#SHAREorg

SHARE
Technology · Connections · Results

Getting the most out of your OSA Adapter with z/OS Comm Server

Mike Fitzpatrick – mfitz@us.ibm.com
IBM Raleigh, NC

Friday, February 8th, 8:00am
Session: 12862

SHARE in San Francisco 2013

Trademarks, notices, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries or both:

- Advanced Peer-to-Peer Networking®
- AIX®
- alphaWorks®
- AnyNet®
- AS/400®
- BladeCenter®
- Candle®
- CICS®
- DataPower®
- DB2 Connect
- DB2®
- DRDA®
- e-business on demand®
- e-business (logo)
- e business(logo)®
- ESCON®
- FICON®
- GDDM®
- GDPS®
- Geographically Dispersed Parallel Sysplex
- HiperSockets
- HPR Channel Connectivity
- HyperSwap
- i5/OS (logo)
- i5/OS®
- IBM eServer
- IBM (logo)®
- IBM®
- IBM zEnterprise™ System
- IMS
- InfiniBand ®
- IP PrintWay
- IPDS
- iSeries
- LANDP®
- Language Environment®
- MQSeries®
- MVS
- NetView®
- OMEGAMON®
- Open Power
- OpenPower
- Operating System/2®
- Operating System/400®
- OS/2®
- OS/390®
- OS/400®
- Parallel Sysplex®
- POWER®
- POWER7®
- PowerVM
- PR/SM
- pSeries®
- RACF®
- Rational Suite®
- Rational®
- Redbooks
- Redbooks (logo)
- Sysplex Timer®
- System i5
- System p5
- System x®
- System z®
- System z9®
- System z10
- Tivoli (logo)®
- Tivoli®
- VTAM®
- WebSphere®
- xSeries®
- z9®
- z10 BC
- z10 EC
- zEnterprise
- zSeries®
- z/Architecture
- z/OS®
- z/VM®
- z/VSE

* All other products may be trademarks or registered trademarks of their respective companies.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries or both:

- Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license there from.
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- InfiniBand is a trademark and service mark of the InfiniBand Trade Association.
- Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
- IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

Notes:

- Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
- IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
- All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
- This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
- All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
- Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
- Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Refer to www.ibm.com/legal/us for further legal information.

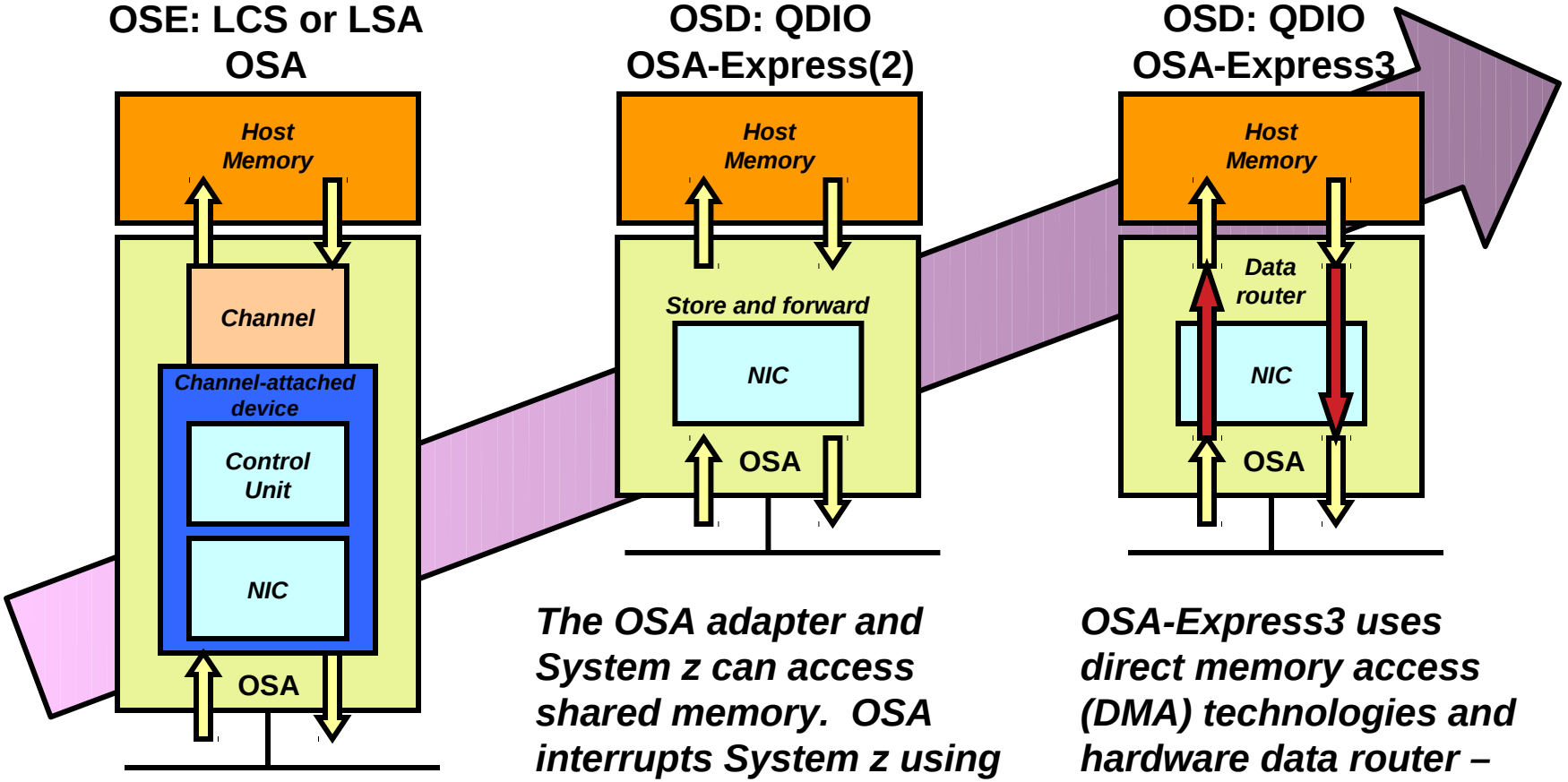
Agenda

- What is QDIO?
- OSA inbound “routing”
- Overview of selected recent OSA enhancements
- Appendix

Disclaimer: All statements regarding IBM future direction or intent, including current product plans, are subject to change or withdrawal without notice and represent goals and objectives only. All information is provided for informational purposes only, on an “as is” basis, without warranty of any kind.

What is QDIO?

Queued Direct IO (QDIO) – How System z accesses a LAN

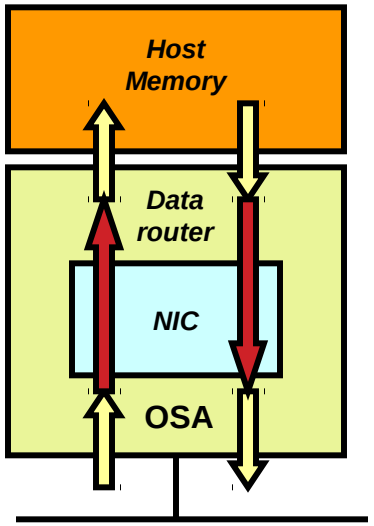


The same way a mainframe connected to an old IBM 3172, or a Cisco CIP

The OSA adapter and System z can access shared memory. OSA interrupts System z using PCI, while System z "interrupts" OSA via Signaling (SIGA instruction)

OSA-Express3 uses direct memory access (DMA) technologies and hardware data router – allowing data to flow through OSA without slow store and forward processing.

OSA and QDIO work together

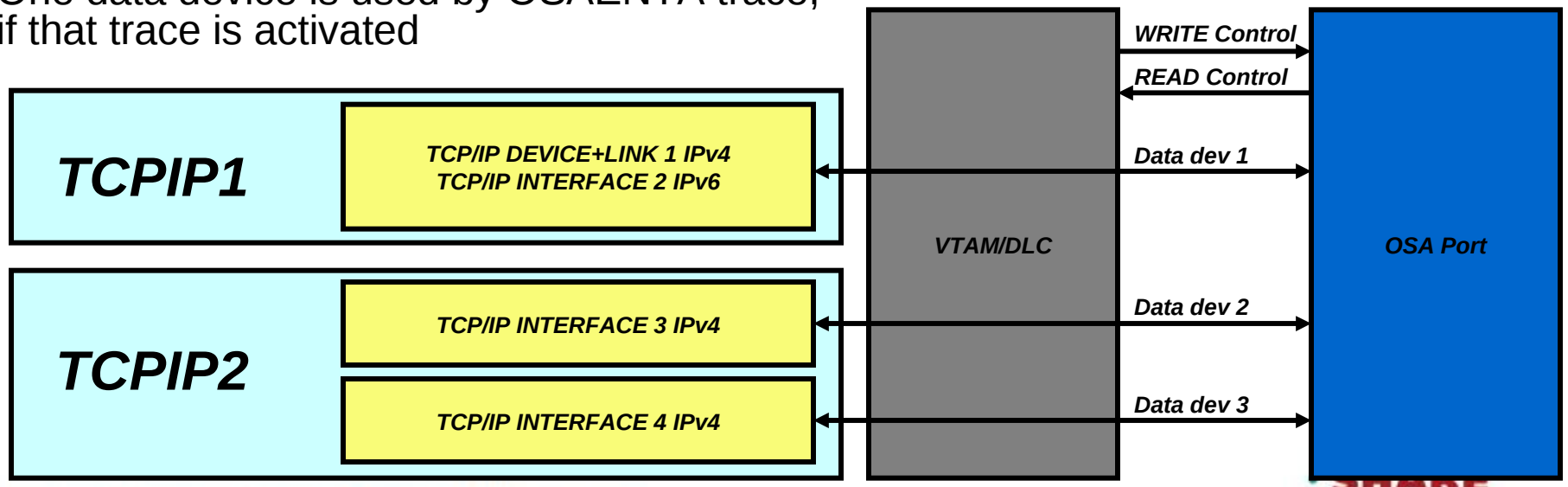


- **QDIO Layer 3 – for TCP/IP traffic only (IPv4 and IPv6)**
  - z/OS, z/VM, z/VSE, z/TPF, and Linux on System z
  - OSA IP Processing Assist (IPA) offloads many TCP/IP functions to the OSA adapter, of which some are:
    - Performs all ARP processing (IPv4 only)
    - Provides Multicast support
    - Builds MAC and LLC headers
    - Performs checksum processing
    - Performs outbound TCP segmentation
  - For SNA/APPN/HPR traffic with QDIO layer 3: use TN3270 and Enterprise Extender
- **QDIO Layer 2 – network protocol agnostic**
  - z/VM V5.2 with PTFs and Linux on System z
    - z/OS does not support QDIO Layer 2
  - Network protocol may be TCP/IP, SNA LLC2, NetBIOS, etc. (no IP Assist)
  - Linux uses QDIO Layer 2 for SNA LLC2 traffic in/out of Linux
    - IBM Communications Server for Linux and Communications Controller for Linux
- **OSA and QDIO are designed for high speed communication between System z and the Local Area Network**
  - Reduced TCP/IP path length
  - QDIO IP Processing Assist
  - LPAR-to-LPAR communication with port sharing
  - Direct Memory Access (DMA) Protocol
    - Memory-to-memory communication
      - *I/O interrupts minimized*
      - *Continuous direct data exchanges*
  - Dynamic customization
- **10 Gigabit Ethernet, Gigabit Ethernet, 1000BASE-T Ethernet (1000, 100, and 10 Mbit)**

A little more on QDIO and devices

- Control data between z/OS CS and the OSA port is exchanged over one read device and one write device (READ even address, WRITE odd address)
  - The READ and WRITE device pair is shared among all the IP interface definitions in an LPAR that use the same OSA port
- Data is exchanged over data devices
- One data device is used per:
  - A set of exactly one IPv4 and one IPv6 interface
  - IPv4: DEVICE+LINK, IPv6: INTERFACE
  - One IPv4 interface defined with INTERFACE
  - One IPv6 interface
- One data device is used by OSAENTA trace, if that trace is activated

