TCP/IP for CICS Systems Programmers

Alfred B Christensen – alfredch@us.ibm.com
IBM Raleigh, NC, USA

Session: 8281
Friday, March 4, 2011: 8:00 AM-9:00 AM
TCP/IP for CICS Systems Programmers

<table>
<thead>
<tr>
<th>Session number:</th>
<th>8281</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date and time:</td>
<td>Friday, March 4, 2011: 8:00 AM-9:00 AM</td>
</tr>
<tr>
<td>Location:</td>
<td>Room 205B (Anaheim Convention Center)</td>
</tr>
<tr>
<td>Program:</td>
<td>Application Architecture Development</td>
</tr>
<tr>
<td>Project:</td>
<td>CICS</td>
</tr>
<tr>
<td>Track:</td>
<td>Application Technologies and Architectures and CICS Systems Programming</td>
</tr>
<tr>
<td>Classification:</td>
<td>Technical</td>
</tr>
<tr>
<td>Speaker:</td>
<td>Alfred B Christensen, IBM</td>
</tr>
<tr>
<td>Abstract:</td>
<td>In the past, the CICS systems programmer had to consider and understand how SNA and VTAM can impact CICS end-user response time and availability. Similar issues now must be addressed in regards to TCP/IP. In this session the speaker will discuss TCP/IP as it relates to the CICS systems programmer. The speaker will review CICS and TCP/IP parameters and highlight areas that might require customization depending on your site’s TCP/IP network configuration. If your installation has, or is about to implement TCP/IP in CICS, come to this session and learn that there is much more to enabling TCP/IP in CICS than simply specifying TCP=YES in the SIT.</td>
</tr>
</tbody>
</table>
Trademarks, notices, and disclaimers

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

- Advanced Peer-to-Peer Networking®
- AIX®
- alphaWorks®
- AnyNet®
- AS/400®
- BladeCenter®
- Candle®
- CICS®
- DataPower®
- DB2 Connect
- DB2®
- DRDA®
- e-business on demand®
- e-business (logo)®
- ESCON®
- FICON®
- GDDM®
- GDPS®
- Geographically Dispersed Parallel Sysplex
- HiperSockets
- HPR Channel Connectivity
- HyperSwap
- i5/OS (logo)
- i5/OS®
- IBM eServer
- IBM (logo)®
- IBM®
- IBM System i5
- IBM System p5
- IBM System z5
- IBM System z9®
- IBM System z10
- IBM zEnterprise System
- IMS
- InfiniBand®
- IP PrintWay
- IPDS
- iSeries
- LANDP®
- Language Environment®
- M-Series®
- MVS
- NetView®
- OMEGAMON®
- e-business on demand®
- Open Power
- Operating System/2®
- Operating System/400®
- OS/2®
- OS/390®
- OS/400®
- Parallel Sysplex®
- POWER®
- POWER7®
- PowerVM
- PR/SM
- pSeries®
- zEnterprise
- Rational Suite®
- Rational®
- Redbooks
- Redbooks (logo)
- Systplex Timer®
- System i5
- System p5
- System x®
- System z®
- System z9®
- Tivoli (logo)®
- Tivoli®
- VTAM®
- WebSphere®
- xSeries®
- zSeries®
- z/Architecture
- z/OS®
- z/VSE
- * All other products may be trademarks or registered trademarks of their respective companies.

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

- Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both, and is used under license there from.
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- InfiniBand is a trademark and service mark of the InfiniBand Trade Association.
- Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- UN*X is a registered trademark of The Open Group in the United States and other countries.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
- IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

Notes:

- Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput 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 improvements equivalent to the performance ratios stated here.
- IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
- All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
- This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
- All statements regarding IBM’s future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
- Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
- Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

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

© 2011 SHARE and IBM Corporation
Agenda

- What is CICS Sockets and what is CICS Sockets Domain?
- Introduction to CICS Sockets Domain
- Introduction to CICS Sockets (aka IP sockets)
- What is hot and sizzling in the z/OS IP community?
- CICS IP activity – monitoring from the TCP/IP side

Note:
CICS Sockets Domain is part of CICS TS and provides support for HTTP(S), IIOP(S), IPIC, and ECI access to CICS transactions. CICS Sockets is part of the z/OS Communications Server and provides generalized sockets programming interface support for CICS transactions.

Disclaimer: All statements regarding IBM future direction or intent, including current product plans, are subject to change or withdrawal without notice and represent goals and objectives only. All information is provided for informational purposes only, on an “as is” basis, without warranty of any kind.
TCP/IP for CICS Systems Programmers

What is CICS Sockets and what is CICS Sockets Domain?
A CICS Sockets transaction has direct access to the TCP/IP socket and can issue native sockets calls to receive and send data over the socket. IPv6 is supported. Secure connections via AT-TLS. No restrictions in application layer protocol.

A CICS Sockets Domain transaction does not have direct access to the socket, but communicates with CICS Sockets Domain services to receive a request and to send a reply over a socket. Secure connections are supported via native system SSL calls. Restricted to supported application layer protocols.

Application-layer protocol is transparent to the enhanced CICS Sockets listener.

These services are based on the Sockets Extended sockets APIs (provided by Communications server)

Inbound and outbound connections, UDP and multicast support, IPv4 and IPv6

TCPIPServices represent the "application" layer protocols supported by CICS Sockets Domain

Inbound connections to supported application protocols and outbound from all except ECI.
z/OS Sockets programming interfaces

Sockets application programs or subsystems utilizing sockets APIs

CICS Sockets transaction

CICS Sockets Domain

Pascal

ESMACRO

ASMOCT

EZASOKET

EZASMI

UNIX Systems Services provided callable BPX sockets API

UNIX Systems Services provided Logical File System (LFS)

UNIX Systems Services and TCP/IP provided Physical File Systems (PFS) - AF_INET and AF_INET6

