

IPv6 on z/OS

Gus Kassimis – kassimis@us.ibm.com
Adrian Jones – jonesad@us.ibm.com

IBM Software Group
Enterprise Networking Solutions

Tuesday, August 3, 2010 – 11:00AM-12:30 PM



SHARE in Boston

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®
- ▶ DB2 Connect
- ▶ DB2®
- ▶ DRDA®
- ▶ e-business on demand®
- ▶ e-business (logo)
- ▶ e business (logo)®
- ▶ ESCON®
- ▶ FICON®
- ▶ GDDM®
- ▶ HiperSockets
- ▶ HPR Channel Connectivity
- ▶ HyperSwap
- ▶ i5/OS (logo)
- ▶ i5/OS®
- ▶ IBM (logo)®
- ▶ IBM®
- ▶ IMS
- ▶ IP PrintWay
- ▶ IPDS
- ▶ iSeries
- ▶ LANDP®
- ▶ Language Environment®
- ▶ MQSeries®
- ▶ MVS
- ▶ NetView®
- ▶ OMEGAMON®
- ▶ Open Power
- ▶ OpenPower
- ▶ Operating System/2®
- ▶ Operating System/400®
- ▶ OS/2®
- ▶ OS/390®
- ▶ OS/400®
- ▶ Parallel Sysplex®
- ▶ PR/SM
- ▶ pSeries®
- ▶ RACF®
- ▶ Rational Suite®
- ▶ Rational®
- ▶ Redbooks
- ▶ Redbooks (logo)
- ▶ Sysplex Timer®
- ▶ System i5
- ▶ System p5
- ▶ System x
- ▶ System z
- ▶ System z9
- ▶ Tivoli (logo)®
- ▶ Tivoli®
- ▶ VTAM®
- ▶ WebSphere®
- ▶ xSeries®
- ▶ z9
- ▶ zSeries®
- ▶ z/Architecture
- ▶ z/OS®
- ▶ z/VM®
- ▶ z/VSE

- 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.
- Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
- Red Hat is a trademark of Red Hat, Inc.
- SUSE® LINUX Professional 9.2 from Novell®
- Other company, product, or service names may be trademarks or service marks of others.
- This information is for planning purposes only. The information herein is subject to change before the products described become generally available.
- 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.

All performance data contained in this publication was obtained in the specific operating environment and under the conditions described and is presented as an illustration. Performance obtained in other operating environments may vary and customers should conduct their own testing.

