

Communications Server z/OS V1R5 and V1R6 Technical Update

Network Management

© Copyright International Business Machines Corporation 2004. All rights reserved.



Topics

z/OS V1R5

- Netstat enhancements
- New network management interfaces
 - TCP/IP event notification
 - Real-time packet tracing and formatting
 - TCP connection initiation and termination notifications
 - Application data for TN3270 server and FTP event data
 - TCP/IP polling interface
 - TCP listeners (server processes)
 - TCP connections (detailed information about individual connections)
 - UDP endpoints
 - CS storage usage
 - Enterprise Extender management data

z/OS V1R6

- Re-vamped netstat documentation
- Netstat enhancements

Netstat in z/OS V1R5

Copyright International Business Machines Corporation 2004. All rights reserved.



Overview over Netstat changes in z/OS V1R5



- Enhance remaining Netstat reports to support IPv6 information and FORMAT LONG/SHORT option
- Add new host name filter to Netstat connection related reports
- Add existing IP address filter to Netstat BYTEINFO/-b report
- Add existing interface name filter to Netstat HOME/-h report
- Add interface statistics to Netstat DEVLINKS/-d report
- Miscellaneous Changes

Long format reports: room for IPv6 addresses



- Add the LONG format support to all of Netstat reports. The existing stack-wide output format option (FORMAT LONG/SHORT) configured on the IPCONFIG profile statement, or Netstat FORMAT/-M option, can be used to instruct all Netstat reports to produce output according to either the old or new format if the stack is not enabled for IPv6 processing.

- Following Netstat reports are enhanced to support two different formats of the reports by using the FORMAT/-M LONG/SHORT option:
 - D TCPIP,,NETSTAT,ACCESS,NETWORK (IPv6-enabled)
 - Netstat CACHINFO/-C
 - Netstat IDS/-k
 - Netstat VCRT/-V
 - Netstat VDPT/-O
 - Netstat VIPADCFG/-F
 - Netstat VIPADYN/-v

Message identifiers in TSO netstat are gone!



- For all TSO NETSTAT reports, the report data does not have message identifiers displayed when the LONG format report is requested or the stack is IPv6-enabled. Error messages will continue to have message identifiers.

- The following Netstat reports are not affected by IPv6 support. But for completeness, these reports are also changed to not display message identifiers when a LONG format report is requested or the stack is IPv6-enabled. Other than that, information displayed in the LONG format report is the same as was displayed in the previous releases.
 - ▶ Netstat ARP/-R (IPv6 doesn't use ARP)
 - ▶ Netstat CLIENTS/-e (report doesn't contain IP address information)
 - ▶ Netstat GATE/-g (report only contains IPv4 routing information)
 - ▶ Netstat SLAP/-j (report doesn't contain IP address information)

- For TSO NETSTAT HELP report, the report data does not have message identifiers displayed for both LONG and SHORT format reports.

Hostname filter



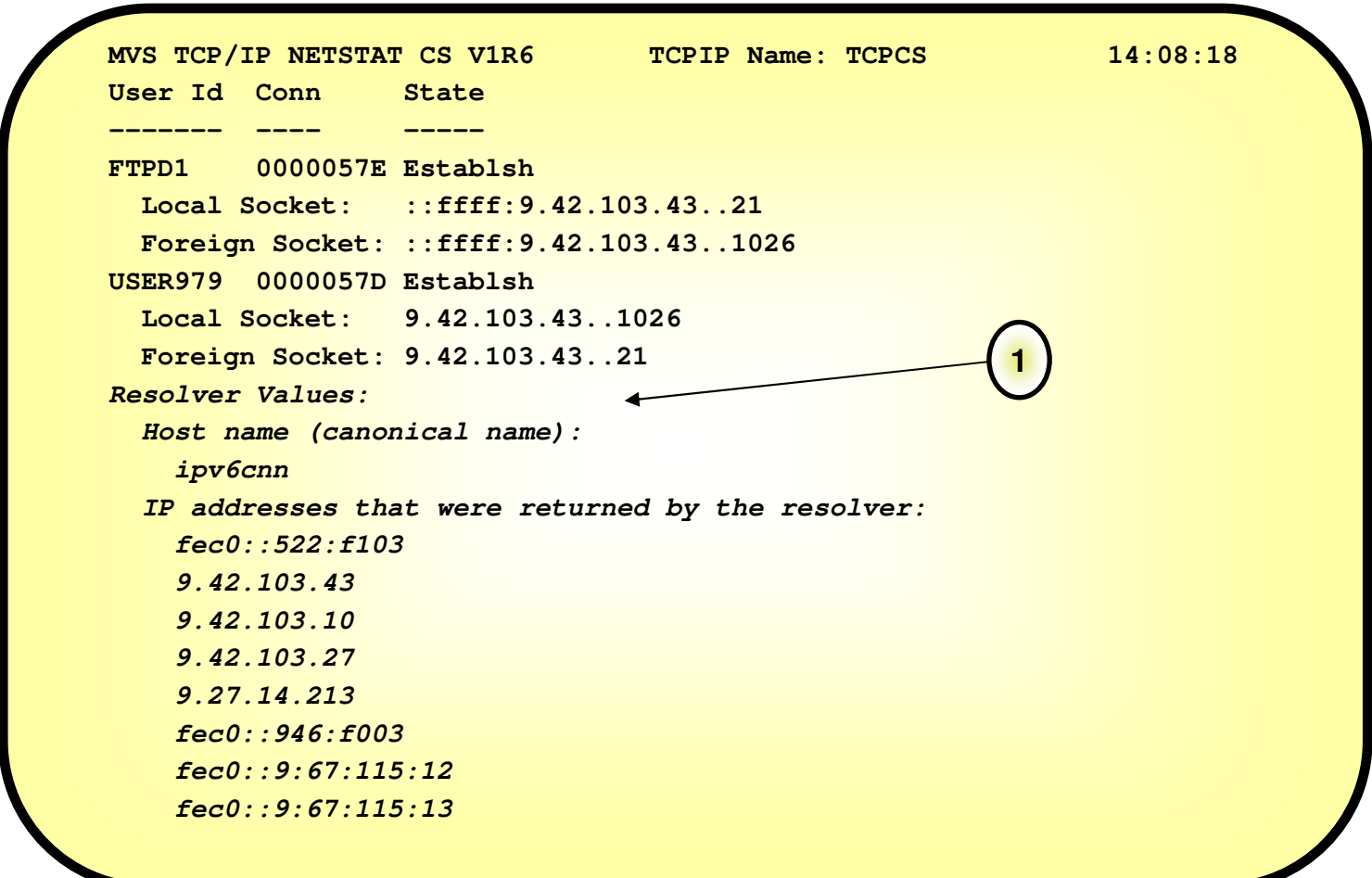
- Added host name filter support (HOSTName/-H) to connection oriented reports: ALL/-A, ALLCONN/-a, CONN/-c, BYTEINFO/-b, SOCKETS/-s, TELNET/-t and VCRT/-V.
- Allow only a single host name filter value to be specified at a time.
- The host name filter value will be passed to the resolver which will resolve it to one or more IP addresses. These IP addresses will be used as filters for the data on the report. If the host name can not be resolved, Netstat will issue the "Unknown host" message and stop the command process.
- In addition, at the end of the report, Netstat will display the host name that the resolver used for the resolution and the list of IP addresses returned from the resolver which were used as filters.
- The host name filter value does not support wildcard characters.
- Using HOSTName/-H filter may cause delays in the output due to resolution of the host name value, depending upon resolver and DNS configuration.

