e-business

Communications Server z/OS V1R5 and V1R6 Technical Update

The journey towards the next generation Internet - IPv6

z/OS V1R5

eserver

Topics

e-business

- z/OS V1R5
  - Sendmail 8.12 IPv6 support (*)
  - CICS sockets IPv6 enabled (*)
  - IPv6-enabled another batch of applications
    - –SNTPD, SyslogD, TFTPD, DCAS, remote execution commands and servers (*)
  - IPv6-enabled SNMP environment
  - New IPv6 interface support
    - –XCF - dynamic and static
    - –IUTSAMEH
    - –MPCPTP6
  - Policy agent IPv6 enabled - IPv6 QoS policy support (*)

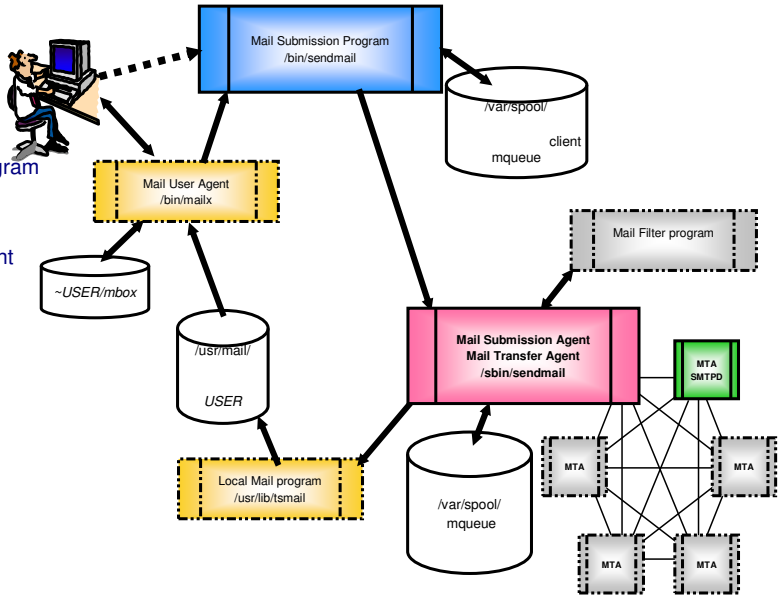
(*) - non-IPv6 enhancements included in this section

z/OS V1R5 - Sendmail 8.12 - IPv6, SSL/TLS, Milter

Background Information

e-business

- Mail is a complex environment
  - MUA
    - Mail User Agent
  - MSP
    - Mail Submission Program
  - MSA
    - Mail Submission Agent
  - MTA
    - Mail Transfer Agent
  - Milter
    - Mail Filter
  - Local Mailer Program



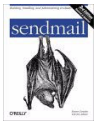
Sendmail 8.12 shipped with z/OS V1R5

Sendmail 8.12.1 vs 8.8.7

What's new in CS for z/OS V1R5

- IPv6 Support
- Mail filters (Milters)
- SSL Support
- Mail Submission Program (MSP)

- Sendmail has changed considerably in the last five years
- New functions were added by sendmail.org
  - Old sendmail did not support IPv6 or SSL
  - Security issues of two separate configuration files (MSP and MUA) and TLS (SSL) were added
  - Mail filters enhance function that customers can exploit
  - New techniques were added for I/O that may or may not help an installation.
    - see /usr/lpp/tcpip/samples/sendmail/TUNING
- By re-porting sendmail we get the latest level of function.
- We upgrade our support at the same time as new bat book (ISBN 1-56592-839-3)
- We align ourself with major changes in sendmail
- Continue to allow customers to migrate from SMTP to sendmail
  - sendmail offers function not available in SMTP
  - some function in SMTP still not available in sendmail
    - NJE Gateway and JES Support



IPv6 and Milter support

- Sendmail 8.12 supports listening on an IPv6 socket.
  - This is specified in the DAEMON_OPTIONS mc statement
    - DAEMON_OPTIONS('Name=MTA-6, Family=inet6')dnl
    - EZZ9929I issued if not IPv6 enabled
  - To listen only on IPv4 ports, specify inet family
    - DAEMON_OPTIONS('Name=MTA-4, Family=inet')dnl
  - The shipped sample specifies both inet and inet6 to allow IPv4 only and IPv6 installations to start with the sample
  - A message is issued to syslogd to explain which socket(s) are being used.
- Sendmail 8.12 also supports mail filtering (Milter)
  - Milter is a sendmail API to work with incoming mail at every step of the SMTP session: accepting, rejecting, discarding, altering.
  - A "milter daemon" is written by the installation and runs on the same host or potentially a different host in the background, parallel to the sendmail process, with sendmail communicating with it.
  - At every step of the SMTP session, callbacks to the milter daemon are issued, and the milter daemon advises the sendmail process what to do -- continue to accept the message, reject the message, etc.
  - Timeouts are provided for each stage (connect, send,read) and another timeout for overall elapsed time.
  - Milters use their own log level for debug, can be applied in any order, can use local or TCP sockets.
  - A simple sample is shipped in
    - /usr/lpp/tcpip/samples/sendmail/milter/lf_smpl.c

Details on Milter support

- Possible uses for filters include
  - Spam rejection
  - Virus filtering
  - Content control
- Address site-wide filter concerns
- Scalable and dynamicly changable
- Not for client-level concerns like sorting
- Separate function from SMTP provides
  - Security as non-root application
  - Simple API
  - Reliable (dumps affect one piece of mail)

- Milters can run on z/OS or other platforms
- Compile like any c program
  - cc -I. -o filter lf_smpl.c libmilter.a
- Works off a vector table for each type of filter to be used (connection, HELO, envelope, etc)
- Return codes specify whether to continue but content can be changed.
- Milters can be required or optional.
  - Configure for what happens if milter connection is lost? Continue to send mail or abort?
- Milters can be applied in any order
- Milters can be for connections to specific ports or all ports

SSL/TLS support and Mail Submission Program (MSP)

- Sendmail 8.12 supports SSL/TLS connections
  - Sendmail 8.12 ships with Open SSL APIs
    - CS for z/OS 1.5 uses System SSL APIs
  - Security encryption / functions are the same for TLS and SSL
  - System SSL Programming is used throughout z/OS
  - Provides compatibility with all other z/OS applications
  - Same information needed for SSL and TLS. Different format
  - /etc/mail/zOS.cf file new for SSL
    - KeyfilePath, ServerKeyfile, etc specified
    - Same as other CS for z/OS applications
    - Specified with ZOSCFFile mc option
  - Note sendmail SSL is not end-to-end encryption for mail
    - Provides encryption only to next mail hop
    - MUA could be non-sendmail

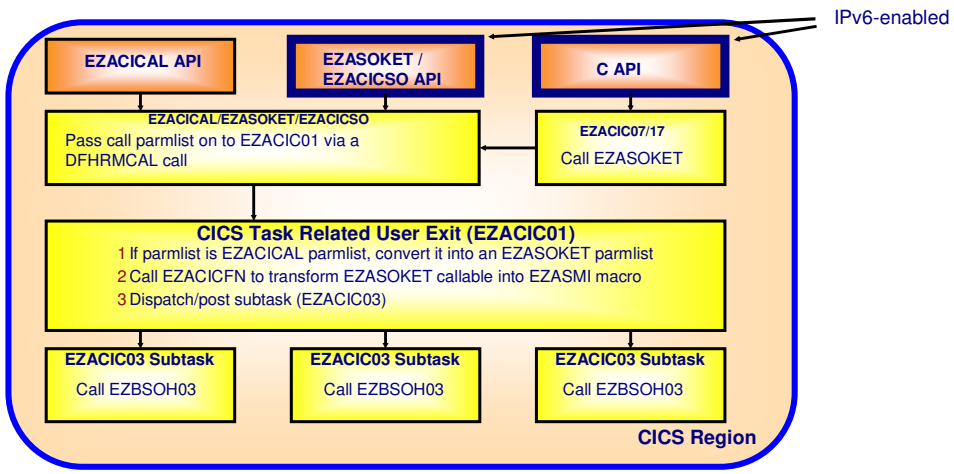
- Mail Submission Program (MSP)
  - Mail Submission Program (MSP) uses a separate config file (submit.cf)
  - Allows separation of hub's Mail Transfer Agent (MTA) and end user's Mail User Agent (MUA) functions and security
  - For most installations the default submit.cf file can just be copied to /etc/mail/submit.cf
  - A second binary /bin/sendmail shipped for MUA in a Program Control Environment
    - Program Control enhances Unix security
    - http://www.ibm.com/servers/eserver/zseries/zos/unix/faq/chuid.html
  - /bin/sendmail never setuid() so it provides more security
  - must chmod and chown it to the sendmail uid
    - chown 25:25 /bin/sendmail
    - chmod 6755 /bin/sendmail

CICS Sockets - IPv6 support

CICS sockets enhancements in z/OS V1R5

- IP CICS Sockets APIs support the *Basic Socket Interface Extensions for IPv6*
- Support both IPv4 and IPv6 Standard and Enhanced Listener
- For the IP CICS Sockets Extended and IP CICS C Sockets APIs support the IPv6 Multicast options for GETSOCKOPT and SETSOCKOPT.
- For the IP CICS C Sockets API support IPv4 Multicast options for getsockopt and setsockopt.
- Provide IPv6 and Assembler samples

CICS sockets APIs



Both CICS C-sockets and Call EZACICAL socket programs are transformed into calls to the sockets extended callable API before the socket calls are passed down to the socket communicating subtasks, making the full CICS socket implementation much more streamlined. The subtasks now only have to do call routing on behalf of the CICS task.

