SHARE

Session 8322

Anaheim

Treasure Hunt:

Buried Gems in z/OS Communications Server V1R10, V1R11, VR12

Gwen Dente (gdente@us.ibm.com)

Wednesday, March 2, 4:30 PM - 6:00 PM

Room 212A
(Convention Center)

IBM Advanced Technical Skills

Abstract

- z/OS Communications Server V1R12 is now here and you probably have not even caught your breath over the major enhancements in z/OS V1R10 and V1R11.
- Did you know there were a lot of hidden gems in these releases that could make your life easier? This session presents practical examples of a treasure chest of diamonds, rubies, sapphires, and pearls for your Communications Server z/OS implementation. This knowledge should help you feel somewhat caught up for that next move to z/OS V1R10, V1R11, or V1R12.
- Disclaimer: There are no IPv6, Enterprise Extender, or Sysplex/VIPA topics in this presentation, as these two subjects have been on the radar screen for quite a while now and are covered extensively in many other presentations. Therefore, you will find some of these items documented in the appendices of this presentation.
- Therefore, this session tends to focus on subjects that have been "under the radar" and that have escaped many an implementer's attention.

Acknowledgments: Many visuals are modified from presentations produced by Alfred Christensen, Mike Fox, Dave Herr, and Tom McSweeney of z/OS Communications Server Development.

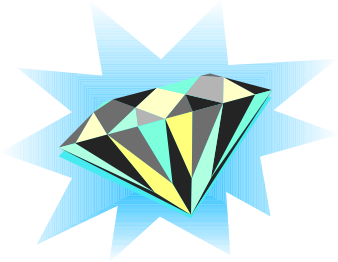
Why Make Plans to Move to z/OS V1R11 or V1R12 Now?

- **z/OS V1R10 has "end-of-service" date of September 30, 2011! Time to move to z/OS V1R11 or V1R12!**
- **The tables below indicate the end-of-service dates for z/OS releases.**
  - **An asterisk (*) indicates projected date. Actual end of marketing or end of service date has not been announced yet. See: http://www-03.ibm.com/servers/eserver/zseries/zos/support/zos_eos_dates.html**

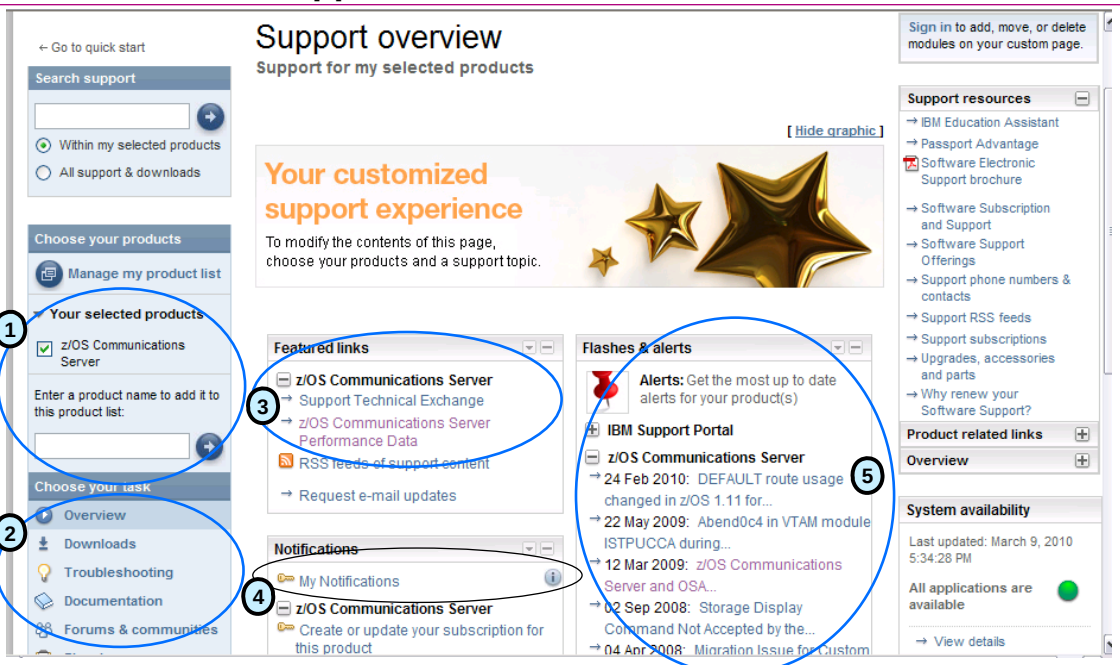
Program Number	Version Release Modification	Announced	Available	Withdrawn from Marketing	Service Discontinued
5694-A01	1.12.0	2010/07/22	2010/09/24	2011/09*	
5694-A01	1.11.0	2009/08/18	2009/09/25	2010/09/25	2012/09*
5694-A01	1.10.0	2008/08/05	2008/09/26	2009/10/26	2011/09 ****
5694-A01	1.09.0	2007/08/08	2007/09/28	2008/10/27	2010/09/30 ***
5694-A01	1.08.0	2006/08/08	2006/09/29	2007/10/22	2009/09/30**
5694-A01	1.07.0	2005/07/27	2005/09/30	2006/10/23	2008/09/30
5694-A01	1.06.0	2005/08/10	2004/09/24	2005/10/24	2007/09/30
5694-A01	1.05.0	2004/02/10	2004/03/26	2004/09/09	2007/03/31
5694-A01	1.04.0	2002/08/13	2002/09/27	2004/09/09	2007/03/31
5694-A01	1.03.0	2002/02/19	2002/03/29	2002/09/12	2005/03/31
5694-A01	1.02.0	2001/09/11	2001/10/26	2002/03/14	2004/10/31
5694-A01	1.01.0	2000/10/03	2001/03/30	2001/10/11 or 2002/06/25	2004/03/31

* indicates a Projected Date. **** Support for z/OS V1R10 is planned to be withdrawn on September 30, 2011. ** Support for z/OS V1.8 was withdrawn on September 30, 2009 and ***Support for z/OS V1.9 was withdrawn on September 30, 2010. The IBM Lifecycle Extension for z/OS V1.8 (5636-A01), the IBM Lifecycle Extension for z/OS V1.9 (5646-A01), and the IBM Lifecycle Extensioin for z/OS V1.10 (5656-A01) provide fee-based corrective service (a fix, bypass, or restriction to a problem) for up to two years beyond the withdrawal of service dates listed above.

Gems: Migration Planning Improvements



IBM Support Portal: Overview & Product List



http://www.ibm.com/support/entry/portal
https://www.ibm.com/support/mynotifications
https://www-912.ibm.com/x_dir/xfeedback.nsf/feedback?OpenForm

NEW and ready for you! The new IBM Support Portal provides complete, customized support for all IBM software, hardware, and services.

Start using the IBM Support Portal today!

http://www.ibm.com/support/entry/portal

Manage your My notifications subscriptions, or send questions and comments.

Subscribe or Unsubscribe - https://www.ibm.com/support/mynotifications

Feedback - https://www-912.ibm.com/x_dir/xfeedback.nsf/feedback?OpenForm

To ensure proper delivery please add mynotify@stg.events.ihost.com to your address book.

Notice how you can customize your IBM Support Panel View.

(1) The panel here is customized to view only z/OS Communications Server.

(2) The z/OS Communications Server selection provides you with the opportuity to select an Overview, to select the Downloads for z/OS Communications Server (like Configuration Assistant for building policies), to select Troubleshooting, Documentation, and even subscribe to Forums and communities.

(3) The featured links on the z/OS Communications Server page can connect you to the Performance data that will help you determine the contrasting performance of SSL/TLS and no SSL/TLS, the contrasting performance of TCP/IP applications from release to release.

(4) "My Notifications" are particularly important here if you want to be advised of any additions to the z/OS Communications Server site by email.

This is how you can learn about new APARs, technical notes, and so on.

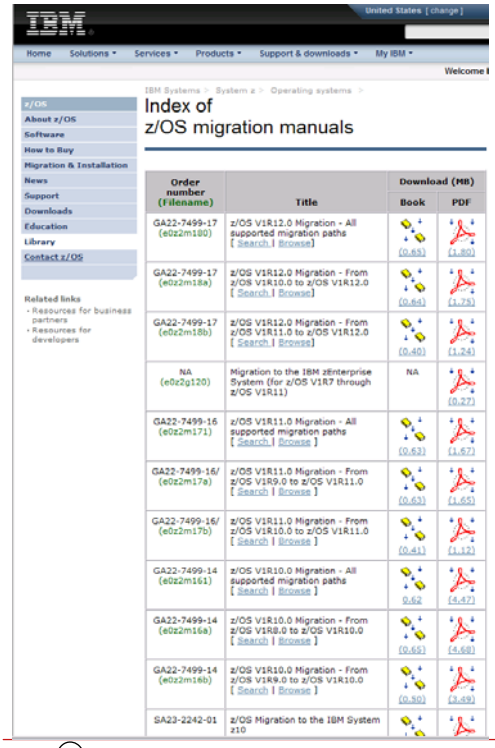
(5) For example, through a subscription to "My Notifications" an email would have advised you in February 2010 about the change in he DEFAULT route usage.

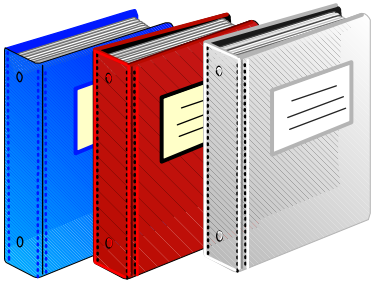
First Things First: 4 Manuals to Get You Started with Migration

z/OS Comm Server information in system books

- **z/OS Migration**
  - Lists Comm Server function that requires you to take action to migrate to V1R12
  - This information is not provided in this format in the Communications Server library
- **z/OS Summary of Message and Interface Changes**
  - Lists all new and changed Comm Server commands, parameters, socket API changes, FTP and Telnet changes, etc.
  - This information is not provided in this format in the Communications Server library
- **z/OS Introduction and Release Guide**
  - Presents high-level function descriptions with pointers to the detailed descriptions in New Function Summary
- **z/OS Communications Server New Function Summary**
  - Detailed descriptions of new CS functions

Index of Migration Manuals (V1R10-V1R12)



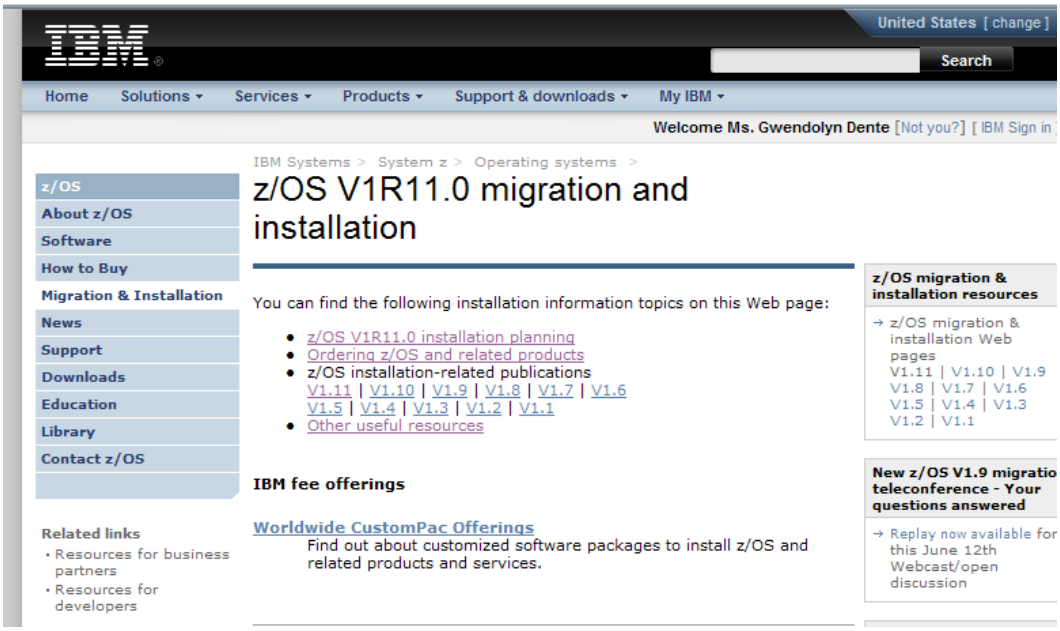


http://www-03.ibm.com/systems/z/os/zos/bkserv/zos_migration_manuals.html

1. Find this list at: http://www-03.ibm.com/systems/z/os/zos/bkserv/zos_migration_manuals.html

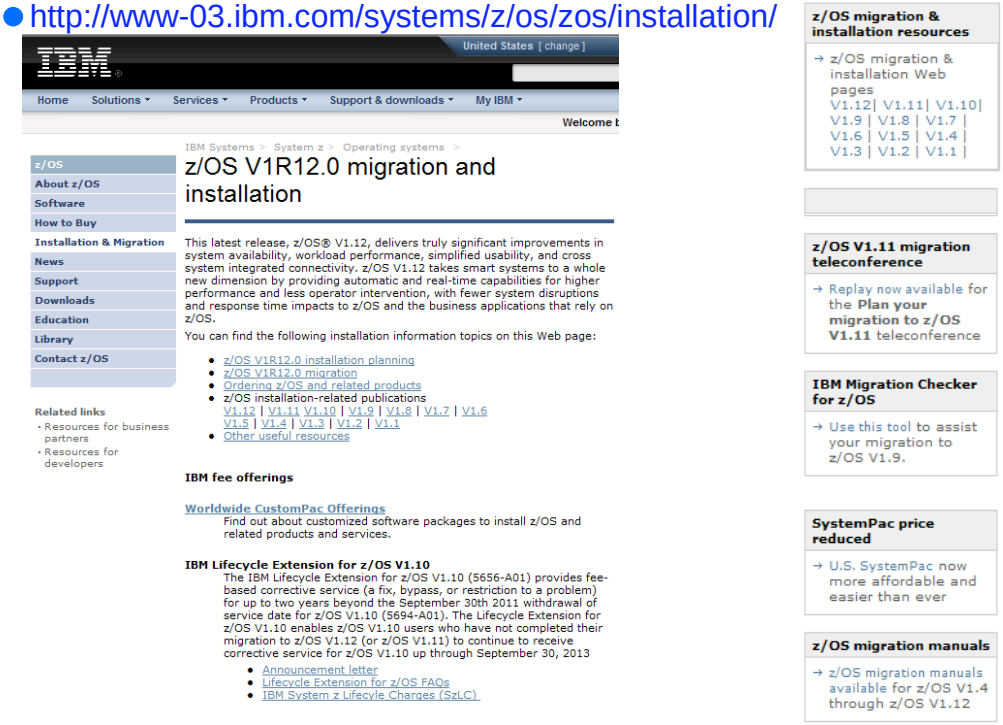
Resources for Migration to z/OS V1R11

- http://www-03.ibm.com/systems/z/os/zos/installation/



Resources for Migration to z/OS V1R12

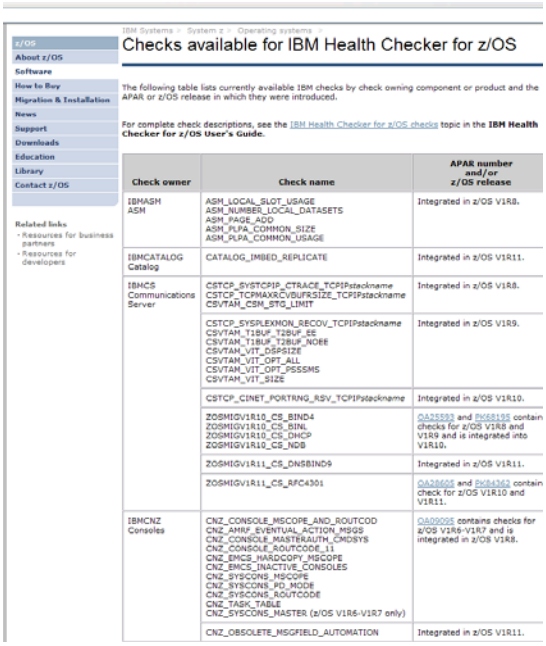
- http://www-03.ibm.com/systems/z/os/zos/installation/



Health Checks Available for z/OS Communications Server

http://www-03.ibm.com/systems/z/os/zos/hchecker/check_table.html

IBM Health Checker for z/OS User's Guide (GA22-7994)



IBM Systems > System z > Operating systems >

z/OS
About z/OS
Software
How to Buy
Migration & Installation
News
Support
Downloads
Education
Library
Contact z/OS

Related links
- Resources for business partners
- Resources for developers

Checks available for IBM Health Checker for z/OS

The following table lists currently available IBM checks by check owning component or product and the APAR or z/OS release in which they were introduced.

For complete check descriptions, see the IBM Health Checker for z/OS checks topic in the **IBM Health Checker for z/OS User's Guide**.

Check owner	Check name	APAR number and/or z/OS release
IBMASH ASM	ASM_LOCAL_SLOT_USAGE ASM_NUMBER_LOCAL_DATASETS ASM_PAGE_ADD ASM_PLPA_COMMON_SIZE ASM_PLPA_COMMON_USAGE	Integrated in z/OS V1R8.
IBMCATALOG Catalog	CATALOG_IMBED_REPLICATE	Integrated in z/OS V1R11.
IBMCS Communications Server	CSTCP_SYSTCPIP_CTRACE_TCPIPstackname CSTCP_TCPMAXRCVBUFRSIZE_TCPIPstackname CSVTAM_CSM_STG_LIMIT	Integrated in z/OS V1R8.
	CSTCP_SYSPLEXMON_RECOV_TCPIPstackname CSVTAM_T1BUF_T2BUF_EE CSVTAM_T1BUF_T2BUF_NOEE CSVTAM_VIT_DSPSIZE CSVTAM_VIT_OPT_ALL CSVTAM_VIT_OPT_PSSSMS CSVTAM_VIT_SIZE	Integrated in z/OS V1R9.
	CSTCP_CINET_PORTRNG_RSV_TCPIPstackname	Integrated in z/OS V1R10.
	ZOSMIGV1R10_CS_BIND4 ZOSMIGV1R10_CS_BINL ZOSMIGV1R10_CS_DHCP ZOSMIGV1R10_CS_NDB	OA25593 and PK68195 contain checks for z/OS V1R8 and V1R9 and is integrated into V1R10.
	ZOSMIGV1R11_CS_DNSBIND9	Integrated in z/OS V1R11.
	ZOSMIGV1R11_CS_RFC4301	OA28605 and PK84362 contain check for z/OS V1R10 and V1R11.
IBMCNZ Consoles	CNZ_CONSOLE_MSCOPE_AND_ROUTCOD CNZ_AMRF_EVENTUAL_ACTION_MSGS CNZ_CONSOLE_MASTERAUTH_CMDSYS CNZ_CONSOLE_ROUTCODE_11 CNZ_EMCS_HARDCOPY_MSCOPE CNZ_EMCS_INACTIVE_CONSOLES CNZ_SYSCONS_MSCOPE CNZ_SYSCONS_PD_MODE CNZ_SYSCONS_ROUTCODE CNZ_TASK_TABLE CNZ_SYSCONS_MASTER (z/OS V1R6-V1R7 only)	OA09056 contains checks for z/OS V1R6-V1R7 and is integrated in z/OS V1R8.
	CNZ_OBSOLETE_MSGFIELD_AUTOMATION	Integrated in z/OS V1R11.

1. You will probably want to download the IBM Health Checker for z/OS User's Guide to investigate how to implement Health Checker and to understand the various types of health checks that are available to you, including those in IBM Communications Server.
2. The User's Guide points you to a web page that is kept updated for all currently available health checks:
   1. http://www-03.ibm.com/systems/z/os/zos/hchecker/check_table.html
3. The web page provides you the name of the RFC4301 health check that you will want your z/OS Systems Programmer to implement for you.

Selected Health Checks to Verify for Migration

Resources for Migration to z/OS V1R10: Health Checker

- **z/OS CS "Best Practice" Health Check**
  - Check that the ***BPXPRMxx INADDRANYPORT and INADDRANYCOUNT*** specifications match correct ***TCP/IP PORT/PORTRANGE*** definitions
  - These ports must be reserved to OMVS - if not, an abend EC6 may occur when Common INET tries to use one of them
- **z/OS CS now implements migration checks within the z/OS Healthchecker "Best Practice" infrastructure**
  - A check to determine if Boot Information Negotiation Layer (BINL) server function is in use on the system.
  - A check to determine if Berkeley Internet Name Domain 4.9.3 (BIND 4.9.3) DNS server function is in use on the system.
  - A check to determine if Dynamic Host Configuration Protocol (DHCP) server function is in use on the system.
  - A check to determine if Network Database (NDB) server function is in use on the system.
- **The migration health checks that are delivered as part of z/OS V1R10 CS, were also rolled back to z/OS V1R8 & V1R9**

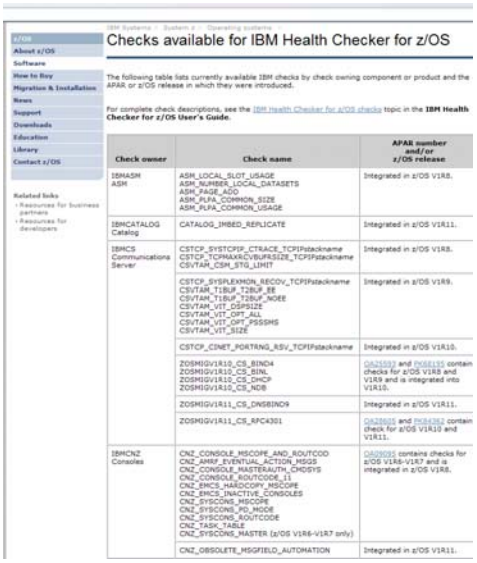
```
HZS0001I CHECK(IBMCS,ZOSMIGV1R10_CS_BIND4):
ISTM004E BIND 4.9.3 DNS server function is in use on this
system during this IP.
```


1. The V1R10 Migration Health Checks:
   1. Objective is to provide programmatic migration checks that can give you an early warning if you are using functions that will be significantly changed or removed in future releases
2. Traffic Regulation Policies
   1. z/OS V1.9 is the last release of z/OS Communications Server which will support the configuration of Traffic Regulation (TR) policy as part of the Quality of Service discipline. The TR configuration function remains supported, but IBM recommends that you implement it as part of the Intrusion Detection Services (IDS) policy configuration made available in z/OS V1.8. This change is only for the TR policy configuration. The TR policy functions themselves remain unaffected. For more information, please refer to z/OS V1.8 Communications Server's IP Configuration Guide, chapter 16, "Intrusion Detection Services", and IP Configuration Reference, chapter 23, "Intrusion Detection Services policy".
   2. z/OS CS Network Data Base (NDB) server removal
      1. In a future release of z/OS, the Network Database (NDB) function will be removed from the z/OS Communications Server component. Customers who currently use or plan to use the NDB function should investigate the distributed data facility (DDF) provided by z/OS DB2, and the DB2 Run-Time Client. DDF allows client applications running in an environment that supports DRDA to access data at DB2 servers.
   3. z/OS CS Dynamic Host Configuration Protocol (DHCP) server removal
      1. In a future release of z/OS, the Dynamic Host Configuration Protocol (DHCP) server function will be removed from the z/OS Communications Server component. Customers who currently use or plan to use the z/OS DHCP server should investigate using a DHCP server on Linux for System z.
   4. z/OS CS Boot Information Negotiation Layer (BINL) removal
      1. In a future release of z/OS, the Boot Information Negotiation Layer (BINL) function will be removed from the z/OS Communications Server component. Customers using this function should investigate the use of IBM Tivoli Provisioning Manager for OS Deployment for network-based operating system installation services.
   5. NOTE: All statements regarding IBM's plans, directions, and intent are subject to change or withdrawal without notice.

