Improving OSA Performance with z/OS Communications Server

Hugh Hockett – hhockett@us.ibm.com
Jerry Stevens – sjerry@us.ibm.com
IBM Raleigh, NC

Sept 21 2010

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®
- DB2 Connect
- DB2®
- DRDA®
- e-business on demand®
- e-business (logo)
- e business(logo)®
- ESCON®
- FICON®
- GDDM®
- HiperSockets
- HPR Channel Connectivity
- HyperSwap
- i5/OS (logo)
- i5/OS®
- IBM (logo)®
- IBM®
- IMS
- 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®
- PR/SM
- pSeries®
- RACF®
- Rational Suite®
- Rational®
- Redbooks
- Redbooks (logo)
- Sysplex Timer®
- System i5
- System p5
- System x
- System z
- System z9
- Tivoli (logo)®
- Tivoli®
- VTAM®
- WebSphere®
- xSeries®
- z9
- zSeries®
- z/Architecture
- z/OS®
- z/VM®
- z/VSE

- 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.
- Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
- Red Hat is a trademark of Red Hat, Inc.
- SUSE® LINUX Professional 9.2 from Novell®
- Other company, product, or service names may be trademarks or service marks of others.
- This information is for planning purposes only. The information herein is subject to change before the products described become generally available.
- 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.

All performance data contained in this publication was obtained in the specific operating environment and under the conditions described and is presented as an illustration. Performance obtained in other operating environments may vary and customers should conduct their own testing.

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

Agenda

- Introduction to zCS and OSA Performance
- OSA-Express3 Highlights
- INTERFACE Statement
- Dynamic LAN Idle Timer
- Optimized Latency Mode
- WLM Priority Queuing
- Segmentation Offload
- QDIO Accelerator
- QDIO Inbound Workload Queueing (V1R12)
- zCS Performance Summaries Online

- Appendix A: HiperSockets Overview & zIIP Assisted HiperSockets Multiple Write
- Appendix B: Virtual MAC Addrs (VMACs)

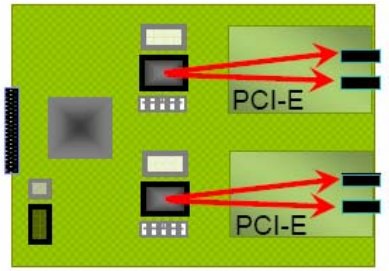
Disclaimer: All statements regarding IBM future direction or intent, including current product plans, are subject to change or withdrawal without notice and represent goals and objectives only. All information is provided for informational purposes only, on an "as is" basis, without warranty of any kind.

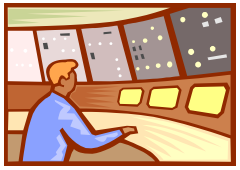
Introduction

- OSA performance has improved over the years from both a hardware and software perspective
- New OSA-Express3 hardware is smarter and faster than previous OSAs
- z/OS Communications Server has made a number of OSA related enhancements to improve performance
  - Latency Improvements
  - Throughput Improvements
  - Reduced CPU utilization
    - Offloading processing
    - Accelerating forwarded traffic
- z/OS Communications Server has also made improvements to simplify OSA configuration and network topologies

OSA-Express3 Highlights

- New generation of OSA-Express features
  - New hardware data router bypasses firmware for packet construction, inspection, routing, etc
  - New microprocessor (660 MHz versus 500/448 MHz)
  - New PCI bus (PCI Express)
  - New LC Duplex SM connectors for 10 Gbe feature
  - Dual density adapters
    - Up to four ports per feature, two ports per CHPID
  - Up to 45% improvement in latency over OSA-Express2
  - 4x improvement over OSA-Express2 for 10g Ethernet feature (line speed)
- Available only on the IBM System z10 platform
- http://www-03.ibm.com/systems/z/hardware/networking/features.html

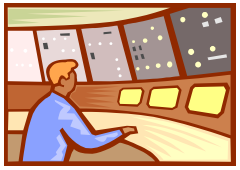




Improving OSA Performance with z/OS Communications Server

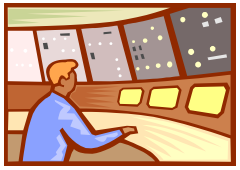
INTERFACE Statement

INTERFACE Statement



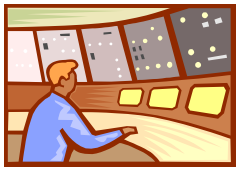
- Based on the IPv6 INTERFACE statement
- Added for IPv4 (IPAQENET) in V1R10 (to support multiple VLANs)
- Improves usability
  - Combines function of DEVICE/LINK/HOME into one statement
  - Easier to add/modify/delete
  - Improved source VIPA specification
  - Provides control over VIPA ARP processing
- Many new features are configurable on the INTERFACE statement only
- Separate datapath device for each INTERFACE statement (IPv4 and IPv6)
- To convert a DEVICE/LINK/HOME to an INTERFACE statement (V1R12):
  - IP Configuration Guide has cookbook style steps
  - CONVERT parameter on the TCPIPCS PROFILE subcommand

INTERFACE Statement: IPv4 Source VIPA



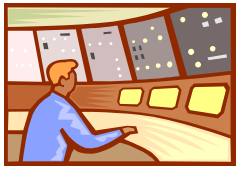
- With IPv4 DEVICE/LINK the order of the home list controls the source VIPA selection
- With IPv4 INTERFACE statement the stack uses IP address of the VIPA specified on SOURCEVIPAINTERFACE parameter
  - Similar to IPv6 source VIPA

INTERFACE Statement: Control VIPA ARP processing



- DEVICE/LINK processing for QDIO ARP offload
  - Stack tells OSA to ARP for all VIPAs in home list
  - Results in many unnecessary gratuitous ARPs
  - Can cause confusion in routers and sniffer traces
- INTERFACE statement with subnet mask
  - Stack tells OSA to only perform ARP processing for VIPAs in the same subnet
  - Eliminates unnecessary gratuitous ARPs

INTERFACE Statement: IPv4 example



Example of DEVICE/LINK/HOME statements:

```
DEVICE QDIO4101  MPCIPA   PRIROUTER
LINK   QDIO4101L IPAQENET     QDIO4101
  INBPERF DYNAMIC
;
HOME
  172.16.1.1     QDIO4101L
```

Example of an INTERFACE statement after conversion:

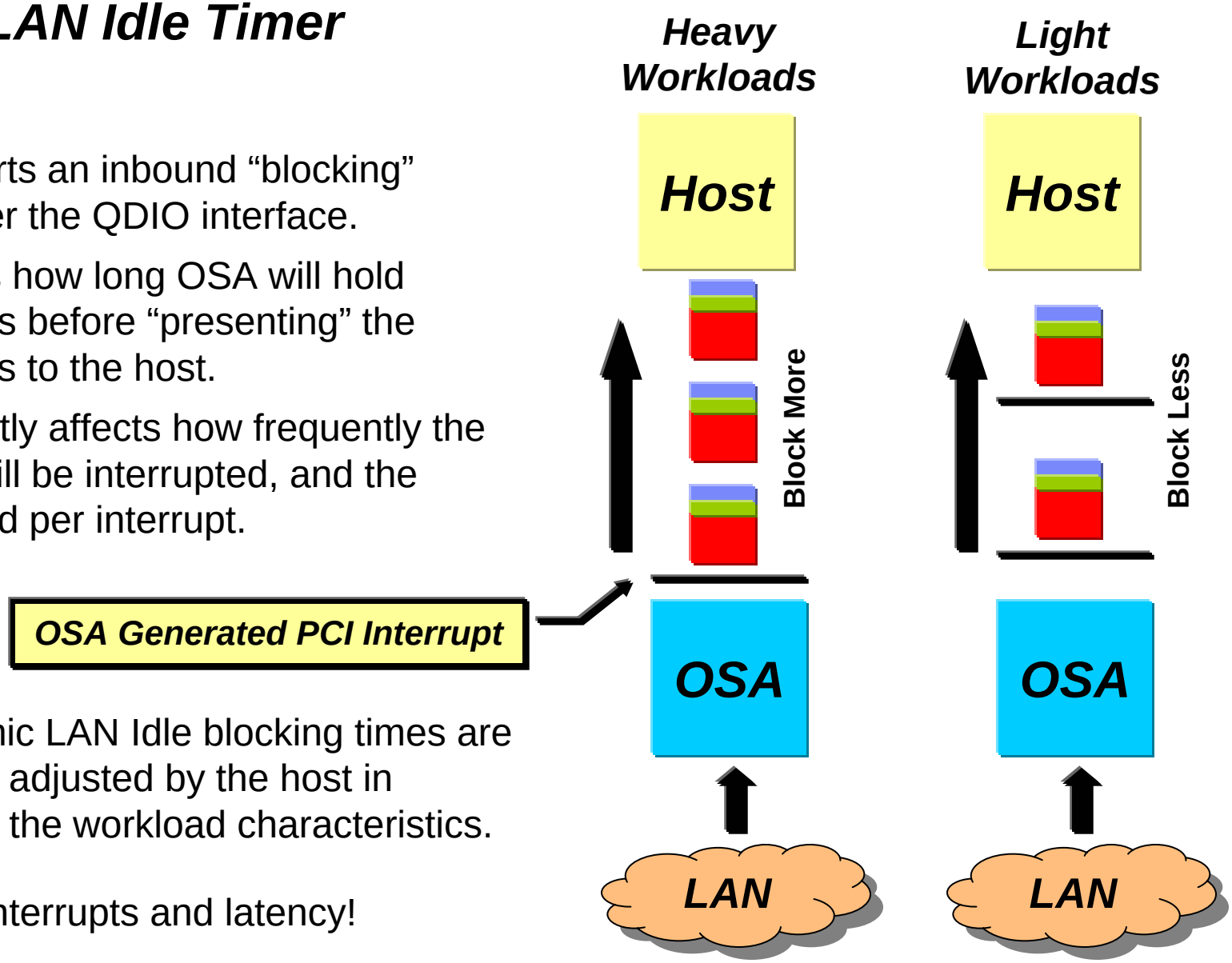
```
INTERFACE QDIO4101L
  DEFINE IPAQENET
  IPADDR 172.16.1.1/24
  PORTNAME QDIO4101
  INBPERF DYNAMIC
  PRIROUTER
```

Improving OSA Performance with z/OS Communications Server

Dynamic LAN Idle Timer (INBPERF DYNAMIC)

Dynamic LAN Idle Timer

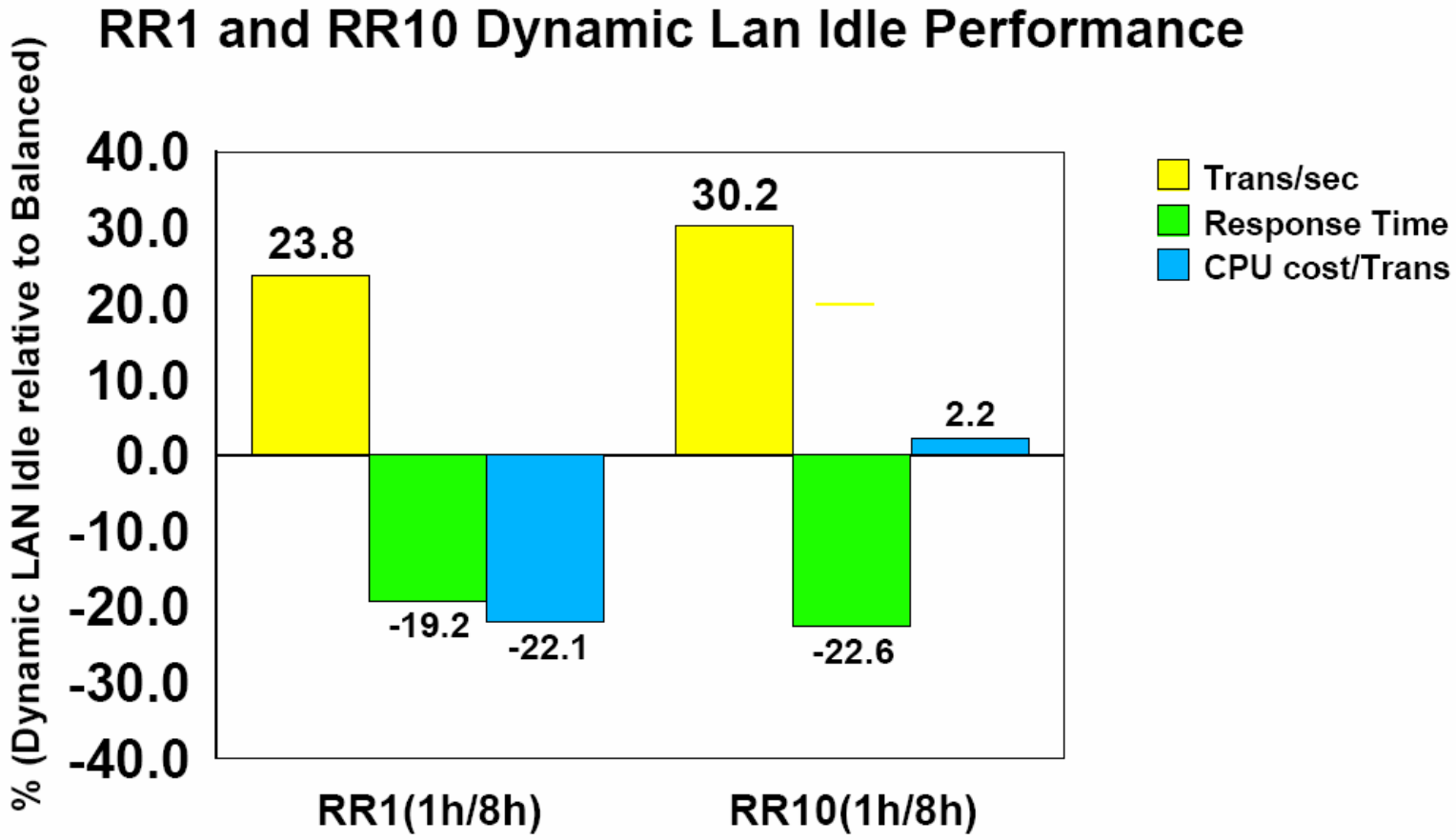
- OSA supports an inbound “blocking” function over the QDIO interface.
  - Affects how long OSA will hold packets before “presenting” the packets to the host.
  - Indirectly affects how frequently the host will be interrupted, and the payload per interrupt.



- With Dynamic LAN Idle blocking times are dynamically adjusted by the host in response to the workload characteristics.
- Optimizes interrupts and latency!

Dynamic LAN Idle Timer: Performance

- For RR1, the transactions per second is improved by 23.8% and for RR10 it is improved by 30.2%.