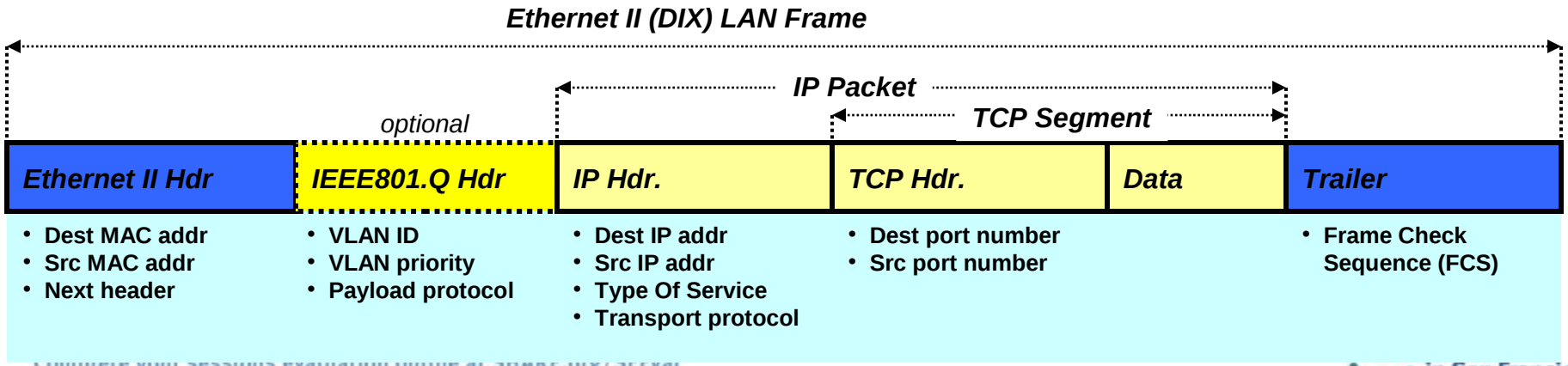
```
LNCTL=MPC,
READ=(2FD6),
WRITE=(2FD7),
MPCLEVEL=QDIO,
DATAPATH=(2FD8,2FD9,2FDA,2FDB),
PORTNAME=(O3ETHB1P),
PORTNUM=(1)
```



OSA inbound “routing”

Some basic LAN technology overview

- The LAN infrastructure transports “Frames” between Network Interface Cards (NICs) that are attached to the LAN media (Copper or fiber optic)
  - Ethernet II (DIX) – MTU 1500 and jumbo frame MTU 9000 (most common)
  - IEEE802 – MTU 1492 and jumbo frame MTU 8992
- Each NIC has a physical hardware address
  - A Media Access Control (MAC) address
    - Burned in (world-wide unique by vendors) or alternatively locally administered
- Every frame comes from a MAC and goes to a MAC
  - There are special MAC values for broadcast and multicast frames
- Every frame belongs to the physical LAN or to one of multiple Virtual LANs (VLAN) on the physical LAN
  - A VLAN ID is in the IEEE801.Q header if VLAN technologies are in use
- A frame carries a payload of a specified protocol type, such as ARP, IPv4, IPv6, SNA LLC2, etc.



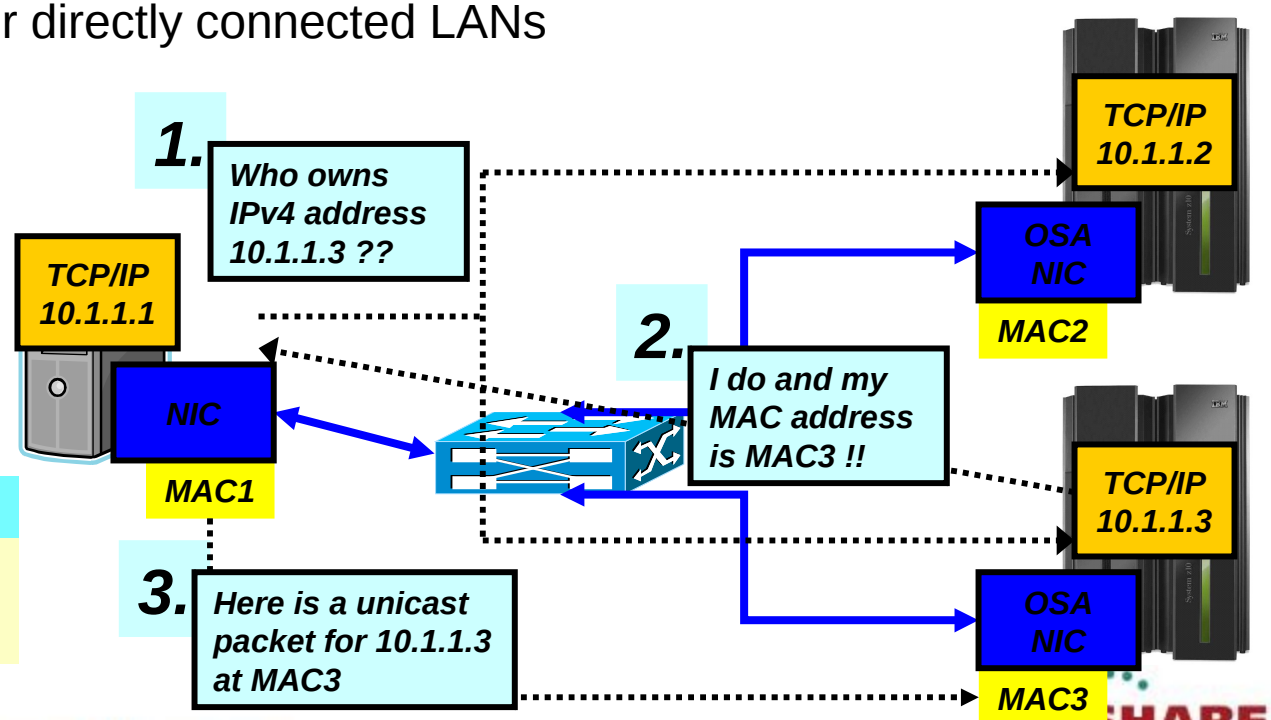
Correlation of IPv4 addresses and MAC addresses on a LAN – Address Resolution Protocol (ARP)

- An IPv4 node uses the ARP protocol to discover the MAC address of another IPv4 address that belongs to the same IPv4 subnet as it does itself.
- ARP requests are broadcasted to all NICs on the LAN
- The one NIC that has a TCP/IP stack with the requested IPv4 address responds directly back to the IPv4 node that sent out the broadcast
- Each IPv4 node maintains a cache of IPv4 addresses and associated MAC addresses on their directly connected LANs

IPv6 uses a similar method, but based on use of multicast. The IPv6 protocol is known as ND for Neighbor Discovery.

ARP Cache

IP address	MAC
10.1.1.2	MAC2
10.1.1.3	MAC3

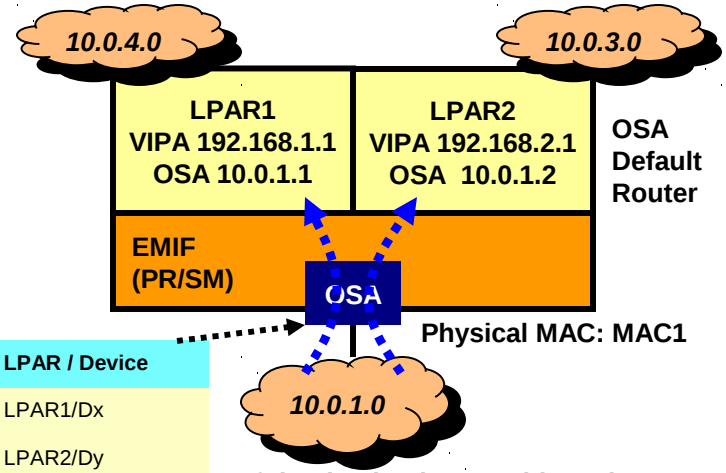


An example – formatted by Wireshark (http://www.wireshark.org)

- **A sample Address Resolution Request frame**

```
Ethernet II, Src: 02:00:00:13:89:01 (02:00:00:13:89:01), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
      Address: Broadcast (ff:ff:ff:ff:ff:ff)
      .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory default)
  Source: 02:00:00:13:89:01 (02:00:00:13:89:01)
      Address: 02:00:00:13:89:01 (02:00:00:13:89:01)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory default)
    Type: 802.1Q Virtual LAN (0x8100)
802.1Q Virtual LAN
    000. .... .... .... = Priority: 0
    ...0 .... .... .... = CFI: 0
    .... 0001 0101 1111 = ID: 351
    Type: ARP (0x0806)
Address Resolution Protocol (request)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (0x0001)
    Sender MAC address: 02:00:00:13:89:01 (02:00:00:13:89:01)
    Sender IP address: 10.3.51.16 (10.3.51.16)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 10.3.51.17 (10.3.51.17)
```

So how does this work in a System z environment with shared (virtualized) OSA adapters?



OSA Address Table (OAT)

IP address	LPAR / Device
192.168.1.1	LPAR1/Dx
192.168.2.1	LPAR2/Dy
10.0.1.1	LPAR1/Dx
10.0.1.2	LPAR2/Dy
Primary	LPAR2/Dy
Secondary	LPAR1/Dx

If the destination IP address in a frame that arrives at the OSA adapter doesn't belong to any of the LPARs that share the adapter, that IP packet is given to the LPAR that acts as the Default OSA router. There are separate IPv4 and IPv6 default OSA router specifications.

LAN frames

Dest_MAC	Dest_IPv4	Who gets it?	Why?
MAC1	10.0.1.1	LPAR1	Registered
MAC1	192.168.1.1	LPAR1	Registered
MAC1	10.0.1.2	LPAR2	Registered
MAC1	192.168.2.1	LPAR2	Registered
MAC1	10.0.3.5	LPAR2	Default router
MAC1	10.0.4.6	LPAR2	Default router

- An OSA NIC has a physical MAC address, just like all NICs
- An OSA NIC is often used by multiple LPARs and TCP/IP stacks
  - The NIC is virtualized so it functions as the NIC for multiple TCP/IP stacks, each with their own IP address
- If someone ARPs for 10.0.1.1 in LPAR1, OSA will return MAC1
- If someone ARPs for 10.0.1.2 in LPAR2, OSA will also return MAC1
- So what does OSA then do when a unicast frame arrives with a destination MAC address of MAC1?
  - It peeks into the IP header inside the frame, and consults a table known as the OSA Address Table (OAT) to see which LPAR the IP address inside the frame belongs to

Setting a few facts straight about OSA and "routing"

- **OSA is not a full-function IP router.**
  - OSA can analyze a destination IP address in a LAN frame to decide which LPAR an incoming IP packet belongs to.
  - OSA does not participate in dynamic routing updates.
  - OSA does not act as a general IP router
    - Depends on the TCP/IP stacks to do that and handle all IP routing issues.

- **By default, the OSA NIC has a single physical MAC address that is shared among all the LPARs that share the OSA port.**
  - All IP packets to all LPARs can be destined for one and the same MAC address
  - In that case, OSA selects correct stack based on destination IP address and the OSA Address Table

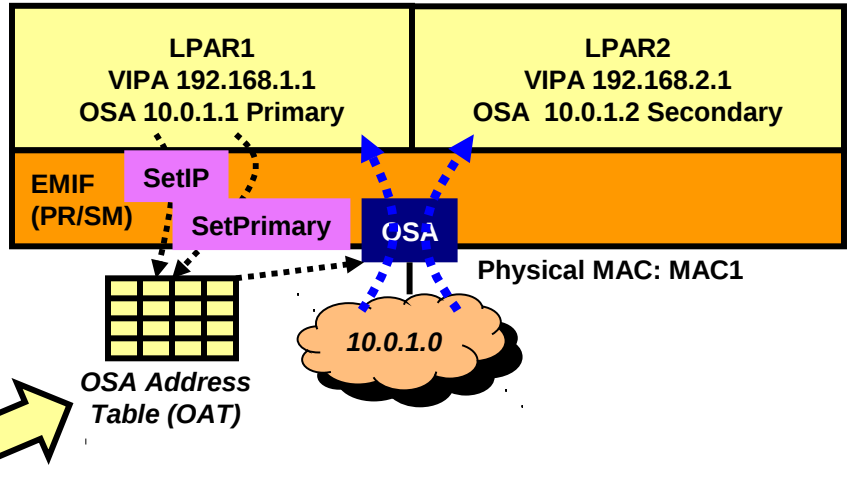
- **If z/OS acts as an IP router to IP networks behind z/OS, the destination address may be any IP address on those networks behind z/OS.**
  - LPAR designated as default router will receive such packets
  - Can become extremely cumbersome to set up if LPARs that share an OSA port are connected to different back-end networks

- **When a port is defined as an OSE/LCS port, the content of the OAT must be maintained *manually* through OSA/SF interaction.**

- **When a port is defined as a OSD/QDIO port, the content of the OAT is maintained *automatically* by the sharing TCP/IP stacks.**

QDIO layer-3: less administration, more dynamics – but OAT remains an important element of OSA inbound routing

QDIO Layer 3 - the OSA adapter is IP-aware and offloads certain functions from the TCP/IP stacks – check-summing, ARP processing, TCP segmentation, etc.