Health Checks Available for z/OS Communications Server V1R10 and V1R11

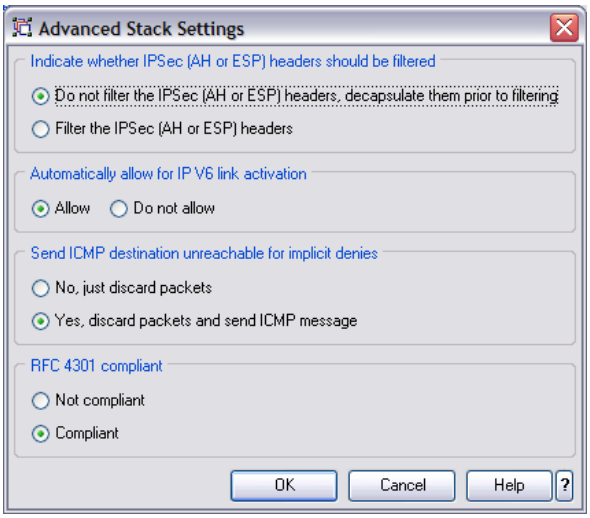
http://www-03.ibm.com/systems/z/os/zos/hchecker/check_table.html

IBM Health Checker for z/OS User's Guide (GA22-7994)



1. You will probably want to download the IBM Health Checker for z/OS User's Guide to investigate
   1. how to implement Health Checker and
   2. to understand the various types of health checks that are available to you, including those in IBM Communications Server.
2. The User's Guide points you to a web page that is kept updated for all currently available health checks:
   1. http://www-03.ibm.com/systems/z/os/zos/hchecker/check_table.html
3. The web page provides you the name of the RFC4301 health check that you will want your z/OS Systems Programmer to implement for you.

RFC4301 Compliance & GUI Migration Assistance (V1R10, 11)



- **Routed Traffic Rules must not contain:**
  - Port Numbers
  - ICMP(v6) Code Types
  - OSPF Types
- **For Migration:**
  - Use the z/OS Migration Manual
  - Use the GUI
  - Use the z/OS V1R11 HealthChecker (available at V1R10)

- When given a choice, always try to configure IPSec with RFC4301 compliance. After V1R11 you **must** configure RFC4301 compliance and will not want to be forced to reconfigure your policies!

1. RFC4301 "Security Architecture for the Internet Protocol" specifies the base architecture for IPSec compliant systems
   1. – Includes restrictions on the routing of fragmented packets
   2. .. In z/OS V1R10 and V1R11, RFC4301 compliance enforcement is an optional setting in the z/OS IPSec policy
   3. – Changing an IPSec policy from non-compliant to compliant might require minor changes to IP filters for IP traffic that is routed through z/OS
2. RFC4301 - "Security Architecture for the Internet Protocol"
   1. Prior to RFC 4301 support, IPsec filters all routed IP fragments using a policy of first possible filter match (RFC4301 compliance=no)
      1. port, type, or code specifications are allowed on routed traffic rules
      2. filter all IP fragments by first possible filter match - except: non-initial IP fragments only match rules covering All ports, types, or codes
   2. RFC 4301 introduces rules and restrictions to ensure proper classification of fragments (RFC4301 compliance=yes)
      1. Use and enforce the RFC 4301 restrictions on IP filter rules: no port, type, or code specifications on routed traffic rules
      2. RFC4301Compliance parameter on the IpFilterPolicy statement
3. To be RFC4301-compliant, you should not filter on ports/type/code for routed traffic
   1. You can have the GUI enforce this compliance by discarding non-compliant rules and issuing a report. Or you can just issue a GUI health check warning
   2. If you use z/OS V1R10 or z/OS V1R11 Configuration Assistant you can change the RFC 4301 Compliance setting to 'Compliant'. This generates an RFC 4301 Compliance Report, and marks rules **incomplete**. This is similar to importing a V1R10 or V1R11 backing store into V1R12 Configuration Assistant. (At V1R12, you find a 'Fix Incomplete Rules' button)
      1. If you do not change the compliance setting to "Compliant," you can use the 'Health Check' button in the IPSec Perspective panel. This generates an RFC 4301 Compliance report and identifies noncompliant rules even though the RFC 4301 Compliance setting is 'Not compliant'
   3. A z/OS migration health check in z/OS V1R11 will determine if you have such filter rules:
      1. – ISTM010E IPSec filter rules that violate RFC4301 compliance are in use on this system during this IPL
4. This restriction can be temporarily suspended up through z/OS V1R11 until you update your policy to comply with the restriction. As an interim measure, you can configure the stack as Not compliant as indicated by one of the radio buttons in this GUI panel.
5. You may choose to relax the restriction until you have updated your configuration. If you choose to relax the restriction, you should be aware that the vulnerabilities cited in RFC 4301 concerning routed traffic and fragmented packets will apply to you.
6. At V1R12, you are no longer given a choice to be non-RFC4301-compliant.

Resources for Migration to z/OS V1R11: Health Checker

```
F HEALTHCK,DISPLAY,CHECKS
HZS0200I 10.25.57 CHECK SUMMARY
CHECK OWNER CHECK NAME                          STATE STATUS
IBMCS       CSTCP_CINET_PORTRNG_RSV_TCPCS1      AE    SUCCESSFUL
IBMCS       CSTCP_SYSPLEXMON_RECOV_TCPCS1       AE    EXCEPTION-LOW
IBMCS       CSTCP_TCPMAXRCVBUFRSIZE_TCPCS1      AE    SUCCESSFUL
IBMCS       CSTCP_SYSTCPIP_CTRACE_TCPCS1        AE    EXCEPTION-LOW
IBMCS       CSVTAM_T1BUF_T2BUF_NOEE             AE    SUCCESSFUL
IBMCS       CSVTAM_T1BUF_T2BUF_EE               AD    ENV N/A
IBMCS       CSVTAM_VIT_OPT_ALL                  AE    EXCEPTION-LOW
IBMCS       CSVTAM_VIT_DSPSIZE                  AE    EXCEPTION-LOW
IBMCS       CSVTAM_VIT_OPT_PSSSMS               AE    SUCCESSFUL
IBMCS       CSVTAM_VIT_SIZE                     AE    EXCEPTION-LOW
IBMCS       CSVTAM_CSM_STG_LIMIT                AE    SUCCESSFUL
IBMUSS      USS_MAXSOCKETS_MAXFILEPROC          AD    UNEXP ERROR
IBMUSS      USS_AUTOMOUNT_DELAY                 AD    ENV N/A
IBMUSS      USS_FILESYS_CONFIG                  AE    EXCEPTION-MED
IBMIXGLOGR  IXGLOGR_ENTRYTHRESHOLD              AE    SUCCESSFUL
```

- ***No check for TCPRCVBUFRSIZE --but verify anyway that it is at least 64K so that you can take advantage of "Dynamic Right Sizing" in z/OS V1R11***

1. To setup Health Checker to run, you must:
   1. 1 Allocate the HZSPDATA data set to save check data between restarts
   2. 2 Set up the HZSPRINT utility
   3. 3 Define log streams
2. •If you want to maintain an historical record of your check output
   1. 4 Create security definitions
   2. •Give the Health Checker proc update access to the HZSPDATA data set
   3. •Give the Health Checker proc read access to the HZSPRMxx parmlib members
   4. •Give the Health Checker proc read access to each Health Checker logstream
   5. •Authorize HZSPRINT users to QUERY and MESSAGES services
   6. •Authorize SDSF support for Health Checker message output
   7. 5 Create multilevel security definitions, if necessary
   8. 6 Create HZSPRMxx from the HZSPRM00 parmlib member
3. •If you want to make permanent changes to check values & parameters
4. •If you want to deactivate a check
   1. 7 Start the IBM Health Checker for z/OS proc
5. ..Step-by-Step details for setting up Health Checker can be found in .. IBM Health Checker for z/OS User's Guide and at http://www-03.ibm.com/systems/z/os/zos/hchecker/check_table.html
6. Setting the TCP Buffer size to a minimum of 64K is important if you want to take advantage of "DYNAMIC RIGHTSIZING" in z/OS V1R11.
   1. Streaming workload over large bandwidth and high latency networks (such as satellite links) is in general constrained by the TCP window size. The problem is that it takes time to send data over such a network. At any given point in time data filling the full window size is 'in-transit' and cannot be acknowledged until it starts arriving at the receiver side. The sender can send up to the window size and then must wait for an ACK to advance the window size before the next chunk can be sent.
   2. If it were possible to dynamically adjust the window size to what it takes to fill the network in-between the sender and the receiver, higher throughput might be achieved.
   3. This support will, on the receiver side, dynamically adjust the window size upward (beyond 180K if so needed) in an attempt to 'fill' the pipe between the sender and the receiver. The aim is that as soon as the sender has sent the end of its window, the sender receives an ACK from the receiver. That ACK allows the sender to advance the window and send another chunk onto the network.
   4. The dynamic right sizing (DRS) algorithm is based on a paper that was published by Los Alamos National Laboratory. The goal of DRS is to keep the pipe full and prevent sender from being constrained by the advertised window. The window size may grow as high as 2 Mbytes. The TCP/IP stack will disable the function if the application doesn't keep up. A netstat all report wil show the DRS-adjusted receive buffer size.
   5. NOTE: Be sure to check the size of your TCPRCVBUFRSIZE and adjust to 64K or higher; otherwise the Dynamic Right Sizing function in V1R11 may not work for you; the receive buffer must be equal to or larger than 64K. There is no healthchecker available to verify the size of the TCPRCVBUFRSIZE ... there is only one for TCPMAXRCVBUFRSIZE.

New Health Check in z/OS CS V1R12: Routing Table too Large

Problem: High Utilization from Routing Changes

Large routing table (2000 or more routes) in a TCP/IP stack can potentially cause high processor utilization for the route changes

Most customer sites typically use 50-500 unique routes

Noticeable performance degradation in OMPROUTE, OMVS, and TCP/IP stack as the number of routes increase and worsens with tracing enabled

The time to process route updates might exceed OMPROUTE's Dead Router Interval for OSPF routes resulting in lost adjacencies with neighbors and network connectivity problems

Solution: Monitor the number of indirect routes

New counters monitor the number of indirect routes in IPv4 and IPv6 routing tables for a TCP/IP stack:

- Current number
  - Total number of indirect routes after adds and deletes
- High interval number
  - Peak number of indirect routes during a time interval

1. A routing table that is considered to be excessive (2000 routes or more) can cause inefficiency in network design and less than optimal performance for OMPROUTE and TCP/IP. Most z/OS sites appear to have 50-500 unique routes. IBM service frequently tells customers with more than 2000 routes to reduce the number of routes after determining that performance degradations in OMPROUTE and TCP/IP were caused by the excessive number of routes. The overall performance degrades further with tracing enabled.
2. There have been a small number of customers over the years who have attempted to configure many thousands of routes (from both dynamic and static routing protocols) on z/OS when they only needed 100 or so. Most of the time, having many thousands of routes will not cause a problem. However, if all of the routes ever need to be deleted or added at the same time, then high processor consumption might be seen in the TCP/IP stack or in OMVS. Many thousands of routing updates have to be processed to make the routing changes.
3. Also, because the OSPF routing protocol in OMPROUTE uses short-interval timers, the time to process the many thousands of routing updates might exceed the OSPF dead router intervals. This results in OSPF adjacency losses with neighbors and contributes to network connectivity problems.
4. These counters are used by IBMHC for the health check monitoring and for input into the informational and warning messages. The current number is incremented and decremented at times of the route table updates. The high interval number is set to the peak number of indirect routes during a time interval and is reset to the current number for the next time interval.
5. IBM Health Checker will perform checks at these times:
   1. One-time check (30 minutes after TCP/IP initialization)
6. For initial health state after routing table updates by TCP/IP and OMPROUTE
   1. Not done if IBM Health Checker started 30 minutes after TCP/IP initialization or if interval check is less than 30 minutes
7. Interval checks (defaults to 168 hours or weekly)
8. Immediate checks (at any time) when:
   1. A counter has exceeded the maximum threshold (default 2000)
   2. A maximum threshold value has been dynamically modified by an operator

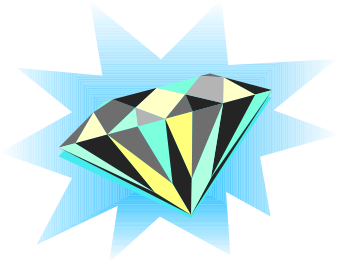
Resources for Migration to z/OS V1R12: Health Checker

SDSF option CK (S.CK) display

```
Display  Filter  View  Print  Options  Search  Help
------------------------------------------------------------------------------
SDSF HEALTH CHECKER DISPLAY  MVS052                  LINE 21-56 (135)
COMMAND INPUT ===>                                  SCROLL ===> CSR
NP   NAME                                 State              Status
     CSTCP_IPMAXRT4_TCPCS                 ACTIVE(ENABLED)    SUCCESSFUL
     CSTCP_IPMAXRT6_TCPCS                 ACTIVE(ENABLED)    SUCCESSFUL
     CSTCP_SYSPLEXMON_RECOV_TCPCS         ACTIVE(ENABLED)    EXCEPTION-LOW
     CSTCP_SYSTCPIP_CTRACE_TCPCS          ACTIVE(ENABLED)    SUCCESSFUL
     CSTCP_TCPMAXRCVBUFRSIZE_TCPCS        ACTIVE(ENABLED)    SUCCESSFUL
     CSVTAM_CSM_STG_LIMIT                 ACTIVE(ENABLED)    SUCCESSFUL
     CSVTAM_T1BUF_T2BUF_EE                ACTIVE(DISABLED)   ENV N/A
     CSVTAM_T1BUF_T2BUF_NOEE              ACTIVE(ENABLED)    SUCCESSFUL
     CSVTAM_VIT_DSPSIZE                   ACTIVE(ENABLED)    EXCEPTION-LOW
     CSVTAM_VIT_OPT_ALL                   ACTIVE(ENABLED)    EXCEPTION-LOW
     CSVTAM_VIT_OPT_PSSSMS                ACTIVE(ENABLED)    SUCCESSFUL
     CSVTAM_VIT_SIZE                      ACTIVE(ENABLED)    EXCEPTION-LOW
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=IFIND     F6=BOOK
 F7=UP        F8=DOWN      F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE
```


1. This is a screen capture of a SDSF Health Checker Display for the list of checks. The display lists the new checks CSTCP_IPMAXRT4_TCPCS and CSTCP_IPMAXRT6_TCPCS. IBMHC has determined that these checks have been successful. The active check states are also displayed here.
2. To setup Health Checker to run, you must:
    1. 1 Allocate the HZSPDATA data set to save check data between restarts
    2. 2 Set up the HZSPRINT utility
    3. 3 Define log streams
3. •If you want to maintain an historical record of your check output
    1. 4 Create security definitions
    2. •Give the Health Checker proc update access to the HZSPDATA data set
    3. •Give the Health Checker proc read access to the HZSPRMxx parmlib members
    4. •Give the Health Checker proc read access to each Health Checker logstream
    5. •Authorize HZSPRINT users to QUERY and MESSAGES services
    6. •Authorize SDSF support for Health Checker message output
    7. 5 Create multilevel security definitions, if necessary
    8. 6 Create HZSPRMxx from the HZSPRM00 parmlib member
4. •If you want to make permanent changes to check values & parameters
5. •If you want to deactivate a check
    1. 7 Start the IBM Health Checker for z/OS proc
6. ..Step-by-Step details for setting up Health Checker can be found in .. IBM Health Checker for z/OS User's Guide and at http://www-03.ibm.com/systems/z/os/zos/hchecker/check_table.html
7. Setting the TCP Buffer size to a minimum of 64K is important if you want to take advantage of "DYNAMIC RIGHTSIZING" in z/OS V1R11.
    1. Streaming workload over large bandwidth and high latency networks (such as satellite links) is in general constrained by the TCP window size. The problem is that it takes time to send data over such a network. At any given point in time data filling the full window size is 'in-transit' and cannot be acknowledged until it starts arriving at the receiver side. The sender can send up to the window size and then must wait for an ACK to advance the window size before the next chunk can be sent.
    2. If it were possible to dynamically adjust the window size to what it takes to fill the network in-between the sender and the receiver, higher throughput might be achieved.
    3. This support will, on the receiver side, dynamically adjust the window size upward (beyond 180K if so needed) in an attempt to 'fill' the pipe between the sender and the receiver. The aim is that as soon as the sender has sent the end of its window, the sender receives an ACK from the receiver. That ACK allows the sender to advance the window and send another chunk onto the network.
    4. The dynamic right sizing (DRS) algorithm is based on a paper that was published by Los Alamos National Laboratory. The goal of DRS is to keep the pipe full and prevent sender from being constrained by the advertised window. The window size may grow as high as 2 Mbytes. The TCP/IP stack will disable the function if the application doesn't keep up. A netstat all report wil show the DRS-adjusted receive buffer size.
    5. NOTE: Be sure to check the size of your TCPRCVBUFRSIZE and adjust to 64K or higher; otherwise the Dynamic Right Sizing function in V1R11 may not work for you; the receive buffer must be equal to or larger than 64K. There is no healthchecker available to verify the size of the TCPRCVBUFRSIZE ... there is only one for TCPMAXRCVBUFRSIZE.

Gems: Convert QDIO from DEVICE/LINK to INTERFACE to Prepare for Future Enhancements, Ensemble, Inbound Workload Queueing, New Displays



Example: INTERFACE for IPv4 (V1R10)

```
INTERFACE NSQDIO411 DEFINE IPAQENET
IPADDR 172.16.11.1/24
PORTNAME NSQDIO1
VLANID 411
MTU 1492
VMAC
SOURCEVIPAINTERFACE LVIPA1
;
; LVIPA1 is the name of a static VIPA
; from a previous LINK statement
;
INTERFACE NSQDIO412 DEFINE IPAQENET
IPADDR 172.16.12.1/24
PORTNAME NSQDIO1
VLANID 412
MTU 1492
VMAC
SOURCEVIPAINTERFACE LVIPA2
```

- **HOME eliminated:**
  - IPADDR
- **Subnet Mask in definition**
  - OMPROUTE conflicts detected
- **MTU in definition**
  - OMPROUTE conflicts detected
- **SOURCEVIPAINTERFACE in definition**

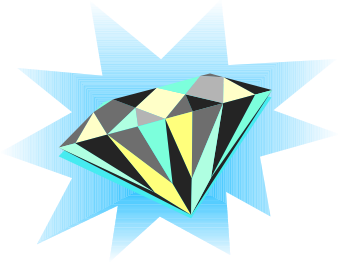
1. If you define the OSA using DEVICE/LINK statements, then the stack will inform OSA to perform ARP processing for all VIPAs in the home list which can result in numerous unnecessary gratuitous ARPs for VIPAs in an interface takeover scenario.
2. However, if you use the IPv4 INTERFACE statement for IPAQENET, you can control this VIPA ARP processing by configuring a subnet mask for the OSA. If you specify a non-0 num_mask_bits value on the IPADDR parameter of the INTERFACE statement, then the stack will inform OSA to only perform ARP processing for a VIPA if the VIPA is configured in the same subnet as the OSA (as defined by the resulting subnet mask).
3. This is an example of multiple VLAN definitions with two INTERFACE statements for IPAQENET. Each statement defines an IPv4 interface associated with the same OSA-Express port NSQDIO1. Each specifies a subnet mask of 24 bits (‘FFFFFF00”x) and defines a unique subnet.
4. The statements contain different VLAN IDs, and each requests that OSA generate a virtual MAC address (and defaults to ROUTEALL). Each statement specifies the link_name of a static VIPA for the source VIPA function.
5. Because so many definitions that used to reside in the HOME list and in BSDROUTINGPARMS are now included in the INTERFACE definition, it is easier to add and delete interfaces dynamically without having to modify the HOME LIST>
   1. If there is any mismatch between OMPROUTE values (MTU and SUBNET MASK), error messages are generated and the values from OMPROUTE are used.
      1. EZZ8163I stack_name MTU value stack_val for interface differs from omproute_procname MTU value omproute_val
      2. EZZ8164I stack_name subnet mask value stack_val for interface differs from omproute_procname subnet mask value omproute_val

Benefits of Migration to INTERFACE Statement

- **Gratuitous ARPs for VIPAs in non-OSA subnet eliminated if Subnet Mask is coded on the IP address**
- **VIRTUALIZATION of the OSA Port into multiples in single Stack with VLAN and VMAC**
- **At V1R11, Optimized Latency Mode on an OSA-E3 takes effect only if coded with**
  - INTERFACE
  - TCPCONFIG TCPRCVBufrsize 64K
- **At V1R12, Inbound Workload Queuing takes effect only if coded with**
  - INTERFACE
- **At V1R12, OSX device is defined only with**
  - INTERFACE
- **At V1R12, "D OSAINFO" command only displays output with the**
  - INTERFACE statement

1. If you define the OSA using DEVICE/LINK statements, then the stack will inform OSA to perform ARP processing for all VIPAs in the home list which can result in numerous unnecessary gratuitous ARPs for VIPAs in an interface takeover scenario.
2. However, if you use the IPv4 INTERFACE statement for IPAQENET, you can control this VIPA ARP processing by configuring a subnet mask for the OSA. If you specify a non-0 num_mask_bits value on the IPADDR parameter of the INTERFACE statement, then the stack will inform OSA to only perform ARP processing for a VIPA if the VIPA is configured in the same subnet as the OSA (as defined by the resulting subnet mask).
3. This is an example of multiple VLAN definitions with two INTERFACE statements for IPAQENET. Each statement defines an IPv4 interface associated with the same OSA-Express port NSQDIO1. Each specifies a subnet mask of 24 bits (‘FFFFFF00”x) and defines a unique subnet.
4. The statements contain different VLAN IDs, and each requests that OSA generate a virtual MAC address (and defaults to ROUTEALL). Each statement specifies the link_name of a static VIPA for the source VIPA function.
5. Because so many definitions that used to reside in the HOME list and in BSDROUTINGPARMS are now included in the INTERFACE definition, it is easier to add and delete interfaces dynamically without having to modify the HOME LIST>
   1. If there is any mismatch between OMPROUTE values (MTU and SUBNET MASK), error messages are generated and the values from OMPROUTE are used.
      1. EZZ8163I stack_name MTU value stack_val for interface differs from omproute_procname MTU value omproute_val
      2. EZZ8164I stack_name subnet mask value stack_val for interface differs from omproute_procname subnet mask value omproute_val

Gems with Interfaces and LANs



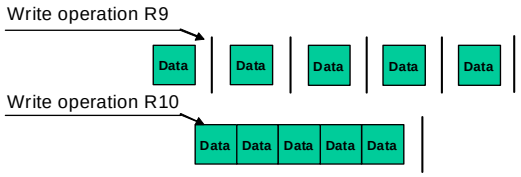
HiperSockets Multiple Write and zIIP Offload (V1R10)