Example of report using the host name filter



➤ Filter the Netstat ALLCONN/-a report on HOSTName/-H *ipv6cnn*

```
MVS TCP/IP NETSTAT CS V1R6          TCPIP Name: TCPCS          14:08:18
User Id  Conn      State
-----  ----      -
FTPD1    0000057E  Establish
  Local Socket:  ::ffff:9.42.103.43..21
  Foreign Socket: ::ffff:9.42.103.43..1026
USER979  0000057D  Establish
  Local Socket:  9.42.103.43..1026
  Foreign Socket: 9.42.103.43..21
Resolver Values:
  Host name (canonical name):
    ipv6cnn
  IP addresses that were returned by the resolver:
    fec0::522:f103
    9.42.103.43
    9.42.103.10
    9.42.103.27
    9.27.14.213
    fec0::946:f003
    fec0::9:67:115:12
    fec0::9:67:115:13
```



Extend use of existing IP address filter to the BYTEINFO report



- Add support for the IPAddr/-I filter to the Netstat BYTEINFO/-b report.
- Up to six IP address filter values can be specified at a time.
- For an IPv6 enabled stack, both IPv4 and IPv6 IP address filter values are accepted and can be mixed on the IPAddr/-I option.
- For an IPv4 only stack, only IPv4 IP address filter values are accepted.
- Wildcard characters are not supported for IPv6 IP address filter values.

Extend use of existing interface name filter to the HOME list report



- Add support for the INTFNAME/-K filter to the Netstat HOME/-h report.
- Only one filter value can be specified at a time which can either be an IPv4 link name or an IPv6 interface name.
- The report will only include information for that particular interface/link. For an IPv6 interface, since multiple IP addresses can be defined for the same interface, all IP addresses for the specified interface will be included in the report.

Enhance the statistics information that is displayed as part of a DEVLINKS report



- The selected subset of interface statistics will always be displayed for all of interfaces or links except for VIPAs (these statistics are not maintained for VIPA links or interfaces).
- The existing BytesIn and BytesOut fields in the Netstat DEVLINKS/-d report are moved into the new statistics section for all interfaces or links except for VIPAs:
 - Programs that are screen-scraping Netstat DEVLINKS/-d reports for byte counters will have to be updated
- The new statistics section heading for an IPv4 link entry is called Link Statistics and for an IPv6 interface entry is called Interface Statistics.

Example of new statistics



➤ Netstat DEVLINKS/-d

```
MVS TCP/IP onetstat CS V1R5          TCPIP Name: TCPCS          12:55:20
DevName: OSAQDIO4                    DevType: MPCIPA
DevStatus: Ready
LnkName: OSAQDIOLINK                  LnkType: IPAQENET   LnkStatus: Ready
NetNum: 0   QueSize: 0   Speed: 0000000100
IpBroadcastCapability: No
CfgRouter: Non                       ActRouter: Non
ArpOffload: Yes                       ArpOffloadInfo: Yes
ActMtu: 1492
VLANid: 1260                          VLANpriority: Enabled
ReadStorage: GLOBAL (8064K)          InbPerf: Balanced
ChecksumOffload: Yes
BSD Routing Parameters:
MTU Size: 00000                       Metric: 00
DestAddr: 0.0.0.0                     SubnetMask: 255.255.255.192
Multicast Specific:
Multicast Capability: Yes
Group          RefCnt
-----
224.0.0.1     0000000001
Link Statistics:
BytesIn                = 11476
Inbound Packets        = 10
Inbound Packets In Error = 0
Inbound Packets Discarded = 0
Inbound Packets With No Protocol = 0
BytesOut               = 6707
Outbound Packets       = 10
Outbound Packets In Error = 0
Outbound Packets Discarded = 0
```

1

Miscellaneous changes to Netstat



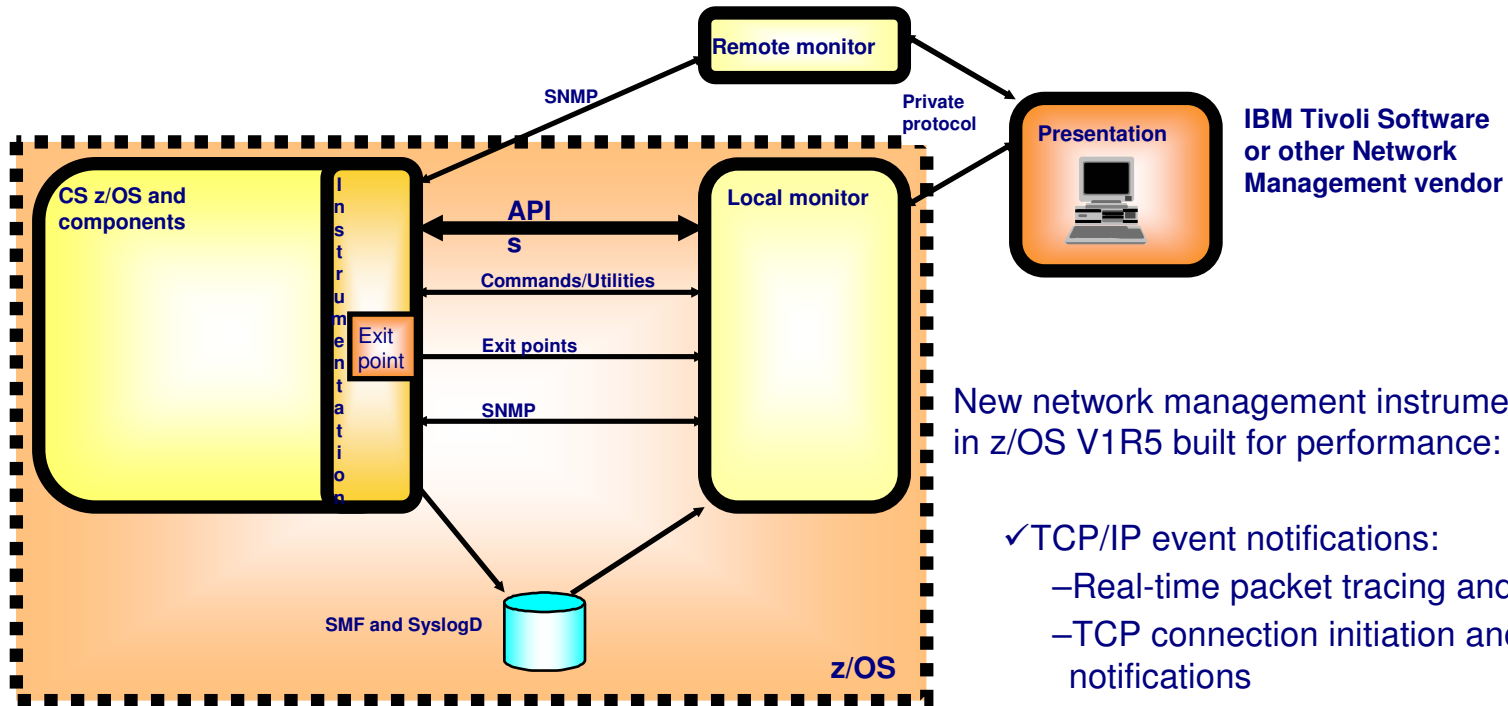
- In order to retrieve requested information from TCP/IP stack, Netstat needs to get enough private storage in the user's address space for the requested information. If the requested storage is low, Netstat may not be able to get storage to process a request. The request will fail but there is no external indication of the low storage problem.
 - Add a new message when Netstat can not get storage to retrieve requested information from TCP/IP stack.
 - EZZ2391I Cannot obtain storage to process option request
- The Netstat CONFIG/-f report currently displays the LogProtoErr field in the Configured TCP Information and Configured UDP Information sections. But the LogProtoErr can not be configured by users via TCPCONFIG and UDPCONFIG profile statements, and the stack never supported such information.
 - Remove the LogProtoErr field from the Configured TCP Information and Configured UDP Information sections of the Netstat CONFIG/-f report.
- The DELAYACKS information can be configured by users via TCPCONFIG profile statement but currently is not displayed under the Configured TCP Information section of the Netstat CONFIG/-f report.
 - Add the configured DELAYACKS information to the Configured TCP Information sections of Netstat CONFIG/-f report.

