IBM Software Group

Using z/OS Communications Server TCP/IP in a Common INET (multistack) Environment

Clarissa Brown
Rick Armstrong
clarisab@us.ibm.com
rickied@us.ibm.com

WebSphere® Support Technical Exchange

ON DEMAND BUSINESS

Agenda

- Overview
- CINET vs INET - what are the differences
  - INET example
  - CINET example
- CINET pre-router and TCPIP
  - How they work together
  - What do I need to be aware of?
  - Means to control TCPIP and the pre-router
- Technotes and References

Overview

Taken from z/OS V1R7.0 Communications Server IP Configuration Guide...

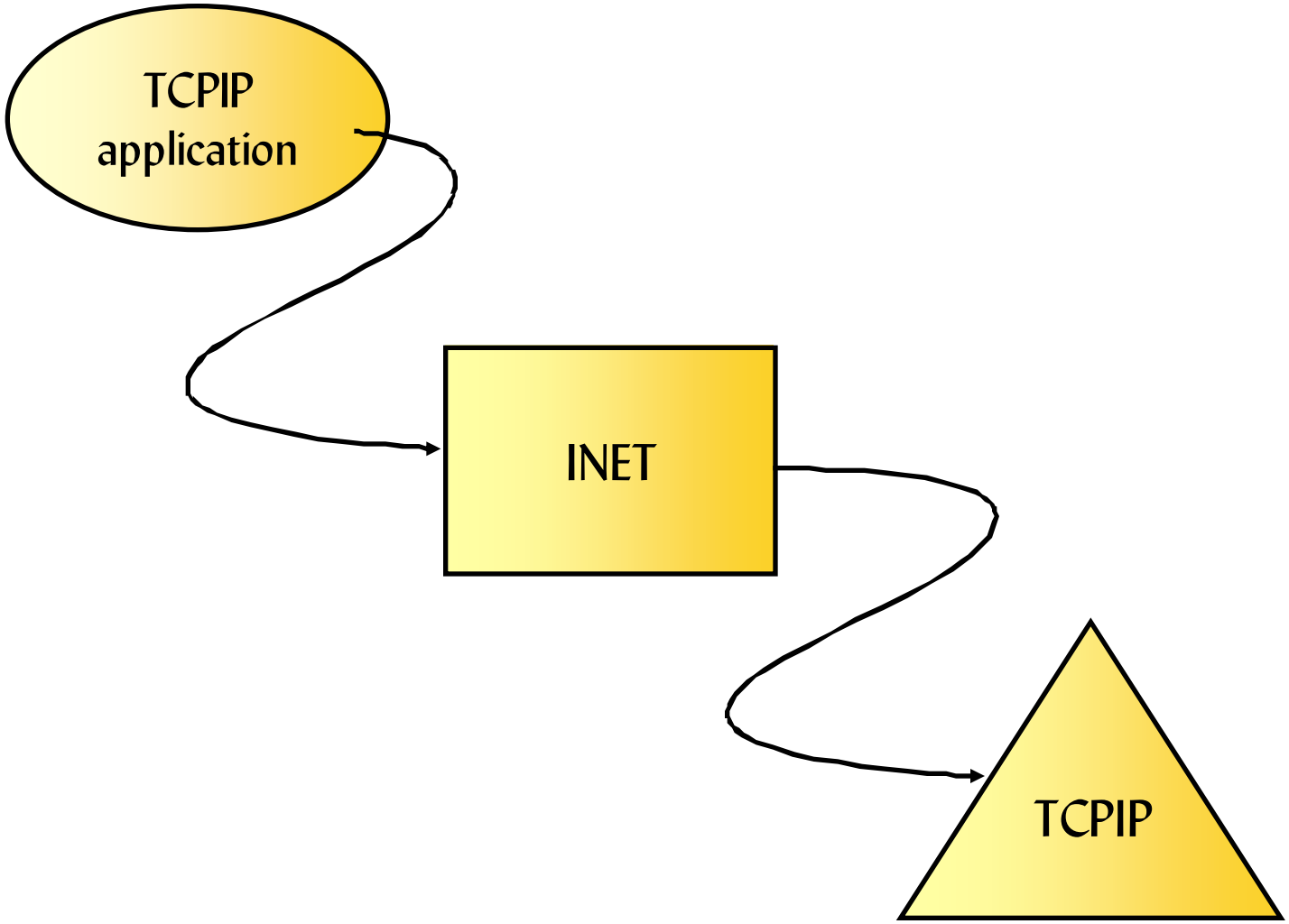
The z/OS Communications Server TCP/IP stack is a multiple-processor capable stack, which means that it can concurrently exploit all available processors on a system. Starting multiple stacks will not yield a significant increase in throughput.

In addition, running multiple z/OS Communications Server TCP/IP stacks requires additional system resources, such as storage, CPU cycles, and DASD. It also adds a significant level of complexity to the system administration tasks for TCP/IP.

For these reasons, it is suggested that in most cases you use the INET configuration, which supports a single TCP/IP stack. However, there are some special situations where running multiple stacks can provide a benefit. For example, you might want to run two separate stacks for intranet and Internet traffic, or AnyNet Sockets over SNA in conjunction with one or more TCPIP stacks.

What is INET?