TCP/IP Profile

GLOBALCONFIG IQDMULTIWRITE ZIIP IQDIOMULTIWRITE

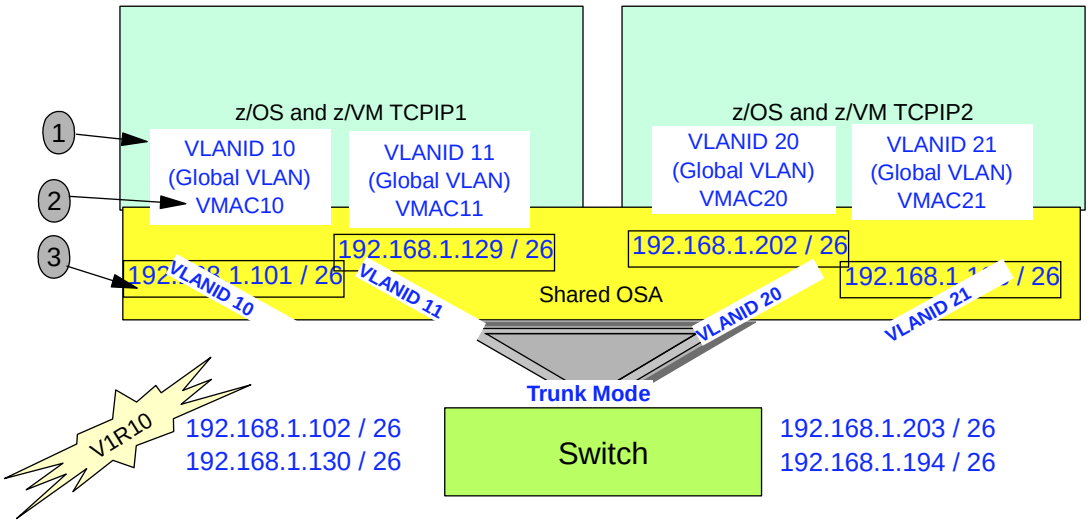
IBM System z10 EC Hipersockets Multiple Write Facility

- Hipersockets can now move multiple output data buffers in one write operation
  - Reduces CPU utilization
  - For large outbound messages
    - Used when message spans Hipersocket frame size



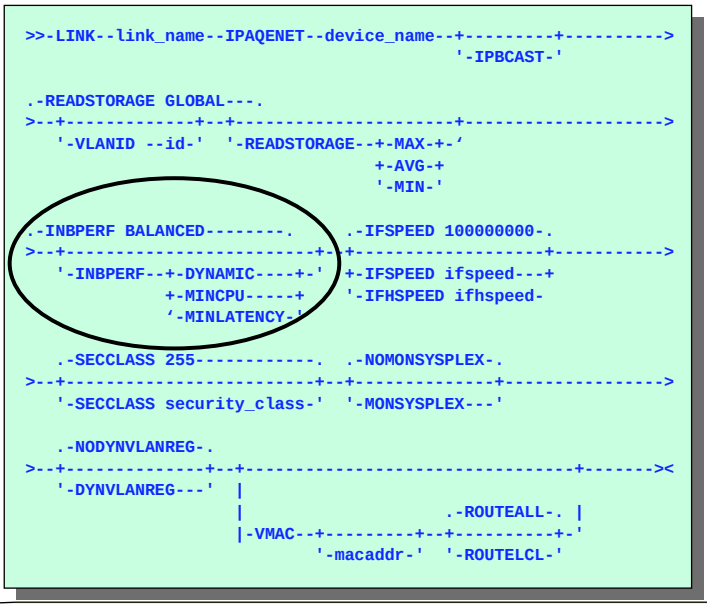
1. The newly announced IBM System z10 includes a new function called Hipersockets Multiple Write. This allows multiple data buffers to be moved from one system image to another across Hipersockets with one operation. This can reduce CPU utilization.
2. With HiperSockets Multiple Write enabled you should see a performance improvement and reduction in CPU utilization for large outbound messages.
   1. .. zIIP assist will also help reduce costs associated with general CPU utilization.
   2. Valid for .. Both HiperSockets Multiple Write and zIIP-Assisted HiperSockets
   3. Multiple Write are disabled by default. Enable them using the new options on the GLOBALCONFIG statement.
   4. .. There are no WLM (enclave) configuration changes required.
   5. .. The PROJECTCPU function in z/OS Workload Manager can be used to project zIIP effectiveness.
3. When enabled, HiperSockets Multiple Write will be used anytime a message spans the Hipersockets frame size, thus requiring multiple output buffers to transfer the message. Therefore, it will only be used for larger outbound messages. Spanning multiple output data buffers can be affected by a number of factors including:
   1. Hipersocket frame size
   2. Application socket send size
   3. TCP send size
   4. MTU size
4. SUMMARY: HiperSockets Multiple Write
   1. Requirements
      1. • IBM System z10
   2. Restrictions
      1. • Unsupported if z/OS is running as a guest in a z/VM environment.
      2. • Supported for large outbound messages only
5. SUMMARY: .. zIIP-Assisted HiperSockets Multiple Write
   1. Requirements
      1. • HiperSockets Multiple Write must be enabled
   2. Restrictions
      1. • Will only be used for large outbound TCP messages (that originate in this host).

Multiple VLAN Support in z/OS CS (V1R10)



- At V1R10 you can have up to 8 VLANs per stack, per OSA port, per IP version.
  1. With multiple VLAN IDs per stack on an OSA port, you must assign a VLAN ID to every one of the multiple Interfaces on that OSA port and
  2. You must assign or default to separate VMACs on each VLAN ID.
  3. As usual, each VLAN ID must be on a separate subnet.

Adjusting for Throughput & Latency on OSA Interfaces (V1R11)



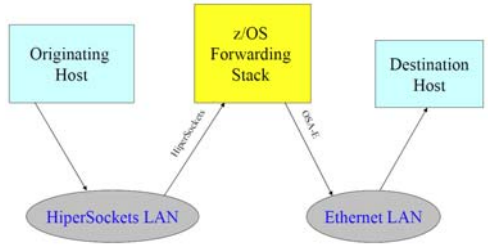
- On Device/Link
- On INTERFACE
- Recommend:
  - "Dynamic"
- At V1R12:
  - add WORKLOADQ on INBPERF DYNAMIC
    - Requires VMAC

Should see a significant throughput improvement for a single-session interactive workload
Some throughput improvement for multiplesession interactive workload
For streaming workloads the operating characteristics should be similar to the INBPERF parameter value of BALANCED

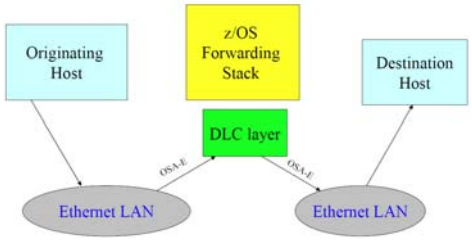
1. LAN idle timer settings have contributed to network latency on zSeries
   1. ┐ Even when the INBPERF parameter is specified with a value of MINLATENCY the permitted inter-packet gap is set to 20 microseconds
   2. ┌ LAN idle timer settings are static and can not be changed unless the connection to OSA connection is terminated and reestablished.
2. Performance studies have shown network latency improvements in environments where the CEC is under low utilization of up to 35% by tuning the Lan Idle timer within the OSA Express2 using a dynamic algorithm that takes workload characteristics. This dynamic algorithm involves taking the current default interpacket gap of 40 microseconds to as low as 1 microsecond.
3. A new INBPERF parameter option of DYNAMIC will now be permitted. This new configurable setting allows the TCP/IP stack to dynamically calculate the best values for the LAN idle timer settings. These settings will indirectly determine how frequently the OSA adapter will interrupt the host for inbound traffic.
4. New DYNAMIC option for the existing INBPERF parameter.
5. ┐ INBPERF parameter can be specified on the OSAExpress QDIO LINK or INTERFACE statement.
6. ┐ New option is valid for OSA-Express2 on an IBM System z9 EC or z9 BC with the corresponding Dynamic LAN Idle functional support
7. ┐ When specified for an OSA-Express device that does not support this new function then the option of BALANCED will be used for INBPERF parameter.
8. INBPERF
9. An optional parameter indicating how frequently the adapter should interrupt the host for inbound traffic. There are three supported static settings (MINCPU, MINLATENCY, and BALANCED). The static settings use static interrupt timing values. The static values are not always optimal for all workload types or traffic patterns, and cannot account for changes in traffic patterns.
10. There is also one supported dynamic (DYNAMIC) setting. This setting causes the host (stack) to dynamically adjust the timer-interrupt value while the device is active and in use. This function exploits an OSA hardware function called Dynamic LAN Idle. Unlike the static settings, the DYNAMIC setting reacts to changes in traffic patterns, and sets the interrupt-timing values at the point where throughput is maximized. The dynamic setting does not incur additional CPU consumption which might have been produced by using any of the static settings.
11. Valid settings include:
    1. DYNAMIC: The host to dynamically signals OSA to change the timer-interrupt value based on current inbound workload conditions. The DYNAMIC setting is effective only for Open Systems Adapter-Express2 on an IBM System z9 EC or z9 BC with the corresponding Dynamic LAN Idle functional support. See the 2094DEVICE Preventive Service Planning (PSP) and the 2096DEVICE Preventive Service Planning (PSP) buckets for further information about the level of Open Systems Adapter-Express2 that supports this function. When this setting is specified for a older Open Systems Adapter-Express, the stack reverts to using the BALANCED setting. The DYNAMIC setting should outperform the other three static settings for most workload mixes.
12. MINCPU
13. This setting uses a static interrupt-timing value, selected to minimize host interrupts without regard to throughput. This mode of operation might result in minor queueing delays (latency) for packets into the host, which is not optimal for workloads with demanding latency requirements.
14. MINLATENCY
15. This setting uses a static interrupt-timing value, selected to minimize latency (delay), by more aggressively presenting received packets to the host. This mode of operation generally results in higher CPU consumption than the other three settings. Use this setting only if host CPU consumption is not an issue.
16. BALANCED
17. This setting uses a static interrupt-timing value, selected to achieve reasonably high throughput and reasonably low CPU consumption. This is currently the default value.

"Fast Path" Routing: QDIO Acceleration (V1R11)

Without Fast Path QDIO Acceleration



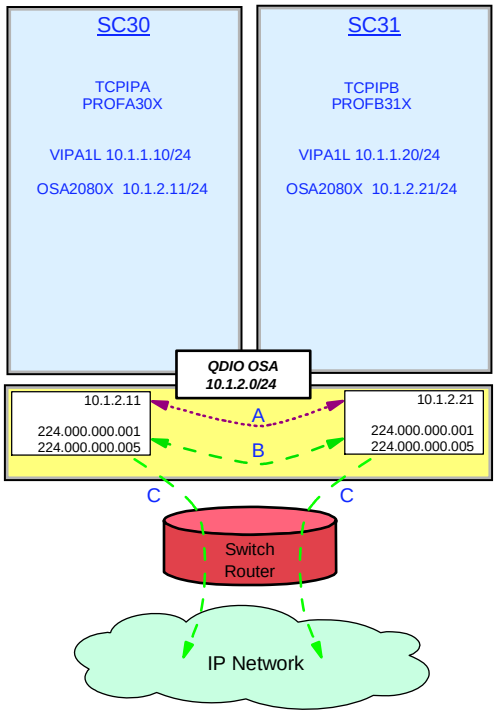
With Fast Path QDIO Acceleration



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

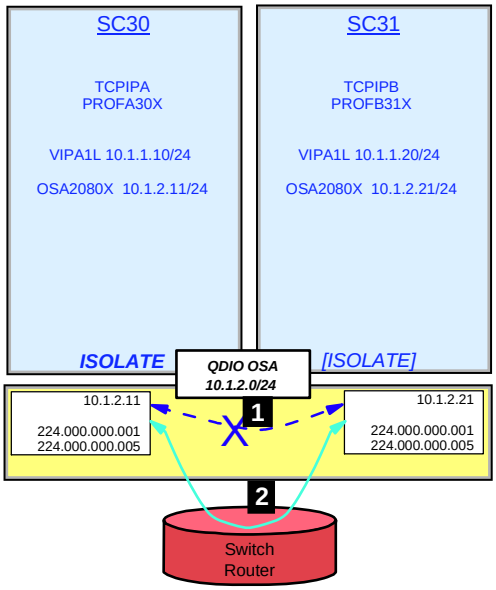
1. A function called "IQDIORouting" was introduced in an earlier release of z/OS to provide a fast path for HiperSockets routing of packets. The new function introduced in V1R11 enhances the "fast path" architecture and has fewer restrictions than IQDIORouting.
2. QDIO Acceleration is a function that allows forwarding of IP Packets from one interface to another interface without having to pass through the upper layers of the TCP/IP protocol stack. It provides "fast path" IP forwarding.
   1. See the visuals to understand how routing looks without and then with QDIO Acceleration.
   2. See the table to understand to which interface flows QDIO Acceleration applies.
3. Requirement:
   1. IPConfig QDIOACCELERATOR is mutually exclusive with IQDIOROUTING
   2. Works with Unfragmented packets only
   3. IPSecurity cannot be enabled
   4. IP Forwarding should be enabled unless you want this function only for SD
4. Specifies that inbound packets that are to be forwarded by this TCP/IP stack are eligible to be routed directly between any of the following combinations of interface types:
   1. A HiperSockets interface and an OSA-Express QDIO interface
   2. Two OSA-Express QDIO interfaces
   3. Two HiperSockets interfaces
5. These packets do not need to be sent to this TCP/IP stack for forwarding. This also applies to packets that would be forwarded by the Sysplex Distributor. This type of routing is called QDIO Accelerator.
6. QDIO Acceleration is supported with or without the VIPAROUTE statement. When QDIO Accelerator is active, the stack dynamically creates QDIO Accelerator routes as it forwards packets in any of the inbound and outbound DLC combinations previously described. The DLC layer can perform accelerated routing for packets across these routes, bypassing the IP forwarding function in the stack. Similarly, the stack dynamically creates QDIO Accelerator routes for packets that would be forwarded by the sysplex distributor in any of the inbound and outbound DLC combinations. The DLC layer can perform accelerated sysplex distributor routing for such packets.
7. Restrictions:
   1. QDIO Accelerator is supported for IPv4 only.
   2. You cannot enable QDIO Accelerator support if you enable IP security on the stack.
   3. If IP forwarding is disabled on the stack, then QDIO Accelerator applies only to packets that are forwarded by the sysplex distributor.
   4. Packets from the sysplex distributor to the target are not accelerated with the VIPAROUTE destination when the outbound interface is HiperSockets.

OSA Connection Isolation (V1R11)



1. Another method available to isolate traffic across a shared OSA port is OSA Connection Isolation. This method can be deployed with or without out assigning a VLAN ID or a VMAC to the OSA port.
2. Many customers share OSA-Express ports across logical partitions, especially if capacity is not an issue. Each stack sharing the OSA port registers certain IP addresses and multicast groups with the OSA.
3. For performance reasons, the OSA-Express bypasses the LAN and routes packets directly between the stacks when possible.
4. For unicast packets, OSA internally routes the packet when the next-hop IP address is registered on the same LAN or VLAN by another stack sharing the OSA port.
   1. A: You see how TCPIPA routes a packet to 10.1.2.21 in TCPIPB over the OSA port without exiting out onto the LAN because the next hop to reach the destination is registered in the OSA Address Table (OAT); the TCPIPA routing table indicates that the destination can be reached by hopping through the direct connection to the 10.1.2.0/24 network.
   2. B For multicast (e.g., OSPF protocol packets), OSA internally routes the packet to all sharing stacks on the same LAN or VLAN which registered the multicast group. Note how TCPIPA and TCPIPB have each registered multicast addresses for OSP (224.000.000.00n) in the OSA port.
   3. C OSA also sends the multicast/broadcast packet to the LAN. For broadcast (not depicted), OSA internally routes the packet to all sharing stacks on the same LAN or VLAN.
5. Some customers express concerns about this efficient communication path and wish to disable it; they may wish to disable the function because traffic flowing internally through the OSA adapter bypasses any security features implemented on the external LAN

OSA Connection Isolation (V1R11)

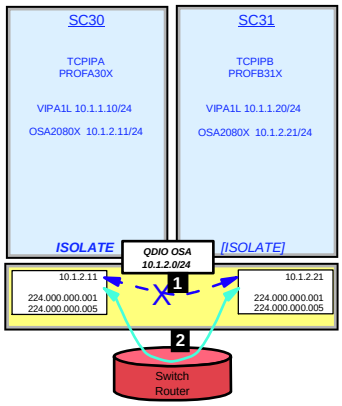


1 = Direct route between TCPIPA and TCPIPB
2 = Indirect route between TCPIPA and TCPIPB

```
INTERFACE OSA2080X
 DEFINE IPAQENET
 PORTNAME OSA2080
 IPADDR 10.1.2.11/24
 MTU 1492
 VLANID 10
 VMAC ROUTEALL
 ISOLATE

INTERFACE OSA2080X
DEFINE IPAQENET
 PORTNAME OSA2080
 IPADDR 10.1.2.21/24
 MTU 1492
 VLANID 10
 VMAC ROUTEALL
 [ISOLATE ]
```


1. Some environments require strict controls for routing data traffic between servers or nodes. In certain cases, the LPAR-to-LPAR capability of a shared OSA port can prevent such controls from being enforced. For example, you may need to ensure that traffic flowing through the OSA adapter does not bypass firewalls or intrusion detection systems implemented on the external LAN. We have described several ways to isolate traffic from different LPARs on a shared OSA port, with one of these methods being OSA Connection Isolation.
2. The feature is called OSA Connection Isolation in z/OS, but it is also available in z/VM, where it is called QDIO data connection isolation or VSWITCH port isolation. It allows you to disable the internal routing on a QDIO connection basis, providing a means for creating security zones and preventing network traffic between the zones. It also provides extra assurance against a misconfiguration that might otherwise allow such traffic to flow as in the case of an incorrectly defined IP filter. With interface isolation, internal routing can be controlled on an LPAR basis. When interface isolation is enabled, the OSA will discard any packets destined for a z/OS LPAR that is registered in the OAT as isolated.
3. QDIO interface isolation is supported by Communications Server for z/OS V1R11 and all OSA-Express3 and OSA-Express2 features on System z10, and by all OSA-Express2 features on System z9, with an MCL update. Refer to the appropriate Preventive Service Planning bucket for details regarding your System z server.
4. Coding ISOLATE on your INTERFACE statement enables the function. It tells the OSA-Express not to allow communications to this stack other than over the LAN.
    1. As the visual depicts, the ISOLATE parameter is available only on the INTERFACE statement. To eliminate the direct path through the OSA between the two depicted LPARs, you need code ISOLATE on only one of the two INTERFACEs. We have coded it on both in order to assure, that if any other LPAR starts sharing the OSA port, that other LPAR cannot use the direct path to communicate even with TCPIPB..
5. If you attempt to code ISOLATE on an INTERFACE that does not support the ISOLATE function, you receive a message:
    1. EZD0022I INTERFACE OSA2080X DOES NOT SUPPORT THE ISOLATE FUNCTION
6. Dynamic routing protocol implementations with RIP or OSPF require careful planning on LANs where OSA-Express connection isolation is in effect; the dynamic routing protocol learns of the existence of the direct path but is unaware of the isolated configuration, which renders the direct path across the OSA port to the registered target unusable. If the direct path that is operating as ISOLATEd is selected, you will experience routing failures.
7. If the visibility of such errors is undesirable, you can take other measures to avoid the failure messages. If you are simply attempting to bypass the direct route in favor of another, indirect route, you can accomplish this as well with some thoughtful design.
8. For example, you might purposely bypass the direct path by using Policy Based Routing (PBR) or by coding static routes that supersede the routes learned by the dynamic routing protocol. You might adjust the weights of connections to favor alternate interfaces over the interfaces that have been coded with ISOLATE.
9. If, however, TCPIPA and TCPIPB do need to exchange information, you will need to deploy an effective route that bypasses the direct route between them. Therefore, at TCPIPA you might add a non-replaceable static route to an IP address in TCPIPB; the static route in the BEGINROUTES block points to the next-hop router on the path indicated with (2) in the visual.

