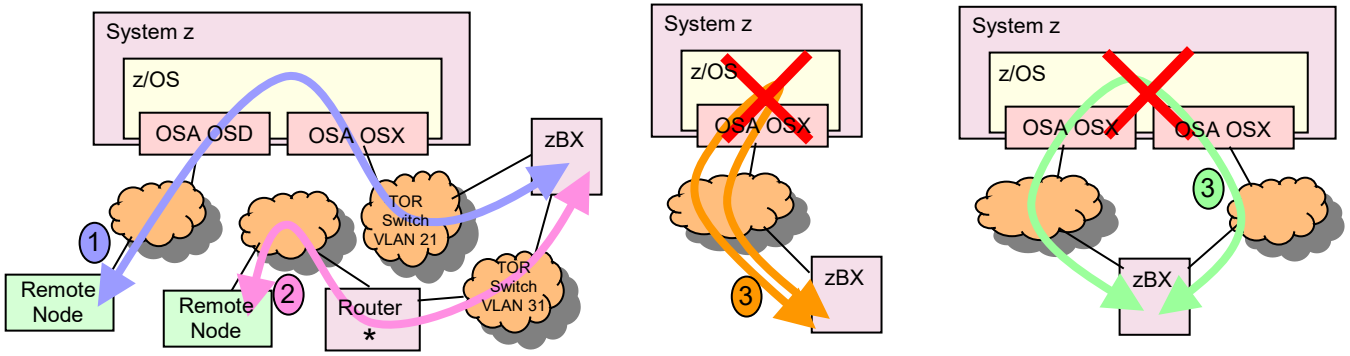
Considerations

- OSA VIPA Limitations
  - OSA devices have a limit on the number of IP addresses that can be registered to the device. The limit is dependent on the microcode level of the OSA. This limit applies across all TCP/IP stacks that share the OSA. When defining a large number of VIPAs, take care not to exceed this limit. If the limit is exceeded, IP addresses beyond the limit will not be registered with the OSA, and incoming packets with those IP addresses will not be routed to the correct stack unless that stack is designated as the primary router.
- RIP Routing and VIPA in Same (Sub)Network as OSAs
  - If using the RIP routing protocol and host route broadcasting is not supported by adjacent routers (that is, adjacent routers are unable to learn host routes), the following restrictions for VIPA addresses must be applied in order to benefit from fault tolerance support:
    - If you use subnetting and VIPA addresses are in the same network as the physical IP addresses, the subnetwork portion of any VIPA addresses must not be the subnetwork portion of any physical IP addresses in the network. In this case, assign a new subnetwork for the VIPA address.
    - If subnetting is not used on any physical interface, the network portion of any VIPA address must not be the network portion of any physical IP address in the network. In this case, assign a new network for the VIPA address, preferably a class C network address.
  - If using the RIP routing protocol and host route broadcasting is supported by adjacent routers (that is, adjacent routers are able to learn host routes), the network or subnetwork portions of VIPA addresses can be the same across multiple z/OS TCP/IP stacks in the network.
- Spanning Tree Protocol
  - If using a DVIPA when connecting an OSA QDIO device to a intelligent bridge or switch, ensure that the Spanning Tree Protocol (STP) on the intelligent bridge or switch is configured properly for DVIPA giveback and takeover operations. See the IP Configuration Guide for more details on STP problems.
- Port Fast Mode
  - If using VIPA in the same subnet as the OSA, along with an intelligent bridge or switch in access mode, ensure that ’Port fast mode’ (Cisco) is enabled. This helps to decrease the amount of time the VIPA is unreachable in scenarios where there is dynamic movement of VIPA. For more information, see your bridge or switch manual.
- 1000BaseT Auto-Negotiation
  - OSA requires same to same connection; Auto-Negotiation on both OSA and Switch Port (or 100 Full).
  - OSA can only set either 100 Full or Auto-Negotiation; there is no option for 1000 Full duplex setting.
  - Only way to get to 1000 Full duplex is by Auto-negotiation.
  - Changing OSA setting from 100 Full to Auto-Negotiation is disruptive but with high availability setting the alternate path should pick up the traffic.

OMPROUTE OSPF Coding

- Optimize Performance
  - To minimize routing table size and advertisements that have to be processed, try to put z/OS and the sysplex into a stub or totally stubby area or isolate areas with BGP or EIGRP.
  - To minimize OSPF adjacencies, try to avoid OMPROUTE becoming the designated router.
  - Only use debug tracing when necessary. Use CTRACE tracing whenever possible.
- OSPF uses multicast packets
  - Disable multicast snooping on switches with shared OSAs attached to them.
- 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.
  - 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.