- In its simplest form it is a single TCPIP stack in an LPAR



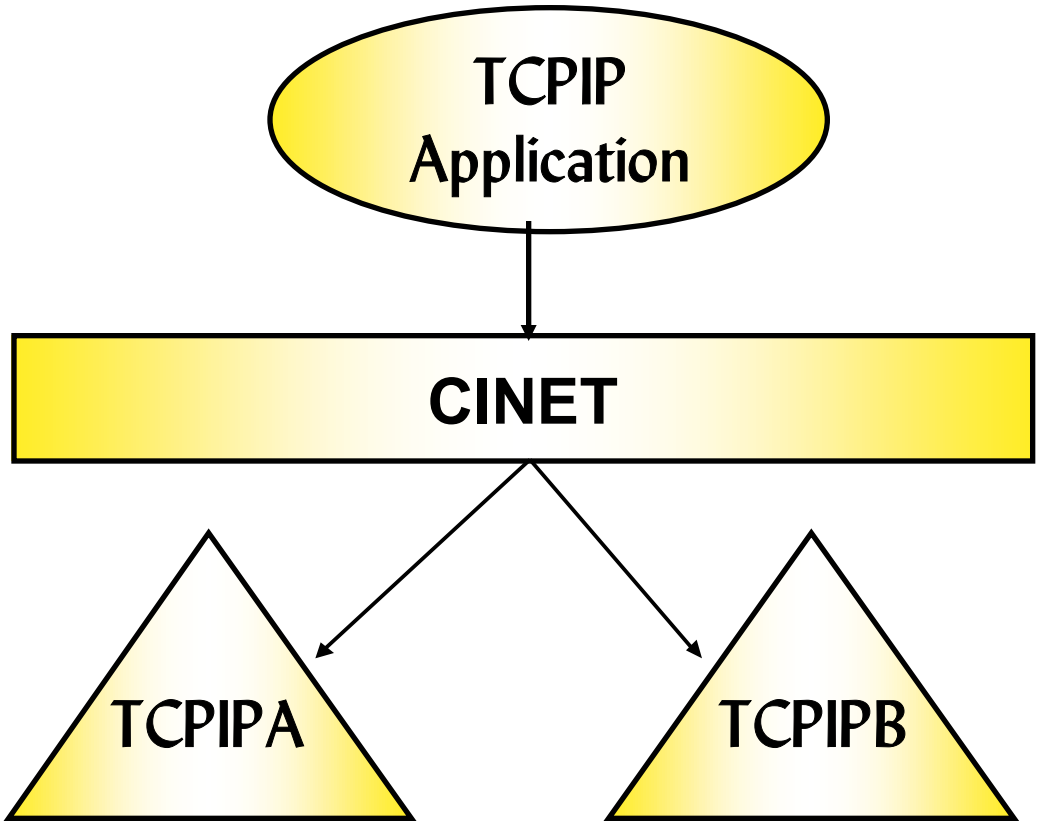
Customizing BPXPRMxx for INET

```
FILESYSTYPE TYPE(INET)
            ENTRYPOINT(EZBPFINI)
  NETWORK DOMAINNAME(AF_INET)
            DOMAINNUMBER(2)
            MAXSOCKETS(64000)
            TYPE(INET)
  NETWORK DOMAINNAME(AF_INET6)
            DOMAINNUMBER(19)
            MAXSOCKETS(64000)
            TYPE(INET)
```

- Benefits
  - no concern for stack affinity
  - simplified application design
  - single routing table

What is CINET?

- CINET provides multiple transport providers (TCPIP stacks)



What is unique about CINET?

- Unique to OS/390 and z/OS platform
- OMVS was built into existing MVS services
  - There were 2 existing Transports
    - VTAM--> ANYNETSockets over SNA
    - TCPIP
- Possibility to use more than one Transport service
- Ability to test one service level of a transport while running production of another service level
- Ability to host services for Internet and intranet traffic within a single LPAR

BPXPRMxx example of CINET

```
FILESYSTYPE TYPE(CINET)
            ENTRYPOINT(BPXTCINT)
SUBFILESYSTYPE NAME(TCPIPA)       /* First TCPIP stack   */
               TYPE(CINET)
               ENTRYPOINT(EZBPFINI)
                DEFAULT
SUBFILESYSTYPE NAME(TCPIPB)       /* Second TCPIP stack */
               TYPE(CINET)
               ENTRYPOINT(EZBPFINI)
NETWORK DOMAINNAME(AF_INET)
DOMAINNUMBER(2)
MAXSOCKETS(64000)
```

What is the CINET prerouter?

- Learns interface list and routing information from each stack
- Determines which stack gets a socket request
  - when stack affinity is not defined
  - bind() and connect() are the most prolific
- Makes decision based on interface list and routing information from each stack
  - Routing is just like TCPIP
    - ➔ Selects the most specific route first. (i.e. host then network route)
  - If routes are the same with the same metric, the default stack is used
  - Default routes are first checked at the default stack, if none available then check the other stacks
  - There is no load balancing performed by the pre-router