OSA Connection Isolation: Dynamic Routing Considerations (V1R11)



1 = Direct route between TCPIPA and TCPIPB
2 = Indirect route between TCPIPA and TCPIPB

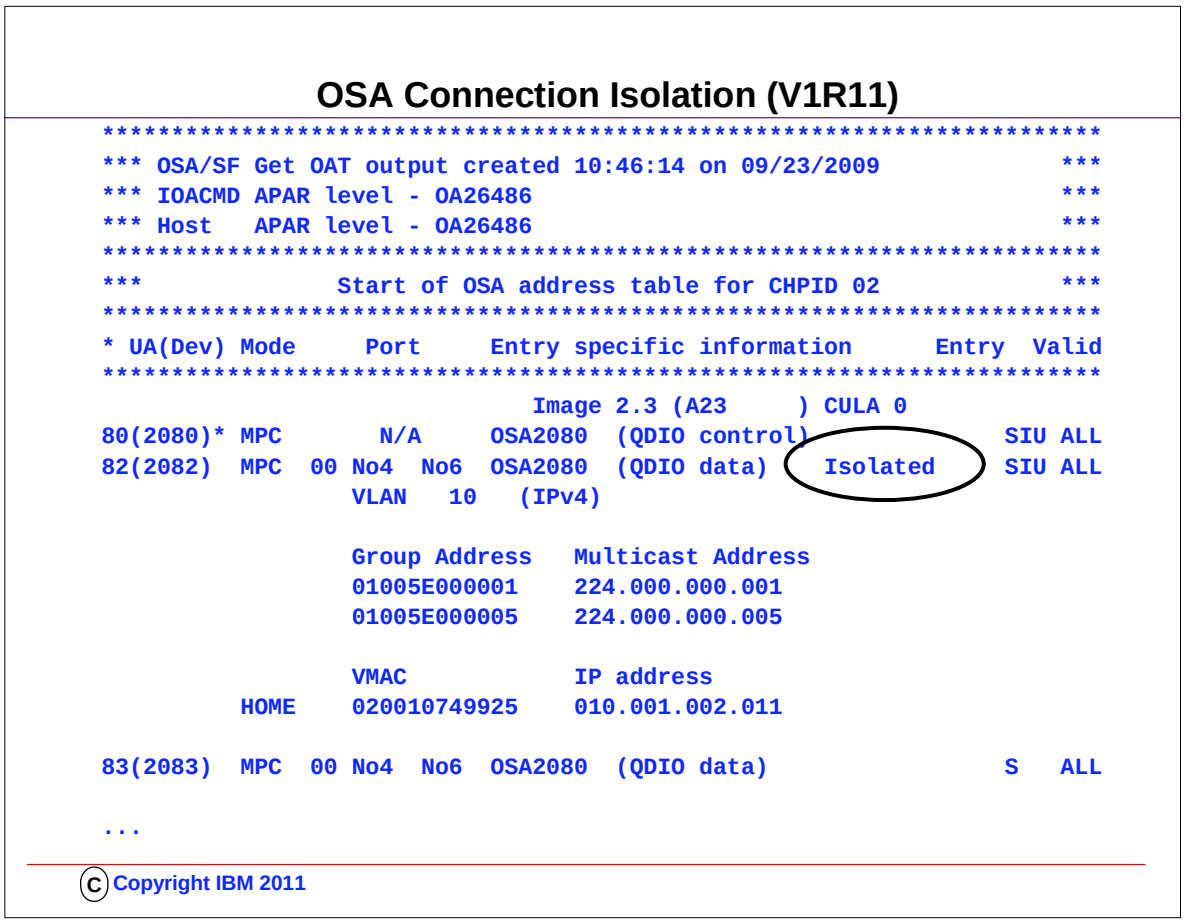
```
;TCPIPA.TCPPARMS(ROUTA30X)
; AUTOLOG LIST:  INITIALIZE OMPROUTE
...
BEGINRoutes
; Direct Routes - Routes directly connected to my interfaces
; Destination  Subnet Mask   First Hop     Link Name   Packet Size
 ROUTE 10.1.2.0/24          10.1.2.240    OSA2080X    mtu 1492
 ROUTE 10.1.1.0/24          10.1.2.240    OSA2080X    mtu 1492
 ROUTE 10.1.1.20/32         10.1.2.240    OSA2080X    mtu 1492
 ENDRoutes
```

- Combine OMPROUTE with Static Routes to bypass direct routing through OSA port.

```
;TCPIPB.TCPPARMS(ROUTB31X)
; AUTOLOG LIST:  INITIALIZE OMPROUTE
...
BEGINRoutes
; Direct Routes - Routes directly connected to my interfaces
; Destination  Subnet Mask   First Hop     Link Name   Packet Size
;
 ROUTE 10.1.2.0/24          10.1.2.240    OSA2080X    mtu 1492
 ROUTE 10.1.1.0/24          10.1.2.240    OSA2080X    mtu 1492
 ROUTE 10.1.1.10/32         10.1.2.240    OSA2080X    mtu 1492
 ENDRoutes
```


1. Some environments require strict controls for routing data traffic between servers or nodes. In certain cases, the LPAR-to-LPAR capability of a shared OSA port can prevent such controls from being enforced. For example, you may need to ensure that traffic flowing through the OSA adapter does not bypass firewalls or intrusion detection systems implemented on the external LAN. We have described several ways to isolate traffic from different LPARs on a shared OSA port, with one of these methods being OSA Connection Isolation.
2. The feature is called OSA Connection Isolation in z/OS, but it is also available in z/VM, where it is called QDIO data connection isolation or VSWITCH port isolation. It allows you to disable the internal routing on a QDIO connection basis, providing a means for creating security zones and preventing network traffic between the zones. It also provides extra assurance against a misconfiguration that might otherwise allow such traffic to flow as in the case of an incorrectly defined IP filter. With interface isolation, internal routing can be controlled on an LPAR basis. When interface isolation is enabled, the OSA will discard any packets destined for a z/OS LPAR that is registered in the OAT as isolated.
3. QDIO interface isolation is supported by Communications Server for z/OS V1R11 and all OSA-Express3 and OSA-Express2 features on System z10, and by all OSA-Express2 features on System z9, with an MCL update. Refer to the appropriate Preventive Service Planning bucket for details regarding your System z server.
4. Coding ISOLATE on your INTERFACE statement enables the function. It tells the OSA-Express not to allow communications to this stack other than over the LAN.
    1. As the visual depicts, the ISOLATE parameter is available only on the INTERFACE statement. To eliminate the direct path through the OSA between the two eepcited LPARs, you need code ISOLATE on only one of the two INTERFACEs. We have coded it on both in order to assure, that if any other LPAR starts sharing the OSA port, that other LPAR cannot use the direct path to communicate even with TCPIPB..
5. If you attempt to code ISOLATE on an INTERFACE that does not support the ISOLATE function, you receive a message:
    1. EZD0022I INTERFACE OSA2080X DOES NOT SUPPORT THE ISOLATE FUNCTION
6. Dynamic routing protocol implementations with RIP or OSPF require careful planning on LANs where OSA-Express connection isolation is in effect; the dynamic routing protocol learns of the existence of the direct path but is unaware of the isolated configuration, which renders the direct path across the OSA port to the registered target unusable. If the direct path that is operating as ISOLATEd is selected, you will experience routing failures.
7. If the visibility of such errors is undesirable, you can take other measures to avoid the failure messages. If you are simply attempting to bypass the direct route in favor of another, indirect route, you can accomplish this as well with some thoughtful design.
8. For example, you might purposely bypass the direct path by using Policy Based Routing (PBR) or by coding static routes that supersede the routes learned by the dynamic routing protocol. You might adjust the weights of connections to favor alternate interfaces over the interfaces that have been coded with ISOLATE.
9. If, however, TCPIPA and TCPIPB do need to exchange information, you will need to deploy an effective route that bypasses the direct route between them. Therefore, at TCPIPA you might add a non-replaceable static route to an IP address in TCPIPB; the static route in the BEGINROUTES block points to the next-hop router on the path indicated with (2) in the visual.
10. The effect of ICMP redirect packets: To avoid the override of the ICMP redirect packets that would most likely occur from the router to the originating host, you need to disable the receipt of ICMP redirects in the IP stacks or disable ICMP redirects at the router. If you are using OMPROUTE, ICMP redirects are automatically disabled, as evidenced by the message that appears during OMPROUTE initialization:
    1. EZZ7475I ICMP WILL IGNORE REDIRECTS DUE TO ROUTING APPLICATION BEING ACTIVE
11. The visual shows the coding for Static non-replaceable routes at TCPIPA and TCPIPB to override direct route through OSA port

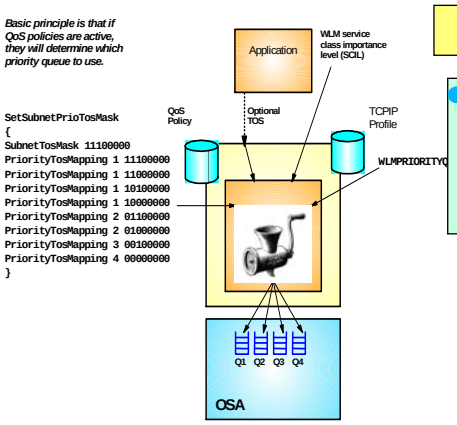
OSA Connection Isolation (V1R11)



```
**********************************************************************
*** OSA/SF Get OAT output created 10:46:14 on 09/23/2009           ***
*** IOACMD APAR level - OA26486                                    ***
*** Host   APAR level - OA26486                                    ***
**********************************************************************
***             Start of OSA address table for CHPID 02           ***
**********************************************************************
* UA(Dev) Mode     Port     Entry specific information      Entry  Valid
**********************************************************************
                              Image 2.3 (A23     ) CULA 0
80(2080)* MPC       N/A     OSA2080  (QDIO control)              SIU ALL
82(2082)  MPC  00 No4  No6  OSA2080  (QDIO data)    Isolated    SIU ALL
                 VLAN   10   (IPv4)

                 Group Address   Multicast Address
                 01005E000001    224.000.000.001
                 01005E000005    224.000.000.005

                 VMAC            IP address
          HOME   020010749925    010.001.002.011

83(2083)  MPC  00 No4  No6  OSA2080  (QDIO data)               S   ALL

...
```


1. Even the OSA/SF display shows where ISOLATE is enabled, as you can see from the display.

Exploiting QDIO Priority Queueing with WLM Service Classes (V1R11)



IPCONFIG WLMPRIORITYQ

- Establish use of outbound QDIO queues for a traffic type either
  - with PAGENT & SETSUBNETPRIOTOSMASK, or
  - with WLMPRIORITYQ

SYSTEM tasks are always assigned QDIO Priority of 1
Default IOPRIORITIES for Importance Levels:

0. SYSSTC service class
1. User defined services classes Importance level 1
2. User defined services classes with Importance level 2
3. User defined services classes with Importance level 3
4. User defined services classes with Importance level 4
5. User defined services classes with Importance level 5
6. User defined service classes associated with a Discretionary goal

WLMPRIORITYQ: YES
IOPRI1 0
IOPRI2 1
IOPRI3 2 3
IOPRI4 4 5 6 FWD

```
policyRule        telnetd       # telnet traffic
{
   protocolNumberRange  6
   SourcePortRange    23
   policyActionReference  interactive1
}

policyAction   interactive1
{
   policyScope    DataTraffic
   OutgoingTOS    10000000
}
```

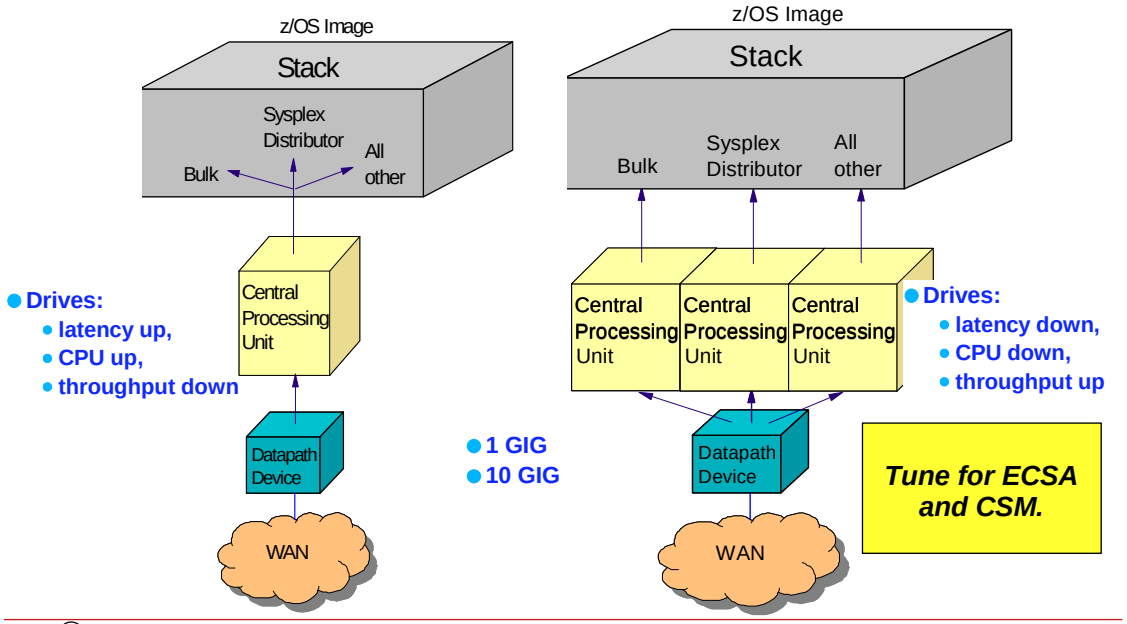
10000000 Send on QDIO Q1

1. The QDIO OSAs are implemented with four internal queues. Outbound Data traffic is distributed over these four queues based upon a Quality of Service (QoS) definition that established Types of Service in the "Precedence Bits" of the IP Header. Most applications fail to establish these precedence bits; Enterprise Extender is an exception to this. Other applications are assigned precedence bits based upon a QoS policy that you may have defined with z/OSMF or with z/OS Configuration Assistant GUI and then installed with Policy Agent into the TCP/IP stack.
2. The first visual in the upper left shows you the four QDIO queues and shows you how different Types of Service are mapped within Policy Agent to distributed traffic outbound over each of the four queues.
3. The visual below the aforementioned visual shows you a sample policy that might be used to assign a high priority (TOS of 10000000) to Telnet traffic and therefore cause it to be dispatched on QDIO OSA Queue #1.
4. In general most shops do little to nothing to prioritize their OSA-Express outbound data, missing any benefits the prioritization provides
5. Beginning with V1R11, it is now possible to allow outbound traffic to be assigned precedence bits based upon WLM priorities and "Service Class Importance Levels."
   1. Since the WLM service classes should already be assigned to the jobs, all that needs to be done is to give the stack ‘permission’ to use it for prioritization.
   2. Defaults are provided that should give a good distribution of work across the priority queues.
   3. If QoS or the application has assigned an IPv4 ToS/IPv6 Traffic Class then enabling this function will only affect those packets assigned a ToS/Traffic Class value of zeros.
   4. Enterprise Extender always assigns a non-zero ToS/Traffic Class so unless it is changed to zero by QoS, Enterprise Extender traffic is not affected.
6. Therefore, with V1R11, all you need to do is enable the use of WLM Service Class importance Level as a means of assigning traffic to the QDIO queues. You do this by enabling:
   1. IPCONFIG WLMPRIORITYQ or GLOBALCONFIG WLMPRIORITYQ (WLMPRIORITYQ: YES on a Netstat Config indicates that WLMPRIORITYQ is enabled) WLMPRIORITYQ specifies that OSA-Express QDIO write priority values should be assigned to packets associated with WorkLoad Manager service class values and to forwarded packets.
      1. If you do not want to accept the default queueing, you may override it with a parameter of IOPRIn. Below you see the default settings for IOPRIn when you sepcify WLM:PRIORITYQ by itself on the IPCONFIG statement.
      2. IOPRI1 0 OSA-Express priority queue 1 is used for packets from jobs with a control value 0 (SYSSTC)
      3. IOPRI2 1 OSA-Express priority queue 2 is used for packets from jobs with control value 1 (services classes with Importance level 1)
      4. IOPRI3 2 3 OSA-Express priority queue 3 is used for packets from jobs with control values 2 and 3 (services classes with Importance levels 2 and 3)
      5. IOPRI4 4 5 6 FWD OSA-Express priority queue 4 is used for packets from jobs with control values 4, 5, and 6 (services classes with Importance levels 4 and 5 and discretionary) as are all non-accelerated forwarded packets
7. Points to remember:.
   1. WLMPRIORITYQ has little effect unless there is enough traffic to cause contention for the OSA-Express resources
   2. WLMPRIORITYQ has no effect unless packet IPv4 ToS/IPv6 Traffic Class is zeros. This is typically the case if you have not defined a network QoS policy
   3. WLMPRIORITYQ does not affect accelerated packet priority.

Inbound Workload Queuing: SD and Bulk Data (V1R12)

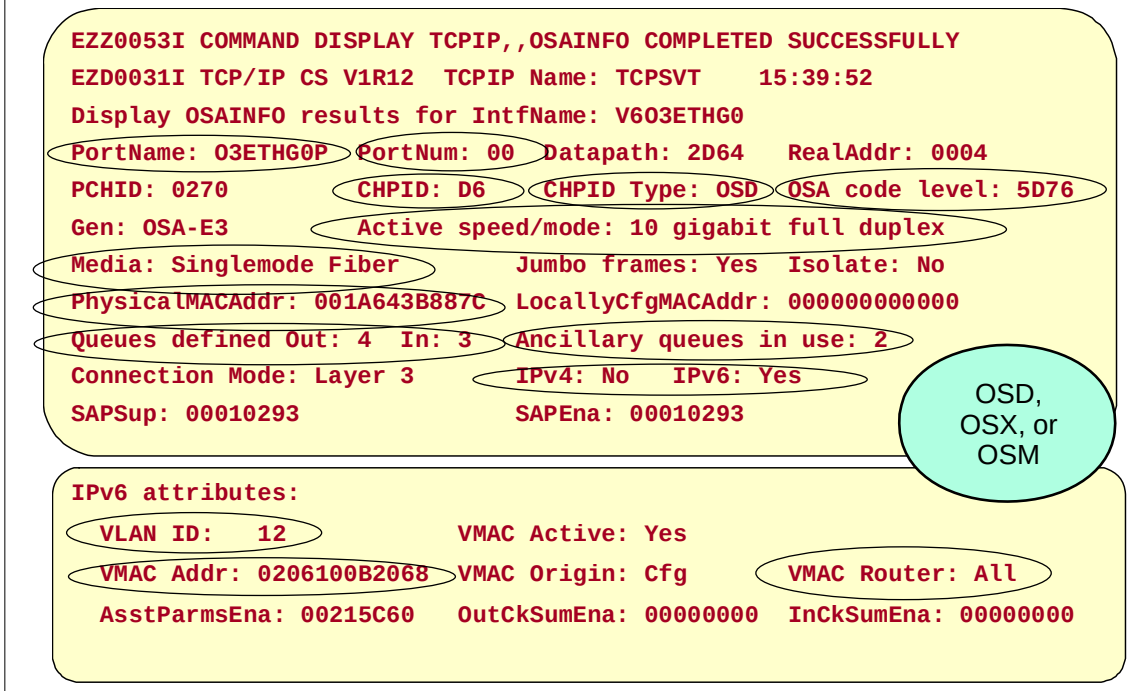
Prior to V1R12: Only 1 OSA Read Queue, but 4 OSA Write Queues!
With V1R12: 4 OSA Read Queues and still 4 OSA Write Queues.

INTERFACE ... INBPERF DYNAMIC WORKLOADQ



1. Prior to z/OS V1R12, all inbound QDIO traffic is received on a single read queue regardless of the data type. The maximum amount of storage available for inbound traffic is limited to the read buffer size (64K read SBALs) times the maximum number of read buffers (126). A single process is used to package the data, queue it, and schedule the TCP/IP stack to process it. This same process also performs acceleration functions, such as Sysplex Distributor connection routing accelerator.
2. The TCP/IP stack must separate the traffic types to be forwarded to the appropriate stack component that will process them. For these reasons, z/OS Communications Server is becoming the bottleneck as OSA-Express3 10GbE nears line speed. z/OS Communications Server is injecting latency and increasing processor utilization. This can impede scalability.
3. Under the pre-V1R12 z/OS Communications Server model, another QDIO input process will eventually be driven, and another TCP/IP stack thread, thus allowing multiple threads to process the one inbound read queue. However, this is only done when the OSA detects the host is now "falling behind" using the QDIO interrupt threshold algorithm.
4. z/OS Communications Server is becoming the bottleneck as OSA nears 10GbE line speed, this behavior Injects latency, increases processor utilization, and impedes scalability. For BULK Data, multiple processes are used for inbound traffic when data is accumulating on the read queue. This can cause bulk data packets for a single TCP connection to arrive at the TCP layer out of order. Each time the TCP layer on the receiving side sees out of order data, it transmits a duplicate ACK. Overall, throughput is harmed for bulk data traffic.
5. With z/OS Communications Server V1R12, inbound traffic separation is supported using multiple read queues. TCP/IP will register with OSA which traffic to be received on each read queue. The OSA-Express Data Router function routes traffic to the correct queue.
6. With z/OS Communications Server V1R12, inbound traffic separation is supported using multiple read queues. TCP/IP will register with OSA which traffic to be received on each read queue. The OSA-Express Data Router function routes traffic to the correct queue.
7. Each read queue can be serviced by a separate process. The primary input queue is used for general traffic. One or more ancillary input queues (AIQs) are used for specific traffic types. Sysplex distributor and bulk data traffic is presorted by OSA and routed to z/OS Communications Server on unique AIQs. All other traffic is routed to z/OS Communications Server on the primary input queue. z/OS Communications Server can now process sysplex distributor, bulk data, and other traffic concurrently and independently.
8. The primary queue is always assigned Queue Identifier 1 (QID 1). Each ancillary queue is assigned a Queue Identifier based on when it gets internally registered.
9. The supported traffic types are streaming bulk data and sysplex distributor. Examples of bulk data traffic are FTP, TSM, NFS, and TDMF.
10. Both IP versions are supported for all types of traffic.
    1. With bulk data traffic separated onto its own read queue, TCP/IP will service the bulk data queue from a single processor. This solves the out of order delivery issue – there are no more race conditions.
    2. With sysplex distributor traffic separated onto its own read queue, it can be efficiently accelerated or presented to the target application.
    3. All other traffic is processed simultaneous with the bulk data and sysplex distributor traffic
    4. The dynamic LAN idle timer is updated independently for each read queue. This ensures the most efficient processing of inbound traffic based on the traffic type.
11. The QDIO inbound workload queuing function is enabled with the INBPERF DYNAMIC WORKLOADQ setting on IPAQENET and IPAQENET6 INTERFACE statements. WORKLOADQ is not supported for INBPERF DYNAMIC on IPAQENET LINK statements. WORKLOADQ does require the VMAC on the INTERFACE definition, but you can allow just a dynamically generated value for VMAC. For steps to convert from IPv4 IPAQENET DEVICE, LINK, and HOME definitions to the IPv4 IPAQENET INTERFACE statement refer to z/OS Communications Server: IP Configuration Guide.
12. Each ancillary queue will consume:
    1. Approximately nine additional pages of ECSA
    2. An additional but tunable amount of fixed CSM data space as specified by the READSTORAGE parameter

OSA-E3 QDIO Address Table Displays: D OSAINFO (V1R12)



1. D OSAINFO is valid on an OSA-E3 in QDIO Mode (either CHPID Type of OSD, OSX, or OSM) as long as the interface has been defined with the INTERFACE Statement.
2. OSA requirements:
   1. OSA-Express3 Ethernet features in QDIO mode running on an
   2. IBM System z10
   3. See the 2097DEVICE and 2098DEVICE Preventive Service Planning (PSP) buckets for the required MCL levels
3. You can issue the DISPLAY OSAINFO command to determine if OSA supports the command
   1. INTFNAME must be defined as IPAQENET or IPAQENET6
   2. INTFNAME must be active
4. The command sorts addresses and ports in ascending order
5. Impact of command on both OSA and Communications Server resources should be insignificant
6. OSA requirements:
   1. OSA-Express3 Ethernet features in QDIO mode running on an IBM System z10
   2. See the 2097DEVICE and 2098DEVICE Preventive Service Planning (PSP) buckets for the required MCL levels
7. Sections of the Output Display:
   1. This part of the sample reply is the start of the BASE section. The BASE section shows general information about the OSA such as the CHPID (in this sample the CHPID is D6).
      1. All of the fields displayed in the reply are documented in z/OS Communications Server IP System Administrator's Commands Version 1 Release 12.
      2. Message EZZ0053I is not part of the report but instead it's issued when the display command is accepted.
      3. Message EZD0031I is the 1st message in the multi-write to operator reply and is issued when all information has been received from OSA
   2. This part of the sample reply is the end of the BASE section. This sample shows information about the IPv6 Layer 3 attributes such as the Global VLAN ID and VMAC information.
      1. If the data device has IPv4 enabled (which this sample does not), the IPv4 Layer 3 attributes are displayed.
      2. If the data device has IPv6 enabled (which this sample does), the IPv6 Layer 3 attributes are displayed.
      3. If the data device has IPv4 and IPv6 enabled , the IPv4 Layer 3 attributes are displayed first, followed by the IPv6 Layer 3 attributes.

