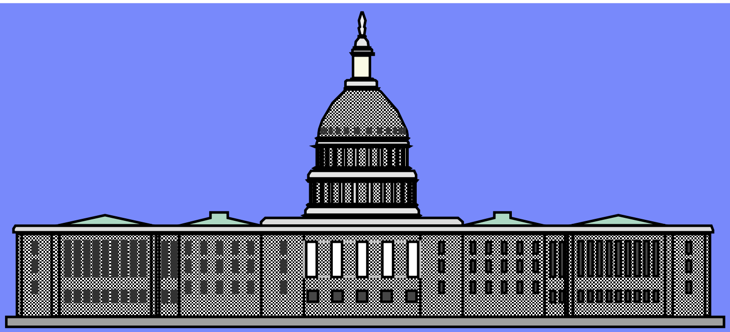
z/OS Communications Server TCP/IP Performance

Linda Harrison
lharriso@us.ibm.com
Washington System Center



Trademarks

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

- **The following are trademarks or registered trademarks of other companies.**
  - Microsoft is a registered trademark of Microsoft Corporation in the United States and other countries.

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

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

Agenda

- Work Load Manager (WLM)
- Delay Acknowledgements
- Fin Wait 2 Timer
- Limits
- Routing
- HiperSockets, Shared Memory Communication, and OSA
- Keep Alive
- Window Size
- Encryption

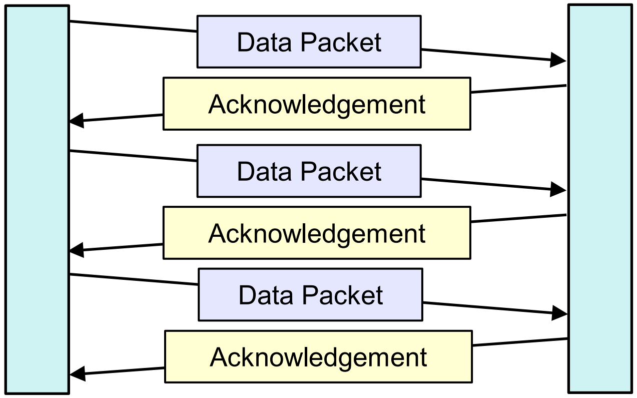
Work Load Manager (WLM)

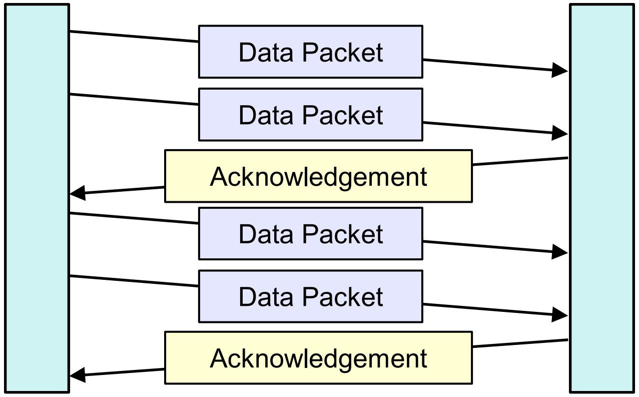
Work Load Manager (WLM)

- VTAM and TCP/IP should be set at a higher dispatching priority than that of the applications that use their services.
- Server applications (OMPROUTE, TN3270E Telnet server, IKED, NSSD, and FTPD) should be set at the same priority value as TCP/IP, or to a priority that is just below that value.
  - Assign these tasks to the SYSSTC service class
  - Make these tasks non-swappable.
- On systems with significant FTP activity, you can improve performance by placing the FTP program objects into the dynamic link pack area (LPA).