Really, EZACICAL calls are transformed directly into EZASMI macro calls by EZACIC01, there's not a transform to EZASOKET first. (According to Bill Kelsey, Oct 2001).

A CICS task may use sockets extended callable sockets, including assembler callable sockets; but not the sockets extended assembler macro API.

There is no change in the linkageedit control statements from V3R1 to V3R2 - for a CICS C-socket program you still need to include EZACIC07, and for both sockets extended and EZACICAL callable programs, you need to include the EZACICAL module

CICS Sockets Extended - new sockets functions in CICS

- New IP CICS Sockets Extended API Resolver commands:
  - FREEADDRINFO
    - Free all the address information structures returned by the GETADDRINFO command addressed by the RES parameter.
  - GETADDRINFO
    - Translates the name of a service location (for example, a host name) and/or service name and returns a set of socket addresses and associated information to be used in creating a socket with which to address the specified service.
      - New utitlity (EZACIC09) to break out pointers and bit strings in an ADDRINFO structure for COBOL programmers
  - GETNAMEINFO
    - Returns the node name and service location of a socket address that is specified in the call.
- New IP CICS Sockets Extended API commands:
  - NTOP
    - Convert an IP address from numeric to presentation
  - PTON
    - Convert an IP address from presentation to numeric

CICS C Sockets - new sockets functions in CICS

- New IP CICS C Sockets API functions:
  - **inet_ntop**
    - Convert an IP address from numeric to presentation
  - **inet_pton**
    - Convert an IP address from presentation to numeric
  - **if_freenameindex**
    - Release if_nameindex array storage
  - **if_indextoname**
    - Given an interface index, return an interface name
  - **if_nameindex**
    - Obtain a list of interface names and their corresponding indices
  - **if_nametoindex**
    - Given an interface name, return an interface index
- New IP CICS C Sockets API Resolver functions:
  - **gai_strerror**
    - Returns a pointer to a text string describing the error value returned from the freeaddrinfo, getaddrinfo or getnameinfo function.
  - **freeaddrinfo**
    - Free all the address information structures returned by the getaddrinfo function addressed by the res parameter.
  - **getaddrinfo**
    - Translates the name of a service location (for example, a host name) and/or service name and returns a set of socket addresses and associated information to be used in creating a socket with which to address the specified service.
  - **getnameinfo**
    - Returns the node name and service location of a socket address that is specified in the call

CICS C Sockets - new header files and macros

- IP CICS C Sockets API "borrows" the TCP C header files. These headers can be found in *hlq*.SEZACMAC. Include the following definition to expose IPv6 data structures and definitions: **`#define CICS_IPV6`**

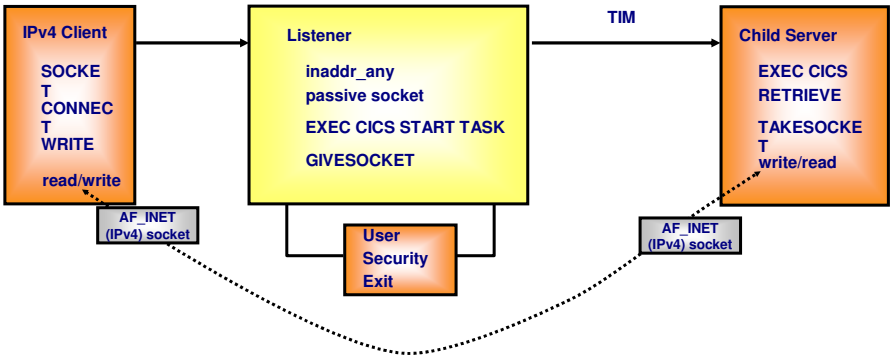
- The affected header files are:
  - if.h
  - in.h
  - inet.h
  - ioctl.h
  - netdb.h
  - socket.h

- New header to define the Task Input Message structure:
  - ezacictm.h - Note: This macro supports IPv4 and IPv6 Standard and Enhanced Child Servers

- New IP CICS C Sockets API Address testing Macros:
  - IN6_IS_ADDR_UNSPECIFIED
  - IN6_IS_ADDR_LOOPBACK
  - IN6_IS_ADDR_MULTICAST
  - IN6_IS_ADDR_LINKLOCAL
  - IN6_IS_ADDR_SITELOCAL
  - IN6_IS_ADDR_V4MAPPED
  - IN6_IS_ADDR_V4COMPAT
  - IN6_IS_ADDR_MC_NODELOCAL
  - IN6_IS_ADDR_MC_LINKLOCAL
  - IN6_IS_ADDR_MC_SITELOCAL
  - IN6_IS_ADDR_MC_ORGLOCAL
  - IN6_IS_ADDR_MC_GLOBAL

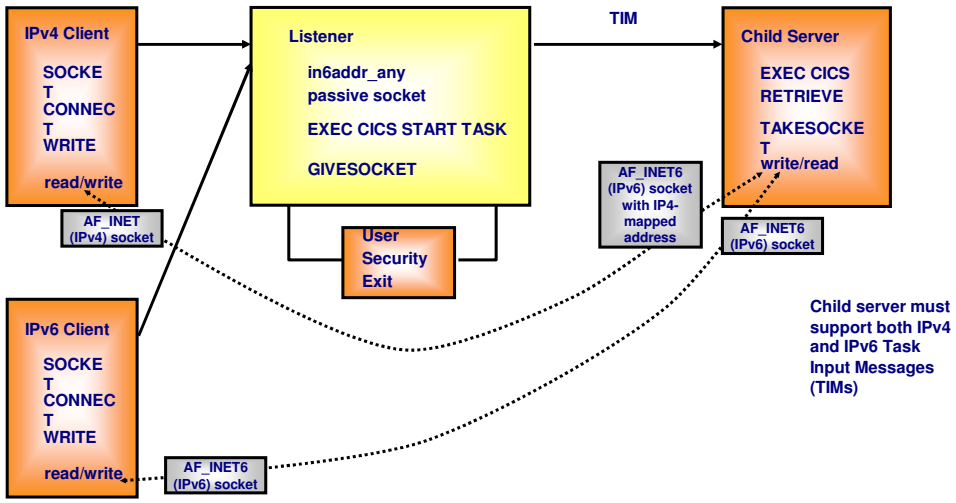
CICS Sockets listener - IPv4 only

- Pre-V1R5 IP CICS standard or enhanced listener - and non-IPv6 enabled CICS standard and enhanced listener in V1R5 and later:



CICS Sockets listener - IPv6 enabled

- V1R5 IP CICS IPv6-enabled standard or enhanced listener:



IPv6 Enhanced Listener Definition

- EZAC Online IPv6 Enhanced Listener definition

```
EZAC,DEFine,LISTENER (enhanced listener)                    APPLID = CICS1A
OVERTYPE TO ENTER
APPLID       ===> CICS1A              APPLID of CICS System
TRANID       ===> ENH6                Transaction Name of Listener
PORT         ===> 07214               Port Number of Listener
AF           ===> INET6               Listener Address Family
IMMEDIATE    ===> YES                 Immediate Startup   Yes|No
BACKLOG      ===> 020                 Backlog Value for Listener
NUMSOCK      ===> 050                 Number of Sockets in Listener
ACCTIME      ===> 060                 Timeout Value for ACCEPT
GIVTIME      ===> 000                 Timeout Value for GIVESOCKET
REATIME      ===> 000                 Timeout Value for READ
CSTRANid     ===> SRV6                Child server transaction name
CSSTTYPe     ===> KC                  STartup method  (KC|IC|TD)
CSDELAY      ===> 000000              Delay interval (hhmmss)
MSGLENgth    ===> 000                 Message length (0-999)
PEEKDATa     ===> NO                  Enter Y|N
MSGFORMat    ===> ASCII               Enter ASCII|EBCDIC
USEREXIT     ===> ENH6EXIT            Name of user/security exit
WLM groups   ===>            ===>            ===>

DEFine       FUNCTION COMPLETED SUCCESSFULLY

PF              3 END                                          12 CNCL
```