OSA-E3 QDIO Address Table Displays: D OSAINFO (V1R12)

```
Registered Addresses:
  IPv4 Unicast Addresses:
    ARP: Yes  Addr: 16.2.16.107
    Total number of IPv4 addresses:      1
  IPv4 Multicast Addresses:
    MAC: 01005E000001  Addr: 224.0.0.1
    Total number of IPv4 addresses:      1
  IPv6 Unicast Addresses:
    Addr: FE80::11:16:32:104
    Total number of IPv6 addresses:      1
  IPv6 Multicast Addresses:
    MAC: 333300000001  Addr: FF02::1
    MAC: 3333FF010001  Addr: FF02::1:FF01:1
    MAC: 3333FF010002  Addr: FF02::1:FF01:2
    MAC: 3333FF010003  Addr: FF02::1:FF01:3
    Total number of IPv6 addresses:      4
```


1. This part of the sample reply is the REGADDRS section in its entirety. Displayed here are all the IPv4 and IPv6 unicast and multicast addresses registered with the OSA.
   1. Note that the IPv4 information conflicts with other sections of the reply. The IPv4 information was inserted here for illustration purposes only.
   2. If the interface has IPv4 enabled (which this sample does), the IPv4 registered unicast and multicast addresses are displayed. The ARP field indicates if the OSA is performing ARP for an IPv4 unicast address.
   3. If the interface has IPv6 enabled (which this sample does), the IPv6 registered unicast and multicast addresses are displayed.
2. Continued sections of the Output display:

OSA-E3 QDIO Address Table Displays: D OSAINFO (V1R12)

```
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
```


1. This part of the sample reply is the REGADDRS section in its entirety. Displayed here are all the IPv4 and IPv6 unicast and multicast addresses registered with the OSA.
2. Note that the IPv4 information conflicts with other sections of the reply. The IPv4 information was inserted here for illustration purposes only.
3. If the interface has IPv4 enabled (which this sample does), the IPv4 registered unicast and multicast addresses are displayed. The ARP field indicates if the OSA is performing ARP for an IPv4 unicast address.
4. If the interface has IPv6 enabled (which this sample does), the IPv6 registered unicast and multicast addresses are displayed.
5. Continued sections of the Output display:
   1. BULKDATA:
      1. This part of the sample reply is the BULKDATA section in its entirety. Displayed here are the source and destination IP address and ports of the TCP connections for which OSA is performing QDIO Inbound Workload Queuing for streaming connections. If the interface has QDIO Inbound Workload Queuing enabled for BULKDATA and there is at least one connection, the BULKDATA section is displayed.
      2. Note that you can see IPv4 or IPv6 addresses here but not both as QDIO Inbound Workload Queuing is not allowed when a single datapath device is used for both IPv4 and IPv6.
   2. Sysplex Distribution:
      1. This part of the sample reply is the SYSDIST section in its entirety. Displayed here are the destination IP address for which OSA is performing QDIO Inbound Workload Queuing for sysplex distributor. If the interface has QDIO Inbound Workload Queuing enabled for sysplex distributor and at least one destination address, the SYSDIST section is displayed.
      2. Note that you can see IPv4 or IPv6 addresses here but not both as Inbound Workload Queuing is not allowed when a single datapath device is used for both IPv4 and IPv6.
      3.
6. The first number shows the total number of lines displayed. The second number shows the total number of lines it's possible to display. The MAX operator can be specified to limit the total number of lines displayed.
7. If MAX=* is specified and more than 65,535 lines are required, Report truncated: Max lines limit reached is displayed instead of the message with the counts.
8.

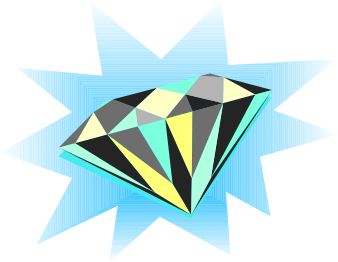
Displaying OSAINFO for an OSM CHPID Interface (V1R12)

```
D TCPIP,,OSAINFO,INTFNAME=EZ6OSM01
EZZ0053I COMMAND DISPLAY TCPIP,,OSAINFO COMPLETED SUCCESSFULLY
EZD0031I TCP/IP CS V1R12  TCPIP Name: TCPIP     15:12:35 212
Display OSAINFO results for IntfName: EZ6OSM01
PortName: IUTMP00A   PortNum: 00   Datapath: 2342  RealAddr: 0002
PCHID: 0531        CHPID: 0A    CHPID Type: OSM      OSA code level: 0906
Gen: OSA-E3        Active speed/mode: 1000 mb/sec full duplex
Media: Copper      Jumbo frames: Yes      Isolate: Yes
PhysicalMACAddr: 00145E7769EC     LocallyCfgMACAddr: 000000000000
Queues defined    Out: 1  In: 1  Ancillary queues in use: 0
Connection Mode: Layer 2
SAPSup: 0009F603               SAPEna: 00082603
Layer 2 attributes:
  VLAN ID: N/A         VMAC Active: Yes
  VMAC Addr: 0200769EC008    VMAC Origin: OSA
15 of 15 lines displayed
End of report
```

Displaying OSAINFO for an OSX CHPID Interface (V1R12)

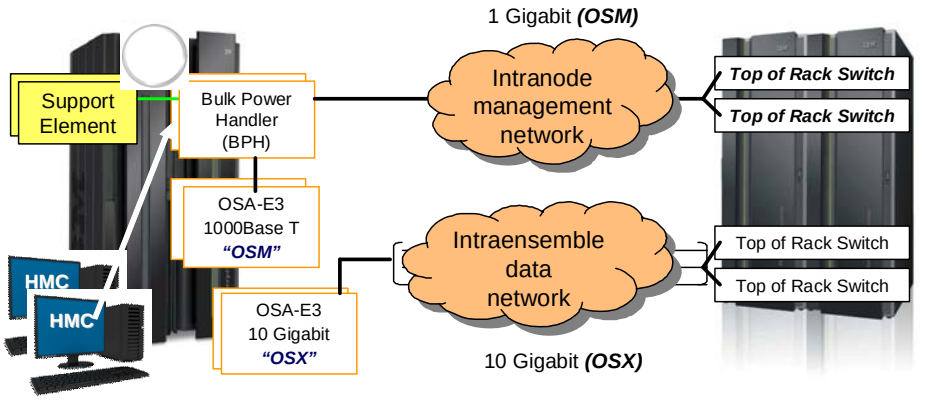
```
D TCPIP,,OSAINFO,INTFNAME=OSX2300
EZZ0053I COMMAND DISPLAY TCPIP,,OSAINFO COMPLETED SUCCESSFULLY
EZD0031I TCP/IP CS V1R12  TCPIP Name: TCPIP     15:06:03 203
Display OSAINFO results for IntfName: OSX2300
PortName: IUTXP018     PortNum: 00    Datapath: 2302   RealAddr: 0002
PCHID: 0590         CHPID: 18      CHPID Type: OSX     OSA code level: 0D0A
Gen: OSA-E3      Active speed/mode: 10 gigabit full duplex
Media: Multimode Fiber          Jumbo frames: Yes   Isolate: No
PhysicalMACAddr: 001A643B2135   LocallyCfgMACAddr: 000000000000
Queues defined Out: 4  In: 1    Ancillary queues in use: 0
Connection Mode: Layer 3    IPv4: Yes   IPv6: No
SAPSup: 000FF603                SAPEna: 0008A603
IPv4 attributes:
  VLAN ID:   99          VMAC Active: Yes
  VMAC Addr: 02BECB000002   VMAC Origin: Cfg    VMAC Router: All
  AsstParmsEna: 00200C57   OutCkSumEna: 0000001A  InCkSumEna: 0000001A
Registered Addresses:
   IPv4 Unicast Addresses:
     ARP: Yes  Addr: 172.30.99.1
     Total number of IPv4 addresses:      1
   IPv4 Multicast Addresses:
     MAC: 01005E000001  Addr: 224.0.0.1
     Total number of IPv4 addresses:      1
23 of 23 lines displayed
End of report
```

Gem Available to Prepare you for zEnterprise™ System and z/OS V1R12



Overview of zEnterprise System and Ensemble Networking

Management Network: IPv6 only
Implement it and then forget about it!
Data Network: IPv4 or IPv6



- Intranode management network (INMN)
  - 1000Base-T OSA-Express3 (copper) --- QDIO (***CHPID Type OSM***) – Cables are 3.2 meters long from OSM to BPH in CEC and 26 meters from BPH to TOR
  - HMC security is implemented with standard practices ***PLUS*** additional security mechanisms:
    - Isolated IPv6 network with ***"link-local"*** addresses only; authentication and authorization and access control, etc.
- Intraensemble data network (IEDN)
  - 10 Gigabit OSA-Express3 --- QDIO (***CHPID Type OSX***) – Cables are maximum of 26 meters long to TOR & 10km long-range
  - Security is implemented with standard practices ***PLUS*** additional security mechanisms: access control, authentication, authorization, application security, routing table restrictions, IP Filtering, etc.
  - Networks can be further isolated using VLAN and VMAC segmentation of the network connections

Simple Migration Plan to Prepare for IEDN in the Ensemble

- In preparation for IEDN OSA port interfaces -- even if using IPv4:
  1. Convert all OSD (QDIO) Definitions from the old statement syntax of DEVICE and LINK to INTERFACE
     - Simplifies definitions by including IP address, source VIPAs, MTU sizes, etc.
  2. Familiarizes you with the new syntax which is REQUIRED even for IPv4 IEDN OSX interfaces

```
TRL14 VBUILD TYPE=TRL
TRL014 TRLE LNCTL=MPC,
            READ=(A60),
            WRITE=(A61),
            DATAPATH=(A62),
            PORTNAME=GIG1F,
            MPCLEVEL=QDIO
```

```
;GbE --- CHPID 1F ---(INTERFACE Version)----
;
INTERFACE     OSDGIG1F
DEFINE        IPAQENET      CHPIDTYPE OSD
PORTNAME      GIG1F
MTU           1492
IPADDR        192.168.20.95/24
VMAC          [ ROUTEALL ]
```

1) INTERFACE Statement

```
;GbE --- CHPID 1F -----(DEVICE/LINK Ve
;
DEVICE GIG1F  MPCIPA PRIROUTER AUTORESTART
LINK LGIG1F   IPAQENET GIG1F
;
HOME
192.168.20.95
;
BEGINROUTES
ROUTE 192.168.20.0/24       =              LGIG1F   MTU 1492
ROUTE DEFAULT               192.168.20.1   LGIG1F   MTU 1492
ENDRoutes
```

1) Device/Link Statement
2) HOME Statement
3) Routing Statement

Since V1R4: Implement IPv6 Without Exploiting It!!

Effect of the IPv6 Addressing on a Participating z/OS Stack

1. Change to BPXPRMxx (UNIX member in PARMLIB)
2. Change to NETSTAT output
   - LONG for IPv6 (or mixed) output (SHORT not supported when IPv6 enabled.)
   - Must use NETSTAT ROUTE to see an IPv6 route
     - and not NETSTAT GATE, which sees only IPv4
   - No Message Identifiers in the LONG Format of *TSO NETSTAT*

LONG Format = IPv6 or IPv4

```
D TCPIP,TCPIPT,N,HOME,FORMAT=LONG
EZD0101I NETSTAT CS V1R10 TCPIPT 034
HOME ADDRESS LIST:
LINKNAME:    VLINK1
  ADDRESS:   192.168.20.102
    FLAGS:   PRIMARY
LINKNAME:    LGIG1F
  ADDRESS:   192.168.20.92
    FLAGS:
…
LINKNAME:    LOOPBACK
  ADDRESS:   127.0.0.1
    FLAGS:
LINKNAME:    LOOPBACK6
  ADDRESS:   ::1
    FLAGS:
7 OF 7 RECORDS DISPLAYED
END OF THE REPORT
```

SHORT Format = IPv4 only

```
D TCPIP,TCPIPT,N,HOME
EZZ2500I NETSTAT CS V1R10 TCPIPT 021
HOME ADDRESS LIST:
ADDRESS           LINK              FLG
192.168.20.102    VLINK1            P
192.168.20.92     LGIG1F
10.1.1.2          EZASAMEMVS
…
127.0.0.1         LOOPBACK
6 OF 6 RECORDS DISPLAYED
END OF THE REPORT
```

Otherwise: IPv6 Usage is Transparent

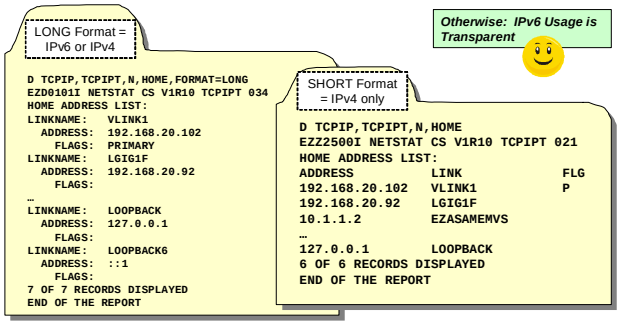
1. Even if you are not yet thinking of implementing an Ensemble Network, you should think about starting to use the LONG FORMAT of the NETSTAT OUTPUT in z/OS. You can define this as a default in the IPCONFIG statement of the TCP/IP Profile.
   1. The benefit is that you are positioning yourself for the change in the z/OS NETSTAT output displays which will occur once you enable the z/OS stack for dual-mode (i.e., IPv4 and IPv6). Once you implement IPv6 in zOS, the LONG format of the display is the only one available.
2. Although INMN uses IPv6, the IPv6 usage in the INMN network is virtually transparent. IPv6 exploitation is unnecessary to create the Ensemble Environment.
3. You do not have to “learn” IPv6 to participate in an Ensemble. You only have to enable the stack to use IPv6, and there are only two changes you must make to do so: Implement IPv6 using definitions in SYS1.PARMLIB(BPXPRMnn) and then optionally change any automated processes you may have to issue and interpret NETSTAT commands to utilize the LONG format of the command. The FORMAT LONG is used to support longer IPv6 addresses. Therefore, LONG FORMAT is always used when IPv6 is enabled. FORMAT SHORT is not supported when IPv6 is enabled. FORMAT can be defined in the IPCONFIG statement of a z/OS TCP/IP stack and thus cause all netstat commands to adopt the LONG format in an IPv4-enabled implementation.
4. Most Netstat Output in a TCP/IP Stack on z/OS that is not enabled for IPv6 can be displayed with either the LONG or the SHORT format. If you have automated operations that are triggered by NETSTAT Short Format messages under TSO (and TSO only), be aware of the fact that NETSTAT in LONG format does not produce Message Identifiers. Under TSO, Netstat output displays in IPv4 can contain messages with the Prefix “EZZ” as with EZZ2761I. These “EZZ” messages do not appear with other forms of the Netstat output, as under UNIX or with the MVS “D TCPIP” variants of the Netstat command. Note that the Message Identifiers under TSO are displayed if the TSO user ID profiles are set to the value PROFILE MSGID and if the TCP/IP stack not enabled for IPv6 processing.
5. NOTE on z/VM: z/VM handles the INMN requirement for IPv6 differently. VM only supports IPv6 on a layer 2 mode Virtual Switch. The main TCP/IP stack does not have to talk IPv6 to be part of an ensemble. Z/VM has another internal z/VM Stack that is used for OSM connectivity. The customer does not use this stack. The netstat output for the IEDN is thus either IPv4 or IPv6, depending on whether the main stack is communicating using IPv6 or not.The customer will not have to configure this stack when Ensemble Managed.

Simple Migration Plan for INMN IPv6 Requirement

- **In preparation for INMN OSA port interfaces:**
  1. **Enforce policy that all NETSTAT commands be executed in the LONG format**
     1. "d tcpip,,n,home,format=long" -- or
     2. Change IPCONFIG in TCP/IP stack to force FORMAT LONG
     3. Create, Execute test plan for this simple change
     4. Rewrite any scripts that cannot accommodate FORMAT LONG
  2. **Eliminate use of NETSTAT GATE command**
     1. Substitute NETSTAT ROUTE for all instances of NETSTAT GATE
     2. Create, Execute test plan for this simple change
  3. **Enable MVS to support IPv6 (Change to hlq.PARMLIB(BPXPRMxx)**
     1. Create, Execute test plan for this simple change

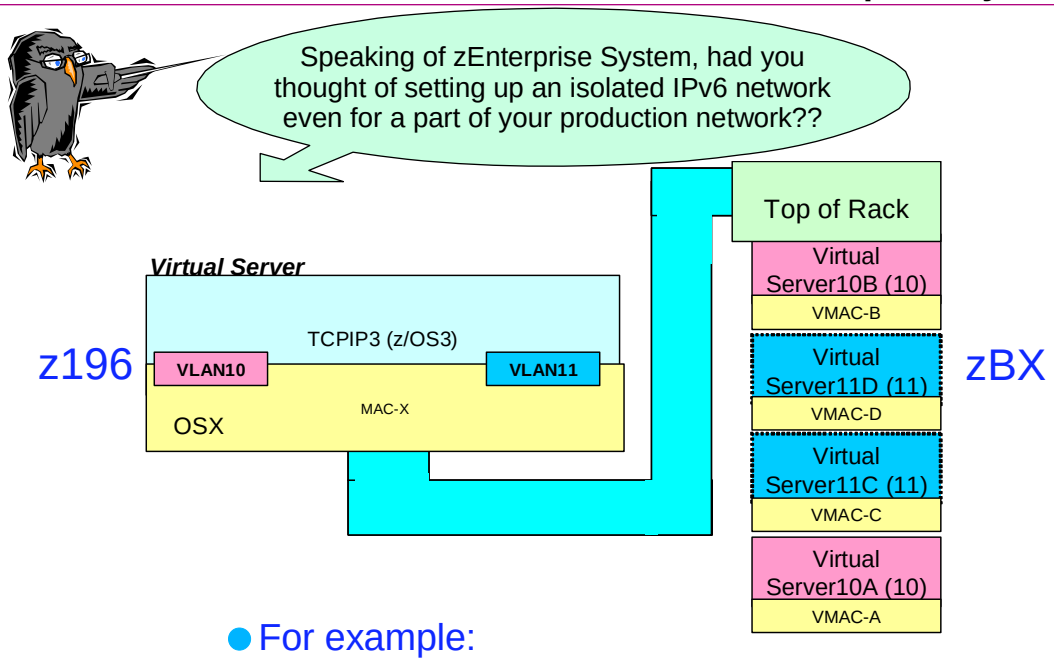
Effect of the IPv6 Addressing on a Participating z/OS Stack

1. Change to BPXPRMxx (UNIX member in PARMLIB)
2. Change to NETSTAT output
   - LONG for IPv6 (or mixed) output (SHORT not supported when IPv6 enabled.)
   - Must use NETSTAT ROUTE to see an IPv6 route
     - and not NETSTAT GATE, which sees only IPv4
   - No Message Identifiers in the LONG Format of *TSO NETSTAT*



1. Note that “NETSTAT” is not regulated by standards; as a result each vendor platform can have a different implementation of the output displays for any of the netstat commands and options. Although you may have written shell scripts or Rexx programs to reformat output displays from a netstat command, every release of software or upgrade of an operating system can introduce changes that cause you to revisit your installation’s customized scripts to process netstat output. In other words, you are already familiar with this process of testing and possibly changing your customized scripts with every new release; the enablement of IPv6 is just another change that you must anticipate as usual. One of the benefits of the netstat display output on z/OS is that, even with an IPv4 network, you can still choose to begin displaying netstat output using the LONG format. This means, that even before a migration to ensemble networking (which will require IPv6 enablement), you can begin the process of modifying your customized scripts and your automated operations that may have been relying on message identifiers. If you have developed REXX programs that issue Netstat commands under TSO and parse the output lines based on message identifiers, you may need to change those REXX programs to use some other token in the output lines to decide the format of the line you are trying to parse.
2. NOTE on z/VM: z/VM handles the INMN requirement for IPv6 differently. VM only supports IPv6 on a layer 2 mode Virtual Switch. The main TCP/IP stack does not have to talk IPv6 to be part of an ensemble. Z/VM has another internal z/VM Stack that is used for OSM connectivity. The customer does not use this stack. The netstat output for the IEDN is thus either IPv4 or IPv6, depending on whether the main stack is communicating using IPv6 or not.The customer will not have to configure this stack when Ensemble Managed.

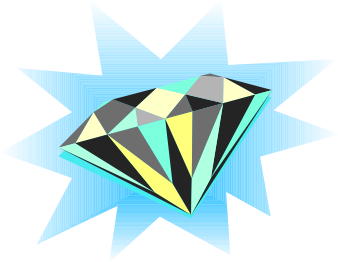
Create an Isolated TEST VLAN for IPv6 with zEnterprise System!



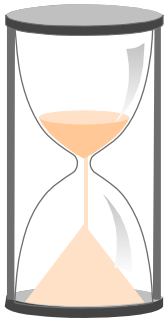
- For example:
  - Make VLAN10 an IPv4 VLAN on the IEDN
  - Make VLAN11 an IPv6 VLAN on the IEDN

1. The zEnteprise System with its segmentation possiblilities via VLAN gives you the perfect opportunity to test or actually put into production pieces of your network using IPv6 protocols -- IPv6 could be in the Virtual Servers of the z196 node or in the zBX blades or in both -- all separated even from IPv4 by means of separate VLAN IDs, as our visual suggests.

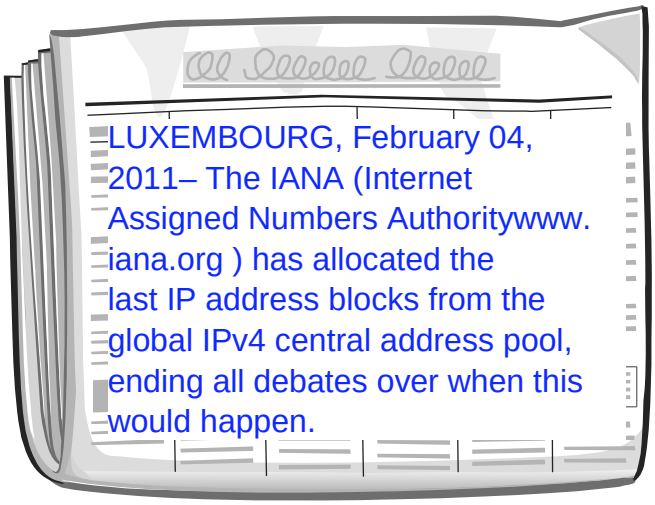
Gems for the Inevitable IPv6 Implementation



US Office of Management and Budget: 1st Phase of IPv6 = 2012

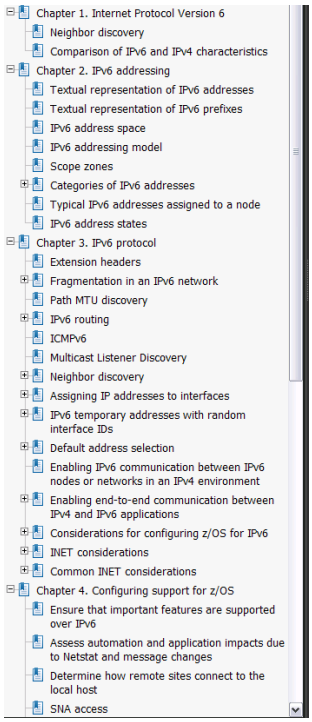


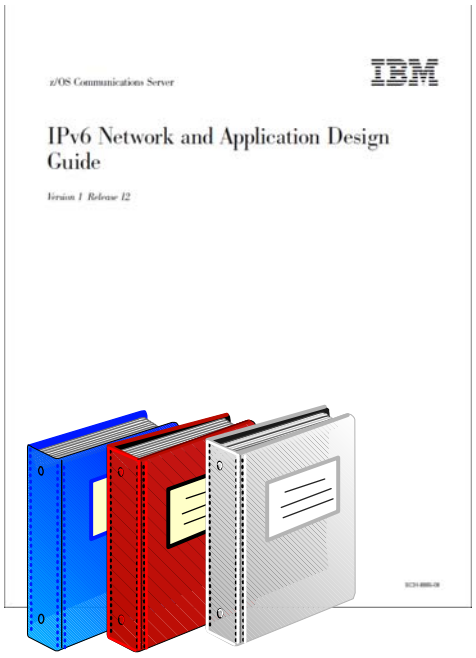
- Time is running out:
  - February 2011: IANA runs out of IPv4 Addresses
  - August 2011: Regional Internet Registries projected to run out.