Delay Acknowlegements

Delay Acknowledgements

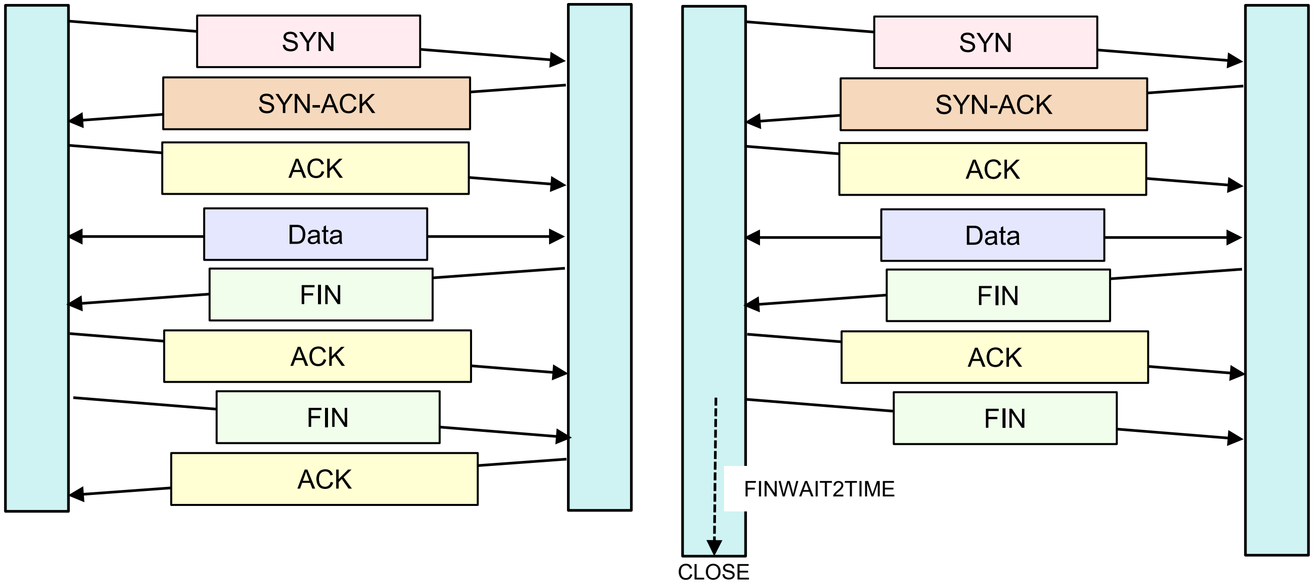




- TCPCONFIG DELAYACKS
  - This useful parameter is enabled by default.
  - By only requiring an acknowledgement to be sent for every other packet rather than every single packets, this is usually a performance improvement.
  - However, some applications wait for acknowledgement prior to sending more data. In this case, disabling Delay Acknowledgements improves performance.
- TCPCONFIG AUTODELAYACKS
  - The best of both worlds. Delay Acknowledgements will be used unless it is automatically detected that NODELAYACKS is needed, at which time it will be implemented.

Fin Wait 2 Timer

Fin Wait 2 Time



- TCPCONFIG FINWAIT2TIME
- Finish Wait 2 Time is the amount of time that a session waits in the FINWAIT2 state before closing it.
- The 600 second default is usually sufficient, because this allows applications to finish closing sessions in that amount of time.
- However, some applications leave sessions hanging in the FINWAIT2 state. In this case, lowering the FINWAIT2TIME can reduce hung sessions.
  - Be aware that a lower value will incur a higher overhead.

Limits

TCP/IP Procedure

- The REGION parameter on the EXEC statement in the startup procedure for a program specifies how much virtual storage the program is enabled to allocate.
  - 0 MB allows the program to allocate all of the storage it requires below and above the 16 MB line.
  - A value greater than 0 MB and less than or equal to 16 MB establishes the size of the private area below 16 MB. The extended region size (above the 16 MB line) is the default value of 32 MB.
  - A value greater than 16 MB and less than or equal to 32 MB gives the program all the storage available below 16 MB. The extended region size is the default value of 32 MB.
  - A value greater than 32 MB gives the program all the storage available below 16 MB. The extended region size is the specified value.
- REGION=0 is recommended in the TCP/IP procedure.
- REGION=0M will set MEMLIMIT=NOLIMIT. However, this NOLIMIT value may have been overriden to a lower value by a MEMLIMIT setting on JOB or EXEC statements.