- 1h/8h indicates 100 bytes In and 800 bytes out

Dynamic LAN Idle Timer: Configuration

- Configure INBPERF DYNAMIC on the INTERFACE statement

```
>>-INTERFace--intf_name----------------------------------------->
  .
    .-INBPERF BALANCED--------.
>--+-------------------------+-------->
    '-INBPERF--+-DYNAMIC----+-'
               +-MINCPU-----+
               '-MINLATENCY-'
  .
```

  - **BALANCED** (default) - a static interrupt-timing value, selected to achieve reasonably high throughput and reasonably low CPU
  - **DYNAMIC** - a dynamic interrupt-timing value that changes based on current inbound workload conditions ⬅ ***Generally Recommended!***
  - **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

  Note: These values cannot be changed without stopping and restarting the interface

Dynamic LAN Idle Timer: Display Configuration

- Use Netstat DEvlinks/-d to display the current INBPERF setting

```
D TCPIP,TCPDLI41,NETSTAT,DEV
.
DEVNAME: GBNS41            DEVTYPE: MPCIPA
  DEVSTATUS: READY
  LNKNAME: LGBNS41           LNKTYPE: IPAQENET   LNKSTATUS: READY
    NETNUM: N/A  QUESIZE: N/A  SPEED: 0000001000
    IPBROADCASTCAPABILITY: NO
    CFGROUTER: PRI                  ACTROUTER: PRI
    ARPOFFLOAD: YES                 ARPOFFLOADINFO: YES
    ACTMTU: 8992
    READSTORAGE: GLOBAL (4096K)     INBPERF: DYNAMIC
    CHECKSUMOFFLOAD: YES            SEGMENTATIONOFFLOAD: YES
    SECCLASS: 255                   MONSYSPLEX: NO
  BSD ROUTING PARAMETERS:
    MTU SIZE: N/A               METRIC: 00
    DESTADDR: 0.0.0.0           SUBNETMASK: 255.255.255.0
  .
```

Dynamic LAN Idle Timer

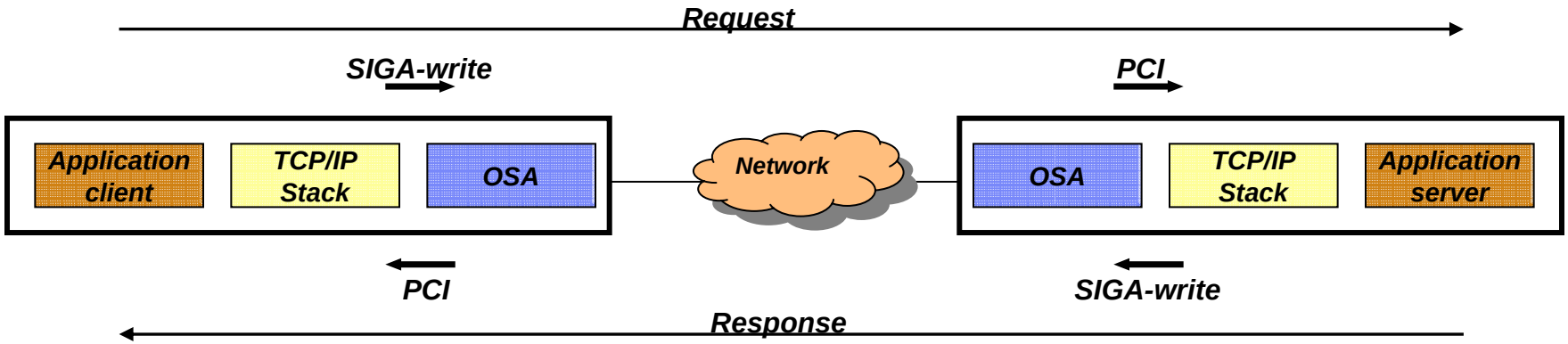
- Static LAN idle timer settings can contribute to network latency on zSeries
  - Even when INBPERF MINLATENCY is specified the inter-packet gap timer is still set to 20 microseconds
- When INBPERF DYNAMIC is specified the stack will dynamically tune the LAN Idle timer values to reflect current workload characteristics
  - The inter-packet gap time can now be reduced as small as a microsecond
- Should see a throughput improvement for interactive workloads
- For streaming workloads the operating characteristics should be similar to the INBPERF parameter value of BALANCED (current default)
- Added in z/OS Communications Server V1R9 for OSA-E2 and E3

Improving OSA Performance with z/OS Communications Server

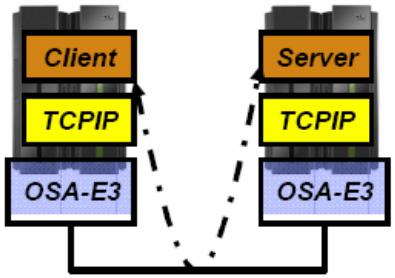
Optimized Latency Mode (OLM)

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
  - Dynamically adjusting “blocking” times in OSA (similar to Dynamic LAN Idle Timer)
- Outbound
  - OSA-Express does not wait for SIGA to look for outbound data (“SIGA reduction”)



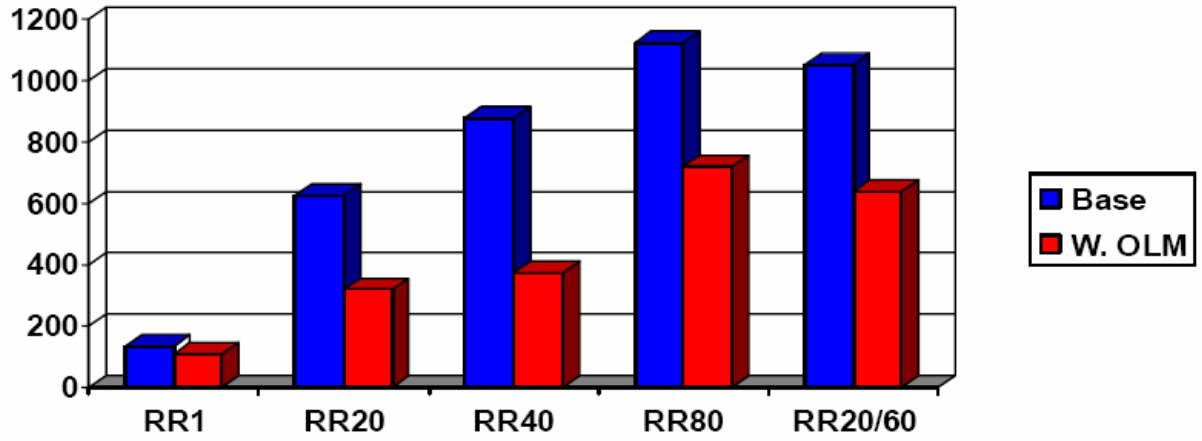
Optimized Latency Mode (OLM): Performance



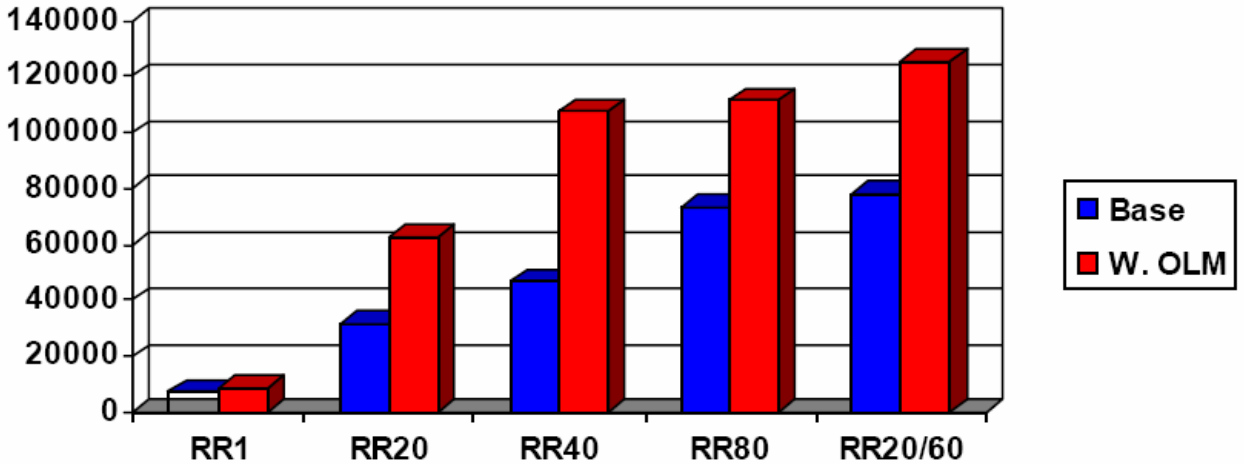
z10 (4 CP LPARs), z/OS V1R11, OSA-E3

- **Client and Server** have almost no application logic
- **RR1 with** one session
  - One byte in, one byte out
- **RR20 with** 20 sessions
  - 128 bytes in, 1024 bytes out
- **RR40 with** 40 sessions
  - 128 bytes in, 1024 bytes out
- **RR80 with** 80 sessions
  - 128 bytes in, 1024 bytes out
- **RR20/60 with** 80 sessions
  - Mix of 100/128 bytes in and 800/1024 out

End-to-end latency (response time) in Micro seconds



Transaction rate – transactions per second



Optimized Latency Mode (OLM): Performance

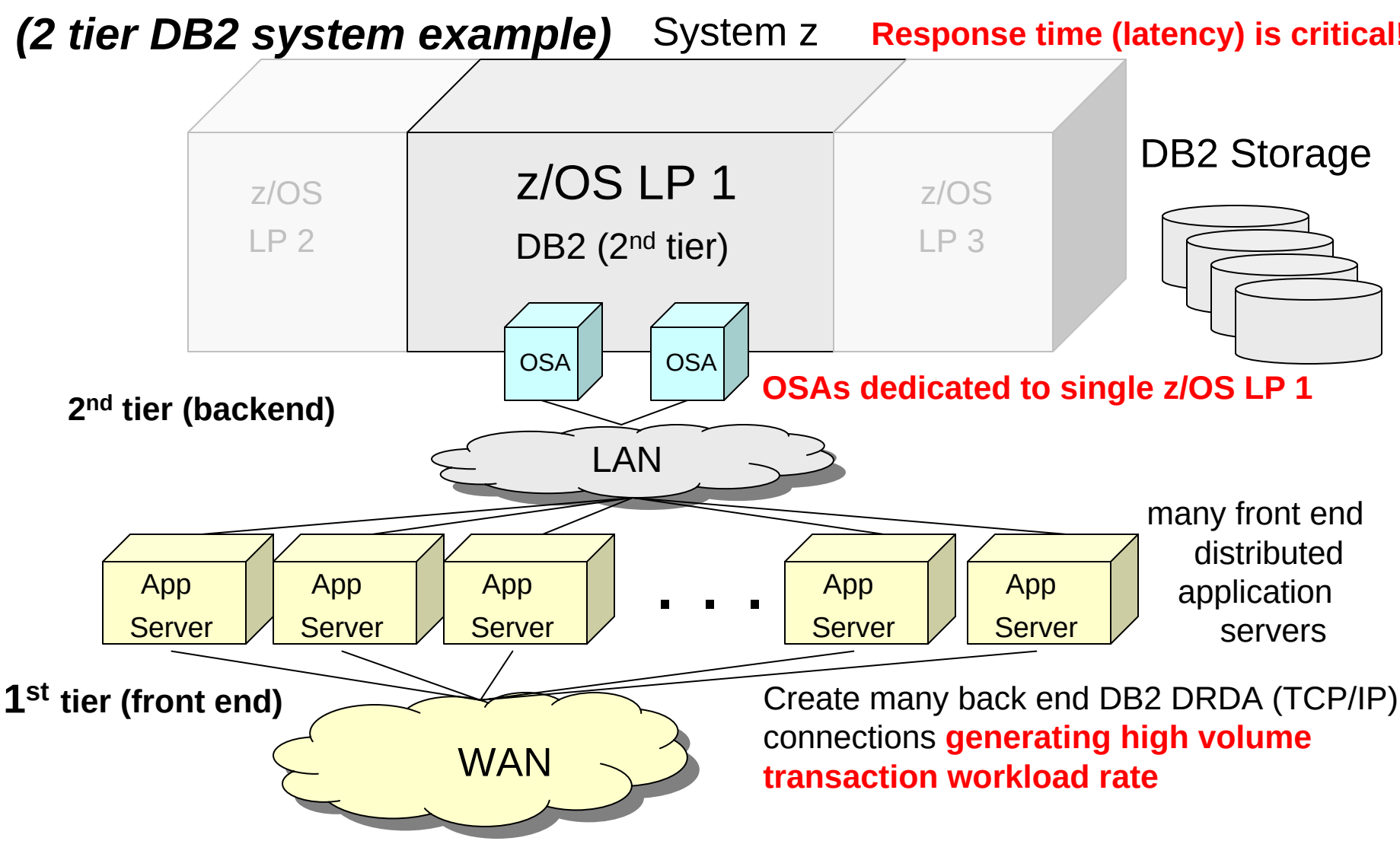
- Performance runs show
  - Single transaction - OLM reduced latency time by 17%
  - 20 simultaneous interactive sessions continually sending data – OLM:
    - Reduced average latency per transaction by 49%
    - Improved overall throughput by 95%!!!
- What happens when OLM is enabled with high volume streaming workloads?
  - z/OS Comm Server will detect and dynamically reduce usage of OLM
  - However, this traffic pattern can result in higher CPU

Optimized Latency Mode (OLM)

- Use OLM for workloads which have demanding QoS requirements for response time (transaction rate):
  - high volume interactive workloads (traffic is predominantly transaction oriented versus streaming)
- With high volume interactive workloads you should see significant latency and throughput improvements (improved transaction rate)
- More aggressive than INBPERF DYNAMIC algorithm, optimized for interactive workloads (can still handle streams but may be non-optimal)
- Only supported on OSA-Express3 with the INTERFACE statement
- Enabled via PTFs for z/OS V1R11
  - PK90205 (PTF UK49041) and OA29634 (UA49172).

OLM Sample Target Customer Environment

(2 tier DB2 system example)



Optimized Latency Mode (OLM): How to configure

```
INTERFACE NSQDIO411 DEFINE IPAQENET
  IPADDR 172.16.11.1/24
  PORTNAME NSQDIO1
  MTU 1492 VMAC OLM
  INBPERF DYNAMIC
  SOURCEVIPAINTERFACE LVIPA1
```

- New OLM parameter
  - IPAQENET/IPAQENET6
  - **Not** allowed on DEVICE/LINK
- Enables Optimized Latency Mode for this INTERFACE only
- Forces INBPERF to DYNAMIC
- Default NOOLM

- Use Netstat DEvlinks/-d to see current OLM configuration

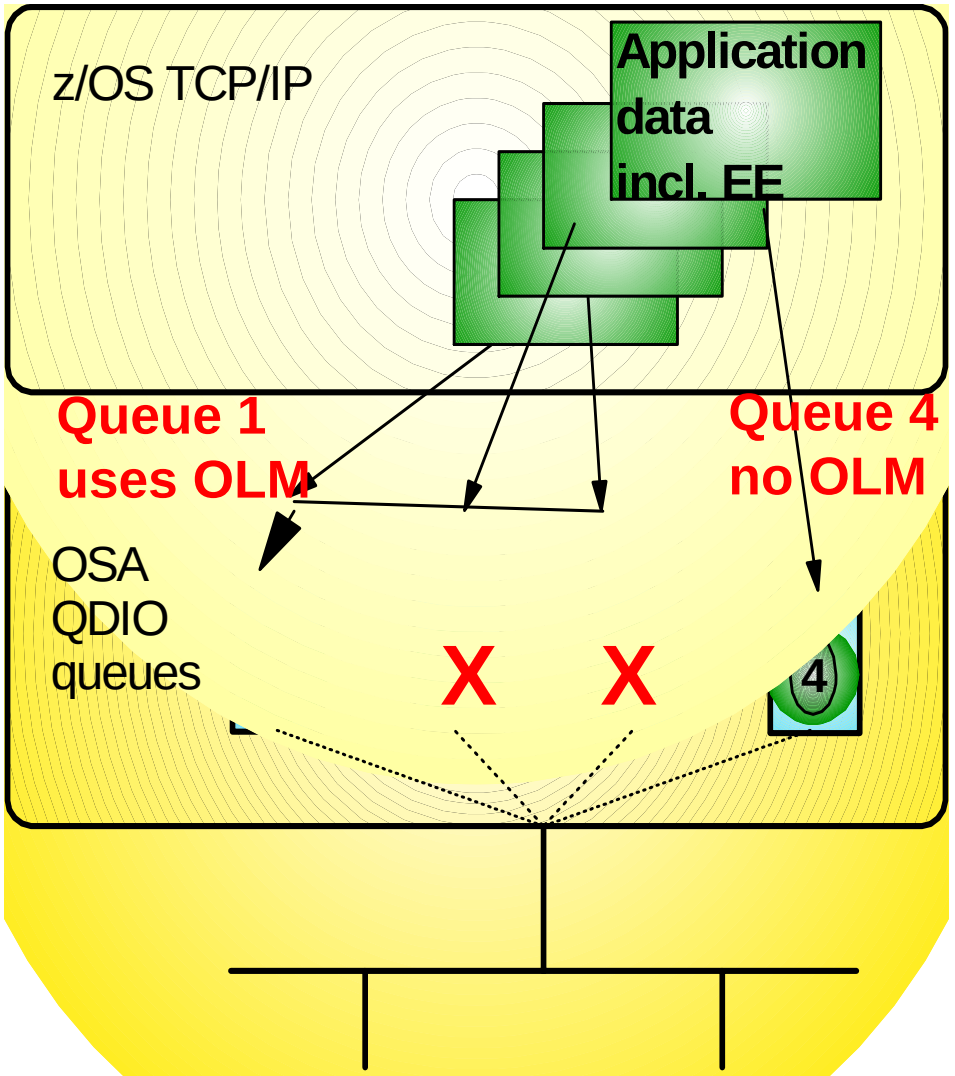
```
d tcpip,tcpcs,netstat,devlinks,intfname=lnsqdio1
JOB    6  EZD0101I NETSTAT CS V1R11 TCPCS
 INTFNAME: LNSQDIO1        INTFTYPE: IPAQENET   INTFSTATUS: READY
     .
     READSTORAGE: GLOBAL (4096K)      INBPERF: DYNAMIC
     .
     ISOLATE: NO                      OPTLATENCYMODE: YES
```

Optimized Latency Mode (OLM): OSA Sharing

- Concurrent interfaces to an OSA-Express port using OLM is limited.
  - If one or more interfaces operate OLM on a given port,
    - Only four total interfaces allowed to that single port
    - Only eight total interfaces allowed to that CHPID
  - All four interfaces can operate in OLM
  - An interface can be:
    - Another interface (e.g. IPv6) defined for this OSA-Express port
    - Another stack on the same LPAR using the OSA-Express port
    - Another LPAR using the OSA-Express port
    - Another VLAN defined for this OSA-Express port
    - Any stack activating the OSA-Express Network Traffic Analyzer (OSAENTA)
- QDIO Accelerator or HiperSockets Accelerator will not accelerate traffic to or from an OSA-Express operating in OLM

Optimized Latency Mode (OLM): Queues

- OLM only enabled for outbound traffic on OSA-E Write Priority Queue 1
- For OLM, z/OS V1R11 Communications Server collapses outbound traffic on queues 1-3 to queue 1
  - Queue 2 and 3 not used
- So traffic must be directed to queues 1-3 to use OLM
- Use GLOBALCONFIG WLMPRIORITYQ (default is ok) or SETSUBNETPRIOTOSMASK to put data on queues 1-3