New Network Management Interfaces (NMI)

Copyright International Business Machines Corporation 2004. All rights reserved.



Network Management Instrumentation Overview



New network management instrumentation APIs in z/OS V1R5 built for performance:

- ✓ TCP/IP event notifications:
 - Real-time packet tracing and formatting
 - TCP connection initiation and termination notifications
 - Application data for TN3270 server and FTP event data
- ✓ APIs to poll information about currently active TCP/IP activity
 - TCP listeners (server processes)
 - TCP connections (detailed information about individual connections)
 - UDP endpoints
 - CS storage usage
- ✓ API to receive and poll for Enterprise Extender management data

Tivoli's IBM Tivoli Monitor / Network Performance (ITM/NP) use these new APIs

The APIs and their documentation will be available as part of CS z/OS for use by network management vendors and customer network management applications. APIs shipped for CS z/OS V1R4 as a PTF (UQ81245).

See info APAR II13699 for details.

Why do we need new network management interfaces for TCP/IP event notification?



➤ Existing network management information had various shortcomings:

Packet trace

- Packet trace data is unavailable except through an external writer or by processing a dump.

-TCP connection information

- Netstat screen-scraping is discouraged.
- Periodic SNMP polling of information (such as connection information) can have a significant impact on performance or on system load.
- An interface for notifying applications of ongoing connection activity was necessary to avoid the system load of polling.
 - Monitoring TCP connection SMF records was inadequate for this problem since that can exact a heavy system load:
 - ★ An application exit is called for every SMF record generated by a system.
 - ★ Monitoring frequent events requires the records for those events be activated, even if there is no desire to permanently record the SMF records.

-Application activity

- There is no existing means to obtain real-time information about the activity of key applications such as FTP and TN3270.

Note: download the documentation at this URL:

z/OS V1R4:

<ftp://ftp.software.ibm.com/s390/zos/commsserver/V1R4NMUG.pdf>

z/OS V1R5:

<ftp://ftp.software.ibm.com/s390/zos/commsserver/V1R5NMUG.pdf>

Basics of event notification interfaces



- All of these problems are amenable to similar solutions:
 - Buffer relevant information in TCP/IP stack.
 - Provide access to this information through an ongoing event notification interface for applications.

- All three interfaces buffer data within the TCP/IP stack
 - Packet trace: Buffer records describing monitored network packets. This includes not only packet trace but also data trace information collected by the TCP/IP stack.
 - Ongoing TCP connection information: Buffer type 119 SMF TCP init/term records
 - Subtype 1 - TCP connection initiation
 - Subtype 2 - TCP connection termination
 - Application activity: Buffer type 119 SMF FTP and Telnet records
 - Subtype 3 - FTP client transfer completion
 - Subtype 20 - TN3270 server SNA session initiation
 - Subtype 21 - TN3270 server SNA session termination
 - Subtype 22 - TSO telnet client initiation
 - Subtype 23 - TSO telnet client termination
 - Subtype 70 - FTP server transfer completion
 - Subtype 72 - FTP server login failure

- Applications obtain data by:
 - Connecting to an AF_UNIX socket made available for each interface:
 - Receiving "tokens" over that socket connection; and
 - Supplying these tokens to a new API call to copy buffers to application storage.

Configuration



- These functions must be enabled before use. This is accomplished using the NETMONitor statement in the TCP/IP PROFILE:

```
NETMONitor OFF
| ON
| [NOPKTTRCService|PKTTRCService]
| [NOTCPCONNService|TCPCONNService]
| [NOSMFService|SMFService]
```

- The status of this config option may be seen using the Netstat CONFIG/-f command.
- A parameter of OFF indicates all services should be inactive, while ON indicates that all services should be made active.
- If no options are specified on the NETMONitor statement, then NETMONitor OFF is assumed.
- If NETMONitor appears in a VARY TCPIP,,OBEYFILE, then the parameters will change the existing settings, e.g.:
 - NETMONITOR OFF - turn all functions off
 - NETMONITOR ON - turn all inactive functions on
 - NETMONITOR NOSMFS - turn off SMF service if active
 - NETMONITOR TCPCONNS - turn on TCP conn service if not active

Programming Interface



- Exploiters of this function (henceforth "clients") first issue a `connect ()` to connect to the appropriate AF_UNIX streams socket:
 - `/var/sock/SYSTCPDA.stackname` - packet trace data
 - `/var/sock/SYSTCPCN.stackname` - TCP connection data
 - `/var/sock/SYSTCPSM.stackname` - application event data

- After connecting:
 - Only the TCP connection interface expects the client to send a record indicating what types of data are requested.
 - The server will send an 'Init' record to the client to indicate that the connection was accepted and is active.

- The format of all records transferred over the interface is described in EZBYTMIA (for assembler) and EZBYTMIH (for C).