ECSA and Pool

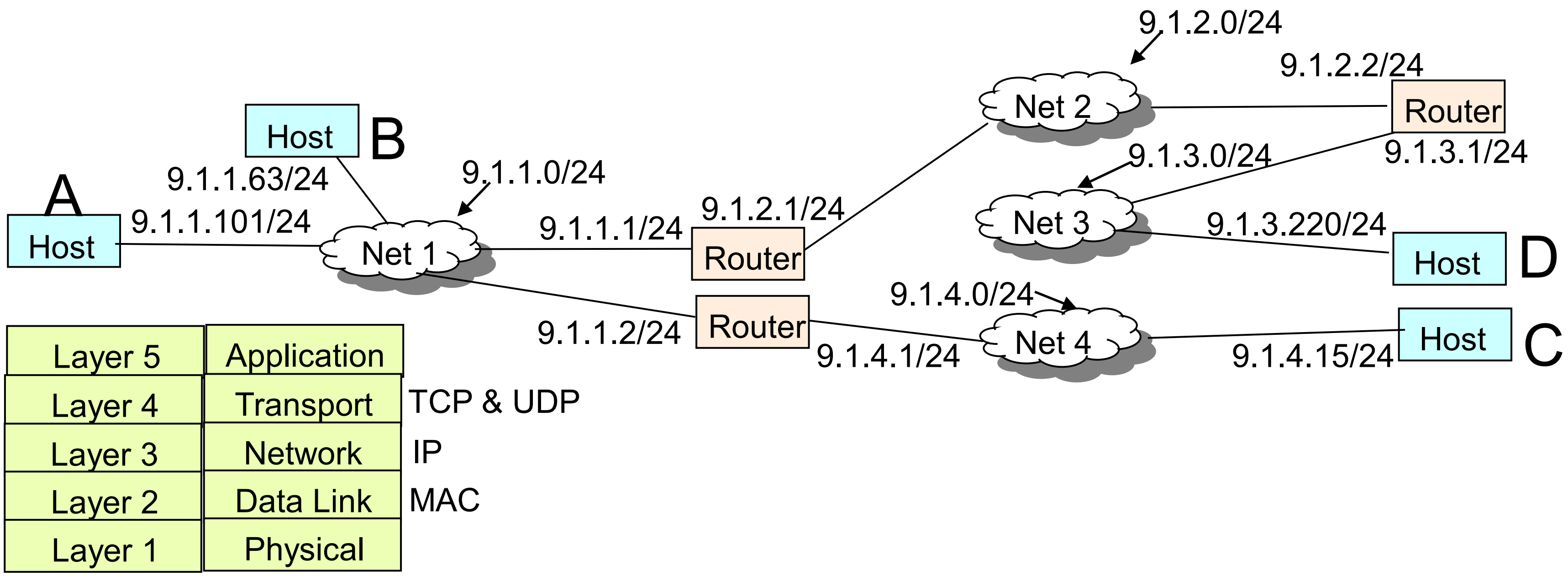
- GLOBALCONFIG ECSALIMIT
- GLOBALCONFIG POOLLIMIT
- In a perfect world, if you have plenty of storage on your system then defining limits is not needed.
- If you define storage limits and your system reaches those limits, then VTAM and TCP/IP function will be impacted. On the other hand, when storage limits are defined then useful messages are issued when storage consumption approaches the limits. In addition, when limits are reached applications may have the storage they need to finish important tasks before the storage shortage stops all activity.
- In order to define limits, you must monitor usage and set the limits high enough to handle your highest normal peaks.
  - DISPLAY TCPIP,,STOR

Unix System Services (USS)

- There are several unix limits that should be monitored and increased as needed.
  - D OMVS,L
    - MAXPROCSYS - Maximum processes
    - MAXPROCUSER - Maximum processes per user
    - MAXPTYS - Maximum number of shell sessions
    - MAXSOCKETS - Maximum sockets
    - MAXTHREADS - Maximum threads
    - MAXTHREADTASKS - Maximum number of tasks per process
    - MAXUIDS - Maximum users
    - MAXUSERMOUNTSYS – Maximum number of non-privileged user mounts
    - MAXUSERMOUNTUSER – Maximum number of non-privileged user mounts per user
    - PRIORITYGOAL – List of service class names when in goal mode
    - PRIORITY PG – List of performance group numbers when in goal mode

Routing

Layer 2 versus Layer 3 Routing



Layer 5	Application	
Layer 4	Transport	TCP & UDP
Layer 3	Network	IP
Layer 2	Data Link	MAC
Layer 1	Physical	

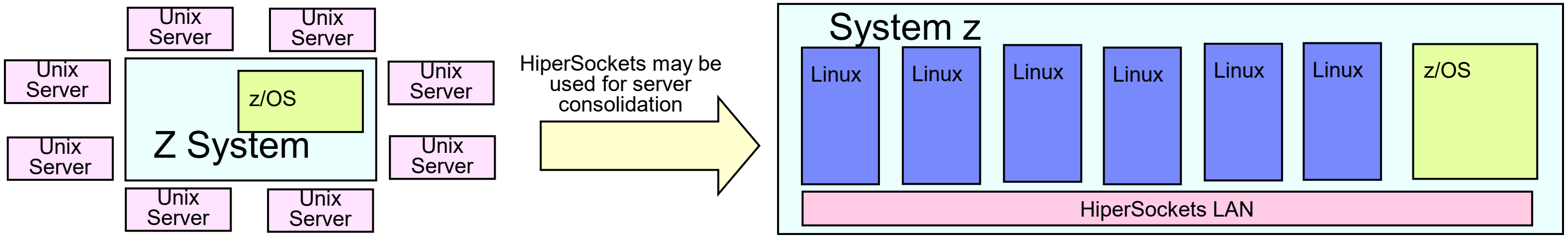
- Hosts in the same subnet avoid Layer 3 IP Routing.
  - Hosts in the same subnet use what is also referred to as Layer 2 routing.
  - Performance is always better when Layer 3 IP Routing is avoided.
    - When a high volume of data is passed between two hosts, the performance is better if they are in the same subnet.