OAT is maintained dynamically by TCP/IP

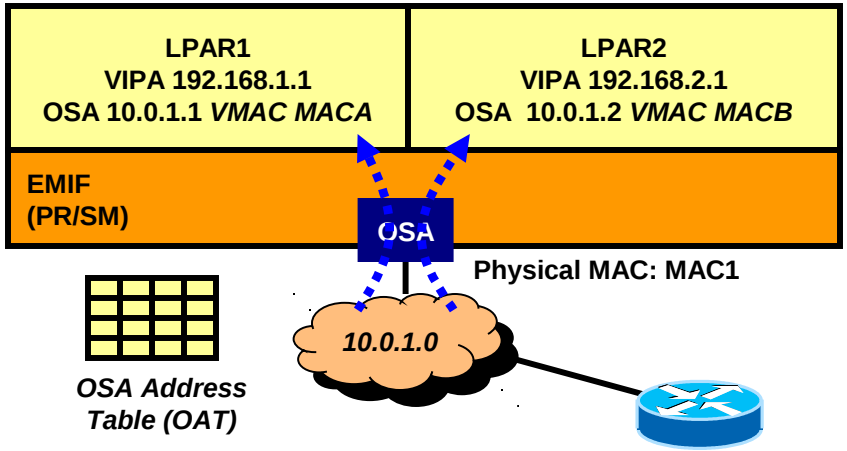
IP address	LPAR / Device
192.168.1.1	LPAR1/Dx
192.168.2.1	LPAR2/Dy
10.0.1.1	LPAR1/Dx
10.0.1.2	LPAR2/Dy
Primary	LPAR1/Dx
Secondary	LPAR2/Dy

OAT Updates:

- **LCS: operator intervention via OSA/SF commands**
- **QDIO: dynamic by software via QDIO control channel**

- QDIO device definitions in the TCP/IP stacks are used to dynamically establish the stack as the OAT default router, secondary router, or non-router.
- Whenever a QDIO device is activated or the TCP/IP home list is modified (through OBEYFILE command processing or through dynamic changes, such as dynamic VIPA takeover), the TCP/IP stack updates the OAT configuration dynamically with the HOME list IP addresses of the stack.
- The OAT includes all (non-LOOPBACK) HOME IP addresses of all the stacks that share the OSA adapter.
- The fact that the OSA microcode is IP address-aware (as it is in this scenario) is the reason for referring to this as QDIO layer 3 processing (layer 3 is generally the networking layer in an OSI model - the IP networking layer when using TCP/IP)

Wouldn't it be so much easier with a MAC address per z/OS TCP/IP network interface?



YES !!

The whole mess with PRIROUTER and SECROUTER is gone !!

It also removes issues with external load balancers that use MAC-level forwarding.
It makes a system z LPAR look like a "normal" TCP/IP node on a LAN.

Router's ARP cache

IP Address	MAC Address
10.0.1.1	MACA
10.0.1.2	MACB

- A packet for 192.168.1.1 arrives at the router from the downstream network
  - Router determines the destination is not directly attached to the router
  - Router looks into its routing table and determines next-hop IP address for this destination is 10.0.1.1, which is on a network that is directly attached to the router
  - Router looks into its ARP cache and determines the associated MAC address is MACA
  - Router forwards the IP packet in a LAN frame to MACA
- Frame arrives in OSA
  - OSA determines which LPAR it belongs to based on the virtual MAC address
  - The OAT may optionally still play a role in inbound routing decisions by the OSA adapter:
    - If the destination address (192.168.1.1) were not in the home list of this LPAR (and hence not in the OAT for this LPAR), should OSA still send it up to the LPAR or not?
    - You decide via a configuration option
      - *ROUTEALL: send all packets with my VMAC to me*
      - *ROUTELCL: send only packets to me if they are in my HOME list*
- The OAT remains in use for other functions, such as ARP ownership, etc.

OSA-Express virtual MAC while operating in QDIO layer-3 mode

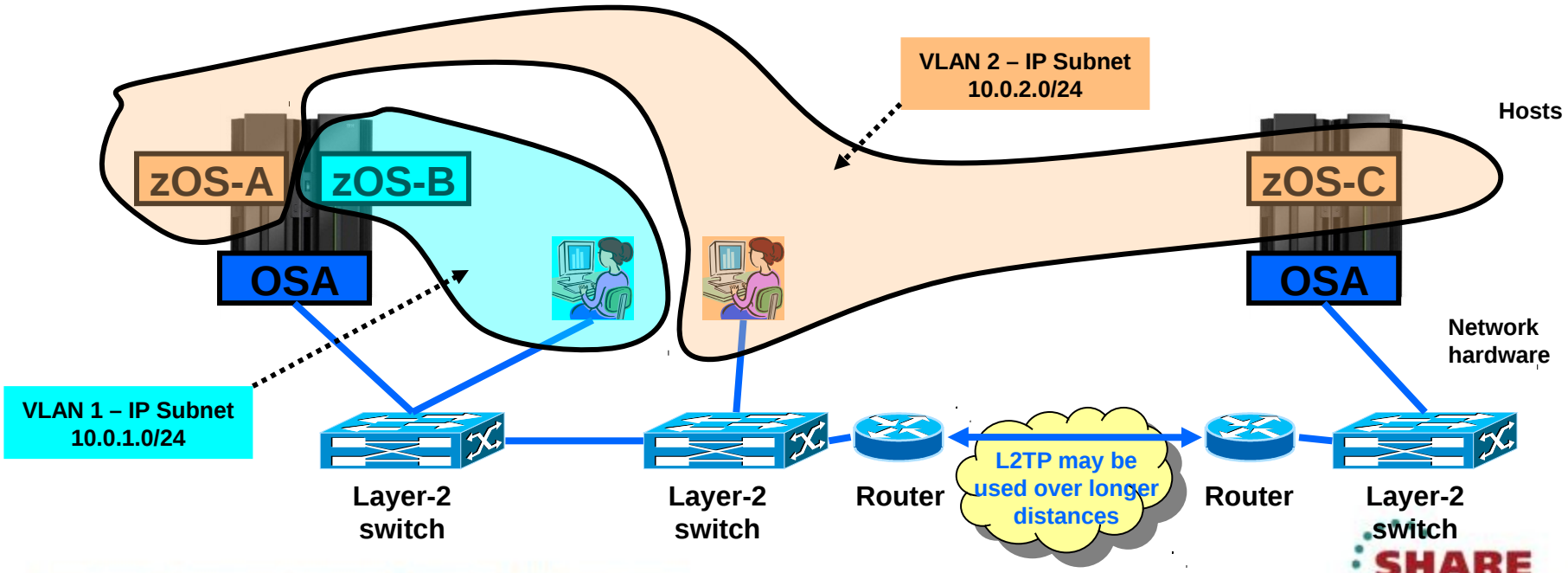
- **OSA MAC sharing problems do not exist if each stack has its own MAC**
  - A "virtual" MAC
  - To the network, each stack appears to have a dedicated OSA port (NIC)
- **MAC address selection**
  - Coded in the TCP/IP profile
  - Generated and assigned by the OSA adapter
- **All IP addresses for a stack are advertised with the virtual MAC**
  - by OSA using ARP for IPv4
  - by the stack using ND for IPv6
- **All external routers now forward frames to the virtual MAC**
  - OSA will “route” to an LPAR/Stack by virtual MAC instead of IP address
  - All stacks can be "routing" stacks instead of 1 PRIROUTER stack
- **Simplifies configuration greatly**
  - No PRIROUTER/SECROUTER!
- **Supported on System z9, z10, z196, and EC12**

No reasons not to use VMACs. Use them!!!

What is a Virtual LAN (a VLAN)?

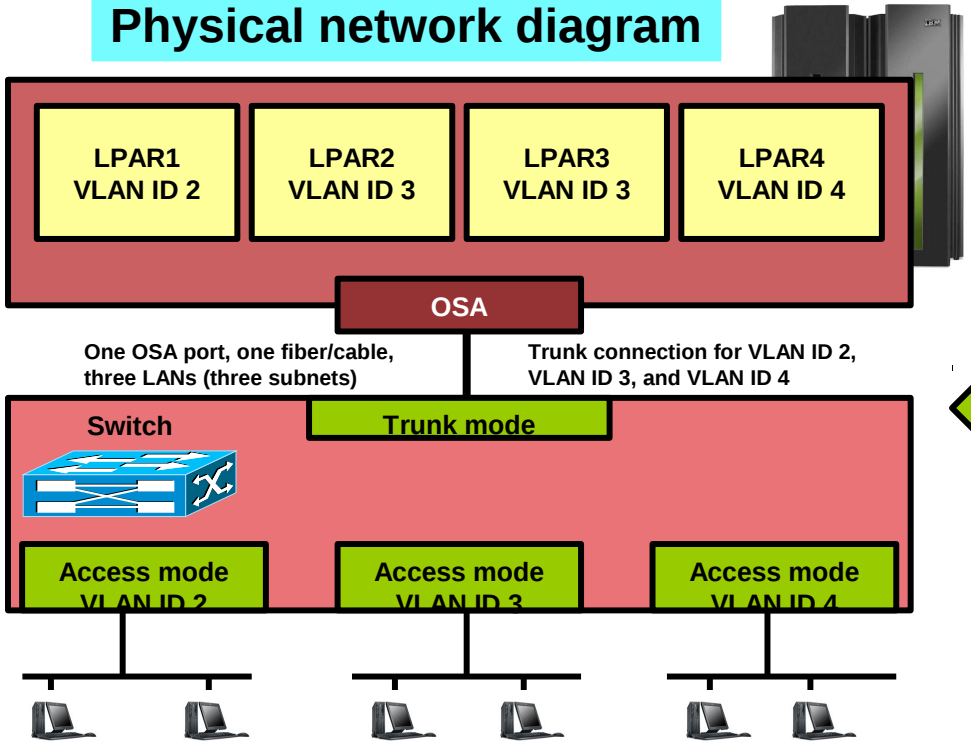
Wikipedia:

- A **virtual LAN**, commonly known as a **VLAN**, is a ***group of hosts*** with a common set of requirements that communicate as if they were ***attached*** to the same ***broadcast domain***, regardless of their physical location.
- A VLAN has the same attributes as a physical LAN, but it allows for end stations to be grouped together even if they are not located on the same network switch.
- Network reconfiguration can be done through software instead of physically relocating devices.