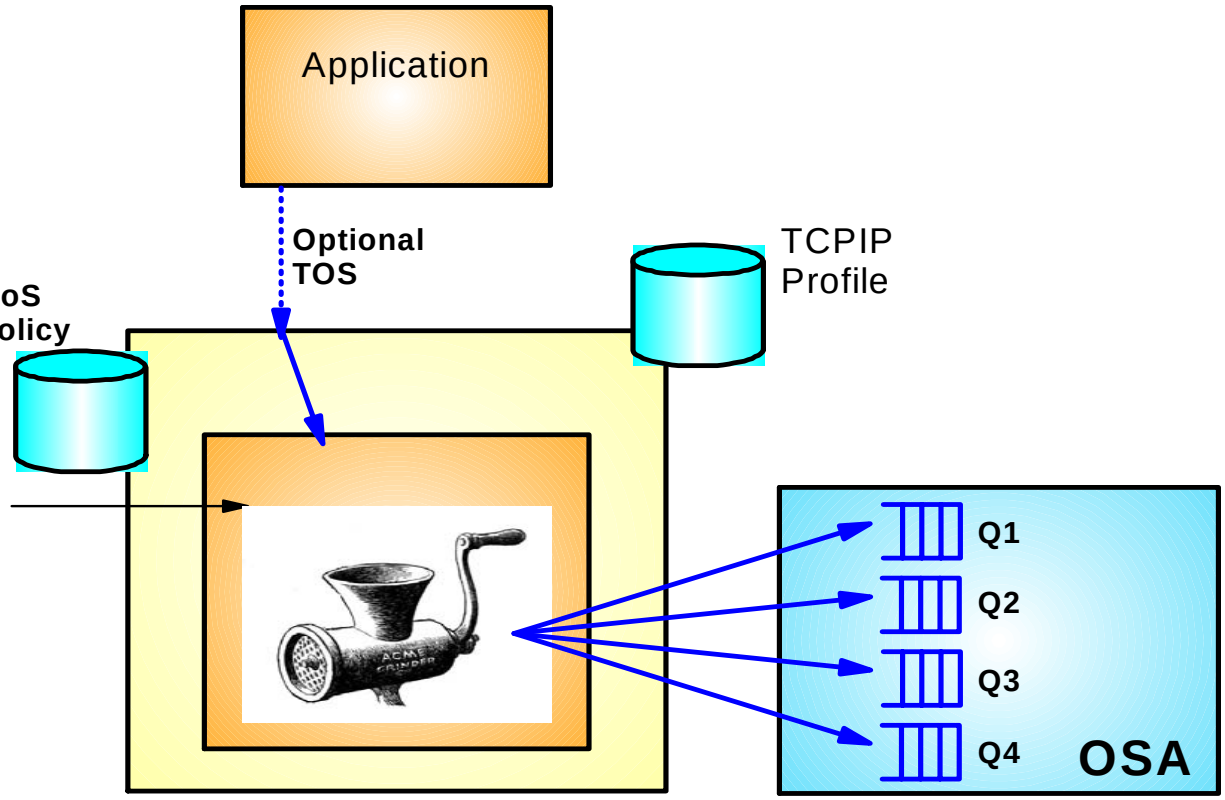
Improving OSA Performance with z/OS Communications Server

WLM Priority Queueing

OSA Express Outbound priority queuing

Prior to z/OS V1R11 you have the ability to control which outbound priority queue is used for your network traffic using TCP/IP configuration and QoS policies (PagenT)

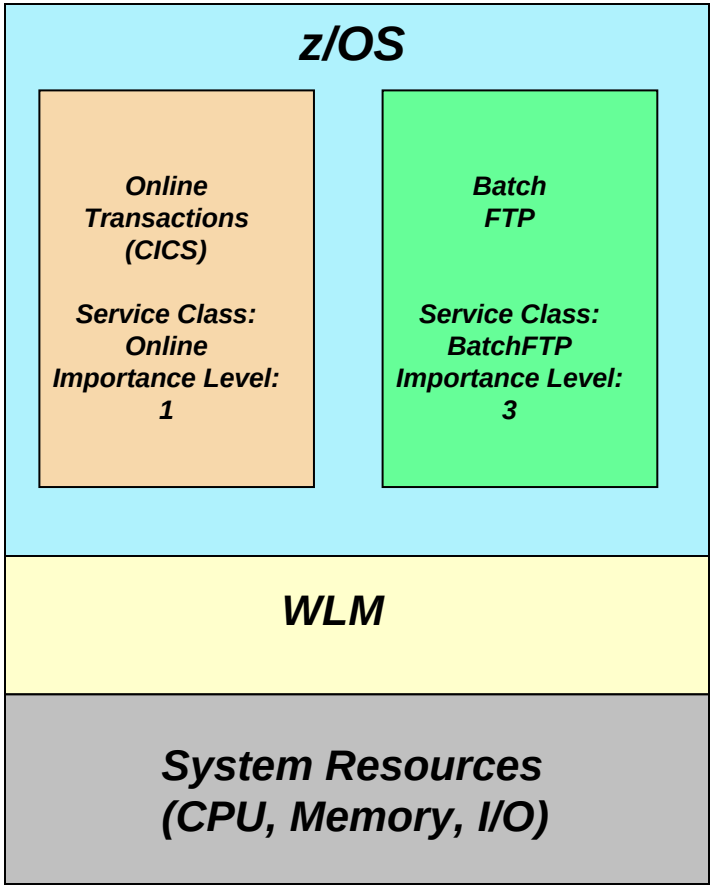
```
SetSubnetPrioTosMask
{
SubnetTosMask 11100000
PriorityTosMapping 1 11100000
PriorityTosMapping 1 11000000
PriorityTosMapping 1 10100000
PriorityTosMapping 1 10000000
PriorityTosMapping 2 01100000
PriorityTosMapping 2 01000000
PriorityTosMapping 3 00100000
PriorityTosMapping 4 00000000
}
```



- While this feature allows for very flexible means of prioritizing outbound network traffic it has not been widely exploited by users
  - *How can we simplify its exploitation?*

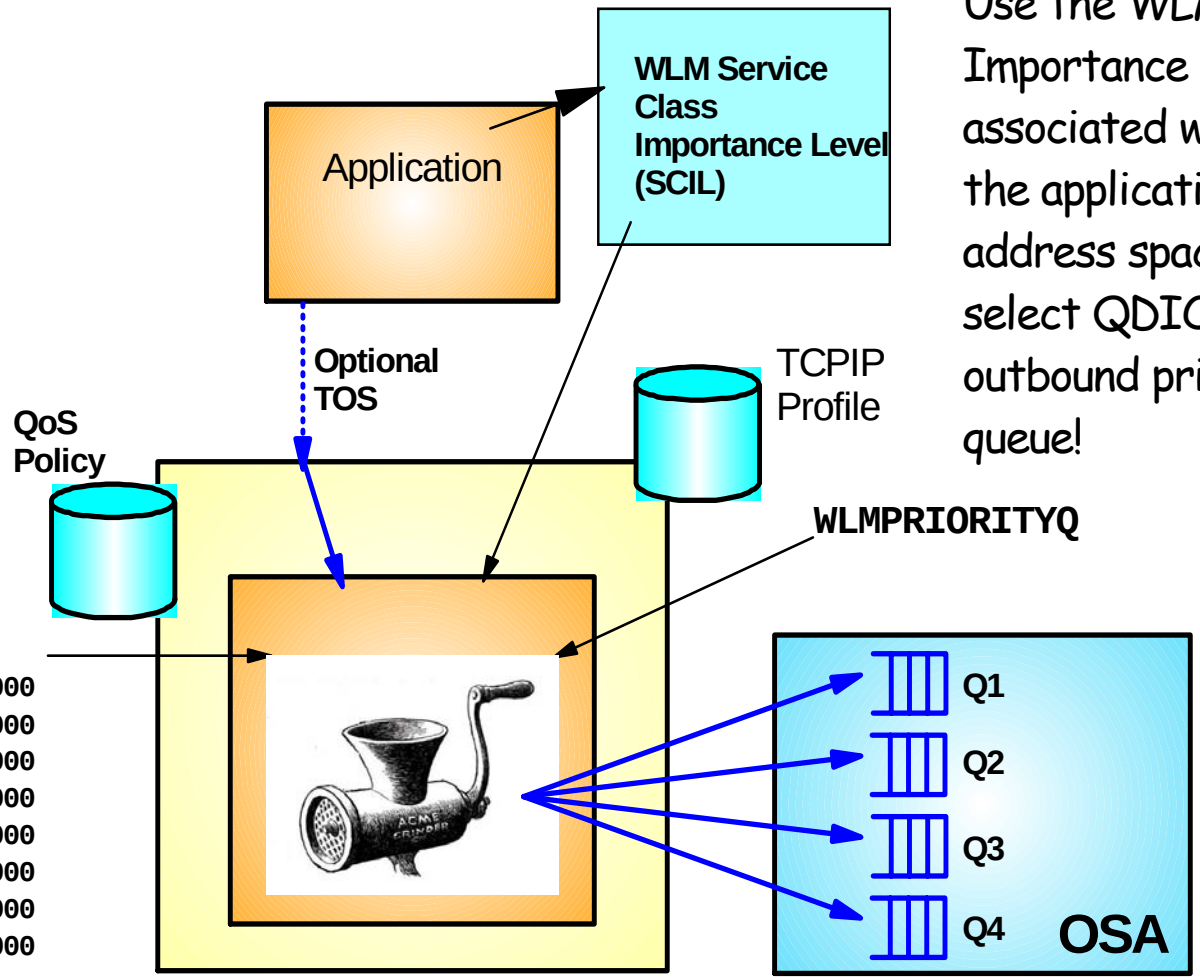
z/OS Workload Manager (WLM)
Managing workloads of different business priorities

- WLM policy allows users to specify the business goals and priorities for all their z/OS workloads
  - Sysplex-wide goals
  - WLM manages key system resources (memory, CPU) to help workloads achieve their specified goals
  - What happens when resources are over-committed?
    - WLM begins prioritizing access to system resources based on the specified Importance Level of each Service Class associated with the workloads currently executing
      - Emphasis is placed on meeting the goals for the more important workloads
  - Over time WLM resource priority management has been expanded to also include I/O prorities (DASD and Tape)
    - But what about Network I/O priority?



Extending WLM priorities to Outbound Network I/O (OSA Express)

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



Use the WLM Importance Level associated with the 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
}
```

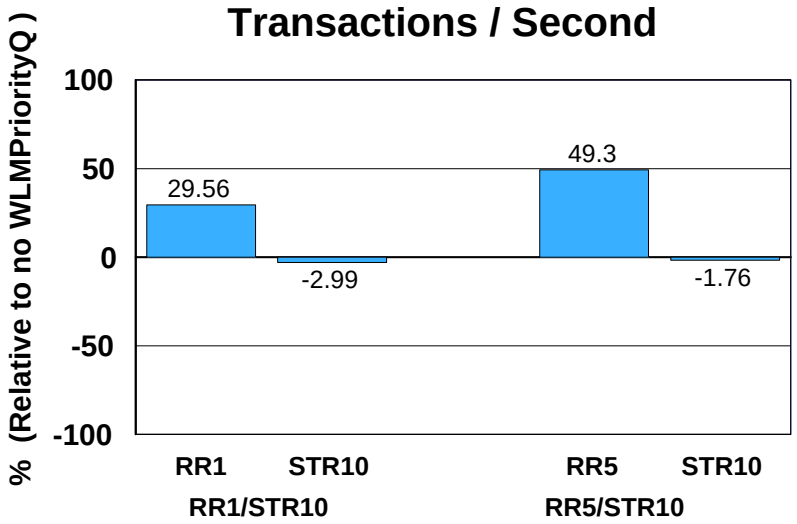
The default QDIO priority queue mapping

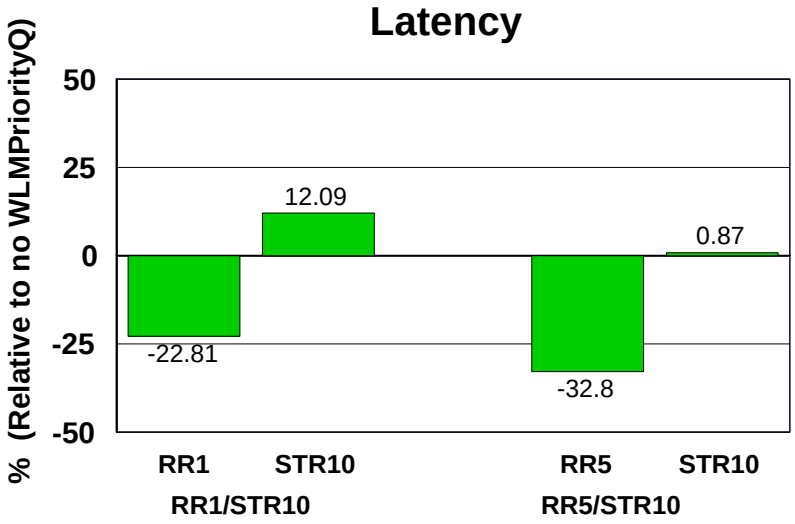
WLM Service classes	TCP/IP assigned	Default QDIO queue mapping
SYSTEM	n/a	Always queue 1
SYSSTC	0	Queue 1
User-defined with IL 1	1	Queue 2
User-defined with IL 2	2	Queue 3
User-defined with IL 3	3	Queue 3
User-defined with IL 4	4	Queue 4
User-defined with IL 5	5	Queue 4
User-defined with discretionary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

```
GLOBALCONFIG … WLMPRIORITYQ
   IOPRI1 0
   IOPRI2 1
   IOPRI3 2 3
   IOPRI4 4 5 6 FWD
```

FWD indicates forwarded (or routed) traffic, which by default will use QDIO priority queue 4

OSA Express (QDIO) WLM Outbound Priority Queuing





- 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 29.56 to 49.3% higher throughput for interactive workloads compared to V1R11 without WLM I/O Priority (Avg= 39.43% higher).
- z/OS V1R11 with WLM I/O Priority provides 22.81 to 32.8% lower latency compared to V1R11 without WLM I/O Priority (Avg= 27.80% lower).

Note: The performance measurements discussed in this presentation are preliminary z/OS V1R12 Communications Server numbers and were collected using a dedicated system environment. The results obtained in other configurations or operating system environments may vary.

Which QDIO priority queues are being used?

From Display tcpip,,n,devlinks:

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

From VTAMLST MACLIB:

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

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

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
```

Example of enabling WLMPRIORITYQ

VTAM TNSTATS before enabling WLMPRIORITYQ

```
IST1233I DEV      = 2E02       DIR      = WR/1
..
IST1236I BYTECNTO =          0 BYTECNT  =         72
IST1810I PKTIQDO  =          0 PKTIQD   =          0
IST1811I BYTIQDO  =          0 BYTIQD   =          0
IST924I -------------------------------------------
-
IST1233I DEV      = 2E02       DIR      = WR/2
..
IST1236I BYTECNTO =          0 BYTECNT  =          0
IST1810I PKTIQDO  =          0 PKTIQD   =          0
IST1811I BYTIQDO  =          0 BYTIQD   =          0
IST924I -------------------------------------------
-
IST1233I DEV      = 2E02       DIR      = WR/3
..
IST1236I BYTECNTO =          0 BYTECNT  =          0
IST1810I PKTIQDO  =          0 PKTIQD   =          0
IST1811I BYTIQDO  =          0 BYTIQD   =          0
IST924I -------------------------------------------
-
IST1233I DEV      = 2E02       DIR      = WR/4
..
IST1236I BYTECNTO =          0 BYTECNT  =      34738
IST1810I PKTIQDO  =          0 PKTIQD   =          0
IST1811I BYTIQDO  =          0 BYTIQD   =          0
```

