Improving OSA Performance with z/OS Communications Server

Hugh Hockett – hhockett@us.ibm.com
IBM, Raleigh, NC, US

Thursday August 5, 2010 - 9:30 AM
**Improving OSA performance with z/OS Communications Server**

<table>
<thead>
<tr>
<th>Date and time:</th>
<th>Thursday August 5, 2010 - 9:30 AM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location:</td>
<td>Room 109 (Hynes Convention Center)</td>
</tr>
<tr>
<td>Program:</td>
<td>Communications Infrastructure</td>
</tr>
<tr>
<td>Project:</td>
<td>Communications Server</td>
</tr>
<tr>
<td>Track:</td>
<td>SNA/IP Integration, If you are in Network support and management</td>
</tr>
<tr>
<td>Classification:</td>
<td>Technical</td>
</tr>
<tr>
<td>Speaker:</td>
<td>Hugh Hockett, IBM</td>
</tr>
<tr>
<td>Abstract:</td>
<td>There are a number of performance features available on z/OS Communications Server and the OSA platform. Are you getting the most out of them? This session will discuss a number of OSA related z/OS Communications Server features that will help you speed up your data, reduce latency, and reduce CPU utilization. Topics that will be covered in this session include Dynamic LAN Idle Timer, Optimized Latency Mode, TCP Segmentation Offload, and QDIO Accelerator. This session will also introduce you to the OSA-Express 3 and the latest V1R12 performance feature called QDIO Inbound Workload Queueing.</td>
</tr>
</tbody>
</table>
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
- 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 Addr (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](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 SOURCEVIPAIN INTERFACE 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%.

```
RR1 and RR10 Dynamic Lan Idle Performance

<table>
<thead>
<tr>
<th>% (Dynamic LAN Idle relative to Balanced)</th>
<th>RR1(1h/8h)</th>
<th>RR10(1h/8h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trans/sec</td>
<td>23.8</td>
<td>30.2</td>
</tr>
<tr>
<td>Response Time</td>
<td>-19.2</td>
<td>-22.1</td>
</tr>
<tr>
<td>CPU cost/Trans</td>
<td>-22.1</td>
<td>-22.6</td>
</tr>
</tbody>
</table>

1h/8h indicates 100 bytes In and 800 bytes out
```
Dynamic LAN Idle Timer: Configuration

- Configure INBPERF DYNAMIC on the INTERFACE statement

```plaintext
>>-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
- 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

Generally Recommended!
**Dynamic LAN Idle Timer: Display Configuration**

- Use `Netstat DEvlkinks/-d` to display the current INBPERF setting

```plaintext
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

- **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).
**OML Sample Target Customer Environment**

*(2 tier DB2 system example)*

System z

Response time (latency) is critical!

- **2nd tier (backend)**
  - DB2 (2nd tier)
  - OSAs dedicated to single z/OS LP 1

- **1st tier (front end)**
  - Many front end distributed application servers
  - Create many back end DB2 DRDA (TCP/IP) connections generating high volume transaction workload rate

Requires high QoS standards for response time!
**Optimized Latency Mode (OLM): How to configure**

- New OLM parameter
  - IPAQENET/IPAQENET6
  - **Not** allowed on DEVICE/LINK

- Enables Optimized Latency Mode for this INTERFACE only
- Forces INBPERF to DYNAMIC
- Default NOOLM

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

- 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 WLM_PRIORITYQ (default is ok) or SET_SUBNET_PRIORITY_OSMASK to put data on queues 1-3
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):**