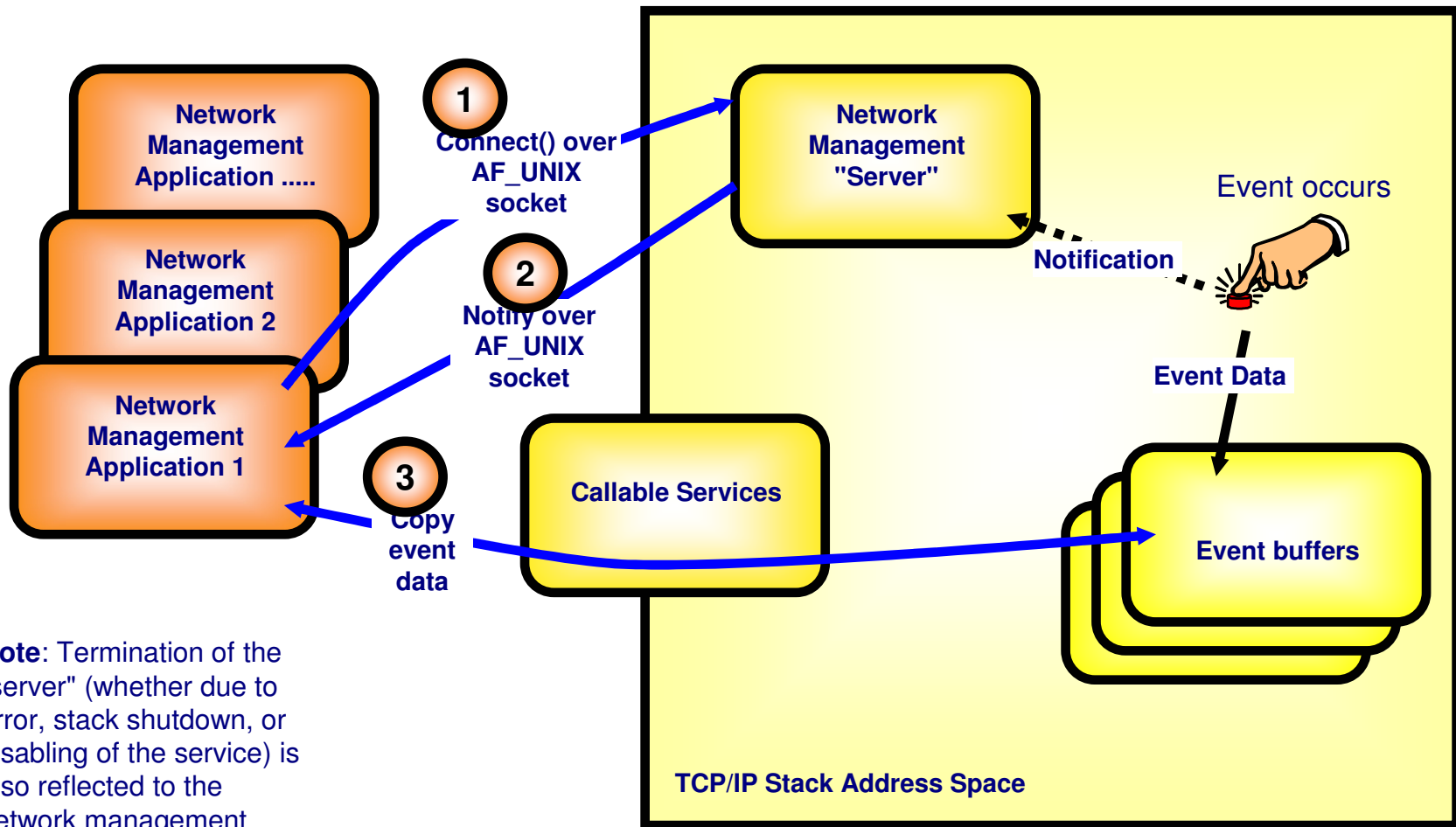
- The server will periodically send *token* records to the client. These tokens represent the actual data buffered for the client.

- The client will pass these tokens to the callable service EZBTMIC1 (TMI_CopyBuffer in C) to copy the buffered data to the client's private storage. The buffered data may be up to 64kB in size.

- One or more records of actual data is present in the buffer copied for the client. The format of these records will be described in detail in the informational APAR II13699.

- The server sends a 'Termination' record to the client when the connection is closed:
 - If the server is disabled using a VARY TCPIP,,OBEYFILE command;
 - If the client performs an error (*e.g.*, sending data to the server when it shouldn't); or
 - On any other error condition.

TCP/IP event notification to network management application



Note: Termination of the "server" (whether due to error, stack shutdown, or disabling of the service) is also reflected to the network management application by sending a notification over the AF_UNIX connection to the client.

Network management services can be protected through SERVAUTH profiles:
-EZB.NETMGMT.sysname.stackname.SYSTCPxx

Programming Interface - Notes



NOTES

- If the security resource EZB.NETMGMT.sysname.stackname.SYSTCPxx (where xx represents the given interface) is defined in the SERVAUTH class, then the userid of the client process must be permitted to this resource in order to access the function. If the resource is not defined, then the client process must be running as superuser in order to access the function.
- If access is denied then a termination record is sent to the client indicating the denial.

- The TCP connection interface requires the client to indicate whether it desires:
 - a list of TCP init records for pre-existing connections;
 - ongoing notification of TCP init and term records for new and terminated connections; or
 - both
- For the TCP connection interface, the server will not send the 'Init' record until the client's request record is received.
- Note that because of internal integration of the SYSTCPDA packet trace interface with the TCP/IP PKTTRACE function, only enabled PKTTRACE data will be reported over the SYSTCPDA interface, rather than all packets. However, the other two functions (SYSTPCPN, SYSTCPSM) do not require the corresponding SMF records to be enabled.

- Tokens are sent to the client either when a data buffer is filled, or when a buffer has remained inactive and unreported for several seconds.
- EZBTMIC1 supports client storage that is ALET-qualified and/or 64-bit.
- The format of the actual records in the data buffer differs for each interface. However, the records are arranged in the buffer in a common format, using a Component Trace Element (CTE) as a prefix to each record to indicate the length of the record (and thus the offset to the next record). The CTE is defined in ITTCTE (for assembler) and EZBYTMIH (for C).
- Because the TCP/IP stack stores the event data in a set of circular buffers, applications should be prompt to copy data after receiving a token. Otherwise the data may have been overwritten.

- Multiple applications are supported by this interface. In that case, all applications will receive the same event notifications.

Why do we need new polling-type network management interfaces?



- Network management applications which run locally on a z/OS system where they monitor TCP/IP activity and status need a high-speed, low-overhead interface to access data about TCP connections and UDP end-points.
 - SNMP protocols can be used to access such information, but adds processing overhead that in some situations has proven to be unacceptable, especially when obtaining information for a large set of end-points (e.g. walking the TCP connection table).
 - Some applications have attempted to parse the output from netstat to access the necessary information. This is error-prone and inefficient, and can put a severe strain on TCP/IP stack resources when done frequently.
 - An application that receives asynchronous events may need to use a polling interface to query (sample at selected intervals) the current status of one or more existing TCP or UDP connections to monitor their progress.
- The new network management callable API is a high-performance and low overhead polling interface that can return the following types of information at a given point in time:
 - **GetTCPListeners** - Information about all, or selected, active TCP end-points that listen for incoming connections ("servers").
 - **GetConnectionDetail** - Information about all, or selected, active non-listener TCP connections.
 - **GetUDPTable** - Information about all, or selected, active UDP end-points.
 - **GetStorageStatistics** - Information about TCP/IP stack usage of common and private storage.