TCP/IP provided TCP/IP protocol stack

TCP/IP provided C sockets API

LE provided C/C++ sockets API

SUN 4.0

DFE

X11

DPI 2.0

SUN 3.9

NCS

X11

RFC 1006

SUN 4.0

XPG 4.2

DPI 1.2

X-WIN

CICS Sockets transaction

CICS Sockets Domain
An attempt at a comparison

<table>
<thead>
<tr>
<th>Attribute</th>
<th>CICS Sockets</th>
<th>CICS Sockets Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of use from a programmer perspective</td>
<td>Easy if you are a sockets programmer, otherwise very difficult</td>
<td>Easy if you are a CICS programmer</td>
</tr>
<tr>
<td>Development productivity</td>
<td>Low to medium</td>
<td>Very high if one of the CICS Sockets Domain application layer protocols can be used</td>
</tr>
<tr>
<td>Application layer protocol flexibility (message formats, code pages, interaction model, error processing, etc.)</td>
<td>Very high - this is the main reason for using CICS Sockets instead of CICS Sockets Domain – the user protocol needed is unique and not supported by CICS Sockets Domain</td>
<td>Low</td>
</tr>
<tr>
<td>Sysplex CICS transaction routing</td>
<td>Limited to CICS regions in an LPAR (sharing a TCP/IP stack)</td>
<td>No GIVE/TAKE Socket support, but DPL can be used across a Sysplex. Response must be sent from same CICS region into which the request arrived</td>
</tr>
<tr>
<td>IPv6 support</td>
<td>Yes</td>
<td>Yes from CICS TS 4.1</td>
</tr>
<tr>
<td>Web services support (SOAP, XML)</td>
<td>No specific support</td>
<td>Yes</td>
</tr>
<tr>
<td>Secure connections</td>
<td>Yes (via AT-TLS)</td>
<td>Yes (via native system SSL usage)</td>
</tr>
<tr>
<td>OTE support</td>
<td>Yes</td>
<td>Yes – generally supported with a few exceptions</td>
</tr>
<tr>
<td>Application control over socket options in use (KEEPALIVE, TCP_NODELAY, etc.)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>CICS as a client (outbound connections)</td>
<td>Yes</td>
<td>Yes for all services except ECI</td>
</tr>
<tr>
<td>Support for connectionless sockets (UDP including multicast)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Management (configuration), trace/debug, and monitoring integral part of CICS</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Standard client support</td>
<td>No</td>
<td>Yes (browsers, IIOP, etc.)</td>
</tr>
<tr>
<td>Connection persistence</td>
<td>Somewhat complicated – requires use of an iterative server design or home-written listener</td>
<td>Yes</td>
</tr>
<tr>
<td>Cost of high-volume transaction processing</td>
<td>Perceived lower</td>
<td>Perceived higher</td>
</tr>
</tbody>
</table>

IPv6 support: Yes from CICS TS 4.1
Web services support (SOAP, XML): No specific support
Secure connections: Yes (via AT-TLS)
OTE support: Yes – generally supported with a few exceptions
Application control over socket options in use (KEEPALIVE, TCP_NODELAY, etc.): Yes
Support for connectionless sockets (UDP including multicast): Yes
Management (configuration), trace/debug, and monitoring integral part of CICS: No
Standard client support: Yes (browsers, IIOP, etc.)
Connection persistence: Somewhat complicated – requires use of an iterative server design or home-written listener
Cost of high-volume transaction processing: Perceived lower
TCP/IP Considerations for the CICS Systems Programmer

Introduction to CICS Sockets Domain
Explanation of a few of the TCPIPService options

```
OVERTYPE TO MODIFY

CEDA ALTER TCPIPSERVICE( HTTP )
TCPIPSERVICE = HTTP
GROUP = SOCKETS
DESCRIPTION = ABC HTTP SERVER
URM = DFHWBAAX
PORTNUMBER = 05081
STATUS = Open
PROTOCOL = HTTP
TRANSACTION = CWXN
BACKLOG = 00020
TSQPREFIX =
IPADDRESS =
SOCKETCLOSE = No
MAXDATALEN = 000032
SECURITY
SSL = No
CERTIFICATE =

When a client connects to your service, it is according to the underlying application protocol expected to send a request for the service to process. If the client is in error and doesn’t send any input data after having connected, how long should your service wait before it closes the connection down? Leave this at No if you want to use persistent connections!

The TCP/IP port your service will operate on - value should be coordinated with your TCP/IP systems programmer to have him/her reserve that port in the TCP/IP profile for this purpose only (through port reservation or RACF SERVAUTH resource definitions)

Backlog is used to specify the maximum number of connections waiting in TCP/IP to be serviced by your service. If the backlog queue is full, new connection requests will be rejected until the backlog queue falls below this value again. This has nothing to do with how many concurrent connections your service can process at any point in time!

Note: Make sure your TCP/IP systems programmer has specified an SOMAXCONN value that supports the maximum backlog you want/need!

IP address is used to turn your service into a bind-specific server - only servicing connection requests that are received for this local IP address.

```

The TCP/IP port your service will operate on - value should be coordinated with your TCP/IP systems programmer to have him/her reserve that port in the TCP/IP profile for this purpose only (through port reservation or RACF SERVAUTH resource definitions)

Backlog is used to specify the maximum number of connections waiting in TCP/IP to be serviced by your service. If the backlog queue is full, new connection requests will be rejected until the backlog queue falls below this value again. This has nothing to do with how many concurrent connections your service can process at any point in time!

Note: Make sure your TCP/IP systems programmer has specified an SOMAXCONN value that supports the maximum backlog you want/need!
How do you make your CICS Sockets Domain services bind-specific?

There are two ways you can do it:

1. Specify the local IP address to bind to when defining your TCPIP service:

   OVERTYPE TO MODIFY                                        CICS RELEASE = 0650
   CEDA  ALter TCpipservice( HTTP     )
   TCpipservice   : HTTP                                                       GROup          : SOCKETS                      ...                              Backlog      ==> 00020              0-32767                                 TSqprefix    ==>>
   Ipaddress    ==> 9.42.104.161
   SOcketclose  ==> No                 No | 0-240000 (HHMMSS)                  Maxdatalen   ==> 000032             3-524288
   SECURITY
   SSL          ==> No                 Yes | No | Clientauth                   CErtificate  ==>
   PORT
   5081 TCP IMWEBSRV BIND 9.42.104.161 ; z/OS HTTP server
   5081 TCP CICSTS32 BIND 9.42.105.45 ; CICS HTTP service

2. Or have the TCP/IP systems programmer control it in the TCP/IP configuration data set (the TCP/IP Profile)

   It might be best to let TCP/IP systems programmer control it - by leaving the field empty in the CEDA panel or enter the IP address as 0.0.0.0
How do you decide which IP address your server is listening on?