Listener - User/Security Exit interface for an IPv6-enabled listener

- The commarea the User/Security exit receives from the listener is changed to support IPv6 data structures.
- A new Assembler DSECT, EZACICSX, is provided to help map the commarea.

```
EZACIC_SECEXIT           DSECT ,  SECURITY/TRANSACTION EXIT STRUCTURE
EZACIC_TRANID            DS CL4   CICS TRANSACTION ID
EZACIC_CLIENT_DATA       DS CL35  USER DATA FROM CLIENT
EZACIC_EXPIND            DS CL1   EXPANDED FORMAT INDICATOR
EZACIC_RESERVED1         DS CL4   RESERVED FOR IBM USE
EZACIC_ACT               DS CL2   METHOD TO START TASK (IC/KC/TD)
EZACIC_TIME              DS CL6   IC START INTERVAL (HHMMSS)
EZACIC_AF                DS H     NETWORK ADDRESS FAMILY
EZACIC_PORT              DS H     REQUESTER'S PORT NUMBER
EZACIC_IPV4ADDR          DS F     IPV4 ADDR OF REQUESTER'S HOST
EZACIC_SWITCH1           DS CL1   SWITCH 1 1=PERMIT, ELSE PROHIBIT
EZACIC_SWITCH2           DS CL1   SWITCH 2
EZACIC_TERMID            DS CL4   CICS TERMID (0'S IF NO TERMINAL)
EZACIC_SOCKDESC          DS H     CLIENT'S SOCDESC
EZACIC_USERID            DS CL8   USER ID FIELD
EZACIC_LISTENER_IPV4ADDR DS F     LISTENERS IPV4 ADDRESS
EZACIC_LISTENER_PORT     DS H     LISTENERS PORT
EZACIC_LISTENER_IPV6ADDR DS CL16  LISTENERS IPV6 ADDRESS
EZACIC_LISTENERS_SCOPEID DS CL4   REQUESTORS SCOPE ID
EZACIC_IPV6ADDR          DS CL16  IPV6 ADDR OF REQUESTER'S HOST
EZACIC_SCOPEID           DS CL4   REQUESTORS SCOPE ID
                         DS CL40  RESERVED FOR FUTURE USE
EZACIC_DATA2_LEN         DS H     LENGTH OF DATA RCVD FROM CLIENT
EZACIC_DATA2             DS 0C    DATA RECEIVED FROM CLIENT
```

Task Input Message from an IPv6-enabled listener

- The Task Input Message (TIM) sent when the Child Server task is started by the Listener changed to accommodate storage for both the IPv4 and IPv6 socket address:

```
01 TCPSOCKET-PARM.
       05 GIVE-TAKE-SOCKET                    PIC 9(8) COMP.
       05 LSTN-NAME                                 PIC X(8).
       05 LSTN-SUBNAME                        PIC X(8).
       05 CLIENT-IN-DATA                      PIC X(35).
       05 FILLER                                    PIC X(1).
       05 SOCKADDR-IN-PARM.
     10 SIN-FAMILY              PIC 9(4) BINARY VALUE 0.
     10 SOCK-DATA               PIC X(26) VALUE LOW-VALUES.
     10 SOCK-SIN REDEFINES SOCK-DATA.
        15 SOCK-SIN-PORT             PIC 9(4) BINARY.
        15 SOCK-SIN-ADDR             PIC 9(8) BINARY.
        15 FILLER                    PIC X(8).
        15 FILLER                    PIC X(12).
     10 SOCK-SIN6 REDEFINES SOCK-DATA.
        15 SOCK-SIN6-PORT            PIC 9(4) BINARY.
        15 SOCK-SIN6-FLOWINFO        PIC 9(8) BINARY.
        15 SOCK-SIN6-ADDR.
           20 FILLER                          PIC 9(16) BINARY.
           20 FILLER                          PIC 9(16) BINARY.
        15 SOCK-SIN6-SCOPEID PIC 9(8) BINARY.
   05 FILLER                                        PIC X(68).
   05 CLIENT-IN-DATA-LENGTH           PIC 9(4) BINARY.
   05 CLIENT-IN-DATA-2                        PIC X(xxx).

where xxx is at least equal to the largest MSGLEN parameter for the Listeners
that can start this application.
```

CICS Sockets samples

- Sample programs illustrating IPv4 and IPv6

NAME	Description	Language	IPv4	IPv6
EZACICSC	Child server	COBOL	Yes	No
EZACICSS	Iterative Server	COBOL	Yes	No
EZACIC6C	Child Server	COBOL	No	Yes
EZACIC6S	Iterative Server	COBOL	No	Yes
EZACICAC	Child Server	Assembler	Yes	Yes
EZACICAS	Iterative Server	Assembler	Yes	Yes

- *hlq*.SEZAINST(EZACICCT) contains the DFHCSDUP commands to define the new sample program and transaction to CICS.

The source for these programs can be found in the SEZAINST library and also in the API Guide.

CICS Sockets migration concerns

- No migration concerns for existing IP CICS Sockets Extended IPv4 applications
- No migration concerns for existing IP CICS C IPv4 applications
- No migration concerns for current Listener definitions. Current listener will continue to execute as IPv4 listeners. Current child servers will continue to work as today.
- If you want your listener to be defined as AF_INET6, then you must add the AF=INET6 to the EZACICD listener definition macro. You can also use the EZAC,ALTER,LISTENER online command to dynamically change the AF of the listener.
  - You must update any Child Server transaction programs to make use of the IPv6 socket address structure passed as part of the Task Input Message. The length of data received by the EXEC CICS RETRIEVE call will have to change to handle the storage necessary to contain the IPv6 socket address structure.
  - You must update any Security/User exit programs defined in the listener configuration to accommodate new IPv6 data elements
  - IP CICS C programs requiring IPv6 function must include the define CICS_IPV6 to expose required IPv6 structures and definitions in the TCP C header files.

IPv6-enabled SNTPD, SyslogD, TFTPD, DCAS

Adding IPv6 support

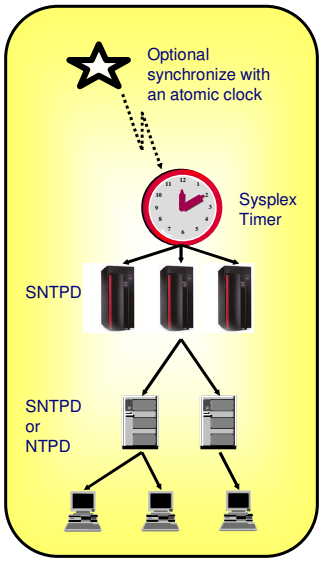
- SNTPD, SyslogD, TFTPD, and DCAS have been IPv6-enabled and are supported on z/OS IPv4-only TCP/IP stacks and on z/OS dual-mode TCP/IP stacks.
  - The syslogd.conf file or dataset will now accept an IPv6 address or host name that resolves to an IPv6 address to which messages will be forwarded.
  - The dcas.conf file or dataset will now accept an IPv6 address or host name that resolves to an IPv6 address on which the DCAS server will listen.
  - There are no new start options for TFTPD or SNTPD to communicate over an IPv6 network.
  - Each application will attempt to create an AF_INET6 socket.
  - If opening an AF_INET6 socket fails, then only IPv4 communication will be used.
  - SNTPD has an AF_INET socket listening for each IPv4 interface as in the past, but only one AF_INET6 socket listening on in6addr_any.