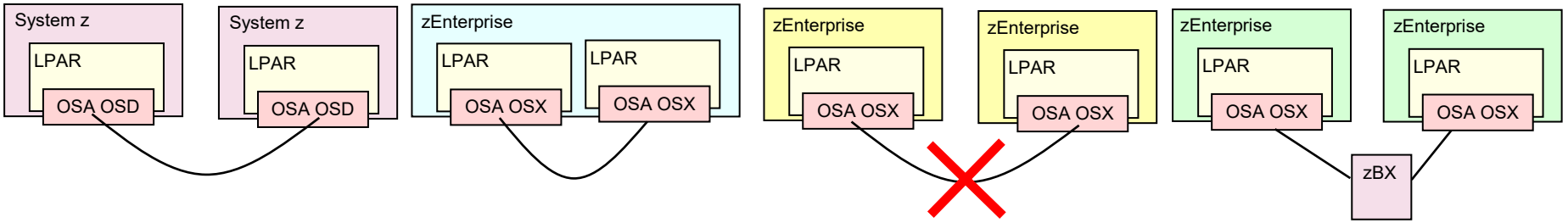
Non-System z to/from zBX



- 1 Traffic from outside the System z may flow into z/OS over OSA in OSD mode and be routed to zBX over OSA in OSX mode.
- 1 Traffic from zBX may flow to z/OS over OSA in OSX mode and be routed to outside the System z over OSA in OSD mode.
- 2 Traffic from outside the System z may flow to the zBX without going through an LPAR.
- 2 Traffic from the zBX may flow outside the System z without going through an LPAR.
- 3 z/OS will not route traffic over an OSA in OSX mode that was received over OSA in OSX mode.

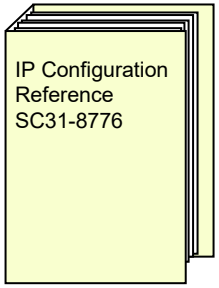
* Router may be any device with IP (Layer 3) Routing capability, including z/OS.

OSA Direct Connection



- Two OSA may be connected directly with an OSA cable without a switch for OSD (QDIO) connectivity.
- Two OSA may be connected directly with an OSA cable without a switch for OSX connectivity for LPAR to LPAR connection on a single Ensemble Node (single CEC).
- Two OSA may **NOT** be connected directly with an OSA cable without a switch for OSX connectivity between two Ensemble Nodes (multiple CECs).
- Two Ensemble OSX connection requires zBX.

Source IP Address



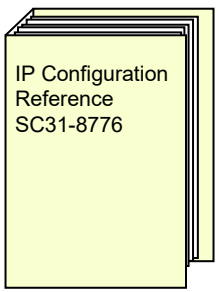
Traffic Path In/Out of z/OS

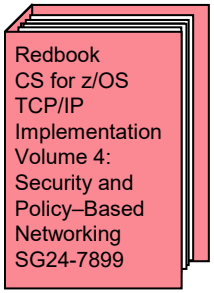
- Inbound Connections
  - When a connection is initiated from a remote node, the remote node sends a connection packet to a z/OS application.
    - Typically the z/OS application swaps the source and destination IP address from the received packet to be used in the response packet.
      - Takes the source addr from received packet and uses it as destination addr in the response packet.
      - Takes the destination addr from the received packet and uses it as source addr in the response packet.
- Outbound Connections
  - When a connection is initiated from a z/OS application the source IP address is determined by the source IP address selection algorithm.
    - See “Source IP Address Selection” section in the IP Configuration Guide manual.
  - When a connection is initiated from a z/OS application the destination IP address is either passed to the application in the connection command (ie. ftp 9.15.42.10) or is determined by domain name resolution of host name (ie. ftp wscftpsrv).
- Routing Table is used to send packet
  - Destination IP address is used with Routing Table (may contain a combination of static and dynamically learned routes), to determine which network path to send packet over.

