

IPv6 on z/OS – Part 2

Mike Fitzpatrick – mfitz@us.ibm.com
IBM Raleigh, NC

Thursday, March 15th, 9:30am
Session: 10831

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® | • GDDM® | • Language Environment® | • Rational Suite® | • zEnterprise |
| • AIx® | • GDPS® | • MQSeries® | • Rational® | • zSeries® |
| • alphaWorks® | • Geographically Dispersed Parallel Sysplex | • MVS | • Redbooks | • z/Architecture |
| • AnyNet® | • HiperSockets | • NetView® | • Redbooks (logo) | • z/OS® |
| • AS/400® | • HPR Channel Connectivity | • OMEGAMON® | • Sysplex Timer® | • z/VM® |
| • BladeCenter® | • HyperSwap | • Open Power | • System i5 | • z/VSE |
| • Candle® | • i5/OS (logo) | • OpenPower | • System p5 | |
| • CICS® | • i5/OS® | • Operating System/2® | • System x® | |
| • DataPower® | • IBM eServer | • Operating System/400® | • System z® | |
| • DB2 Connect | • IBM (logo)® | • OS/2® | • System z9® | |
| • DB2® | • IBM® | • OS/390® | • System z10 | |
| • DRDA® | • IBM zEnterprise™ System | • OS/400® | • Tivoli (logo)® | |
| • e-business on demand® | • IMS | • Parallel Sysplex® | • Tivoli® | |
| • e-business (logo) | • InfiniBand® | • POWER® | • VTAM® | |
| • e business (logo)® | • IP PrintWay | • POWER7® | • WebSphere® | |
| • ESCON® | • IPDS | • PowerVM | • xSeries® | |
| • FICON® | • iSeries | • PR/SM | • z9® | |
| | • iSeries | • pSeries® | • z10 BC | |

* 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.
- UNIX 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.

Other IPv6 Sessions at this SHARE

10395: IPv6 Tunneling Technologies	Wednesday, March 14 th : 1:30pm
10397: IPv6 Basics	Wednesday, March 14 th : 8:00am
10396: IPv6 Addressing	Wednesday, March 14 th : 9:30am
10398: IPv6 Design	Wednesday, March 14 th : 3:00pm
10399: IPv6 Implementation	Wednesday, March 14 th : 4:30pm
10400: IPv6 Planning	Wednesday, March 14 th : 11:00am
10401: Transitioning to IPv6	Thursday, March 15 th : 3:00pm
10413: Common IPv6 Mistakes	Friday, March 16 th : 8:00am
10414: IPv6 Deep Dive	Friday, March 16 th : 11:00am
10830: IPv6 on z/OS - Part 1	Thursday, March 15 th : 8:00am
10836: IPv6 Configuration on z: Hands-on Lab	Thursday, March 15 th : 4:30pm



Agenda

- Enabling IPv6 on z/OS
- What functions are IPv6-enabled?
- Planning how to get there



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.

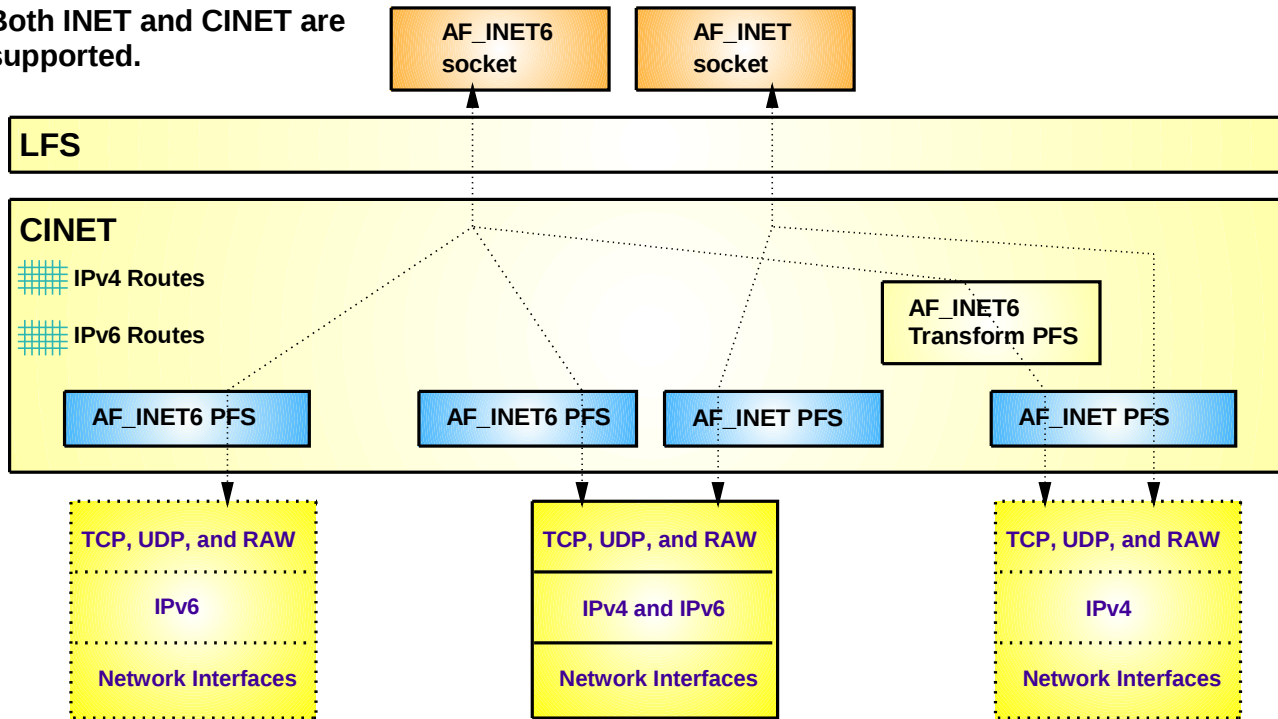
Enabling IPv6 on z/OS



Enabling IPv6 support on z/OS

IPv6 is enabled at an LPAR level via an option in BPXPRMxx to enable AF_INET6 support. Both INET and CINET are supported.

When IPv6 is enabled, a z/OS TCP/IP stack will always have an IPv6 Loopback interface. You can define real IPv6 interfaces in addition to the loopback interface.



IPv6-only TCP/IP Stack

This will not be the case on z/OS for the foreseeable future! An AF_INET6 stack is required to also support AF_INET!

Dual Mode TCP/IP Stack

A z/OS TCP/IP stack will always come up as dual-mode if AF_INET6 is enabled in BPXPRMxx

IPv4-only TCP/IP Stack

(such as an OEM TCP/IP stack)

- ▶ Existing AF_INET sockets programs will continue to work as they always did - no difference in behavior or support.
- ▶ AF_INET6 enabled sockets programs will be able to communicate with IPv4 partners (just as before they were changed to support IPv6), but in addition to that they will also be able to communicate with IPv6 partners.

Netstat output format LONG or SHORT

- When IPv6 is enabled, most netstat reports will look different because of the potential for long IPv6 addresses.
 - Without IPv6 enabled, netstat uses what is known as a SHORT report format
 - It is possible to have both local and remote IPv4 address in one 80-character line
 - You can override the SHORT format by coding IPCONFIG FORMAT LONG
 - With IPv6 enabled, netstat uses a LONG report format
 - Each IPv6 address may potentially be up to 45 characters long, which makes it impossible to have both local and remote IPv6 addresses in a single 80-character line
- Make sure you update any netstat screen-scraping REXX programs you might have developed in the past!

```

MVS TCP/IP NETSTAT CS V1R12          TCPIP Name: TCPCS          12:50:02
User Id  Conn      State
-----  ----      -
MYINETD1 00000025 Listen
  Local Socket:  9.42.104.161..23
  Foreign Socket: 0.0.0.0..0
TN3270A  00000045 Listen
  Local Socket:  ::..23
  Foreign Socket: ::..0
TN3270A  00001B5E Establish
  Local Socket:  ::ffff:9.42.105.45..23
  Foreign Socket: ::ffff:9.50.52.109..58646
Application Data: EZBTNSRV TCPABC81 TS010001 ET B
  