Static versus Dynamic Routing

- Static Routing
  - Uses Gratuitous ARP Takeover for dynamic failover/recovery when an outage occurs.
  - All OSA and VIPA defined to TCP/IP should all be in the same subnet.
- Dynamic Routing
  - Uses Open Shortest Path First (OSPF) for dynamic failover/recovery when an outage occurs.
  - All OSA should be in different subnets.
  - All VIPA should be in a different subnet and the OSAs.
- Otherwise, the two different failover/recovery mechanisms can interfere with each other.

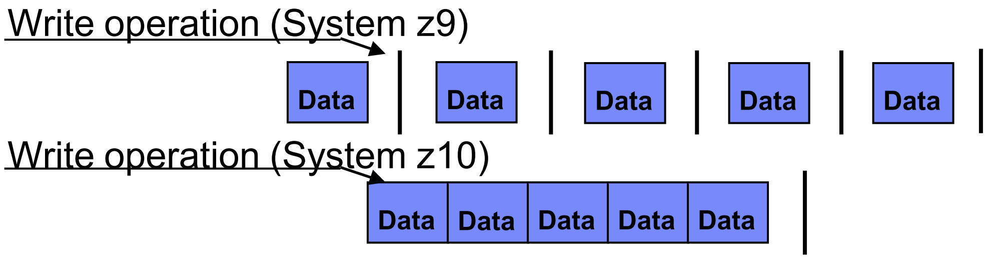
HiperSockets, Shared Memory Communication, and OSA

HiperSockets (iQDIO)



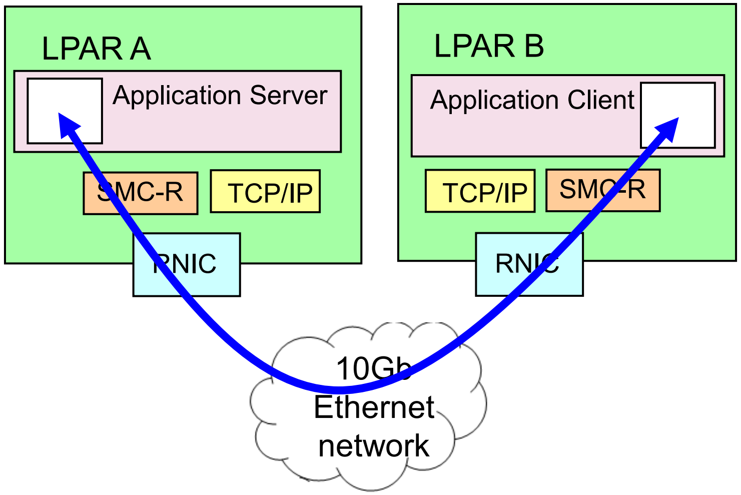
- HiperSockets = Internal Queued Direct Input Output (iQDIO)
  - Was developed from the OSA QDIO architecture
    - INTERFACE IPAQIDIO
  - Also known as HiperSockets device or Z System internal virtual LAN or HiperSockets LAN
  - LPAR to LPAR communication via shared memory
    - High speed, low latency, similar to cross-address-space memory move using memory bus
    - Provides better performance than channel protocols for network access.
  - Multiple HiperSockets may be configured as internal LANs on the Z System box.
  - A HiperSockets LAN may be configured to be part of TCP/IP DynamicXCF.
    - HCD IQDCHPID

HiperSockets Multiple Write Facility and zIIP Offload



- GLOBALCONFIG IQDMULTIWRITE ZIIP IQDIOMULTIWRITE
- HiperSockets can move multiple output data buffers in one write operation
  - ➢ Reduces CPU utilization
- Multiwrite operation can be offloaded to a zIIP
  - ➢ Only for TCP traffic that originates in this host
    - Only large TCP outbound messages (32KB+)