Source IP Address Selection

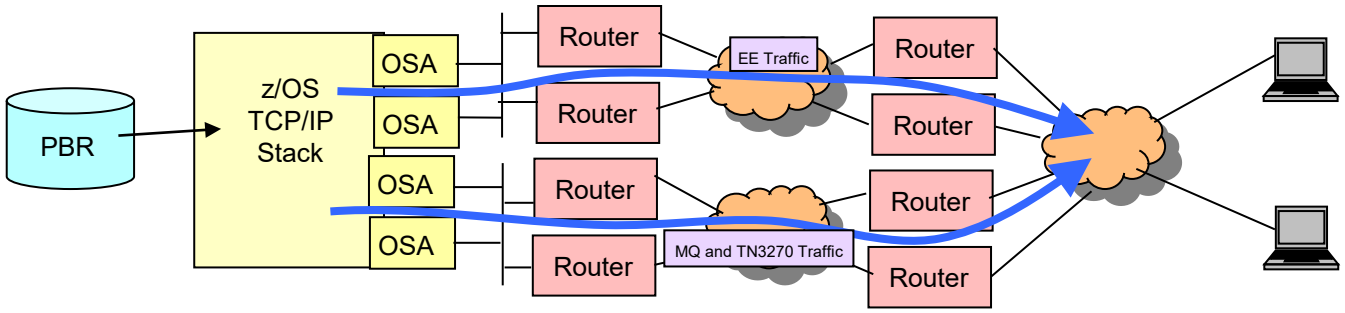
- As per the IP Configuration Guide…
- TCP/IP determines the source IP address for a TCP outbound connection, or for a UDP or RAW outbound packet, using the following sequence, listed in descending order of priority.
  1. Sendmsg( ) using the IPV6_PKTINFO ancillary option specifying a nonzero source address (RAW and UDP sockets only)
  2. Setsockopt( ) IPV6_PKTINFO option specifying a nonzero source address (RAW and UDP sockets only)
  3. Explicit bind to a specific local IP address
  4. PORT profile statement with the BIND parameter
  5. SRCIP profile statement (TCP connections only)
  6. TCPSTACKSOURCEVIPA parameter on the IPCONFIG or IPCONFIG6 profile statement (TCP connections only)
  7. SOURCEVIPA: static VIPA address from the HOME list or from the SOURCEVIPAINTERFACE parameter
  8. HOME IP address of the link over which the packet is sent
- For a TCP connection, the source address is selected for the initial outbound packet, and the same source IP address is used for the life of the connection. For the UDP and RAW protocols, a source IP address selection is made for each outbound packet.

Policy Based Routing



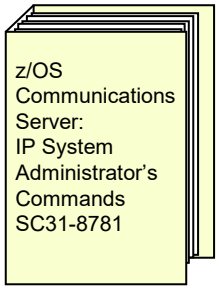


Outbound Routing



- Policy-based Routing (PBR) of Outbound Traffic (traffic that Originates on z/OS)
  - Choose first hop router, outbound network interface (including VLAN), and MTU
  - Choice can be based on more than the usual destination IP address/subnet
  - With PBR, the choice can be based on source/destination IP addresses, source/destination ports, TCP/UDP, etc.
  - Allows an installation to separate outbound traffic for specific applications to specific network interfaces and first-hop routers:
    - Security related
    - Choice of network provider
    - Isolation of certain applications
      - EE traffic over one interface
      - TN3270 traffic over another interface
  - PBR policies will identify one or more routes to use
    - If none of the routes are available, options to use any available route or to discard the traffic will be provided

Commands



Some Useful Commands for OSA Information

- IP Commands
  - See the IP System Administrator's Commands manual for syntax and details.
  - Vary TCPIP,procname,OSAENTA
    - Control the OSA-Express Network Traffic Analyzer (OSAENTA) tracing facility.
  - Vary TCPIP,procname,START/STOP,device_name/interface_name
    - Start or stop device or interface.
  - NETSTAT ARp/-R ip_addr/ALL…
    - Output includes the ip_addr to MAC address mapping.
  - NETSTAT DEvlinks/-d…
    - Output includes information about devices, links, and interfaces.
  - NETSTAT HOme/-h…
    - Output includes the IP Addresses to links/interfaces mapping.
  - NETSTAT SRCIP/-J…
    - Output includes Source IP Address information.
- VTAM Commands
  - See the SNA Operations manual for syntax and details.
  - DISPLAY NET,ID=xcaname…
    - Output includes LINE and PU.
  - DISPLAY NET,TNSTAT
    - Output indicates which TRLEs are collecting statistics if start option TNSTAT is specified.
  - DISPLAY NET,TRL… and DISPLAY NET,TRL,TRLE=…
    - Output includes TRL/TRLE information.
  - MODIFY procname,TRACE/NOTRACE,TYPE=QDIOSYNC…
    - Control QDIO Synchronization Trace