```

Accessing z/OS over IPv6 from a remote site

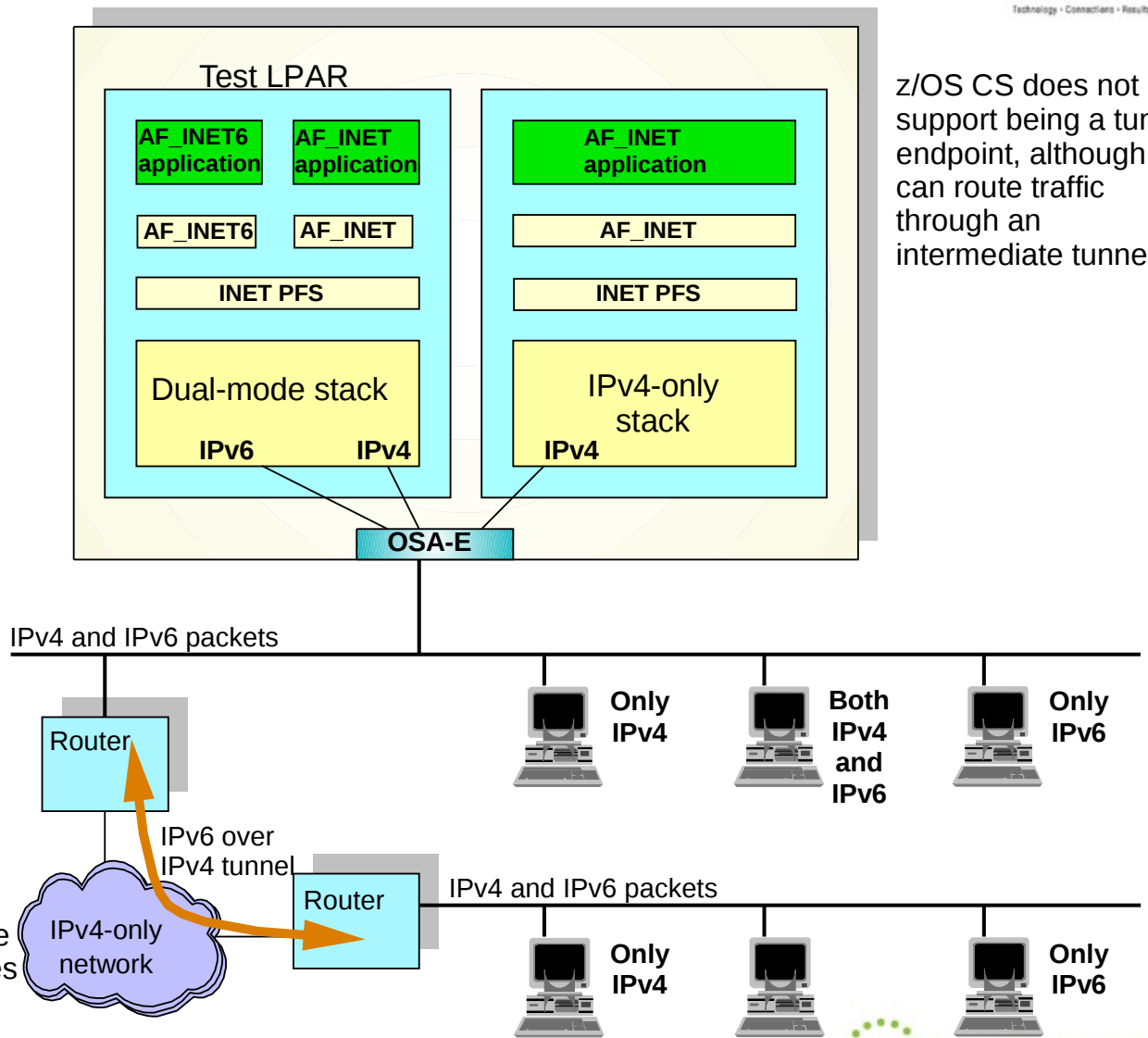
Remember:

You can enable IPv6 today on z/OS without impact to your existing IPv4 users.

Do it on your test system – initially without defining any IPv6 interfaces.

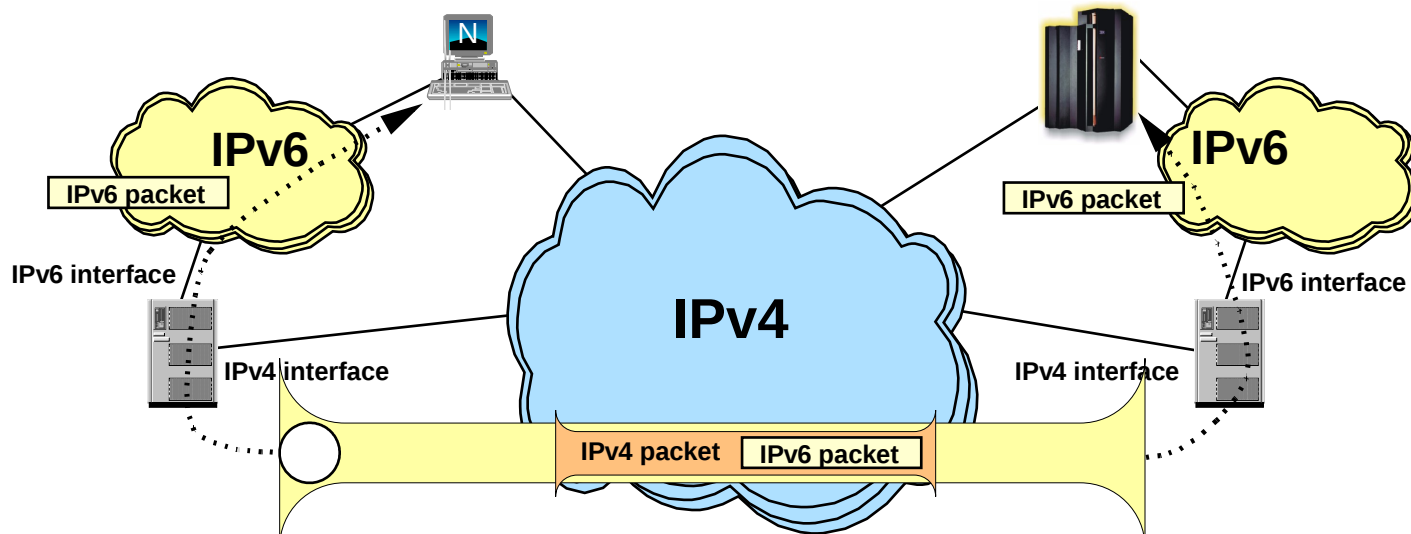
All IPv4 communication continues to work as before.

Use IPv6 over IPv4 tunneling when native IPv6 connectivity does not exist



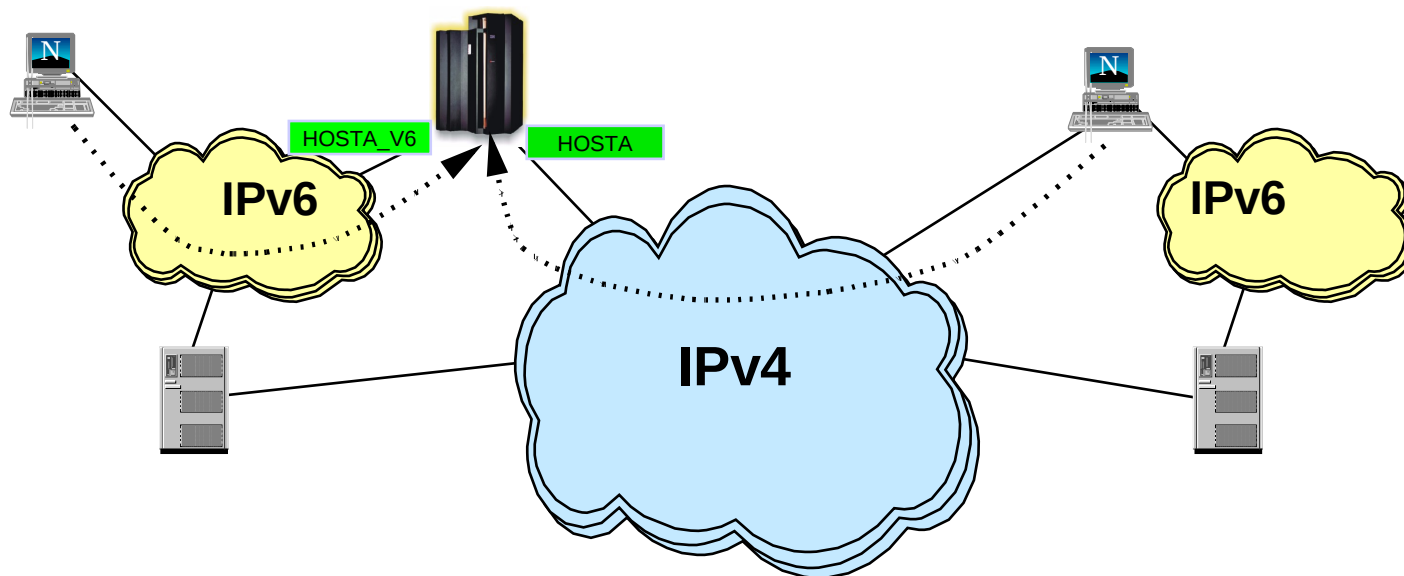
z/OS CS does not support being a tunnel endpoint, although it can route traffic through an intermediate tunnel