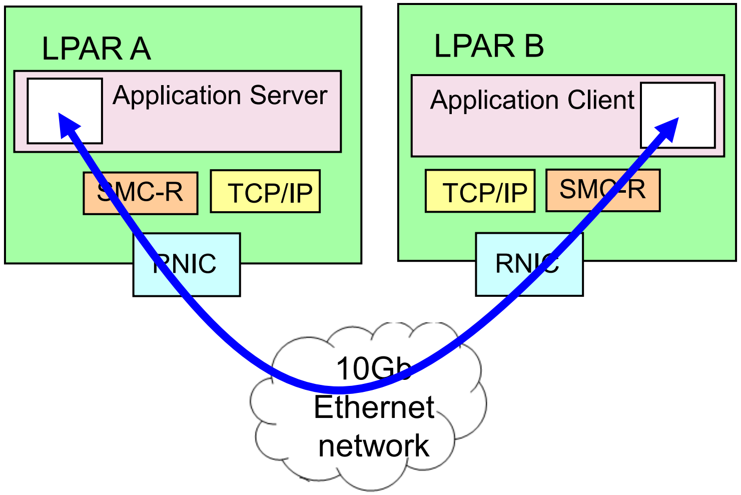
SMC-R



- Shared Memory Communication over RDMA (SMC-R)
  - Is a new sockets over RDMA communication protocol that allows existing TCP applications to transparently benefit from RoCE.
  - Requires no application change.
  - Provides host-to-host direct memory access without the traditional TCP/IP processing overhead.
  - Allows customers to benefit from InfiniBand technology by leveraging their existing 10GbE Ethernet infrastructure.
  - TCP protocol only! No UDP (ie. EE), SNA, etc.
    - All TCP traffic except IPsec.
  - https://datatracker.ietf.org/doc/draft-fox-tcpm-shared-memory-rdma/

- z/OS V2R1+ includes SMC-R support.
- SMC-R is only used over the RoCE Express feature to a partner z/OS V2R1+ with RoCE.
  - While other platform (non-Z) RoCE RNICs might exist in your network, the Z RoCE Express feature is not able to communicate with them without the SMC-R protocol.

SMC-D

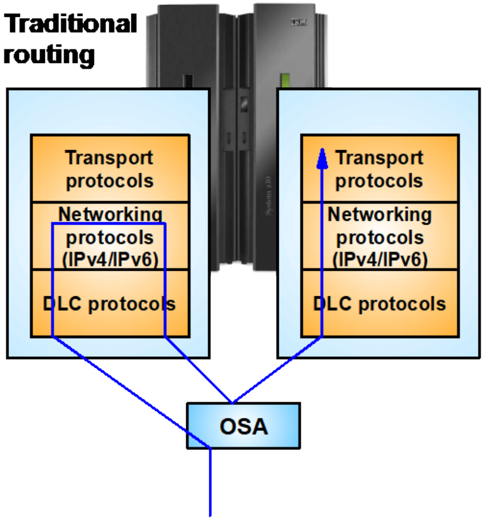


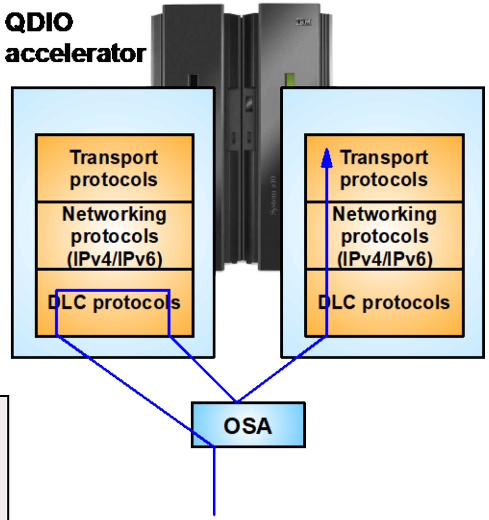
- Shared Memory Communication - Direct (SMC-D)
  - Is a new sockets over RDMA communication protocol that allows existing TCP applications to transparently benefit when on the same hardware box.
  - Requires no application change.
  - Provides host-to-host direct memory access without the traditional TCP/IP processing overhead.
  - TCP protocol only! No UDP (ie. EE), SNA, etc.
    - All TCP traffic except IPsec.

- z/OS V2R3 includes SMC-D support.
- Lower overhead and better performance than HiperSockets.

QDIO Accelerator

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





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

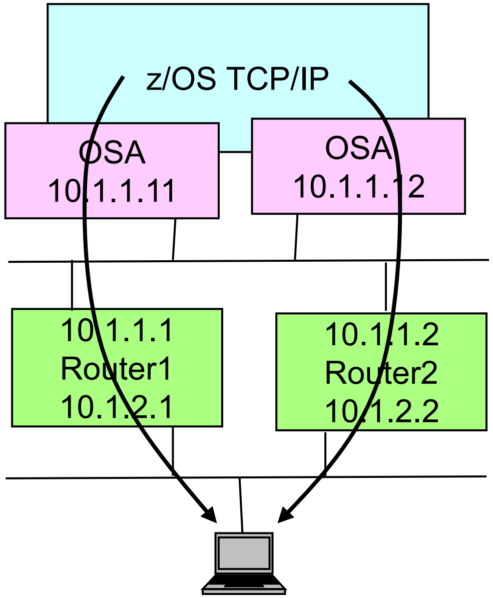
Accelerator requires IP Forwarding to be enabled for non-Sysplex Distributor acceleration.
No Acceleration for:
- IPv6
- Traffic which requires fragmentation in order to be forwarded
- Incoming fragments for a Sysplex Distributor connection
- OSA port interfaces using Optimized Latency Mode (OLM)