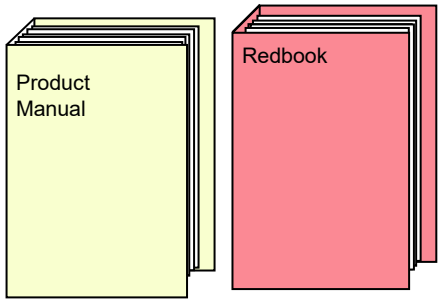
More Information

Web Information

- IBM ATS Technical Documents:
  - http://www.ibm.com/support/techdocs
- z/OS Communications Server
  - http://www.ibm.com/software/network/commserver/zos
- IBM Information Center
  - http://www.ibm.com/support/documentation/us/en
- IBM Education Assistant
  - http://www.ibm.com/software/info/education/assistant
- z/OS Communications Server Publications
  - http://www.ibm.com/systems/z/os/zos/bkserv
- IBM Redbooks
  - http://www.redbooks.ibm.com
- System z main web site:
  - http://www.ibm.com/systems/z/hardware/

IBM Product Manuals and Redbooks

- z/OS Communications Server
  - IP Configuration Guide, SC31-8775
  - IP Configuration Reference, SC31-8776
  - IP System Administrator's Commands, SC31-8781
  - IP Diagnosis Guide, GC31-8782
- z/OS SNA
  - Network Implementation Guide, SC31-8777
  - SNA Resource Definition Reference, SC31-8778
  - SNA Operation, SC31-8779
  - SNA Diagnosis Volume 1, Techniques and Procedures, GC31-6850
- OSA
  - Planning for ESCON, FICON, InfiniBand, Coupling Links, and OSA, GA23-0367
  - OSA-E Customer's Guide and Reference, SA22-7935
  - OSA-E Integrated Console Controller User's Guide, SA22-7990
  - User's Guide for OSA/SF, SC28-1855
- z/OS Hardware Configuration Definition
  - HCD Reference Summary, SX33-9032
  - HCD Planning, GA22-7525
  - HCD User's Guide, SC33-7988
  - IOCP User's Guide, SB10-7037
- zEnterprise
  - Install Manual (EC12 model 2827 is GC28-6913, different one for each processor)
  - Install for Physical Planning (EC12 model 2827 is GC28-6914, different one for each processor)
  - System Overview (EC12 model 2827 is SA22-1088, different one for each processor)
  - Ensemble Planning and Configuring Guide, GC27-2608
  - Intro to Ensembles, GC27-2609
  - HMC Operations Guide for Ensembles, SC27-2615
  - HMC Ensembles, SC27-2622
- Redbook
  - Communications Server for z/OS TCP/IP Implementation Volume 1: Base Functions, Connectivity, and Routing
  - IBM System z Connectivity Handbook, SG24-5444
  - OSA Implementation Guide, SG24-5948



Summary of Features and Other Operating Systems

Appendix: Syntax

Device/Link Syntax

- LCS (non-QDIO OSA OSE mode) and MPCIPA (QDIO OSA OSD mode) DEVICE and LINK

Parameter	DEVICE or LINK
AUTORESTART	DEVICE
device_number	DEVICE LCS only
DYNVLANREG	LINK IPAQENET only
ETHERNET/802.3/ ETHEROR802.3	LINK LCS only
IFSPEED	LINK (LCS default 4000000) (MPCIPA default 100000000)
INBPERF	LINK MPCIPA only
IOBUFFERSIZE	DEVICE LCS only
IPAQENET	LINK MPCIPA only
IPBCAST	LINK MPCIPA only
link_number	LINK LCS only
MONSYSPLEX	LINK
NETMAN	DEVICE LCS only
PRIROUTER	DEVICE MPCIPA only
READSTORAGE	LINK MPCIPA only
SECCLASS	LINK
VLANID	LINK IPAQENET only
VMAC	LINK IPAQENET only