VTAM TNSTATS after enabling WLMPRIORITYQ with defaults

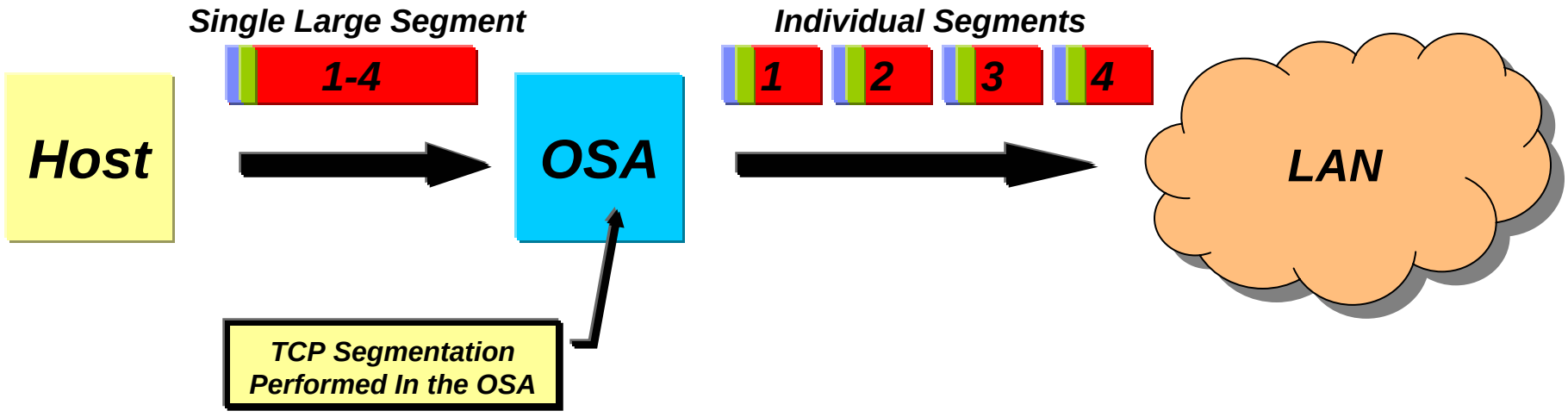
```
IST1233I DEV      = 2E02       DIR      = WR/1
..
IST1236I BYTECNTO =          0 BYTECNT  =       1552
IST1810I PKTIQDO  =          0 PKTIQD   =          0
IST1811I BYTIQDO  =          0 BYTIQD   =          0
IST924I -------------------------------------------
-
IST1233I DEV      = 2E02       DIR      = WR/2
..
IST1236I BYTECNTO =          0 BYTECNT  =      55421
IST1810I PKTIQDO  =          0 PKTIQD   =          0
IST1811I BYTIQDO  =          0 BYTIQD   =          0
IST924I -------------------------------------------
-
IST1233I DEV      = 2E02       DIR      = WR/3
..
IST1236I BYTECNTO =          0 BYTECNT  =          0
IST1810I PKTIQDO  =          0 PKTIQD   =          0
IST1811I BYTIQDO  =          0 BYTIQD   =          0
IST924I -------------------------------------------
-
IST1233I DEV      = 2E02       DIR      = WR/4
..
IST1236I BYTECNTO =          0 BYTECNT  =      90411
IST1810I PKTIQDO  =          0 PKTIQD   =          0
IST1811I BYTIQDO  =          0 BYTIQD   =          0
```

Improving OSA Performance with z/OS Communications Server

TCP Segmentation Offload

TCP Segmentation Offload

- Segmentation consumes (high cost) host CPU cycles in the TCP stack
- V1R7 (PTFed to V1R6) offered new OSA-Express (QDIO mode) feature 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



TCP Segmentation Offload: Performance

- OSAE-2, 1 GbE (versus no segmentation offload):

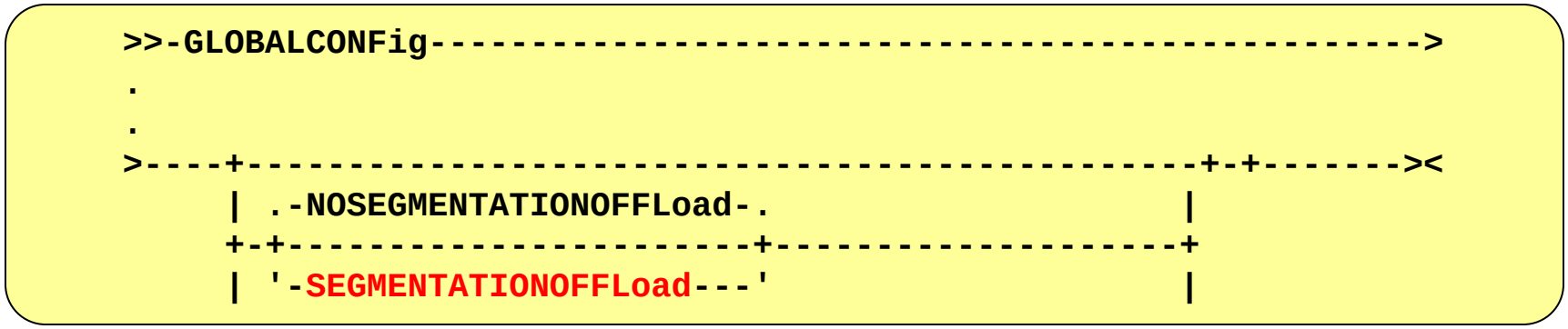
Workload	Trans/Sec Delta %	CPU/Tran Delta %
RR 60	+ 1.3 %	- 0.7 %
CRR 9	+ 2 %	- 0.1 %
STR (1/20M):		
64K(send)/32K(recv)	Equal	- 28.9 %
180K(send)/64K(recv)	Equal	- 36.3 %
256K(send)/64K(recv)	Equal	- 39.2 %

- OSAE-2, 10 GbE (versus no segmentation offload):

Workload	Trans/Sec Delta %	CPU/Tran Delta %
RR 60	+ 1.7 %	- 2 %
CRR 60	+ 5.2 %	- 1 %
STR (1/20M):		
64K(send)/32K(recv)	+ 1.1 %	- 33.4 %
180K(send)/64K(recv)	+ 1.5 %	- 41.5 %
256K(send)/64K(recv)	+ 0.4 %	- 44.9 %

TCP Segmentation Offload: Configuration

- Enabled with GLOBALCONFIG SEGMENTATIONOFFLOAD



- Disabled by default
- TCP/IP stack will still do segmentation for
  - Packets going LPAR to LPAR
  - IPSec encapsulated packets
  - When multipath is in effect (unless all interfaces in the multipath group support segmentation offload)

TCP Segmentation Offload: Packet Trace Enhancements

- **SESSION Report**
  CTRACE COMP(SYSTCPDA) SUB((TCPCS1)) SHORT OPTIONS((SESSION))

```
TcpHdr    IO F         Seq         Ack RcvWnd  Data Delta Time       TimeStamp
      S    I    458316454           0  32768     0   0.000000 00:54:40.567581
    A  S    O    456248587   458316455  32768     0   0.050860 00:54:40.618441
    A       I u  458316455   456248588  32768     0   0.077633 00:54:40.696074
 o AP       O .  456248588   458316455  32768  5752   7.023527 00:54:47.719601
    A       I a  458316455   456254340  27016     0   0.069695 00:54:47.789296
 o AP       O .  456254340   458316455  32768  7190   0.004164 00:54:47.793460
    A       I a  458316455   456261530  19826     0   0.063442 00:54:47.856902

Data Segment Stats:                  Inbound,         Outbound
 Number of data segments:                  0,               39
 Maximum segment size:                  1460,             1460
 Largest segment size:                     0,            16384
 Average segment size:                     0,            12820
 Smallest segment size:                    0,             3808
 Segments/window:                        0.0,              1.0
 Average bytes/window:                     0,            12820
 Most bytes/window:                        0,            16384
 Offload Sends:                                             39            (  100%)
 Offload Segments:                                         365
 Offload Bytes:                                         500000            (  100%)
```

TCP Segmentation Offload: Packet Trace Enhancements

- **SHORT Report**
  CTRACE COMP(SYSTCPDA) SUB((TCPCS1)) SHORT

```
00000004 00:54:47.719601 Packet Trace
 To Interface      : NSQDIO1L          Device: QDIO Ethernet    Full=5804
  Tod Clock        : 2004/11/15 00:54:47.719600                 Intfx: 28
  Sequence #       : 0                 Flags: Pkt Out Offl
 IpHeader: Version : 4                 Header Length: 20
  Tos              : 00                QOS: Routine Normal Service
  Offload Length   : 5804              ID Numbers: 0053-0056
  Fragment         :                   Offset: 0
  TTL              : 64                Protocol: TCP            CheckSum: 0000 9E61
  Source           : 10.1.1.1
  Destination      : 10.1.4.5
 TCP
  Source Port      : 8084  ()          Destination Port: 1027  ()
  Sequence Number  : 456248588         Ack Number: 458316455
  Header Length    : 32                Flags: Ack Psh
  Window Size      : 32768             CheckSum: 190E 0000 Urgent Data Pointer: 0000
  Offload Segments : 4                 Length: 1438
   Option          : NOP
   Option          : NOP
   Option          : Timestamp         Len: 10 Value: 7C7141DC Echo: 7C714194
```

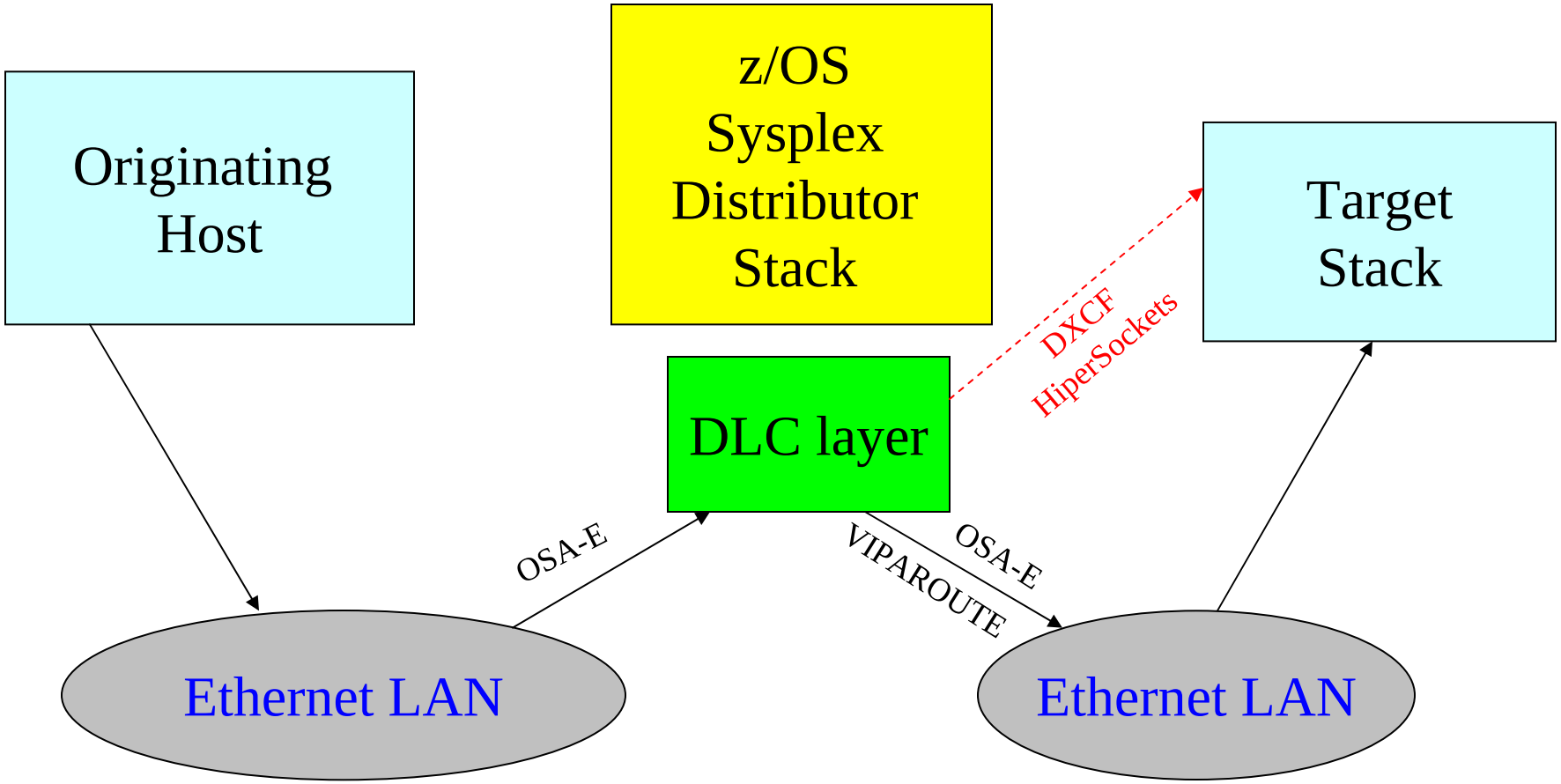
Improving OSA Performance with z/OS Communications Server

QDIO Accelerator

QDIO Accelerator

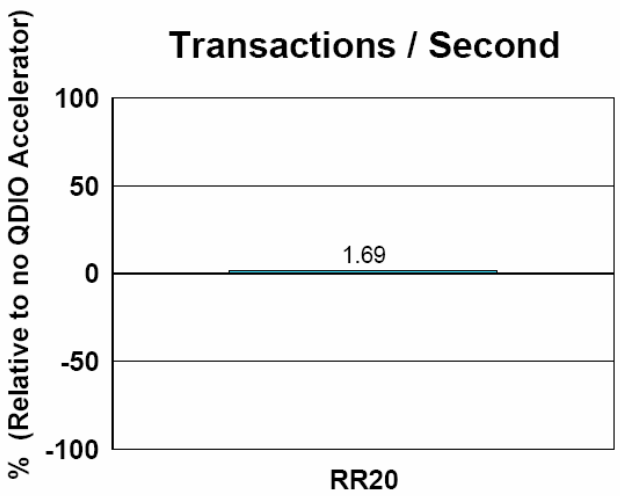
- Accelerates Sysplex Distributor (SD) traffic at the DLC layer
  - Inbound packets over HiperSockets or OSA-E QDIO
  - Outbound when SD gets to the target stack using either:
    - Dynamic XCF connectivity over HiperSockets
    - VIPAROUTE over OSA-E QDIO
- SD registers DVIPAs with the DLC
- When packets arrive in the DLC for a registered DVIPA, the DLC checks with SD to see if it can immediately forward the packet for the connection
- Packets are then forwarded by the DLC layer bypassing the forwarding stack
- Reduces CPU usage and improves performance for such workloads
- IPv4 only, no fragmentation, no IPSECURITY, no OLM

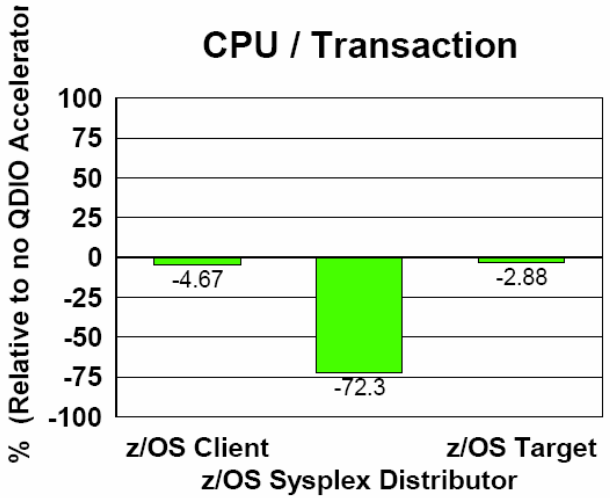
Sysplex Distributor connection routing accelerator



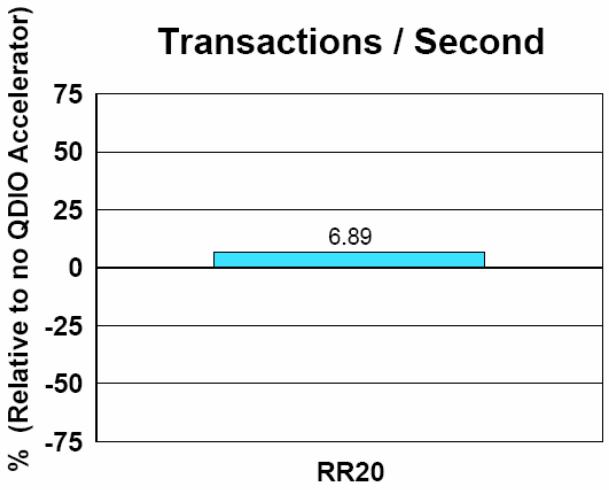
QDIO Accelerator: Performance

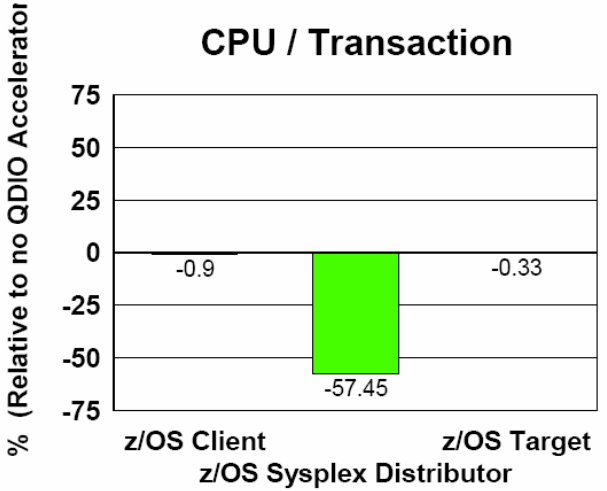
Sysplex Distributor QDIO Accelerator (RR HiperSockets)





Sysplex Distributor QDIO Accelerator (RR GbE)