The programming interface



- Exploiters of this function invoke the EZBNMIFR service routine, passing the following parameters:

```
CALL EZBNMIFR, (TcpipJobName,  
                RequestResponseBuffer,  
                RequestResponseBufferAlet,  
                RequestResponseBufferLength,  
                ReturnValue,  
                ReturnCode,  
                ReasonCode)
```

- The request buffer contains:
 - The header, which identifies the type of network management data to be returned (TCP listener connections, TCP non-listener connections, UDP connections, storage usage).
 - The filters, which identify a subset of connections. Each filter can contain any of the following elements:
 - ASID of the owning socket application address space.
 - Job name of the owning socket application address space.
 - Resource identifier ("Client ID" in NETSTAT displays).
 - Resource identifier of the related server listening connection.
 - Local IP address.
 - Local IP address prefix.
 - Local port.
 - Remote IP address.
 - Remote IP address prefix.
 - Remote port.

Notes



NOTES

- Callers of the EZBNMIFR service must execute:
 - In supervisor state, or in system key, with APF authorization, or as a superuser.
 - In either TCB mode or in SRB mode.
 - In either AMODE(31) or in AMODE(64).
 - In primary ASC mode.
 - With PASN=SASN=HASN.
- The format of the request/response buffer is described in EZBNMRHA (for Assembler programs) and EZBNMRHC (for C/C++ programs).
- Any of the following methods can be used to invoke the EZBNMIFR service:
 - Issue a LOAD macro to obtain the EZBNMIFR service entry point address, and then CALL that address. The EZBNMIFR load module must reside in a linklist dataset (e.g. TCP/IP's SEZALOAD load library), or in LPA.
 - Issue a LINK macro to invoke the EZBNMIFR service. The EZBNMIFR load module must reside in a linklist dataset (e.g. TCP/IP's SEZALOAD load library), or in LPA.
 - Link-edit EZBNMIFR directly into the application load module, and then CALL the EZBNMIFR service. Include SYS1.CSSLIB(EZBNMIFR) in the application load module link-edit.

Filtering support by polling function



Filter item	GetTCPLListeners	GetUDPTTable	GetConnectionDetail	GetStorageStatistics
ASID	yes	yes	yes	no
Resource name (Job name)	yes	yes	yes	no
Resource ID (conn ID)	yes	yes	yes	no
Server resource ID (conn ID of server)	no	no	yes	no
Local IP address	yes	yes	yes	no
Local IP address prefix	yes	yes	yes	no
Local port	yes	yes	yes	no
Remote IP address	no	no	yes	no
Remote IP address prefix	no	no	yes	no
Remote port	no	no	yes	no

Getlistener call - returned data per active TCP listening socket



```
typedef struct {
NWM_uint NWMTCPLIdent;          /* Identifier */
#define NWMTCPLISTENIDENTIFIER 0xD5E6D4E3 /* EyeCatcher "NWMTCPL" */
union {
struct sockaddr_in NWMTCPLLocalAddr4;
                                /* AF_INET address */
struct sockaddr_in6 NWMTCPLLocalAddr6;
                                /* AF_INET6 address */
} NWMTCPLLocal;                /* Local Address */
NWM_ushort NWMTCPLRsvd01;      /* Reserved */
NWM_ushort NWMTCPLAsid;        /* ASID */
char NWMTCPLResourceName.8.;   /* Resource name */
NWM_uint NWMTCPLResourceID;    /* Resource ID */
NWM_uint NWMTCPLSubtask;       /* Address of TCB in address space
                                that opened connection */
NWM_uint NWMTCPLAcceptCount;   /* Number connections accepted */
NWM_uint NWMTCPLExceedBacklog; /* Number connections dropped */
NWM_uint NWMTCPLCurrBacklog;   /* Current connections in backlog*/
NWM_uint NWMTCPLMaxBacklog;    /* Max backlogs allowed */
NWM_uint NWMTCPLCurrActive;     /* Number of current connections */
NWM_ull NWMTCPLStartTime;      /* Listener start time */
NWM_ull NWMTCPLLastActivity;   /* Last time connection processed*/
NWM_ull NWMTCPLLastReject;     /* Last time connection rejected
                                due to backlog exceeded */
} NWMTCPLListenEntry ;
```

Enterprise Extender management



- Network management information for Enterprise Extender (EE) is not very extensive
 - While there are a number of network management tools available for monitoring and for problem determination with TCP/IP and SNA, there is nothing available specifically to assist with EE network management.

- Network management information for High Performance Routing (HPR) exists, but is not easily processed
 - The DISPLAY NET, ID=*rtpname*, HPRDIAG=YES command does provide some HPR statistics for a given HPR connection (added in z/OS V1R4).
 - However, operator must still issue command periodically, and someone must parse and sort the output, to get useful monitoring information.

- Network management information for Common Storage Management (CSM) exists, but is not easily processed
 - The DISPLAY NET, CSM command provides data currently.
 - Performance Monitor Interface (PMI) processing provides application interface to collect the data, but requires monitoring application to open an ACB to operate.

- Rather than require operators or customers to collate the existing data on their own, and rather than require ACB overhead, a new interface is provided for acquisition of information about EE, HPR, and/or CSM.

- The management application can request the pertinent information at regular intervals and thus get a more complete picture of system usage.

- This new SNA Network Monitor Interface (NMI) is a polling interface, based on the AF_UNIX socket interface, for requesting information about Enterprise Extender, High Performance Routing, and Common Storage Management

Information that can be requested over this interface



- Enterprise Extender (EE) connection data: information about all EE connections or a desired set of EE connections as specified by the application using the local IP address or hostname and/or the remote IP address or hostname.
- Enterprise Extender summary data: information comprising a summary of EE activity for this host.
- High Performance Routing (HPR) connection data: information about specific HPR connections Rapid Transport Protocol physical units (RTP PUs) as specified by the application using either 1) the RTP PU name, or 2) the RTP partner CP name with an optional APPN COS specification. These RTP PUs are not limited to those using EE connections.
- Common Storage Manager (CSM) statistics: CSM storage pool statistics and CSM summary information.