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

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

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

<table>
<thead>
<tr>
<th>TcpHdr</th>
<th>IO F</th>
<th>Seq</th>
<th>Ack</th>
<th>RcvWnd</th>
<th>Data</th>
<th>Delta Time</th>
<th>TimeStamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>I</td>
<td>458316454</td>
<td>0</td>
<td>32768</td>
<td>0</td>
<td>0.000000</td>
<td>00:54:40.567581</td>
</tr>
<tr>
<td>A</td>
<td>S</td>
<td>456248587</td>
<td>458316455</td>
<td>32768</td>
<td>0</td>
<td>0.050860</td>
<td>00:54:40.618441</td>
</tr>
<tr>
<td>A</td>
<td>I u</td>
<td>458316455</td>
<td>456248588</td>
<td>32768</td>
<td>0</td>
<td>0.077633</td>
<td>00:54:40.696074</td>
</tr>
<tr>
<td>O</td>
<td>AP</td>
<td>456248588</td>
<td>458316455</td>
<td>32768</td>
<td>5752</td>
<td>7.023527</td>
<td>00:54:47.719601</td>
</tr>
<tr>
<td>A</td>
<td>I a</td>
<td>458316455</td>
<td>456254340</td>
<td>27016</td>
<td>0</td>
<td>0.069695</td>
<td>00:54:47.789296</td>
</tr>
<tr>
<td>O</td>
<td>AP</td>
<td>456254340</td>
<td>458316455</td>
<td>32768</td>
<td>7190</td>
<td>0.004164</td>
<td>00:54:47.793460</td>
</tr>
<tr>
<td>A</td>
<td>I a</td>
<td>458316455</td>
<td>456261530</td>
<td>19826</td>
<td>0</td>
<td>0.063442</td>
<td>00:54:47.856902</td>
</tr>
</tbody>
</table>

**Data Segment Stats:**

- **Inbound:**
  - Number of data segments: 0
  - Maximum segment size: 1460
  - Largest segment size: 0
  - Average segment size: 0
  - Smallest segment size: 0
  - Segments/window: 0.0
  - Average bytes/window: 0
  - Most bytes/window: 0
  - Offload Sends: 39 (100%)
  - Offload Segments: 365
  - Offload Bytes: 500000 (100%)

- **Outbound:**
  - Number of data segments: 39
  - Maximum segment size: 1460
  - Largest segment size: 16384
  - Average segment size: 12820
  - Smallest segment size: 3808
  - Segments/window: 0.0
  - Average bytes/window: 12820
  - Most bytes/window: 16384

---

Page 31 © Copyright International Business Machines Corporation 2010 and SHARE. All rights reserved.
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)

[Graph showing performance metrics for RR20]

- Transactions / Second:
  - RR20: 1.69%

- CPU / Transaction:
  - z/OS Client:
    - RR20: -4.67%
    - z/OS Sysplex Distributor: -2.88%
  - z/OS Target:
    - RR20: -72.3%

Sysplex Distributor QDIO Accelerator (RR GbE)

[Graph showing performance metrics for RR20]

- Transactions / Second:
  - RR20: 6.89%

- CPU / Transaction:
  - z/OS Client:
    - RR20: -0.9%
    - z/OS Sysplex Distributor: -0.33%
  - z/OS Target:
    - RR20: -57.45%
**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 addr and outbound interfaces are registered in the DLC for future fast path forwarding
**Background: IP Forwarding**

- **Originating Host**
- **z/OS Forwarding Stack**
- **Destination Host**

- **Ethernet LAN**
- **OSA-E**

- **HiperSockets LAN**
QDIO Accelerator

Note: Still requires IP Forwarding be enabled
# QDIO Accelerator

<table>
<thead>
<tr>
<th>Function</th>
<th>IQDIOROUTING</th>
<th>QDIOACCELERATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSA-E → HiperSockets</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HiperSockets → OSA-E</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>OSA-E → OSA-E</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>HiperSockets → HiperSockets</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Sysplex Distributor</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
**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_________________________          |
| _QDIOACCELERator__________________________          |
| __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:

- z/OS-A
  - z/OS v1r12
  - OSA EXP-3’s in Balanced, Dynamic, or new IWQ mode
- z/OS-B
  - AIX 5.3 p570
  - 1gb or 10gb ethernet