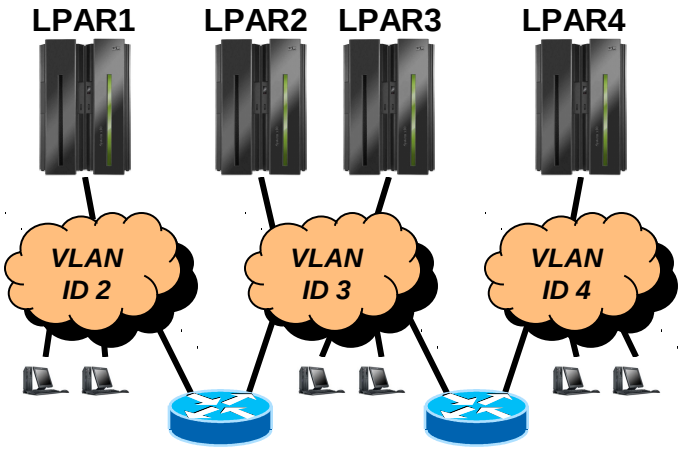


z/OS and VLANs

Physical network diagram



Logical network diagram



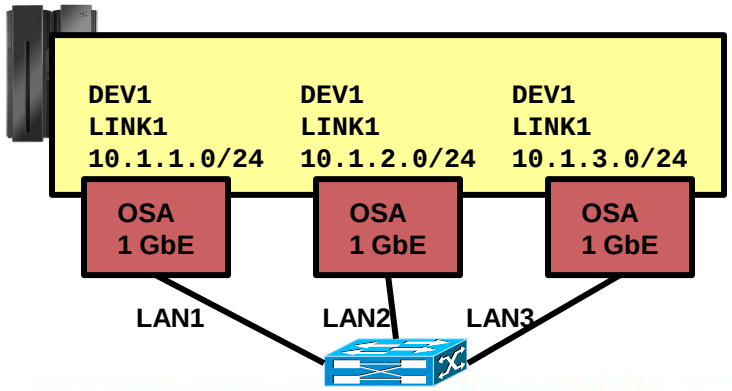
- Depending on switch configuration, the switch may interconnect the VLANs using a layer-3 IP router function.
- The subnets may belong to different routing domains or OSPF areas:
  - Test, production, demo
- The subnets may belong to different security zones:
  - Intranet, DMZ

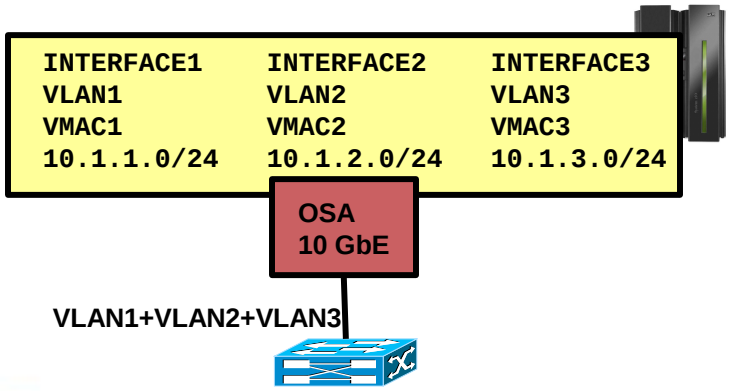
- Each frame on the trunk mode connection carries a VLAN ID in the IEEE802.3 header that allows the network equipment to clearly identify which virtual LAN each frame belongs to.
- On an access mode connection, the switch will transport frames belonging to the configured VLAN ID for that access mode connection only.

VLAN is a LAN media virtualization technology that allows multiple independent IP networks (IP subnets) to share one physical media, such as a cable, an adapter, or a layer-2 switch. Connectivity between VLANs is under control of IP routers.

Multiple network interfaces (VLANs) per OSA port per stack per IP protocol version

- As installations consolidate multiple OSA Gigabit Ethernet ports to a smaller number of 10 Gigabit Ethernet ports, low limits on VLANs has become too restrictive:
  - Not possible to retain existing network interface and IP subnet topology
  - Consolidating multiple LANs to one LAN requires IP renumbering
- z/OS V1R10 added support for eight VLANs per IP protocol per OSA port:
  - Each VLAN on the same OSA port must use unique, non-overlapping IP subnets or prefixes
    - Will be enforced by the TCP/IP stack
  - Each VLAN must be defined using the INTERFACE configuration statement
    - IPv4 INTERFACE statement only supports QDIO interfaces
  - Each VLAN must use layer-3 virtual MAC addresses with ROUTEALL, and each VLAN must have a unique MAC address
- z/OS V1R13 raises the number of VLANs to 32 per IP protocol per OSA port

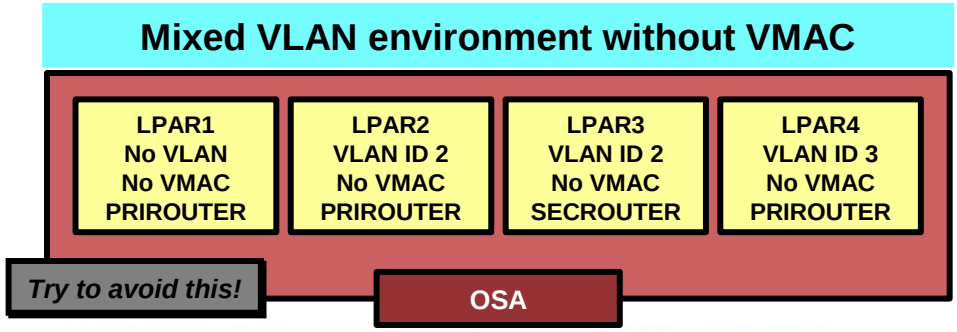




Consolidate multiple low-capacity LAN interfaces to fewer high-capacity LAN interfaces without network topology impact.

A few more words on VLANs with z/OS CS

- z/OS Communications Server supports one VLAN ID per defined network interface
  - This is known as a global VLAN ID
    - Remember: z/OS CS now supports up to 32 interfaces per OSA port – each with its own VLAN ID
  - OSA performs inbound routing based on VLAN IDs in the incoming frames and only sends frames to z/OS with the global VLAN ID z/OS has registered
- Linux on System z supports multiple VLAN IDs per interface
  - OSA sends multiple VLAN IDs up to Linux and Linux then de-multiplexes the different virtual LANs
- Some interfaces that share an OSA port may select to not specify a VLAN ID, while others do specify a VLAN ID (a mixed VLAN environment)
  - **Warning**: be very careful - mixed VLAN environments are not recommended due to complexities with OSA default router selection and other functional issues. Use VLAN IDs on all TCP/IP interfaces that share an OSA port - or not at all.

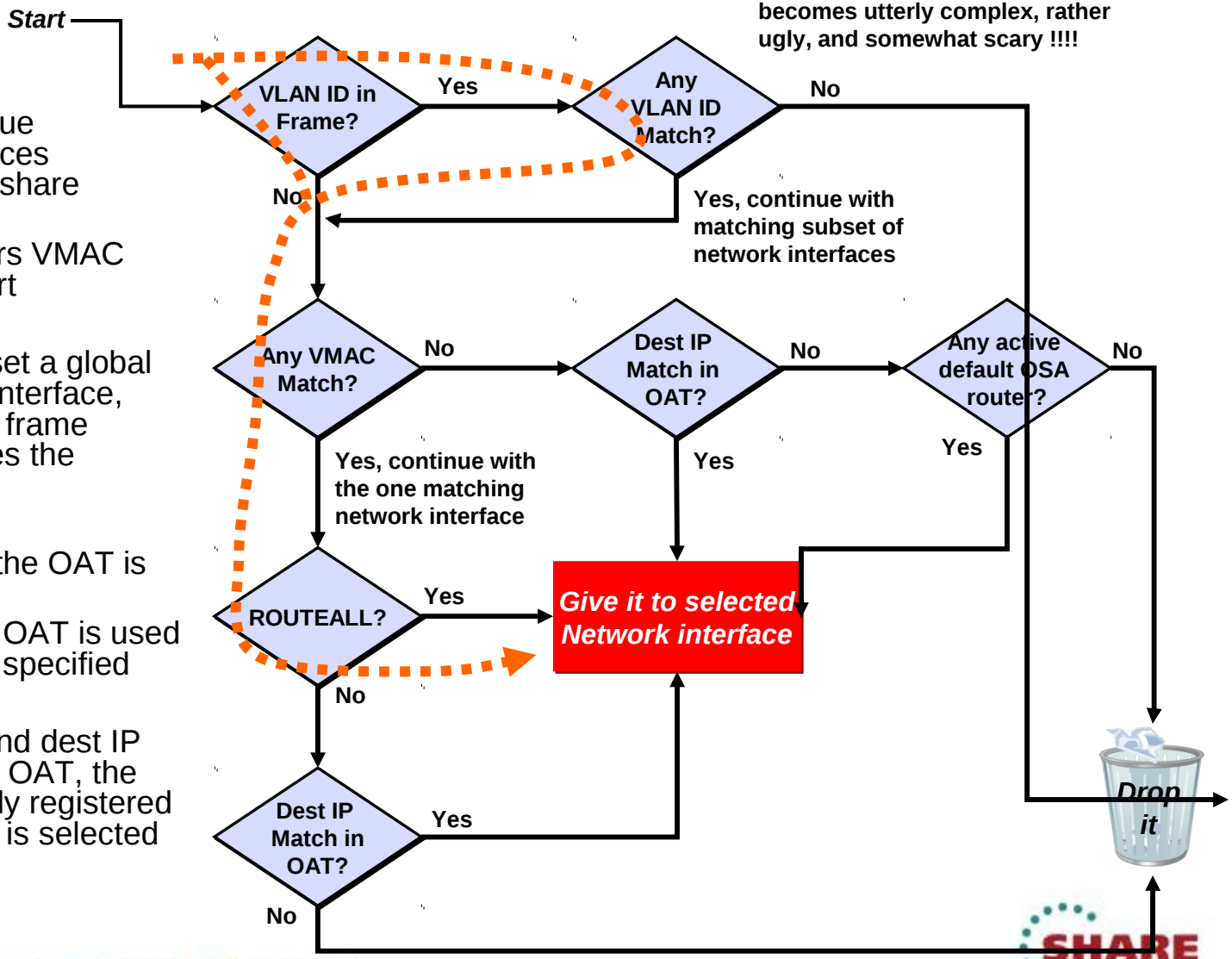


- **LPAR1 acts as PriRouter for:**
  - **Untagged frames**
  - **Frames tagged with a null VLAN ID**
  - **Frames tagged with an unregistered (anything but VLAN 2 or 3) VLAN ID**
  - **Frames tagged with a registered VLAN ID, an unknown destination IP address but no VLAN Pri/SecRrouter**
- **LPAR2 acts as PriRouter for frames tagged with VLAN 2 while LPAR3 acts as SecRouter for that same VLAN.**
- **LPAR4 acts as PriRouter for frames tagged with VLAN 3**

OSA inbound routing as of late 2012

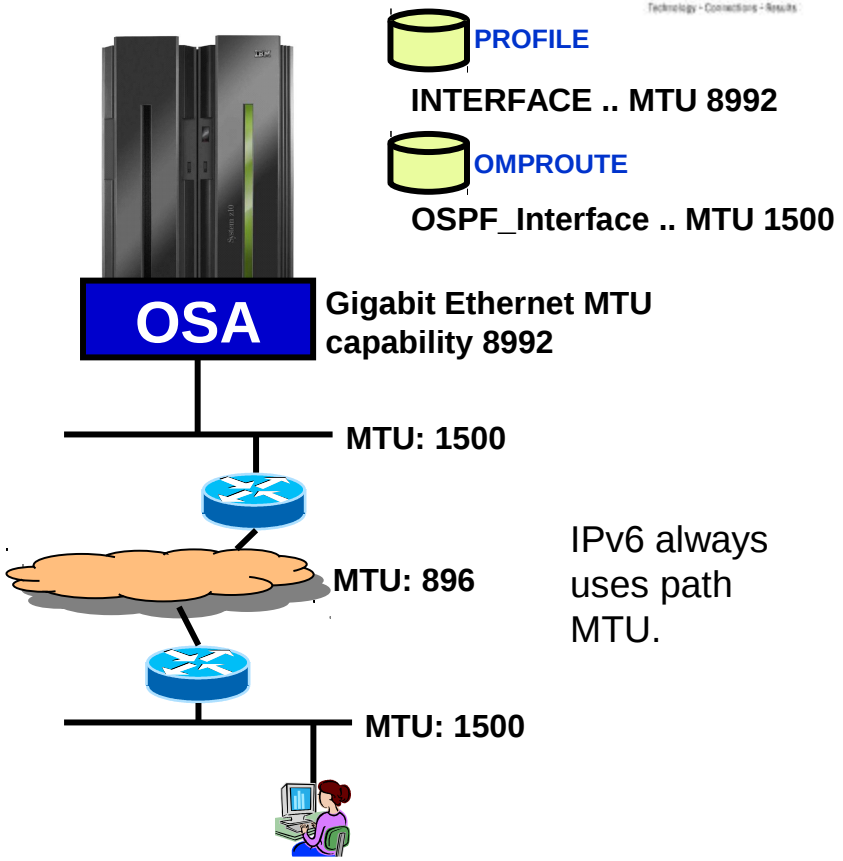
This is correct if all interfaces that share the OSA port use VLAN IDs or none of them do. If some do and others do not, this becomes utterly complex, rather ugly, and somewhat scary !!!!