Filtering support by EE request function



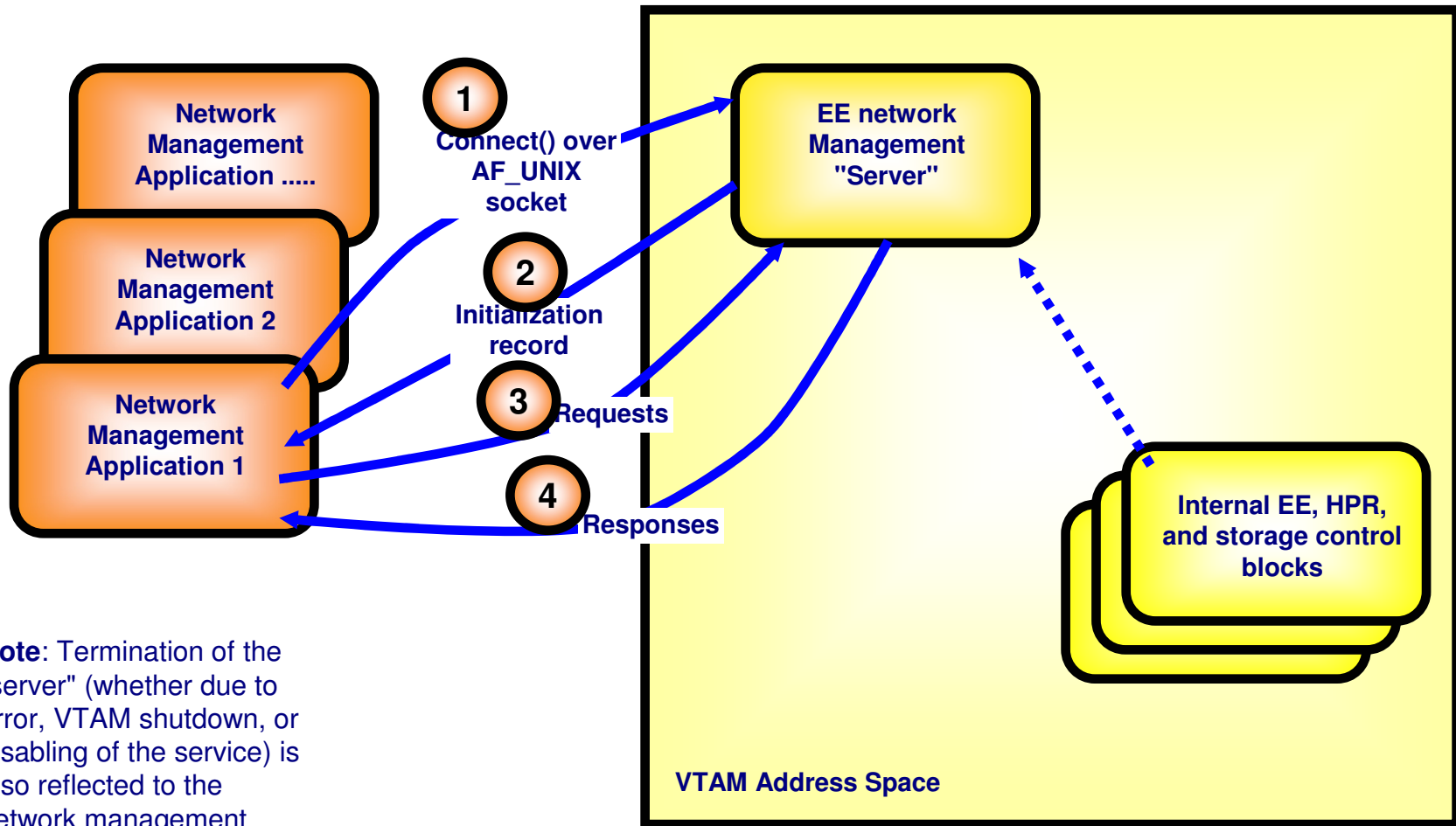
Filter item	EE connection request	EE summary request	HPR connection request	CSM storage request
Local IP address or hostname	Optional. Local hostname is ignored if local IP address is specified.	N/A	N/A	N/A
Remote IP address or hostname	Optional. Remote hostname is ignored if remote IP address is specified.	N/A	N/A	N/A
RTP PU name or partner CP name	N/A	N/A	One is required. Partner CP name is ignored if RTP PU name is specified.	N/A
COS name	N/A	N/A	Optional. Ignored if RTP PU name is specified.	N/A

Programming interface



- The z/OS system administrator may restrict access to this interface by defining the RACF (or equivalent external security manager product) resource `IST.NETMGMT.sysname.SNAMGMT` in the `SERVAUTH` class.
 - *sysname* represents the MVS system name where the interface is being invoked
- For management applications that use the interface, the MVS user ID is permitted to the defined resource.
- If the resource is not defined, then only superusers (users permitted to `BPX.SUPERUSER` resource in the `FACILITY` class and users with `UID=0`) are permitted to it.
- The administrator must define an OMVS segment for VTAM if one is not already defined.
- The VTAM OMVS user ID must have write access to the `/var` directory.
- The VTAM operator makes the SNA NMI function's `AF_UNIX` socket available for use by setting the new VTAM start option `SNAMGMT` to `YES` (via `START` command or `MODIFY VTAMOPTS` processing). The default for `SNAMGMT` is `NO`.
- Client applications then issue a connect to connect to this `AF_UNIX` streams socket:
 - `/var/sock/SNAMGMT`
- When an NMI client application connects to the SNA NMI server, it receives an "initialization" record with the following information (and more):
 - Request Type (i.e., initialization)
 - z/OS version number
 - z/OS functional capabilities

Enterprise Extender network management interface overview



Note: Termination of the "server" (whether due to error, VTAM shutdown, or disabling of the service) is also reflected to the network management application by sending a notification over the AF_UNIX connection to the client.

EE network management services can be protected through SERVAUTH profile:
-IST.NETMGMT.sysname.SNAMGMT

Netstat in z/OS V1R6

Copyright International Business Machines Corporation 2004. All rights reserved.



Extend port filter to netstat portlist report



- The Netstat PORTLIST/-o function is used to display reserved port list information defined via PORT and PORTRANGE TCP/IP profile statements.
- With a large number of ports reserved in a system, the Netstat PORTLIST/-o report can be large. But there is no support to display reserved port information for a particular port number.
- Add support for the PORT/-P filter to the Netstat PORTLIST/-o command.
- Up to six port number filter values can be specified at a time.
- netstat portlist (port 21 23

```
MVS TCP/IP NETSTAT CS V1R6          TCPIP Name: TCPCS          11:15:31
Port# Prot User      Flags   Range
-----
00021 TCP  FTPABC1  DA
00023 TCP  TCPCS    DAU
00023 TCP  MYINETD1 DABU
      BindSpecific: 9.42.104.161
```

Netstat route report changes



➤ Changes to the Netstat ROUTE/-r report:

- Add the configured DELAYACKS/NODELAYACKS setting for a route to the Netstat ROUTE/-r report when the DETAIL modifier is specified.
- Add the MTU size for IPv4 routes to both LONG and SHORT formats of the Netstat ROUTE/-r report when the DETAIL modifier is specified.
- Add the prefix length information for IPv4 routes to the SHORT format of the Netstat ROUTE/-r report to show the subnet mask information.

```
MVS TCP/IP NETSTAT CS V1R6          TCPIP Name: TCPCS          11:19:39
IPv4 Destinations
Destination      Gateway          Flags           Refcnt         Interface
-----
Default          9.42.105.65     UGO             000001        QDIO4
9.42.103.0/24    9.42.105.65     UGO             000000        QDIO4
9.42.103.11/32   0.0.0.0         UH              000000        TR1
```

```
MVS TCP/IP NETSTAT CS V1R6          TCPIP Name: TCPCS          11:21:01
IPv4 Destinations
Destination      Gateway          Flags           Refcnt         Interface
-----
9.42.103.0/24    9.42.105.65     UGO             000000        QDIO4
Metric: 00000007 MTU: 576
MVS Specific Configured Parameters:
MaxReTransmitTime: 120.000   MinReTransmitTime: 0.500
RoundTripGain:     0.125           VarianceGain:     0.250
VarianceMultiplier: 2.000           DelayAcks:         Yes
```