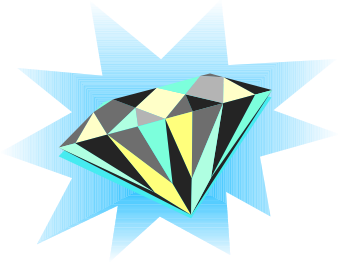
1. The Business Case and Roadmap for Completing IPv6 Adoption in US Government
2. The IPv6 Working Group of the Federal CIO Council has published a draft for review and comment. This document is intended for chief information officers (CIOs), chief architects, and other individuals in federal agencies who are responsible for exploiting information technology (IT) assets to assist in achieving the mission and objectives of the agency. The purpose of this document is to aid in understanding the Federal Government's Internet Protocol version 6 (IPv6) vision and to provide specific guidance for adopting this protocol.
   http://www.whitehouse.gov/omb/egov/documents/DRAFT_Business_Case_&_Roadmap_for_Completing_IPv6_Adoption_in_USG_12242008.pdf
3. Request for Review and Comment
   1. Please submit your comments on the draft IPv6 Business Case and Roadmap using the comment template below and to the right by COB Monday, January 19, 2009 to the FEA mailbox at fea{at}omb.eop.gov
   2. http://www.whitehouse.gov/omb/egov/a-2-EAIPv6.html
   3.
4. In order to facilitate timely and effective IPv6 adoption, agencies shall:
   1. Upgrade public/external facing servers and services (e.g. web, email, DNS, ISP services, etc) to operationally use native IPv6 by the end of FY 20121;
   2. Upgrade internal client applications that communicate with public Internet servers and supporting enterprise networks to operationally use native IPv6 by the end of FY 2014;
   3. Designate an IPv6 Transition Manager and submit their name, title, and contact information to IPv6@omb.eop.gov by October 30, 2010. The IPv6 Transition Manager is to serve as the person responsible for leading the agency's IPv6 transition activities, and liaison with the wider Federal IPv6 effort as necessary; and,
   4. Ensure agency procurements of networked IT comply with FAR requirements for use of the
   5. USGv6 Profile and Test Program for the completeness and quality of their IPv6 capabilities.

How to Learn how to Design & Implement IPv6 Networks





Gems in Netstat



APPLDATA on the NETSTAT Command (V1R9, V1R10)

From MVS Console, TSO, UNIX

- **Display TCPIP,,Netstat,ALLConn,APPLDATA<,filter>**
- **Display TCPIP,,Netstat,COnn,APPLDATA<,filter>**
  - Includes APPLDATA in report if present
- **Display TCPIP,,Netstat,ALLConn,APPLD=xx?xx***
- **Display TCPIP,,Netstat,COnn,APPLD=xx?xx***
  - Includes APPLDATA in report
  - Limited to connections with matching APPLDATA
    - Case insensitive search
    - Wildcards are supported:
      - ? Exactly one arbitrary character
      - * Zero or more arbitrary characters

V1R9: Exploited by CICS Sockets and TN3270
V1R10: Exploited by FTP Client and Server

1. APPLDATA is available for TCP applications if they are instrumented to exploit it. Network Management Interface applications may also choose to exploit this capability.
2. Applications may place non-printable characters in the string. Netstat will display them as '.'. Only printable characters may be entered in Netstat filters. Nonprintable characters must be skipped over with wild card characters in the filter.
3. This support was rolled back to V1R7 and V1R8 at the request of other IBM applications that are interested in exploiting it.
4. For CICS, the support provides the ability to:
   1. identify TCP connections for IP CICS Socket applications:
      1. Listener, child server, and client transactions
5. IOCTL or ioctl()
6. z/OS TCP APIs supported
   1. ┐ Macro – EZASMI
      1. • Assembler programs
   2. ┐ CALL instruction – EZASOKET
      1. • Assembler, COBOL or PL/1 programs
         1. – Batch, CICS or IMS applications
   3. ┐ IP CICS C socket library stubs
      1. • C programs
   4. ┐ IP REXX Socket library
      1. • EXECs

Configurable Maximum for D TCPIP,,NETSTAT MVS (V1R10)

- **Provide a configurable maximum for records displayed by a D TCPIP,,NETSTAT MVS console command**
  - Remember that the maximum value denotes number of records, not number of lines written to the console - these six lines count as two records:

```
SNTPD    0000001B UDP
  LOCAL SOCKET:   0.0.0.0..123
  FOREIGN SOCKET: *..*
SNTPD    0000001C UDP
  LOCAL SOCKET:   ::..123 (IPV6_ONLY)
  FOREIGN SOCKET: *..*
```

  - Default maximum remains 100
  - Can be changed via new GLOBALCONFIG MAXRECS statement
    - Maximum can either be * - no maximum
    - Any value between 1 and 65535
  - If the number of lines displayed as the result of a D TCPIP,,NETSTAT console command exceeds 65535 before MAXRECS is hit, an error message will be issued (instead of an abend D23)

NETSTAT ALL Is Available on MVS Console (V1R10)

- **With this new configuration support available, we add**
  - **D TCPIP,,NETSTAT,ALL** MVS console command support for the NETSTAT ALL report
  - This report can produce significant amounts of output if it is used without filters

```
                    .--MAXRECS 100 -----.
>>-GLOBALCONFig----+-------------------+------------><
                    '-+-MAXRECS * ----+-'
                      '-MAXRECS recs -'
```

- **Provide a configurable maximum for records displayed by a D TCPIP,,NETSTAT MVS console command**
  - Can be changed via new GLOBALCONFIG MAXRECS statement
    - Maximum can either be * - no maximum
    - Any value between 1 and 65535
  - If the number of lines displayed as the result of a D TCPIP,,NETSTAT console command exceeds 65535 before MAXRECS it hit, an error message will be issued (instead of an abend D23)

1. If you use automation programs which process MVS operator command output, and you want these programs to process detailed TCP connection and UDP endpoint data, you can update the programs to invoke the DISPLAY TCPIP,,NETSTAT command with the ALL option.
2. You should also update the programs to detect the new report output line which indicates that the report has been truncated:
   1. REPORT TRUNCATED DUE TO GREATER THAN 65533 LINES OF OUTPUT
3. By checking for this output line, the program will know when the report output is incomplete.

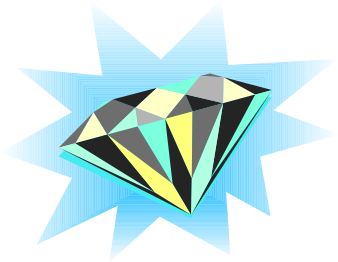
Netstat Dev or Netstat Home: INTFNAME Filter (V1R10)

```
Home address list:
Address         Link         Flg
-------         ----         ---
192.168.115.5   OSAQDIOLINK  P
192.168.113.11  TR1
201.2.10.31     VIPLC9020A1F I
127.0.0.1       LOOPBACK

Address         Interface    Flg
-------         ---------    ---
192.168.125.5      OSAQDIOINTF
```

- INTFNAME may specify the OSA Portname!
  - You can see all associated INTERFACES connected to the same PORT.

1. This is an example of the Netstat HOme/-h report from an IPv4-only TCP/IP stack.
2. This report displays the IPv4 home addresses defined with an INTERFACE statement separately from the others.

Gems with Routing



INCLUDE Statement for OMPROUTE Configuration File (V1R10)

```
AREA
AREA_NUMBER=1.1.1.1
STUB_AREA=NO;

INCLUDE /u/user1/omproute.conf
INCLUDE //'USER1.INC10'
Include //'USER1.&SYSNAME..OMP'

OSPF_INTERFACE
IP_ADDRESS=10.9.128.128
NAME=DUMMY_SASRVA2
SUBNET_MASK=255.255.255.240
ROUTER_PRIORITY=0
ATTACHES_TO_AREA=1.1.1.1;
```

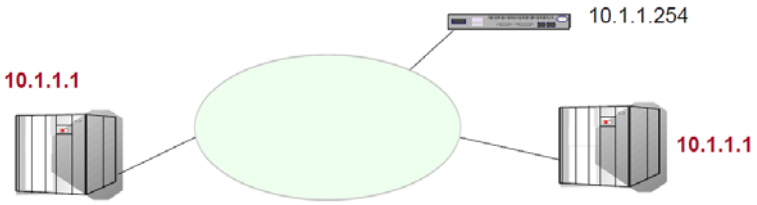
- **OMPROUTE "Include file":**
  - Easier to share common OMPROUTE definitions within a Sysplex

```
>>__ _____________________________________________________><
   |_Include__ _//'fully qualified MVS dataset name'_ _|
             |_/file system absolute pathname_______|
```


1. Common configuration statements can be grouped into separate files and specified in the OMPROUTE configuration via the INCLUDE statement
2. Single, multiple, and nested INCLUDE statements can be used in configuring OMPROUTE
3. Rules:
   1. 1. INCLUDE statement must be the only configuration statement on the line.
   2. 2. INCLUDE statement must not end with semicolon.
   3. 3. There must be no more than 10 nested INCLUDE statements.
   4. 4. Static system symbols can be specified as part of the data set name.
   5. 5. Only 1 INCLUDE statement can be specified per line, anything else that follows the statement will be ignored.
4. If a syntax error is encountered in the final version of the configuration file after INCLUDE file(s) were processed, use debug level d1 to print a copy of the expanded configuration file to your OMPROUTE trace.

OSPF Detection of Duplicate RouterID (V1R11)

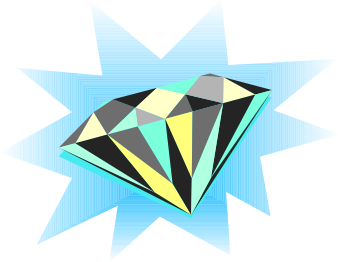
EZZ8165I DUPLICATE IPV4 OSPF ROUTER ID 10.2.3.4 DETECTED

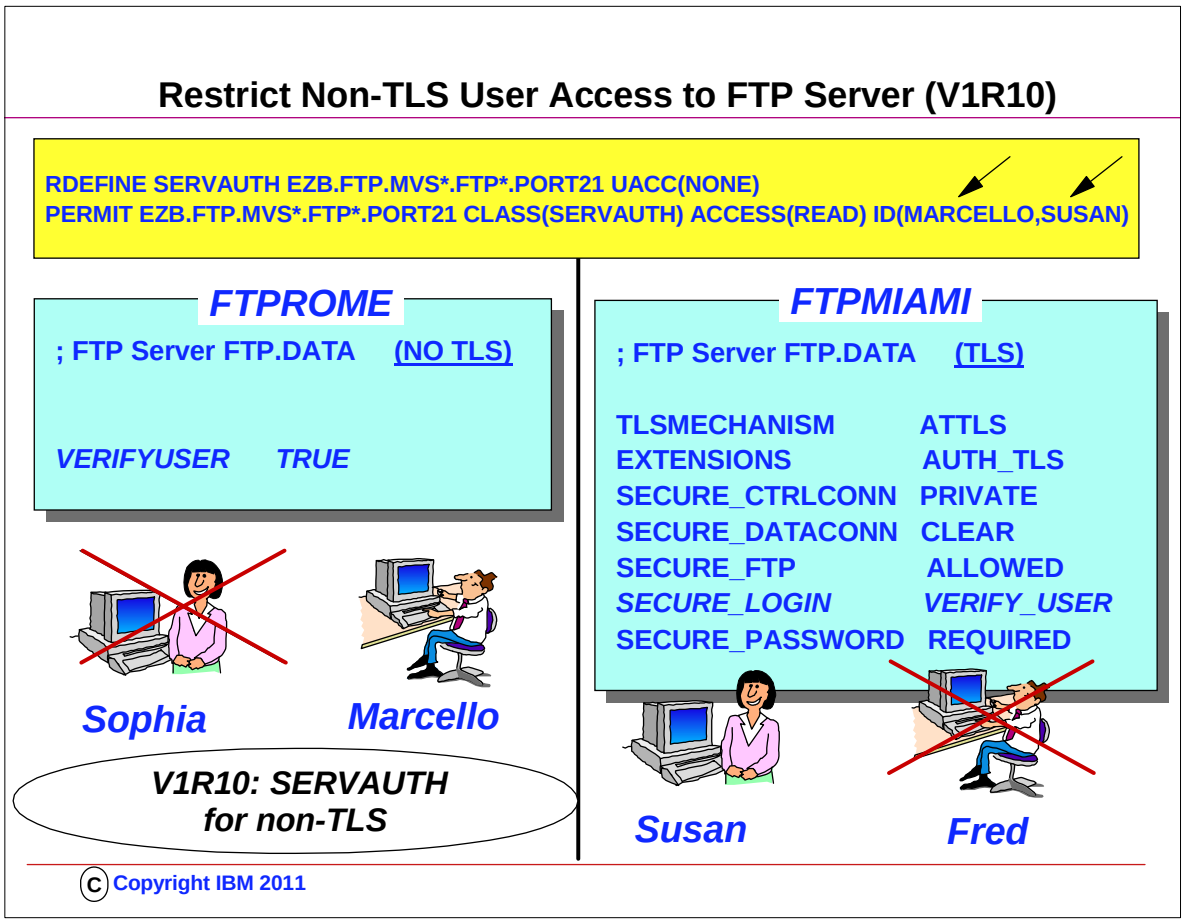


- If multiple OSPF routers use the same router ID, routing problems will occur:
  - Routes are continuously added and deleted by neighboring routers
  - Increased OSPF traffic as designated router floods new LSAs
  - Packet loss or connectivity loss depending upon routing environment
  - Problem can be difficult to diagnose due to varied symptoms
- New Console Message to identify the situation, allowing for correction by System z Networking System Programmer.

1. Although router IDs should be unique, sometimes multiple OSPF routers are configured with the same router ID. This will cause routing problems.
2. The designated router is getting router LSAs from each router with different information. The router will update its routing table and then flood the updated LSAs to other routers in the area.
3. This will cause routes to cycle from active to non-active states, or be constantly added then deleted from the network topography, causing excessive network disruption. Packets can be lost in a routing loop or dropped as these routes consistently change. This can cause intermittent ping timeouts or poor performance on connections. The symptoms will stop if the duplicate router is stopped. This type of problem can be difficult to diagnose.
4. The picture shows three OSPF routers, however two of them are using the same router ID.
5. In V1R11 OMPROUTE will issue message EZZ8165I when OSPF packets are received from a adjacent router with the same router ID OMPROUTE is using. EZZ8165I is issued to the console once every 10 minutes per OSPF version. So, if a router is using the same router ID for both IPv4 and IPv6 OSPF, message EZZ8165I is issued twice.
6. Message EZZ8165I only detects this situation has occurred. Unfortunately, OMPROUTE can't resolve this problem dynamically. The first step is to verify the router ID being used by this OMPROUTE is correct. If the router ID in message EZZ8165I is not the expected router ID, the configuration needs to be verified. OMPROUTE should be configured with a router ID, so the same router ID is used by this OMPROUTE instance. The router ID should not be a DVIPA address, as this address can be active on multiple TCPIP stacks. Message EZZ8134I should have been issue when OMPROUTE started if a DVIPA address had been used. If the router ID in message EZZ8165I is correct for OMPROUTE, someone else in the OSPF autonomous system is incorrectly using the router ID. The designated router should be checked first, using neighbor displays. You are trying to correlate the router ID with an interface address to determine which router is incorrectly using the router ID. A packet trace or sniffer trace can also be used to find the IP address. Once the router has been identified, the router can be configured with the correct router ID.

Gems with FTP





1. By default, any user ID that is valid on the z/OS host can log into FTP. For security purposes, a customer may want to allow only certain user IDs to log into FTP on a certain host. z/OS FTP currently provides two ways to do this:
   1. you can code and install the FTCHKPWD exit routine, or
   2. you can configure TLS level 3 client authentication.
2. The FTCHKPWD exit routine is code written by you which is invoked by the FTP server as part of validating the user ID used to log into FTP. The sample FTCHKPWD in SEZAINST shows one method of using an exit routine to control which user IDs are allowed to log into the FTP server.
3. TLS level 3 client authentication adds a Security Access Facility (SAF) profile check to TLS level 2 client authentication. After configuring TLS level 2 client authentication, you can define a server port profile in the SERVAUTH class, and grant READ access to those user IDs you want to allow to log into the FTP server. FTP will verify each user ID logging in with TLS has at least READ access to the profile.
4. If users log on using SSL/TLS with Client Authentication and the SECURE_LOGIN option is set to VERIFY_USER, the FTP server will check if the user has READ access to EZB.FTP.<systemname>.<ftpdaemonname>.PORTxxxx SERVAUTH resource. Client Authentication requires that the user present an x.509 client certificate.
5. If users do not use SSL/TLS or the VERIFY_USER option isn't set as above, no checking of the SERVAUTH resource is done prior to V1R10
6. Now with V1R10 we are giving installations an easy way to limit use of the FTP server functions in general without requiring TLS with Client Authentication:
   1. Define the EZB.FTP.<systemname>.<ftpdaemonname>.PORTxxxx SERVAUTH SERVAUTH resource with universal access set to NONE
   2. Permit those users who are allowed to use the FTP server with READ access to the SERVAUTH resource
7. Note that if you code this statement in the server FTP.DATA: SECURE_LOGIN VERIFY_USER
   1. And the session is TLS secured, FTP ignores the VERIFYUSER value and checks the server port profile before allowing the login.
8. In the examples shown, Sophia is not allowed access to the non-TLS FTP server because of the SERVAUTH definitions depicted. Fred is not allowed access to the FTP server even if he has a client certificate, because he is also not authorized through the SERVAUTH definitions.

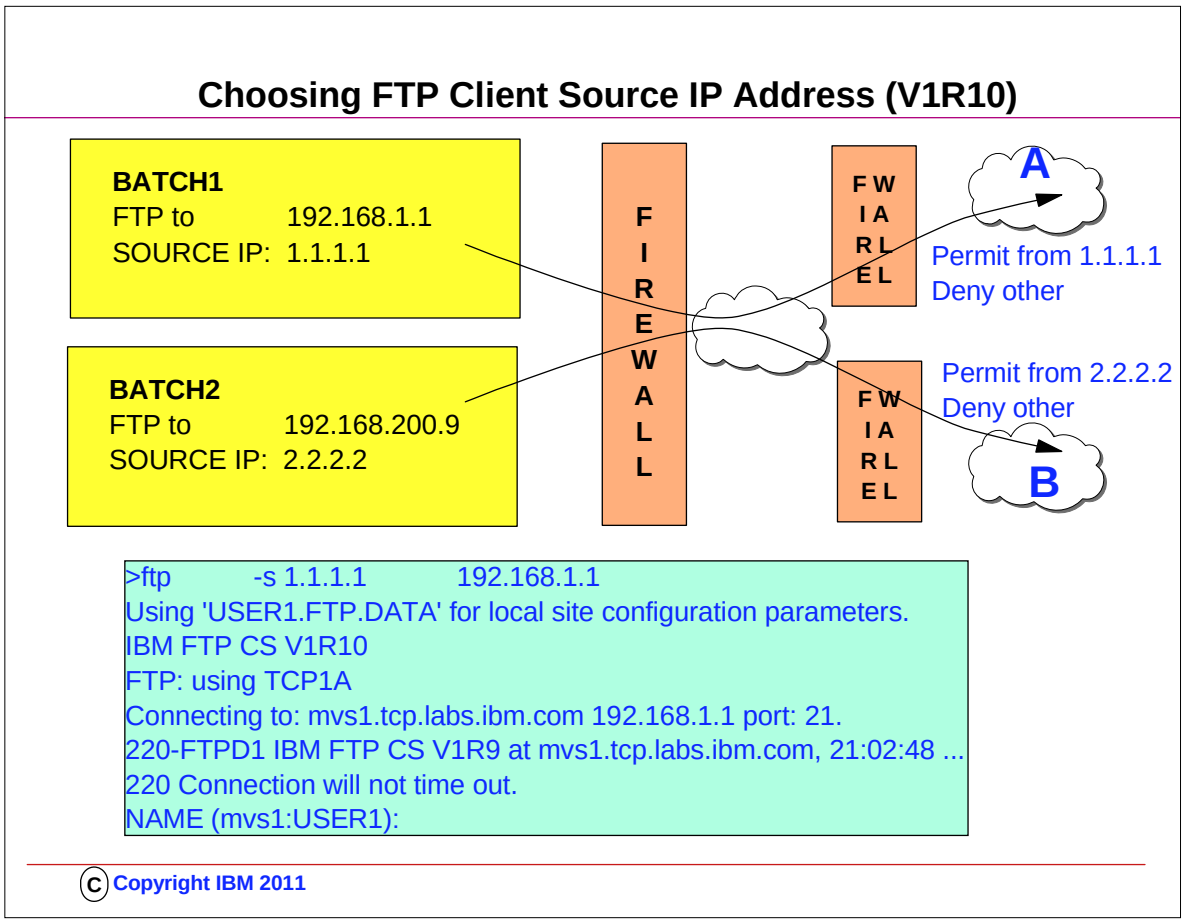
FTP Enhancements: Connection APPLDATA (V1R10)

```
/u/user1 netstat -G EZAFTP*
MVS TCP/IP NETSTAT CS V1R10 TCPIP Name: TCPCS 01:50:14
User Id Conn Local Socket Foreign Socket State
FTPD1 000000BC 1.2.5.36..20 1.2.5.36..1026 Establsh
Application Data: EZAFTP0S D USER2 C PSSS
```

Offset	Description
1 – 8	The string "EZAFTP0S"
9	Blank
10	C for a control connection socket D for a data connection socket
11	Blank
12-20	User ID used to log into FTP
21	Blank
22	Security protection C for Clear; S for Safe; P for private; L for Clear but previously safe or private
…	More fields (see *IP Configuration Reference* for details)

- **z/OS CS V1R9: APPLDATA of 40 characters for a TCP sockets**
  - **exploited by CICS, TN3270**
- **V1R10: APPLDATA**
  - **exploited by FTP Client and Server**

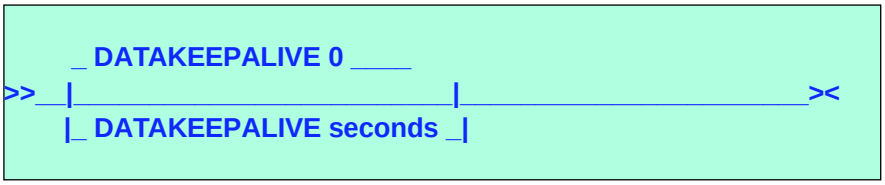
1. This data is kept in the APPLDATA field of the socket. It can be set or updated by a TCP application using an IOCTL sockets call and can be included in NETSTAT ALL, ALLCONN, and CONN reports and used as a filter. This data also can be included in the NMI network monitor interface. The suggested syntax for the field is to use an eight-character application identifier in the first 8 characters of the 40-character APPLDATA field
2. In V1R9, this support is used by CICS Sockets to associate CICS-specific information with CICS sockets endpoints
3. For example: EZACICSO SRV1 0000123 USER1234 CICA
4. Also in V1R9, it is used by the TN3270 server to associate TN3270-specific information with TN3270 sockets endpoints
5. For example: EZBTNSRV TCPABC80 TSO10001 ET B
6. Now in V1R10, both the FTP client and the FTP server will associate FTP-specific information with the FTP sockets endpoints:
   1. FTP component (Client, Server, Daemon)
   2. Type of connection (Control or Data)
   3. User ID
   4. Security characteristics (SSL/TLS, GSSAPI, Ciphers, etc.)
   5. Info about file being transferred (direction, type, location)
7. z/OS V1R9 CS implemented support for TCP applications to associate up to 40 characters of application-specific data with a TCP socket:
   1. Can be set or updated by a TCP application using an IOCTL sockets call
   2. Is included in NETSTAT ALL, ALLCONN, and CONN reports and used as a filter
   3. Is included in the NMI network monitor interface
   4. Suggested format for the field is to use an eight-byte application identifier in the first 8 characters of the 40-character APPLDATA field
8. The string 'PSSS' for the server data socket is part of the other fields mentioned in the table. It indicates a PORT command established the connection; the transfer was inbound to the server, the file type was SEQ, and the file location was a sequential MVS data set.



