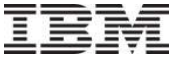
The z Exchange – April 25, 2019

TCP/IP Networking Essentials for the zOS System Programmer

Chris Meyer, CISSP (meyerchr@us.ibm.com)
z/OS Communications Server design and architecture



Before we start

This presentation is NOT a comprehensive overview of TCP/IP!

Rather, it attempts to convey basic networking concepts in simple, easy-to-understand terms. As such, we will generalize and skim over various details at many points.

In some cases, we will select representative use cases to illustrate key points, even though those use cases that are most germane to z/OS networking, but may not represent all possible scenarios.

For more information on specific details or implementation scenarios, please refer to appropriate documentation or tutorials.

Agenda

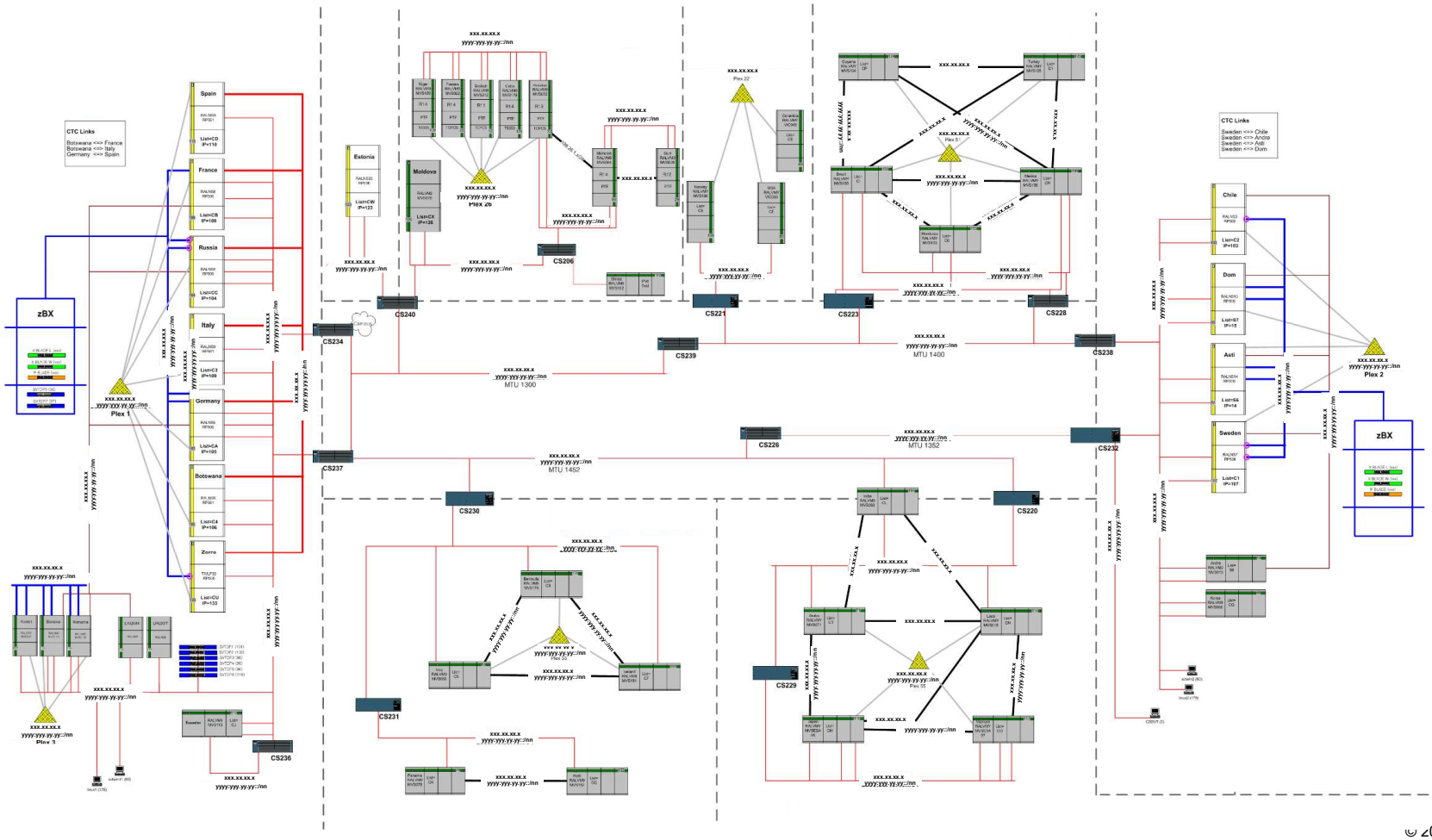
- Core concepts
- Common devices and terms
- Some z/OS-unique features
- Network security – the typical overlap with your job

Agenda

- **Core concepts**
- Common devices and terms
- Some z/OS-unique features
- Network security – the typical overlap with your job

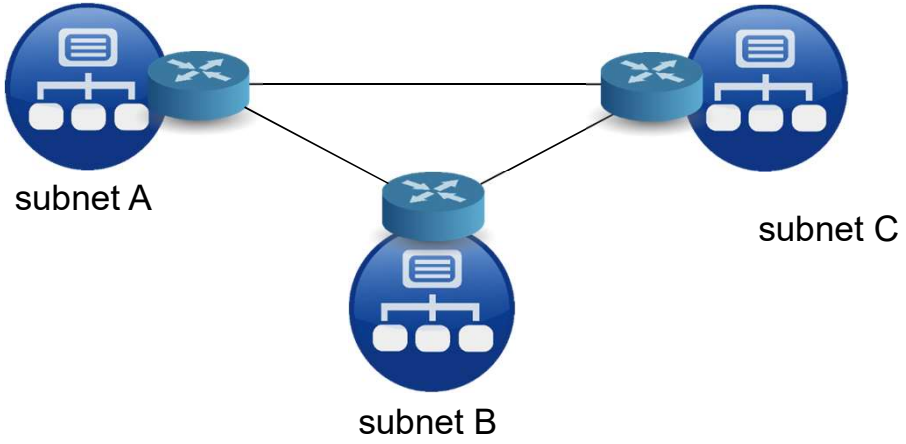
Concepts: What you see…

Concepts: What the network really looks like…



Concepts: A network of networks (subnets)