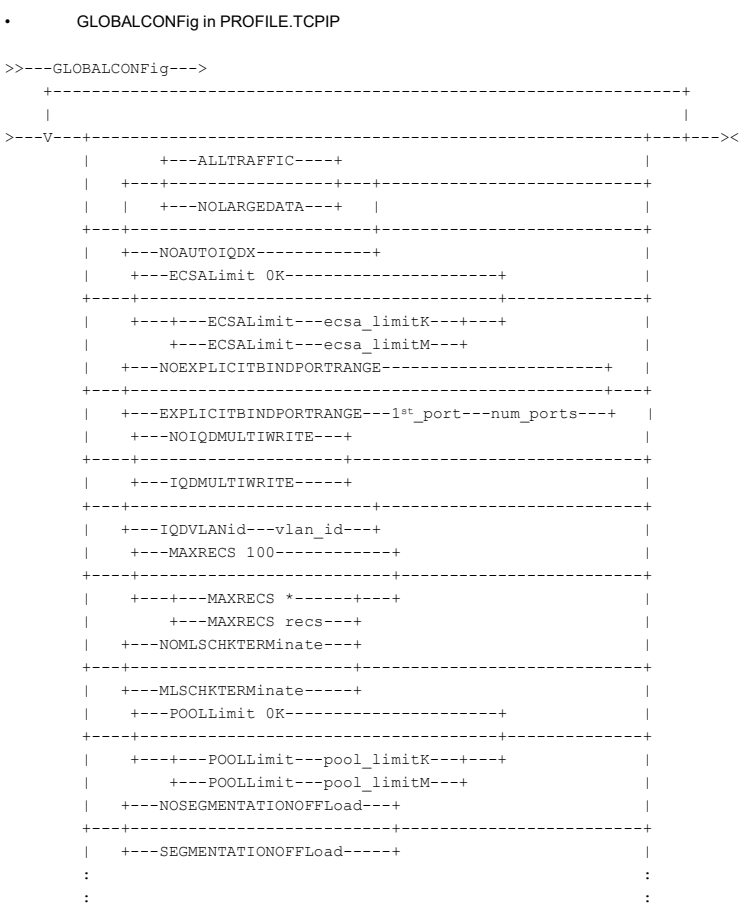
Interface Syntax

- IPAQENET (IPv4) and IPAQENET6 (IPv6) INTERFACE (QDIO only)

Parameter	Interface
CHPID	OSX only
DUPADDRDET	IPAQENET6 only
DYNVLANREG	Both IPAQENET and IPAQENET6
INBPERF	OSD default BALANCED / OSX default DYNAMIC WORKLOADQ
INTFID	IPAQENET6 only
IPADDR	OSD DEFINE (required) / OSX DEFINE (optional)
IPBCAST	IPAQENET only
ipv4_addr	IPAQENET OSD only
ipv4_addr/0	IPAQENET OSD only
ipv4_addr/num_mask_bits	IPAQENET only
ipv6_addr_spec	IPAQENET6 only DEFINE (optional)
ISOLATE	Both IPAQENET and IPAQENET6
MONSYSPLEX	Both IPAQENET and IPAQENET6
MTU	Both IPAQENET and IPAQENET6
OLM	Both IPAQENET and IPAQENET6
PORTNAME	Both IPAQENET and IPAQENET6
PRIROUTER	OSD only
READSTORAGE	Both IPAQENET and IPAQENET6
SECCLASS	Both IPAQENET and IPAQENET6
SOURCEVIPAINTERFACE	Both IPAQENET and IPAQENET6
TEMPPREFIX	IPAQENET6 only
VLANID	OSD (optional) / OSX (required)
VMAC	OSD no default / OSX default ROUTEALL

GlobalConfig Syntax

- GLOBALCONFig in PROFILE.TCPIP



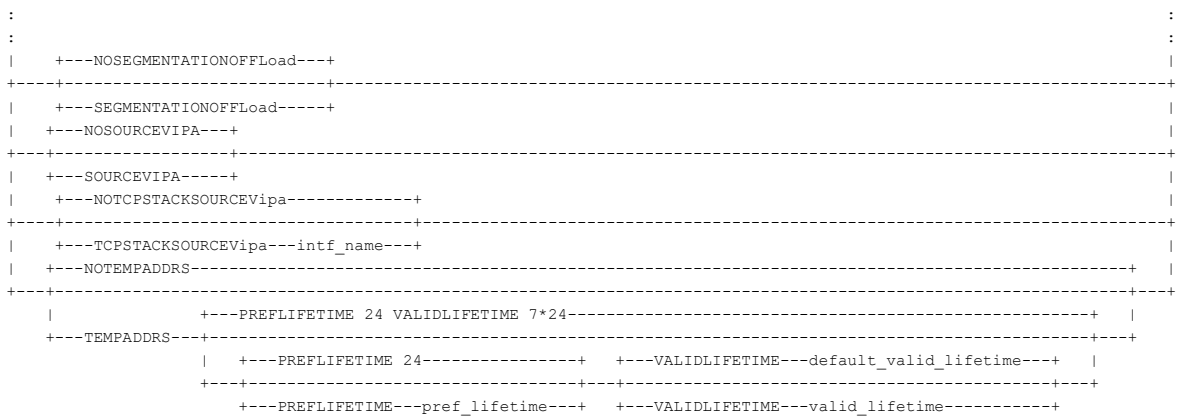
IPCONFIG Syntax

- IPCONFIG (IPv4) in PROFILE.TCPIP

IPCONFIG6 Syntax

- IPCONFIG6 (IPv6) in PROFILE.TCPIP

IPCONFIG6 Syntax (cont.)