- The easiest way is to use the netstat command from either TSO or the UNIX shell (or the MVS console).
  - I have noticed that the CEMT Q TCPIPS command sometimes display the wrong IP address

<p>| TSO: ALLCONN APPLDATA TCP TCPCS ( CLI CICSTS32 |
| MVS TCP/IP NETSTAT CS V1R11 TCPIP Name: TCPCS 13:22:46 |</p>
<table>
<thead>
<tr>
<th>User Id</th>
<th>Conn</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICSTS32 000000A4</td>
<td>Listen</td>
<td></td>
</tr>
<tr>
<td>Local Socket: 9.42.105.45..5081</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Foreign Socket: 0.0.0.0..0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application Data: DFHICICS1A CWXNHTTP HTTP ABC HTTP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CICSTS32 00000045</td>
<td>Listen</td>
<td></td>
</tr>
<tr>
<td>Local Socket: 0.0.0.0..5082</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Foreign Socket: 0.0.0.0..0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application Data: DFHICICS1A CIEPECI ECI CICS ECI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CICSTS32 00000047</td>
<td>Listen</td>
<td></td>
</tr>
<tr>
<td>Local Socket: 0.0.0.0..5083</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Foreign Socket: 0.0.0.0..0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application Data: DFHICICS1A CIRRIIOP IIOP CICS IIO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CICSTS32 00000048</td>
<td>Listen</td>
<td></td>
</tr>
<tr>
<td>Local Socket: 0.0.0.0..5084</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Foreign Socket: 0.0.0.0..0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application Data: DFHICICS1A CISSIPIC IPIC CICS IPI</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The services you did not make bind-specific - in this example ECI on port 5082, IIOP on port 5083, and IPIC on port 5084 show up in your netstat display with the local socket IP address as 0.0.0.0.
- They will receive connection requests that arrive on any of the IP addresses in the HOME list.
Establishing stack-affinity for UNIX System Services sockets (CICS Sockets Domain)

- Affinity for CICS Sockets Domain to one of more stacks in an LPAR can be established at an address-space level, and will be in effect for all CICS Sockets Domain access from that address space.

- The simplest way to establish stack affinity is to add a small job step to your CICS startup procedure.
  - Executing PGM=BPXTCAFF

- All CICS Sockets Domain activity from this address space will only use the TCPCS TCP/IP stack on this LPAR

- CICS Sockets activity is made stack-affinity via the TCPADDR keyword when defining your CICS entry to CICS Sockets (EZACICD macro or EZAC transaction)

```
//DFHSTART PROC START='AUTO',
// INDEX1='DCICS.CICSTS32',
// INDEX2='CICSTS.V3R2M0.CICS',
// REGNAM='1A',
// REG='64M',
// DUMPTR='YES',
// RUNCICS='YES',
// OUTC='*',
// SIP=T,
// TCPHLQ='CS390.CS1B.PRD'
/*
  /* Set affinity to TCPCS stack on mvs098
  */
  */
  Affinity EXEC PGM=BPXTCAFF,PARM='TCPCS'
  */
  .......
  //CICS EXEC PGM=DFHSIP,REGION=&REG,TIME=1440,
  // COND=(1,NE,CICSCNTL),
  // PARM='START=&START,SYSSIN'
```

Note: CICS Sockets stack-affinity is determined via CICS Sockets definitions in the CICS entry in the CICS Sockets configuration data set.
TCP/IP for CICS Systems Programmers

Introduction to CICS Sockets (aka IP sockets)
CICS Sockets overview

- Multiple listeners – each instance separately configurable
- Enhanced listener has no requirements on client input data
- Multiple listeners in many CICS regions can share listener port number
- User ID security
- SSL/TLS support by means of ATTLS
- Configuration file and transaction (EZAC)
- Operations transaction to start/stop individual listeners (EZAO)
- PLT-enabled start and termination
- Reusable subtasks
- OTE enabled
- IPv6 support
- UDP and multicast support

- CICS Sockets is a component of the Communications Server for z/OS, not CICS TS itself.
- It is a general-purpose sockets programming API to be used by CICS application programmers for implementing native (low-level) sockets communication in z/OS CICS transaction programs.
CICS entry in CICS Sockets configuration file - EZAC transaction

EZAC,ALTER,CICS

Overttype to Enter

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLID</td>
<td>CICS1A</td>
<td>APPLID of CICS System</td>
</tr>
<tr>
<td>TCPADDR</td>
<td>TCPCS</td>
<td>Name of TCP Address Space</td>
</tr>
<tr>
<td>NTASKS</td>
<td>100</td>
<td>Number of Reusable Tasks</td>
</tr>
<tr>
<td>DPRTY</td>
<td>010</td>
<td>DPRTY Value for ATTACH</td>
</tr>
<tr>
<td>CACHMIN</td>
<td>010</td>
<td>Minimum Refresh Time for Cache</td>
</tr>
<tr>
<td>CACHMAX</td>
<td>020</td>
<td>Maximum Refresh Time for Cache</td>
</tr>
<tr>
<td>CACHRES</td>
<td>005</td>
<td>Maximum Number of Resolvers</td>
</tr>
<tr>
<td>ERRORTD</td>
<td>CSMT</td>
<td>TD Queue for Error Messages</td>
</tr>
<tr>
<td>MSGSUP</td>
<td>NO</td>
<td>Suppress Task Started Messages</td>
</tr>
<tr>
<td>TERMLIM</td>
<td>000</td>
<td>Subtask Termination Limit</td>
</tr>
<tr>
<td>TRACE</td>
<td>YES</td>
<td>Trace CICS Sockets</td>
</tr>
<tr>
<td>OTE</td>
<td>NO</td>
<td>Open Transaction Environment</td>
</tr>
<tr>
<td>TCBLIM</td>
<td>00000</td>
<td>Number of Open API TCBs</td>
</tr>
<tr>
<td>PLTSDI</td>
<td>NO</td>
<td>CICS PLT Shutdown Immediately</td>
</tr>
<tr>
<td>APPLDAT</td>
<td>YES</td>
<td>Register Application Data</td>
</tr>
</tbody>
</table>

PF 3 END

To get APPLDATA in Netstat for CICS Sockets Sockets, you must specify YES to APPLDAT on the CICS entry.