Netstat config report changes



➤ Changes to the Netstat CONFIG/-f report:

- Display some field values as Yes/No instead of 01/00.
- Change the field name NOUdpQueueLimit to UdpQueueLimit.

```
MVS TCP/IP NETSTAT CS V1R6          TCPIP Name: TCPCS          11:24:34

TCP Configuration Table:
DefaultRcvBufSize: 00065536  DefaultSndBufSize: 00065536
DefltMaxRcvBufSize: 00524288
MaxReTransmitTime: 120.000  MinReTransmitTime: 0.500
RoundTripGain: 0.125  VarianceGain: 0.250
VarianceMultiplier: 2.000  MaxSegLifeTime: 30.000
DefaultKeepALive: 00000120  DelayAck: Yes
RestrictLowPort: No  SendGarbage: No
TcpTimeStamp: Yes  FinWait2Time: 600

UDP Configuration Table:
DefaultRcvBufSize: 00008192  DefaultSndBufSize: 00008192
Checksum: Yes
RestrictLowPort: Yes  UdpQueueLimit: No

.....
```

Netstat devlinks report changes



➤ Changes to the Netstat DEVLINKS/-d report:

- Display a value of 'n/a' instead of '0' in the NetNum and QueSize fields for those interfaces and links to which the NetNum field does not apply.
- Display a value of 'MPCPTP' instead of 'MPC' in the DevType and LnkType for MPCPTP devices and links.
- Display a value of 'MPCPTP6' instead of 'MPC6' in the IntfType field for MPCPTP6 interfaces.
- Display a value of 'Packed/None' instead of 'Yes/No' in the CfgPacking fields, and display a value of 'Packed/None' instead of 'Packed/Unpacked' in the ActPacking fields for CLAW devices.

```
DevName: IUTSAMEH          DevType: MPCPTP
DevStatus: Not Active
LnkName: V4SAMEH           LnkType: MPCPTP       LnkStatus: Not Active
NetNum: n/a  QueSize: n/a
...
IntfName: V6SAMEH         IntfType: MPCPTP6    IntfStatus: Not Active
NetNum: n/a  QueSize: n/a
...

DevName: CLAW2            DevType: CLAW       DevNum: 0D10
DevStatus: Ready         CfgPacking: Packed ActPacking: Packed
LnkName: CLAW2LINK       LnkType: CLAW       LnkStatus: Ready
NetNum: n/a  QueSize: n/a
...
```

Netstat documentation



- The z/OS Netstat command has been enhanced almost every single release. Up to now, it has 26 report/functional options, 19 modifiers (additional keywords for report options), and 9 filters (select-strings for report options), so that Netstat can provide 54 different reports to display the network status of the local host, including information about TCP/IP connections, network clients, gateways, and devices, etc.
- The existing Netstat documentation in the IP System Administrator's Commands book is mainly divided into two sections:
 - The TSO NETSTAT command and its options
 - The z/OS UNIX onetstat/netstat command and its options
- Since Netstat has so many different options and filters which cover almost 200 pages of documentation, it is difficult for customers to find individual option information in the TSO and UNIX sections.
- Most of the Netstat options are supported for both TSO and UNIX environments, so the existing documentation format causes the same information (in Parameters and Examples sections) to be repeated in two places.
- Netstat reports produce a high amount of detailed information to the user. But since we have a number of customers who are not that familiar with TCP/IP concepts, they get confused and are often unable to interpret the information that is provided in the Netstat report.
- For some of the Netstat reports, we do not explain all of the fields displayed on the report.

➤ Restructure Netstat documentation

Netstat

....

TSO NETSTAT command output parsing considerations

....

Provide security product access to Netstat command

....

The TSO NETSTAT command syntax

Purpose

What is the command for.

Syntax

Command syntax diagram.

The z/OS UNIX netstat command syntax

Purpose

What is the command for.

Syntax

➤ Command syntax diagram.

The Netstat parameter overview

Report Options

ALL/-A

General high level description. Refer to its detailed information in the new Report details and examples section.

ALLConn/-a

....

Target

TCp/-p

....

Output

FORMat/-M

....

Filter

CLient/-E

....

Command

DRop/-D n

....

The Netstat report details and examples

General concepts that apply to multiple Netstat reports

TCP connection status

Diagram

Table

UDP socket status

....

Client ID or Connection number

....

Client name or User ID

....

Local IP address

....

Foreign/remote IP address

....

Local port

....

Foreign/remote port

....

Local socket

....

Foreign socket

....

Last touched time

....

Time stamp

....

Redirecting netstat output

(continued in the next page)

The Netstat report details and examples (*continued*)

Netstat ALL/-A report

Description of the report including when it is used

Syntax of the report option

Syntax diagram;

Description of the modifiers and filters if needed;

Command invocation samples for both TSO and UNIX shell versions

Examples

SHORT and LONG formats, with and without modifiers if any, with and without filters if needed

Comprehensive description for each field shown in the example

Netstat ALLConn/-a report

....

z/OS UNIX and TSO Netstat option comparison

....

- Added the documentation for each field displayed in the Netstat reports.

Documentation example for the Netstat COnn/-c report



NOTES

➤ Here is an example of the newly designed documentation for the Netstat COnn/-c report.

Purpose

Displays the information about each active TCP connection and UDP socket. COnn/-c is the default parameter.

TSO Syntax

```
>>--NETSTAT--COnn--|-----|--|-----|--|-----|---><
      |-Target-|  |-Output-|  |-(Filter-|
```

Target:

Provide the report for a specific TCP/IP address space by using **TCp tcpname**. Refer to the **Target** section in **The Netstat parameter overview** for more description about the **TCp** parameter.

Output:

The default output option is to display the output to the user's terminal. For other options, refer to the **Output** section in **The TSO NETSTAT command syntax** and **The Netstat parameters** sections for more information.

Filter:

```
      <-----+
|--|-CLient----|-clientname-|-----|--|
|
|
|-HOSTName----|hostname-----|
|
|
```

Documentation example for the Netstat COnn/-c report (*continued*)



NOTES

```
|
|
|          <-----+
|-IPAddr----|-ipaddr-----|
|          |-ipaddr/subnetmask-|
|          |-ipaddr/prefixLen--|
|
|
|-NOTN3270-----|
|
|          <-----+
|-Port-----|-portnum-|
```

z/OS UNIX Syntax

```
>>--netstat -c --|-----|--|-----|--|-----|--><
| -Target-| | -Output-| | -Filter-|
```

Target :

Provide the report for a specific TCP/IP address space by using **-p tcpname**. Refer to the **Target** section in **The Netstat parameter overview** for more description about the **-p** parameter.

Output :

The default output option is to display the output to the z/OS UNIX shell stdout. For other options, refer to the **Output** section in **The z/OS netstat command syntax** and **The Netstat parameters** sections for more information.