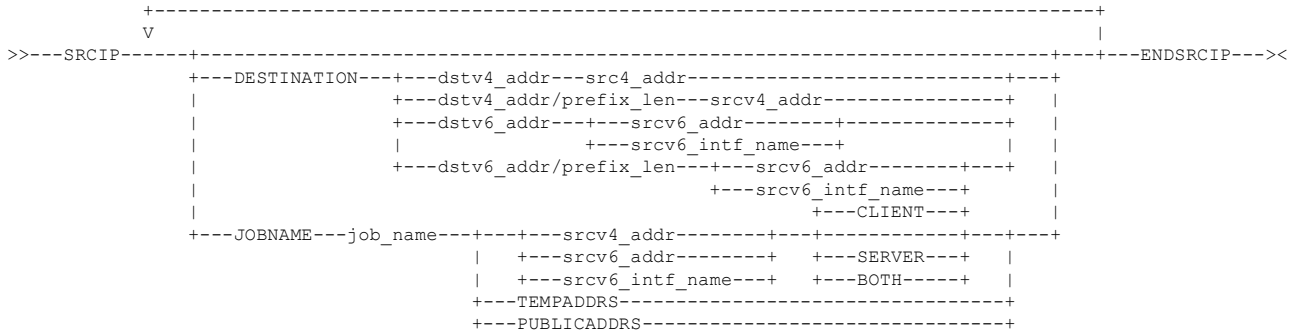
Policy Agent Syntax to Map TOS to Priority

```
QoS SetSubnetPrioTosMask may be specified in the main policy configuration file or the QoS configuration file.

>>---SetSubnetPrioTosMask--->

|---{--->

    +---SubnetAddr---0.0.0.0---+
|---+--------------------------+---Subnet TosMask---mask--->
    +---SubnetAddr---address---+

   +-------------------------------------------------------------------------+
   V   +---PriorityTosMapping----4---------tos--------0------------------+   |
|------+-------------------------------------------------------------------+---+--->
       +---PriorityTosMapping---priority---tos---+--------------------+---+
                                                 +---user_priority---+
|---}--->
```

- { and }
  - Brackets must be on separate lines by themselves.
- address
  - an IP address or a LINK name or an Interface name
- mask
  - eight bits for ToS (Type of Service) or Traffic Class mask
- priority
  - integer 0-4 indicating device priority level
- tos
  - eight bits indicating ToS or Traffic Class
- user_priority
  - integer 0-7 indicating VLAN priority
- If PriorityTosMapping is not specified for a ToS or Traffic Class value, a device priority value of 4, and a user priority value of 0 is set.
- A maximum of 32 PriorityTosMapping parameters can be specified.
- Coding the VLAN user priority causes a frame to be sent out based on the IEEE 802.1Q specification. A VLAN priority-tagged frame is used to convey packet priority to the switches; it has a value of NULL for VLANID. A full VLAN-tagged frame contains both the priority and non-null VLANID. If you have switches in your network that do not support the IEEE 802.1Q standard or that are not properly configured for these types of frames, the frames might be dropped by the switch.