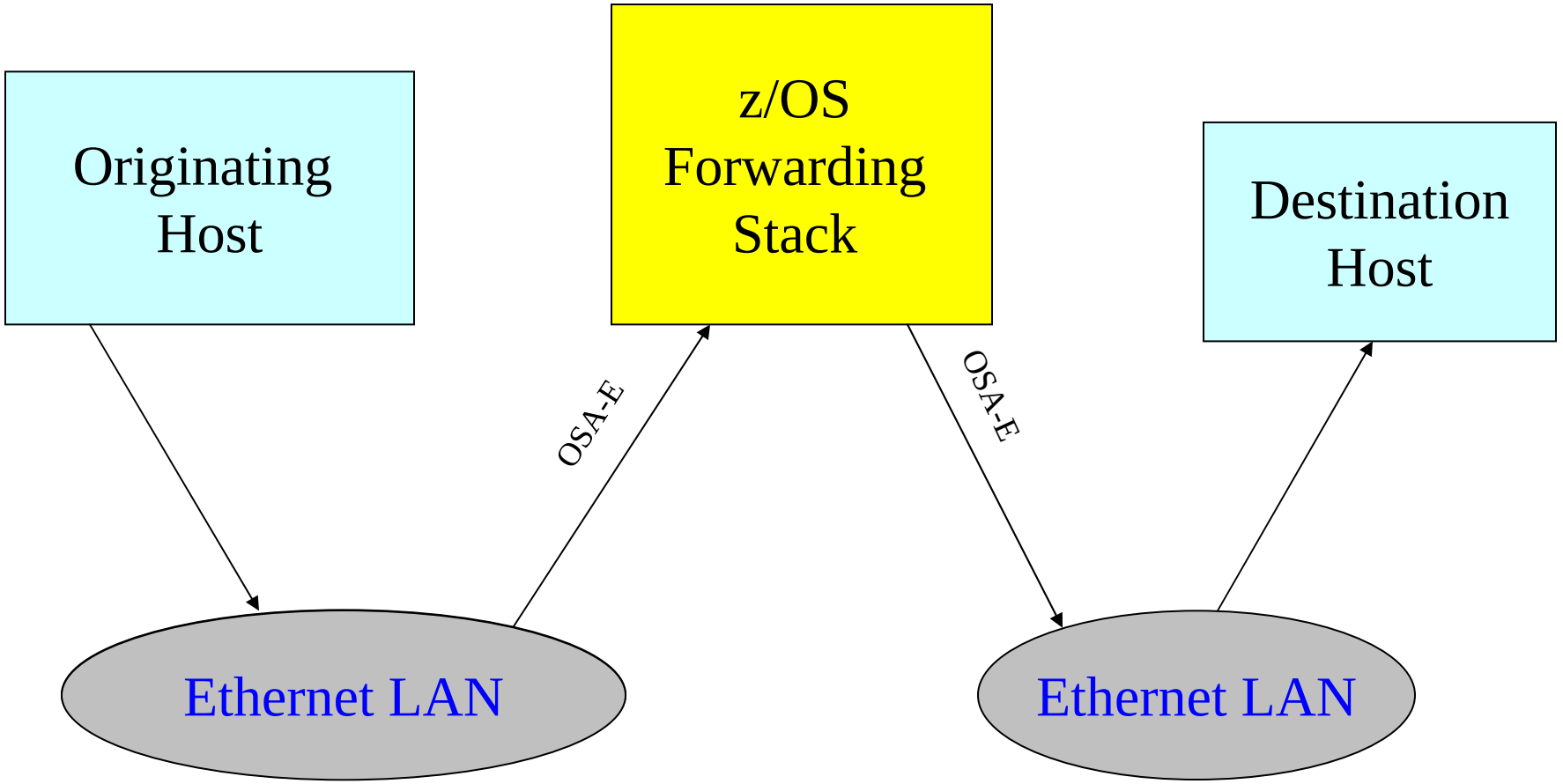




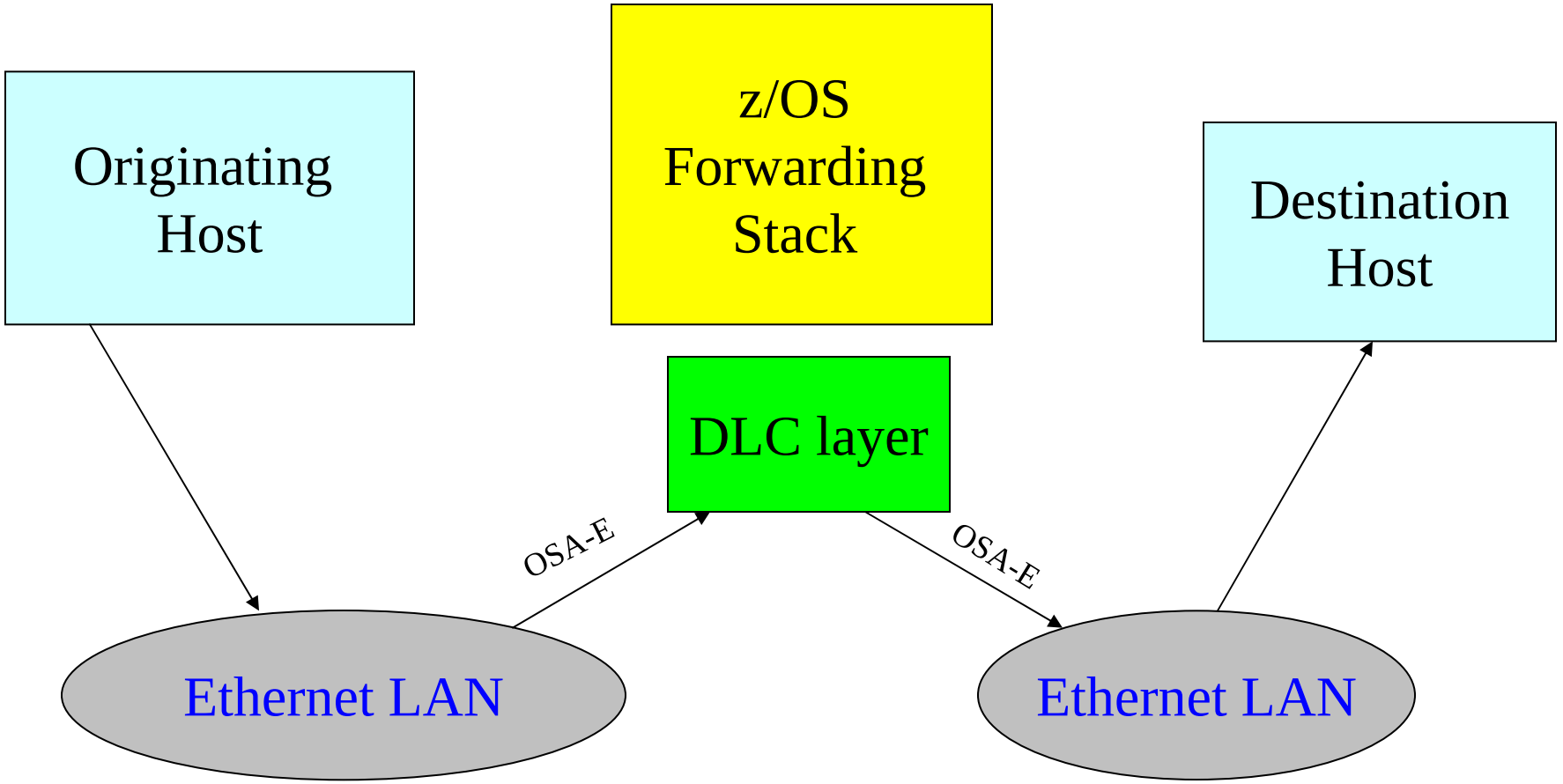
QDIO Accelerator

- HiperSockets Accelerator (V1R2) provides fast path IP forwarding for these DLC combinations for non-SD traffic:
  - Inbound OSA-E QDIO → Outbound HiperSockets
  - Inbound HiperSockets → Outbound OSA-E QDIO
- QDIO Accelerator (V1R11) functionally includes HiperSockets accelerator and also provides fast path IP forwarding for these DLC combinations for non-SD traffic:
  - Inbound OSA-E QDIO → Outbound OSA-E QDIO
  - Inbound HiperSockets → Outbound HiperSockets
- Once an initial packet is forwarded by stack, destination IP addrs and outbound interfaces are registered in the DLC for future fast path forwarding

Background: IP Forwarding



QDIO Accelerator



Note: Still requires IP Forwarding be enabled

QDIO Accelerator

Function	IQDIOROUTING	QDIOACCELERATOR
OSA-E → HiperSockets	Yes	Yes
HiperSockets → OSA-E	Yes	Yes
OSA-E → OSA-E	No	Yes
HiperSockets → HiperSockets	No	Yes
Sysplex Distributor	No	Yes

QDIO Accelerator

- QDIOACCELERATOR enables the acceleration function for all OSA-Express QDIO and HiperSockets interfaces
- Mutually exclusive with IQDIOROUTING
- Can enable or disable dynamically, but only if acceleration is configured in the initial profile

```
IPCONFIG

|  _NOQDIOACCELerator_____________________________          |
|_|_____________________________________________|_________|
| |                     _QDIOPriority 1_________|         |
| |_QDIOACCELerator__|_____________________________|         |
|                    |_QDIOPriority priority_|             |

…
```

QDIO Accelerator

- Netstat CONFIG/-f example
  - Indicates whether QDIO Accelerator is enabled

```
NETSTAT CONFIG

MVS TCP/IP NETSTAT CS V1R11      TCPIP NAME: TCPCS       09:51:02
…
QDIOAccel:  Yes       QDIOAccelPriority: 1
IQDIORoute: n/a
```

QDIO Accelerator

- Netstat VCRT/-V example
  - Indicates which Sysplex Distributor connections are eligible for QDIO Acceleration

```
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
  PolicyRule:    *NONE*
  PolicyAction:  *NONE*
  Intf:   OSAQDIOLINK
    VipaRoute: Yes      Gw: 199.100.1.1
  Accelerator: Yes
```

QDIO Accelerator

- Netstat ROUTe/-r (QDIOACCEL parameter) example
  - Displays QDIO Accelerator routes (non-SD)

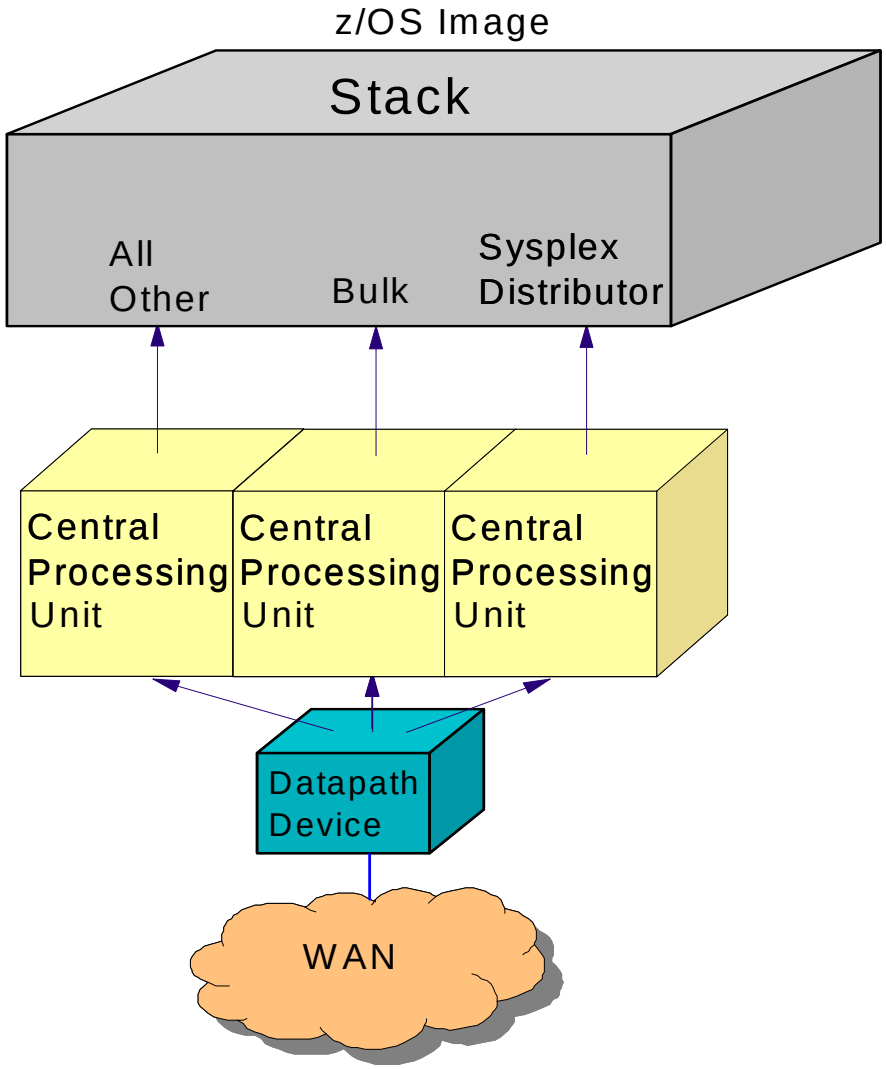
```
NETSTAT ROUTE QDIOACCEL

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
```

Improving OSA Performance with z/OS Communications Server

QDIO Inbound Workload Queueing (V1R12)

QDIO Inbound Workload Queueing

- OSA separates the packets and routes them over 3 different read queues on the same interface
- Each queue can be serviced concurrently by a separate processor
- Stack receives pre-sorted packets



QDIO Inbound Workload Queueing

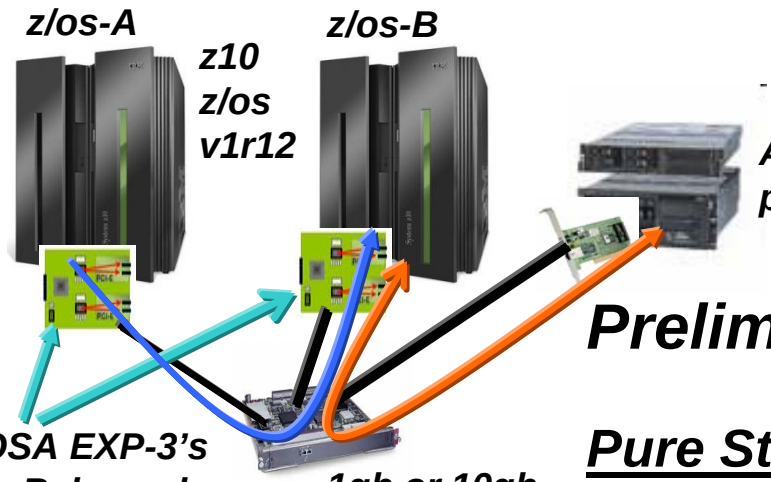
- Traffic separation by using multiple read queues
  - Stack “registers” with OSA which traffic goes to which queue
  - OSA-Express3 hardware data router puts traffic on the correct queue
- Each input queue can be serviced by a separate process
  - Primary input queue for general traffic
  - One or more ancillary input queues (AIQs) for specific traffic types
- Supported traffic types (IPv4 and IPv6)
  - Streaming bulk data (FTP, TSM, NFS, TDMF)
  - Sysplex Distributor

QDIO Inbound Workload Queueing - benefits

- Bulk data traffic queue
  - Serviced from a single process - eliminates the out of order delivery issue
- Sysplex distributor traffic queue
  - SD traffic efficiently accelerated or presented to target application
- All other traffic processed concurrently with bulk data and SD traffic
- Dynamic LAN idle timer updated per queue

QDIO Inbound Workload Queueing – Early Performance Data

Performance Test Configuration:



Your mileage may vary. Performance notes: For z/OS outbound streaming to another platform, 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 when transmission is over "lossy" links; for mixed workloads, performance boost for interactive traffic is possible only if the streaming workload has not consumed all of the bandwidth.