SNTPD - IPv4 vs. IPv6

- IPv4 processing
  - SNTPD opens one socket per interface (including VIPAs) and binds that socket to the interface address.
  - All sockets are closed and re-established on 5 minute intervals to pick up any new interfaces.
  - Cannot bind to inaddr_any due to the way SNTP clients work.
    - Some SNTP clients require the source IP address of the server's reply to equal the destination IP address of their initial request.
    - In a multi-path environment or for VIPAs, the source IP address of the server's reply may not equal the destination address of the client's request.
- IPv6 processing
  - SNTPD opens one socket and binds that socket to in6addr_any.
  - No check for IPv6 interfaces and socket is not closed unless SNTPD exits.
  - Ancillary data is used to solve the source IP address / destination IP address issue.
    - recvmsg() and sendmsg() are used to control the source IP address of the server's reply
- Maintained the IPv4 process due to IPv4-only stack and IPv4 multicast support

SNTPD configurable stratum level

- SNTPD is a TCP/IP daemon that is used to synchronize time between a client and a server.
- Simple Network Time Protocol (SNTP) is a protocol for synchronizing clocks across a WAN or LAN through a specific formatted message. The SNTP protocol is compatible with the NTP protocol and SNTP servers can coexist with NTP servers in a time synchronization tree.
- An External Time Reference (ETR), named stratum 0, is chosen as the highest timer reference used for synchronization. A stratum 1 server is attached to and receives the time from the stratum 0 timer. For example, the z/OS sysplex timer could be a stratum 0 timer, and z/OS Communications Server would be a stratum 1 server.
- A client attached to the stratum 1 server can also be a stratum 2 server, receiving the time from the stratum 1 server, and so on.
- An SNTPD start option has been added to allow the user to specify the stratum level reflecting the accuracy of the z/series clock.
- '-s' is the new start option used to indicate the stratum level for SNTPD.
  - Valid range is 1-15.
  - Default is 1.



SyslogD non-swappable

- SYSLOGD non-swappable
  - Use the following guidelines when using RACF to set the desired state for SYSLOGD:
    - If the FACILITY class resource BPX.STOR.SWAP is not defined to the system, syslogd will run nonswappable and cannot be prevented from running nonswappable
    - If the FACILITY class resource BPX.STOR.SWAP is defined to the system with UACC(NONE), syslogd will run swappable by default (no access to BPX.STOR.SWAP)
      - `ICH408I USER(SYSLOGD ) GROUP(OEA ) NAME(####################) 899 BPX.STOR.SWAP CL(FACILITY) INSUFFICIENT ACCESS AUTHORITY ACCESS INTENT(READ ) ACCESS ALLOWED(NONE )`
      - syslogd can run nonswappable (given at least READ access to BPX.STOR.SWAP)
    - This behavior may differ depending on which security product is used.
    - The SyslogD started task user ID must be UID-0.
  - To define the FACILITY class resource BPX.STOR.SWAP issue the following commands:
    - RDEFINE FACILITY BPX.STOR.SWAP UACC(NONE)
    - SETROPTS RACLIST(FACILITY)REFRESH

Your discussion about BPX.STOR.SWAP may depend on use of RACF or another security product. Your description fits RACF. I am not fully sure it would fit if the customer uses ACF2 or TopSecret. **RACF says a resource is not protected if it isn't defined - hence youn run non-swappable if the resource isn't defined. ACF2 and TopSecret say a resource is protected (and no-one can use it) if it isn't defined - which leads me to believe that you'll probably run swappable if BPX.STOR.SWAP isn't defined in case ACF2 or TopSecert is used.** You may want to add a blurp that this behavior may depend on which security product is used.

Migration Concerns

- IPv6 Support
  - The new function does not impact the current function.
  - For all of the protocols involved, there is no change to the payload due to IPv6 support.
  - DCAS
    - The connection to the LDAP server will continue to be IPv4 only due to lack of support for IPv6 in the LDAP access libraries.
    - The address of the LDAP server will continue to be IPv4 only.
    - DCAS just passes the LDAP server address or name on to System SSL which takes care of the error case if the address is IPv6.
- SNTPD Stratum Level
  - SNTP clients may choose to ignore the time, if the SNTPD stratum level is not 1.
- SYSLOGD can run swappable or nonswappable.
  - When an application makes an address space nonswappable, it might convert additional real storage in the system to preferred storage.
  - Allowing SYSLOGD to run in a nonswappable state can reduce the installation's ability to reconfigure storage in the future, since preferred storage cannot be configured offline.
  - If BPX.STOR.SWAP has not been defined, SYSLOGD will now run as non-swappable.
  - If BPX.STOR.SWAP is defined without permitting SYSLOGD, SYSLOGD will run swappable as usual.
  - If BPX.STOR.SWAP is defined and SYSLOGD is permitted, SYSLOGD will run non-swappable.

IPv6-enabled remote execution commands and servers

IPv6 enabling MVS clients and servers and shipping new UNIX rsh client

- The following clients and server have been IPv6-enabled:
  - TSO rexec client
  - TSO rsh client
  - MVS remote execution daemon
- The z/OS IPv4-only TCP/IP stack and the z/OS dual-mode TCP/IP stack are supported.
- TSO Remote Execution Clients (rexec and rsh)
  - The *foreign host* parameter for the clients may now be an IPv6 address or host name that resolves to an IPv6 address.
  - There are no new options for the clients.
  - Clients were converted from the Pascal API to the LE C API which is IPv6-enabled.
    - Pascal to C conversion had some implications
    - Maintain old behavior for search orders and translation support
- The MVS remote execution daemon was changed from using TCP/IP C sockets to LE C sockets
- An rsh client has been created for the z/OS UNIX environment similar to the existing z/OS UNIX rexec client.

IPv6 support in the MVS remote execution server

- New start parameter for the MVS remote execution daemon
  - IPV6=Y|N indicating whether the server should attempt communication over an IPv6 network.
  - If this option is not specified, then the server will attempt IPv6 communication.
    - If IPv6 communication fails, then IPv4 communication will be used.
  - Specifying N for this option prevents IPv6-only clients from communicating with this server.
    - Only IPv4 communication will be attempted.
  - Specifying Y for this option allows IPv4 clients and IPv6 clients to communicate with this server.
    - If the TCP/IP stack is not IPv6-enabled and start parameter IPV6=Y:
      - Open of AF_INET6 socket will fail
      - Error message will be logged
      - AF_INET socket will be attempted
      - Only IPv4 communication possible
  - This option is useful for installations that have not migrated remote execution user exits to accommodate IPv6 addresses.

MVS Remote execution server user exit - notes

NOTES

- The user exit should have the AMODE(31) and RMODE(24) attributes to provide addressability to the input parameters.
- On entry to the user exit, register 1 points to the following parameter list:

```
Offset Description
0      A pointer to a mixed AF_INET or AF_INET6 address structure
       Rule: The address family must be examined to determine which type of
       address structure is being used.
4      A pointer to JOB statement parameters
8      A pointer to EXEC statement parameters
12     A pointer to an optional JES control statement
```

- The INET address consists of the following fields:

```
Offset Description
0      2 bytes (AF_INET or AF_INET6)
2      2 bytes (server port)
4      4 or 16 bytes (client AF_INET or AF_INET6 address)
```

- If the server is IPv6 enabled and an IPv4 client connects, the IP address is the 4-byte IPv4 address and not the IPv4-mapped IPv6 address.

The new UNIX orsh command

- orsh is a new command for the z/OS UNIX environment.
  - There is a man page for orsh.
  - orsh -? -d -l user_id/password -s port foreign host command
    - -? Displays the help message.
    - -d Activates debug tracing.
    - -l user_id/password Specifies the user ID and password on the foreign host.
    - -s port Specifies the TCP port number of the rsh server on the foreign host.
      - The default is the port number defined in /etc/services.
    - foreign_host Specifies the name or IP address of the foreign host to which you are sending the orsh command.
    - command - Specifies the command that is sent to the foreign host.
      - The command is composed of one or more words.
      - Coding is assigned after checking the prefixed parameters (-l, -s) and assigning the remaining string as the command.
      - The command you specify must not require a response from you to complete.
      - orsh cannot interact with you after you enter data in the command format.