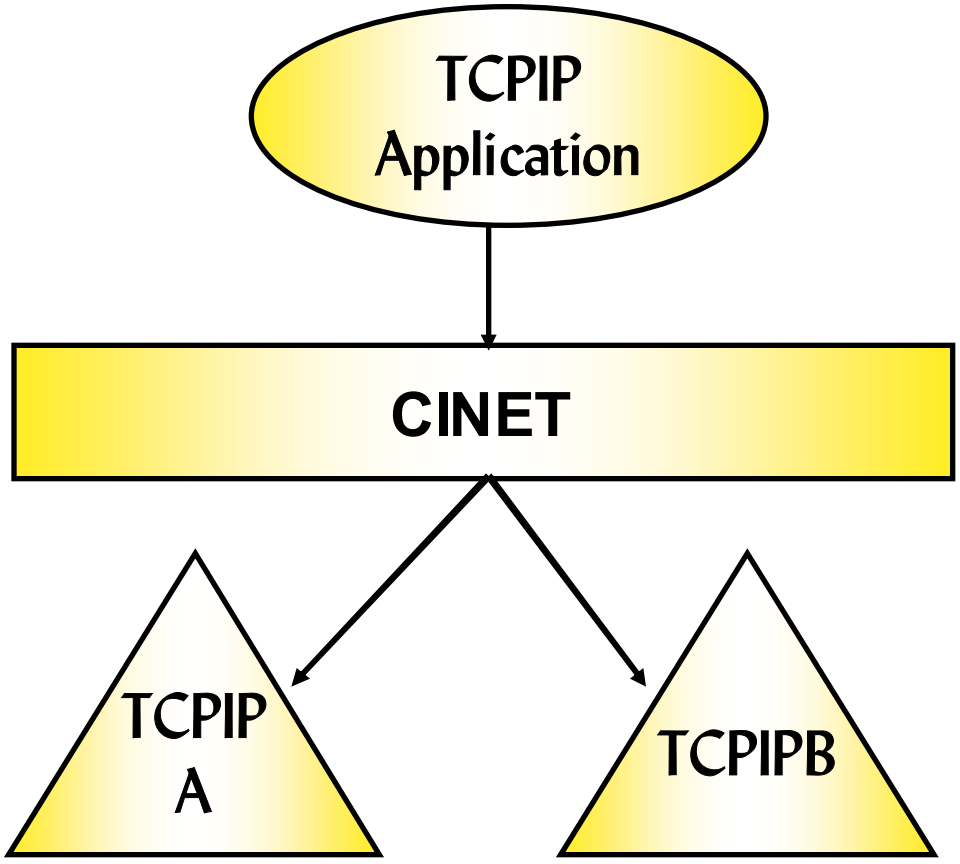
What happens to my socket call?

Every socket call goes through the CINET layer where it is then directed to the stack which stack affinity is set.

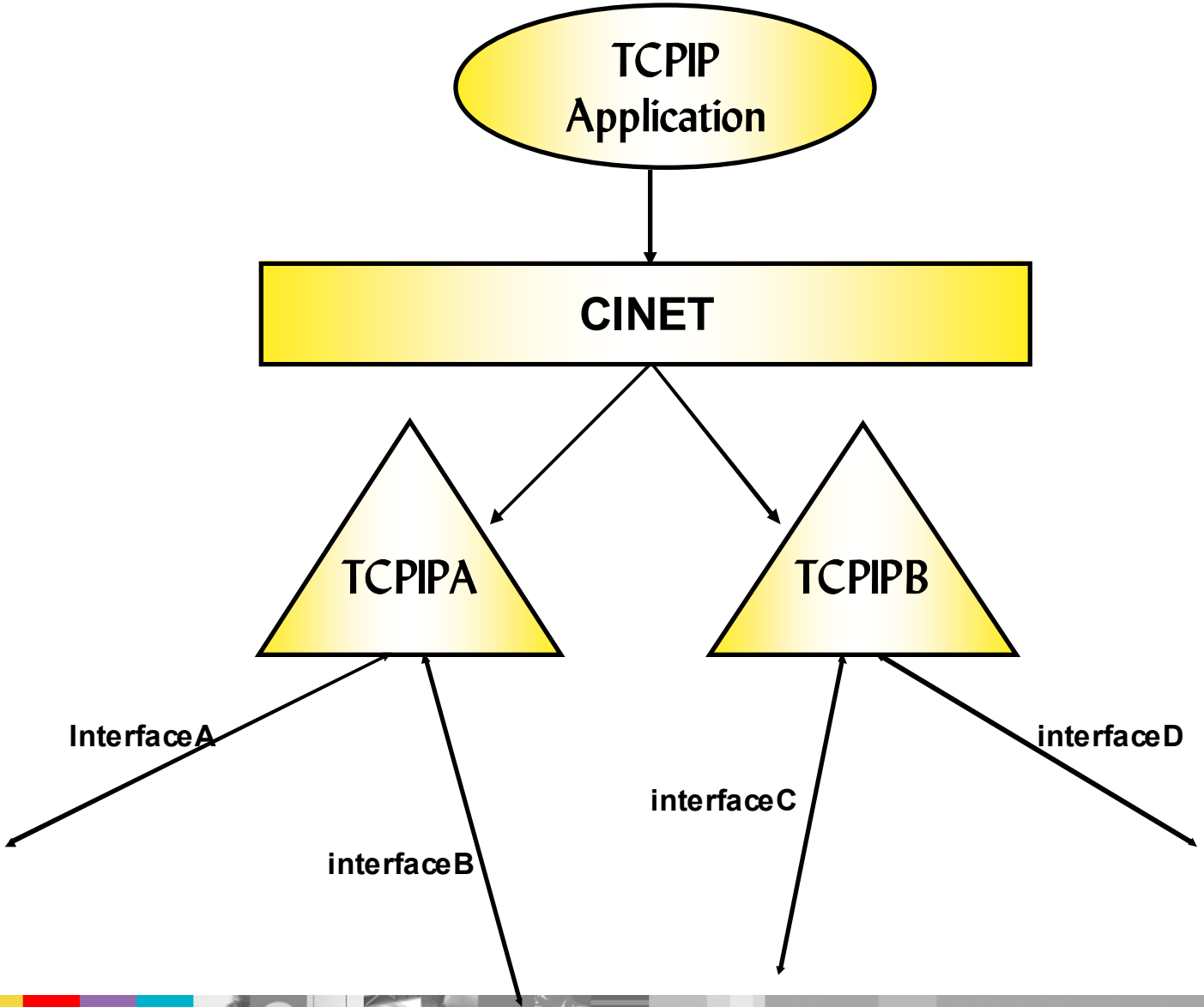
- How CINET selects a TCPIP stack (stack affinity)
  - ➔ Stack affinity set by the application
    - ✓**socket call:**
      setibmopt (IBMTCP_IMAGE) ioctl (SIOCSETRTTD)
    - ✓**environment variable**
      _BPXK_SETIBMOPT_TRANSPORT
    - ✓**BPXTCAFF**
  - ➔ No stack affinity set by the application
    - ✓ **based on interface address(es) per stack**
    - ✓ **based on the CINET pre-router routing table built from each stack**
    - ✓ **bind() goes to specific stack or all available stacks**
  - ➔ DEFAULT (if coded in BPXPRMxx)
    - ✓**All things being equal the DEFAULT stack is used**
    - ✓**If DEFAULT is not coded, the first stack started is DEFAULT**
    - ✓**D OMVS,P will show which stack is DEFAULT**