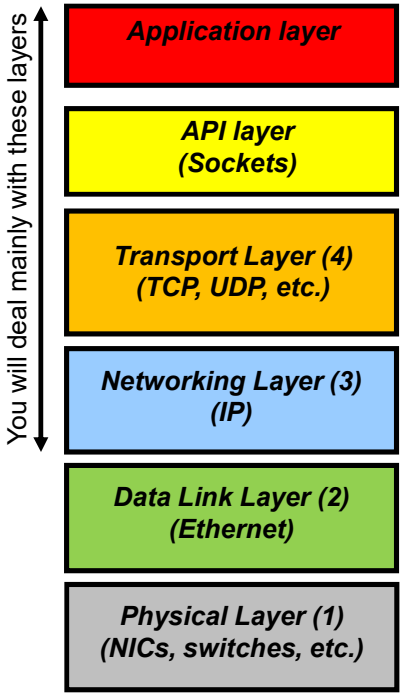
- "IP" stands for Internet Protocol
- From the ground up, this protocol is designed and built to connect multiple separate networks into larger, interconnected networks
- Within your own enterprise, and most likely, even in your home, you have multiple interconnected IP networks. These smaller networks are called *subnets*



- TCP/IP-related protocols are defined by the Internet Engineering Task Force (IETF) and each is published in a document called a *Request For Comment (RFC)*. Each RFC is issued a unique number (ex: RFC 8446)

Concepts: TCP/IP layered model

- Based on the Open Systems Interconnect (OSI) 7-layer reference model (we won't go into that here)
- Each layer provides a distinct set of services:



Applications and middleware like TN3270E server or CICS that use network services

The programming interfaces that the application layer uses to access network services

Provides the semantics of the communication model (connection oriented, datagram model, etc.) as well as distinct application access points (ports)

Knows how to route IP packets between nodes in the IP network using IP addresses

Knows how to move link-specific frames across local links to adjacent devices

Physical (Hardware)

- An implementation of this model is called a *TCP/IP stack*

Concepts: IP, TCP and UDP

TCP – Transmission Control Protocol

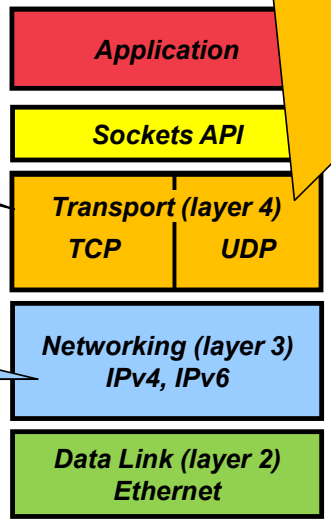
- Sits on top of IP – deals in **TCP segments**
- **Connection-oriented streaming protocol**
- **Guaranteed delivery of data in the proper order**
- **Analogous to a telephone connection**
  - One party calls another, establishing a connection
  - Conversation takes place over that connection
  - When the conversation is over, the parties disconnect
- **TCP-based applications** on a node are identified by a **TCP port** (e.g., the FTP server listens on TCP port 21)
- **Most z/OS application and middleware traffic flows over TCP**

UDP – User Datagram Protocol

- Sits on top of IP – deals in **UDP datagrams**
- **Unreliable, but very lightweight**
- **Analogous to mailing a letter**
  - One party sends a message to another by putting the target address on the envelope and dropping it in a mailbox
  - In most cases, the letter eventually arrives at the recepient, but the sender doesn't know unless the receiver sends another letter back. But even then, the reply message may never get to the first party
  - Up to the parties to re-send letters if they don't think the first one got there
- **UDP-based applications** on a node are identified by a **UDP port** (e.g., the Internet Key Exchange daemon listens on UDP port 500)

IP – Internet Protocol

- Passes **IP packets between nodes** in an IP network
- Nodes are identified by **IP addresses**
  - IPv4: 32-bit addresses in "dotted decimal" notation (e.g., 192.168.1.7)
  - IPv6: 128-bit addresses in "colon hexadecimal" notation (e.g., 2001::0dea:C1AB:0000:00D0:ABCD:0041);
- **An IP address gets you to a specific node** (each network inerface considered a different node)