Preliminary performance results:

Pure Streaming Workloads:

(IWQ throughput boost relative to INBPERF DYNAMIC)

z/OS->z/OS: +30% (20 to 40%)

z/OS->AIX: +40% (39 to 41%)

Mixed Interactive+Streaming Workload:

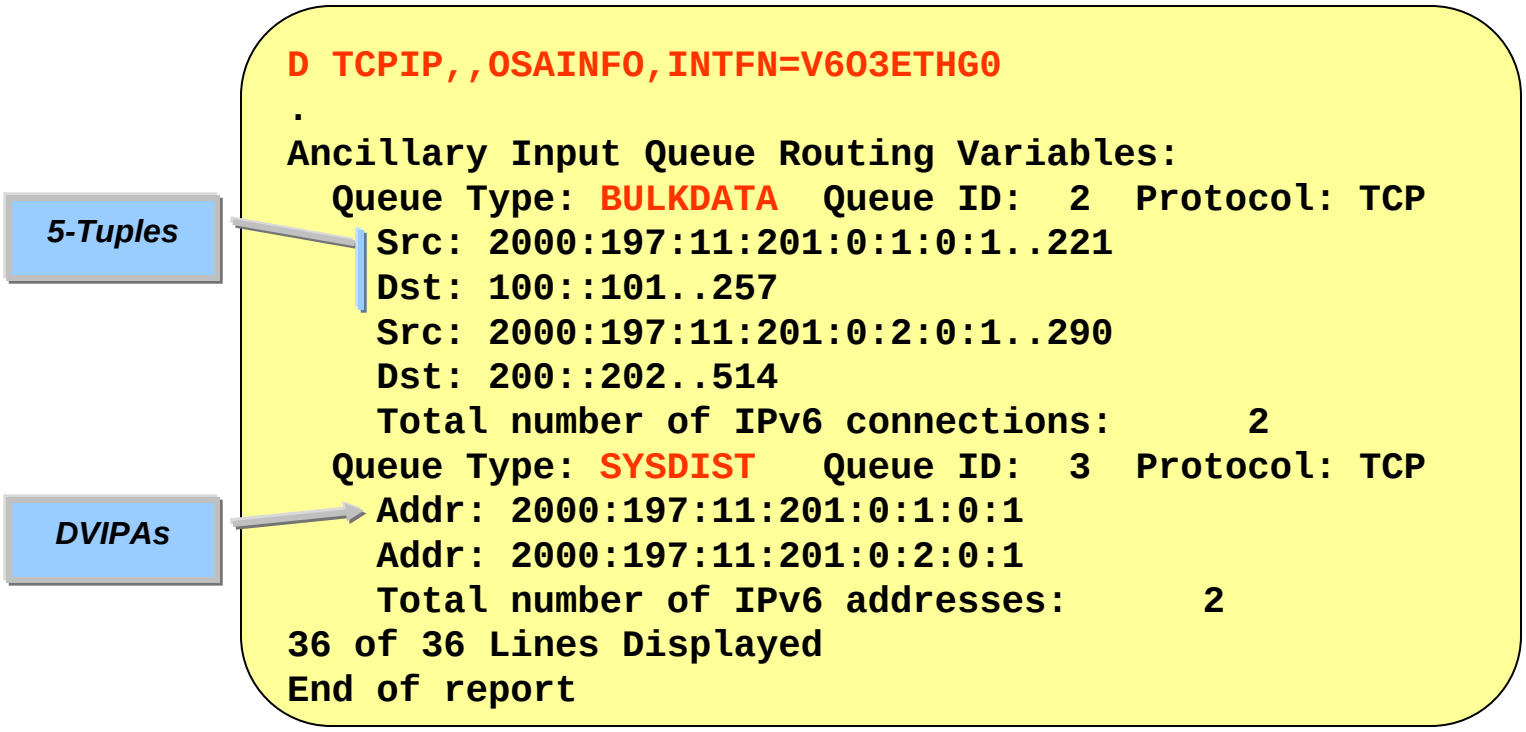
(workload is: interactive request/response workload running between z/OS-B and AIX, while z/OS-B is also receiving streaming traffic from z/OS-A over the same 1Gb OSA-3 handling the R/R traffic. We compare z/OS-B's OSA-3 running in IWQ mode, vs Dynamic Mode. IWQ throughput and response time improvements are relative to INBPERF DYNAMIC.)

z/OS<->AIX R/R Throughput improved 55% (Response Time improved 36%).

Streaming Throughput also improved in this test: +5%

QDIO Inbound Workload Queueing

- Display OSAINFO command (V1R12) shows you what's registered in OSA



- BULKDATA queue registers 5-tuples with OSA (streaming connections)
- SYSDIST queue registers DVIPAs with OSA

QDIO Inbound Workload Queueing

- INBPERF DYNAMIC WORKLOADQ enables QDIO inbound workload queueing

```
>>-INTERFace--intf_name-------------------------------------->
.
.-INBPERF BALANCED--------------------.
 >--+-----------------------------------+-->
    |                    .-NOWORKLOADQ-.   |
    '-INBPERF-+-DYNAMIC-+-------------+-+-'
              |         '-WORKLOADQ---' |
              +-MINCPU------------------+
              '-MINLATENCY--------------'
```

  - INTERFACE statements only - no support for DEVICE/LINK definitions
  - QDIO inbound Workload Queueing requires VMAC

QDIO Inbound Workload Queueing

- Requires OSA-Express3 in QDIO mode running on an IBM System z10
- Not supported when z/OS is running as a z/VM guest with simulated devices (VSWITCH or guest LAN)
- Connections where multiple QDIO interfaces are servicing the transfer (i.e. multipath perpacket) are not put on the Bulkdata queue
- Each ancillary queue will consume:
  - Approximately nine additional pages of ECSA
  - An additional but tunable amount of fixed CSM data space as specified by the READSTORAGE parameter

QDIO Inbound Workload Queueing: Netstat DEvlinks/-d

- Display TCPIP,,Netstat,DEvlinks to see whether QDIO inbound workload queueing is enabled for a QDIO interface

```
D TCPIP,TCPCS1,NETSTAT,DEVLINKS,INTFNAME=QDIO4101L
EZD0101I NETSTAT CS V1R12 TCPCS1
INTFNAME: QDIO4101L          INTFTYPE: IPAQENET   INTFSTATUS: READY
    PORTNAME: QDIO4101  DATAPATH: 0E2A     DATAPATHSTATUS: READY
    CHPIDTYPE: OSD
    SPEED: 0000001000
...
    READSTORAGE: GLOBAL (4096K)
    INBPERF: DYNAMIC
      WORKLOADQUEUEING: YES
    CHECKSUMOFFLOAD: YES
    SECCLASS: 255                    MONSYSPLEX: NO
    ISOLATE: NO                      OPTLATENCYMODE: NO
...
1 OF 1 RECORDS DISPLAYED
END OF THE REPORT
```

QDIO Inbound Workload Queueing: Display TRLE

- Display NET,TRL,TRLE=trlename to see whether QDIO inbound workload queueing is in use for a QDIO interface

```
D NET,TRL,TRLE=QDIO101
IST097I DISPLAY ACCEPTED
...
IST2263I PORTNAME = QDIO4101   PORTNUM =   0   OSA CODE LEVEL = ABCD
...
IST1221I DATA  DEV = 0E2A STATUS = ACTIVE     STATE = N/A
IST1724I I/O TRACE = OFF  TRACE LENGTH = *NA*
IST1717I ULPID = TCPCS1
IST2310I ACCELERATED ROUTING DISABLED
IST2331I QUEUE   QUEUE     READ
IST2332I ID      TYPE      STORAGE
IST2205I ------  --------  ---------------
IST2333I RD/1    PRIMARY   4.0M(64 SBALS)
IST2333I RD/2    BULKDATA  4.0M(64 SBALS)
IST2333I RD/3    SYSDIST   4.0M(64 SBALS)
...
IST924I -------------------------------------------------------------
IST314I END
```

QDIO Inbound Workload Queueing: Netstat ALL/-A

- Display TCPIP,,Netstat,ALL to see whether QDIO inbound workload queueing is in use for BULKDATA.

```
D TCPIP,TCPCS1,NETSTAT,ALL,CLIENT=USER1
EZD0101I NETSTAT CS V1R12 TCPCS1
CLIENT NAME: USER1                    CLIENT ID: 00000046
  LOCAL SOCKET: ::FFFF:172.16.1.1..20
  FOREIGN SOCKET: ::FFFF:172.16.1.5..1030
    BYTESIN:            0000000000023316386
    BYTESOUT:           0000000000000000000
    SEGMENTSIN:         0000000000000016246
    SEGMENTSOUT:        0000000000000000922
    LAST TOUCHED:       21:38:53        STATE:              ESTABLSH
...
Ancillary Input Queue: Yes
    BulkDataIntfName: QDIO4101L
...
    APPLICATION DATA:   EZAFTP0S D USER1     C      PSSS
----
1 OF 1 RECORDS DISPLAYED
END OF THE REPORT
```

QDIO Inbound Workload Queueing: Netstat STATS/-S

- Display TCPIP,,Netstat,STATS to see the total number of TCP segments received on BULKDATA queues

```
D TCPIP,TCPCS1,NETSTAT,STATS,PROTOCOL=TCP
EZD0101I NETSTAT CS V1R12 TCPCS1
TCP STATISTICS
  CURRENT ESTABLISHED CONNECTIONS     = 6
  ACTIVE CONNECTIONS OPENED           = 1
  PASSIVE CONNECTIONS OPENED          = 5
  CONNECTIONS CLOSED                  = 5
  ESTABLISHED CONNECTIONS DROPPED     = 0
  CONNECTION ATTEMPTS DROPPED         = 0
  CONNECTION ATTEMPTS DISCARDED       = 0
  TIMEWAIT CONNECTIONS REUSED         = 0
  SEGMENTS RECEIVED                   = 38611
...
  SEGMENTS RECEIVED ON OSA BULK QUEUES= 2169
  SEGMENTS SENT                       = 2254
...
END OF THE REPORT
```

QDIO Inbound Workload Queueing: VTAM Tuning Statistics

- VTAM tuning statistics indicate whether inbound traffic is using QDIO Inbound Workload Queueing

```
IST1230I TIME     = 16400874   DATE     = 10013      ID  = QDIO101
...
IST1233I DEV      = 0E2A         DIR      = RD/1 (PRIMARY)
IST1719I PCIREALO =          0 PCIREAL  =       7687
...
IST1754I NOREADSO =          0 NOREADS  =          0
IST1721I SBALCNTO =          0 SBALCNT  =         50
...
IST924I -------------------------------------------------------------
IST1233I DEV      = 0E2A         DIR      = RD/2 (BULKDATA)
IST1754I NOREADSO =          0 NOREADS  =          0
IST1721I SBALCNTO =          0 SBALCNT  =       7629
...
IST924I -------------------------------------------------------------
IST1233I DEV      = 0E2A         DIR      = RD/3 (SYSDIST)
IST1754I NOREADSO =          0 NOREADS  =          0
IST1721I SBALCNTO =          0 SBALCNT  =          8
```

QDIO Inbound Workload Queueing Diagnosis: IP traces

- Input queue ID (QID) and QID flag is included in:
  - Packet trace records
  - OSA-Express Network Traffic Analyzer (OSAENTA) trace records

```
8 MVS161   PACKET   00000004 15:39:52.034517 Packet Trace
From Interface    : QDIO4101L        Device: QDIO Ethernet    Full=60
 Tod Clock        : 2010/01/22 15:39:52.034516                Intfx: 35
 Segment #        : 0                Flags:  In QID
 Source           : 172.16.1.5
 Destination      : 10.91.1.1
 Source Port      : 1026             Dest Port: 4006  Asid: 003A TCB: 00000000
 QID              : 3
IpHeader: Version : 4                Header Length: 20
 Tos              : 00               QOS: Routine Normal Service
 Packet Length    : 60               ID Number: 001D
 Fragment         :                  Offset: 0
 TTL              : 64               Protocol: TCP            CheckSum: C22E FFFF
 Source           : 172.16.1.5
 Destination      : 10.91.1.1
```

Improving OSA Performance with z/OS Communications Server

z/OS Communications Server Performance Summaries

z/OS Communications Server Performance Summaries

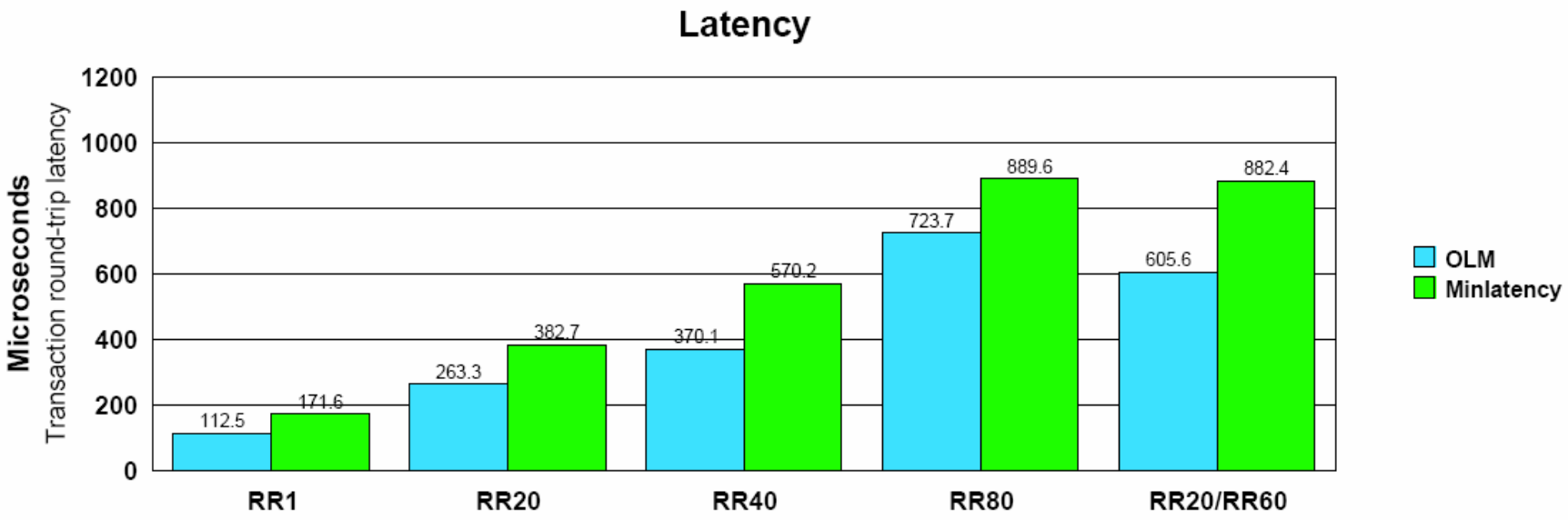
- Performance of each z/OS Communications Server release is studied by an internal performance team
- Summaries are created and published on line
  - http://www-01.ibm.com/support/docview.wss?rs=852&uid=swg27005524
- Ex: The z/OS V1R11 Communications Server Performance Summary includes:
  - Release to release performance comparisons (z/OS V1R11 CS versus z/OS V1R10 CS)
  - Performance of z/OS V1R11 Communications Server line items
  - Capacity planning performance for:
    - TN3270 (Clear Text, AT-TLS, and IPSec with and without zIIP processors)
    - FTP (Clear Text, AT-TLS, and IPSec with and without zIIP processors)
    - CICS Sockets

z/OS CS V1R11 vs V1R10 Performance Summary by Workload

CS Workload	V1R11 Throughput relative to V1R10	V1R11 CPU/Transaction relative to V1R10
AWM Primitives (1 GbE) RR60 (100/800) CRR9 (64/8K) STR10 Client (1/20M) STR10 Server (1/20M)	 + 1.37 % - 0.24 % + 1.04 % + 1.04 %	 - 3.94 % - 3.28 % - 9.92 % - 20.09 %
FTP Server (1 GbE)	- 1.62 %	- 3.80 %
TN3270 Server (1 GbE)	Equal (with think time)	- 0.89 %
CICS Sockets (1 GbE)	Equal (with think time)	- 1.38 %
Enterprise Extender	+ 4.93 %	- 8.46 %
AT-TLS RR20 (100/800) CRR20 (64/8K) STR5 (20M/1)	 - 1.07 % + 8.56 % - 0.51 %	 + 1.2 % - 7.6 % - 3.6 %

- On average, z/OS V1R11 increases throughput by 1.23% for these workloads.
- On average, z/OS V1R11 reduces CPU cost by 5.61% for these workloads.

QDIO Performance Enhancements (OSA Latency Optimization)



- OLM (Optimized Latency Mode) vs. Minlatency
- Request-Response workload measuring the round-trip latency of a transaction
- RR1 (1/1): 1 session, TCP, 1 / 1
- RR20 (128/1024): 20 sessions, TCP, 128 / 1024
- RR40 (128/1024): 40 sessions, TCP, 128 / 1024
- RR80 (128/1024): 80 sessions, TCP, 128 / 1024
- RR20/RR60 (128/1024 and 100/800): Mixed workload 20 and 60 sessions, TCP, 128 / 1024 and 100 / 800
- Hardware: z10 using OSA-E3 (1 GbE)
- Software: z/OS V1R11

- z/OS V1R11 with OLM provides 22.25 to 54.07% lower latency compared to V1R11 with Minlatency (Avg= 44.12% lower).