CICS Sockets always uses one TCP/IP stack only - which one is specified with the TCPADDR keyword.
**Listener entry in CICS Sockets configuration file - EZAC transaction - screen 1 of 2**

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLID</td>
<td>CICS1A</td>
<td>APPLID of CICS System</td>
</tr>
<tr>
<td>TRANID</td>
<td>CSKL</td>
<td>Transaction Name of Listener</td>
</tr>
<tr>
<td>PORT</td>
<td>03001</td>
<td>Port Number of Listener</td>
</tr>
<tr>
<td>AF</td>
<td>INET</td>
<td>Listener Address Family</td>
</tr>
<tr>
<td>IMMEDIATE</td>
<td>YES</td>
<td>Immediate Startup Yes</td>
</tr>
<tr>
<td>BACKLOG</td>
<td>040</td>
<td>Backlog Value for Listener</td>
</tr>
<tr>
<td>NUMSOCK</td>
<td>100</td>
<td>Number of Sockets in Listener</td>
</tr>
<tr>
<td>ACCTIME</td>
<td>060</td>
<td>Timeout Value for ACCEPT</td>
</tr>
<tr>
<td>GIVTIME</td>
<td>000</td>
<td>Timeout Value for GIVESOCKET</td>
</tr>
<tr>
<td>REALTIME</td>
<td>000</td>
<td>Timeout Value for READ</td>
</tr>
<tr>
<td>RTYTIME</td>
<td>015</td>
<td>Stack Connection Retry Time</td>
</tr>
<tr>
<td>LAPPLD</td>
<td>INHERIT</td>
<td>Register Application Data</td>
</tr>
</tbody>
</table>

You specify if the listener is an IPv4 or an IPv6 listener (INET or INET6)

Verify parameters, press PF8 to go to screen 2
or ENTER if finished making changes

**To get APPLDATA in Netstat for this listener, specify YES or INHERIT (inherit from the CICS entry)**
### Listener Entry in CICS Sockets Configuration File - EZAC Transaction - Screen 2 of 2

**EZAC, ALTer, LISTENER (standard listener. screen 2 of 2) APPLID = CICS1A**

Overtypes to Enter

<table>
<thead>
<tr>
<th>Field</th>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINMSGL</td>
<td>==&gt; 004</td>
<td>Minimum Message Length</td>
</tr>
<tr>
<td>TRANTRN</td>
<td>==&gt; NO</td>
<td>Translate TRNID Yes</td>
</tr>
<tr>
<td>TRANUSR</td>
<td>==&gt; NO</td>
<td>Translate User Data Yes</td>
</tr>
<tr>
<td>SECEXIT</td>
<td>==&gt;</td>
<td>Name of Security Exit</td>
</tr>
<tr>
<td>GETTID</td>
<td>==&gt; NO</td>
<td>Get AT-TLS ID (YES</td>
</tr>
<tr>
<td>USERID</td>
<td>==&gt;</td>
<td>Listener User ID</td>
</tr>
</tbody>
</table>

Verify parameters, press PF7 to go back to screen 1
or ENTER if finished making changes

**PF 3 END  7 PREV  12 CNCL**
CICS Sockets program categories in CICS

**Remote Sockets Server**
- Connect
- Send/Receive

**Remote Sockets Client**
- Connect
- Send/Receive

**Connect**
- Send TRM

**Send/Receive**
- Connect

**Non-socket transaction initiation - 3270 terminal input, IC, batch input, etc.**
- Started via PLT or via EZAO transaction.

**Transaction Request Message (TRM):**
Please note that use of the Enhanced Sockets Listener removes the requirement for the client sending a transaction request message - in reality removing any requirements from the CICS Sockets infrastructure on the application-level protocol between the client and the server running in CICS.
Concurrent CICS Sockets server - overview

**Listener**
- Do Forever
  - Accept connection request
  - [Read TRM from client]
  - [Call security exit]
  - Givesocket
  - EXEC CICS START passing TIM
    - If errors, send err message
    - end

**Transaction Initiation Message**
- Listener socket descriptor number
- Listener TCP/IP ID
- TRM user data
- Remote client socket address

**Child Server**
- EXEC CICS Retrieve of TIM
- Takesocket
- [Send OK Message to client]
- Read request data from client
- Process request
- Send reply to client
- Close socket

**Client**
- Connect to listener
  - [Send TRM]
  - [Read OK/Error Message]
- Send Request
- Read Reply
- Close socket
Three ways to launch CICS transactions:
- Via a Transaction Request Message – standard listener
- Via a listener configuration option to associate listener instance (and port) with one specific CICS transaction code
- Via the listener security user exit, driven by the listener

With the last two options, data may be sent by the client in completely free format.
TCP/IP for CICS Systems Programmers

What is hot and sizzling in the z/OS IP community?
Information technology today is limited by the technology and architecture configurations available

- Business processes and the applications that support them are becoming more service oriented, modular in their construction, and integrated.
- The components of these services are implemented on a variety of architectures and hosted on heterogeneous IT infrastructures.
- Approaches to managing these infrastructures along the lines of platform architecture boundaries cannot optimize: alignment of IT with business objectives; responsiveness to change; resource utilization; business resiliency; or overall cost of ownership.
- Customers need a better approach: The ability to manage the IT infrastructure and Business Application as an integrated whole in a much simplified manner.
IBM zEnterprise System Overview

Connecting the pieces with zManager (aka. Unified Resource Manager)!

1 All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represents goals and objectives only.
IBM zEnterprise node with internal networks

Customer managed management network

zEnterprise Node

Intra Node Management Network (INMN)

Intra Ensemble Data Network (IEDN)

z196

OSM

OSX

zBX

BladeCenter rack

Customer managed data networks
IBM zEnterprise – OSA and Network Types

zEnterprise node

Customer external data network access OSA OSD

Firewall

Customer external data network

Intra Ensemble Data Network (IEDN) OSA OSX 10 GbE

Intra Node Management Network (INMN) OSA OSM 1 GbE (1000BaseT)

IEDN may extend to other zEnterprise nodes within an ensemble

© 2011 SHARE and IBM Corporation
Use of multiple VLANs on the IEDN – no routing, but “application layer gateways” between VLANs

Virtual servers that act as “application layer gateways” have, under zManager control, access to two VLANs.