Maximum Transmission Unit (MTU)

- All devices on a subnet should all define the same MTU.
- Each device type has a hardware MTU associated with it.
- The MTU that is used is the least of the MTU values defined:
  - Hardware MTU
  - Interface statement MTU
  - Routing definition MTU
- If a packet is sent with a larger MTU and there is a subnet in the path with a smaller MTU, then the router “fragments” the packet into smaller pieces. The remote host “reassembles” the packet.
  - An ICMP message is sent back to the sending host to have the transmission size reduced to the new value.
  - If routers in the network block ICMP messages, then the fragmentation/reassembly continues.
- Path MTU Discovery (IPCONFIG PATHMTUDISCOVERY) discovers the lowest MTU in the path prior to sending data.
  - This uses ICMP messages, so if those are blocked in the network, then this capability will not work.
  - This parameter is recommended when defining Jumbo Frames (MTU 8992).
- Using a larger MTU, without Fragmentation, should provide better performance.

Multipath

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



- IPCONFIG MULTIPATH
  - “Load balances" outbound packets over multiple OSAs to the same destination.
  - Static Routing and OMPROUTE OSPF support Multipath.
- Is this a performance enhancement?
  - If the capacity of a single OSA is exceeded, then Multipath may very well improve performance.

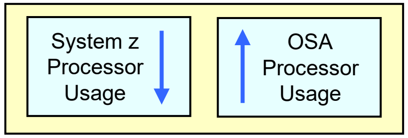
Inbound Load Balance

Inbound load balance is really determined by the first hop router.

If the first hop router is capable of load balancing traffic across multiple OSAs when the destination is a VIPA address, then inbound traffic will be truly load balanced.

Routers have static and OSPF load balancing capability similar to z/OS outbound Multipath. See
http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094820.shtml

OSA Offload

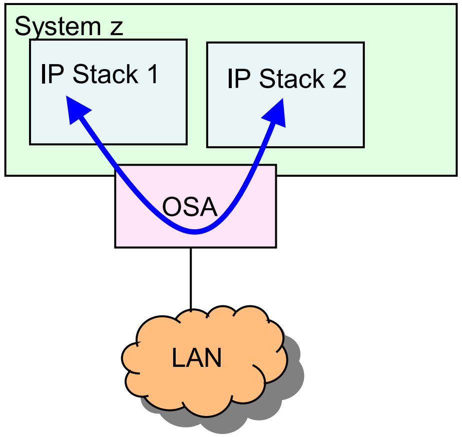


System z Processor Usage ↓

↑ OSA Processor Usage

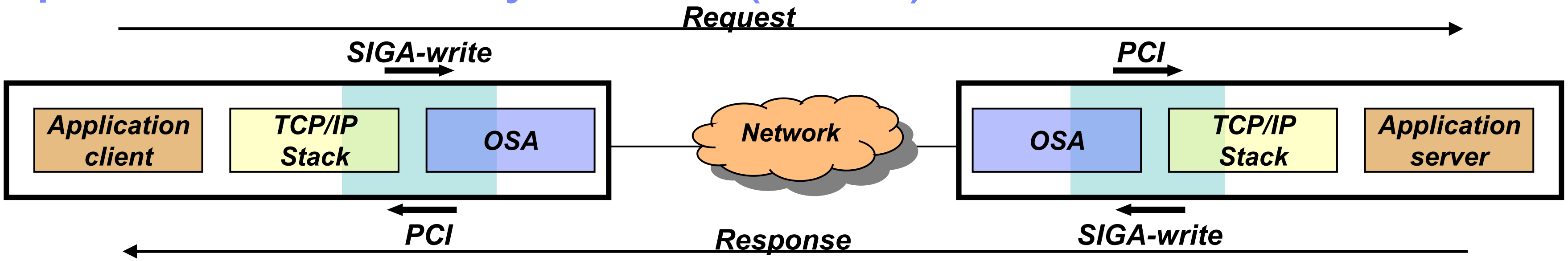
- The following OSA Offload Support avoids CPU processing:
  - Checksum Offload (IPCONFIG CHECKSUMOFFLOAD)
    - A checksum of the packet data is calculated and sent with the packet to provide integrity of the data.
  - Segmentation Offload (IPCONFIG SEGMENTATIONOFFLOAD)
    - Also known as Large Send for TCP/IP traffic, Segmentation Offload allows larger amounts of data to be sent by the TCP/IP stack because the OSA provides the segmentation of that data.
  - IPv4 ARP Offload
    - OSA automatically offloads ARP processing.
- Depending upon the traffic, there is variable benefit from these offloads.
  - Traffic with smaller then 32K messages might perform better over OSA than HiperSockets due to these offload capabilities.
  - Dynamic VIPA, VIPA Route (VIPADYNAMIC VIPAROUTE) option is recommended to send traffic over OSA (or HiperSockets) rather than the default XCF links.