- **Dest MAC**
  - VMACs are unique across all interfaces and VLANs that share an OSA port
  - z/OS CS registers VMAC with the OSA port
- **VLAN ID**
  - If z/OS CS has set a global VLAN ID for an interface, OSA verifies the frame VLAN ID matches the global VLAN ID
- **Dest IP address**
  - Without VMAC, the OAT is always used
  - With VMAC, the OAT is used if ROUTELCL is specified
- **Default OSA router**
  - If OAT is used and dest IP address is not in OAT, the interface currently registered as default router is selected



This elusive Maximum Transmission Unit – what is the size really?

- **Interface MTU**
  - Configured on the INTERFACE statement or learned from the device
    - For IPv4, OSA reports 1492/8992 – even when you use Ethernet II frames
    - For IPv6, OSA reports 1500/9000 – because IPv6 always uses Ethernet II frames
  - Reported as ActMtu: in a DEVLINKS netstat report
- **Configured route MTU**
  - For static routes: the value coded in the BEGINROUTES section for the destination
  - For dynamic routes: the MTU value from the OMPROUTE interface over which the route was learned
  - If no MTU size defined in OMPROUTE, a default of 576 will be used for IPv4
- **Actual route MTU**
  - Minimum of the interface MTU and the configured route MTU
  - If path MTU is not enabled, this is the MTU that will be used for that route
- **Path MTU**
  - With path MTU enabled, TCP/IP starts out trying to use the actual route MTU and may then lower the MTU to what is discovered
    - With path MTU enabled, all frames are sent with the Do Not Fragment bit
    - Path MTU values are cached, but will timeout after a certain amount of time to accommodate topology changes that allows a higher MTU



IPv6 always uses path MTU.

MTU type	MTU value
Interface MTU	8992
Configured route MTU	1500
Actual route MTU	1500
Path MTU to this destination	896

Some examples on MTU specifications

- MTU 1500 configured on an IPv4 INTERFACE stmt:
  - **`CfgMtu: 1500`** **`ActMtu: 1500`**
  - OSA reports 8992 in this case, but our configured MTU overrides that and determines the actual MTU
- No MTU configured on an IPv4 INTERFACE stmt:
  - **`CfgMtu: None`** **`ActMtu: 8992`**
  - We have no configured MTU to override what OSA reports, so the actual MTU ends up being 8992 as reported by OSA for an IPv4 interface
- No MTU configured on an IPv6 INTERFACE stmt:
  - **`CfgMtu: None`** **`ActMtu: 9000`**
  - Since this is an IPv6 interface, OSA reports the full jumbo frame size of 9000 bytes as the MTU it supports. Since we did not override that with a configured MTU, the 9000 ends up being the actual MTU size.
- In the sample setup, an MTU of 1500 is coded in the OMPROUTE configuration file, so all actual route MTUs end up being 1500
  - **`Default 9.42.105.65 UGO 0000000002 QDIO4A`**
  - **`Metric: 00000001 MTU: 1500`**

Overview of selected recent OSA enhancements

Inbound OSA/QDIO performance: Dynamic LAN idle timer

- Dynamic LAN idle timer is designed to reduce latency and improve network performance by dynamically adjusting the inbound blocking algorithm.
  - When enabled, the z/OS TCP/IP Stack is designed to adjust the inbound blocking algorithm to best match the application requirements.

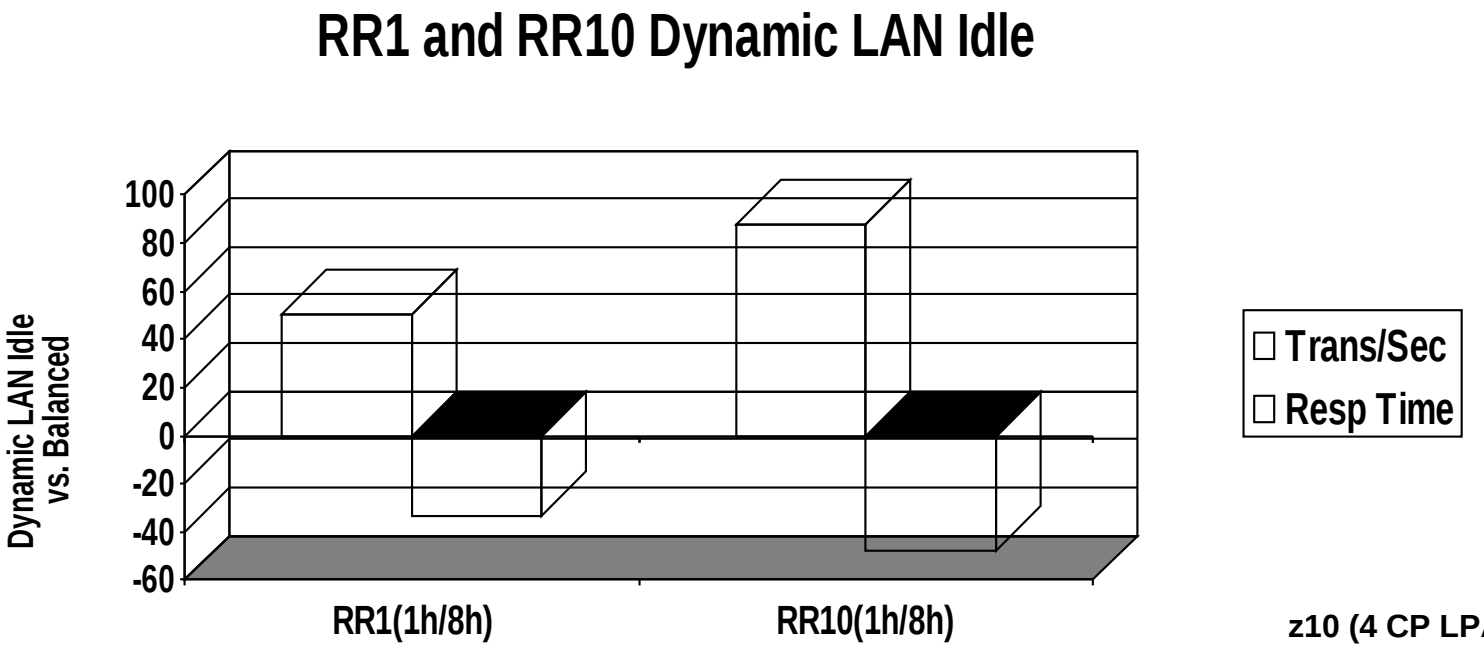
- For latency sensitive applications, the blocking algorithm is modified to be "latency sensitive." For streaming (throughput sensitive) applications, the blocking algorithm is adjusted to maximize throughput. In all cases, the z/OS TCP/IP stack dynamically detects the application requirements, making the necessary adjustments to the blocking algorithm.
  - The monitoring of the application and the blocking algorithm adjustments are made in real-time, dynamically adjusting the application's LAN performance.

- System administrators can authorize the z/OS TCP/IP stack to enable a dynamic setting, which was previously a static setting.
  - The z/OS TCP/IP stack is designed to dynamically determine the best setting for the current running application based on system configuration, inbound workload volume, CPU utilization, and traffic patterns.

- For an OSA Express port in OSD/QDIO mode the z/OS TCP/IP profile **INBPERF** parameter can be specified with one of the following options:
  - **MINCPU** - a static interrupt-timing value, selected to minimize host interrupts without regard to throughput
  - **MINLATENCY** - a static interrupt-timing value, selected to minimize latency
  - **BALANCED** (default) - a static interrupt-timing value, selected to achieve reasonably high throughput and reasonably low CPU
  - **DYNAMIC** - implements the new dynamic LAN idle algorithm ← recommended!

Dynamic LAN Idle Timer: Performance Data

Dynamic LAN Idle improved RR1 TPS 50% and RR10 TPS by 88%. Response Time for these workloads is improved 33% and 47%, respectively.



z10 (4 CP LPARs), z/OS V1R13, OSA-E3 1Gbe

1h/8h indicates 100 bytes in and 800 bytes out

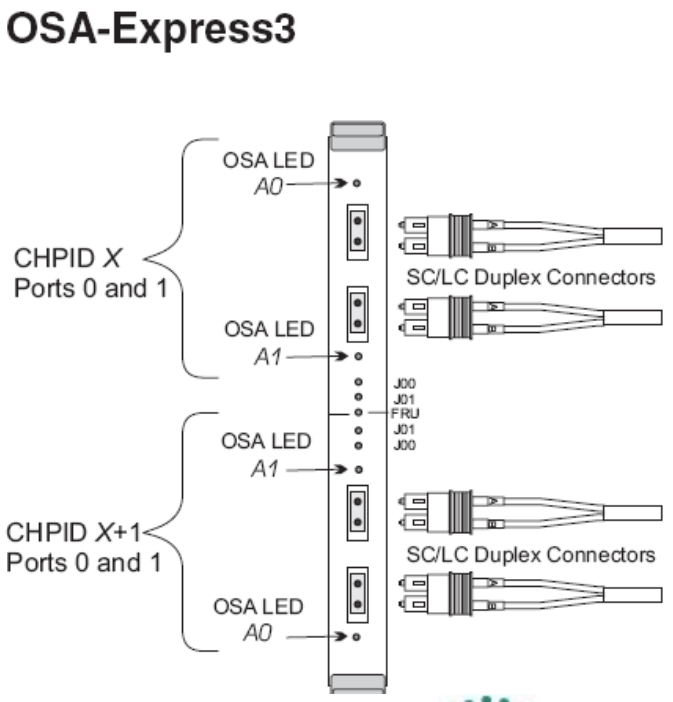
Note: The performance measurements discussed in this presentation are z/OS V1R13 Communications Server numbers and were collected using a dedicated system environment. The results obtained in other configurations or operating system environments may vary.

OSA-Express3 error handling

- Transparent error handling
  - Prior to this support, if one packet within a QDIO read operation was in error, all packets within that entire read operation were discarded as part of error handling.
  - With the new transparent error handling of OSA-Express3, individually packets with errors are marked as invalid.
  - The z/OS Communications Server later discards the packets which are marked as invalid.
  - This new efficiency reduces the number of unnecessary retransmissions which improves overall I/O efficiency.
  - Support for the new "transparent error handling" function is available with z/OS V1R10

OSA-Express3 dual port support by z/OS CS

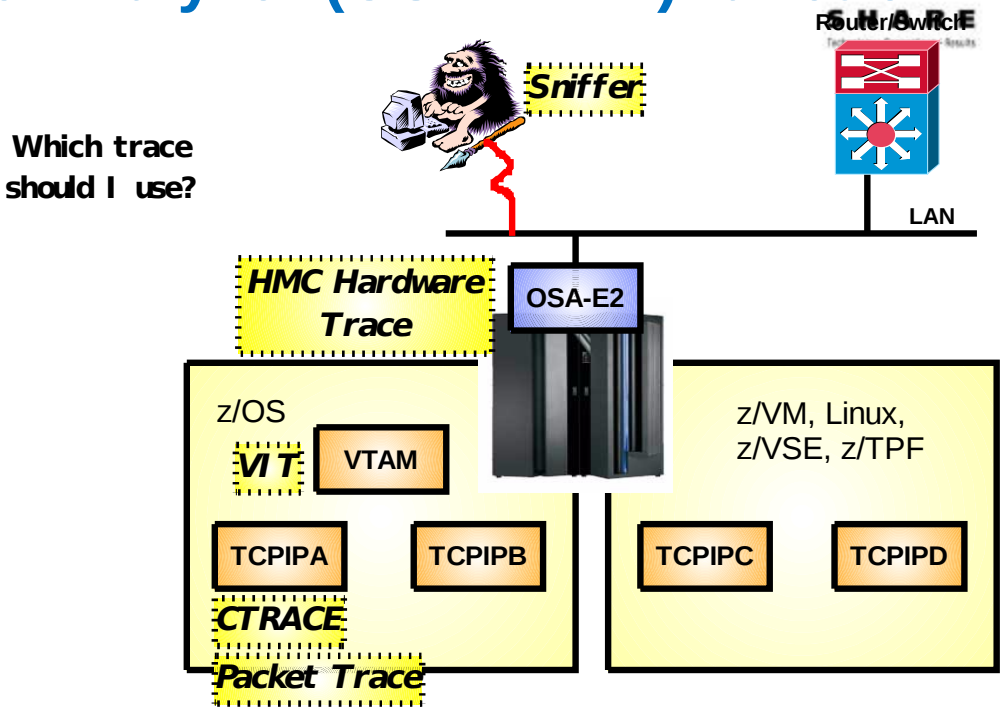
- Without specific z/OS CS support and TRLE PORTNUM definitions, only the two port 0 ports on an OSA-E3 adapter can be used
  - One port 0 per CHPID
- Support for both port 0 and port 1 (the ability to configure PORTNUM in the TRLE) is available with z/OS V1R10