These nodes “should” be configured with forwarding disabled. z/OS is by default prevented from routing between IEDN VLANs.
Sysplex Distributor support for DataPower

- Introduced in z/OS V1R11 Communications Server
  - DataPower Support in Firmware 3.8.1

- Allows Sysplex Distributor to load balance connections to a cluster of DataPower appliances that “front-end” a z/OS Sysplex environment (Tier 1)
  - Complements Sysplex Distributor support for back-end workflows (DataPower to z/OS – Tier 2)

- Sysplex Distributor and DataPower communicate over a control connection
  - Allows SD to have awareness of state and utilization levels of each DataPower instance
  - Facilitates TCP connection management and use of GRE to preserve client’s IP address visibility to DataPower
When do our z/OS customers believe they will need IPv6?

- The majority of z/OS customers do not know
  - Expectations are that it will be needed slightly earlier on other platforms than z/OS
- It is time to start thinking, learning, and preparing now!

Source: Survey conducted by ENS early 2009 among a selected set of customers (39 responses to this question)
IPv4 address usage since early 1993

- Projected Internet Assigned Numbers Authority (IANA) Unallocated Address Pool Exhaustion
  - Happened Feb 1, 2011

- Projected Regional Internet Registries (RIR) Unallocated Address Pool Exhaustion
  - August 2011

- z/OS Communications Server continues to focus on IPv6 standards currency
  - US DoD/NIST
  - IPv6 Forum

If you want to stay in business after 2011/2012, you’d better start paying attention! Do not worry too much; the sky isn’t falling – IPv4 and IPv6 will coexist for many years to come. Your applications need to be able to use both. If you write directly to the TCP/IP sockets layer, you need to start changing those applications.

What is the upper practical limit (the ultimate pain threshold) for number of assigned IPv4 addresses? Some predictions said 250,000,000 (250 million), others go up to 1,000,000,000 (one billion or one milliard).

Source: https://www.isc.org/solutions/survey
Source: http://penrose.uk6x.com/
How the IPv4 address space is managed

Is Doomsday approaching?


This is less than six months from now!!!!

z/OS Communications Server keeps the pace, adding required new IPv6 support...
Tracking RIR allocations

Projected RIR Unallocated Address Pool Exhaustion: 09-Aug-2011

So - what is IPv6?

- **IPv6 is an evolution of the current version of IP, which is known as IPv4**
  - Work on new IETF standard started in early 90's
  - Not backward compatible, but migration techniques defined

- **Today's IPv4 has 32 bit addresses**
  - Theoretical limit is around 4 billion addresses
  - Due to IPv4 address assignment structure and policies, the practical limit is less than 1 billion useable global addresses

- **IPv6 provides almost unlimited number of addresses**
  - IPv6 addresses are 128 bits
  - No practical limit on global addressability
  - Enough address space to meet all imaginable needs for a while
  - More addresses *cannot* be retrofitted into IPv4

- **Other improvements important, but to some extent secondary:**
  - Facilities for automatic configuration
  - Improved support for site renumbering
  - End to end IP security
  - Mobility with route optimization (important for wireless)
  - Miscellaneous improvements aimed at improving router performance
z/OS TCP/IP is a dual-mode TCP/IP stack

- A dual-mode (or dual-stack) TCP/IP implementation supports both IPv4 and IPv6 interfaces – and both old AF_INET and new AF_INET6 applications.

- The dual-mode TCP/IP implementation is a key technology for IPv4 and IPv6 coexistence in an internet.

- For AF_INET6 applications, the common TCP or UDP transport layer determines per communication partner if the partner is an IPv4 or an IPv6 partner - and chooses IPv4 or IPv6 networking layer component based on that.

- Raw applications make the determination themselves when they choose IPv4 or IPv6 raw transport.
IPv6 and CICS

- CICS TS 4.1 IPv6 enables the CICS Sockets Domain services
- CICS Sockets was IPv6 enabled a few years ago
- CICS transaction Gateway for z/OS is IPv6 enabled
Enable a CICS sockets listener for IPv6 connectivity

```plaintext
EZAC,DISPLAY,LISTENER (standard listener.  Screen 1 of 2)    APPLID = CICS1A

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLID</td>
<td>CICS1A</td>
<td>APPLID of CICS System</td>
</tr>
<tr>
<td>TRANID</td>
<td>LSN0</td>
<td>Transaction Name of Listener</td>
</tr>
<tr>
<td>PORT</td>
<td>06000</td>
<td>Port Number of Listener</td>
</tr>
<tr>
<td>AF</td>
<td>INET6</td>
<td>Listener Address Family</td>
</tr>
<tr>
<td>IMMEDIATE</td>
<td>YES</td>
<td>Immediate Startup Yes</td>
</tr>
<tr>
<td>BACKLOG</td>
<td>020</td>
<td>Backlog Value for Listener</td>
</tr>
<tr>
<td>NUMSOCK</td>
<td>050</td>
<td>Number of Sockets in Listener</td>
</tr>
<tr>
<td>ACCTIME</td>
<td>060</td>
<td>Timeout Value for ACCEPT</td>
</tr>
<tr>
<td>GIVTIME</td>
<td>000</td>
<td>Timeout Value for GIVESOCKET</td>
</tr>
<tr>
<td>REALTIME</td>
<td>000</td>
<td>Timeout Value for READ</td>
</tr>
<tr>
<td>RTYTIME</td>
<td>015</td>
<td>Stack Connection Retry Time</td>
</tr>
<tr>
<td>LAPPLD</td>
<td>YES</td>
<td>Register Application Data</td>
</tr>
</tbody>
</table>

Verify parameters, press PF8 to go to screen 2
Press ENTER or PF3 to exit

PF 3 END         8 NEXT          12 CNCL
```