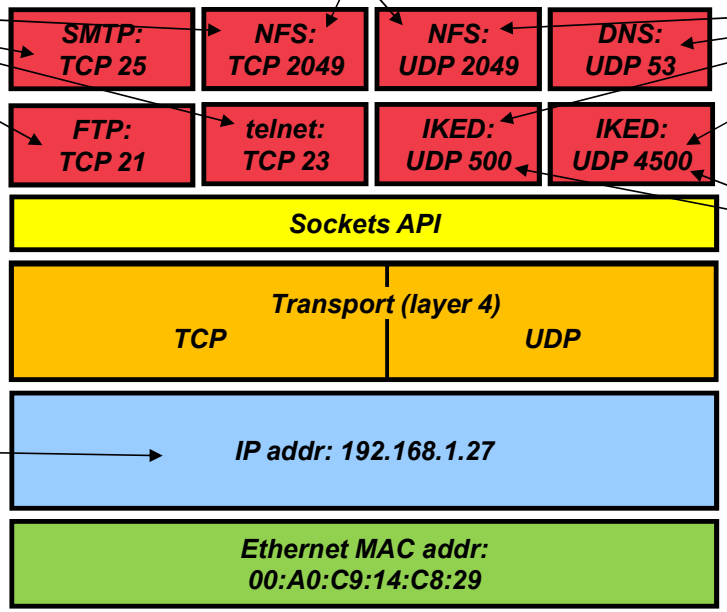
Concepts: Examples of an IP host

IP address + port gets you to a specific application on the host

Some applications use both TCP and UDP

TCP applications

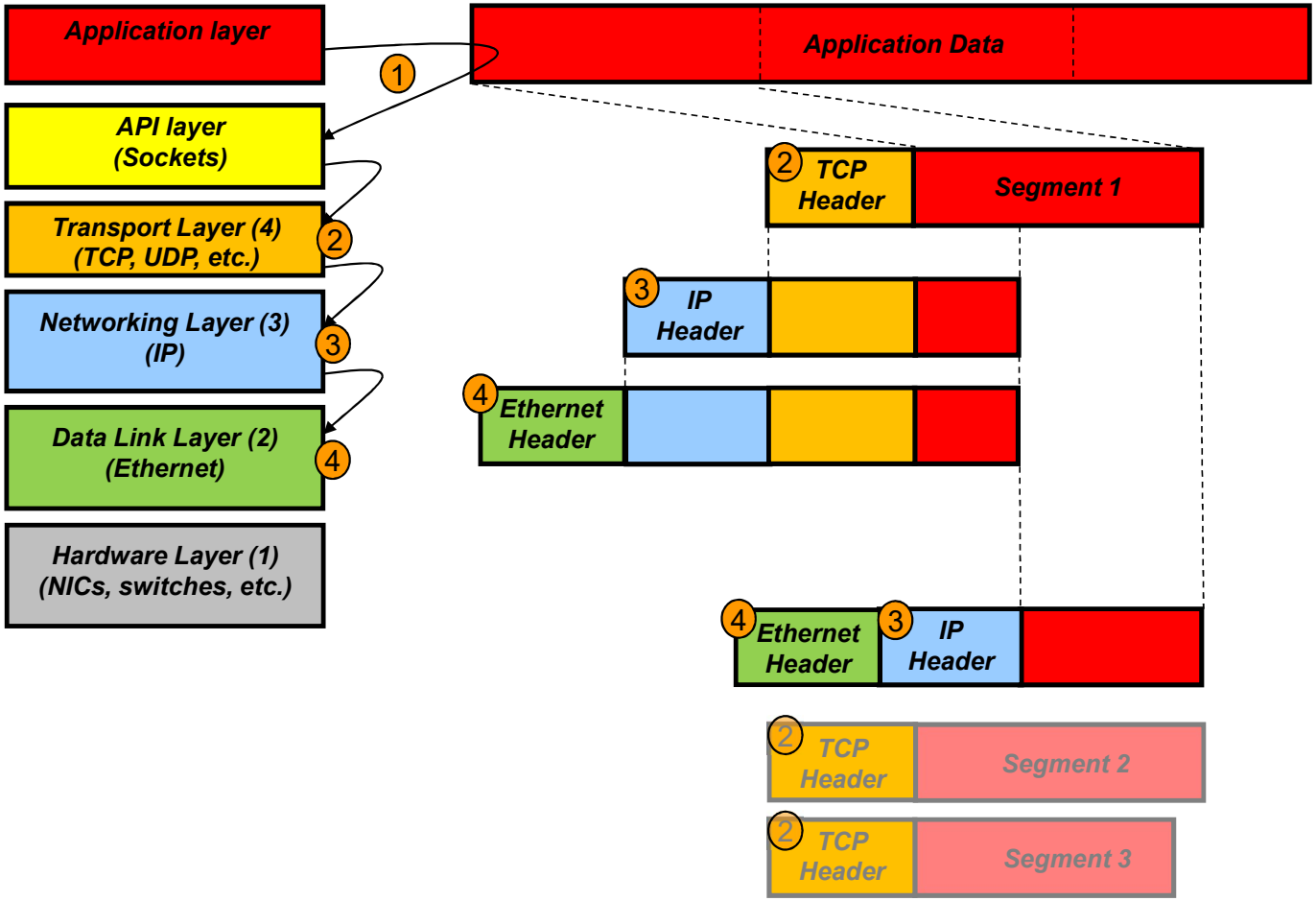
UDP applications



Some applications use multiple ports of the same protocol

IP address gets you to the host

Concepts: “The stack” and network data encapsulation



Concepts: IPv4 addresses

- IPv4 is protocol that was established in the 1970s and eventually took the world by storm
- Most of the western hemisphere still operates largely on IPv4 and parts of the eastern do as well

- IPv4 uses 32-bit addresses that are described in four “octets” that are separated by dots. This is called *dotted decimal notation*. For example:

  192.168.1.1

- Each IPv4 address identifies a specific network (called a *subnet*) and a specific host within that subnet
- Most modern IP networks use variable length *subnet masking* which allows network administrators to decide how many of those 32 bits (starting from the left) identify the network and how many identify a host within that network. The important thing to know is the notation that describes this split. This is called *Classless Inter-Domain Routing (CIDR)* notation, which takes the form:

  aa.aa.aa.aa/pp where ‘aa’ represents an octet of the address and ‘pp’ specifies the number of prefix bits

- The number of IPv4 addresses in a given subnet is roughly $2^{(32-\text{number of prefix bits})}$-2 (-2 because values of all ‘0’B or all ‘1’B cannot serve as host addresses – these are the subnet and broadcast addresses, respectively). For example,

  the prefix /24 gives 253 host addresses (2^{32-24}-2 = 2^8-2 = 255-2 = 253), so subnet 192.168.1.0/24 supports hosts 192.168.1.1 through 192.168.1.254

- You may also encounter something called *network masks* – we won’t describe them here – if need be, look them up

Concepts: IPv6 addresses

- IPv6 addressed the limited number of possible IPv4 addresses (we "ran out" of IPv4 addresses sometime in 2015). IPv6 supports enough addresses to "…assign an IPV6 address to EVERY ATOM ON THE SURFACE OF THE EARTH, and still have enough addresses left to do another 100+ earths." (Steve Leibson, EDN Network, 2008)
- IPv6 became a draft standard in 1998 (and an official standard only in 2017!)
- IPv6 usage is growing, especially in the eastern hemisphere. This growth has accelerated globally with the advent of mobile devices. However, IPv6 adoption on the mainframe is currently rather low.