See appendix for sample IOCP, TRLE, and INTERFACE definitions

OSA-Express Network Traffic Analyzer (OSAENTA) function

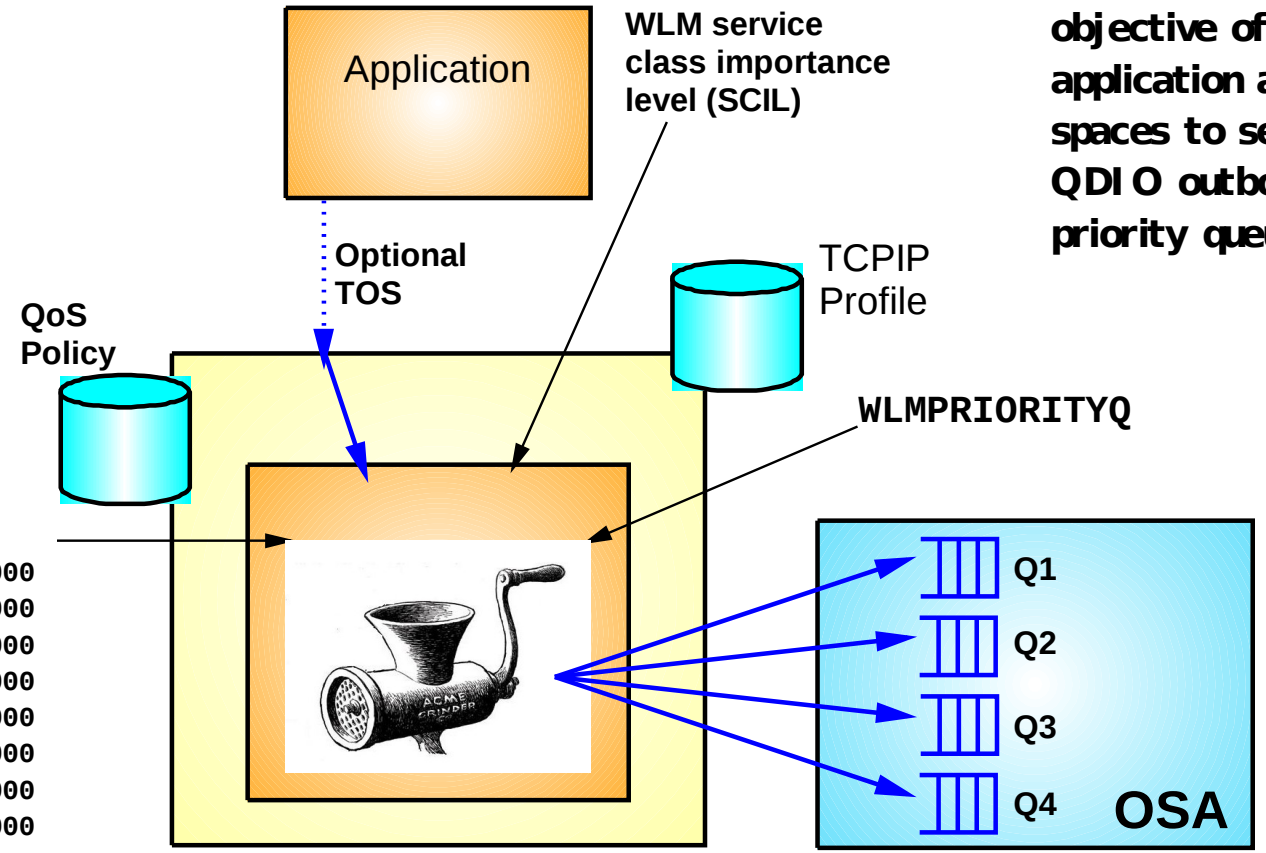
- Diagnosing OSA-Express QDIO problems can be somewhat difficult
  - TCP/IP stack (CTRACE and/or packet trace)
  - VTAM (VIT)
  - OSA (hardware trace)
  - Network (sniffer trace)
- Often not clear where the problem is and which trace(s) to collect
  - Offloaded functions and shared OSAs can complicate the diagnosis
    - ARP frames
    - Segmentation offload



- Supported with OSA-2, OSA-3, and OSA-4
- Allows z/OS Communications Server to collect Ethernet data frames from the OSA adapter
  - Not a sniffer trace (but similar in some aspects)
  - No promiscuous mode
  - Minimizes the need to collect and coordinate multiple traces for diagnosis
  - Minimizes the need for traces from the OSA Hardware Management Console (HMC)
  - Formatted with the standard z/OS CS packet trace formatter
    - Including the ability to convert to Sniffer/Wireshark format

QDIO priority queue selection based upon WLM importance level

Basic principle is that if QoS policies are active, they will determine which priority queue to use.



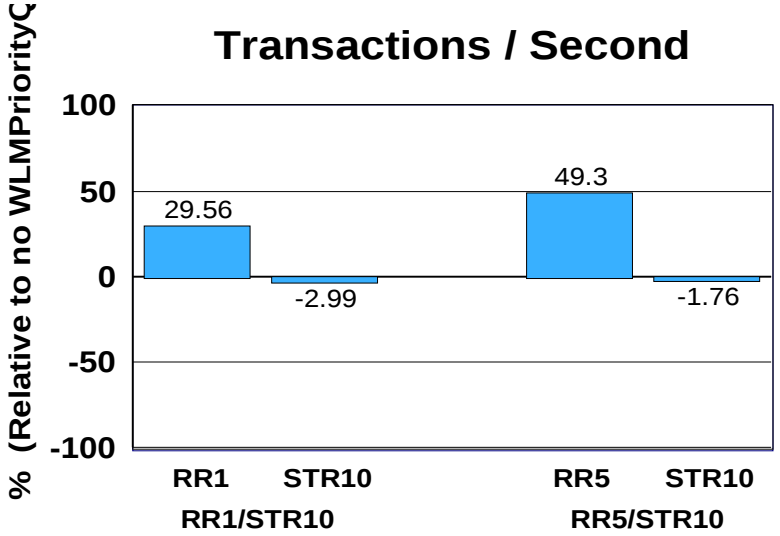
Use workload objective of application address spaces to select QDIO outbound priority queue

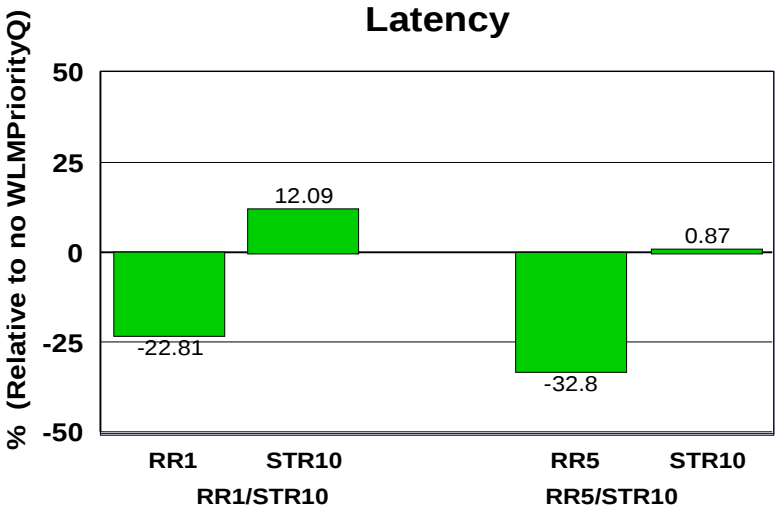
```
SetSubnetPrioTosMask
{
SubnetTosMask 11100000
PriorityTosMapping 1 11100000
PriorityTosMapping 1 11000000
PriorityTosMapping 1 10100000
PriorityTosMapping 1 10000000
PriorityTosMapping 2 01100000
PriorityTosMapping 2 01000000
PriorityTosMapping 3 00100000
PriorityTosMapping 4 00000000
}
```

An easy way to take advantage of QDIO's four outbound priority queues!

OSA Express (QDIO) WLM Outbound Priority Queuing

V1R11





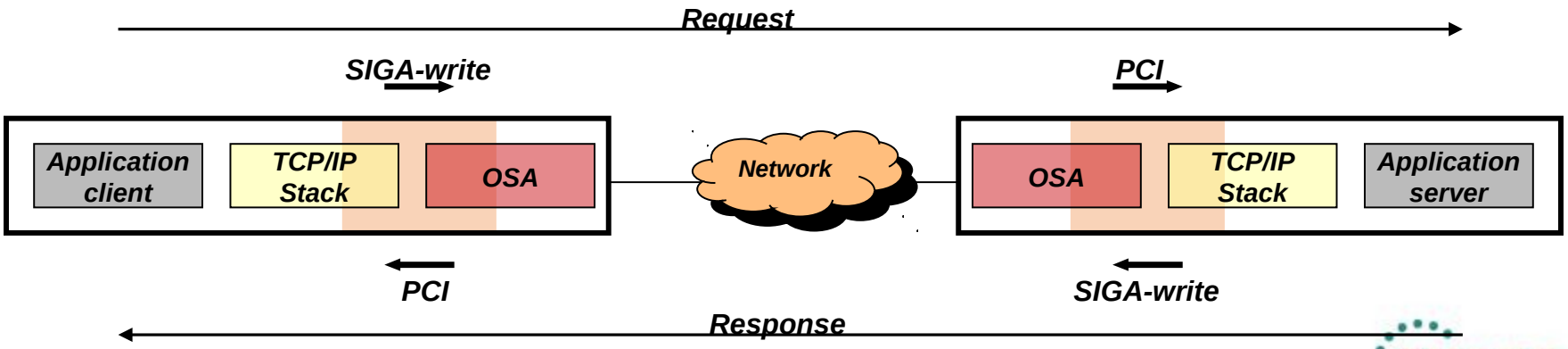
- Request-Response and Streaming mixed workload
- RR1/STR10: 1 RR session, 100 / 800 and 10 STR sessions, 1 / 20 MB
- RR5/STR10: 5 RR sessions, 100 / 800 and 10 STR sessions, 1 / 20 MB
- WLMPRIORITYQ assigned importance level 2 to interactive workloads and level 3 to streaming workloads
- The z/OS Workload Manager (WLM) system administrator assigns each job a WLM service class
- Hardware: z10 using OSA-E2 (1 GbE)
- Software: z/OS V1R11
- z/OS V1R11 with WLM I/O Priority provides 30 to 50% higher throughput for interactive workloads compared to V1R11 without WLM I/O Priority
- z/OS V1R11 with WLM I/O Priority provides 23 to 33% lower latency compared to V1R11 without WLM I/O Priority

Note: The performance measurements discussed in this presentation are preliminary z/OS Communications Server numbers and were collected using a dedicated system environment. The results obtained in other configurations or operating system environments may vary.