Shared OSA



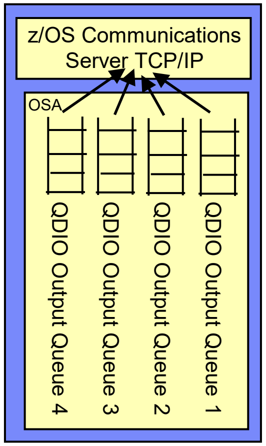
- INTERFACE NOISOLATE (the default)
- All IP addresses in HOME list are added to the OSA Address Table (OAT).
- When a packet is sent from one of the systems sharing the OSA and the destination is an IP address in the OAT, the packet is sent directly to the destination without going out onto the LAN.
- This can be a performance improvement when sending data between hosts that share an OSA port.

Optimized Latency Mode (OLM)



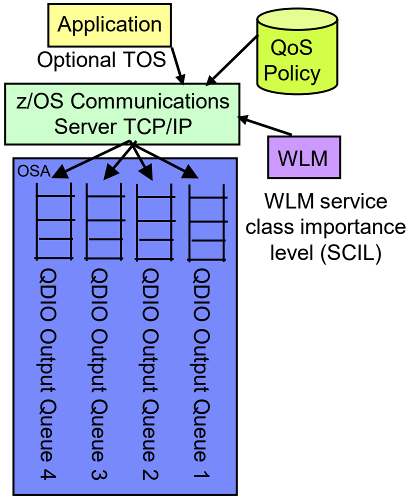
- INTERFACE IPAQENET OLM
- Define OLM when,
  - Latency is the most critical factor (ie. More important than CPU overhead).
  - Traffic is not streaming bulk data (ie. FTP).
- GLOBALCONFIG WLMPRIORITYQ or QoS configuration statement SETSUBNETPRIOTOSMASK may be necessary to benefit from OLM.
  - OLM will not change traffic patterns if all the traffic is being sent to the fourth OSA queue.
- Inbound
  - OSA signals the host if data is “on its way” (“Early Interrupt”).
  - Host looks more frequently for data from OSA.
- Outbound
  - OSA does not wait for SIGA to look for outbound data (“SIGA reduction”).
- Restrictions:
  - When OLM is defined QDIO Accelerator will not accelerate the traffic.
  - When OLM is specified INBPERF is automatically set to DYNAMIC.
  - Interfaces sharing an OSA port using OLM is limited to four. CHPID sharing is limited to eight.

Inbound Workload Queuing (IWQ)



- INTERFACE IPAQENET INBPERF DYNAMIC WORKLOADQ
- IWQ automatically provides unique input queues for:
  - Sysplex Distributor traffic
  - Bulk data (streaming) traffic
  - Enterprise Extender (EE) traffic
  - Default (Interactive)
- Prevents inbound and outbound out of order packets, and the overhead that goes with it.
- IWQ can improve performance but there is an overhead associated with it.
  - Testing should be done to determine if the overhead required is worth the performance improvement provided.

Outbound Priority Routing Queues



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

Default mapping of ToS values to priorities:

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

WLM Service Class

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

- GLOBALCONFIG WLMPRIORITYQ
- WLM IO Priority Enhancement
  - When a packet with a ToS or traffic class value of 0 is sent over a QDIO OSA port, TCP/IP sets the QDIO write priority of the packet based on the priority value provided by the WLM service class.