CINET

- Remember this?



CINET


TCPIP
Application
CINET
TCPIPA
TCPIPB
InterfaceA
interfaceB
interfaceC
interfaceD

TCPIP Configuration

TCPIPA Profile

```
HOME
1.1.1.1      interfaceA
2.2.2.2      interfaceB
BEGINROUTES
Route   1.1.0.0/16    =    interfaceA   mtu 1500
Route   2.2.0.0/16    =    interfaceB   mtu 1500
ENDROUTES
```

D TCPIP,TCPIPA,NETSTAT,ROUTE

EZZ2500I NETSTAT CS V1R7 TCPIPA 576

DESTINATION	GATEWAY	FLAGS	REFCNT	INTERFACE
1.1.0.0	0.0.0.0	UG	000000	INTERFACEA
2.2.0.0	0.0.0.0	UG	000000	INTERFACEB
127.0.0.1	0.0.0.0	UH	000003	LOOPBACK

TCPIPB Profile

```
HOME
3.3.3.3      interfaceC
4.4.4.4      interfaceD
BEGINROUTES
Route        3.3.0.0/16   =    interfaceC    mtu 1500
Route        4.4.0.0/16   =    interfaceD    mtu 1500
ENDROUTES
```

D TCPIP,TCPIPB,NETSTAT,ROUTE

EZZ2500I NETSTAT CS V1R7 TCPIPB 576

DESTINATION	GATEWAY	FLAGS	REFCNT	INTERFACE
3.3.0.0	0.0.0.0	UG	000000	INTERFACEC
4.4.0.0	0.0.0.0	UG	000000	INTERFACED
127.0.0.1	0.0.0.0	UH	000003	LOOPBACK

CINET internals

- TCPIPA starts and OMVS is notified that a new PFS is active
  - OMVS will issue ioctls to get the interface list from TCPIPA
  - OMVS will issue ioctls to get the routing table from TCPIPA
- TCPIPB starts...
  - OMVS will issue ioctls...interface list from TCPIPB
  - OMVS will issue ioctls... routing table from TCPIPB
- CINET prerouter now has an interface list and routing table from each of the TCPIP stacks
  - D OMVS,CINET=All will display routing information for CINET prerouter

CINET route table

D OMVS, CINET=All

IPV4 HOME INTERFACE INFORMATION

TP NAME	HOME ADDRESS	FLAGS
TCPIPA	001.001.001.001	
TCPIPA	002.002.002.002	
TCPIPB	003.003.003.003	
TCPIPB	004.004.004.004	

IPV4 HOST ROUTE INFORMATION

TP NAME	HOST DESTINATION	METRIC
TCPIPA	001.001.001.001	0
TCPIPA	002.002.002.002	0
TCPIPB	003.003.003.003	0
TCPIPB	004.004.004.004	0
TCPIPA	127.000.000.001	0
TCPIPB	127.000.000.001	0