Migration Concerns

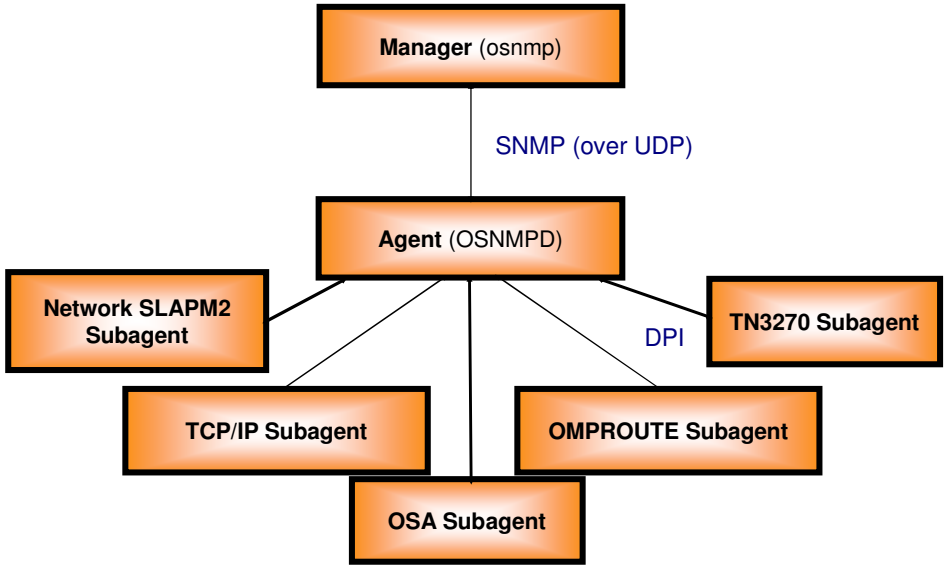
- The new function does not impact the current function.
  - For the remote execution protocol, there is no change to the payload due to IPv6 support.
- The TCP/IP stack on your system must support IPv6 networking.
  - If not, these applications will operate in IPv4 mode.
- Unless IPV6=N is specified, the remote execution user exits must be changed to operate with IPv6 clients.
  - The user exit must be able to receive an IPv6 socket address structure.
  - Specifying IPV6=N will preserve operation of user exits that do not support IPv6.
  - **Must specify IPV6=N or change the user exit - default behavior will break current user exits.**
  - If server supports IPv6 and IPv4 client connects, pointer to client's IPv4-mapped IPv6 address will be passed to the user exit.
- Automation on certain messages may be an issue.
  - For the remote execution clients that were converted from Pascal to C, some debug messages were changed
  - Some messages have been maintained.
  - For the remote execution daemon, messages did not change.

IPv6-enabling SNMP on z/OS

SNMP Overview

- SNMP is a set of protocols describing management data and the operations for conveying it across heterogeneous systems.
- The primary SNMP functional entities are:
  - manager
    - Application that requests management data
  - agent
    - Server that responds to requests for management data
  - subagent
    - Assists agent by supporting a particular set of management data
- SNMP applications can in z/OS V1R5 communicate over an IPv6 connection:
  - osnmp command
  - SNMP agent (OSNMPD)
  - Trap Forwarder daemon
  - subagents
  - pwtokey and pwchange commands now accept IPv6 addresses

SNMP Overview



osnmp command and configuration changes

- osnmp queries management information from one or more SNMP agents that may be running at IPv6 addresses
- OSNMP.CONF is used to define target agents and, for SNMPv3, security parameters used in communicating with target agents
- Can now specify an IPv6 address in OSNMP.CONF or on -h option when invoking osnmp command
- In OSNMP.CONF, an IPv6 address can optionally be followed by two periods (..) and a port number

```
v2c              127.0.0.1       snmpv2c
v2c_ipv6         12ab::2         snmpv2c
v2c_ipv6_port    12ab::2..1061   snmpv2c
```

  - > osnmp -h v2c_ipv6 get sysDescr.0
  - > osnmp -h 12ab::2 -c get sysUpTime.0
  - > osnmp -v -h v2c_ipv6_port walk ipAddrTable

- All existing OSNMP.CONF entries will work as they previously did
  - For consistency, customers may want to change any OSNMP.CONF entries that use a single colon (:) as the delimiter between hostname / IPv4 address and port number to use two periods (..) instead, e.g.

```
v1c  9.42.105.68:161  snmpv1
   would become
v1c  9.42.105.68..161  snmpv1
```

  - The :portnum notation is not supported on OSNMP.CONF entries that specify IPv6 addresses. So if you don't change your existing IPv4 entries that use the :portnum notation, and you add new IPv6 entries, you may end up with both kinds of syntax (:portnum and ..portnum) for specifying destination port numbers.

SNMPD agent configuration changes

- The files used to configure the SNMP agent are either:
  - PW.SRC and SNMPTRAP.DEST (to support only SNMPv1 and v2c, but not v3 security)
    OR
  - SNMPD.CONF (to support all types of security, including SNMPv3).

- OSNMPD accepts SNMP requests from management applications, authenticates requests using community- or user-based security, responds to the requests, and sends asynchronous notifications to configured management applications

- Can now specify IPv6 addresses in PW.SRC or SNMPD.CONF to authenticate SNMPv1/v2c requests from management applications in IPv6 networks

- Can now configure IPv6 addresses in SNMPTRAP.DEST or SNMPD.CONF as notification destinations

- IPv6 prefix values are supported for convenience because colon-hexadecimal form can be cumbersome to deal with. For example, the following PW.SRC entries are equivalent:

```
comm1       12ab:34cd:56ef::       ffff:ffff:ffff::
comm1       12ab:34cd:56ef::       48
```

- IPv4 prefix values (0-32) can now be used as an alternative to IPv4 network masks in dotted decimal notation. For example, the following PW.SRC entries are equivalent:

```
comm1       9.67.0.0       255.255.0.0
comm1       9.67.0.0       16
```

- In general, refer to the IP configuration reference manual for the details on configuring OSNMPD for IPv6

DPI Enhancements

- Distributed Protocol Interface (DPI) version 2.0 library now supports IPv6
- `DPIconnect_to_agent_TCP()` and `DPIconnect_to_agent_UNIXstream()` support an IPv6 address in colon hexadecimal form for `hostname_p` parameter
- `lookup_host6()` attempts to resolve a hostname to an IPv6 address
- If a DPI subagent passes a hostname for the hostname_p parameter to DPI connect routines, the initial SNMP query for the DPI port number or path name will be sent using IPv4 or IPv6 packets. Which type of packet is used depends on how the hostname is resolved, but it can be important because it determines whether the source address of the initial SNMP query will be IPv4 or IPv6. The SNMP agent must be configured to allow the source address for the community name used by the subagent.
- In this situation, a subagent can force the initial SNMP query to be sent using IPv4 packets or IPv6 packets by doing its own hostname-to-address resolution using lookup_host() or lookup_host6() (respectively) and passing the IP address string to the DPI connect routine.

If a DPI subagent passes a hostname for the `hostname_p` parameter to DPI connect routines, the initial SNMP query for the DPI port number or path name will be sent using IPv4 or IPv6 packets. Which type of packet is used depends on how the hostname is resolved, but it can be important because it determines whether the source address of the initial SNMP query will be IPv4 or IPv6. The SNMP agent must be configured to allow the source address for the community name used by the subagent.

In this situation, a subagent can force the initial SNMP query to be sent using IPv4 packets or IPv6 packets by doing its own hostname-to-address resolution using `lookup_host()` or `lookup_host6()` (respectively) and passing the IP address string to the DPI connect routine.

Migration considerations

- Algorithm used in generating an engineID for SNMPv3 authentication and privacy has been changed to support a more current standard
  - Existing key definitions will still work as long as the SNMP agent engineID is not changed
  - If agent engineID is changed, for example, by deleting the SNMPD.BOOTS file and letting the agent regenerate its engineID...
    - Must use pwtokey to regenerate any localized SNMPv3 keys used by the SNMP agent and SNMPv3 network managers (such as osnmp)
- If the SNMP agent obtains an IPv6 address for itself when it initializes, then all SNMPv1 traps generated will encode 0.0.0.0 as the IP address
  - Limitation in SNMP architecture
    - The SNMP architecture requires the source IP address of an SNMPv1 trap to be encoded in a 4-byte field. Consequently, IPv6 addresses cannot be represented.
    - If the agent is started with the -A option, then SNMPv1 traps are guaranteed to encode the "real" IPv4 source address of the agent.
  - Can prevent this by specifying -A when starting the agent
    - Forces agent to obtain an IPv4 address for itself

SNMP TCP/ip stack subagent enhancements

- IPv6 management data support
  - Added support for the version-neutral (both IPv4 and IPv6) MIB data in the following new, IETF internet drafts:
    - **IP-MIB: draft-ietf-ipv6-rfc2011-update-01.txt**
    - **IP-FORWARD-MIB: draft-ietf-ipv6-rfc2096-update-02.txt**
    - **TCP-MIB: draft-ietf-ipv6-rfc2012-update-01.txt**
  - IF-MIB support now includes IPv6 interfaces
  - SNMP TCP/IP Enterprise-specific MIB updates:
    - **Existing ibmTcpipMvsTcpListenerTable now supports both IPv4 and IPv6 Servers**
    - **New MIB objects added to ibmTcpipMvsPortTable to support an IPv6 IP address specified on the BIND keyword of the PORT Profile statement**
- Added the SACACHETIME keyword to the SACONFIG Profile statement so that the Subagent cache time can be configured by a means other than an SNMP command. The SACONFIG Profile statement is used to configure the TCP/IP Subagent.
- New message EZZ3231I issued to console if Subagent unable to process a request due to low private area storage. This message will be issued only if it has been 15 minutes since the last time the message was issued.
- New ibmTcpipMvsPktTraceTable added to the TCP/IP Enterprise-specific MIB to provide packet trace management data
- New MVS image MIB objects added to the TCP/IP Enterprise-specific MIB:
  - ibmMvsSystemName - This value comes from the CVTSNAME field of the system. This is the SYSNAME specified in the IEASYSxx member of SYS1.PARMLIB.
  - ibmMvsSysplexName - This value comes from the ECVTSPLX field of the system. This is the SYSPLEX value on the COUPLE statement in the COUPLExx member of SYS1.PARMLIB.