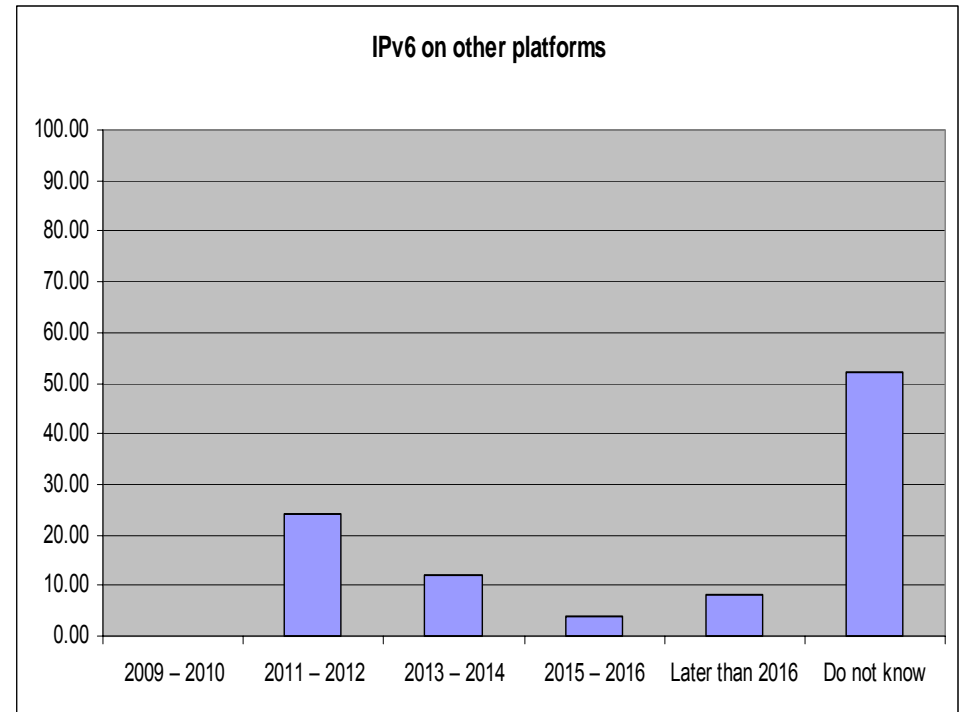
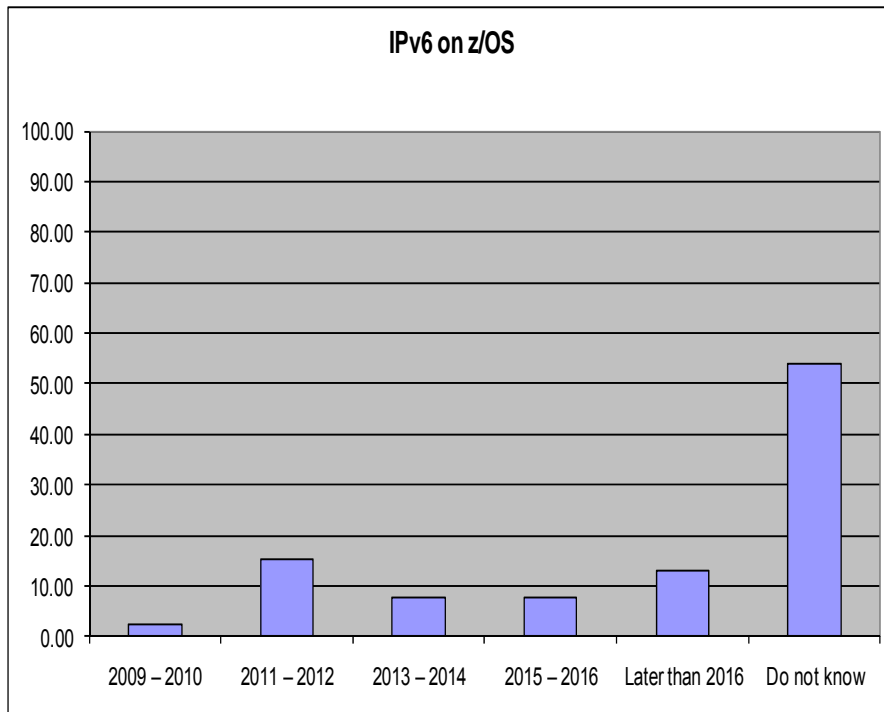
Refer to www.ibm.com/legal/us for further legal information.

Agenda

- Why is IPv6 on z/OS important?
- What IPv6 features are supported by z/OS?
- How do you enable IPv6 support on z/OS?
- How do you configure z/OS CS IPv6 support?
- How do you access z/OS from a remote client?
- How do you verify that IPv6 is working?
- How do you manage IPv6 support on z/OS?
- What are some of the considerations when enabling IPv6 support?
- What are some of the steps to begin the transition to IPv6?