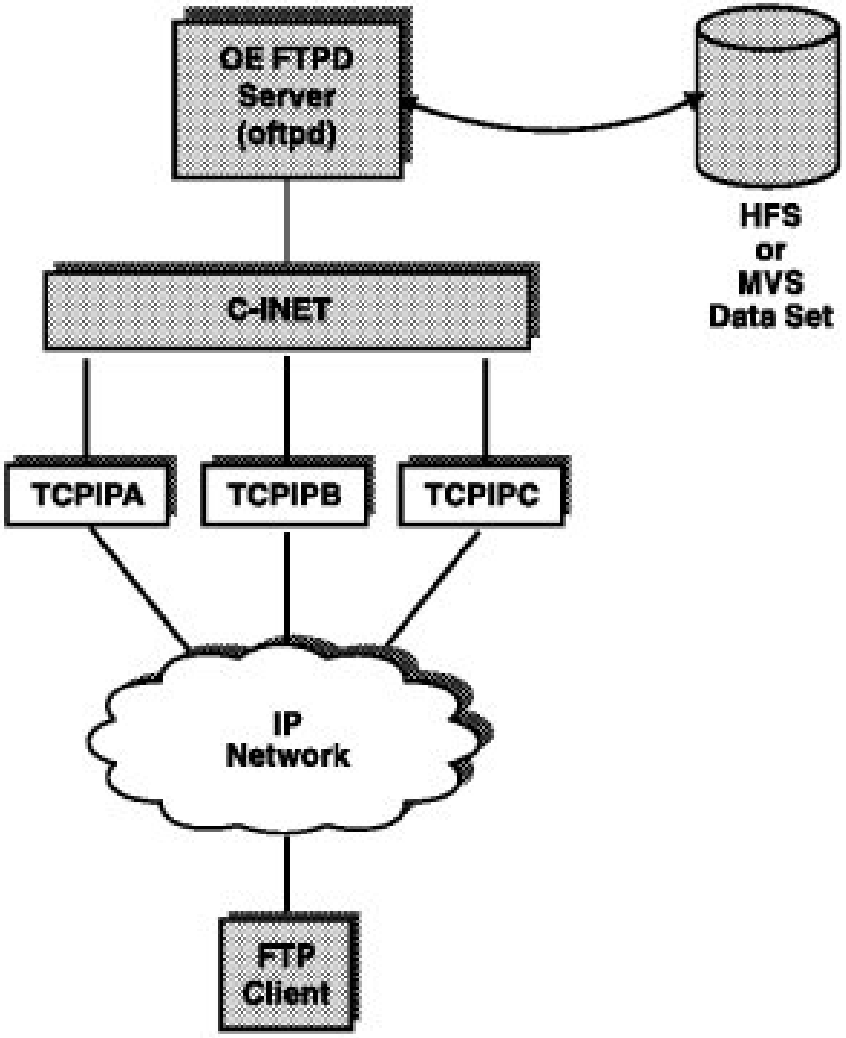
IPV4 NETWORK ROUTE INFORMATION

TP NAME	NET DESTINATION	NET MASK	METRIC
TCPIPA	001.001.000.000	255.255.000.000	0
TCPIPA	002.002.000.000	255.255.000.000	0
TCPIPB	003.003.000.000	255.255.000.000	0
TCPIPB	004.004.000.000	255.255.000.000	0

Bind() and Connect()

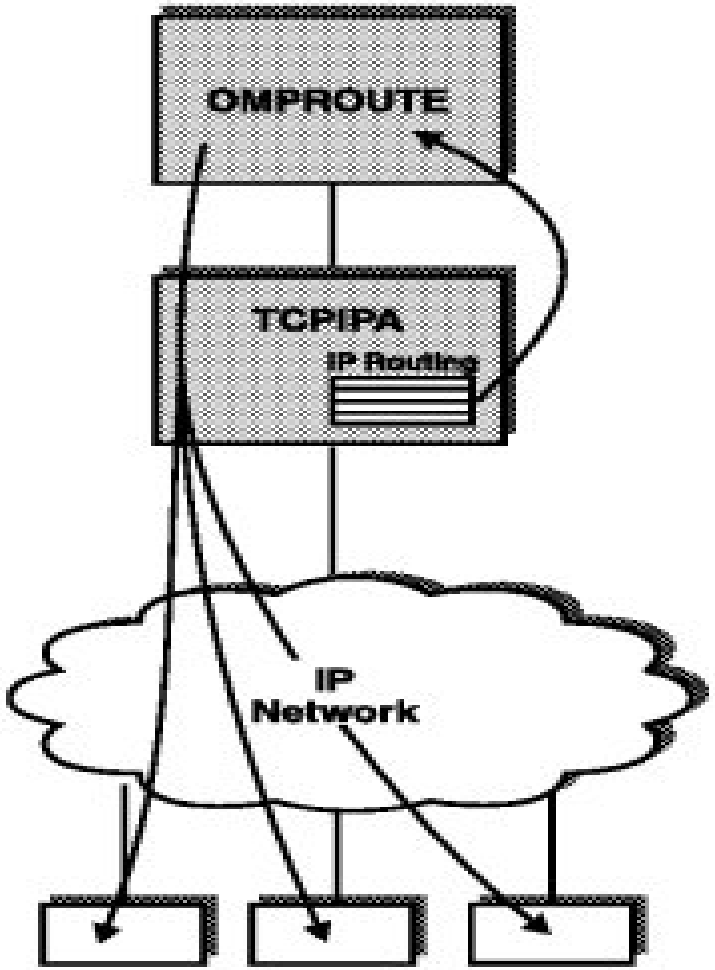
- TCPIP Application issues a bind() to inaddrany (0.0.0.0)
  - CINET interprets the bind() and sends to both TCPIPA and TCPIPB
- TCPIP Application issues a bind() to address 1.1.1.1
  - CINET interprets the bind() and sends to TCPIPA
- TCPIP application issues a connect() to address 3.3.32.19
  - CINET interprets the connect() and sends it to TCPIPB

Generic Server



- The FTP Server shipped with z/OS Comm Server is an example of a generic server
- A generic server is a server that is not dependent upon a particular transport provider (stack)
- A generic server can communicate concurrently over any number of stacks
  - If desirable, a generic server can be configured to set affinity to a particular stack

Servers Requiring Affinity



- Omproute shipped with z/OS Comm Server is an example of a server requiring stack affinity
- This type of server is dependant upon the internal functioning of a particular stack
- This type must have affinity to a specific stack

Common Problem1

- What would happen if?