INET6 instead of INET
IPv6-enabled transaction initiation message

```
*---------------------------------------------------------------*
* Transaction Initiation Message from CICS listener            *
*---------------------------------------------------------------*

01 CICS-listener-TIM.
   05 give-take-sd         pic 9(8) Binary value zero.
   05 lstn-asname         pic x(8).
   05 lstn-subtask        pic x(8).
   05 client-in-data      pic x(35).
   05 lstn-ote           pic x(1).
   05 sockaddr-in.
      10 sin-family      pic 9(4) Binary.
      10 sock-data       pic x(26).
      10 sock-sin-data redefines sock-data.
         15 sock-sin-port  pic 9(4) Binary.
         15 sock-sin-addr  pic 9(8) Binary.
         15 filler          pic x(20).
      10 sock-sin6-data redefines sock-data.
         15 sock-sin6-port  pic 9(4) Binary.
         15 sock-sin6-flowinfo pic 9(8) Binary.
         15 sock-sin6-addr  pic x(16).
         15 sock-sin6-scope pic 9(8) Binary.
      05 filler            pic x(68).
      05 client-indata2-len pic 9(4) Binary.
      05 client-indata2    pic x(1).
```
IPv6 enabled CICS Sockets child server transaction

*---------------------------------------------------------------*
* Receive TIM from the CICS Listener.                           *
*---------------------------------------------------------------*
move 160 to cleng.
extec cics retrieve
  into(CICS-listener-TIM)
  length(cleng)
end-exec.

if sin-family = 19 then
  move sin-family to ntop-family
  move 45 to addrlen
  Call 'EZASOKET' using soket-ntop
    ntop-family
    sock-sin6-addr
    startup-sin-addr
    addrlen
    errno
    retcode
  if retcode < 0 then
    move 'Ntop failed' to ezaerror-text
    perform write-ezaerror-msg thru
      write-ezaerror-msg-exit
  end-if
End-if

TPICICSS Startup parms: 01 CICSTS32 00043L 0019 01135 ::FFFF:9.42.104.232

Receive the TIM from the listener – specify a length that is at least


NTOP convert binary IPv6 address to printable text.

In this case, the client is really an IPv4 client.
Sysplex Distributor hot standby support

- Have a single target server to receive all new connection requests
  - While other target servers are active but not receiving any new connection requests
  - Automatically route traffic to a backup target server when the active target server is not available

- Enable using a new HOTSTANDBY distribution method
  - One preferred target
    - AUTOSWITCHBACK option - switch to the preferred target if it becomes available
    - No auto switch back if reason for original switch was health problems
    - Use a V TCPIP Quiesce and Resume sequence
  - One or more backup targets ranked in order of preference
  - A target is not available when:
    - Not ready OR
    - Route to target is inactive OR
    - If HEALTHSWITCH option configured – target is not healthy when
      - TSR = 0% OR
      - Abnormal terminations = 1000 OR
      - Server reported Health = 0%

VIPADEFINE DVIPA1
VIPADISTRIBUTE DISTMETHOD HOTSTANDBY
AUTOSWITCHBACK HEALTHSWITCH
DVIPA1 PORT nnnn
DESTIP XCF1 PREFERRED
DESTIP XCF2 BACKUP 50
DESTIP XCF3 BACKUP 100
Performance improvements for fast local sockets

- **Fast local sockets (FLS)**
  - Optimized path through TCP/IP
  - Bypassing the IP layer
    - Data placed on TCP send queue
    - Data is then moved to TCP receive queue
    - ACKs built and sent from receive side
  - Used when socket end-points are on same stack
  - Dynamic; no configuration required

- **Faster local sockets (Turbo FLS)**
  - Bypasses processing on both sending and receiving side
    - Data no longer placed on TCP send queue
    - Data is placed directly onto receive queue bypassing TCP inbound processing
    - Data no longer ACKed
  - Enabled automatically; no configuration changes
    - Reverts to fast local sockets if packet trace or AT-TLS is enabled
    - No impact for data trace
TCP/IP for CICS Systems Programmers

CICS Sockets – monitoring from the TCP/IP side
### Table 204. Registered application data - CONNECT

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-8</td>
<td>The component ID of the IP CICS socket interface. For an outbound IP CICS socket client, this data always comprises the characters EZACICSO.</td>
</tr>
<tr>
<td>9</td>
<td>Blank</td>
</tr>
<tr>
<td>10-13</td>
<td>The CICS/TS transaction identifier. This is the CICS/TS transaction ID that is assigned to the program that issued the CONNECT socket command.</td>
</tr>
<tr>
<td>14</td>
<td>Blank</td>
</tr>
<tr>
<td>15-21</td>
<td>The task number of the transaction identifier in bytes 10-13.</td>
</tr>
<tr>
<td>22</td>
<td>Blank</td>
</tr>
<tr>
<td>23-30</td>
<td>The user ID that is assigned to the transaction identifier in bytes 10-13.</td>
</tr>
<tr>
<td>31</td>
<td>Blank</td>
</tr>
<tr>
<td>32-35</td>
<td>The CICS system name where the transaction is running.</td>
</tr>
<tr>
<td>36-40</td>
<td>Blank</td>
</tr>
</tbody>
</table>
### Table 205. Registered application data - GIVESOCKET

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-8</td>
<td>The component ID of the IP CICS Socket listener. For the IP CICS Sockets listener, this data always comprises the characters EZACIC02.</td>
</tr>
<tr>
<td>9</td>
<td>Blank</td>
</tr>
<tr>
<td>10-13</td>
<td>The CICS/TS transaction identifier. This is the transaction ID that the listener expects to take the specified socket.</td>
</tr>
<tr>
<td>14</td>
<td>Blank</td>
</tr>
<tr>
<td>15-21</td>
<td>This data is the task number of the CICS task that gives the accepted socket to a child process.</td>
</tr>
<tr>
<td>22</td>
<td>Blank</td>
</tr>
<tr>
<td>23-30</td>
<td>The user ID to be assigned to the transaction identifier in bytes 10-13.</td>
</tr>
<tr>
<td>31</td>
<td>Blank</td>
</tr>
<tr>
<td>32-35</td>
<td>The CICS system name where the transaction is to be assigned.</td>
</tr>
<tr>
<td>36-40</td>
<td>Blank</td>
</tr>
</tbody>
</table>
### Table 207. TAKESOCKET