IPv6 paths are preferred over IPv4



- IPv6 connectivity is preferred over IPv4
 - In many cases, only if one of the nodes does not support IPv6 will IPv4 be used
 - Can lead to undesirable paths in the network
 - Data may be tunneled over the IPv4 network even when a native IPv4 path exist
- May lead to longer connection establishment to an AF_INET application on a dual-stack node
 - IPv6 addresses will be tried before attempting to connect via IPv4
 - A "well behaved" client will cycle through all addresses returned and try the IPv4 address
 - But this takes time and network resources
 - And not all clients are "well behaved" or bug-free

Use of distinct IPv4 and IPv6 host names



- To avoid undesirable tunneling, configure two host names in DNS
 - Continue to use the existing host name for IPv4 connectivity
 - Create a new host name to be used for IPv6 connectivity
 - Optionally, a third host name which may be used for both IPv4 and IPv6 can be configured
- Client chooses type of connection based on host name
 - Using the existing host name results in IPv4 connectivity
 - Using the new host name results in IPv6 connectivity

Note: Use of distinct host names is only necessary during the initial transition phases when native IPv6 connectivity does not exist

What functions are IPv6-enabled?



Defining IPv6 interfaces

- IPv6 interfaces are defined using an INTERFACE statement in the TCP/IP profile
 - Combines the definitions of DEVICE, LINK and HOME into one statement
 - In order for one physical device to support both IPv4 and IPv6 traffic, DEVICE, LINK and HOME statements (or INTERFACE statement) have to be specified in the profile to define the IPv4 side and an INTERFACE statement must be specified to define the IPv6 side
- A single IPv6 interface may have one or more IPv6 addresses at any given time
 - There will always be a link-local address, which is automatically assigned during interface activation
 - There may be 0-n local-unicast and/or global IPv6 addresses as well
- For physical interfaces, IP addresses (except for the link-local address) may be manually configured or may be autoconfigured

```
INTERFACE OSAQDI015 DEFINE IPAQENET6 PORTNAME OSAQDI01
```

```
INTERFACE OSAQDI025 DEFINE IPAQENET6 PORTNAME OSAQDI02  
IPADDR FC00::9:67:115:5  
2001:0DB8::9:67:115:5
```

Netstat DEVLINKS/-d

```
MVS TCP/IP onetstat CS V1R12          TCPIP Name: TCPCS          12:55:20
DevName: OSAQDI04          DevType: MPCIPA
DevStatus: Ready
LnkName: OSAQDIOLINK          LnkType: IPAQENET          LnkStatus: Ready
NetNum: 0          QueSize: 0          Speed: 0000000100
IpBroadcastCapability: No
CfgRouter: Non          ActRouter: Non
ArpOffload: Yes          ArpOffloadInfo: Yes
ActMtu: 1492
VLANid: 1260          VLANpriority: Enabled
ReadStorage: GLOBAL (8064K)          InbPerf: Balanced
ChecksumOffload: Yes
BSD Routing Parameters:
MTU Size: 00000          Metric: 00
DestAddr: 0.0.0.0          SubnetMask: 255.255.255.192
Multicast Specific:
Multicast Capability: Yes
Group          RefCnt
-----          -
224.0.0.1          0000000001
Link Statistics:
BytesIn          = 11476
Inbound Packets          = 10
Inbound Packets In Error          = 0
Inbound Packets Discarded          = 0
Inbound Packets With No Protocol          = 0
BytesOut          = 6707
Outbound Packets          = 10
Outbound Packets In Error          = 0
Outbound Packets Discarded          = 0
```

Netstat DEVLINKS/-d (continued)

```
IntfName: OSAQDI046          IntfType: IPAQENET6 IntfStatus: Ready
NetNum: 0   QueSize: 0   Speed: 0000000100
MacAddress: 000629DC21BC
SrcVipaIntf: VIPAV6
DupAddrDet: 1
CfgRouter: Pri                ActRouter: Pri
RtrHopLimit: 5
CfgMtu: 4096                  ActMtu: 1492
VLANid: 1261                  VLANpriority: Enabled
IntfID: 0000:0000:0000:0001
ReadStorage: GLOBAL (8064K)   InbPerf: Balanced

Packet Trace Setting:
Protocol: *                    TrRecCnt: 00000000  PckLength: FULL
SrcPort: *                     DestPort: *
IpAddr/PrefixLen: 9::44/128

Multicast Specific:
Multicast Capability: Yes
RefCnt      Group
-----
0000000001  ff02::1:ff15:5
0000000001  ff02::1:ff00:2

Interface Statistics:
BytesIn                = 12655
Inbound Packets        = 12
Inbound Packets In Error = 0
Inbound Packets Discarded = 0
Inbound Packets With No Protocol = 0
BytesOut               = 4590
Outbound Packets       = 11
Outbound Packets In Error = 0
Outbound Packets Discarded = 0
```

IPv6 VIPA and SOURCEVIPA

- Static VIPAs are defined on a VIRTUAL6 interface
 - Each VIRTUAL6 interface must be manually configured with one or more IPv6 addresses
- Use the SOURCEVIPAINterface parameter to associate a physical interface to a specific VIRTUAL6 interface
 - No ordering considerations like DEVICE/LINK/HOME for IPv4
 - The TCP/IP stack will choose the "best" address as the source IP address using the Default Address Selection algorithms defined by the IETF
 - More than one physical interface can point to the same VIRTUAL6 interface
- IPCONFIG6 SOURCEVIPA definition makes the SOURCEVIPA function available for all IPv6 interfaces configured with SOURCEVIPAINterface.

IPCONFIG6 SOURCEVIPA

```
INTERFACE VIPAV61 DEFINE VIRTUAL6  
IPADDR FC00::9:67:115:5 2001:0DB8::9:67:115:5
```

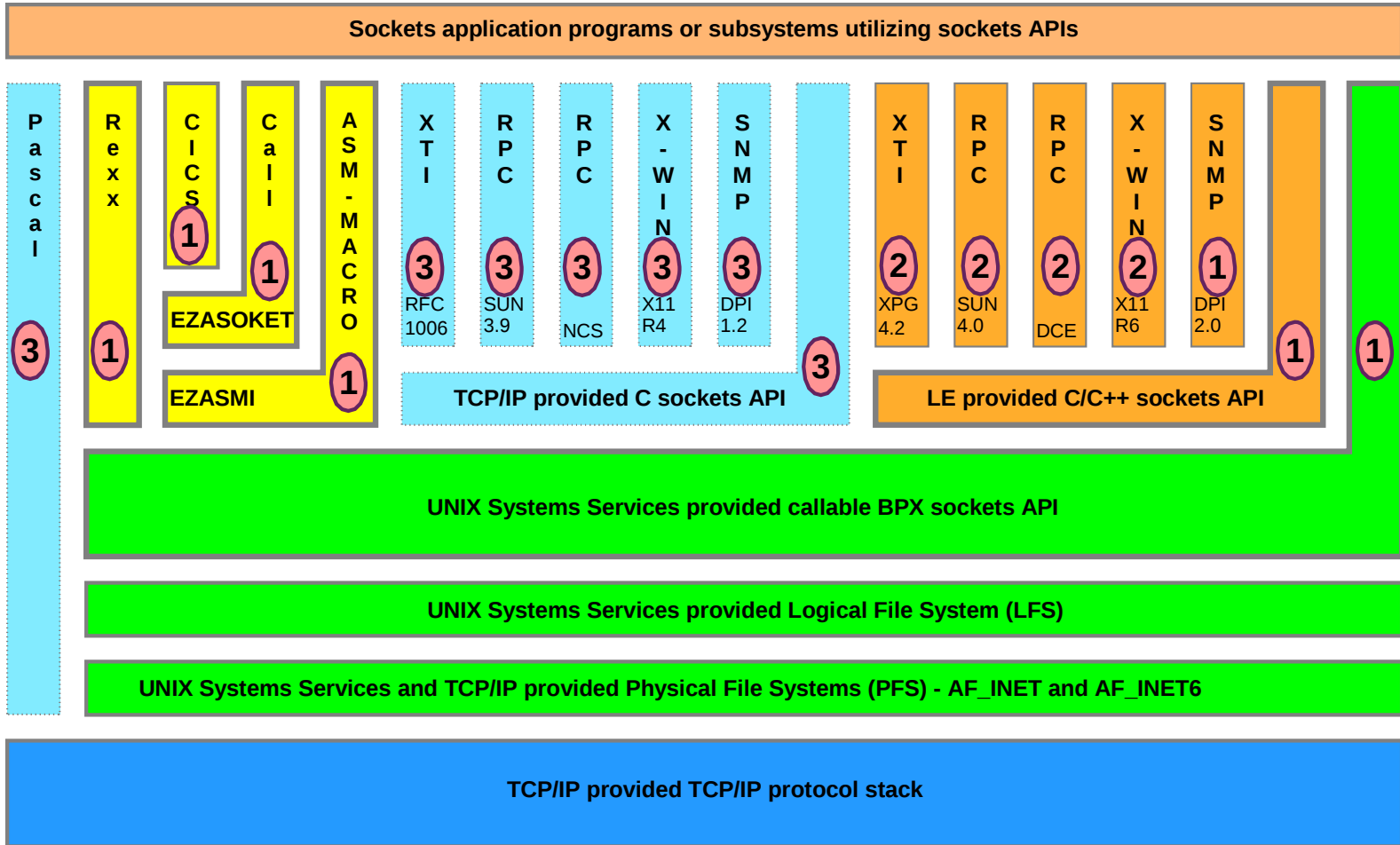
```
INTERFACE VIPAV62 DEFINE VIRTUAL6  
IPADDR FC00::9:67:115:6 2001:0DB8::9:67:115:6
```

```
INTERFACE OSAQDI016 DEFINE IPAQENET6 PORTNAME OSAQDI01  
SOURCEVIPAINT VIPAV61
```

```
INTERFACE OSAQDI026 DEFINE IPAQENET6 PORTNAME OSAQDI02  
SOURCEVIPAINT VIPAV62
```

```
INTERFACE OSAQDI036 DEFINE IPAQENET6 PORTNAME OSAQDI03
```