SRCIP Statement

- PROFILE.TCPIP



  - DESTINATION
    - When destination IP address matches then source IP address or Interface is defined.
    - Source address cannot be a distributed DVIPA unless GLOBALCONFIG EXPLICITBINDPORTRANGE is configured and one of the following situations exist:
      - Application issued a connect request without a prior explicit bind.
      - Source port was explicitly bound to port 0, to a port less than 1024, or to a port that is reserved for this job by a PORT or PORTRANGE.
    - If source port is explicitly bound to an ephemeral port that is not reserved for this job, the connection request fails.
    - If source port < 1024 or is a port that is reserved for this job and specified source is a distributed DVIPA, you must ensure that multiple outbound connections to same destination IP addr and port cannot occur concurrently with same source IP addr and port.
    - If you use distributed DVIPAs for the source IP address in a SRCIP DESTINATION entry, you should specify the SYSPLEXPORTS keyword on the VIPADISTRIBUTE statement for those distributed DVIPAs.
    - The specified IP address does not need to be defined prior to the processing of the SRCIP block, but it must be defined before the first TCP connect request is issued for the associated destination; otherwise, the connect request fails.
  - JOBNAME
    - When job name matches then source address or interface is defined.
    - CLIENT, SERVER, and BOTH designate the type of socket function call on which the source IP address should be used.
    - Application can be a server if it binds to the IPv4 INADDR_ANY address or to the IPv6 unspecified address (in6addr_any), and the keyword SERVER or BOTH is specified with the SRCIP JOBNAME statement specified with a value other than JOBNAME *.
    - Designate whether to prefer a temporary IPv6 address (TEMPADDRS) or a public IPv6 address (PUBLICADDRS) when algorithm for default source address selection is used to select source IP address for an application that has specified job name.
    - TEMPADDRS specifies that this JOBNAME entry causes temporary IPv6 addresses to be preferred over public IPv6 addresses.
    - PUBLICADDRS specifies that this JOBNAME entry causes public IPv6 addresses to be preferred over temporary IPv6 addresses.

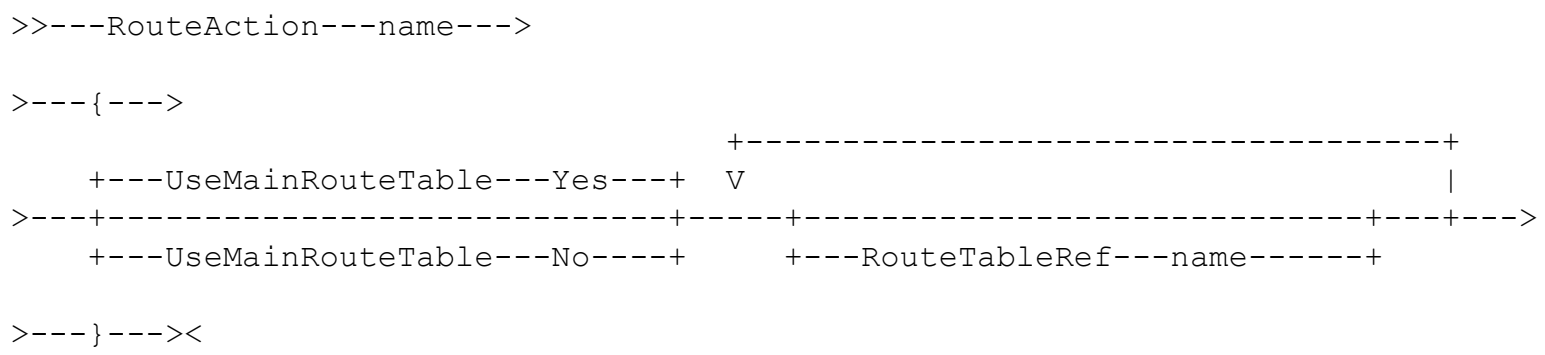
Route Table

- Policy-based Routing (PBR) configuration file

  - RouteTable is used to create a subset of the whole routing table.
    - UseGlobal causes the Multipath defined in the IPCONFIG statement to be used.

Routing Action

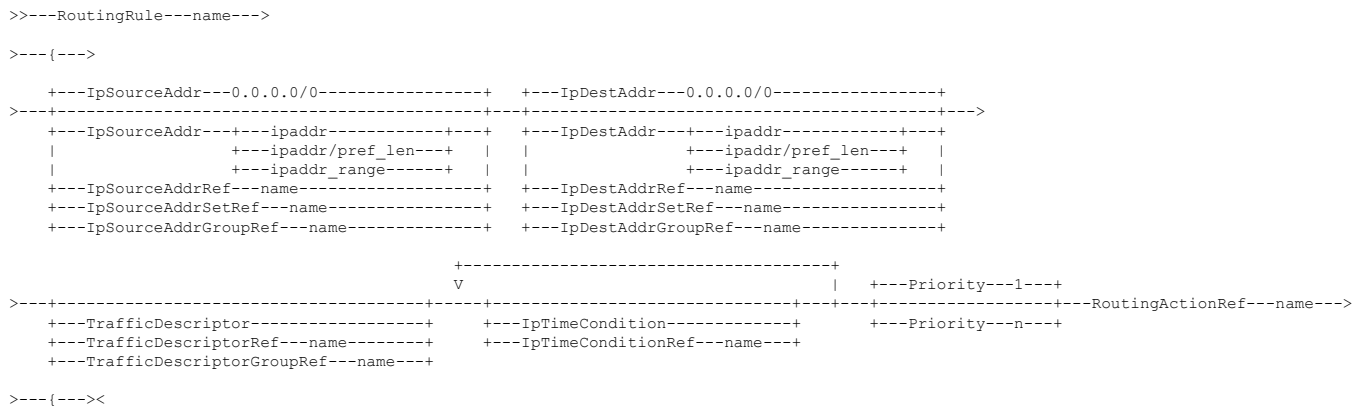
- Policy-based Routing (PBR) configuration file