"version-neutral" MIB object definitions are based on SNMP textual conventions defined in the INET-ADDRESS-MIB from RFC 3291 and IETF draft draft-ietf-ops-rfc3291bis-00.txt.

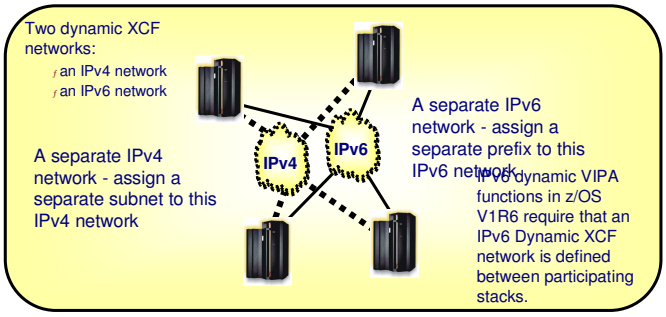
The SNMP TCP/IP Enterprise-specific MIB is installed in the HFS in the /usr/lpp/tcpip/samples directory as files: mvstcpip.mi2 (SMIv2) and mvstcpip.mib (SMIv1).

IPv6 Interface Support for SAMEHOST, XCF, and ESCON

New IPv6 connectivity options

- Provide IPv6 connectivity over MPCPTP to another z/OS V1R5 stack
  - XCF to other stacks in same sysplex
    - The VTAM ISTLSXCF major node must be active for XCF connectivity (same requirement as for IPv4)
  - IUTSAMEH to other stacks on same LPAR
  - ESCON/FICON to another z/OS image
- Provide automatic connectivity option to other stacks in same sysplex (XCF Dynamics for IPv6)

Dynamic XCF for IPv6 will not use HiperSockets in this release for intra-CEC communication because iQDIO isn't IPv6 enabled yet.



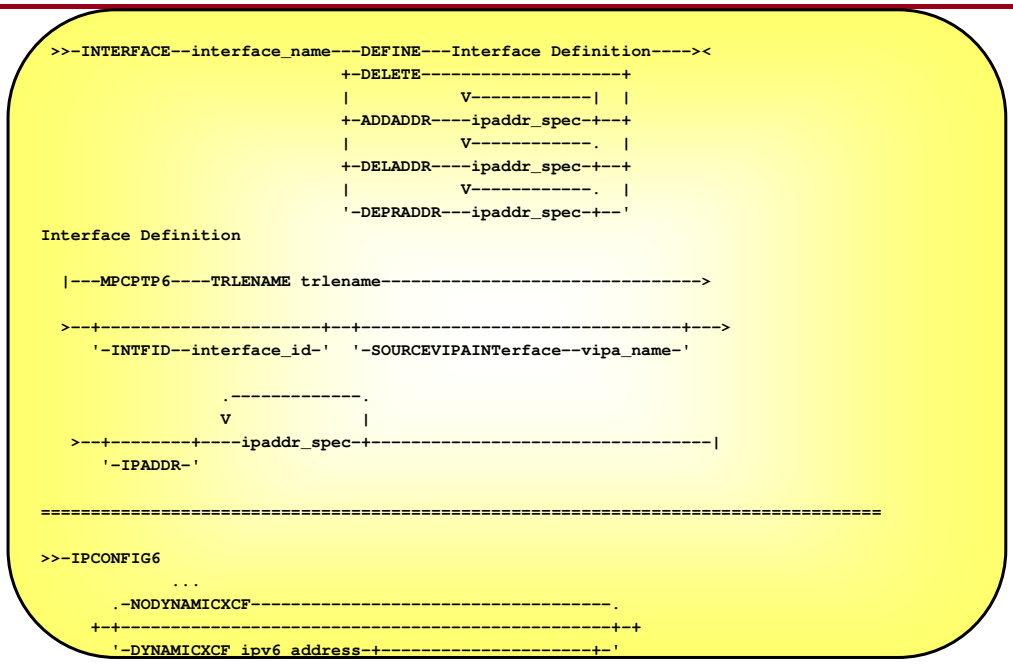
INTERFACE statement MPCPTP6

- Configure INTERFACE statement for MPCPTP6
  - TRLENAME keyword identifies the TRLE (or VTAM CPname for XCF, or reserved name IUTSAMEH)
  - To use MPCPTP for both IPv4 and IPv6, specify the same TRLENAME on the INTERFACE statement as the device_name on the DEVICE statement
    - A single TRLE definition can be shared by IPv4 traffic, IPv6 traffic, and SNA traffic
  - Optional INTFID keyword to specify interface ID
    - Allows for predictable link-local address
    - Otherwise interface ID is randomly generated
- Optional INTFID keyword also added to INTERFACE statement for IPAQENET6 (to override interface ID generated by OSA)
- Similar attributes to existing IPv6 support
  - INTERFACE statement options to:
    - Add/delete/deprecate addresses
    - Specify a VIPA for SOURCEVIPA
  - Use interface name for START/STOP and on static routes (BEGINROUTES)
  - Separate START/STOP of IPv4 and IPv6
  - Separate interface counters for IPv4 and IPv6

DYNAMICXCF for IPv6

- Use IPCONFIG6 DYNAMICXCF for dynamic IPv6 XCF and IUTSAMEH connectivity to other stacks in the sysplex
- TCP/IP automatically generates and activates IPv6 INTERFACE definitions (similar to IPCONFIG DYNAMICXCF for IPv4 sysplex connectivity)
  - Interface name EZ6XCFnn for XCF (where nn is sysclone value)
  - Interface name EZ6SAMEMVS for IUTSAMEH
  - No HiperSockets IPv6 support
- Optional INTFID keyword to specify interface ID (otherwise interface ID is randomly generated)
- Existing IPCONFIG DYNAMICXCF is limited to IPv4 sysplex connectivity
  - XCF Dynamics could result in IPv4 HiperSockets and IPv6 XCF connectivity between two stacks
- Cannot mix static and dynamic IPv4 and IPv6 definitions for XCF or IUTSAMEH
  - IPv4 and IPv6 XCF links must be generated the same way to the same hosts - IPv4 Dynamic XCF + IPv6 Dynamic XCF OR IPv4 static XCF + IPv6 static XCF.
- Once the IPv6 Dynamic XCF address has been established, or enabled, it cannot be changed without stopping and restarting the TCP stack.