1. The TCP/IP stack determines the source IP address. This can be based on TCP/IP configuration options such as Job-Specific Source IP or it may be determined when the route to the FTP server is found.
2. In some situations the FTP client may want to use a different source IP address when connecting to different FTP servers.
3. In firewall configurations, it may be necessary to use a specific source IP address for the firewall to allow the connection.
4. But, there is no way for the FTP client, itself, to specify the source IP address that should be used.
5. This diagram shows an example of when the FTP client may want to specify the source IP address.
6. In the diagram, the customer has a network setup where the z/OS system running the FTP client has two interfaces into the network.
7. The customer needs to be able to FTP into two other networks which are protected by firewalls. The firewalls are configured to only allow connections from specific IP addresses.
8. So the only way to successfully FTP into “Customer A network”, is to use a source IP address of 1.1.1.1 or into Customer B network is to use a source IP address of 2.2.2.2.
9. Since there is no way for the FTP client to specify a source IP address without this V1R9 improvement, there is no guarantee that the TCP/IP stack would choose the correct interface UNLESS you implement the SRCIP block in an appropriate manner.
10. Since there are two interfaces into the network the TCP/IP stack may choose either interface.

FTP KeepAlive (V1R10)

- **FTP.DATA for both server and client**
  - Use the DATAKEEPALIVE statement to define the data connection keepalive timer.



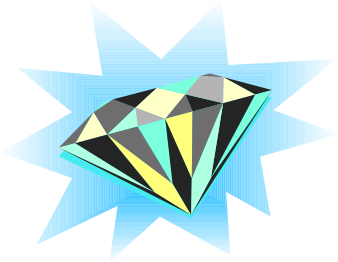
SITE DATAKEEPALIVE=xxx

LOCSITE DATAKEEPALIVE=xxx

- **DATAKEEPALIVE Parameters**
  - seconds The number of seconds of inactivity before a keepalive packet is sent out on the FTP data connection. The valid range is 0 (not used) through 86400 (24 hours). The default is 0.
- **Usage Notes**
  - Specify 0 to use the keepalive interval specified in the TCP/IP stack.
  - Specify 86400 to prevent any keepalive packet from being sent

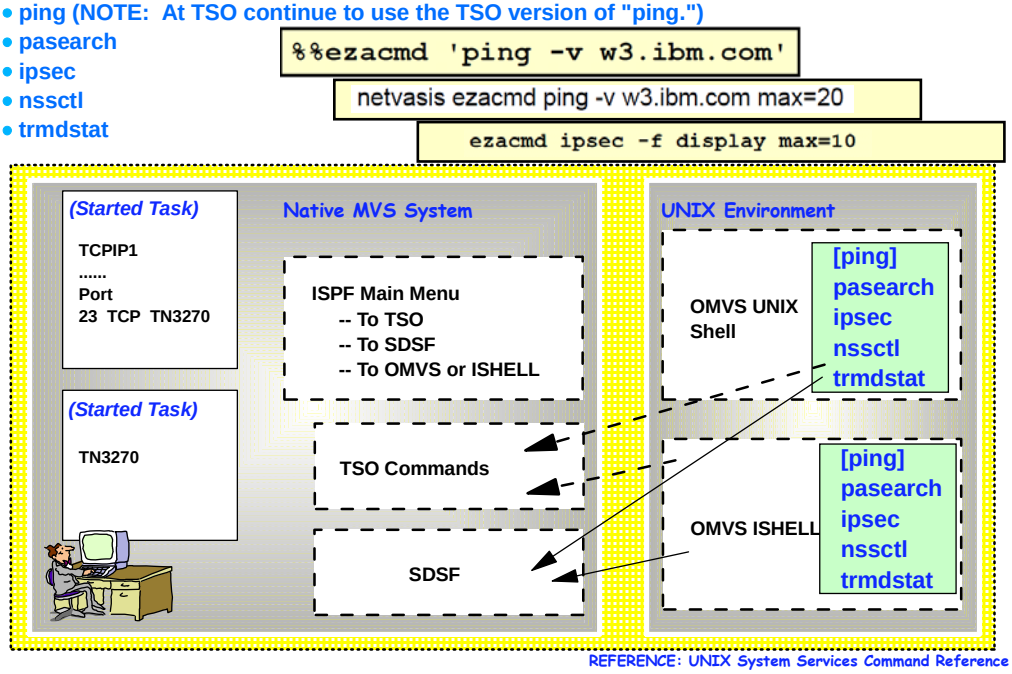
1. Any TCP/IP connection is subject to monitoring by the network. The session may be canceled if no activity on the session is detected within a defined period of time as determined by the network device. Cancelling the session prevents any further communication between the session partners. In cancelling the session, the session partners may not be notified of this cancellation which can result in a hung session if the session partners do not provide for this situation.
2. To prevent cancellation, TCP/IP sends keepalive packets. A keepalive packet contains one byte of data and uses a sequence number of a packet that was already sent.
3. The remote session partner discards the data packet because they have already received the packet.
4. The benefit is that any device monitoring the session will detect that the session is active.
5. The FTP control connection is the connection over which FTP commands are sent from the client to the server. The keepalive interval can be customized by the KEEPALIVE statement in the FTP.DATA configuration file instead of utilizing the TCP/IP stack's keepalive interval.
6. The FTP data connection, over which file data flows during a file transfer between the client and server, does not support any customization of the keepalive interval.
7. This session is susceptible to being cancelled if the connection stays idle too long. A long running DB/2 query or a job submitted to JES that has not completed can cause this to occur.
8. While the TCP/IP stack provides the ability to configure when keepalive packets are generated, this interval may exceed that needed by FTP.
9. Without the ability to customize a keepalive interval on the FTP data connection, FTP must rely on the keepalive interval defined to the stack.
10. The FTP connection may be monitored by a device whose cancellation timer is less than the stack keepalive timer.
11. Each network has its specific needs and a single stack keepalive interval may not be able to cover all of these networks.
12. The value may be set through the SITE and LOCSITE commands as well.
13. **PASSIVE MODE:** When logging in from a non-z/OS FTP client to a z/OS FTP server and using passive mode, the SITE command is not supported by a non-z/OS client. Use the QUOTE SITE subcommand to have the z/OS FTP server initiate keepalive packets to keep the data connection from being cancelled because of inactivity.

Gems with TN3270 and Telnet ("otelnetd")



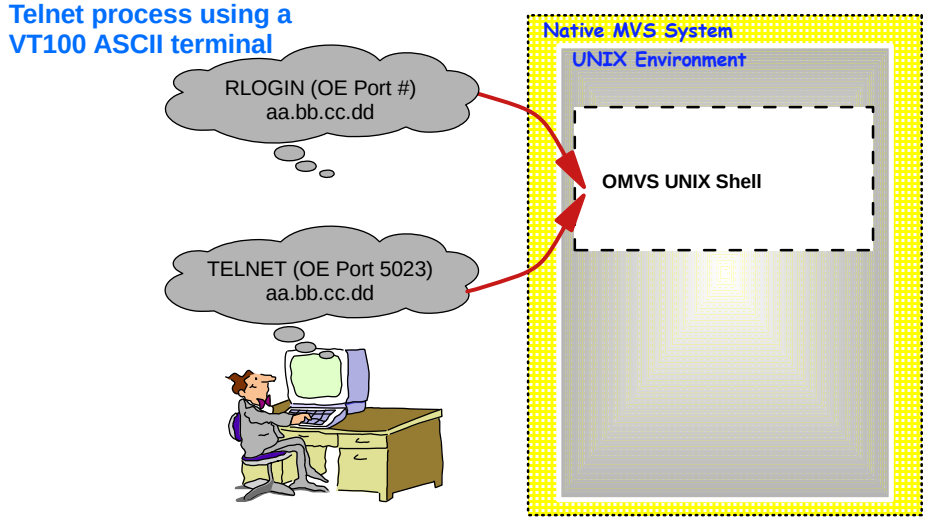
Formerly UNIX-only Commands Now at MVS Console (V1R11)

- Following Commands that were formerly UNIX-only, may now be executed at the MVS Console, from NetView, and from TSO using the EZACMD interface:
  - ping (NOTE: At TSO continue to use the TSO version of "ping.")
  - pasearch
  - ipsec
  - nssctl
  - trmdstat



1. z/OS V1R11 makes the z/OS UNIX commands listed above available in the three new command environments:
   1. z/OS console, NetView, and TSO. Only the z/OS UNIX commands listed above are made available in these environments.
2. Ping is not a policy-related command, but customers have asked for ping from the z/OS console for many years.
3. The infrastructure that was built for the policy-related commands was very easily expanded to also support ping.
4. Since TSO has a native TSO ping command already, the z/OS UNIX ping was not made available in TSO.
5. The command examples for PING show you, in sequence, the syntax for: MVS Console, NetView, TSO.
6. EZACMD is a generalized interface to the selected set of z/OS UNIX commands. The same EZACMD command is used from TSO, the z/OS console, and NetView. Each environment has specific requirements and characteristics, which you should read about in the IP Configuration Guide. For setting up z/OS System REXX in general for the MVS Console and for TSO, refer to the z/OS publication "MVS Programming: Authorized Assembler Services Guide", chapter 31 "System REXX" and z/OS "MVS Initialization and Tuning Reference", Chapter 8 "AXR00 (default System REXX data set concatenation)".
   1. EZACMD should be copied into the REXX libraries used by TSO and by NetView for its use there.
7. The EZACMD supports UNIX commands. Command options are case sensitive and must be entered exactly as documented for the z/OS UNIX command in question.
   1. The MAX keyword can be entered in any case and it may be present anywhere after the command-name.
   2. Output from the commands is displayed as-is. Some commands in some of the supported environments will produce output lines that are too long for the display environment. Such long output lines will be folded onto the following line. No attempt is made to re-format the output from the existing z/OS UNIX commands.
8. EZACMD is delivered as a compiled REXX program in two different system libraries. One library is SYS1.SAXREXEC, which is the system REXX system library. This is a VB, LRECL=255 library. The second library is tcpip.SEZAEXEC, which is the z/OS Communications Server REXX library. This is an FB, LRECL=80 library.
9. SYS1.SAXREXEC is used from the z/OS console by means of the system REXX infrastructure, which requires a VB, 255 library.
   1. tcpip.SEZAEXEC is used from TSO and NetView.
10. Remember System REXX requires that all REXX libraries used by System REXX are VB, LRECL=255
    1. – TSO and NetView might have been set up to use either FB, LRECL=80 or VB, LRECL=255
       1. • SYS1.SAXREXEC is VB, 255
       2. • tcpip.SEZAEXEC is FB, 80
    2. – EZACMD is delivered in both libraries
    3. Consider SERVAUTH profiles for especially the ipsec command usage

Accessing UNIX Shell with UNIX Telnet (Port 23): Banners ...




```
#=====================================================================
# service | socket | protocol | wait/ | user | server  | server program
# name    | type   |          | nowait|      | program |   arguments
#=====================================================================
login    stream tcp nowait OMVSKERN /usr/lpp/tcpip/rlogind rlogind -m
#
otelnet   stream tcp nowait OMVSKERN /usr/lpp/tcpip/sbin/otelnetd -l
# telnet stream tcp nowait OMVS /usr/sbin/otelnetd otelnetd -l -n -h
```

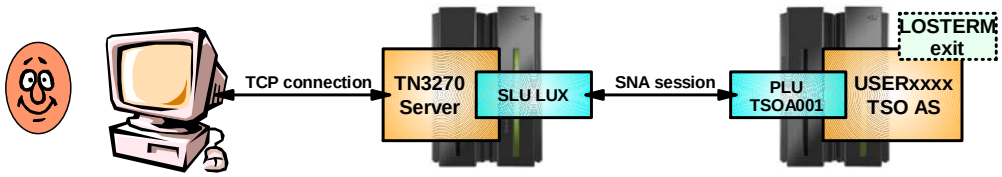

1. You may also implement an ASCII version of TELNET that operates only in a UNIX environment; Telnet (sometimes called "otelnet") also defaults to using Port 23, but many people assign a different port to it, for example we have used Port 5023 here. You use an ASCII terminal or terminal emulator (like the the VT100 emulator) to work in this UNIX environment.
   1. Telnet does not have its own "listener" and so it uses the services of INETD, which listens for connections to Telnet.
   2. UNIX Telnet is implemented by defining it to INETD and then starting INETD as a UNIX process from /etc/rc.
2. You can enter UNIX commands once you have entered the UNIX environment.
3. This visual shows you two methods to enter the UNIX environment:
   1. Use RLOGIN to arrive at the UNIX shell, or
   2. Use OTELNET (TELNET) to the OTELNETD port to arrive at the shell.

Two Telnet Banners: Before and After Login (V1R11)

UNIX Telnet Shell Screen: Display banners or suppress with "-h" initialization

1. This is the type of UNIX shell screen you would see if you used either TELNET or RLOGIN to the UNIX telnet port.
2. The z/OS UNIX Telnet server (otelnetd) provides access to z/OS UNIX shell applications on the host using the Telnet protocol. The z/OS UNIX Telnet server lets hosts in an IP network log on to the z/OS shell environment directly, without going through TSO.
3. Otelnetd provides a customizable banner that is presented to a user after a user logs in. This banner is located in /etc/banner.
4. Customers want the ability to have a banner page presented before a user logs in to otelnetd. They wanted to be able to provide information in this banner, such as which system a user is about to log in to and possibly other information. A new banner to accommodate this requirement was introduced with z/OS V1R11 Communications Server.
   1. The new banner is called /etc/otelnetd.banner. **If the existing –h parameter is specified for otelnetd in /etc/inetd.conf, it now disables the display of both /etc/banner and /etc/otelnetd.banner. An example of how to code this is shown.**

LOGONHERE (V1R11)



If old SNA session exists, when user attempts reconnect, disconnect old SNA session and proceed with TSO logon reconnect.

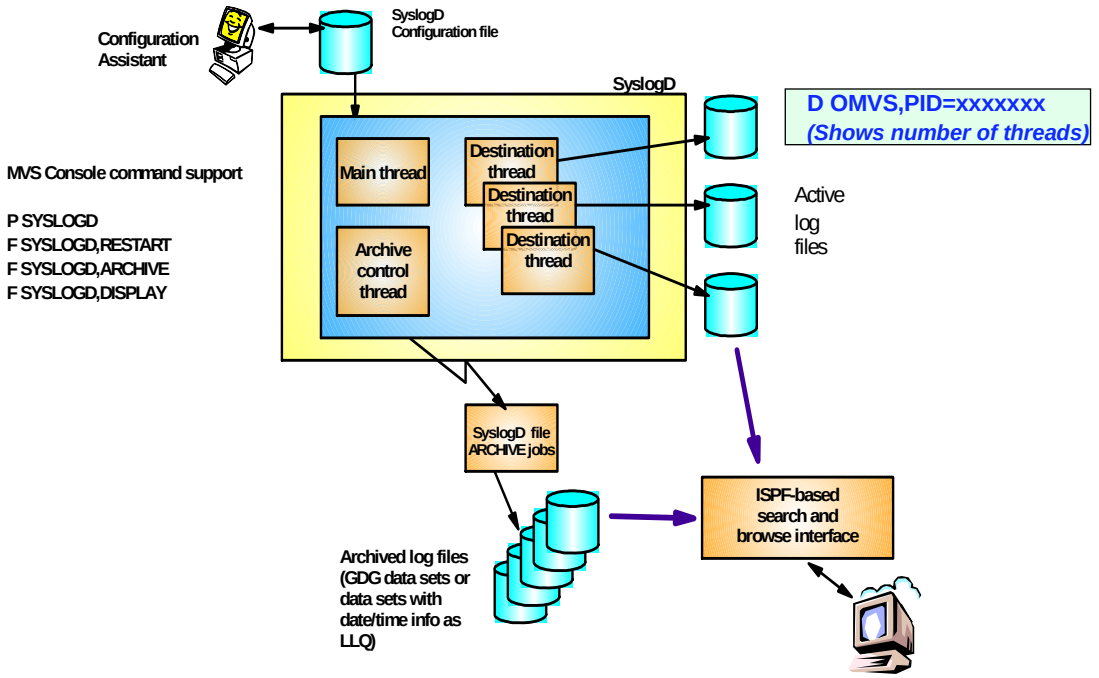
- **Combined effort by TSO and CS development**
- **New LOGONHERE option in IKJTSOxx member to enable new support**
- **Enables reconnecting TSO user from a new SNA session**
- **Helps further reduce number of "USERID already in use" errors**

TSO Reconnect Possible	Single session	Multiple sessions	NATed connecti-vity
TKOGENLU[RECON]	✓		
CheckClientConn	✓	✓	
TKOSPECLU[RECON]	✓	✓	✓
TSO LOGONHERE	✓	✓	✓
TIMEMARK/SCANINTERVAL	✓	✓	✓

1. TSO reconnect has so far been supported in the case where the original SNA session had been disconnected. If an SNA session was still active, it was not possible to use reconnect. In the case where a TN3270 client lost a TCP connection with the TN3270 server, but the SNA session remained active, you could not use reconnect.
2. A modification has been made to TSO. The modification is governed by a new option in IKJTSOxx – a LOGONHERE option. With that option enabled, users can reconnect even when an old SNA session exists. The old SNA session is being disrupted (LOSTERM exit), and reconnect for the new session is processed.
3. This function is an alternative to other TN3270 server options for cleaning up old TCP connections and or SNA session.

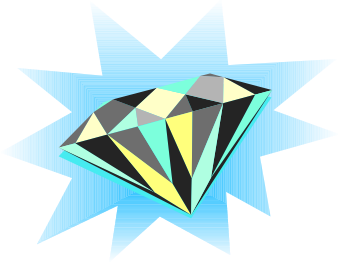
Gems with SYSLOGD

SYSLOGD Management and Performance (V1R11)



1. This slide shows a high-level overview of the new and improved syslogd components.
2. Syslogd is now a multi-threaded implementation allowing for more parallel processing in peak periods. Syslogd continues to write log messages to z/OS UNIX files. A new archive function will archive the content of a z/OS UNIX log file to an MVS data set. The MVS data set can either be a sequential data set (low level qualifiers specify date and time) or a new generation of a generation data group (GDG). The archive operation can be initiated by an operator. At a specific point in time (for example, shortly after midnight). Or when the utilization of one of the file systems the z/OS UNIX log files are written to exceeds a configurable threshold.
3. Command support includes the ability to shut syslogd down using a P command. Syslogd will in R11 not change address space name after it has started. If you start a procedure by the name of SYSLOGD – the resulting address space name remains SYSLOGD.
4. The ISPF browser starts by reading the syslogd configuration file, locates the active z/OS UNIX files, and all available MVS archives. It supports browsing individual files or data sets, in addition to performing extensive searches in one or a series of files or data sets.

Gems with Security



IPSec Standards Compliance (V1R10)

RFC	Department of Defense Advanced UNIX Server Profile	National Institute of Standards and Technology Host Profile	z/OS CS V1R10
2407 ISAKMP DOI	MUST	MUST	✓ (already supported)
2408 ISAKMP	MUST	MUST	✓ (already supported)
2409 IKE	MUST	MUST	✓ (already supported)
3948 UDP-encap ESP	N/A	MAY	✓ (already supported)
4109 IKE algorithms	MUST	MUST	✓ (already supported)
4301 IPsec	MUST	SHOULD+	✓ (new in V1R10)
4302 IP AH	MUST	MAY	✓ (already supported)
4303 IP ESP	MUST	MUST	✓ (already supported)
4304 ESN	SHOULD	MUST	✓ (new in V1R10)
4305 IPsec algorithms	SHOULD+	SHOULD+	✓ (already supported)
4308 Crypto suites	MUST	MAY	✓ (new in V1R10)

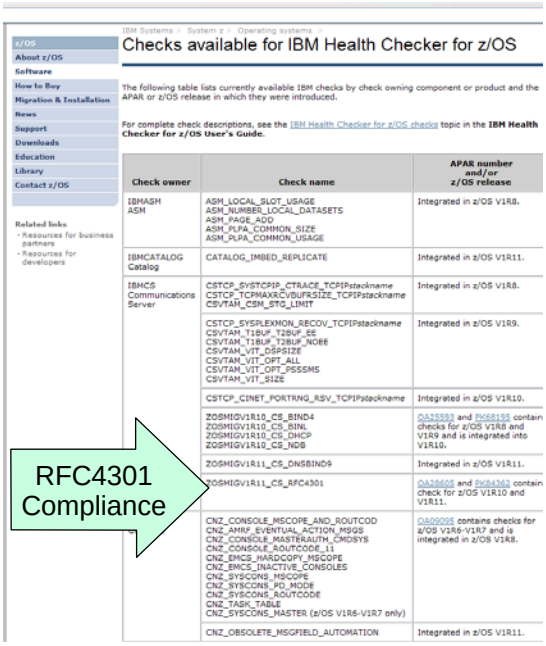
Security mandates can require compliance with standards.
Auditors must check for these on all platforms.

1. z/OS Communications Server V1R10 supports all IPsec RFCs at levels currently required by DOD and NIST profiles.
2. Note that some elements of RFC 4301 require the use of IKEv2; for example, support for dynamic tunnels that cover a range of ports. Since z/OS Communications Server does not support IKEv2, these elements are not supported on z/OS Communications Server.
3. Note that the z/OS Communications Server support for RFC 4304 extends to recognition of ESN proposals during the negotiation of security associations, but not to supporting the use of ESN. z/OS Communications Server IKED will reject an SA proposal that includes ESN. If there are SA acceptable proposals without ESN then z/OS Communications Server IKED will accept them.