Customer Survey on IPv6

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



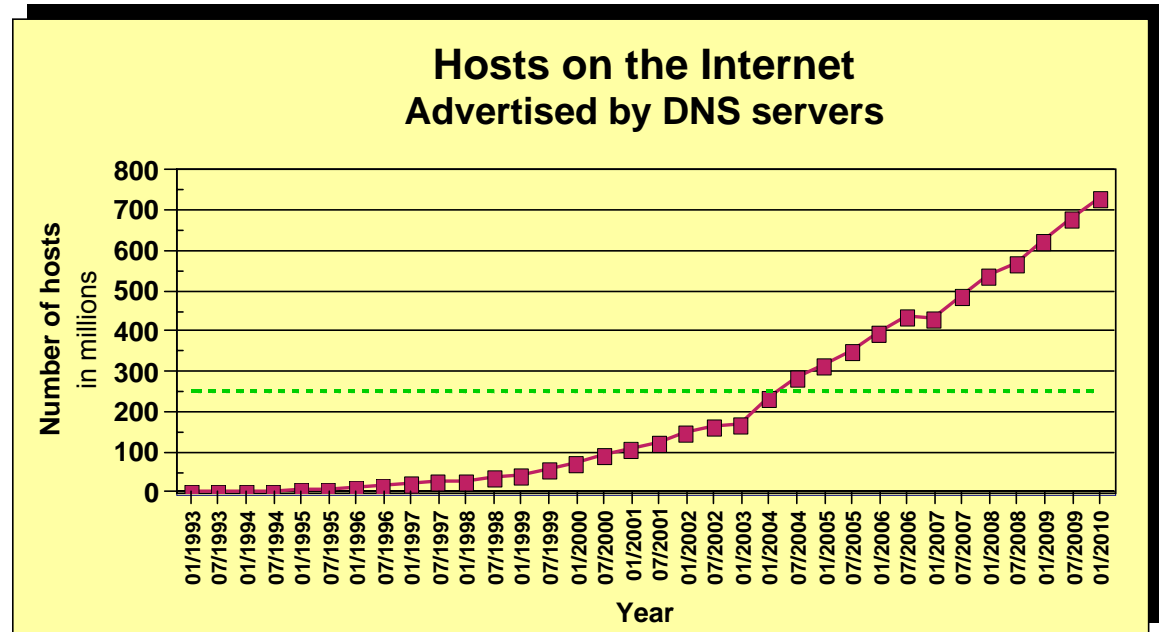
Source: Survey conducted by ENS early 2009 among a selected set of customers (39 responses to this question)

Visible IPv4 hosts growth on the Internet through the past years

- **Projected Internet Assigned Numbers Authority (IANA) Unallocated Address Pool Exhaustion**
 - ▶ July 2011

- **Projected Regional Internet Registries (RIR) Unallocated Address Pool Exhaustion**
 - ▶ January 2012