  – RouteAction is used to indicate Routing Table subsets and if the Main Routing Table is to be used when a usable route is not found in any of the referenced subsets.

Routing Rule

- Policy-based Routing (PBR) configuration file



  - RoutingRule is used to define the type of traffic for the Routing Action to be used on.
    - IpSourceAddrRef indicates the name of a globally defined IpAddr statement that is used for the source IP address specification.
    - IpSourceAddrSetRef indicates the name of a globally defined IpAddrSet statement that is used for the source IP address prefix or range specification.
    - IpSourceAddrGroupRef indicates the name of a globally defined IpAddrGroup statement that is used for the source IP address specification.
    - IpDestAddrRef indicates the name of a globally defined IpAddr statement that is used for the destination IP address specification.
    - IpDestAddrSetRef indicates the name of a globally defined IpAddrSet statement that is used for the destination IP address prefix or range specification.
    - IpDestAddrGroupRef indicates the name of a globally defined IpAddrGroup statement that is used for the destination IP address specification.
    - TrafficDescriptorRef indicates the name of a globally defined TrafficDescriptor statement.
    - TrafficDescriptorGroupRef indicates the name of a globally defined TrafficDescriptorGroup statement.
    - IpTimeConditionRef indicates the name of a globally defined IpTimeCondition statement.

IP Address Reusable Statements

- Policy-based Routing (PBR) configuration file

```
>>---IpAddr---+----------+--->
              +---name---+
>---{--->

>---Addr---ip_addr--->

>---{--->< 


>>---IpAddrGroup---+----------+--->
                   +---name---+
>---{--->

   +--------------------------------+
   V                                |
>-----+---IpAddr----------------+---+--->
      +---IpAddrRef---name------+
      +---IpAddrSet-------------+
      +---IpAddrSetRef---name---+

>---{--->< 


>>---IpAddrSet---+----------+--->
                 +---name---+
>---{--->

>---+---Prefix---ipaddr/prefix_length---+--->
    +---Range---ipaddr_range------------+

>---{--->< 
```

Traffic Descriptor and IP Time Condition Reusable Statements

- Policy-based Routing (PBR) configuration file

```
>>---TrafficDescriptor---+----------+--->
                         +---name---+
>---{--->

    +---Protocol---All-----------+   +---SourcePortRange---0---------------+   +---DestinationPortRange---0---------------+
>---+----------------------------+---+-------------------------------------+---+-----------------------------------------+---+--------------------+--->
    +---Protocol---+---Tcp---+---+   +---SourcePortRange---+---n-------+---+   +---DestinationPortRange---+---n-------+---+   +---Jobname---name---+
                   +---6-----+                             +---n---m---+                                 +---n---m---+
                   +---Udp---+
                   +---17----+
                   +---All---+
>---+--------------------------+---+-------------------------+--->
    +---SecurityZone---name---+   +---SecurityLabel---name---+

>---{--->< 


>>---TrafficDescriptorGroup---name--->

>---{--->

   +---------------------------------------+
   V                                       |
>-----+---TrafficDescriptor--------------+---+--->
      +---TrafficDescriptorRef---name---+

>---{--->< 


>>---IpTimeCondition---+----------+--->
                       +---name---+
>---{--->
                                          +---MonthOfYearMask---111111111111---+   +---DayOfMonthMask---31---1's---+   +---TimeOfDayRange---0---24---+
>---+-----------------------------------+---+------------------------------------+---+-------------------------------+---+-----------------------------+--->
    +---ConditionTimeRange---range---+   +---MonthOfYearMask---12---n's-------+   +---DayOfMonthMask---31---n's---+   +---TimeOfDayRange---n---m----+
                                         +---DayOfMonthMask---62---n's---+
    +---DayOfWeekMask---1111111---+
>---+-----------------------------+--->
    +---DayOfWeekMask---7---n's---+

>---{--->< 
```

OSAENTA Syntax

The End

The End