Sockets-related AF_INET6 enablement



- 1 Currently supported
- 2 Possible future candidates
- 3 Not planned

z/OS Communications Server Applications IPv6-enabled



- IPv6-enabled applications in z/OS
 - inetd
 - ftp and ftpd
 - telnetd
 - USS rshd and rexecd servers
 - USS rexec client
 - ping
 - tracert
 - netstat
 - tftpd (trivial file transfer server)
 - syslogd
 - dcas (digital certificate access server)
 - sntpd (simple network time protocol server)
 - sendmail 8.12.x (new port of sendmail picks up IPv6 enablement too)
 - MVS rshd/rexecd server
 - TSO rsh/rexec clients
 - Updated version that can be used in all z/OS environments (batch, TSO, REXX, etc.)
 - New UNIX rsh client that is IPv6-enabled from start
 - CICS Listener (including CICS socket APIs)

FTP

- FTP server
 - To enable IPv6 support in the FTP server, activate IPv6 stack support
 - No new configuration commands are provided or needed to enable IPv6 support
 - User Exit routines
 - Update server exit routines for IPv6 addressing
 - Trace and Extended Trace
 - Update DUMP IPADDR() and DEBUG IPADDR() as needed
 - NETRC data set
 - Update with IPv6 addresses as needed
 - SMF recording
 - Update SMF statements in client and server FTP.DATA commands
- FTP client
 - For the client, you may specify the host as an IPv4 address, a hostname, an IPv4 mapped IPv6 address, or as an IPv6 address
 - Examples:

ftp fc00:197:11:105::1

ftp 9.67.21.33 and ftp ::ffff:9.67.21.33 are equivalent

ftp linuxipv6.tcp.raleigh.ibm.com

TN3270

- To enable IPv6 support in the TN3270 server, activate IPv6 stack support
 - No new configuration commands are provided or needed to enable IPv6 support
- IPv6-enable the TN3270 server
 - Support clients with IPv6 addresses
 - Support IPv6 addresses in USS messages, displays, command responses, etc.
 - Support IPv6 addresses as client identifiers for all mapping statements in TN3270 server configuration that allows an IP address client identifier
 - Includes SSL/TLS support
 - Changes made to VTAM to support TN3270 visibility when clients are IPv6 clients
 - IPv6 addresses are passed to VTAM
 - VTAM displays that include IP addresses are enhanced to accommodate IPv6 addresses

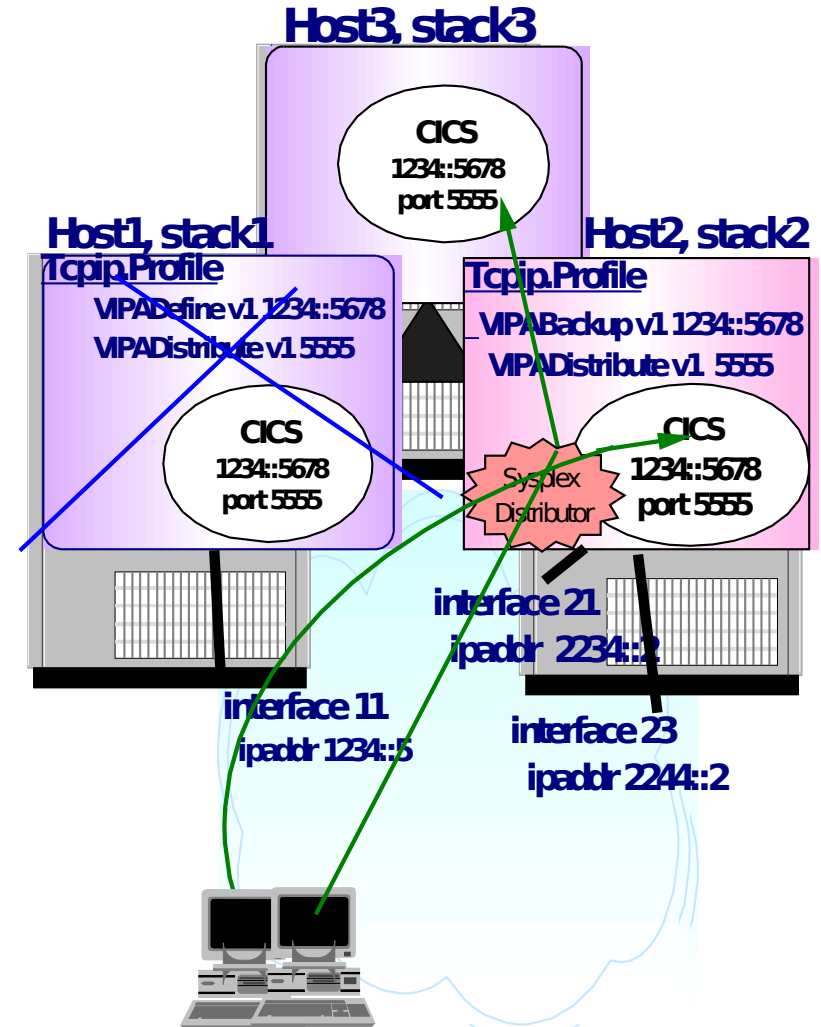
Enterprise Extender



- Enterprise Extender has support for IPv6 in z/OS
 - Allows Enterprise Extender to exploit an IPv6-enabled network
 - Architectural changes needed since HPR passes IP addresses in protocol data and is supported on multiple platforms
 - Changes to VTAM exits to pass IPv6 addresses, hostnames, and port numbers:
 - SME (Session Management exit)
 - Login exit
- IPv6 support requires use of the HOSTNAME keyword (start option, GROUP, path definition)
 - Existing IPADDR keywords (start option, path definition in SMN) are IPv4-only
- EE Connection networks are IPv4-only or IPv6-only
 - Nodes supporting both IPv4 and IPv6 must define an IPv4 VRN and an IPv6 VRN