Documentation example for the Netstat CConn/-c report (*continued*)



NOTES

Filter:

```
<-----+
|--| -E -|clientname|-----|--|
|
| -H ---hostname-----|
|
| <-----+
|-- -I -|ipaddr-----|--|
|   |ipaddr/subnetmask-|
|   |ipaddr/prefixLen--|
|
| <-----+
|-- -P -|portnum|-----|--|
|
|-- -T -----|--|
```

Filter Description

CLient/-E *clientname*

Filter the output of the CConn/-c report using the specified client name *clientname*. You can enter up to six filter values and each specified value can be up to 8 characters long.

HOSTName/-H *hostname*

Filter the output of the CConn/-c report using the specified host name *hostname*. You can enter one filter value at a time and the specified value can be up to 256 characters long.

Notes:

1 The HOSTName/-H filter does not support wildcard characters.

Documentation example for the Netstat COnn/-c report (*continued*)



NOTES

2. At the end of the report, Netstat will display the host name that the resolver used for the resolution and the list of IP addresses returned from the resolver which it used as filters.
3. Using HOSTName/-H filter might cause delays in the output due to resolution of the hostname value depending upon resolver and DNS configuration.

IPAddr/-I *ipaddr*
ipaddr/subnetmask
ipaddr/prefixlength

Filter the report output using the specified IP address *ipaddr*, *ipaddr/subnetmask*, or *ipaddr/prefixlength*. You can enter up to six filter values. Each specified IPv4 *ipaddr* value can be up to 15 characters long and each selected IPv6 *ipaddr* value can be up to 45 characters long.

ipaddr

Filter the output of the COnn/-c report using the specified IP address *ipaddr*. For IPv4 addresses, the default subnet mask of 255.255.255.255 is used. For IPv6 addresses, the default prefixlength of 128 is used.

ipaddr/subnetmask

Filter the output of the COnn/-c report using the specified IP address and subnet mask *ipaddr/subnetmask*. The IP address *ipaddr* in this format must be IPv4 IP address.

ipaddr/prefixlength

Filter the output of the COnn/-c report using the specified IP address and prefix length *ipaddr/prefixlength*. For a IPv4 address, the prefix length range is 1-32. For an IPv6 address, the prefix length range is 1-128.

Notes:

1. The filter value *ipaddr* can be either the local or remote IP address
2. The filter value for an IPv6 address does not support wildcard characters.

Documentation example for the Netstat CConn/-c report (*continued*)



- 3 For an IPv6 enabled stack, both IPv4 and IPv6 ipaddr values are accepted and can be mixed on the IPAddr/-I option. For an IPv4 only stack, only IPv4 ipaddr values are accepted.
- 4 For an IPv6 enabled stack, an IPv4-mapped IPv6 address is accepted as a valid ipaddr value and will usually provide the same result as its IPv4 address does.

NOTN3270/-T

Filter the output of the CConn/-c report excluding TN3270 server connections.

PORt/-P portnum

Filter the output of the CConn/-c report using the specified port number *portnum*. You can enter up to six filter values.

Note:

The port number can be either a local or remote port.

The filter value for CLient/-E and IPAddr/-I can be a complete string or a partial string using wildcard characters. A wildcard character can be an asterisk (*), which matches a null string or any character or character string, at the same position. A wildcard character can be a ?, which matches any single character at the same position. For example, a string "searchee" matches with "*ar?he*", but the string "searhee" does not match with "*ar?he*". If you want to use the wildcard character on the IPAddr/-I filter, you must specify the value in the ipaddr format. The wildcard character is not accepted for the ipaddr/subnetmask or ipaddr/prefixlen format of IPAddr/-I values.

When you use z/OS UNIX netstat/onetstat command in a z/OS UNIX shell environment, care should be taken if you use a z/OS UNIX MVS special character in a character string. It may cause an unpredictable result. To be safe, if you want to use a z/OS UNIX MVS special character in a character string, the character string should be surrounded by single or double quotation marks. For example, to use an asterisk (*) in the IP address, 10.*.0.0 for the -I filter, issue the command as: netstat -c -I '10.*.0.0' or netstat -c -I "10.*.0.0".

NOTES

Documentation example for the Netstat COnn/-c report (*continued*)



NOTES

Command syntax examples

From TSO environment:

```
NETSTAT CONN
```

Display information for all active TCP connections and UDP sockets in the default TCP/IP stack.

```
NETSTAT CONN TCP TCPCS6
```

Display information for all active TCP connections and UDP sockets in TCPCS6 stack.

```
NETSTAT CONN TCP TCPCS8 (IPADDR 9.43.1.1 9.43.2.2
```

Display information for these active TCP connections and UDP sockets in TCPCS8 stack whose local or remote IP addresses match the specified filter IP address values.

```
NETSTAT CONN (PORT 2222 6666 88
```

Display information for those active TCP connections and UDP sockets in the default TCP/IP stack whose local or remote ports match the specified filter port numbers.

From UNIX shell environment:

```
netstat -c
```

```
netstat -c -p tcpcs6
```

```
netstat -c -p tcpcs6 -I 9.43.1.1 9.43.2.2
```

```
netstat -c -P 2222 6666 88
```

Documentation example for the Netstat COnn/-c report *(continued)*



NOTES

Report examples

Following examples are generated by using TSO NETSTAT command. Using the z/OS UNIX netstat command displays the data in the same format as the TSO NETSTAT command.

Not IPv6 enabled (SHORT format)

```
MVS TCP/IP NETSTAT CS V1R6          TCPIP NAME: TCPCS          17:40:36
User Id  Conn      Local Socket      Foreign Socket      State
-----  ----  -
FTPD1    0000003B 0.0.0.0..21      0.0.0.0..0         Listen
FTPD1    0000003D 9.37.65.146..21  9.67.115.5..1026   Establish
SYSLOGD1 00000010 0.0.0.0..514    *.*                UDP
...
```

IPv6 enabled or request for LONG format

```
MVS TCP/IP NETSTAT CS V1R6          TCPIP NAME: TCPCS          17:40:36
User Id  Conn      State
-----  ----  -
FTPD1    0000004A Listen
  Local Socket:  ::::21
  Foreign Socket: ::::0
FTPD1    00000052 Establish
  Local Socket:  ::ffff:9.67.115.5..21
  Foreign Socket: ::ffff:9.67.115.65..1026
SYSLOGD1 0000002C UDP
  Local Socket:  0.0.0.0..529
  Foreign Socket: *.*
...
```


Documentation example for the Netstat CConn/-c report (*continued*)



**N
O
T
E
S**

Report field description

User Id

Refer to the Client name or User Id in the General concept section for detailed description.

Conn

Refer to the Client ID or Connection Number in the General concept section for detailed description.

Local Socket

Refer to the Local Socket in the General concept section for detailed description.

Foreign Socket

Refer to the Foreign Socket in the General concept section for detailed description.

State

Refer to the TCP connection status and UDP socket status in the General concept section for detailed description.

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 NON-INFRINGEMENT. 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.

© Copyright International Business Machines Corporation 2005. All rights reserved.

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.