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-8</td>
<td>The component ID of the IP CICS Socket interface. For the IP CICS Sockets interface and listener, this data always comprises the characters EZACICSO.</td>
</tr>
<tr>
<td>9</td>
<td>Blank</td>
</tr>
<tr>
<td>10-13</td>
<td>The CICS/TS transaction identifier. This is the transaction ID that now owns the socket.</td>
</tr>
<tr>
<td>14</td>
<td>Blank</td>
</tr>
<tr>
<td>15-21</td>
<td>The task number of the transaction identifier in bytes 10-13.</td>
</tr>
<tr>
<td>22</td>
<td>Blank</td>
</tr>
<tr>
<td>23-30</td>
<td>The user ID that is assigned to the transaction identifier in bytes 10-13.</td>
</tr>
<tr>
<td>31</td>
<td>Blank</td>
</tr>
<tr>
<td>32-35</td>
<td>The CICS system name where the transaction is running.</td>
</tr>
<tr>
<td>36-40</td>
<td>Blank</td>
</tr>
</tbody>
</table>
Table 206. Registered application data - LISTEN

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-8</td>
<td>The component ID of the IP CICS socket interface. For the IP CICS sockets listener, this data always comprises the characters EZACICS0.</td>
</tr>
<tr>
<td>9</td>
<td>Blank</td>
</tr>
<tr>
<td>10-13</td>
<td>The CICS/TS transaction identifier. This is the CICS/TS transaction ID assigned to the EZACIC02 program or a user-designed listener transaction program.</td>
</tr>
<tr>
<td>14</td>
<td>Blank</td>
</tr>
<tr>
<td>15-21</td>
<td>The task number of the transaction identifier.</td>
</tr>
<tr>
<td>22</td>
<td>Blank</td>
</tr>
<tr>
<td>23-30</td>
<td>The user ID that is assigned to the transaction identifier in bytes 10-13.</td>
</tr>
<tr>
<td>31</td>
<td>Blank</td>
</tr>
<tr>
<td>32-35</td>
<td>The CICS system name where the transaction is executing.</td>
</tr>
<tr>
<td>36-40</td>
<td>Blank</td>
</tr>
</tbody>
</table>
The IBM z/OS Communications Server Network Utility Assistant

- There is a new tool available for download from the z/OS Communications Server web pages:

- The IBM z/OS Communications Server Network Utility Assistant tool is a TSO/ISPF front-end to the z/OS Communications Server TSO NETSTAT line-mode command.

```
*--------------------- z/OS V1R12 CS TCP/IP NETSTAT --------------------------*
Command ===>                                                                   Select a report option by number or name ==>  
  1 ALL          2 ALLConn      3 ARp          4 BYTEInfo     5 CLients     
  6 CONFig      7 CONN         8 DEVlinks     9 Gate         10 HOne       
 11 PORTList    12 ROUTe       13 SOCKETs     14 TELnet       15 UN       
 16 CACHinfo    17 SLAP        18 VIPADYN     19 VPADCFG     20 VCRT       
 21 VDPT        22 IDS         23 STATS       24 ND          25 SRCIP      
 26 DROP        27 TTLs        28 RESCache    29 DEFADDRT    30 TCP/IP     
 90 TN3270      91 CICSSock    92 FTP         93 CICSTS      

Enter optional command modifiers and selection filters:
Do you want to specify optional command modifiers   ==> N  (Y/N)              
Do you want to specify optional selection filters   ==> N  (Y/N)              

Enter optional TCP/IP stack name and general options:
Stack name    ==> TCPCS      Leave blank for default stack              
Interval      ==> 5          Seconds for interval display                
Report format ==> LONG       (Short/Long) Leave blank for stack-default  
Excl. TN3270  ==> N          (Y/N) Reply Y to exclude TN3270 connections 
Netstat debug ==> N          (Y/N) Reply Y to see debug messages from Netstat  
EZANS debug   ==> N          (Y/N) Reply Y to see debug messages from EZANS  

Enter required arguments for ARP and DROP commands:
ARP address   ==> ALL        ARP (specify an IPv4 address or ALL)        
Conn id       ==> DROP        DROP (Specify connection ID to drop)        
```
What you can do with APPLDATA in Netstat – CICS Sockets

- APPLDATA is identification data a sockets application can associate with a sockets end point.
- CICS Sockets uses that feature to associate CICS-specific identification data with sockets that are used by the CICS Sockets.
- APPLDATA can be displayed with netstat, it is included in TCP/IP SMF records, and in the Network Management API.

```
*----------------------------- MVS TCP/IP NETSTAT CS z/OS V1R10 -----------------------------*
Command ===>
Please enter optional selection criteria for CICS Sockets connection overview – or press END to continue without any selection criteria.

Remote IP address   ==>  CICS listener server port
Local IP address    ==>  CICS address space that owns socket
CICS Sockets server port  ==>  CICS assigned user ID
CICS address space name  ==>  CICS transaction identifier
CICS user ID         ==>  CICS internal task number
CICS transaction code==>  CICS name transaction assigned to
CICS task number     ==>  Listener, Given, Taken, Connect
CICS system name     ==>  CICS Sockets type
CICS Sockets type    ==>  

If you want a display of all your CICS Socket connections, leave all selection fields above blank.
```
APPLDATA socket states

- **Remote client**
  - Connect()
  - Accept()
  - GiveSocket()

- **Remote server**
  - CONNECT
  - Connected Socket

- **Listener transaction**
  - LISTENER
  - Listener socket
  - Connected Socket (given)

- **Child server transaction**
  - TAKEN
  - TakeSocket()
  - Connected Socket (taken)