- IPv6 uses 128-bit addresses that are described in 8 groups of 4 hexadecimal digits that are separated by colons (*colon hexadecimal notation*). A basic IPv6 address looks like this:

  fe80::a05d:bd43:acfd:d2e0 (which is really fe80:0000:0000:0000:a05d:bd43:acfd:d2e0)

  Note that two colons ("::") can be used to represent a sequence of one or more quartets of X'00000000'

- Like IPv4, each IPv6 address identifies a specific subnet and a specific host within that subnet
- With IPv6, variable length subnet masking is the only way to specify how those 128 bits are split between network and host identifiers. Again CIDR notation is used, taking the form:

  aaaa:aaaa:aaaa:aaaa:aaaa:aaaa:aaaa:aaaa /ppp where 'aaaa' represents 16 bits of the address and 'ppp' specifies the number of prefix bits (usually 64 to accommodate something called stateless autoconfig)

The number of addresses in a given subnet is determined in a manner similar to IPv4 addresses, but uses the formula $2^{(128-\text{number of prefix bits})}-1$ since IPv6 addresses are 128 bits long and since there are no broadcast addresses in IPv6

Agenda

- Core concepts
- **Common devices and terms**
- Some z/OS-unique features
- Network security – the typical overlap with your job

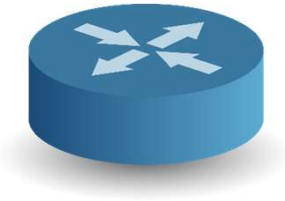
Common devices and terms (1 of 3)

- Network Interface Cards (NICs) and MAC addresses (Physical and Layer 2)
  - The network adapters in your machines - think of laptop wireless adapter or your z14's OSA
  - Every Ethernet and WiFi NIC has a burned-in address called a *Media Access Control (MAC) address*. The address format is long enough that every NIC produced by every manufacturer on the planet has a unique MAC address – plus plenty of space for *virtual MACs*
  - MAC addresses are used to address devices at Layer 2 (the Data Link Layer)
  - Ethernet Layer 2 traffic is confined to a single Local Area Network (LAN) segment (i.e., to the other devices that are connected to the same Ethernet switch).
  - Layer 2 traffic is not routable. For that, Layer 3 (IP) is required.

- Hosts (Layer 3 (IP))
  - These are typically endpoint devices that are attached to the network using one or more NICs
  - A host with more than one NIC is called a *multi-homed* host
  - A host supports an operating system which includes a TCP/IP stack. Typically, hosts are tasked with running application programs for one or more specific purposes
  - At the very least, each NIC on a host is assigned its own IP address (layer 3)

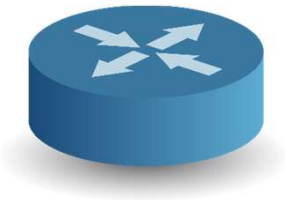
Common devices and terms (2 of 3)

- Routers (Layer 3 (IP))
  - Network devices (or specially configured hosts) that route IP packets (Layer 3) through the network using routing protocols like OSPF, RIP and BGP
  - Have at least two NICs (commonly more), each of which represents a connection to a different segment of the network
  - Can also have other functions built in (NAT, firewall, etc.)



- Proxies (General concept)
  - Devices or programs that act as a man-in-the-middle on behalf of the users on one “side” of the proxy
  - Typically used to control access to remote resources. For example, a corporate web proxy that prevents a company’s users from establishing secure connections to untrusted web sites

- Gateways (General concept)
  - Devices (typically routers) that provide access between two separate networks. Think of them as a entry points to other networks (for example, the default gateway you configure in your home’s wireless LAN to connect to your Internet Service Provider (ISP))

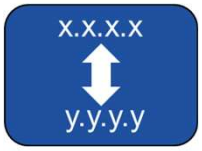


Common devices and terms (3 of 3)

- Firewalls
  - Devices or programs that control access to local or remote IP addresses, TCP and UDP ports or combinations of addresses and ports

- Intrusion Detection Services (IDS) / Intrusion Prevention Services (IPS)
  - Devices or programs that scan network traffic for a wide variety of intrusions and either report them or take protective action to prevent them
  - Most network-based IDS/IPS devices are *signature based* – the signatures tell the device what to look for (often in the application payloads of the network messages). New signatures can be created and loaded into the devices as new types of intrusions are discovered
  - In most cases, these are invisible to network users, but can come into play where cryptographic network protocols are used

- Network Address Translation (NAT)
  - Devices or programs that translate private IP addresses (inside your enterprise) into public addresses or addresses/ports for consumption on the open Internet.
  - Just for fun, Google “What is my IP address” from your workstation. You probably won’t recognize the resulting address -- most likely, it came from a NAT!

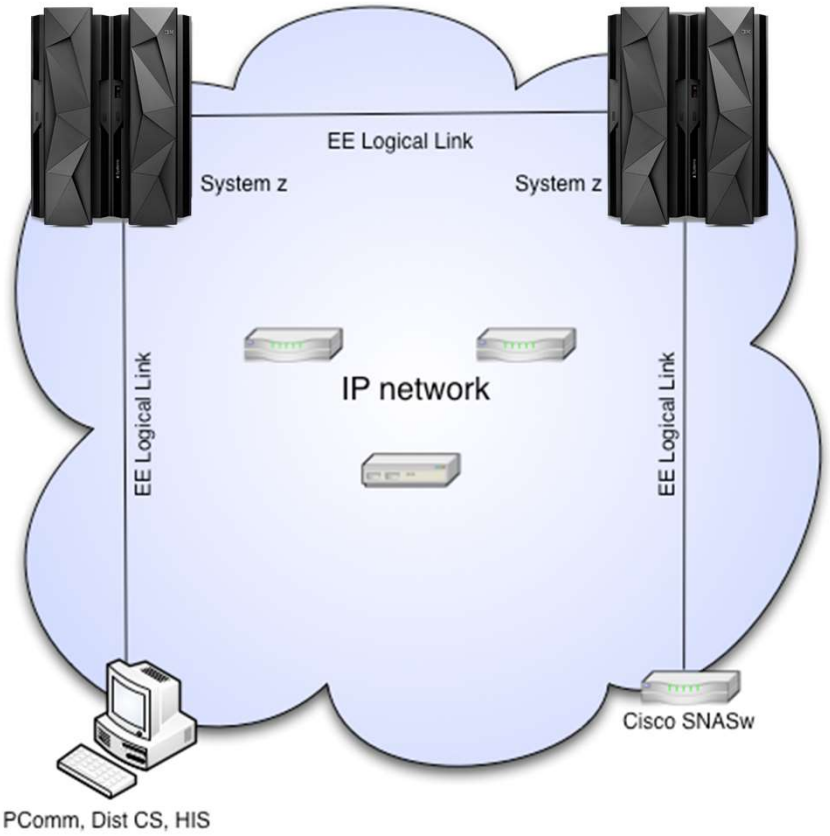


Agenda

- Core concepts
- Common devices and terms
- **Some z/OS-unique features**
- Network security – the typical overlap with your job