Given:

TCPIPA PROFILE

.
.
PORT TCP 21 FTPD1
.

TCPIPB PROFILE

.
.
PORT TCP 21 FTPSERV
.

- FTPD starts and FTPD1 will bind to port 21 and inaddrany
  - In this case, CINET will send the bind to both stacks
- When the bind gets to TCPIPA:
  - It succeeds because TCPIPA has port 21 reserved for FTPD1
- When the bind gets to TCPIPB:
  - It will fail with errno x'6F' (dec 111) EACCESS
    - ➔Cause: FTPD1 tried to bind to port 21 and TCPIPB has it reserved for FTPSERV

Common problem2

TCPIPA

VIPADYNAMIC
VIPARANGE DEFINE 255.255.255.0 8.8.8.0
ENDVIPADYNAMIC

TCPIPB

(*No VIPARANGE)

- APPLX issues a bind to 8.8.8.6
  - CINET will check the interface list, no match is found and the bind() is sent to TCPIPA (default transport provider)
  - Bind() succeeds because 8.8.8.6 falls within the VIPARANGE and the DVIPA is created
- If TCPIPB happened to be the default transport provider:
  - Bind() will fail with errno EADDRNOTAVAIL and errno2 JRInvalidAddr.
  - Cause: TCPIPB does not have a VIPARANGE defined for 8.8.8.6

Common problem3

TCPIPA

VIPADYNAMIC
VIPARANGE DEFINE 255.255.255.0 8.8.8.0
ENDVIPADYNAMIC

PORT

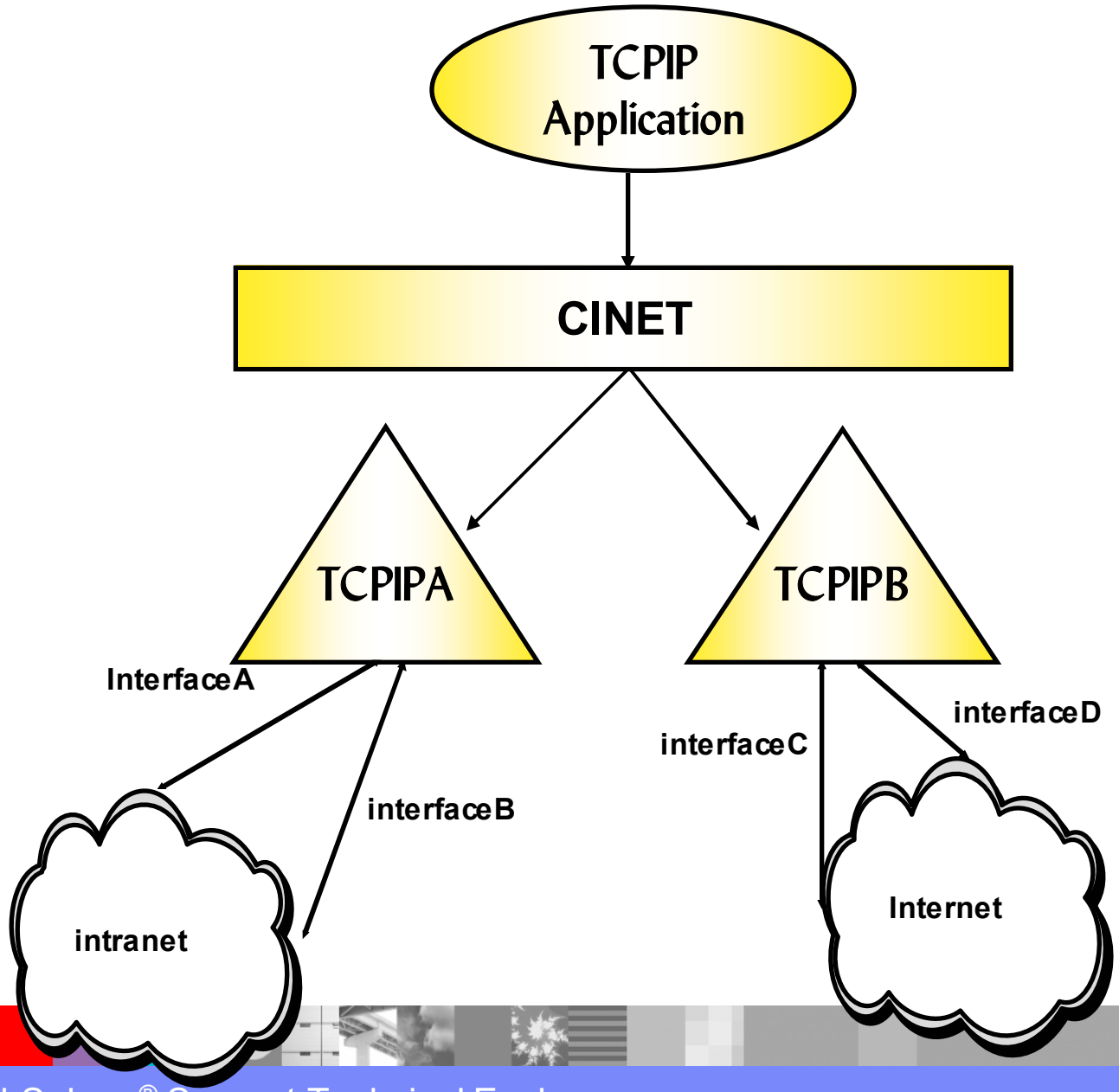
2413 TCP APPLX BIND 8.8.8.8

TCPIPB

(*No VIPARANGE)