INTERFACE statement for MPCPTP6 and IPCONFIG6 for dynamic XCF

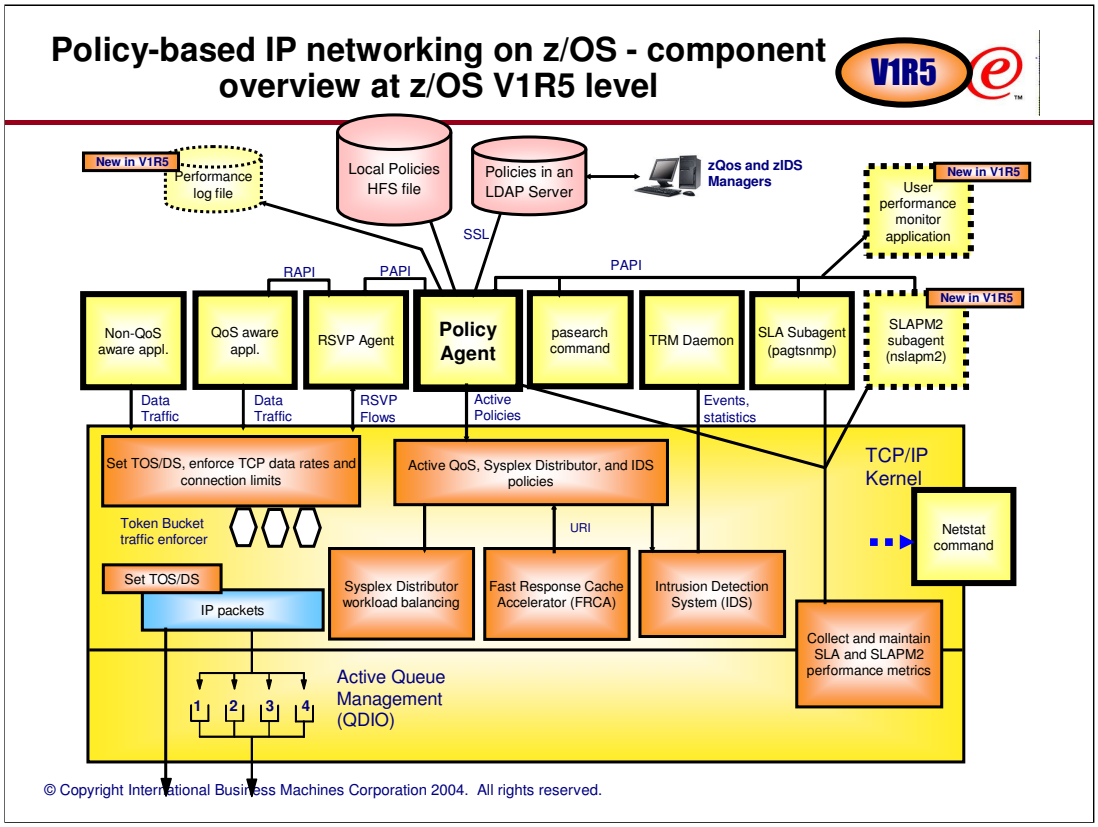


Netstat

- Netstat DEVLINKS/-d enhanced to display MPCPTP6 interfaces

```
DevName: IUTSAMEH          DevType: MPC
  DevStatus: Ready
  LnkName: LSAMEH            LnkType: MPC        LnkStatus: Ready
...
  IntfName: SAMEH6           IntfType: MPC6      IntfStatus: Ready
    NetNum:  0    QueSize: 0
    CfgMtu:  None                    ActMtu:    65535
    IntfID:  0010:0000:0003:0001
  Multicast Specific:
    Multicast Capability: Yes
    RefCnt      Group
    ------      -----
    0000000002  ff02::1:ff03:1
    0000000001  ff02::1
  Interface Statistics:
    BytesIn                          = 0
    Inbound Packets                  = 0
    Inbound Packets In Error         = 0
    Inbound Packets Discarded        = 0
    Inbound Packets With No Protocol = 0
    BytesOut                         = 144
    Outbound Packets                 = 6
```

Policy Agent support for IPv6 and a few other things



NOTES:

QoS-aware (Integrated Services) applications and non-QoS-aware (Differentiated Services) applications can both utilize QoS support in the stack

QoS-aware applications use the RSVP API (RAPI) to communicate with the RSVP Agent

Non-QoS-aware applications can pass data classification information dynamically

RSVP Agent communicates with other RSVP Agents on routers/hosts

RSVP Agent is supported as an end system only, not as a router

Policy Agent reads policies from local files and/or an LDAP server and installs them into the Policy Table in the stack

pasearch command displays active and inactive policies

Netstat command displays active QoS policy statistics

Policy Rule = Condition(s), Time period, and an Action

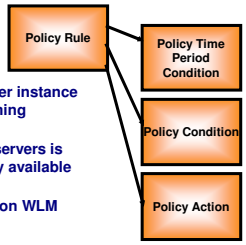
e-business

Classification - Conditions

- Source/Destination IP addresses (host's identification)
- Source/Destination port numbers (application identification)
- Protocol id (e.g., UDP, TCP, ICMP)
- Application name (use when port is not known)
- Application data (use for content based classification - used with Web URI or, in V1R2, with ancillary 'sendmsg()' data)
- Application priority (indicator sent by application)
- Routing - inbound/outbound interface/subnet
- Time periods when rule should be active

Sysplex Distributor Actions (DataTraffic)

- Determines to what target server instance within a Sysplex to route incoming connection request
- If none of the specified target servers is available, option to route to any available server
- Target server is chosen based on WLM and network QoS load status



Differentiated Services Actions (DataTraffic)

- TCP maximum/minimum rate - cwnd/srtt
- Setting ToS/DSCP - Type of Service/Differentiated Services Code Point
- Map ToS/DSCP to appropriate QDIO queue
- Map VLAN Priority to VLANs
- Number of concurrent TCP connections allowed
- Token bucket - policing access bandwidth excess traffic is either dropped or transmitted with a different ToS/DSCP value

Integrated Services Actions (RSVP)

- Limit the number of RSVP flow reservations per node or per subnet/interface
- Limit how much bandwidth that can be reserved per flow
- Limit burst size per reservation
- Reservation over ATM subnet will activate an ATM VC with QoS parameters that are mapped from RSVP reservation parameters
- Token bucket is used to meter reserved traffic

Intrusion Detection Services Actions and Traffic Regulation Actions (TR+)

- IDS: Attacks, Scans, Other Traffic Regulation (TCP, UDP, RAW)
  - Flooding prevention - a denial of service attack
  - Reporting, Logging, Notifying
- Manage total number of TCP connections per application - total connections allowed
- Manage number of TCP connections per client - percentage of remaining connections - Prevent greedy client(s) from monopolizing application and system resources
- Control action is either in Limiting (managing connection counts) and/or Logging (for problem analysis).
- Limit the number of RSVP flow reservations per node or per subnet/interface

Policy agent performance monitoring log file

- Prior to z/OS V1R5, there was no external API to collect the QoS performance data that can be used to see the state at which the QoS policies are operating.
- z/OS v1R5 adds a PAGENT configuration file statement to enable or disable policy performance data collection.
  - PolicyPerformanceCollection
- z/OS V1R5 also provides a C APIs that a user can utilize to get information about the performance of the QoS policies.
  - Sample C programs are provided to show how to utilize the APIs.
- PAGENT will in z/OS V1R5 log QoS performance data retrieved by the Policy Agent into log files.
- If policies are coded, performance data collection will always take place in the stack - the PolicyPerformanceCollection statement only enables retrieval of this data by Pagent

Policy agent performance monitoring log file
How to enable

- PolicyPerformanceCollection statement
  - PolicyPerformanceCollection {Enable | Disable}
  - Parameters:
    - DataCollection {Rule | Action}
      - type of performance data that needs to be collected (can have multiple types)
    - MinimumSamplingInterval minSampInt
      - smallest value, in seconds, that can be requested from an application, to retrieve performance data from the stack (default is 30); an algorithm is used to determine the actual interval
    - LogSamplingInterval logSampInt
      - interval, in seconds, at which the performance data will be retrieved from the stack and logged into the log file defined by PerformanceLogFile parameter
    - PerformanceLogFile logFile
      - name of the file to which the collected performance data should be written
    - SizeOfLogFile logFileSize
      - log file size, in kilobytes (default is 300)
    - NumberOfLogFiles numLogFiles
      - number of performance log files to be maintained (default is 3)

Policy agent performance monitoring log file
Improved NETSTAT SLAP report

- Commands: NETSTAT SLAP/netstat -j
  - Changed output to display the new performance data fields
- Policy name filter support has been added to the NETSTAT SLAP command
  - Netstat SLAP/-j POLicy/-Y policyname
  - Netstat SLAP/-j SUMmary

```
MVS TCP/IP NETSTAT CS V1R5       TCPIP Name: TCPCS
PolicyRuleName:  defaultRule
  FirstActTime:     03/05/2003 20:09:37
  LastMapTime:      03/05/2003 20:13:41
  TotalBytesIn:     63659
  TotalBytesOut:    1490
  TotalInPackets:   74
  TotalOutPackets:  38
  OutBytesInProf:   0
  OutPacksInProf:   0
  TotalBytesReTrn:  0
  TotalPacksReTrn:  0
  ReTrnTimeouts:    0
  AcceptConn:       2
  DeniedConn:       0
  ActConnMap:       1             Status:            Active
  SmoothRTTAvg:     72            SmoothRTTMdev:     83
  SmoothConnDlyAvg: 0             SmoothConnDlyMdev: 0
  AcceptQDelayAvg:  6             AcceptQDelayMdev:  0
```

Policy agent
Performance collection API (PAPI)

- Performance Collection APIs
  - Client library calls to connect, disconnect, get and free storage for QoS Performance Collection data from Policy Agent
  - Use an AF_UNIX connected socket to communicate with Pagent

API Name	Purpose
`int papi_debug(papiDebug_t debugValue)`	Allow debug information to be displayed for PAPI functions. Called by the application to either turn debug on or off during the PAPI processing
`int papi_connect(void **papiHandle, void *regReq)`	Used to open a connection and register with the Policy Agent. Most other PAPI functions will need the handle created here to be passed in as input
`int papi_get_perf_data(void *papiHandle, int typeFlag, void *filter, int *acceptableCachedTime,`	Used to retrieve the policy performance data from the Policy Agent. Format of the returned data is described later in this document.
`int papi_free_perf_data(void *perfDataHandle)`	Used to free the memory associated with the policy performance data returned by the **papi_get_perf_data()** API
`int papi_disconnect(void *papiHandle)`	Used to terminate a connection with the Policy Agent.