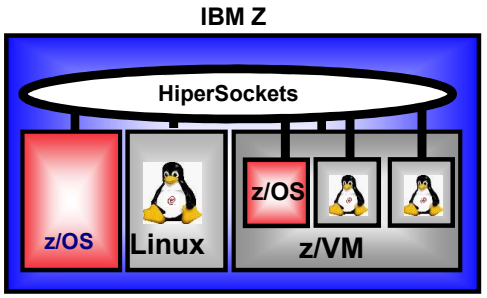
z/OS unique features: Enterprise Extender (EE)

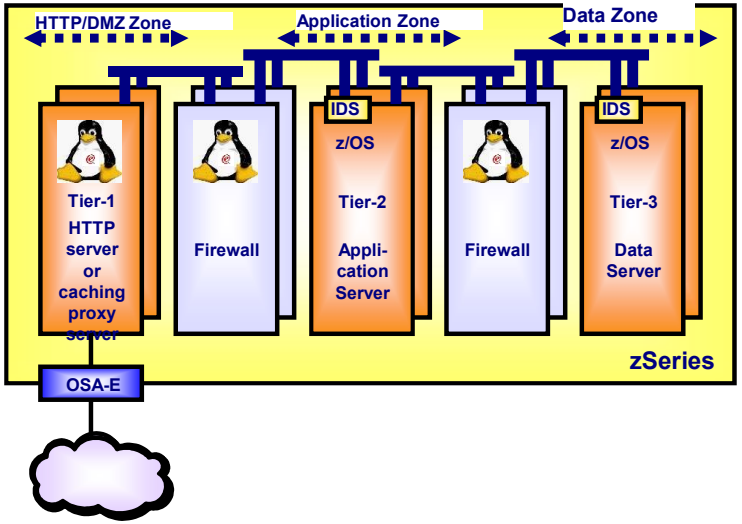
- Enterprise Extender provides a way to flow System Network Architecture (SNA) traffic over IP networks
- This has become critical since
  - Native SNA hardware is a thing of the past
  - But there are still tons of mission-critical SNA-based applications
- SNA has all of its own sophisticated flow control (ensuring data arrives at its destination in the correct order, etc.)
  - No need for such services from TCP
  - Only needs is a way to get SNA frames from one SNA node to another
  - UDP is a perfect fit for this
- EE uses a series of 5 UDP ports (12000-12004 by default) to allow the IP network to act as an SNA link to another EE node



z/OS unique features: Hipersockets

- Hipersockets provides memory-based IP networking among virtual servers within an IBM Z CEC
  - Up to 16 internal IP networks that interconnect z/OS, Linux on Z, VSE/ESA, and z/VM including z/VM guest operating systems
  - Very low latency
  - Highly secure (no external cables to tap into)
  - Highly available
  - No external networking equipment needed
  - Flexible and enables a variety of intra-CEC scenarios
  - Simple to install, operate, maintain
- A Hipersockets network
  - Looks like an internal LAN
  - Allows operating system images on the same processor complex to exchange IP traffic virtually at memory speed
- V2R3 provides Hipersockets Converged Interface
  - A single IP address is backed by both OSA AND Hipersockets
  - For outbound traffic, the TCP/IP stack decides at transmission time which is the best path to follow





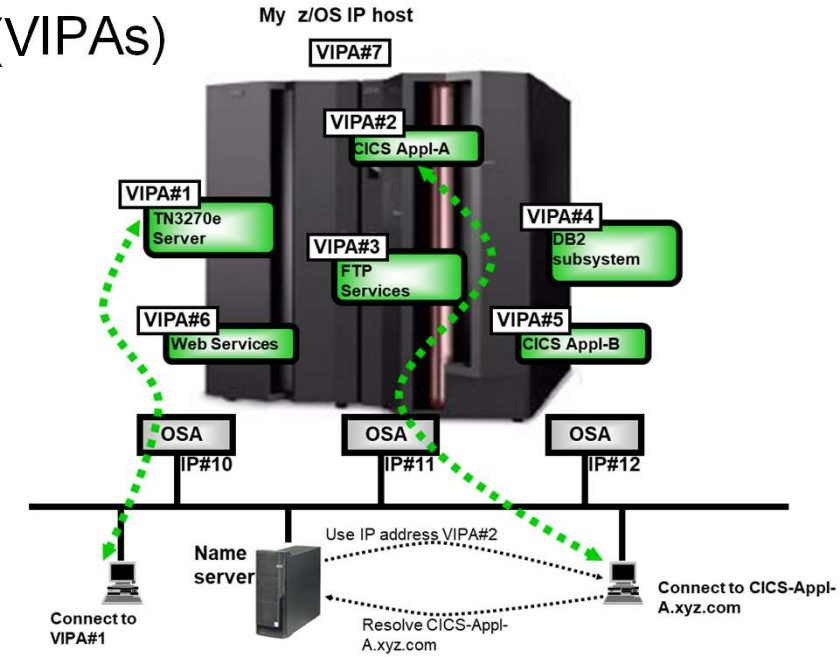
z/OS unique features: VIPAs, DVIPAs and Distributed DVIPAs (1 of 3)

Like everything else, z/OS can virtualize IP addresses.

A Virtual IP Address (VIPA) is an IP address that is not tied to a specific physical interface and instead represents a higher level concept such as a host, an application, or a load balancer.