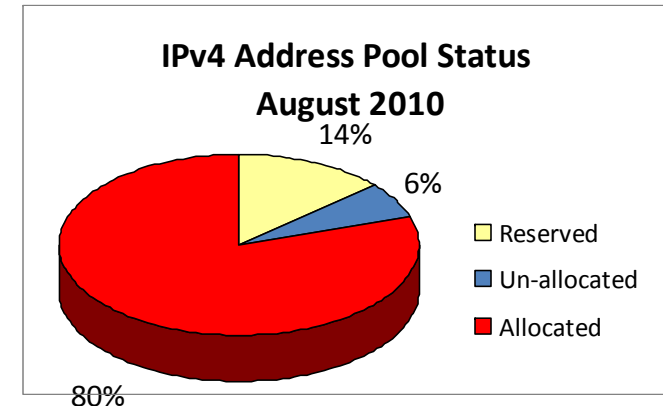
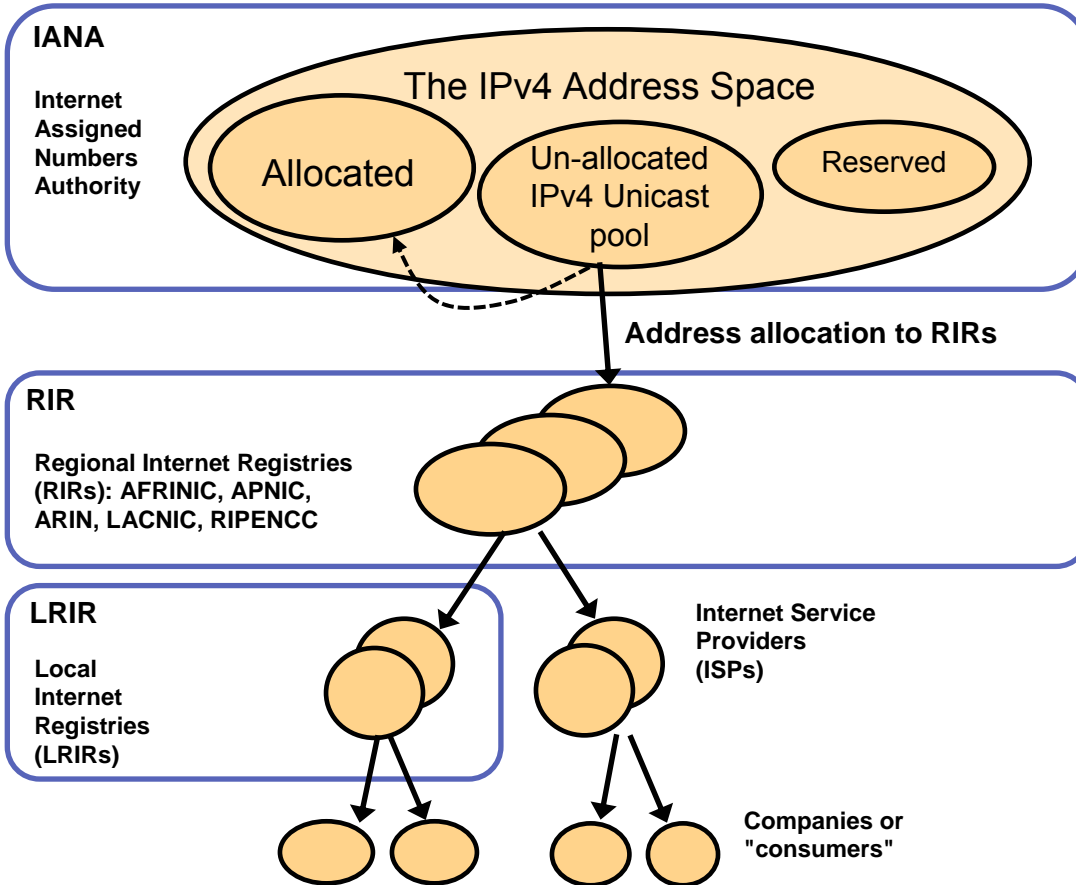
- **z/OS Communications Server continues to focus on IPv6 standards currency**
 - ▶ US DoD/NIST
 - ▶ IPv6 Forum



- ▶ What is the upper practical limit (the ultimate pain threshold) for number of assigned IPv4 addresses? Some predictions said 250 million, others go up to one billion.
- ▶ Source: <https://www.isc.org/solutions/survey>
- ▶ Source: <http://www.potaroo.net/tools/ipv4/index.html>

If you want to stay in business after 2011/2012, you'd better start paying attention!
Do not worry, 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.

How the IPv4 address space is managed



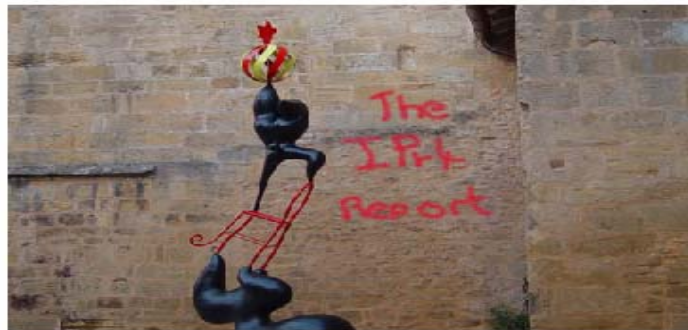
The most current predictions estimate that the un-allocated pool will be exhausted July 2011. Some of the assigned but not advertized space may potentially be re-used to prolong the life of IPv4.