For more information

URL	Content
http://www.twitter.com/IBM_Commserver	IBM Communications Server Twitter Feed
http://www.facebook.com/IBMCommserver	IBM Communications Server Facebook Fan Page
http://www.ibm.com/systems/z/	IBM System z in general
http://www.ibm.com/systems/z/hardware/networking/	IBM Mainframe System z networking
http://www.ibm.com/software/network/commserver/	IBM Software Communications Server products
http://www.ibm.com/software/network/commserver/zos/	IBM z/OS Communications Server
http://www.ibm.com/software/network/commserver/z_lin/	IBM Communications Server for Linux on System z
http://www.ibm.com/software/network/ccl/	IBM Communication Controller for Linux on System z
http://www.ibm.com/software/network/commserver/library/	IBM Communications Server library
http://www.redbooks.ibm.com	ITSO Redbooks
http://www.ibm.com/software/network/commserver/zos/support/	IBM z/OS Communications Server technical Support – including TechNotes from service
http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/TechDocs	Technical support documentation from Washington Systems Center (techdocs, flashes, presentations, white papers, etc.)
http://www.rfc-editor.org/rfcsearch.html	Request For Comments (RFC)
http://www.ibm.com/systems/z/os/zos/bkserv/	IBM z/OS Internet library – PDF files of all z/OS manuals including Communications Server

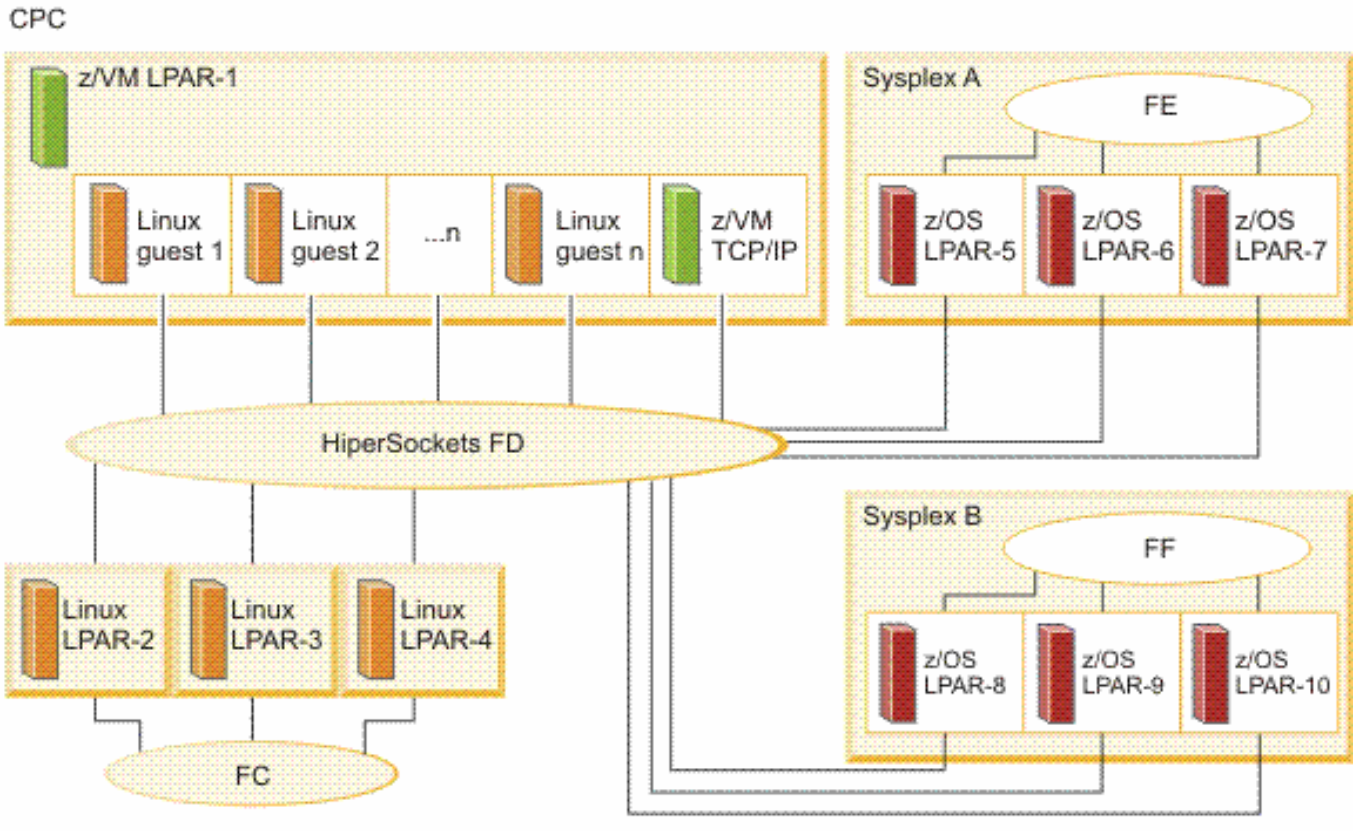
For pleasant reading ….

Improving OSA Performance with z/OS Communications Server

Appendix A: HiperSockets

HiperSockets Introduction

- HiperSockets is a technology that provides high-speed internal TCP/IP connectivity between logical partitions within a System z.



HiperSockets Introduction

- The HiperSockets implementation is based on the OSA-Express Queued Direct I/O (QDIO) protocol, hence HiperSockets is also called internal QDIO, or IQDIO.
- The communication is through the system memory of the processor, so servers are connected to form an "internal LAN."
- Eliminates the need for any physical cabling or external networking connection between servers running in different LPARs
- Since HiperSockets does not use an external network, it can free up system and network resources, eliminating attachment costs while improving availability, performance and security.
- Recent performance runs show 12Gbps+ for certain workloads
- HiperSockets Implementation Guide
  http://www.redbooks.ibm.com/redbooks/pdfs/sg246816.pdf

HiperSockets Benefits

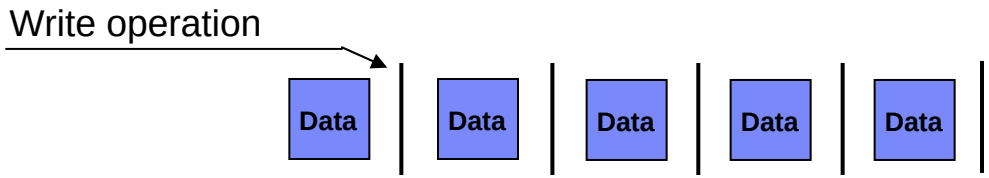
- High performance: Consolidated servers that have to access corporate data residing on the System z can do so at memory speeds with latency close to zero, by bypassing all the network overhead and delays.
- Availability: With HiperSockets, there are no network hubs, routers, adapters, or wires to break or maintain. The reduced number of network external components greatly improves availability.
- Secure: Because there is no server-to-server traffic outside the System z, HiperSockets has no external components, and therefore it provides a very secure connection.
  - Supports multiple VLANs on a single HiperSockets CHPID
- HiperSockets can also improve TCP/IP communications within a Sysplex environment when the DYNAMICXCF facility is used.

Improving OSA Performance with z/OS Communications Server

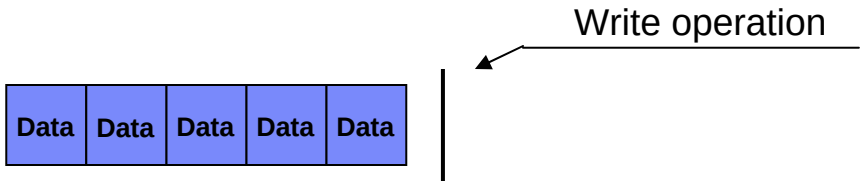
zIIP Assisted HiperSockets Multiple Write

HiperSockets Multiple Write

- HiperSockets can now be configured to move multiple output data buffers in one write operation. (V1R10; enabled via PTF in V1R9)
  - **Disabled**: 1 output data buffer is moved in 1 write operation



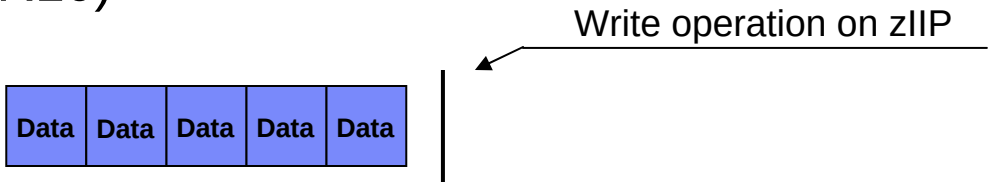
  - **Enabled**: multiple output data buffers are moved in 1 write operation, reducing CPU utilization related to large outbound messages.



zIIP Assisted HiperSockets Multiple Write

- IBM System z10 Integrated Information Processor (zIIP)
  - A specialty central processing unit (CPU) designed to free up general computing capacity and lower software costs for select workloads.

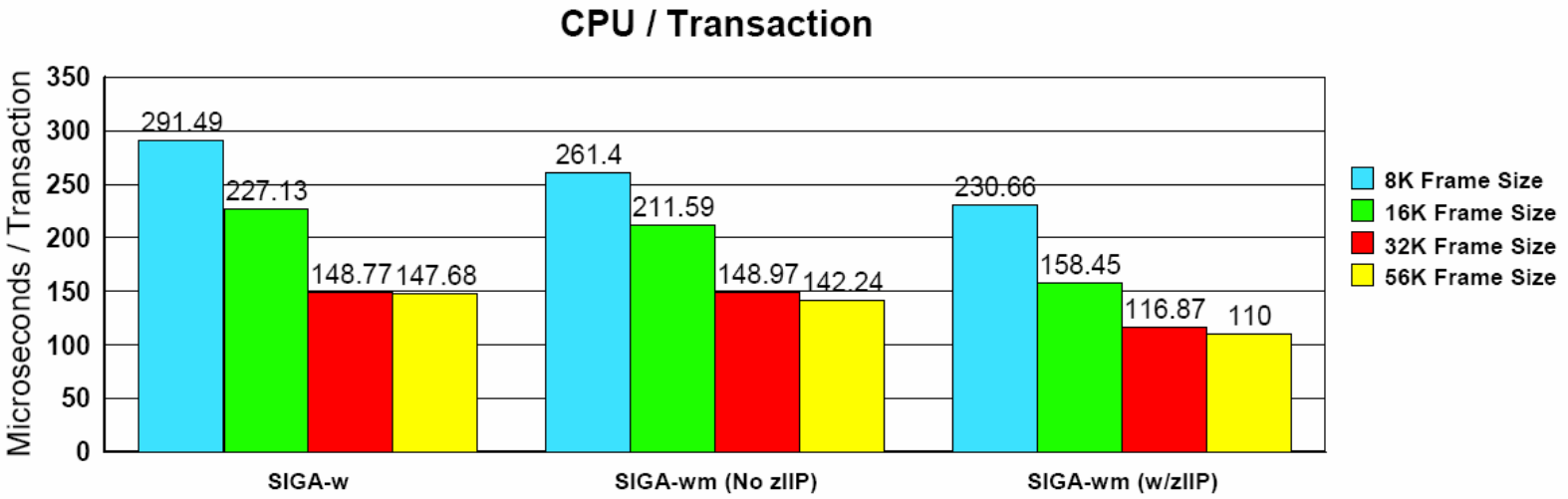
- HiperSockets can now process large outbound messages on an available zIIP. (V1R10)



  - Reduces general CPU usage and software licensing costs
  - Asynchronously moves data without blocking the sending application
  - Application socket send size must be >= 32k to be eligible for zIIP

zIIP Assisted HiperSockets Multiple Write

SIGA-wm and zIIP Assist for HiperSockets Performance (CPU/Transaction)



- Workload: RR, 4 sessions, 65000/65000
- SIGA-w: SIGA write, SIGA-wm (No zIIP): SIGA write multiple (No zIIP), SIGA-wm (w/zIIP): SIGA write multiple (with zIIP)
- SIGA-wm is only used for HiperSockets and when the data size is greater than 32 KB
- All transactions are memory to memory (No DASD used)
- Hardware: z10 (4 CPs) using HiperSockets for SIGA-w and SIGA-wm (No zIIP), z10 (4 CPs, 2 zIIPs) using HiperSockets for SIGA-wm (w/zIIP)
- Software: z/OS V1R10

- z/OS V1R10 SIGA-wm (No zIIP) provides 10.3 % lower to 0.13 % higher CPU cost per transaction compared to V1R10 SIGA-w (Avg= 5.2 % lower)
- z/OS V1R10 SIGA-wm (w/zIIP) provides 20.9 % to 30.2 % lower CPU cost per transaction compared to V1R10 SIGA-w (Avg= 24.5 % lower)

zIIP Assisted HiperSockets Multiple Write

- HiperSockets Multiple Write

```
>>-GLOBALCONFig-----------------------------------------------------><
.
.
>----+-----------------------------------------------+-+-------><
     | .-NOIQDMULTIWRITE-.                           |
     +-+-----------------+---------------------------+
       '-IQDMULTIWRITE---'
```

- zIIP-Assisted HiperSockets Multiple Write

```
>>-GLOBALCONFig-----------------------------------------------------><
.
.
>----+-----------------------------------------------+-+-------><
     |        .-NOIPSECURITY-.                       |
     +-ZIIP-+-+-------------+------+----------------+
            | '-IPSECURITY---'      |
            | .-NOIQDIOMULTIWRITE-. |
            +-+-------------------+-+
              '-IQDIOMULTIWRITE---'
```

zIIP Assisted HiperSockets Multiple Write

- Shows if HiperSockets Multiple Write is enabled for an interface and whether a zIIP will be used (if available).

```
NETSTAT DEVLINKS
MVS TCP/IP NETSTAT CS V1R10       TCPIP Name: TCPCS
14:23:39
DevName: IUTIQDIO           DevType: MPCIPA
  DevStatus: Ready
  LnkName: IQDIOLNK0A3D0001  LnkType: IPAQIDIO   LnkStatus:
Ready
    IpBroadcastCapability: No
    CfgRouter: Non                    ActRouter: Non
    ArpOffload: Yes                   ArpOffloadInfo: No
    ActMtu: 8192
    ReadStorage: GLOBAL (2048K)
    SecClass: 255
    IQDMultiWrite: Enabled (ZIIP)
  BSD Routing Parameters:
    MTU Size: 8192              Metric: 00
    DestAddr: 0.0.0.0           SubnetMask: 255.255.0.0
 .
 .
```

Improving OSA Performance with z/OS Communications Server

Appendix B: Virtual MAC Address VMAC

Virtual MAC Address (VMAC)

- Gives each INTERFACE (statement) its own virtual MAC address instead of one physical MAC address for all interfaces (like having a virtual OSA)
- One VMAC per IP version (IPv4 and IPv6) per INTERFACE
- Support added in z/OS Communications Server V1R8 (OSA-E2 & E3)
- Solves many OSA sharing, forwarding, and load balancing issues
- PRIROUTER/SECROUTER is ignored if VMAC specified
  - VMAC simplifies OSA sharing, no longer need a PRIROUTER
  - True for DEVICE/LINK and INTERFACE
  - PRIROUTER/SECROUTER now only applies to stacks sharing the OSA that do not use VMAC
- VMAC is required for some features such as QDIO Inbound Workload Queueing
- It is recommended VMACs be used anytime the OSA is shared.

Virtual MAC Address (VMAC): Configuration

- Configure VMAC on the INTERFACE statement

```
>>-INTERFace--intf_name----------------------------------------->
  .
>--+-------------------------------+-------------------------|
   |                    .-ROUTEALL-. |
   '-VMAC--+---------+--+----------+-'
           '-macaddr-'  '-ROUTELCL-'
  .
```

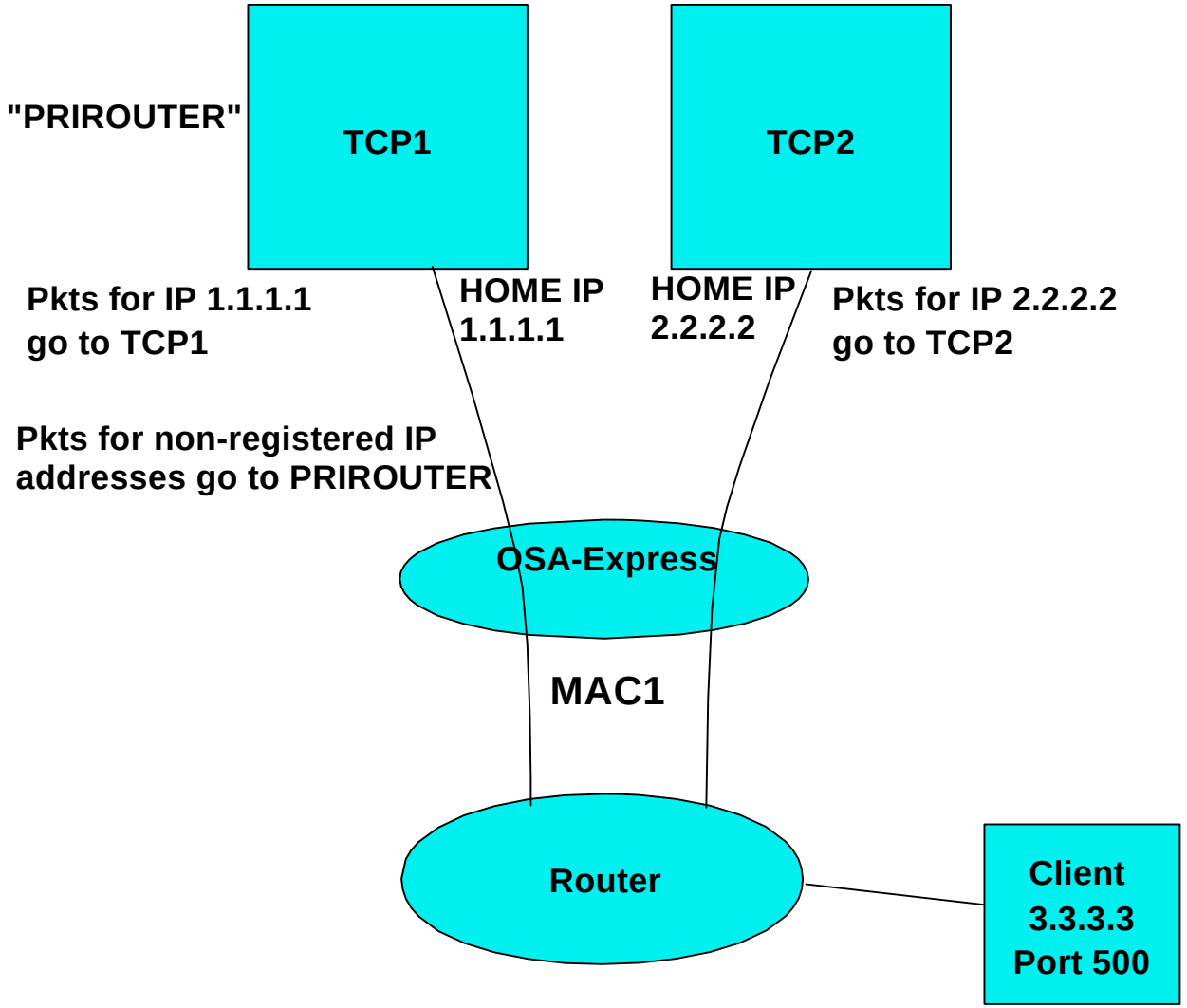
- Appendix has VMAC address scheme (for both OSA and user generated)
- ROUTEALL - route all packets destined for the VMAC to this stack
  - Even if IP address not registered
  - This is the default
- ROUTELCL - only route packets that match IP addresses in home list
  - Note: Use only if this stack will not forward OSA traffic