There are several types of Virtual IP Addresses (VIPAs)

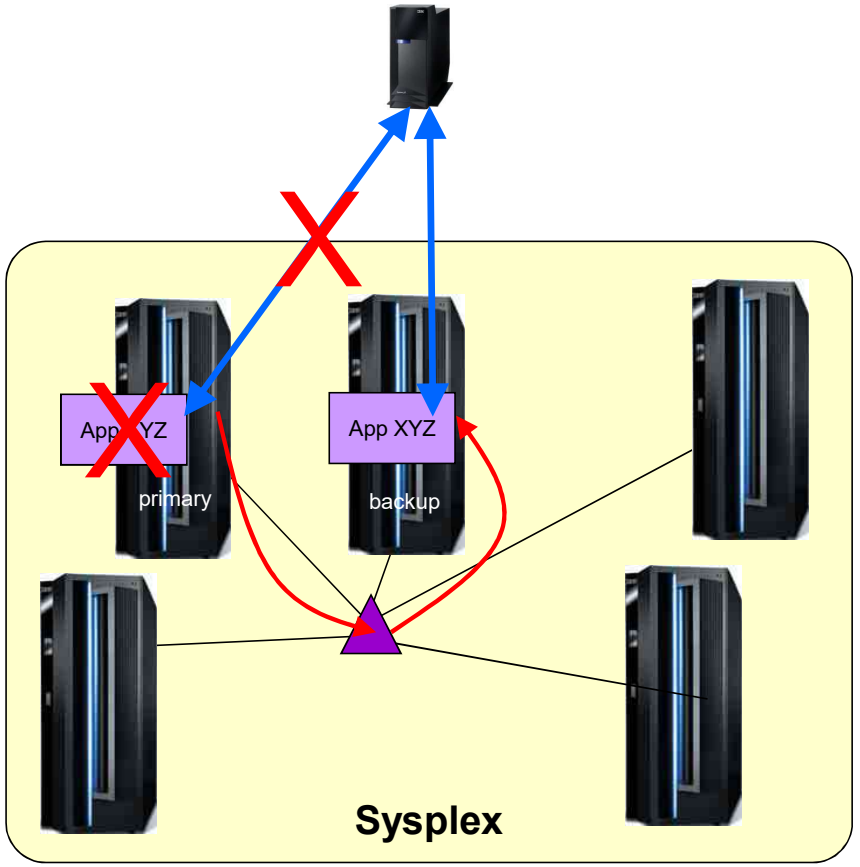
- Static VIPA
  - Allows application communications to continue even with physical network interface failures
  - As long as a single network interface is operational on the host, communications continue



z/OS unique features: VIPAs, DVIPAs and Distributed DVIPAs (2 of 3)

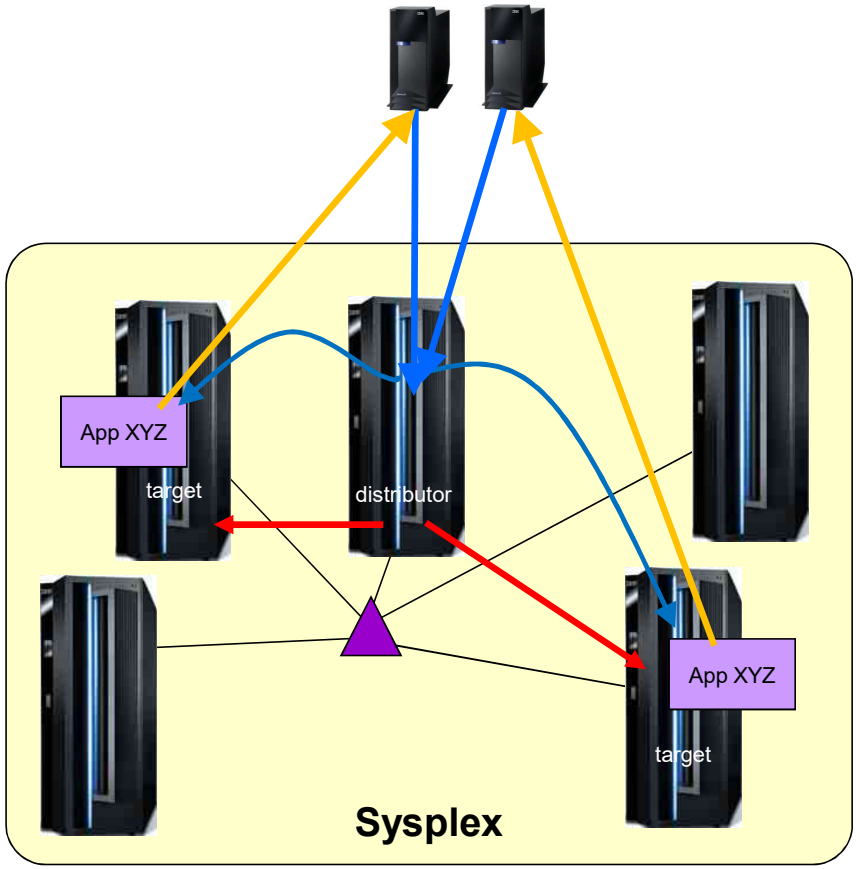
- Dynamic VIPA (DVIPA)
  - A VIPA that is defined in multiple z/OS systems within a sysplex
  - For TCP traffic only
  - Can move from one location to another through planned action or for failover
  - Uses the z/OS Coupling Facility to maintain state across the sysplex
  - Allows an address to move to a different system in the sysplex due to planned or unexpected outage

- Application-specific DVIPA
  - A DVIPA that is instantiated when a specific application binds to the address and deleted when the application terminates
  - Retains all the attributes of a defined DVIPA, but is bound to the existence of the application
  - Sometimes called a *VIPARANGE* DVIPA



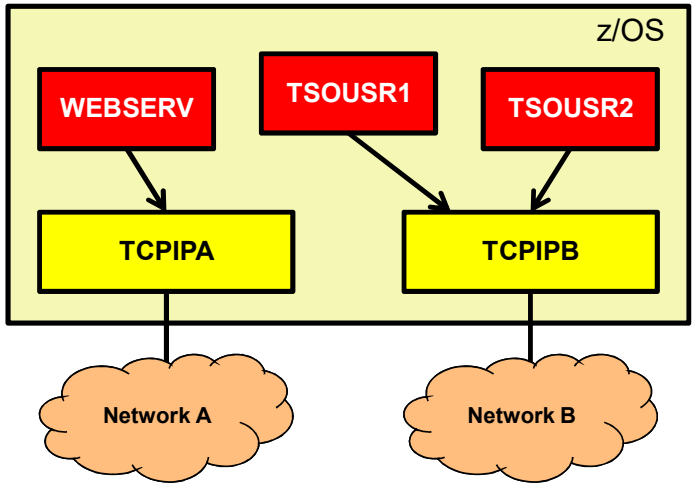
z/OS unique features: VIPAs, DVIPAs and Distributed DVIPAs (3 of 3)