- APPLX starts and will bind to port 2413 and Inaddrany so that it will be converted to a specific bind to create a DVIPA
  - CINET will send the bind to both stacks
- When the bind gets to TCPIPA:
  - Bind is established on port 2413 to 8.8.8.8
- When the bind gets to TCPIPB:
  - Bind is established on port 2413 to 0.0.0.0

CINET with different network types



Default routes

Another layer of complexity:

TCPIPA and TCPIPB PROFILE data

TCPIPA

```
HOME
1.1.1.1     interfaceA
2.2.2.2     interfaceB
BEGINROUTES
Route       1.1.0.0/16 =    interfaceA        mtu 1500
Route       2.2.0.0/16 =    interfaceB        mtu 1500
Route       Default   1.1.1.254  interfaceA     mtu 1500
Route       Default   2.2.2.254  interfaceB     mtu 1500
ENDROUTES
```

TCPIPB

```
HOME
3.3.3.3     interfaceC
4.4.4.4     interfaceD
BEGINROUTES
Route       3.3.0.0/16 =    interfaceC        mtu 1500
Route       4.4.0.0/16 =    interfaceD        mtu 1500
Route       Default   3.3.3.254  interfaceC     mtu 1500
Route       Default   4.4.4.254  interfaceD     mtu 1500
ENDROUTES
```

CINET route table

CINET's view of TCPIP routing:

D OMVS,CINET=All

IPV4 HOME INTERFACE INFORMATION

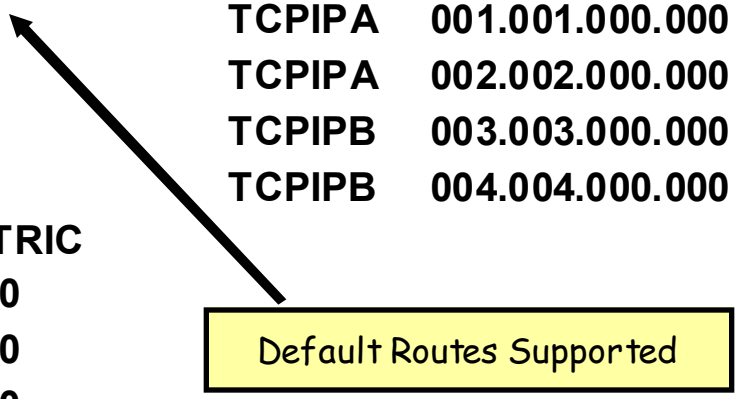
TP NAME	HOME ADDRESS	FLAGS
TCPIPA	001.001.001.001	DRS
TCPIPA	002.002.002.002	DRS
TCPIPB	003.003.003.003	DRS
TCPIPB	004.004.004.004	DRS

IPV4 HOST ROUTE INFORMATION

TP NAME	HOST DESTINATION	METRIC
TCPIPA	001.001.001.001	0
TCPIPA	002.002.002.002	0
TCPIPB	003.003.003.003	0
TCPIPB	004.004.004.004	0
TCPIPA	127.000.000.001	0
TCPIPB	127.000.000.001	0

IPV4 NETWORK ROUTE INFORMATION

TP NAME	NET DESTINATION	NET MASK	METRIC
TCPIPA	001.001.000.000	255.255.000.000	0
TCPIPA	002.002.000.000	255.255.000.000	0
TCPIPB	003.003.000.000	255.255.000.000	0
TCPIPB	004.004.000.000	255.255.000.000	0



Common problem4

- Given:

TCPIP APPLA **is an application that connects to users on the company intranet**

TCPIP APPLB **is an application that connects to users on the Internet**

- APPLA issues a connect to 1.1.35.143
  - CINET will check the routing table and send the connect() to TCPIPA
- APPLB connects to 207.25.253.21, which is on the Internet
  - Which stack will CINET prerouter send the connect to?
    - ➔It will be routed to TCPIPA because it is defined as DEFAULT in the BPXPRMxx member
    - ➔The connect will time out because TCPIPA is NOT connected to the Internet

connect()

Controlling which stack gets a connect() call

- Establish Stack affinity

➔socket call:

✓**setibmopt(IBMTCP_IMAGE)**

✓**ioctl(SIOSETRTTD)**

➔environment variable _BPXK_SETIBMOPT_TRANSPORT

✓**_BPXK_SETIBMOPT_TRANSPORT=TCPIPA**

For applications using the Language Environment (LE)

```
//FTPD   EXEC PGM=&MODULE,REGION=4096K,TIME=NOLIMIT,
//       PARM=('POSIX(ON) ALL31(ON)',
//            'ENVAR("_BPXK_SETIBMOPT_TRANSPORT=TCPIPA"',
//            '"TZ=EST")/&PARMS')
```