Health Checks Available for z/OS Communications Server

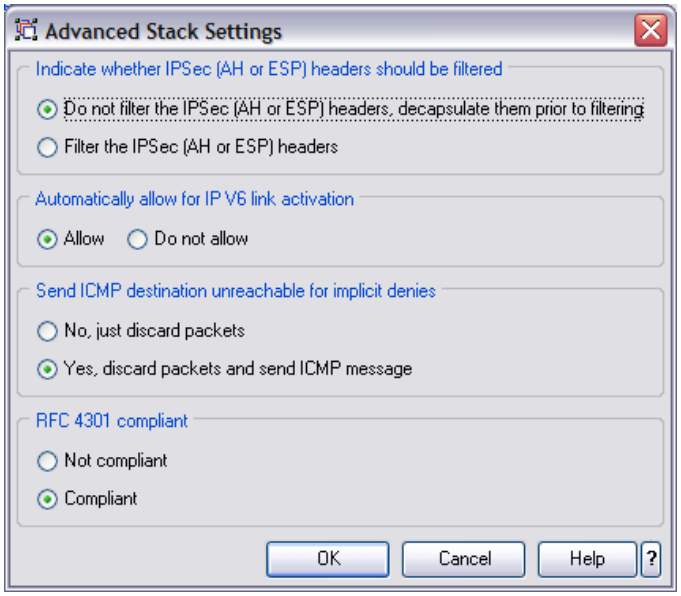
http://www-03.ibm.com/systems/z/os/zos/hchecker/check_table.html

IBM Health Checker for z/OS User's Guide (GA22-7994)



1. You will probably want to download the IBM Health Checker for z/OS User's Guide to investigate how to implement Health Checker and to understand the various types of health checks that are available to you, including those in IBM Communications Server.
2. The User's Guide points you to a web page that is kept updated for all currently available health checks:
   1. http://www-03.ibm.com/systems/z/os/zos/hchecker/check_table.html
3. The web page provides you the name of the RFC4301 health check that you will want your z/OS Systems Programmer to implement for you.

RFC4301 Compliance: IP Filtering and IPSec VPNs

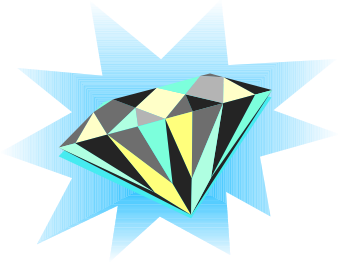


- **Routed Traffic Rules must not contain:**
  - Port Numbers
  - ICMP(v6) Code Types
  - OSPF Types
- **For Migration:**
  - Use the z/OS Migration Manual
  - Use the GUI
  - Use the z/OS V1R11 HealthChecker (available at V1R10)

- When given a choice, always try to configure IPSec with RFC4301 compliance. After V1R11 you **must** configure RFC4301 compliance and will not want to be forced to reconfigure your policies!

1. RFC4301 "Security Architecture for the Internet Protocol" specifies the base architecture for IPSec compliant systems
   1. – Includes restrictions on the routing of fragmented packets
   2. .. In z/OS V1R10 and V1R11, RFC4301 compliance enforcement is an optional setting in the z/OS IPSec policy
   3. – Changing an IPSec policy from non-compliant to compliant might require minor changes to IP filters for IP traffic that is routed through z/OS
2. RFC4301 - "Security Architecture for the Internet Protocol"
   1. Prior to RFC 4301 support, IPsec filters all routed IP fragments using a policy of first possible filter match (RFC4301 compliance=no)
      1. port, type, or code specifications are allowed on routed traffic rules
      2. filter all IP fragments by first possible filter match - except: non-initial IP fragments only match rules covering All ports, types, or codes
   2. RFC 4301 introduces rules and restrictions to ensure proper classification of fragments (RFC4301 compliance=yes)
      1. Use and enforce the RFC 4301 restrictions on IP filter rules: no port, type, or code specifications on routed traffic rules
      2. RFC4301Compliance parameter on the IpFilterPolicy statement
3. To be RFC4301-compliant, you should not filter on ports/type/code for routed traffic
   1. You can have the GUI enforce this or just issue a GUI health check warning
   1. A z/OS migration health check in z/OS V1R11 will determine if you have such filter rules:
      1. – ISTM010E IPSec filter rules that violate RFC4301 compliance are in use on this system during this IPL
4. This restriction can be temporarily suspended up through z/OS V1R11 until you update your policy to comply with the restriction. As an interim measure, you can configure the stack as Not compliant as indicated by one of the radio buttons in this GUI panel.
5. You may choose to relax the restriction until you have updated your configuration. If you choose to relax the restriction, you should be aware that the vulnerabilities cited in RFC 4301 concerning routed traffic and fragmented packets will apply to you.
6. At V1R12, you are no longer given a choice to be non-RFC4301-compliant.

Gems for Performance and Miscellaneous Items



Performance Items Presented Previously

1. Inbound Workload Queuing
2. Optimized Latency Mode
3. QDIO Acceleration
4. HiperSockets Multiwrite and zIIP Offload
5. WLM Service Priority associated with Outbound Queuing

Do You Know Your TCP/IP Stack's Hostname? (V1R10)

EZZ0162I HOSTNAME FOR tcpstackname IS hostname

- Some applications issue: GETHOSTBYNAME
- If your resolver file is not set up correctly, you could be providing an unintended hostname!
  - Need to understand Global Resolver processing - which changed radically way back at V1R2!.
- An IPL is required to change a hostname.
- Verify at startup what the hostname of your running TCP/IP job is!

1. TCPIP learns its hostname at startup.
2. ..Applications can issue a gethostname call to get the hostname TCPIP learned at startup.
3. ..Applications may fail if gethostname returns an unexpected hostname.
4. ..TCPIP needs to be recycled to learn a new hostname.
5. Search Order for Hostname:
   1. The name on the stack's TCPIP.DATA HOSTNAME statement is used.
      1. The z/OS UNIX search order is used to find the stack's TCPIP.DATA statements unless you use the Global Resolver. Refer to "Search orders used in the z/OS UNIX environment" in the z/OS Communications Server IP Configuration Guide for more information on the search order
      2. If there is no valid HOSTNAME statement, the VMCF node name with which VMCF was started is used.
      3. If VMCF was not active when the stack was started, the CVTSNAME value (the SYSNAME=value in IEASYSxx that was IPLed) is used.
6. To see where TCPIP is finding the hostname, you can add a SYSTCPT DD to the TCPIP proc. This will provide a resolver trace that will tell you what tcpip.data files are being used by TCPIP to find the hostname.
7. Applications can get hostnames to use in different ways. One way is to issue a gethostname call to get the hostname TCPIP learned at startup. If an unexpected hostname is returned, this can cause the application to not run properly.
8. Since TCPIP needs to be recycled to learn a new hostname, this can be an outage for the user – with the application not working until TCPIP is recycled to pick up the correct hostname.
9. With this enhancement in V1R10, you can find out the first time you initialize your TCP/IP stack what hostname it is using and make any corrections before things turn critical.

Security: Limiting Access to Unreserved Ports - (V1R10)

```
PORT   UNRSV   TCP   MYAPP1
PORT   UNRSV   TCP     *     SAF   RES2
PORT   UNRSV   TCP     *     SAF   GENERIC  WHENLISTEN
PORT   UNRSV   UDP     *     SAF   GENERIC  EPHEMERAL
```

- Controls TCP listens on unreserved ports
  1. Denies all TCP listens on an unreserved port,
     - **except for application MYAPP1**
  2. Denies all TCP listens on an unreserved port, except for all users permitted to the specified SAF resource:
     - **EZB.PORTACCESS.sysname.stackname.RES2**
  3. Prevents all users from opening any unreserved TCP ports as servers, unless they have access to the SAF resource:
     - **EZB.PORTACCESS.sysname.stackname.GENERIC**
  4. Prevents all users from explicitly binding to any unreserved UDP ports, unless they have access to the SAF resource:
     - **EZB.PORTACCESS.sysname.stackname.EPHEMERAL**

1. UDP and TCP port usage by server programs can be controlled via port reservations in the TCP/IP profile
2. If there is no port reservation for a given port number, then any application can use it as a server port
3. Prior to V1R10, you could use RESTRICTLOWPORTS to prevent users and jobs that are not authorized or UID(0) from choosing ports below 1024.
4. Port access can be controlled by server jobname or server userID access authorization to a SAF resource that is associated with the port.
5. This new function only controls application-specified ports. It does not affect generic binds or use of ephemeral ports (meaning, port number chosen by the stack).
6. **Caution:** PORT UNRSV controls could have broad and unexpected consequences
   1. .. For example: Client programs may execute under many different user IDs, so all address spaces where the client program can execute may need to be authorized.
7. POSSBILE IMPLEMENTATION APPROACH:
   1. Determine unreserved ports used by your applications
      1. Define following: .. PORT UNRSV protocol * SAF xyz WHENBIND
      2. Create.. SERVAUTH profile with UACC(READ)
         1. .. Audit the successes to determine the names of the applications
      3. Reserve ports for your discovered applications
         1. .. PORT or PORTRANGE profile statements
         2. .. Enable PORT UNRSV control by DENY or SAF UACC(NONE)
            1. .. Monitor failures and reserve ports as appropriate

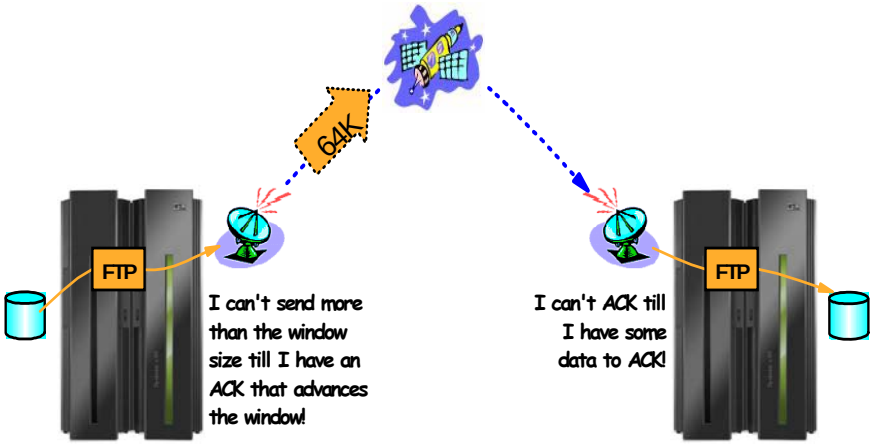
Verbose Ping (V1R11)

- z/OS ping has been made to look more like ping on other platforms.
  - A new verbose (or –v) option

```
USER1:/u/user1: >ping -v w3.ibm.com
CS V1R11: Pinging host w3.ibm.com (9.17.137.11)
with 256 bytes of ICMP data
ping #1 from 9.17.137.11: bytes=264 seq=1 ttl=242 time=56.64 ms
ping #2 from 9.17.137.11: bytes=264 seq=2 ttl=242 time=56.90 ms
ping #3 from 9.17.137.11: bytes=264 seq=3 ttl=242 time=57.96 ms
Ping statistics for w3.ibm.com (9.17.137.11)
  Packets: Sent=3, Received=3, Lost=0 (0% loss)
  Approximate round trip times in milliseconds:
  Minimum=56.64 ms, Maximum=57.96 ms, Average=57.17 ms, StdDev=0.70 ms
USER1:/u/user1: >
```


1. z/OS ping has been made to look like ping on most other platforms.
2. The new verbose or –v option will by default send three echo requests, calculate statistics for those three requests, and display the statistical summary as the response.

Dynamic Right Sizing (V1R11)

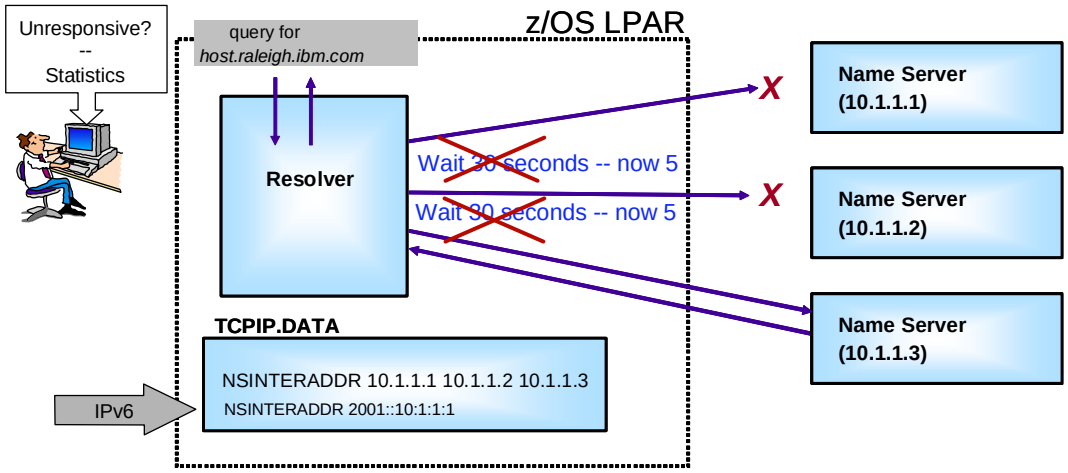


- **Improves performance for inbound streaming TCP connections over networks with large bandwidth and high latency by automatically tuning the ideal window size for such TCP connections.**
- **This function does not take effect for applications which use a TCP receive buffer size smaller than 64K.**
  - **The enhancement implements an algorithm known as "DYNAMIC RIGHT SIZING"**

1. Setting the TCP Buffer size to a minimum of 64K is important if you want to take advantage of "DYNAMIC RIGHTSIZING" in z/OS V1R11.
2. Streaming workload over large bandwidth and high latency networks (such as satellite links) is in general constrained by the TCP window size. The problem is that it takes time to send data over such a network. At any given point in time data filling the full window size is 'in-transit' and cannot be acknowledged until it starts arriving at the receiver side. The sender can send up to the window size and then must wait for an ACK to advance the window size before the next chunk can be sent.
3. If it were possible to dynamically adjust the window size to what it takes to fill the network in-between the sender and the receiver, higher throughput might be achieved.
4. This support will, on the receiver side, dynamically adjust the window size upward (beyond 180K if so needed) in an attempt to 'fill' the pipe between the sender and the receiver. The aim is that as soon as the sender has sent the end of its window, the sender receives an ACK from the receiver. That ACK allows the sender to advance the window and send another chunk onto the network.
5. NOTE: Be sure to check the size of your TCPRCVBUFRSIZE and adjust to 64K or higher; otherwise the Dynamic Right Sizing function in V1R11 may not work for you; the receive buffer must be equal to or larger than 64K. There is no healthchecker available to verify the size of the TCPRCVBUFRSIZE ... there is only one for TCPMAXRCVBUFRSIZE.

Resolver Enhancements: Communicate with IPv6 DNSs; Detecting Unresponsive DNS Servers (V1R12)

- Prior to V1R12:
  - Name server or network outages can introduce long resolver delays
  - Default setting for the timeout value is 30 seconds
  - Resolver uses the list, as coded, for every DNS query, regardless of past results



- Important changes:
  - RESOLVERTIMEOUT default changed to 5 seconds
  - Alerts sent to operator for unresponsive Domain Name Servers
  - Resolver can communicate with DNS at an IPv6 address

1. The combination of configuration options (NSINTERADDR, RESOLVERUDPRETRIES, multiple domain names specified with SEARCH, and RESOLVERTIMEOUT) in the TCPDATA (or resolv.conf) file can cause a resolver to continue using name servers even after unsuccessful queries can result in situations where long delays occur during name resolution. In this example, three name servers have been configured on the NSINTERADDR statement. Resolver will send the request for host.raleigh.ibm.com to the first DNS name server. No value for RESOLVETIMEOUT was coded, so a default setting of 30 seconds is used. That means resolver waits for 30 seconds for the name server at 10.1.1.1 to respond before moving to the next name server in the list. If the first two name servers are not responding, it takes a full minute for resolver to send the request to the third name server in the list. This one-minute delay applies to every request sent by resolver, as long as the first two name servers are not responding. Identifying situations where resolver is the bottleneck for transaction delays, and why it is the bottleneck, has proven to be a challenge for users.
2. In order to make these delays more visible to users, resolver is now going to maintain statistics on name server responsiveness to resolver queries. Two values are being kept: the total number of queries sent to the name server, and the total number of instances when the name server did not respond to a query. These numbers are maintained on a sliding five-minute window. Resolver does not maintain any long-term historical information beyond the most recent five minutes. Resolver uses these statistics to determine whether a name server is being responsive or not to resolver queries. To do this, every minute resolver calculates the percentage of failures over the most recent five minute interval. This percentage is then compared against a user-specified threshold at which a name server is considered to be unresponsive. If the five-minute percentage for this name server exceeds the threshold setting, resolver generates messages to the operator console to alert the operator of the unresponsive name server. If the name server had been unresponsive, but now the percentage is below the threshold, resolver generates different messages to the operator console.
3. The z/OS default for RESOLVERTIMEOUT has been changed to 5 seconds from 30.
4. Resolver tracenow displays the statistics for the Domain Name Servers.
5. In addition, starting with IPv6 the resolver can communicate with a Domain Name Server that is identified with an IPv6 address.

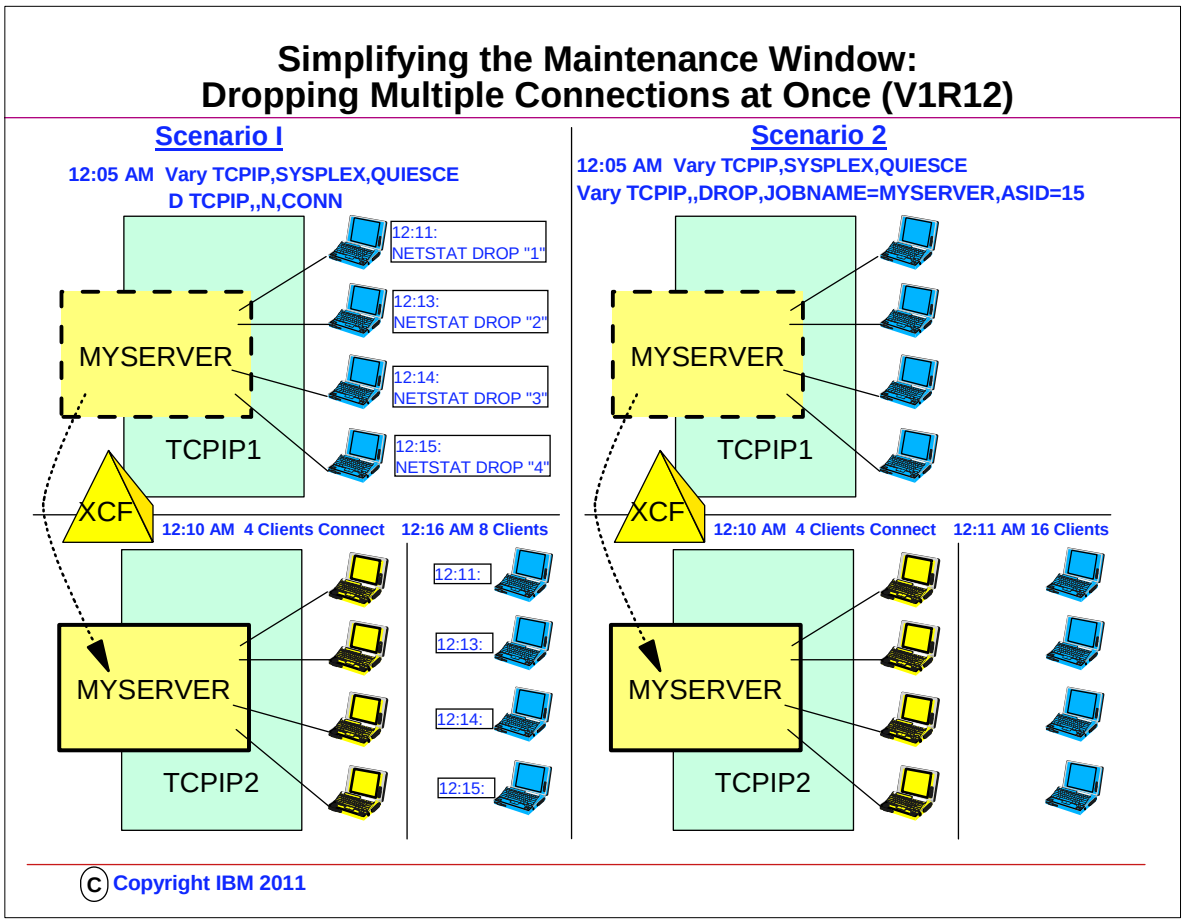
Unresponsivethreshold for Resolver (V1R12)

Display of contents of Resolver Setup File

F RESOLVER,DISPLAY

EZZ9298I DEFAULTTCPIPDATA - None
EZZ9298I GLOBALTCPIPDATA -
SYS1.TCPPARMS(TCPDATA)
EZZ9298I DEFAULTIPNODES - USER1.ETC.IPNODES
EZZ9298I GLOBALIPNODES - None
EZZ9304I COMMONSEARCH
EZZ9304I CACHE
EZZ9298I CACHESIZE - 200M
EZZ9298I MAXTTL – 214748364
EZZ9298I ***UNRESPONSIVETHRESHOLD - 25*** New Statement
EZZ9293I DISPLAY COMMAND PROCESSED

1. You can display the current resolver setup settings at any time using the MODIFY RESOLVER,DISPLAY command. The output from this command is also displayed when the resolver starts, and after a successful MODIFY RESOLVER,REFRESH command. An example of the output, showing the setting of the new UNRESPONSIVETHRESHOLD setup statement, is included on the slide. This example shows a 25% threshold setting, which as previously noted, is the default value.



1. SCENARIO I:
   1. If you want to move workload from one server application to another, for instance for maintenance purposes, you can quiesce the creation of new connections to the old server, but persistent connections need to be ended using the Netstat DROP/-D command.
   2. You would need to issue a Netstat CONN/-c display command to get the connection ID of each persistent session to be reset, and issue a Netstat DROP/-D command for each connection.
   3. If a server has dozens of persistent connections, this can be tedious.
   4. As you see in Scenario I, the original server has 4 connections from terminals that continue to survive after the SYSPLEX QUIESCE command is issued at 12:05 AM.
      1. Then the operator begins displaying the remaining connections to obtain the connection ID. Finally, the operator drops the connections one by one with a NETSTAT DROP command. (The operator could also have terminated the application to get rid of all connections at once, but in some cases this is not desirable.)
      2. The sessions connect back in, but this time to the takeover TCP/IP stack (TCPIP2).
2. SCENARIO 2:
   1. To address this problem, z/OS V1R12 Communications Server extended the existing VARY TCPIP,,DROP command with new parameters to allow all TCP connections associated with a server matching the specified filter to be reset. The structure of the new parameters is modeled after the parameters of the existing VARY TCPIP,,SYSPLEX,QUIESCE command.
      1. You can specify the job name and optionally the address space ID or the port number and optionally job name and address space ID for the servers. The command processor will scan the TCP connection table for listeners matching the supplied filters. If a match is found, it will scan the table again for all child connections associated with that listener. For each one found, it will reset the connection.
      2. If more than one server application is found to match the input filter values, the command will be failed. You can re-issue the command specifying additional filter parameters to identify a specific server application.
3. The Netstat DROP/-D command was not extended in this release, and can still reset only one connection per command invocation.
4. The format of the command is:

```
1. >>-VARY--TCPIP-+---------------+--,--+--DRop,------+--+-connid-----------------------------------+
2.                '- procname--'       '-CMD=DRop,- -'  '-CONNection=connid--------------------'
3.                                                       '-PORT=portnum -| opt_parms |------'
4.                                                       '-JOBNAME=jobname--+-----------------'
5.                                                                          '-ASID= <asid> ---'
```

End of Topic

End of Topic