V1R11

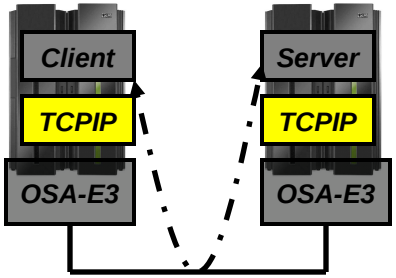
OSA-Express optimized latency mode (OLM)

- OSA-Express3 has significantly better latency characteristics than OSA-Express2
- The z/OS software and OSA microcode can further reduce latency:
  - If z/OS Communications Server knows that latency is the most critical factor
  - If z/OS Communications Server knows that the traffic pattern is not streaming bulk data
- Inbound
  - OSA-Express signals host if data is “on its way” (“Early Interrupt”)
  - Host looks more frequently for data from OSA-Express
- Outbound
  - OSA-Express does not wait for SIGA to look for outbound data (“SIGA reduction”)
- Enabled via PTFs for z/OS V1R11
  - PK90205 (PTF UK49041) and OA29634 (UA49172)



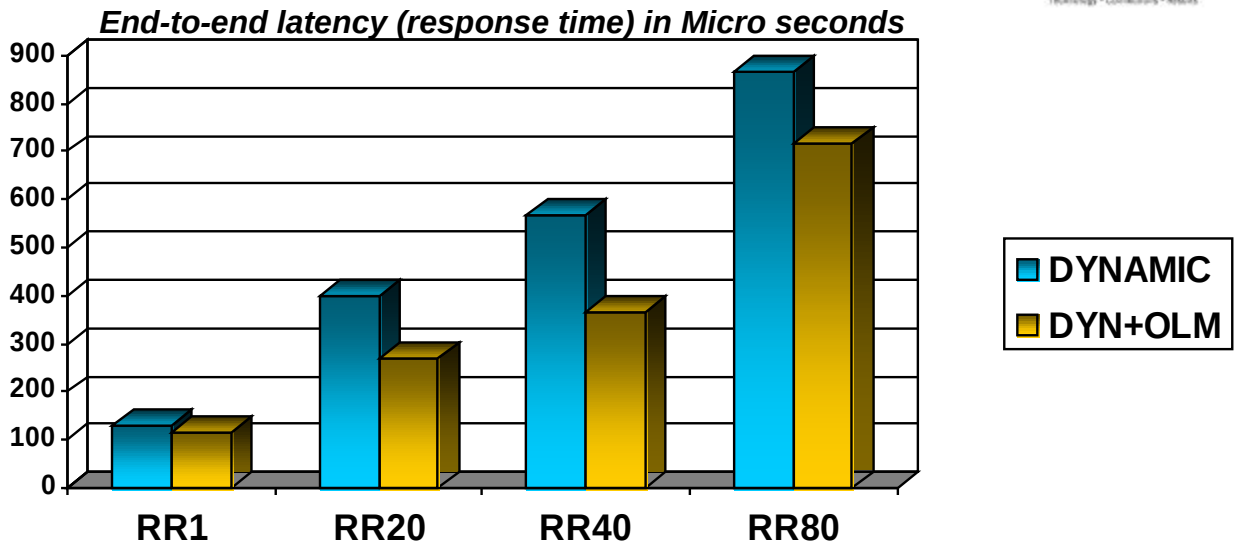
Performance indications of OLM for interactive workloads

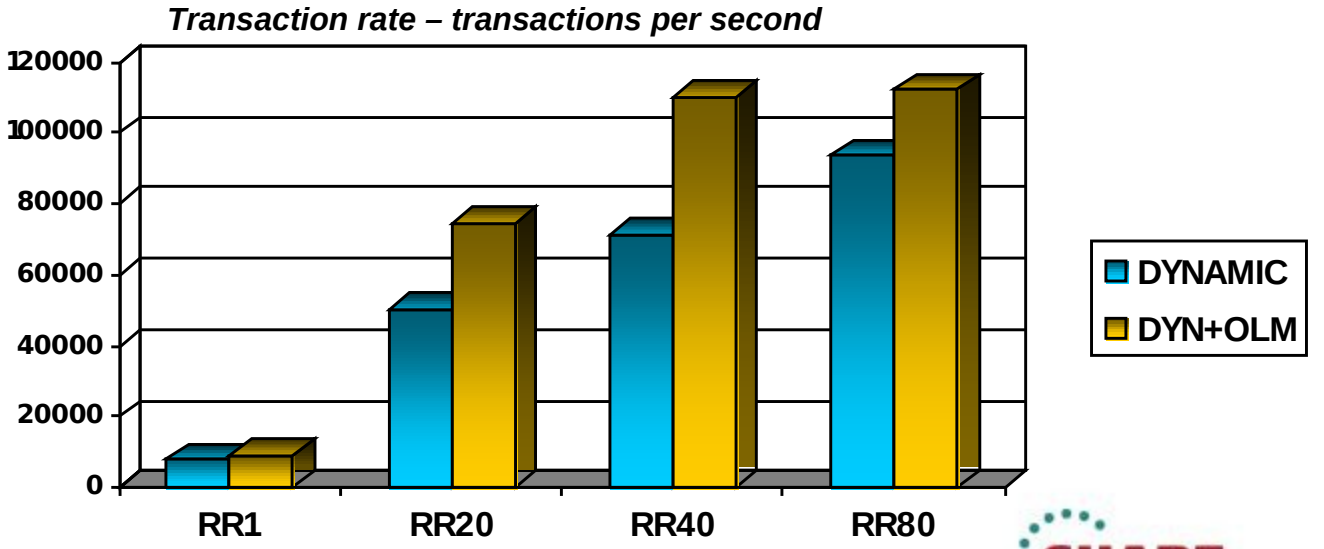
V1R11



z10 (4 CP LPARs), z/OS V1R13, OSA-E3

- Client and Server
  - Has close to no application logic
- RR1
  - 1 session (1 byte in 1 byte out)
- RR20
  - 20 sessions (128 bytes in, 1024 bytes out)
- RR40
  - 40 sessions (128 bytes in, 1024 bytes out)
- RR80
  - 80 sessions (128 bytes in, 1024 bytes out)
- RR20/60
  - 80 sessions (mix of 100/128 bytes in and 800/1024 out)





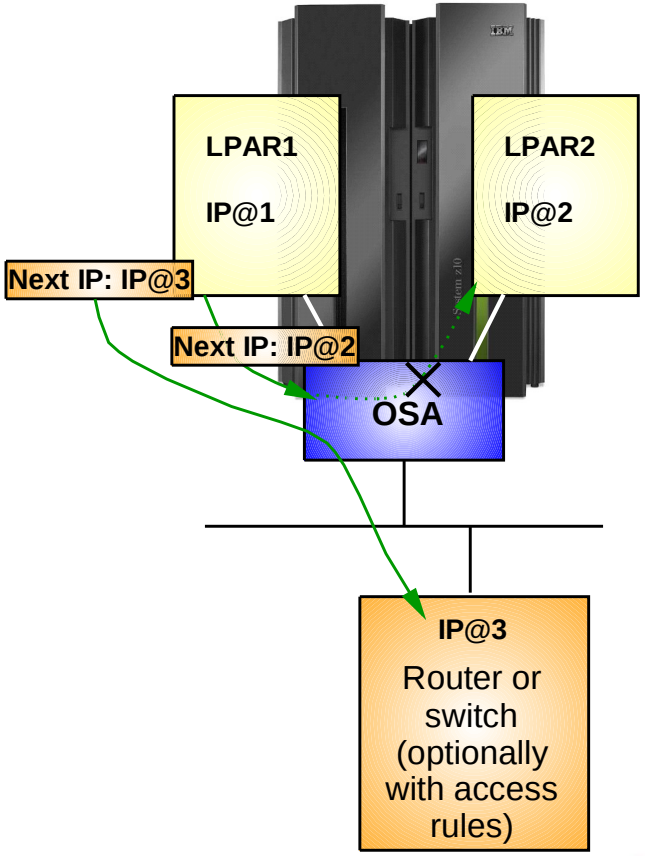
Note: The performance measurements discussed in this presentation are preliminary z/OS Communications Server numbers and were collected using a dedicated system environment. The results obtained in other configurations or operating system environments may vary.

OSA interface isolation

V1R11

- New function added to the OSA adapter
  - z/OS Communications Server adds support for this new function in z/OS V1R11
- Allow customers to disable shared OSA local routing functions
  - ISOLATE/NOISOLATE option on QDIO network INTERFACE definition
- OSA local routing can in some scenarios be seen as a security exposure

Be careful using ISOLATE if you use OSPF and share a subnet between stacks that share an OSA port.

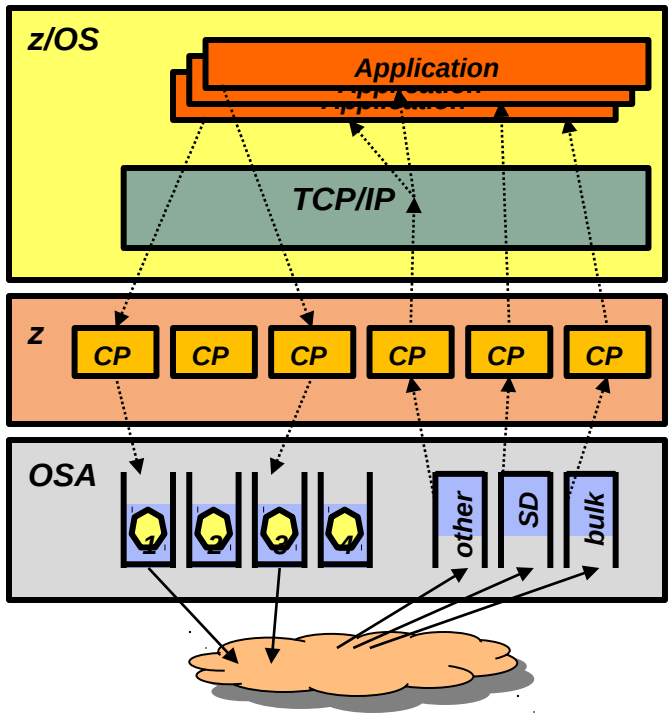


If you enable ISOLATE, packets with a nexthop IP address of another stack that share the OSA port, will be discarded.

OSA multiple inbound queue support: improved bulk transfer and Sysplex Distributor connection routing performance

V1R12

- Allow inbound QDIO traffic separation by supporting multiple read queues
  - "Register" with OSA which traffic goes to which queue
  - OSA-Express Data Router function routes to the correct queue
- Each input queue can be serviced by a separate process
  - Primary input queue for general traffic
  - One or more ancillary input queues (AIQs) for specific traffic types
- Supported traffic types
  - Bulk data traffic queue
    - Serviced from a single process - eliminates any out of order delivery issues
  - Sysplex distributor traffic queue
    - SD traffic efficiently accelerated or presented to target application
  - All other traffic not backed up behind bulk data or SD traffic
- Dynamic LAN idle timer updated per queue

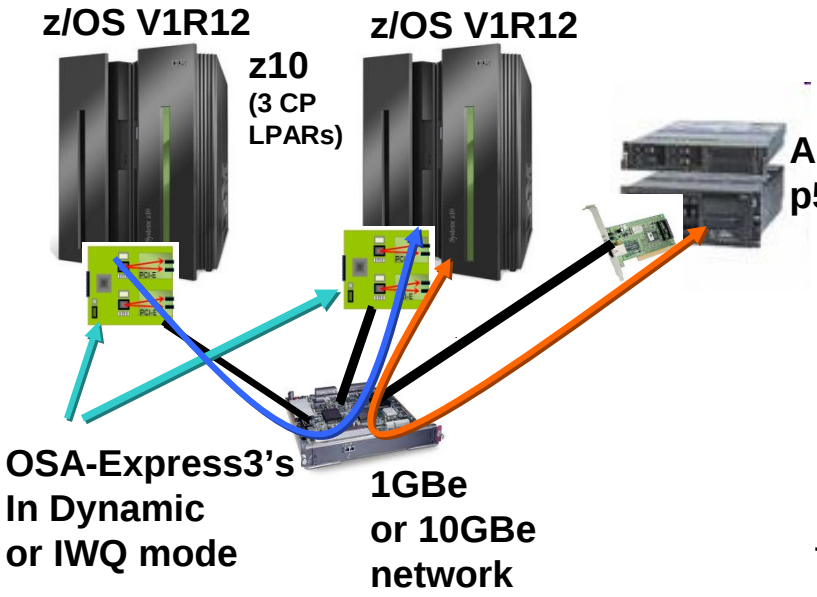


TCP/IP defines and assigns traffic to queues dynamically based on local IP address and port

Bulk traffic
- ***Application predominately sends/receives data in one direction***
- ***Registered per connection (5-tuple)***

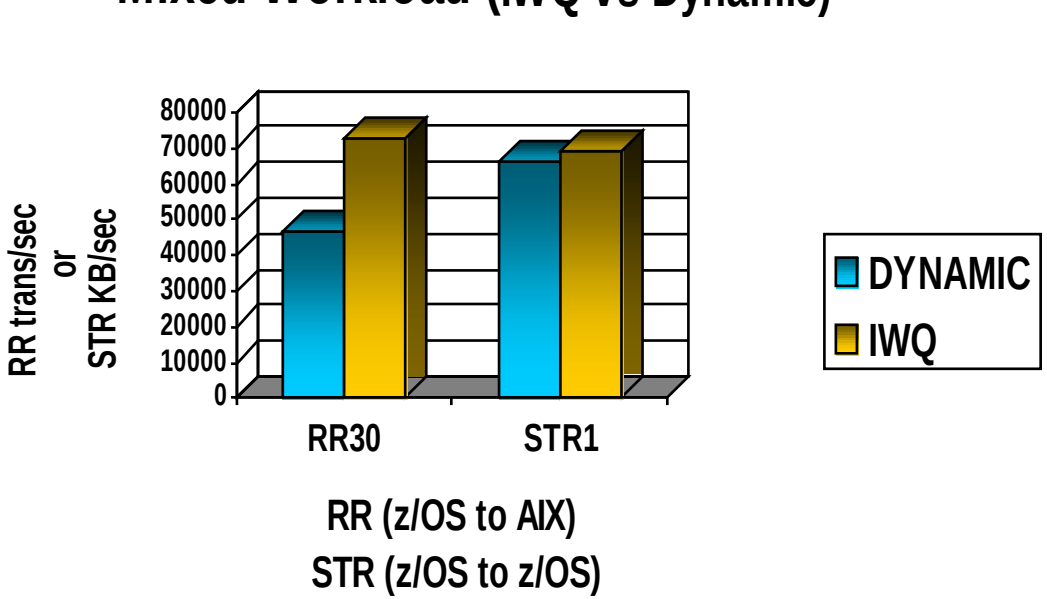
SD traffic
- ***Based on active VIPADISTRIBUTE definitions***
- ***Registered on DVIPA address***

Inbound Workload Queuing: Performance Data



IWQ: Mixed Workload Results vs DYNAMIC:

- **z/OS<->AIX R/R Throughput improved 55% (Response Time improved 36%)**
- **Streaming Throughput also improved in this test: +5%**



For z/OS outbound streaming to another platform, the degree of performance boost (due to IWQ) is relative to receiving platform's sensitivity to out-of-order packet delivery. For streaming INTO z/OS, IWQ will be especially beneficial for multi-CP configurations.