✓**Can be set prior to starting a program**

✓**Set in the PARM= statement to have LE issue setibmopt()**

✓**Set in the _CEE_ENVFILE**

connect()

- ➔BPXTCAFF
  - ✓Set for the duration of an address space
  - ✓Applies to all UNIX processes running in the address space
  - ✓Intended for use with non-C or POSIX(OFF)

```
//STEP0 EXEC,PGM=BPXTCAFF,PARM=TCPIPB
//REALSTEP EXEC,PGM=MYPGM,PARM='MyParm
```

- No stack affinity set by the application
  - ➔based on the CINET pre-router routing table built from each stack
- **DEFAULT** if coded in BPXPRMxx
  - ➔ All things being equal the DEFAULT stack is used
  - ➔ Or, if no matches are found, errno x'45C' (dec 1118) ENETUNREACH is returned to a connect() call
  - ➔ If DEFAULT is not coded the first stack started is DEFAULT

bind()

Controlling which stack gets a bind() call

- Bind specific
  - Binds to a specific IP address
  - Usually not a problem if the correct interface addresses are defined
  - Binds to specific DVIPA addresses requires:
    - ✓ DVIPA should already be active via VIPADEFINE or VIPADISTRIBUTE DEFINE statement
    - ✓ DVIPA not yet active, IP address should be defined via VIPARANGE statement
      - ❖ Stack affinity should be set to ensure it goes to the correct stack
- Bind inaddrany
  - Establish Stack affinity
    - ➔ socket call:
      - ✓ **setibmopt(IBMTCP_IMAGE)**
      - ✓ **ioctl(SIOSETRTTD)**
        - **Sets affinity of the associated socket to a particular stack**
        - **Removes any previous affinity**

bind()

- environment variable _BPXK_SETIBMOPT_TRANSPORT
  - **_BPXK_SETIBMOPT_TRANSPORT=TCPIPA**

    For applications using the Language Environment (LE)

```
//FTPD   EXEC PGM=&MODULE,REGION=4096K,TIME=NOLIMIT,
//       PARM=('POSIX(ON) ALL31(ON)',
//            'ENVAR("_BPXK_SETIBMOPT_TRANSPORT=TCPIPA"',
//             '"TZ=EST")/&PARMS')
```

  - **Can be set prior to starting a program**
  - **Set in the PARM= statement to have LE issue setibmopt()**
  - **Set in the _CEE_ENV file**

- BPXTCAFF - if started via batch
  - **//STEP0 EXEC,PGM=BPXTCAFF,PARM=TCPIPB**
  - **//REALSTEP EXEC,PGM=MYPGM,PARM='MyParm**

- No stack affinity set by the application
  - Based on the CINET pre-router interface address list
  - Directed to multiple stacks or only one stack

bind()

- **DEFAULT** if coded in BPXPRMxx
  - If no matches are found for a specific IP address, the bind is sent to the DEFAULT stack
  - If a VIPARANGE is not defined for the specific IP address, the bind will fail with errno EADDRNOTAVAIL and errno2 JRInvalidAddr
  - If DEFAULT is not coded, the first stack started is DEFAULT

Additional awareness

- Connect()
  - Coding NETACCESS in the TCPIP PROFILE can have an affect

```
NETACCESS
192.168.113.0    255.255.255.0   SUBNET1
192.168.112.0    255.255.248.0   SUBNET2
ENDNETACCESS
```

Applications doing the connect() permitted to:
EZB.NETACCESS.sysname.tcpname.SUBNET1

- Bind()
  - Coding PORT statement in the TCPIP PROFILE can have an affect

```
PORT
3000 TCP CICSTCP  SAF CICS
```

CICSTCP must be permitted to:
EZB.PORTACCESS.sysname.tcpname.CICS

Caution!

It is not recommended to run z/OS Communications Server TCPIP stacks in a CINET environment.

Technotes and References

- **Technotes:**

http://www-306.ibm.com/software/network/commserver/zos/support/

  - 1079109 -- Creating DVIPAs with the BIND Keyword on Port reservations in a CINET environment
  - 1079111 -- Creating DVIPAs using bind in a CINET environment

- **Reference manuals:**
  - z/OS V1R4.0 UNIX System Services File System Interface Reference
  - z/OS V1R4.0 UNIX System Services Planning
  - z/OS V1R4.0 Communications Server IP Configuration Guide
  - z/OS V1R4.0 C/C++ Programming Guide
  - z/OS V1R4.0 MVS System Commands

Additional WebSphere Product Resources

- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at: www.ibm.com/developerworks/websphere/community/
- Learn about other upcoming webcasts, conferences and events: www.ibm.com/software/websphere/events_1.html
- Join the Global WebSphere User Group Community: www.websphere.org
- Access key product show-me demos and tutorials by visiting IBM Education Assistant: www.ibm.com/software/info/education/assistant
- Learn about the Electronic Service Request (ESR) tool for submitting problems electronically: www.ibm.com/software/support/viewlet/ESR_Overview_viewlet_swf.html
- Sign up to receive weekly technical My support emails: www.ibm.com/software/support/einfo.html
- Attend WebSphere Technical Exchange conferences or Transaction and Messaging conference: http://www-304.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageType=page&c=a0011317

Questions and Answers