- Distributed DVIPA
  - A DVIPA that is not only defined in multiple z/OS systems in the sysplex, but can be active in multiple places at the same time
  - For TCP traffic only
  - One stack is designated at the *distributor* (which owns the address) for a given Dynamic DVIPA, and the rest are *targets*.
  - The distributor distributes new inbound connections to the targets based on one of several distribution methods (including one based on WLM).
  - Allows multiple concurrent instances of same application to appear to outside world as a single system image



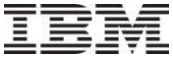
z/OS unique features: Multiple TCP/IP stacks on a single z/OS system

- Unlike most other operating systems, a single instance of z/OS can support multiple TCP/IP stacks (up to 8).
- While this is not recommended in most cases, there is a scenario where multiple stacks can be effective in isolating network traffic to specific applications within a z/OS system



Agenda

- Core concepts
- Common devices and terms
- Some z/OS-unique features
- **Network security – the typical overlap with your job**



As a z/OS security administrator, you may have to…

- Define user profiles for TCP/IP-related applications and middleware
  - TN3270E server
  - Policy Agent
  - IKED (for IPsec)
  - DMD
  - TRMD
  - FTP daemon
  - CSSMTP
  - NSSD (for IPsec)
  - syslogd
  - …and others
- Define and grant access to SAF profiles for TCP/IP-related resources:
  - All Communications Server resources are defined in the SERVAUTH class
  - OPERCMDS class to control access to certain TCP/IP-related commands
  - CSFSERV class to control access to ICSF resources when cryptographic protocols like TLS/SSL, IPsec and SSH are used
- Manage X.509 digital certificates in your ESM
  - Become friendly with your company's Public Key Infrastructure (PKI) team!
  - You will be the middleman between them and your application/middleware owners
  - If you need it, there are some good introductions to basic cryptography and digital certificates available on the web

The SAF SERVAUTH resource class

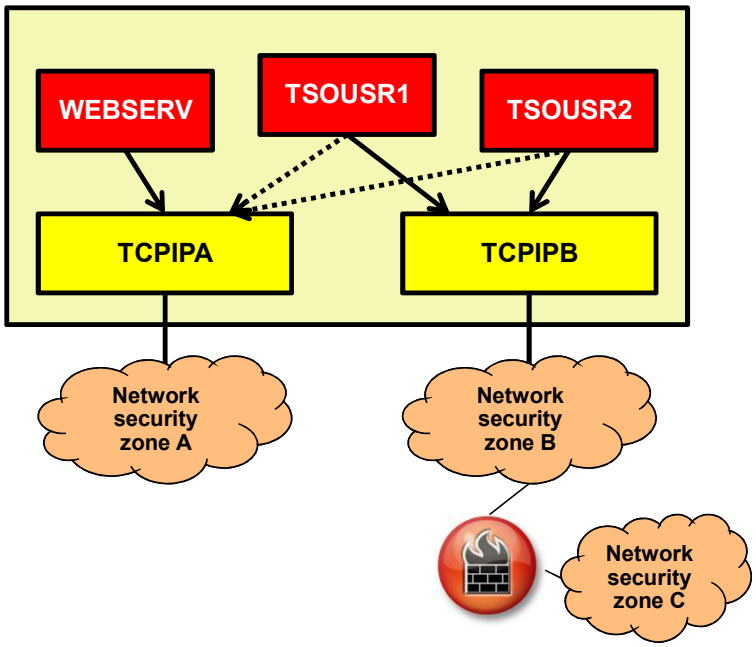
- The SERVAUTH resource class is used to specifically define and protect numerous TCP/IP-related resources
- General SERVAUTH profile format:

EZB.resource_category.system_name.jobname.resource_name

- EZB designates that this is a TCP/IP resource
- resource_category is a capability area to be controlled e.g. TN3270, Stack Access, etc.
- system_name is the name of the system (LPAR) - can be wild-carded (*)
- jobname is the jobname associated with the resource access request - can be wild-carded (*)
- optional resource_name - one or more qualifiers to indicate name of resource to be protected - can be wild-carded (*)

- To protect one of the supported TCP/IP resources, define a SERVAUTH profile with universal access NONE and then permit authorized user IDs to have READ access to that profile
- If using OEM security packages, beware of the differences between defined/not defined resource actions
- All the "traditional" SAF protection of datasets, authorized MVS and z/OS UNIX functions, etc. on a z/OS system applies to TCP/IP workload just as it applies to all other types of workload.
- Be careful with anonymous services such as anonymous FTP that can be configured to allow unauthenticated users access to selected MVS data sets and/or HFS files.

A few selected SERVAUTH resources (1 of 3): STACKACCESS



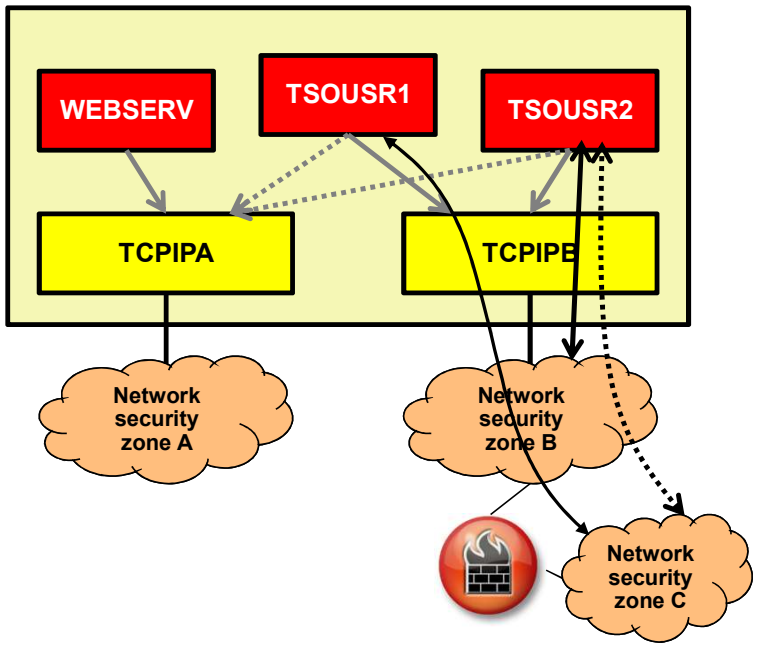
EZB.STACKACCESS.*.TCPIPA
WEBSRV permitted, all others not