Source: "IPv4 Address Report" - <http://www.potaroo.net/tools/ipv4/>

IPv4 Address Report - Windows Internet Explorer

http://www.potaroo.net/tools/ipv4/index.html

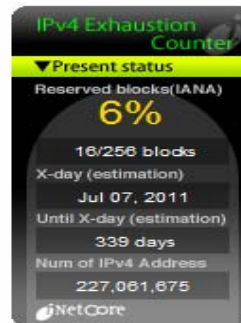
IPv4 Address Report



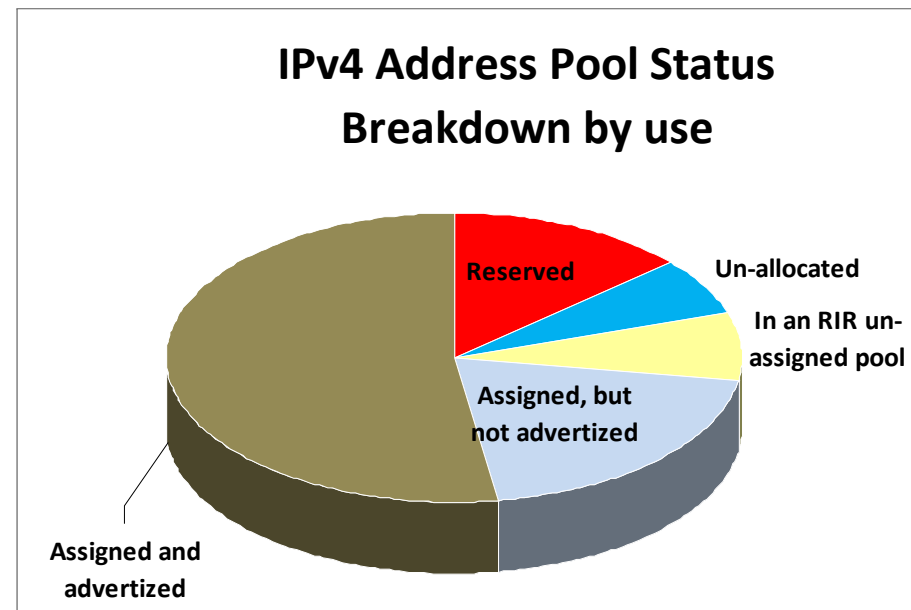
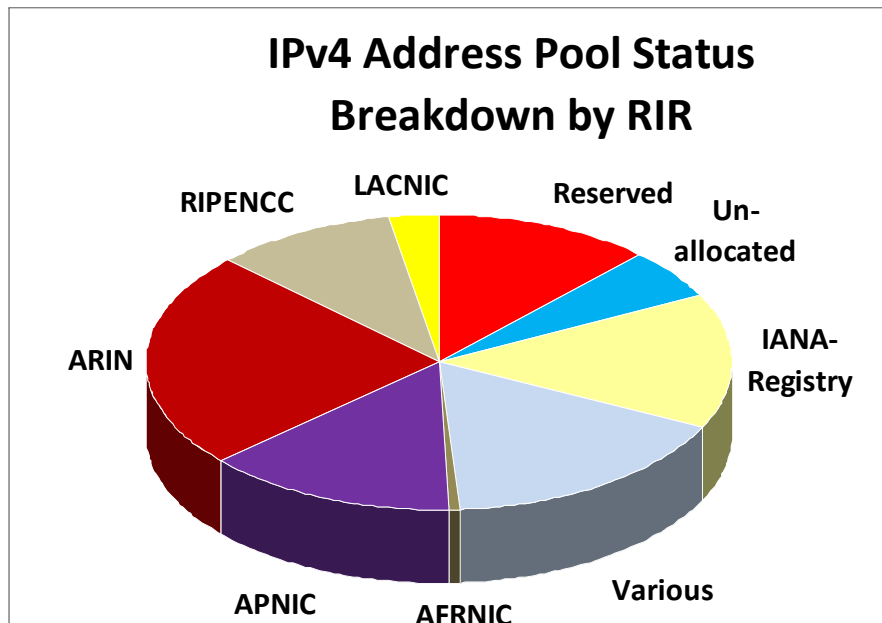
This report is auto-generated by a daily script. The report you are seeing here was generated at 02-Aug-2010 07:58 UTC.