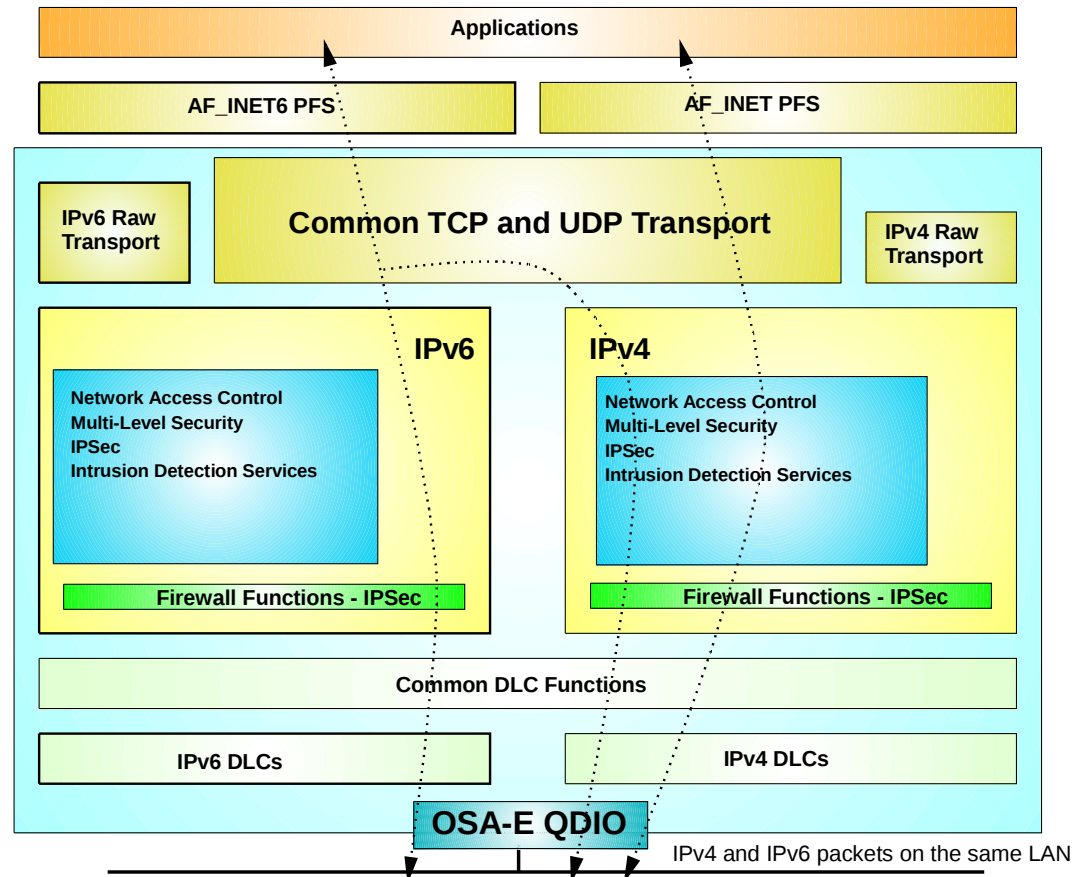
Sysplex functions that support IPv6

- Almost all Sysplex functions support IPv6
 - Dynamic VIPA (DVIPA)
 - Dynamic VIPA Takeover
 - Sysplex Distributor
 - Sysplex Sockets
 - TCPSTACKSOURCEVIPA
 - Sysplexports
 - Fast Connection Reset after System Failure
 - Enhance Workload Distribution (Application Server Affinity)
 - Dynamically Assign Sysplex Ports
 - Activation of DVIPAs through VIPABACKUP
 - DYNAMICXCF and SOURCEVIPAINIT
 - Sysplex Distributor load balancing algorithms
 - Sysplex Distributor Policy
 - z/OS Load Balancing Advisor



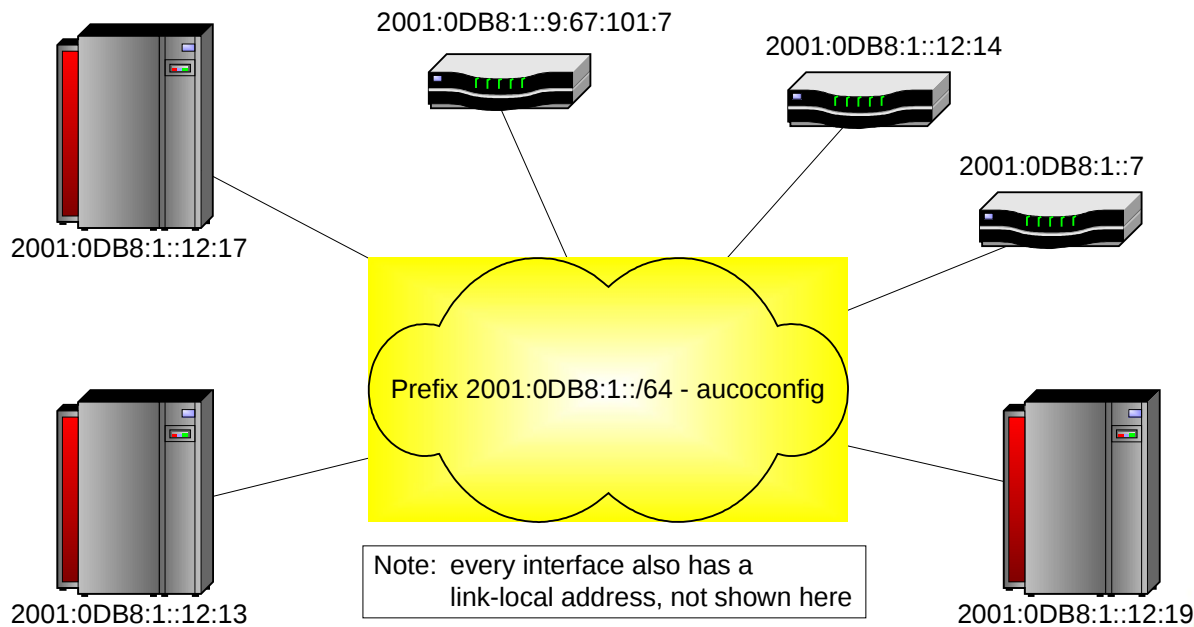
Securing your IPv6 network

- z/OS V1R5 provided the first set of security features for the IPv6 transport
 - Includes support for both Network Access Control and Multi-Level Security
- z/OS V1R8 provided IPsec and Firewall filters support for both IPv4 and IPv6
 - IKEv2 support (for both IPv4 and IPv6 made available in z/OS V1R12)
 - IPv6 IPsec currently does not support Sysplex Wide Security Associations (SWSA)
- Intrusion Detection Services (IDS) support for IPv6 shipped in z/OS V1R13



Dynamic routing

- Support IPv6 RIP (RIPng) in V1R5 and IPv6 OSPF (OSPFv3) in V1R6
 - Implementation done in OMPROUTE
 - One and the same daemon for both IPv4 (RIP and OSPF) and IPv6 (RIPng and OSPFv3)
- Based on IPv4 specifications with IPv6-specific updates
 - IPv6 RIP includes minimal changes
 - Replacement for RIPv1 and RIPv2 used in IPv4 networks
 - IPv6 OSPF is protocol independent
 - Separate IP addressing and network topology where possible
- Needed for Sysplex-related functions such as dynamic IPv6 VIPA movement



Managing your IPv6 network

- Network management SNMP support
 - Support SNMP agent (OSNMPPD)
 - DPI 2.0 enabled for AF_INET6 (used between SNMP subagents and SNMP manager)
 - Support TCPIP (stack) subagent
 - osnmp command
 - The trap forwarder daemon enabled for AF_INET6
 - IPv6 MIB support
 - New RFCs have been published that are IP version neutral - support will gradually converge from supporting version-specific MIBs to the new version-neutral MIBs
 - *RFC 2011 (IP and ICMP)*
 - *RFC 2012 (TCP)*
 - *RFC 2096 (IP routes)*
 - *RFC 2333 (Interfaces) - this one is not version neutral*

Testing network connectivity

- Ping and Traceroute support for IPv6
 - IPv6 IP addresses, or host names that resolve to IPv6 IP addresses, can be used for destinations
 - IPv6 IP addresses can be used as the source IP address for the command's outbound packets
 - IPv6 IP addresses or interface names can be used as the outbound interface
 - A new ADDRTYPE/-A command option can be specified to indicate whether an IPv4 or IPv6 IP address should be returned from host name resolution
- IPv4-mapped IPv6 IP addresses are not supported for any option value