Operator command to query and display OSA information

V1R12

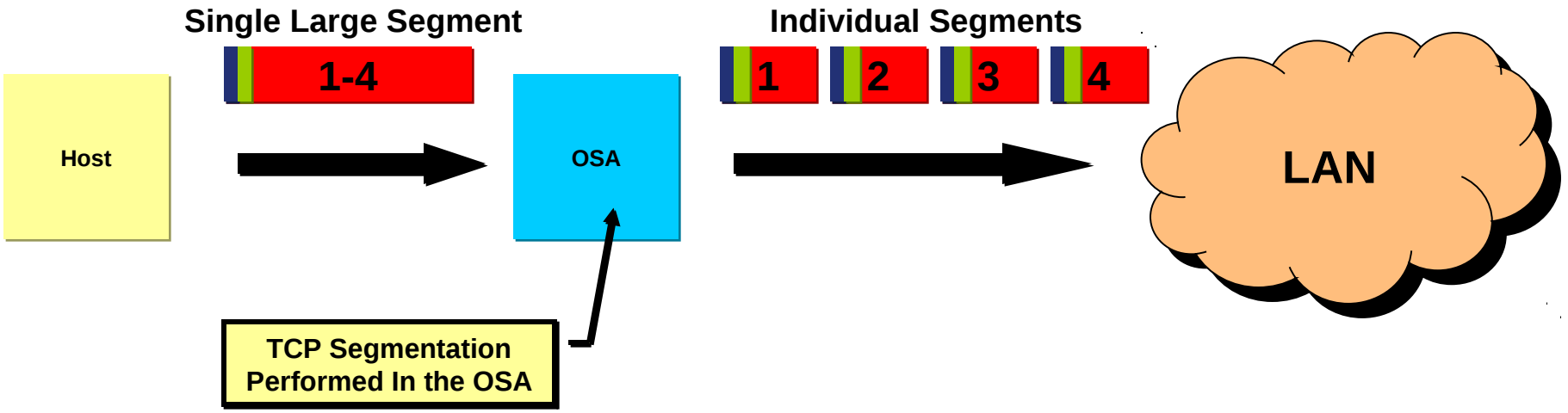
- OSA/SF has been used for years to configure OSA and display the configuration. OSA/SF has played a more central role for OSE devices (pre-QDIO) than for today's OSD devices (QDIO).
- OSD devices exclusively use IPA signals exchanged with the host to enable and configure features and register IP addresses to OSA.
- However, there was no mechanism to display the information directly from OSA without OSA/SF.
- z/OS V1R12 implements a new D TCPIP,,OSAINFO command for use with OSA-3 and OSA-4:
  - Base OSA information
  - OSA address table information
  - Information related to the new multiple inbound queues

```
D TCPIP,,OSAINFO,INTFN=V6O3ETHG0,MAX=100

EZZ0053I COMMAND DISPLAY TCPIP,,OSAINFO COMPLETED SUCCESSFULLY
EZD0031I TCP/IP CS V1R12  TCPIP Name: TCPSVT    15:39:52
Display OSAINFO results for Interface: V6O3ETHG0
PortName: O3ETHG0P  PortNum: 00  DevAddr: 2D64    RealAddr: 0004
PCHID: 0270         CHPID: D6    CHPID Type: OSD  OSA code level: 5D76
Gen: OSA-E3         Active speed/mode: 10 gigabit full duplex
Media: Singlemode Fiber        Jumbo frames: Yes  Isolate: No
PhysicalMACAddr: 001A643B887C  LocallyCfgMACAddr: 000000000000
Queues defined Out: 4  In: 3   Ancillary queues in use: 2
Connection Mode: Layer 3       IPv4: No   IPv6: Yes
SAPSup: 00010293
…
```

TCP Segmentation Offload

- Segmentation consumes (high cost) host CPU cycles in the TCP stack
- New OSA-Express (QDIO mode) feature called Segmentation Offload (also referred to as “Large Send”)
  - Offload most IPv4 TCP segmentation processing to OSA
  - Decrease host CPU utilization
  - Increase data transfer efficiency for IPv4 packets

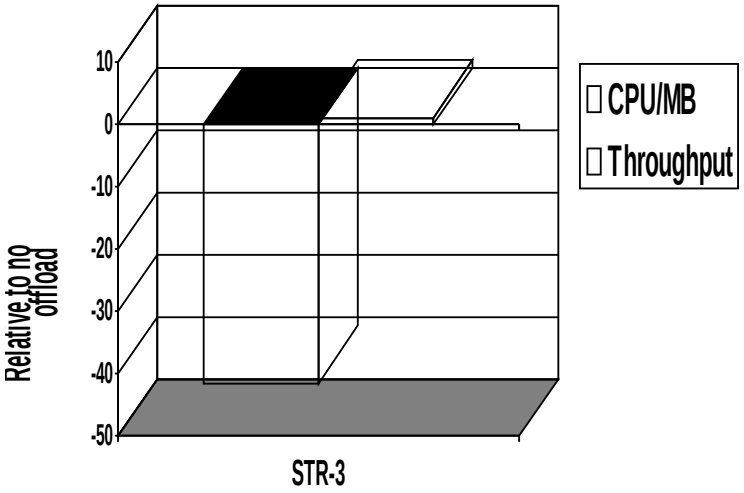


z/OS V1R13 segmentation offload performance measurements

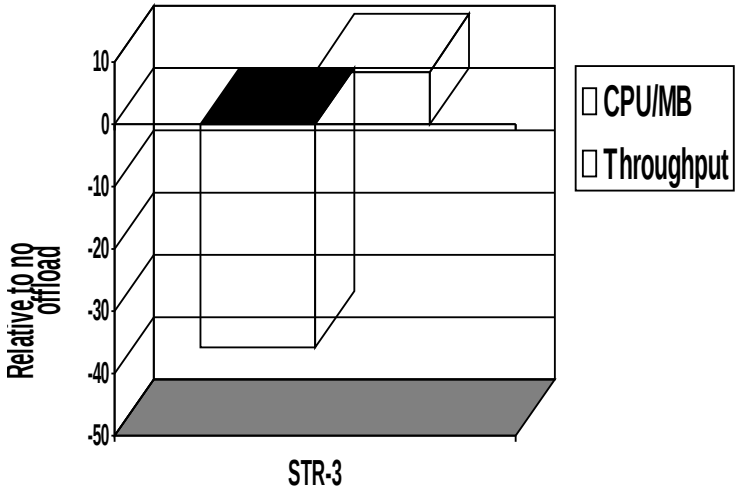
Note: The performance measurements discussed in this presentation were collected using a dedicated system environment. The results obtained in other configurations or operating system environments may vary.

SHARE
Technology · Connections · Results

OSA-Express3 10Gb



OSA-Express4 10Gb



Segmentation offload is generally considered safe to enable at this point in time. Please always check latest PSP buckets for OSA driver levels.

Send buffer size: 180K for streaming workloads

Segmentation offload may significantly reduce CPU cycles when sending bulk data from z/OS!

OSA Express 4 Enhancements

- Improved on-card processor speed and memory bus provides better utilization of 10GB network
- Enterprise Extender inbound workload queue provides internal optimizations
  - EE traffic processed quicker
  - Avoids memory copy of data
- Checksum Offload support for IPv6 traffic
- Segmentation Offload support for IPv6 traffic

Please fill out your session evaluation

- Getting the most out of your OSA Adapter with z/OS Comm Server
- Session # 12862
- QR Code:

IBM Comm Server

Find us on Facebook at
http://www.facebook.com/IBMCommserver

Follow us on Twitter at
http://www.twitter.com/IBM_Commserver

Read the z/OS Communications Server blog at
http://tinyurl.com/zoscsblog

Visit the z/OS CS YouTube channel at
http://www.youtube.com/user/zOSCommServer

Appendix A:
OSA-3 dual port definitions

OSA-Express3 IOCP definitions

```
CHPID PATH=(CSS(0,1),FD),SHARED,                                    *
      PCHID=581,TYPE=OSD

CNTLUNIT CUNUMBR=2FD0,PATH=((CSS(0),FD),(CSS(1),FD)),UNIT=OSA

IODEVICE  ADDRESS=(2FD0,14),CUNUMBR=2FD0,UNIT=OSA,                  *
      UNITADD=00
IODEVICE  ADDRESS=(2FDE,1),CUNUMBR=2FD0,UNIT=OSAD,                  *
      UNITADD=FE
```

```
10.37.39  d m=chp(fd)
      10.37.39  IEE174I 10.37.39 DISPLAY M 825                                C
      CHPID FD:  TYPE=11, DESC=OSA DIRECT EXPRESS, ONLINE
      DEVICE STATUS FOR CHANNEL PATH FD
           0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
      02FD +  +  +  +  +  +  +  +  +  +  +  +  +  +  +  .
      SWITCH DEVICE NUMBER = NONE
      PHYSICAL CHANNEL ID = 0581
      ************************ SYMBOL EXPLANATIONS ************************
      + ONLINE    @ PATH NOT VALIDATED   - OFFLINE    . DOES NOT EXIST
      * PHYSICALLY ONLINE   $ PATH NOT OPERATIONAL
```

OSA Express3 TRLE definitions

```
O3ETHB0T TRLE  LNCTL=MPC,                                    X
               READ=(2FD0),                                  X
               WRITE=(2FD1),                                 X
               MPCLEVEL=QDIO,                                X
               DATAPATH=(2FD2,2FD3,2FD4,2FD5),               X
               PORTNAME=(O3ETHB0P),                          X
               PORTNUM=(0)

O3ETHB1T TRLE  LNCTL=MPC,                                    X
               READ=(2FD6),                                  X
               WRITE=(2FD7),                                 X
               MPCLEVEL=QDIO,                                X
               DATAPATH=(2FD8,2FD9,2FDA,2FDB),               X
               PORTNAME=(O3ETHB1P),                          X
               PORTNUM=(1)
```

OSA Express3 TCP/IP Interface definitions

```
INTERFACE O3ETHB0 DEFINE IPAQENET
 PORTNAME O3ETHB0P
 IPADDR 16.11.16.105/20
 SOURCEVIPAINT LGEVIPA1
 MTU 1500
 VLANID 3
 READSTORAGE GLOBAL
 INBPERF BALANCED
 IPBCAST
 MONSYSPLEX
 DYNVLANREG
 OLM
 VMAC 0204100B1069 ROUTEALL

INTERFACE O3ETHB1 DEFINE IPAQENET
 PORTNAME O3ETHB1P
 IPADDR 16.11.17.105/20
 SOURCEVIPAINT LGEVIPA1
 MTU 1500
 VLANID 3
 READSTORAGE GLOBAL
 INBPERF BALANCED
 IPBCAST
 MONSYSPLEX
 DYNVLANREG
 OLM
 VMAC 0204100B1169 ROUTEALL
```