Projected IANA Unallocated Address Pool Exhaustion: 07-Jul-2011

Projected RIR Unallocated Address Pool Exhaustion: 24-Jan-2012



IPv4 address space data as of August 2010...



- Reserved:** Reserved by the IETF
- Un-allocated:** Available to be allocated to the RIRs
- IANA-Registry:** Addresses assigned directly by the IANA from before the time of regional registries
- Various:** Space allocated to various registries (before regional registries were introduced)
- AFRNIC:** Africa, portions of the Indian Ocean
- APNIC:** Portions of Asia, portions of Oceania (includes Australia, China, India)
- ARIN:** Canada, United States, islands in the Caribbean Sea and North Atlantic Ocean
- RIPENCC:** Europe, the Middle East, Central Asia
- LACNIC:** Latin America, portions of the Caribbean

Why IPv6? It's this simple: IPv4 addresses are running short!

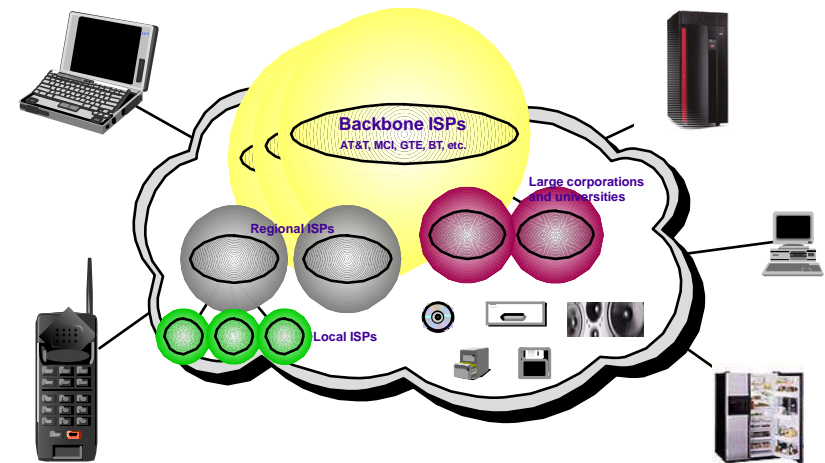
➤ IPv6 deployment is largely inevitable

- ▶ Literally running out of IPv4 addresses
 - IPv4 address pool projected to be exhausted in 1-2 years
 - To minimize disruption, IPv6 needs to be in place and in actual use before exhaustion occurs
- ▶ No other credible alternative to IPv6
 - Only alternative is IPv4 with significant increase in NAT
 - Increased use of private addresses and resulting address collisions
 - Complete loss of globally unique addressing

➤ All major vendors have maturing IPv6 product lines

- ▶ All IBM operating systems support IPv6, with middleware and application support fairly widely available
- ▶ Router vendors (such as Cisco) have supported IPv6 for several years
- ▶ Microsoft VISTA was IPv6-enabled “out of the box”

The Internet - a worldwide digital utility.



Connectivity for **anyone** from **anywhere** (car, plane, home, office) to **anything!**

IPv6 promises true end-to-end connectivity for peer-based collaborative solutions.

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
 - ▶ 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 the whole world and for generations to come
 - ▶ More addresses *cannot* be retrofitted into IPv4
- Other improvements important, but secondary:
 - ▶ Facilities for automatic configuration
 - ▶ Improved support for site renumbering
 - ▶ End to end IP security
 - ▶ Mobility with route optimization (important for wireless)
 - ▶ Miscellaneous minor improvements