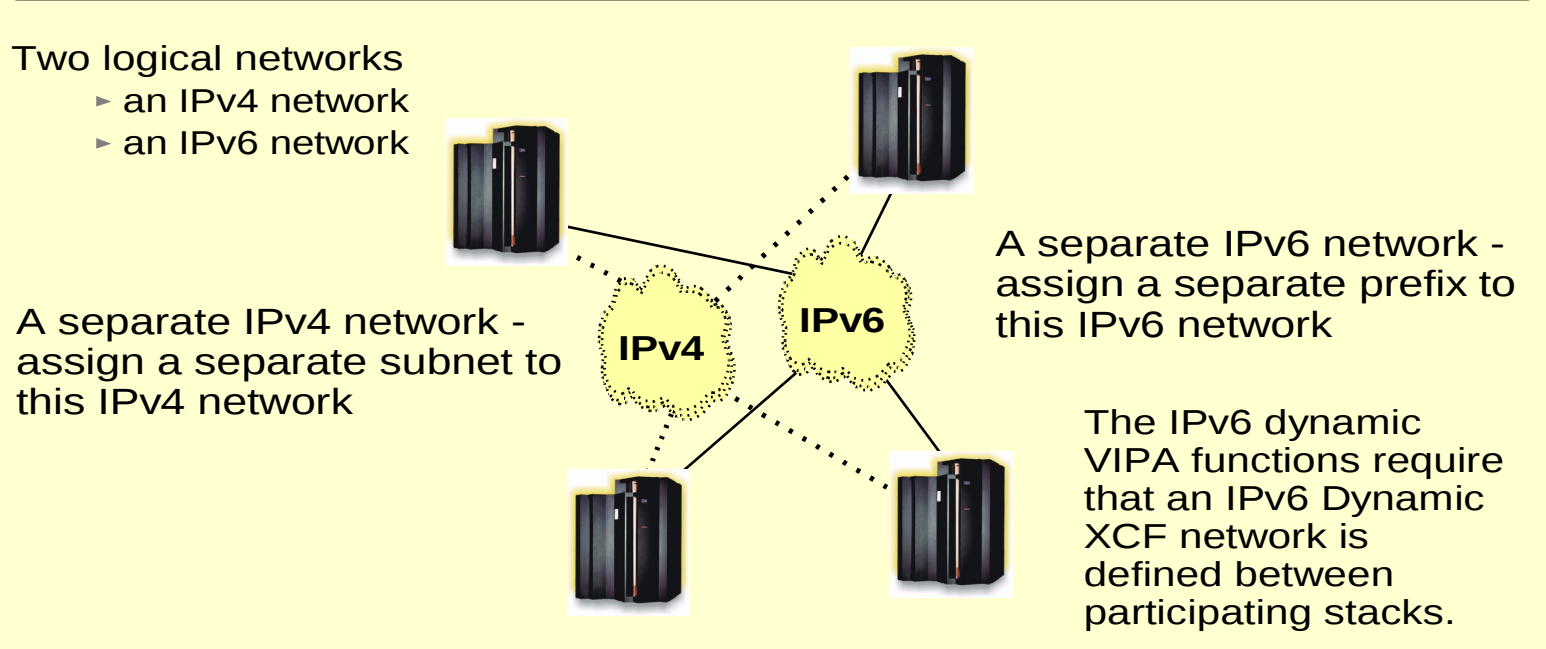
Planning how to get there



Steps for moving to an IPv6 Environment

- **Network access**

- A LAN can carry both IPv4 and IPv6 packets over the same media
- An OSA-EXPRESS port can be used for both IPv4 and IPv6
- Update TCP/IP Profile to include the INTERFACE statement(s) for any IPv6 interfaces
- For LPAR-LPAR communication for IPv6, several options exist:
 - Using QDIO to a shared LAN (or a Shared OSA)
 - MPCPTP6 interfaces (via XCF if on the same sysplex or ESCON CTC links)
 - IPv6 HiperSocket connections (if on the same CEC)



Steps for moving to an IPv6 Environment...

- **IPv6 address selection**

- Obtain an address block from your ISP, use one of your IPv4 addresses to create a 6to4 prefix, or create local IPv6 unicast addresses using the FC00::/7 prefix
 - You provide the MAC address of one of your OSA adapters, and sites such as <http://www.kame.net/~suz/gen-ula.html> can generate a ULA prefix for you
 - Remember: IPv6 sitelocal addresses have been deprecated and should not be used
- IPv6 addresses can be assigned to the IPv6 Interfaces and static VIPAs
- Addresses can be manually configured on the INTERFACE statement in the TCP/IP Profile or auto-configured using Neighbor Discovery Stateless Auto-configuration
 - VIPA addresses must be manually configured

Steps for moving to an IPv6 Environment...

- **INET or Common INET**

- Both are supported for IPv6, but INET is much simpler
- Running IPv4-only and dual-mode stacks under CINET is not recommended - run dual-mode stacks in a separate LPAR from IPv4-only stacks
 - All z/OS TCP/IP stacks in an LPAR are either IPv4-only or dual-mode - (Based on your BPXPRMxx definitions)
 - Only case where this could become an issue is if you start CA's TCPAccess TCP/IP stack side-by-side with a z/OS TCP/IP stack in an LPAR that have been enabled for IPv6 in the BPXPRMxx parmlib member
- AF_INET6 NETWORK statement must be coded in BPXPRMxx before starting IPv6-enabled stacks

Steps for moving to an IPv6 Environment...

- TCP/IP profile – IP Configuration for IPv6
 - Basic IPv6 configuration is done using the IPCONFIG6 statement
 - Similar to IPCONFIG, which continues to be used for IPv4
 - Separate statements for IPv4 and IPv6 allow different values to be specified for IPv4 and IPv6
 - Most of the defaults on the IPCONFIG6 statement are good choices
 - However, we recommend that you code the SOURCEVIPA parameter
 - SOURCEVIPA allows a VIPA to be used as the source IP address for connections which are established by this node
 - It also allows DNS address-to-name translation to work
 - It is not enabled by default, but is more important in an IPv6 environment
 - You may want to enable IP forwarding using the DATAGRAMfwd parameter
 - The default is to **not** forward IP packets, the same as for IPv4

IPCONFIG6 SOURCEVIPA
DATAGRAMFWD

Steps for moving to an IPv6 Environment...

- **DNS setup**

- A DNS BIND 9 Name Server can be used for both IPv4 and IPv6 resources
- Continue to use the existing host name for IPv4 connectivity to avoid possible disruption in network connectivity and IPv4-only applications on an IPv6-enabled stack
- Create a new host name to be used for IPv6 and IPv4 connectivity
- Optionally, a third host name which may be used only for IPv6 can be configured
- If using stateless auto-configuration to define IPv6 addresses, static VIPA addresses should be stored in DNS since the auto-configured addresses will change over time

- **Never** add link-local addresses to DNS
 - They can't be used beyond the link on which they are defined, and aren't intended for general-purpose applications

- **Resolver setup**

- The Resolver sends queries to DNS server using IPv4 or IPv6 (IPv6 support added in z/OS V1R12)
 - If the IPv4 protocol is used to communicate, it does not affect what type of records are returned (you can still resolve host names to IPv6 addresses and vice-versa)
- Resolver communication with DNS name servers
 - Name query sends AAAA query to DNS and receives AAAA records in response
 - Reverse query sends PTR query to the 'ip6.arpa' domain and receives results from the 'ip6.arpa' domain

Steps for moving to an IPv6 Environment...

- **Resolver setup....**

- Use `/etc/ipnodes` as the local host file for IPv6 name queries
 - Local database that associates host names with IP addresses
 - May be used to locate both IPv4 and IPv6 addresses (using the COMMONSEARCH System Resolver option)
 - Extended version of `/etc/hosts`
 - Uses the same format as `/etc/hosts`, but may be used to store both IPv4 and IPv6 addresses
- `/etc/hosts` may continue to be used to store IPv4 addresses
 - But may not be used to store IPv6 addresses (same is true for files created with MAKESITE utility - HOSTS.SITEINFO and HOSTS.ADDRINFO)
- May want to consider using a local host file for early testing
 - Use the `LOOKUP LOCAL|DNS` resolver directive

9.67.43.100	HOSTNAME1	
9.67.43.126	HOSTNAME2	
9.67.43.222	HOSTNAME3	HOSTNAME3_IPV4
129.34.128.245	HOSTNAME4	
1::2	HOSTNAME2_IPV6	
1:2:3:4:5:6:7:8	HOSTNAME3_IPV6	

Recommendations when adding IPv6 addresses to DNS



- Add Static VIPAs in DNS
 - You don't need to add addresses assigned to physical interfaces if using VIPA and SOURCEVIPAs
 - z/OS autoconfigured addresses are not suitable for placement in DNS
 - May (and likely will) change each time a z/OS stack is recycled
 - If you need to place addresses assigned to physical interfaces in DNS, then you should manually configure the addresses
- Configure two (and optionally three) host names in DNS
 - Continue to use the existing host name for IPv4 connectivity
 - Create a new host name to be used for IPv6 and IPv4 connectivity
 - Optionally, a third host name which may be used only for IPv6 can be configured
- Be careful when adding Unique Local Unicast addresses to DNS
 - Unique Local unicast addresses are not globally unique and must not be returned to hosts outside the local site
 - Similar to how private addresses are handled in IPv4
- **Never** add link-local addresses to DNS
 - They can't be used beyond the link on which they are defined, and aren't intended for general-purpose applications