Keep Alive

Keep Alive, Inactive, and Scan Interval

- Keep Alive
  - TCPCONFIG INTERVAL (default 120 minutes)
    - TCPCONFIG KEEPALIVEPROBEINTERVAL (default 75 seconds) is the time in between probes that are being sent.
    - TCPCONFIG KEEPALIVEPROBES (default 10) is the number of probes sent prior to the connection take down.
  - When keep alive processing is occurring probes are sent. If a response is received for any probe, then the session is still working so it will not be dropped due to keep alive processing.
  - Keep Alive processing may be disabled by defining TCPCONFIG INTERVAL 0
  - The default Keep Alive Interval is 2 hours. So by default, after 2 hours 10 probes are sent, one every 75 seconds. If no response to any of the probes is received the session will be dropped 75 seconds after the last probe.
- TN3270 profile Inactive
  - INACTIVE (default 0) applies to the SNA traffic over that TCP/IP connection. If there is no application traffic (SNA VTAM traffic) over the connection in the Inactive time interval then the connection will be dropped. Setting INACTIVE to 0 (zero) disables this timer so the session will never be dropped due to SNA VTAM traffic inactivity.
- TN3270 profile SCANINTERVAL and TIMEMARK
  - Every SCANINTERVAL (default 1800 seconds) time the connection is checked for inbound traffic from the client. The default SCANINTERVAL is 30 minutes.
  - If there has been no inbound traffic from the client in the TIMEMARK time (default 10800 seconds) then a TIMEMARK probe is sent. TIMEMARK defaults to 3 hours.
  - At the next SCANINTERVAL time, if no response or other inbound traffic from the client has been received, then the session is dropped.
- As an example, using the defaults, after 30 minutes ScanInterval checks and there has been inbound traffic in the past 3 hours. After 30 more minutes ScanInterval checks and there has been no inbound traffic in the past 3 hours so a TimeMark probe is sent. After 30 more minutes ScanInterval checks and there has still been no inbound traffic in the past 3 hours (including no response to the probe) so the session is dropped.
- If you are trying to eliminate the server from a problem, you could define INACTIVE=0, TCPCONFIG INTERVAL=0, and either use the defaults for SCANINTERVAL and TIMEMARK or increase them to the max 99999999.
- On the client side:
  - Keep Alive may be enabled. The Time Out interval defaults to 180 seconds (3 minutes) but may be increased to 99999 seconds.
- If a problem is occurring due to a firewall in the path throwing away packets, of course none of these settings will help.

Window Size

Window Size

- Send and Receive TCP Buffer Sizes
  - TCPMAXRCVBUFRSIZE (default 256 KB)
  - TCPMAXSENDBUFRSIZE (default 256 KB)
  - TCPRCVBUFRSIZE (default 65536)
  - TCPSENDBFRSIZE (default 65536)
- The send buffer size is the amount of application send data that TCP/IP can buffer for a connection.
  - The value can increase/decrease due to size information received from the remote system.
  - The value can increase to the maximum defined.
- Outbound Right Sizing allows the buffer to increase to 2 MB.
  - Requires original buffer to be at least 64 KB.
- Define TCP Send and Receive Buffers to at least 64K.

Encryption

AT-TLS Heap Pool

- Reduce AT-TLS overhead by adding this to the TCP/IP proc:

```
//CEEOPTS  DD  *  HEAPPOOLS64(ON)
```

Cryptographic Hardware

- There are two stages to TLS and IPsec connections.
  - Stage 1 uses public/private key pairs using asymmetric encryption.
    - TLS authenticates the server (and optionally the client) and negotiates the encryption used to send the application data.
    - IPsec authenticates both sides and negotiates the encryption used for two tunnels, the first between the IKE Daemons, and the second to send the application data.
    - Stage 1 will be offloaded to the Cryptographic Cards if they are defined to z/OS and the Integrated Cryptographic Service Facility is defined. Otherwise, it may be offloaded to the CPACF (CP Assist for Cryptographic Functions), depending upon the algorithms used.
  - Stage 2 uses symmetric encryption to send the application data.
    - There is a lower overhead associated with symmetric encryption.
    - TLS stage 2 may be offloaded to the CPACF, depending upon the algorithms used.
    - IPsec may be offloaded to zIIP (GLOBALCONFIG ZIIP IPSECURITY), otherwise it may be offloaded to CPACF, depending upon the algorithms used.

Cryptographic Algorithms

- Some cryptographic algorithms have a larger overhead than others.
- Newer Z hardware provides lower overhead for some algorithms.
  - Perfect Forward Secrecy (PFS) protects against session keys being compromised in the event that the private key is compromised.
  - Ephemeral Elliptic Curve Diffie-Hellamn (ECDH) provides PFS and a new ECDH key pair must be created for each full handshake, which are both costly functions. z15+ provides Elliptic-Curve Cryptography (ECC) support in CPACF, which is a large relief for this overhead compared to z14 where this is all done in software.
- Test with different algorithms to determine which algorithm provides the necessary protection that is needed with an overhead that can be tolerated.

More Information

Web Pages

- The z/OS Communications Server Performance Index web site that provides benchmark data:
  - https://www.ibm.com/support/pages/node/317829
- URLs for Publications
  - http://www.ibm.com/systems/z/os/zos/bkserv/index.html
  - http://www.redbooks.ibm.com
- PKI Services web site:
  - http://www.ibm.com/servers/eserver/zseries/zos/pki
- PKI Services Red Book:
  - http://www.redbooks.ibm.com/abstracts/sg246968.html
- IBM Washington Systems Center Technical Sales Support
  - http://www.ibm.com/support/techdocs/
- Request for Comment (RFC)
  - http://www.rfc-editor.org/rfcsearch.html
  - http://www.rfc-editor.org/

End of Presentation

z/OS Communications Server TCP/IP Performance