**CICS**
CICS Sockets customized netstat displays

- The socket type may be:
  - Listener
    - listener socket
  - Given
    - Given by the listener, but not yet taken by the child server
  - Taken
    - taken and currently owned by the child server
  - Connect
    - outbound connection

```
*----------------------------- MVS TCP/IP NETSTAT CS z/OS V1R10 ------- Row 1 to 3 of 3 Command ===>                               ...
Command ===>
CICS Sockets overview
Line command: S Connection summary, P Ping remote address, L Listener details, and D Drop connection

<table>
<thead>
<tr>
<th>Socket</th>
<th>CICS</th>
<th>CICS Socket</th>
<th>CICS Tran</th>
</tr>
</thead>
<tbody>
<tr>
<td>S ConnID</td>
<td>status</td>
<td>ASName</td>
<td>ID</td>
</tr>
</tbody>
</table>

0000F7 Listen CICSTS32 CICT Listener 3001 0000131 CSKL CICSUSER
0000F8 Listen CICSTS32 CICT Listener 6000 0000132 LSN0 CICSUSER
0000FD Establish CICSTS32 CICT Taken 6000 0000134 SRV1 CICSUSER
```
**MVS TCP/IP NETSTAT CS z/OS V1R11**

**Command ==>

Details for CICS Sockets connection ID: 0000058E

Local address . . .: 9.42.105.45..6000
Foreign address . .: 9.42.104.161..1127
Connection status .: Establish
Last touched . . .: 23:17:35 (UTC)
Last touched . . .: 18:17:35 (LCL) Idle time . . . . .: 00:00:13

Security information

ATTLS protected . .: No Certificate userID.: N/A
FIPS 140 . . . .: N/A Mapping type. . . .: N/A
SSL/TLS protocol. .: N/A
CipherSuite. . . .: N/A

CICS information

CICS address space : CICSTS32 CICS system name. .: CICT
CICS tran code. . .: SRV1 CICS task number. .: 0000044
CICS tran user ID .: CICSUSER CICS Sockets type .: Taken
CICS local port . .: 6000

TCP flow information

TCP segments in . .: 4 TCP segments out. .: 2
TCP bytes in . . .: 5 TCP bytes out . . .: 50
Send data queued. .: 0 Receive data queued: 0
Current send window: 1,048,160 Current recv window: 131,067
Congestion window .: N/A Max segment size. .: 8,940
Total retransmits .: 0 Duplicate ACKs . . .: 0
TCP RTT (msec) . . .: 258.00 TCP RTT variance : 844.00
TCP NODELAY set . .: No KEEPALIVE set . . .: No
What you can do with APPLDATA in Netstat – CICS Sockets Domain

- CICS TS 3.2 also uses the APPLDATA feature to associate CICS-specific identification data with sockets that are used by the CICS Sockets Domain.
- APPLDATA can be displayed with netstat, it is included in TCP/IP SMF records, and in the Network Management API.

```
*---------------------------- MVS TCP/IP NETSTAT CS z/OS V1R10 -----------------------------*
Command ===> 

Please enter optional selection criteria for CICS sockets domain sockets – or press END to continue without any selection criteria.

  Remote IP address     ==> 
  Local IP address      ==> 
  Local port number     ==> TCP port number (Listening port)
  Connection direction  ==> In (listen / accept), Out (connect)
  Owning CICS APPLID    ==> CICS application ID that owns socket
  CICS transaction ID   ==> CICS trans that created socket
  CICS network protocol ==> ECI, HTTP, IIOP, IPIC, or USER
  Listener service name ==> Service name if listening socket
  IPCCONN name          ==> IPCCONN name
  APPLID of partner system ==> Partner APPLID for connected IPIC

If you want a display of all your CICS sockets domain sockets, leave the selection fields above blank.
```
CICS Sockets Domain customized netstat displays

*------------------------ MVS TCP/IP NETSTAT CS z/OS V1R10 ------- Row 1 to 5 of 5
Command ===>                               Scroll ===> PAGE

CICS Sockets domain overview

Line command: S Connection summary, P Ping remote IP address, L Listener information, and D Drop connection

<table>
<thead>
<tr>
<th>Socket</th>
<th>CICS</th>
<th>CICS</th>
<th>Dir</th>
<th>Tran</th>
<th>Proto-</th>
<th>Service</th>
<th>IPCONN</th>
<th>Partner</th>
</tr>
</thead>
<tbody>
<tr>
<td>000045</td>
<td>Listen</td>
<td>CICSTS32</td>
<td>CICS1A</td>
<td>In</td>
<td>CIEP</td>
<td>ECI</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>000047</td>
<td>Listen</td>
<td>CICSTS32</td>
<td>CICS1A</td>
<td>In</td>
<td>CIRR</td>
<td>IIOP</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>000048</td>
<td>Listen</td>
<td>CICSTS32</td>
<td>CICS1A</td>
<td>In</td>
<td>CISS</td>
<td>IPIC</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>0000A4</td>
<td>Listen</td>
<td>CICSTS32</td>
<td>CICS1A</td>
<td>In</td>
<td>CWXN</td>
<td>HTTP</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>000123</td>
<td>Establish</td>
<td>CICSTS32</td>
<td>CICS1A</td>
<td>In</td>
<td>CWXN</td>
<td>HTTP</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

************************************************************************** Bottom of data ****************************
CICS Sockets Domain customized netstat displays – socket details

*-------------------- MVS TCP/IP NETSTAT CS z/OS V1R10 -----------------------*

Command ==> 

Details for CICS Sockets domain connection ID: 00000123

Local address . . : 9.42.105.45..5081
Foreign address . . : 9.65.253.59..1539
Connection status . : Establish
Last touched . . : 18:46:01 (UTC)
Last touched . . : 13:46:01 (LCL)  Idle time . . . . : 00:01:55

CICS information

CICS address space : CICSTS32  CICS APPLID . . : CICS1A
CICS tran code. . : CWXN  CICS protocol . . : HTTP
CICS connection dir: In  CICS local port . : 5081
CICS IPCONN name. . : N/A  CICS partner APPLID: N/A
CICS service name . : HTTP  CICS service desc . : ABC HTTP

TCP flow information

TCP segments in . : 11  TCP segments out. : 10
TCP bytes in . : 2,725  TCP bytes out . : 2,488
Send data queued. : 0  Receive data queued: 0
Current send window: 65,535  Current recv window: 130,150
Congestion window . : N/A  Max segment size. : 536
Total retransmits . : 0  Duplicate ACKs . : 0
TCP RTT (msec) . : 30.00  TCP RTT variance : 478.00
TCP NODELAY set . : No  KEEPALIVE set . : No
# For more information

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

*For pleasant reading ....*