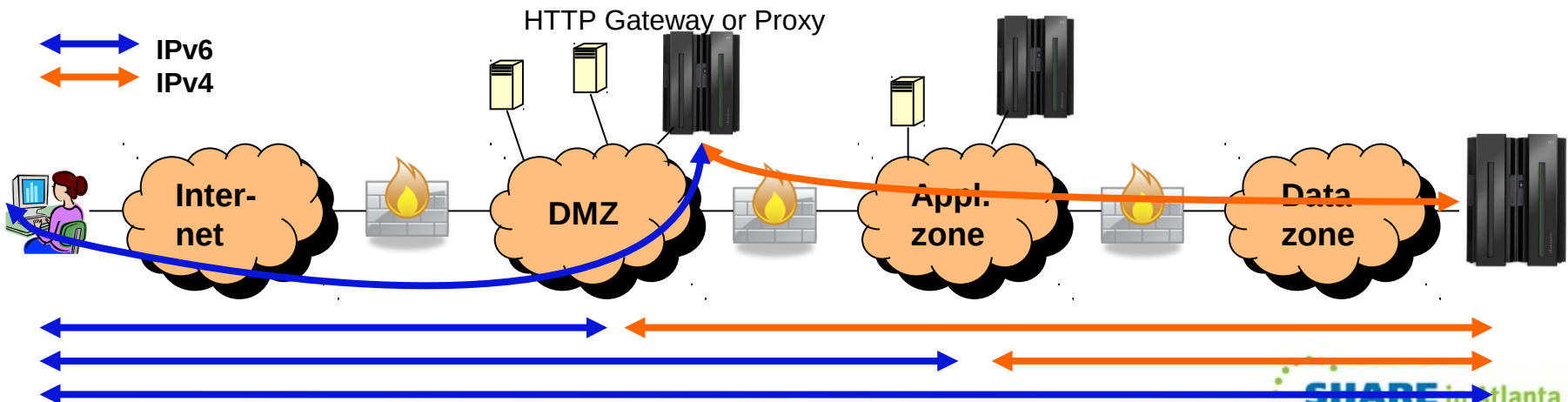
Resolver communication with DNS Name Server



- The results of Resolver queries varies based on interface availability
 - Resolver may omit IPv4 or IPv6 results if there aren't any physical interfaces which support the network protocol
 - Resolver sorts the addresses returned based on local interface availability
 - Default Address Selection algorithms govern both source address selection and destination address selection
 - Destination Address Selection is performed by Resolver as part of the name-to-address mapping
 - Source Address Selection is performed by the TCP/IP stack after the destination address is chosen
 - RFC 3484 “*Default Address Selection for Internet Protocol version 6 (IPv6)*” defines configurable rules for how parts of the source and destination IP address selection logic works – the default source and destination IP address selection
 - This rule-based logic kicks in after all the existing z/OS TCP/IP logic for selection of source and destination IP addresses has been exhausted

Steps for moving to an IPv6 Environment...

- **Selection and placement of IPv6 to IPv4 translators or application gateway**
 - z/OS does not implement any functions that will allow IPv6-only nodes to communicate with z/OS-resident AF_INET applications, so an outboard protocol converter or application-layer gateway component may be needed
 - This component will only be needed if the test configuration includes IPv6-only platforms
 - Various technologies are being made available by various vendors
- **Connectivity to non-local IPv6 locations**
 - Tunneling may be needed between a router connected to the LAN that z/OS is connected to, and a router at another location where IPv6 test equipment is located



Sockets API considerations when moving to AF_INET6

- IPv6 addresses are 128-bit in size as compared to 32 bits for IPv4
 - Data structures which store IP addresses must be modified to handle the larger size
- DNS Resolver library changes
 - New DNS routines - getaddrinfo() and getnameinfo()
- Textual representation of the IP address has changed
 - IPv4 addresses use dotted-decimal format
 - IPv6 addresses use colon-hex notation
- IPv6 has several scopes for IP addresses
 - An address is only unique within its given scope
 - On multi-homed hosts, an IP address alone may be insufficient to select the interface over which to route
- IP addresses should not be assumed to be permanent
 - Long-term use of an address is discouraged due to renumbering
 - Applications should rely on DNS resolvers to obtain the appropriate IP addresses
- sockaddr_in6
 - Analogous to sockaddr_in, but holds 128-bit IPv6 address, port numbers, plus Flow Label and Interface Identifier
- in6_addr
 - Analogous to 32-bit in_addr, but holds 128-bit IPv6 address
- Socket calls to investigate for possible changes
 - socket(), bind(), connect(), sendmsg(), sendto(), accept(), recvfrom(), recvmsg(), getpeername(), getsockname()
- New calls
 - inet_pton(), inet_ntop()

What can you do today? Start learning, planning, and testing!



- ❑ Develop a multi-step plan
 - Eventual goal is fully IPv6-enabled dual-stack operating environment
- ❑ Choose a target date for being IPv6-enabled
 - Work backwards in developing a timeline on when key steps need to be completed
- ❑ Develop detailed plan for each sub-step
 - To resolve critical dependencies in the necessary timeframe
- ❑ Not too early to begin planning today
 - Need for IPv6 may occur quickly and with little advanced warning
 - Rapid realization that IPv6 is needed
 - Take several years to actually get IPv6 deployed
 - Need to have IPv6 already in use (and tested) before it becomes a requirement that it be used operationally
- ❑ Develop an internal addressing plan for distributing/managing IPv6 addresses
 - Determine how IPv6 addresses will be obtained
 - Either from your ISP, or from a Regional Internet Registry (RIR)
 - Consider whether Unique Local Addresses are appropriate
- ❑ Understand your ISPs IPv6 plans
- ❑ Perform a detailed inventory of all systems
 - Determine what is involved in IPv6-enabling them
 - All network hardware and software
 - All client and server hardware, software and applications
- ❑ Develop plans to ensure all components are IPv6-enabled according to a workable timeline
 - Work with vendors to understand their plans for adding IPv6 support for all critical components
- ❑ Determine how end users will use IPv6 services
 - Likely involve tunneling initially
 - But need IPv6-capable routers on the edge links where clients connect
 - Need to provide remote IPv6 access
- ❑ Develop plans for IPv6 training, education and consulting

z/OS and IPv6 – certifications

- z/OS V1R5 is IPv6 Ready Phase 1 certified by the IPv6 Forum
- z/OS V1R8 and z/OS V1R11 are IPv6 Ready Phase 2 certified by the IPv6 Forum
- z/OS V1R10 is IPv6 certified according to the US DoD IPv6 requirements!
 - See the “Special Interoperability Test Certification of the IBM z/OS Version 1.10 Operating System for IBM Mainframe Computer Systems for Internet Protocol Version 6 Capability”
 - From US government, Defense Information Systems Agency, Joint Interoperability Test Command
 - (http://jitc.fhu.disa.mil/adv_ip/register/certs/ibmzosv110_dec08.pdf)

“The IBM z/OS Version 1.10 operating system for IBM mainframe computer systems has met the Internet Protocol (IP) Version 6 (IPv6) Capable interoperability requirements of an Advanced Server as described in the Department of Defense (DoD) Information Technology Standards Registry, “DoD IPv6 Standard Profiles for IPv6 Capable Products Version 2.0,” 1 August 2007, reference (c). The IBM z/OS Version 1.10 operating system for IBM mainframe computer systems has successfully completed the related IPv6 Interoperability portions of the “DoD IPv6 Generic Test Plan (GTP) Version 3,” August 2007, reference (d), and is certified for listing on the Unified Capabilities (UC) Approved Products List (APL) as IPv6 Capable.”

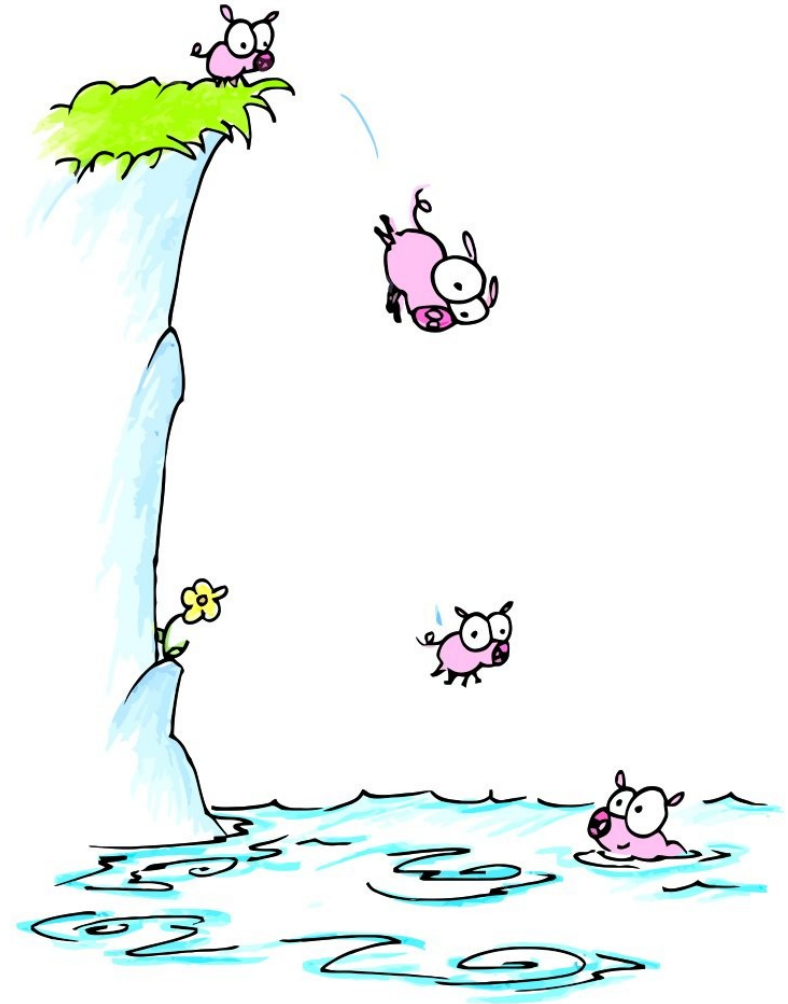
Things to consider ..

z/OS is ready for IPv6 – are you?