Policy agent
Performance collection API (PAPI) - *(continued)*

- Performance Collection APIs
  - Helper functions to access policy performance data:

API Name	Purpose
`int papi_get_policy_instance(void *perfDataHandle)`	Used to obtain the policy instance number for the set of policies in the policy performance data returned by the **papi_get_perf_data()** API
`int papi_get_rules_count(void *perfDataHandle)`	Used to obtain the number of rules in the policy performance data returned by the **papi_get_perf_data()** API
`int papi_get_actions_count(void *perfDataHandle)`	Used to obtain the number of actions in the policy performance data returned by the **papi_get_perf_data()** API
`RulePerfInfo *papi_get_rule_perf_info(void *perfDataHandle, int ruleNum)`	Used to obtain the performance information on a particular rule. Format of this information is described later in this document
`ActionPerfInfo *papi_get_action_perf_info(void *perfDataHandle, int actionNum)`	Used to obtain the performance information on a particular action. Format of this information is described later in this document
`char *papi_strerror(int papiReturnCode)`	Used to obtain a string describing a PAPI return code value.

Note: See *z/OS CS IP Programmer's Reference* for more detailed PAPI documentation

Policy agent
PAPI and performance log file details

- Performance Collection APIs
  - These APIs allow you to do near-real time performance analysis
  - Only available as a C API (i.e. not available in assembler)
  - PAPI return codes defined in papiuser.h
    - Must be included in client application
    - Stored in /usr/include
  - Refer to the C Sample file /usr/lpp/tcpip/samples/pagent/pCollector.c for a more detailed example use of these APIs
    - README in the same directory shows how to build the sample
    - Obtains performance data and displays it to the user in a readable format
- Policy performance log file
  - This log file allows you to do offline performance analysis
  - Information received from the stack will be written to this file in the same structure in which it is received
  - Logged in binary format
  - Stack name is appended to the filename defined by the PerformanceLogFile parameter in the PolicyPerformanceCollection statement
  - Data is logged based on the interval defined by the LogSamplingInterval parameter in the PolicyPerformanceCollection statement
  - C Sample file in /usr/lpp/tcpip/samples/pagent/pLogReader.c
    - README in the same directory shows how to build the sample
    - Displays the binary performance data to the user in a readable format

Policy agent - performance log file formatter pLogReader sample output - notes

NOTES

➢pLogReader sample output:

```
                        time:     04/30/03-15:12:44
                     version:                     1
                 policy name:           defaultRule
                 record type:                     1
                   record id:                     1
           bytes transmitted:               1072470
         packets transmitted:                   744
          active connections:                     1
        accepted connections:                     2
            smoothed rtt avg:                    13
           smoothed rtt mdev:                     8
         bytes retransmitted:                     0
       packets retransmitted:                     0
     smoothed conn delay avg:                     0
    smoothed conn delay mdev:                     0
      accept queue delay avg:                     0
     accept queue delay mdev:                     0
packets transmitted in profile:                   0
  bytes transmitted in profile:                   0
            packets received:                   386
              bytes received:                 12949
packets transmitted timed out:                    0
            denied Connections:                   0
```

New SLAPM2 replaces SLAPM

- SLAPM-MIB (SLA subagent - pagtsnmp)
  - Missing information to help analyze network performance by a network manager
  - Complex and not easy to index, requiring subcomponent information on a per TCP connection basis
  - Counts maintained in words, that wrap quickly in high speed environment
  - Not IPv6 enabled
- NETWORK-SLAPM2-MIB (Network Slapm2 subagent - nslapm2)
  - Contains more counts and monitored values
  - Contains 64-bit counts to minimize wrapping
  - Easier to use then Pagtsnmp, since no subcomponent information is required on a per TCP connection basis
  - Reduces system overhead due to the restructure of the MIB and the interface for subagent to get performance data
  - Provides transparency in supporting either IPv4 or IPv6, since this MIB no longer keeps track of IP addresses

SLAPM2 and SLAPM2 MIBs

- nslapm2 Subagent
  - The Network SLAPM2 subagent (nslapm2) provides support for the Network Service Level Agreement Performance Monitor MIB (NETWORK-SLAPM2-MIB).
  - Maintain Statistics Table
    - Connects with Policy Agent to obtain policy rules
    - This MIB provides information on defined policy rules, and performance statistics for TCP and UDP connections that map to active policies.
  - Processes SNMP requests for defined NETWORK-SLAPM2-MIB objects.
  - Maintain Monitor Table
    - The subagent can monitor TCP connections. When monitoring entry is created, a set of gauges and counters related to the policy rule being monitored are maintained.
  - SNMP Traps
    - The monitor table entries can be configured to send NOT OK SNMP traps when a specified value related to the gauges goes above its 'high' threshold. The entries can also be configured to send OK traps when a specified value goes below its 'low' threshold.
  - SNMP traps can also be sent when a monitored entry or statistics entry is deleted.

- NETWORK-SLAPM2-MIB Objects
  - This is a non-standard MIB
  - This MIB is shipped in z/OS Communications Server
    - /usr/lpp/tcpip/samples/slapm2.mi2

Extend policy definitions and operations to support IPv6

- IPv6 support in Policy Agent in z/OS V1R5:
  - IPv6 source and destination IP addresses are allowed to be specified in policy rules (LDAP and configuration files).
  - Interfaces in policy rules and subnet priority TOS masks are allowed to be specified by name.
    - Allowed for both IPv4 and IPv6 interfaces
    - IPv6 interfaces MUST be specified by name
  - "TOS" in policy definitions means IPv4 Type of Service or IPv6 Traffic Class.

- IPv6 is NOT supported by Policy Agent in z/OS V1R5 for the following:
  - Policy version 1
    - This version of policy is not being enhanced.
  - Intrusion Detection Services (IDS) policies
    - IDS does not support IPv6.
  - RSVP Agent
  - SLAPM MIB subagent (pagtsnmp)
    - New SLAPM2 MIB subagent (nslapm2) does not use IP addresses. The new subagent can still be used for IPv6 traffic (but the old one cannot).
  - LDAP server connection
    - LDAP client API does not support IPv6.
  - Pagent-to-Pagent communication for Sysplex Distributor functions
    - Sysplex Distributor does not support IPv6 in z/OS V1R5

Policy agent configuration change overview

- **LDAP Server**
  - **IPAddressRange attribute**
    - Existing formats for IPv4 addresses:
      - ibm-sourceIPAddressRange:2-ipv4address-prefixmask
      - ibm-sourceIPAddressRange:3-ipv4address1-ipv4address2
      - (same for destination)
    - New formats for IPv6 addresses:
      - ibm-sourceIPAddressRange:4-ipv6address-prefixmask
      - ibm-sourceIPAddressRange:5-ipv6address1-ipv6address2
      - (same for destination)
  - **SubnetAddr attribute**
    - Existing formats for IPv4 addresses:
      - ibm-interface:1-inboundipv4address-outboundipv4address
      - SubnetAddr:ipv4address
    - New formats for IPv6 interfaces:
      - ibm-interface:3-inboundname-outboundname
      - SubnetAddr:interfacename
- **Flat file**
  - **DestinationAddressRange attribute**
    - Existing formats for IPv4 addresses:
      - SourceAddressRange ipv4address1 [ipv4address2]
      - (same for destination)
    - New formats for IPv6 addresses:
      - SourceAddressRange ipv6address1 [ipv6address2]
      - (same for destination)
  - **SubnetAddr attribute (for subnet priority TOS masks)**
    - Existing formats for IPv4 addresses:
      - InboundInterface ipv4address
      - OutboundInterface ipv4address
      - SubnetAddr ipv4address
    - New formats for IPv6 interfaces:
      - InboundInterface interfacename
      - OutboundInterace interfacename
      - SubnetAddr interfacename
  - **ReadFromDirectory statement, LDAP_SchemaVersion parameter**
    - Default changed from 2 to 3

Migration Concerns

- Automation for changed messages may be impacted.
- Tools/automation that operate on pasearch command output may be impacted.
- Default schema version for LDAP policies is now 3 instead of 2.
- New LDAP schema definitions must be installed on the LDAP server to use IPv6 in LDAP policies.
  - ***IP Configuration Guide*** documents which schema definition files need to be installed when migrating from previous releases.
  - Also refer to specific LDAP server documentation for additional details. For the z/OS LDAP server, see ***Security Server LDAP Server Administration and Use***.

Trademarks, Copyrights, and Disclaimers

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

IBM	CICS	IMS	MQSeries	Tivoli
IBM(logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.