- **A single z/OS system can run up to eight different TCP/IP stacks**
- **STACKACCESS limits local users' open sockets or use of TCP/IP stack services (e.g., get hostname, get hostid, etc.)**
- **Access to stack via sockets is allowed if the user has READ access to the following SERVAUTH class SAF resource:**

EZB.STACKACCESS.*sysname.stackname*

- **Define stack profile with UACC(NONE) and permit groups or individual users to allow them access to the stack**
- **In the example, TSOUSR1 and TSOUSR2 are not permitted to use TCPIPA**

A few selected SERVAUTH resources (2 of 3): NETACCESS




```
EZB.STACKACCESS.*.TCPIPA
      WEBSRV permitted, all others not

EZB.NETACCESS.*.TCPIPB.ZONEC
      TSOUSR1 permitted, all others not
```

- **Controls local user's access to network resources**
  - bind to local address
  - send/receive IP packets to/from protected zone
  - Network
  - Subnet
  - Individual host

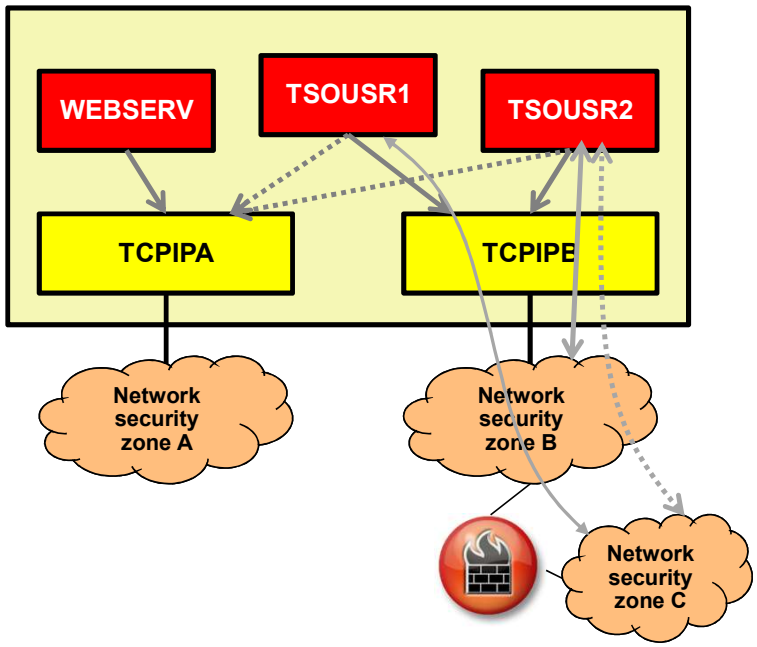
  (Note that firewalls can't distinguish between individual z/OS users)
- **Access to security zone is allowed if the user has READ access to the SERVAUTH class SAF resource associated with the zone:**

  EZB.NETACCESS.*sysname.stackname.zonename*
- **NETACCESS statement in TCP/IP profile defines security zones. For example, stack B may have:**

```
NETACCESS INBOUND OUTBOUND
  192.168.1.0     255.255.248.0 ZONEB
  192.168.0.0/16                ZONEC
  Default         0             WORLD
ENDNETACCESS
```

- **In the example, TSOUSR2 is not permitted to network security zone C**

A few selected SERVAUTH resources (3 of 3): PORTACCESS



- **Limits local users' access to *explicitly bound* ports**
- **Controls whether a started task or userid can establish itself as a server on a given TCP or UDP port.**
- **Access to use port is allowed if the user has READ access to the following SERVAUTH class SAF resource:**

```
EZB.PORTACCESS.sysname.stackname.SAFname
```

- **SAF keyword on PORT or PORTRANGE statement in TCP/IP profile defines SAF resource name. For example, stack A may have:**

```
PORT 80 TCP * SAF WEBPORT
```

```
EZB.STACKACCESS.*.TCPIPA
      WEBSRV permitted, all others not
EZB.PORTACCESS.*.TCPIPA.WEBPORT
      WEBSRV permitted, all others not
EZB.NETACCESS.*.TCPIPB.ZONEC
      TSOUSR1 permitted, all others not
```

- **In the example, only userid WEBSRV is permitted to establish itself as a server on port 80 on stack TCPIPA**
- **RESERVED keyword on PORT or PORTRANGE statement prohibits access for all users.**

Other SERVAUTH resources

There are 30+ different possible TCP/IP-related resource types to protect. Careful use of these can provide a significant level of security administrator-based control over use of TCP/IP-related resources on z/OS

- Command protection
  - ipsec
  - nssctl
  - pasearch
  - netstat
- Network management APIs
  - packet trace
  - realtime SMF data
  - connection data
- Application control
  - FTP port, command access and HFS access
  - Stack access prior to AT-TLS initialization
  - NSS certificate, service, client access
  - broadcast socket options
  - IPv6 advanced socket APIs
- Other resource restrictions
  - Fast Response Cache Accelerator (FRCA) page load
  - SNMP subagent access
  - DVIPA modification control

See *z/OS Communications Server IP Configuration Guide* chapter 3 for a complete list of SERVAUTH resources

Summary

- TCP/IP is a family of protocols
- IP networks are actually networks of networks
- IP, TCP and UDP are the key protocols that package and move data across the network
- There are two IP protocols – IPv4 and IPv6
- z/OS has some unique features that support resiliency, isolation and SNA access
- As a z/OS security administrator, you will be responsible for TCP/IP-related security on a number of levels
- From a SAF perspective, the SERVAUTH class is where most of the action is for securing TCP/IP resources
- Hopefully this presentation provides a good foundation for understanding the terms and concepts you will typically be exposed to as a z/OS security administrator.

Thank you!

Notices and disclaimers (1 of 2)

Notices and disclaimers (2 of 2)

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 about this publication and cannot confirm the accuracy of 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. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

.