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

```
D TCPIP,,OSAINFO,INTFN=V6O3ETHG0
.
Ancillary Input Queue Routing Variables:
  Queue Type: BULKDATA  Queue ID:  2  Protocol: TCP
  Src: 2000:197:11:201:0:1:0:1..221
  Dst: 100::101..257
  Src: 2000:197:11:201:0:2:0:1..290
  Dst: 200::202..514
  Total number of IPv6 connections:      2
  Queue Type: SYSDIST  Queue ID:  3  Protocol: TCP
  Addr: 2000:197:11:201:0:1:0:1
  Addr: 2000:197:11:201:0:2:0:1
  Total number of IPv6 addresses:      2
36 of 36 Lines Displayed
End of report
```

- 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: 00000000000023316386
BYTESOUT: 00000000000000000000
SEGMENTSIN: 000000000000016246
SEGMENTSOUT: 00000000000000000922
LAST TOUCHED: 21:38:53    STATE: ESTABLISH

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)
IST1719I PCIREALO = 0          PCIREAL  = 7629
...  
IST924I -------------------------------------------------------------
IST1233I DEV      = 0E2A       DIR      = RD/3 (SYSDIST)
IST1719I PCIREALO = 0          PCIREAL  = 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
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 online:

- 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

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

- 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

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

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
**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)

<table>
<thead>
<tr>
<th>Microseconds / Transaction</th>
<th>SIGA-w</th>
<th>SIGA-wm (No zIIP)</th>
<th>SIGA-wm (w/zIIP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8K Frame Size</td>
<td>291.49</td>
<td>148.77 147.68</td>
<td>158.45</td>
</tr>
<tr>
<td>16K Frame Size</td>
<td>227.13</td>
<td>211.59 142.24</td>
<td>116.87</td>
</tr>
<tr>
<td>32K Frame Size</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56K Frame Size</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **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
```

Shows if HiperSockets Multiple Write is enabled for an interface and whether a zIIP will be used (if available).
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--inf_name---------------------------------------------------------->
  .
>-----------------------------------------------------------------------------|
  | -ROUTEALL- |
  '-----VMAC----++-----------------------++-----' |
    '-----macaddr-'  '-----ROUTEELCL-'  
```

- 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
- ROUTEELCL - 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**

1. Fixed affinity tells External Load Balancer
   - source 3.3.3.3/500
   - dest 1.1.1.1/500
   - toward 2.2.2.2 (Target stack)

2. OSA-Express MAC1
   - Cisco Forwarding Agent
   - External Load Balancer routing says 2.2.2.2 goes to MAC1
   - (Pkt still destined to 1.1.1.1)

3. OSA Routing says "1.1.1.1 goes to SD!!!!"

Client
- 3.3.3.3
- Port 500
**Solution: VMAC with External Load Balancing**

1. Fixed affinity tells External Load Balancer
   - Source 3.3.3.3/500
   - Dest 1.1.1.1/500
   - Toward 2.2.2.2 (Target stack)

2. OSA Routing says "VMAC 2 goes to target stack 2"
   - External Load balancer routing says 2.2.2.2 goes to VMAC 2
     - (Pkt still destined to 1.1.1.1)

3. Dynamic XCF or VIPAROUTE 2.2.2.2

---

**Diagram:**
- DVIPA 1.1.1.1
- Dynamic XCF or VIPAROUTE 2.2.2.2
- Sysplex Distributor
- Target Stack
- OSA-Express MAC1
- VMAC 1
- VMAC 2
- Cisco Forwarding Agent
- Client 3.3.3.3 Port 500
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

Also, if Stack 2 is SECROUTER
- Not predictable who is doing routing
- If Stack 1 is recycled, Stack 2 is ROUTER

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
**Solution: VMAC and Multiple Routing Stacks**

- Each OSA appears as a “virtual” OSA dedicated to that stack.
  - IP 1.1.1.1

- No definition of PRIROUTER/SECROUTER
  - IP 2.2.2.2

- 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 tcp -s -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
  ArpOffload: Yes
  CfgMtu: 1492
  IpAddr: 100.1.1.1/24
  VLANid: 1261
  DynVLANRegCfg: Yes

  ArpOffloadInfo: Yes
  ActMtu: 1492
  VLANpriority: Enabled
  DynVLANRegCap: No
  ActRouter: Non
```