VMAC Address Scheme

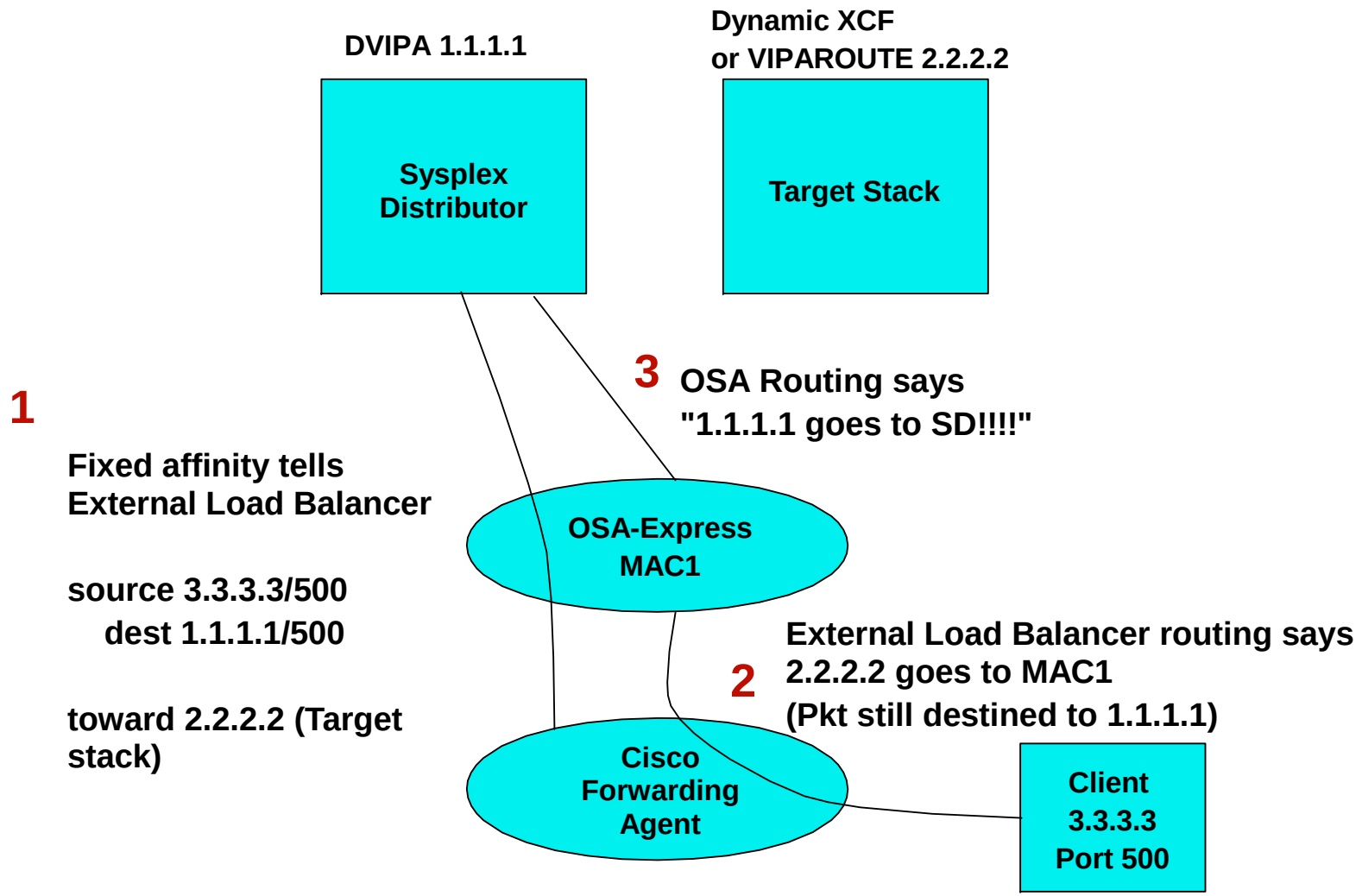
- **OSA Generated**
  - OSA's VMAC generation scheme, to guarantee uniqueness, is as follows:
    - First byte of VMAC will be a constant 02. The 2 bit indicates this is a locally administered MAC address. This will guarantee it is unique from all physical "burned-in" MACs, since the 2 bit is off, indicating they are "universal" addresses.
    - The last 3 bytes will be the last 3 bytes of the physical MAC address. This will guarantee all VMACs on one OSA will be unique from all other VMACs on any other OSA.
    - To guarantee stacks sharing an OSA will get unique addresses, the second and third bytes of the VMAC will be an instance count, incremented each time OSA gives out a VMAC address.
  - TCP/IP will reuse the same generated VMAC address when a device becomes inactive and is reactivated. A new VMAC address will be generated for a given OSA if the stack is stopped and restarted.
- **User Configured**
  - If the VMAC is defined by the user, it must be a 12 digit hexadecimal number, with the X'02' bit in the first byte of the VMAC on, indicating this is a locally administered MAC address. It is up to the user to ensure the uniqueness of the VMAC on the local LAN on which this OSA resides.

Sharing OSAs and PRIROUTER

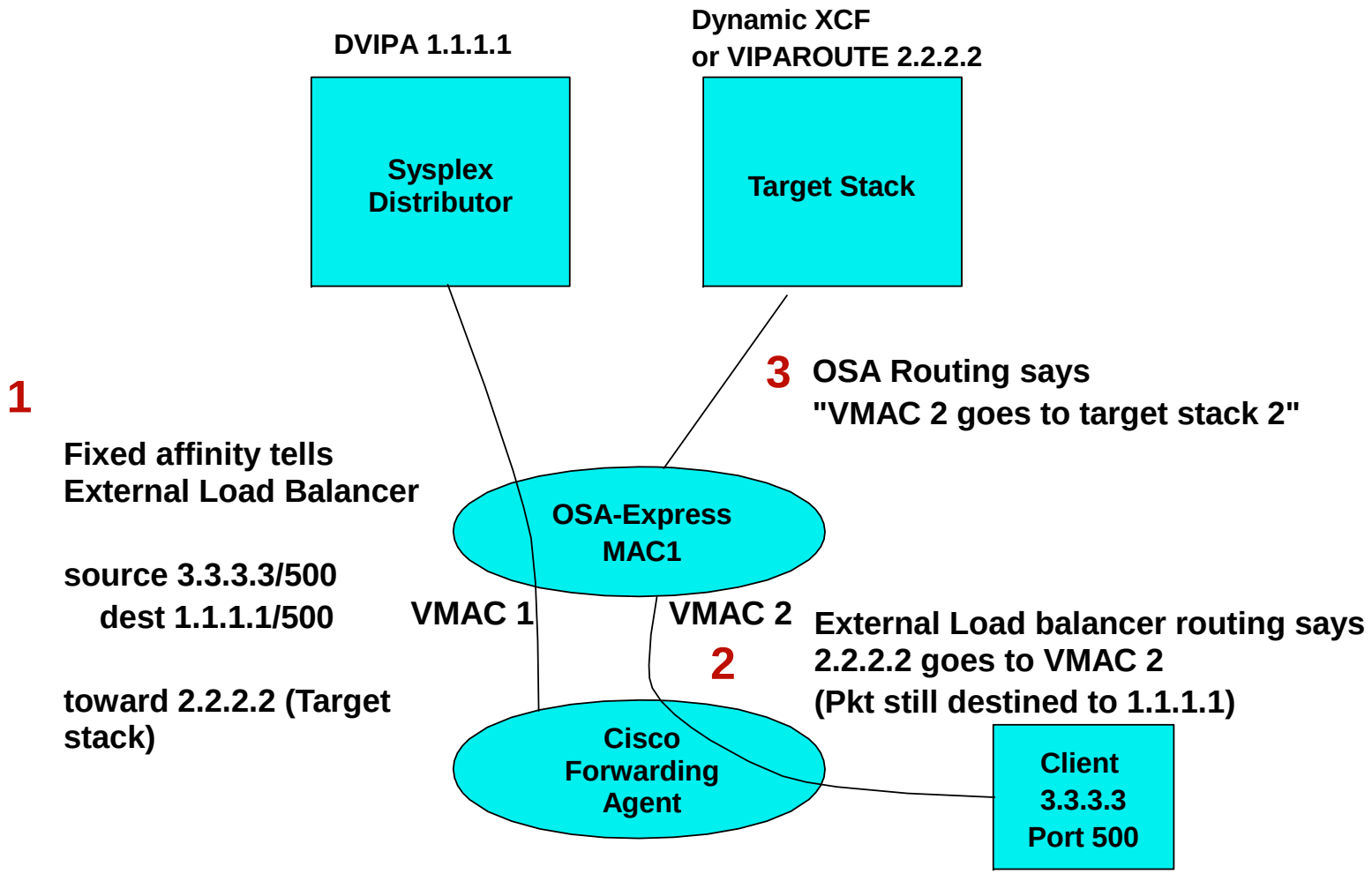
- Allows many stacks, in different LPARs, to share bandwidth
- Even more important with high bandwidth adapters (10 gig, etc)
- Accomplished by registering IP addresses, sharing "burned in" MAC
- One stack may be PRIROUTER for unknown packets



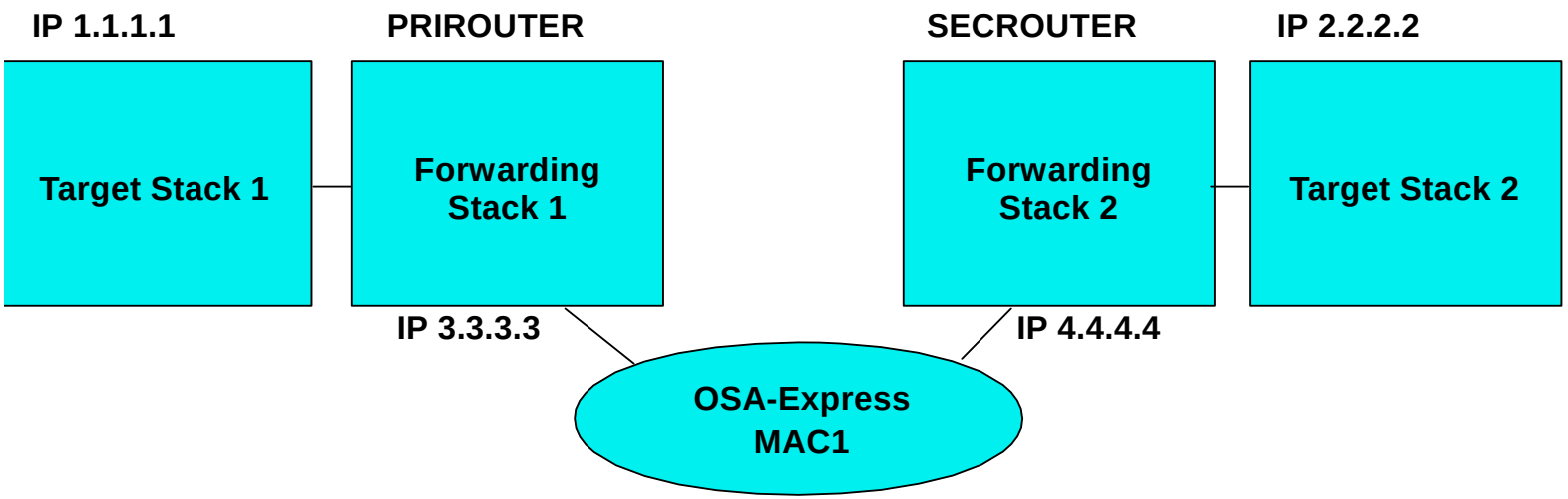
Problem: Sharing an OSA with External Load Balancing



Solution: VMAC with External Load Balancing



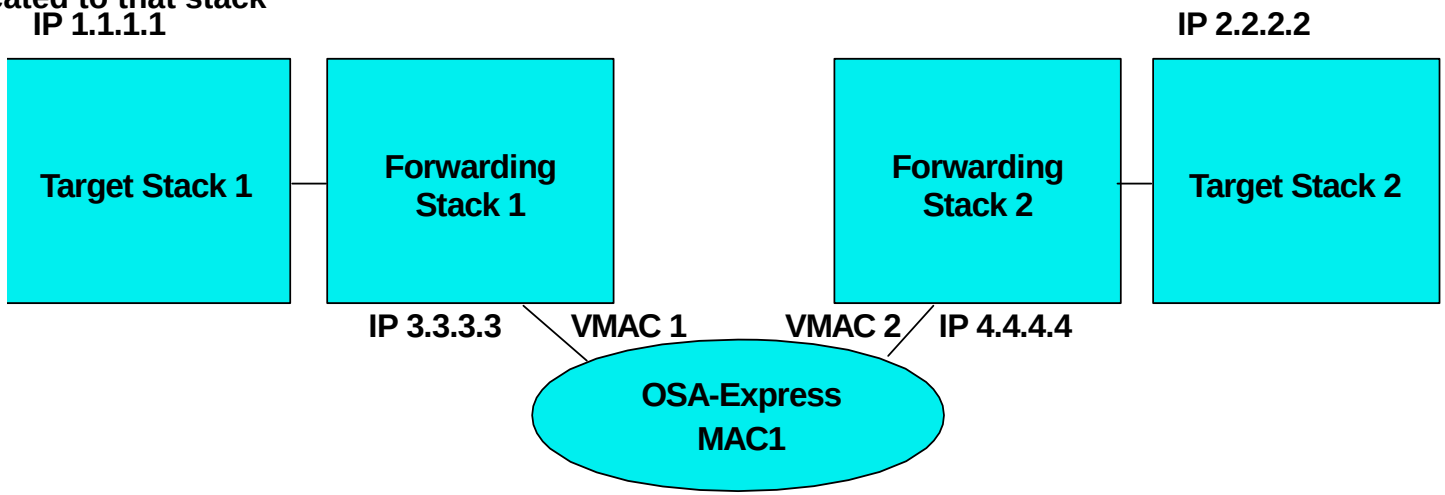
Problem: Only one Routing Stack per OSA



- Routes to IP 1.1.1.1 and 2.2.2.2 are as follows:
  - 1.1.1.1 has a hop through 3.3.3.3
    - Any pkt with hop of 3.3.3.3 goes to MAC1
  - 2.2.2.2 has a hop through 4.4.4.4
    - Any pkt with hop of 4.4.4.4 also goes to MAC1
- OSA gets both packets with same MAC, but....
  - doesn't know either 1.1.1.1 or 2.2.2.2
  - Sends both to PRIROUTER
  - 2.2.2.2 pkt is discarded
- Also, if Stack 2 is SECROUTER
  - Not predictable who is doing routing
  - If Stack 1 is recycled, Stack 2 is ROUTER

Solution: VMAC and Multiple Routing Stacks

- Each OSA appears as a "virtual" OSA dedicated to that stack
- No definition of PRIROUTER/SECROUTER



- Routes to IP 1.1.1.1 and 2.2.2.2 are as follows:
  - 1.1.1.1 has a hop through 3.3.3.3, goes to VMAC1
  - 2.2.2.2 has a hop through 4.4.4.4, goes to VMAC2
- OSA doesn't know either 1.1.1.1 or 2.2.2.2, but...
  - Sends 1.1.1.1 pkt with VMAC1 to Stack 1
  - Sends 2.2.2.2 pkt with VMAC2 to Stack 2

VMAC – Displaying Configuration

- Use Nestat DEvlinks/-d or Display OSAINFO to show VMAC configuration

```
netstat -d -p tcpcs1 -K OSAQDIOINTF
.
IntfName: OSAQDIOINTF       IntfType: IPAQENET   IntfStatus: Ready
    PortName: OSAQDIO2  Datapath: 0E2A     DatapathStatus: Ready
    ChpidType: OSD
    Speed: 0000000100
    IpBroadcastCapability: No
    VMacAddr:   020629DC21BD  VMacOrigin: Cfg  VMacRouter: All
    SrcVipaIntf: VIPAV4
    CfgRouter: Non                      ActRouter: Non
    ArpOffload: Yes                     ArpOffloadInfo: Yes
    CfgMtu: 1492                        ActMtu: 1492
    IpAddr: 100.1.1.1/24
    VLANid: 1261                        VLANpriority: Enabled
    DynVLANRegCfg: Yes                  DynVLANRegCap: No
.
```