Phase-1





Phase-2



The IPv6 pool

For more information



URL		Content
http://www.twitter.com/IBM_Commserver		IBM z/OS Communications Server Twitter Feed
http://www.facebook.com/IBMCommserver		IBM z/OS Communications Server Facebook Page
https://www.ibm.com/developerworks/mydeveloperworks/blogs/IBMCommserver/?lang=en		IBM z/OS Communications Server Blog
http://www.ibm.com/systems/z/		IBM System z in general
http://www.ibm.com/systems/z/hardware/networking/		IBM Mainframe System z networking
http://www.ibm.com/software/network/commserver/		IBM Software Communications Server products
http://www.ibm.com/software/network/commserver/zos/		IBM z/OS Communications Server
http://www.redbooks.ibm.com		ITSO Redbooks
http://www.ibm.com/software/network/commserver/zos/support/		IBM z/OS Communications Server technical Support – including TechNotes from service
http://www.ibm.com/support/techdocs/atmastr.nsf/Web/TechDocs		Technical support documentation from Washington Systems Center (techdocs, flashes, presentations, white papers, etc.)
http://www.rfc-editor.org/rfcsearch.html		Request For Comments (RFC)
http://www.ibm.com/systems/z/os/zos/bkserv/		IBM z/OS Internet library – PDF files of all z/OS manuals including Communications Server
http://www.ibm.com/developerworks/rfe/?PROD_ID=498		RFE Community for z/OS Communications Server
https://www.ibm.com/developerworks/rfe/execute?use_case=tutorials		RFE Community Tutorials

For pleasant reading

Supplemental DNS Configuration

named.conf changes for IPv6

- Additional zone { } statements:
 - New zone statements are required for IPv6 reverse zones.
- Additions to the options { } statement:
 - listen-on-v6 { any; };
 - named can listen on ALL IPv6 interfaces or NONE of them. Hence, the only allowable values for listen-on-v6 { } are 'any;' or 'none;'
- Other (optional) options:
 - transfer-source-v6
 - query-source-v6
 - notify-source-v6
- Other statements not specific to v4 or v6 take either IPv4 or IPv6 addresses as arguments.
 - e.g., the forwarders { } or allow-transfer { } options, or the masters clause in a slave zone statement.

Example IPv6-enabled name server configuration file

```
acl mynets { fec0::/64; fec0:0:0:A::/64;          #IPv6 site-local
              50c9:c2d4::/64; 50c9:c2d4:0:A/64;    #IPv6 global
              9.67.115.0/26; };                   #IPv4 subnet

options {
directory "/etc/dnsdata";
pid-file "/etc/dnsdata/named.pid";
listen-on-v6 { any; };
query-source-v6 address 50c9:c2d4::A:9:67:115:5 port *;
allow-transfer { mynets; };
};

zone "tcp.raleigh.ibm.com" {
type slave;
masters { fec0::9:67:114:45; 9.67.114.45; };
file "db.tcp.slave";
};
zone "A.0.0.0.0.0.0.0.0.0.0.0.0.c.e.f.ip6.int" {
type master;
file "db.ipv6.reverse";
};
```

Forward zone changes for IPv6 DNS

- IPv4 information is stored in A records; IPv6 information is stored in AAAA records.
 - The format is essentially the same, e.g.,
 - www IN A 9.67.115.5
 - www IN AAAA fec0::9:67:115:5
 - The new IPv6 records may coexist with existing IPv4 information (whether one is adding IPv6 records to an existing zone or starting from scratch).
- An alternative record format for IPv6 information is the A6 record.
 - This format is experimental and is not recommended for use
 - The **allow-v6-synthesis { }** named.conf option could be useful if you have to deal with other servers that use A6 records
 - Tells named to query for an A6 first when it receives a AAAA query. If the A6 query fails, then the server tries the original AAAA lookup
 - It takes an address list as an argument--queries from those hosts will invoke this behavior

Example IPv6-enabled forward zone file

```
$TTL 86400
$ORIGIN raleigh.ibm.com.

tcp          SOA      linuxipv63.tcp      dnsadmin.tcp  ( ... )

linuxipv63.tcp  NS      tcp
mvs073.tcp     NS      tcp

;hosts in between snipped

mvs073.tcpA    9.67.115.5
mvs073-v6.tcp  A       9.67.115.5
               AAAA    fec0::A:9:67:115:5      ; site-local VIPA
               AAAA    50c9:c2d4::A:9:67:115:5 ; global VIPA

;other hosts would follow below...
```

Reverse zones for IPv6 DNS

- IPv6 reverse zones work similarly to IPv4 reverse zones, but there are two reverse domains (ip6.int and ip6.arpa) instead of one (in-addr.arpa).
 - In both IPv6 domains, the same 'nibble' label format is used:
 - 5.0.0.0.5.1.1.0.7.6.0.0.9.0.0.0.A.0.0.0.0.0.0.0.0.0.0.0.0.c.e.f.ip6.int.
 - The PTR record is used for the IPv6 reverse mapping, like IPv4:
 - 5.0.0.0.5.1.1.0.7.6.0.0.9.0.0.0.A.0.0.0.0.0.0.0.0.0.0.0.0.c.e.f.ip6.int.
PTR mvs073
 - 5.0.0.0.5.1.1.0.7.6.0.0.9.0.0.0.A.0.0.0.0.0.0.0.0.0.0.0.0.c.e.f.ip6.arpa.
PTR mvs073
- Why two reverse domains?
 - RFC 3152 deprecates the ip6.int domain for IPv6 reverse mapping and says that ALL IPv6 reverse zones should fall under ip6.arpa.
 - Many Resolvers implement IPv6 reverse lookups using the ip6.int domain, but the process of migrating to ip6.arpa is on the way.

Reverse Zones for IPv6 DNS (*continued*)

- Serving two zones containing essentially the same data is more complex, but does not have to be an administrative headache.
 - Carefully done, the same zone data file can be used for both zones!

```
# example named.conf section...
zone "A.0.0.0.0.0.0.0.0.0.0.0.0.c.e.f.ip6.int" {
    type master;
    file "db.ipv6.reverse";
};

zone "A.0.0.0.0.0.0.0.0.0.0.0.0.c.e.f.ip6.arpa" {
    type master;
    file "db.ipv6.reverse";
};
```

- Zone files inherit a default \$ORIGIN value from the name of the zone in named.conf. And since the data in the reverse zones is identical other than the top-level domain, the same file can be used for both zones, and named will append the named.conf \$ORIGIN onto each record.

Example reverse IPv6 zone file

```
$TTL 86400
;The default origin is the name of the zone:
; A.0.0.0.0.0.0.0.0.0.0.0.c.e.f.ip6.int. or ...ip6.arpa.

@ SOA mvs073.tcp.raleigh.ibm.com. hostmaster. ( ... )

@ NS mvs073.tcp.raleigh.ibm.com.
@ NS linuxipv6.tcp.raleigh.ibm.com.

5.0.0.0.5.1.1.0.7.6.0.0.9.0.0.0 PTR mvs073.tcp.raleigh.ibm.com.
7.1.0.0.5.1.1.0.7.6.0.0.9.0.0.0 PTR winipv6.tcp.raleigh.ibm.com.

;other records here...

5.4.0.0.4.1.1.0.7.6.0.0.9.0.0.0 PTR linuxv63.tcp.raleigh.ibm.com.
6.4.0.0.4.1.1.0.7.6.0.0.9.0.0.0 PTR linuxv64.tcp.raleigh.ibm.com.

;other records would continue below...
```