IPv4 Address:
9.67.122.66

IPv6 Address:
2001:0DB8:4545:2::09FF:FEF7:62DC

Trends driving IPv6

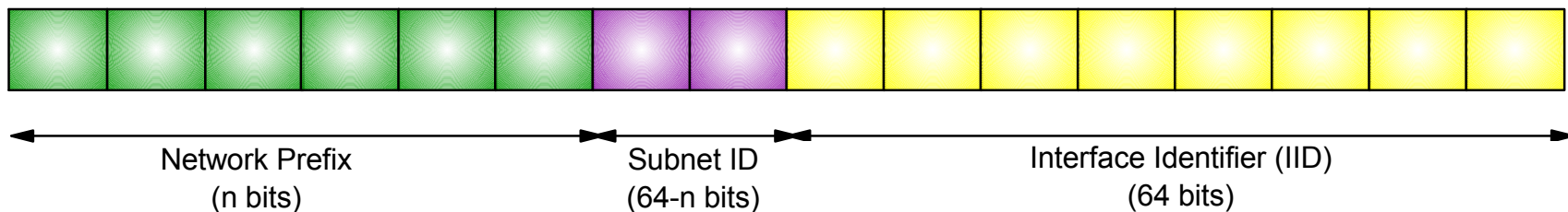
- Growing mobility of users
 - ▶ Internet access from anywhere (car, home, office)
 - ▶ Multiple addresses per person
 - ▶ Pervasive Computing
- Continued rapid growth of the Internet
 - ▶ China plans to roll out ~1 billion Internet nodes, starting with a 320 million student educational network
 - Network operations for 2008 Summer Olympics done solely on IPv6 network
 - ▶ Asia/Pacific, and to a lesser extent Europe, missed out on the early IPv4 address allocations
- Government support
 - ▶ Wide-scale IPv6 promotion underway in China, Japan, Korea and Taiwan
 - ▶ European Commission (EC) encourages IPv6 research, education, and adoption in member countries
 - ▶ US Department of Defense - All platforms offered to DoD must meet very specific IPv6 capabilities
 - ▶ Other US government institutions through the National Institute of Standards and Technologies - NIST has also published detailed IPv6 compliance requirements
- More and more "push" applications being deployed in the wireless market space.
 - ▶ Clients subscribe to services that get pushed out by servers – requires public addresses for clients
- Convergence of voice, video and data on IP
 - ▶ Need for reliable and scalable architecture
 - ▶ "Always-on Connections"

Important IPv6 technical features

- IPv6 header and extensions header
 - ▶ Streamlined IPv6 header
 - ▶ Optional extensions for fragmentation, security, etc.
- Routers no longer fragment forwarded datagrams
- Extended IP Address
 - ▶ 32 bits -> 128 bits (but only 64 bits for routing)
- Neighbor Discovery and Stateless Autoconfiguration
 - ▶ Router Discovery and Neighbor Unreachability Detection (NUD)
 - ▶ Address configuration with no manual or server-based configuration
- IPv4/IPv6 Coexistence and Transition Mechanisms
 - ▶ Coexistence for IPv4 and IPv6
 - ▶ Tunneling and transition mechanisms

Expanded routing and addressing

- Expanded size of IP address space
 - ▶ Address space increased to 128 bits
 - Provides 340,282,366,920,938,463,374,607,431,768,211,456 addresses
 - Enough for 1.8×10^{19} addresses per person on the planet
 - ▶ A 64-bit subnet prefix identifies the link
 - ▶ Followed by a 64-bit Interface Identifier (IID)
- IID derived from IEEE identifier (i.e., MAC address)
 - ▶ Only leftmost 64 bits available for routing and "network addressing"
 - ▶ The rightmost 64-bits identify the host on the target link



IPv6 address textual representation

- Addresses are represented as 8 segments of 4 hex digits (16 bits), separated by colons

2001:0DB8:0:0:240:2BFF:FE3D:71AD

- Two colons in a row can be used to denote one or more sets of zeroes, usually used between the prefix and the interface ID

2001:0DB8::240:2BFF:FE3D:71AD

- The prefix length can be indicated after a slash at the end

2001:0DB8::240:2BFF:FE3D:71AD/64

- A prefix alone is represented as if the interface ID bits are all zero

2001:0DB8::/64

- Obviously, this syntax may be a bit difficult for humans.....

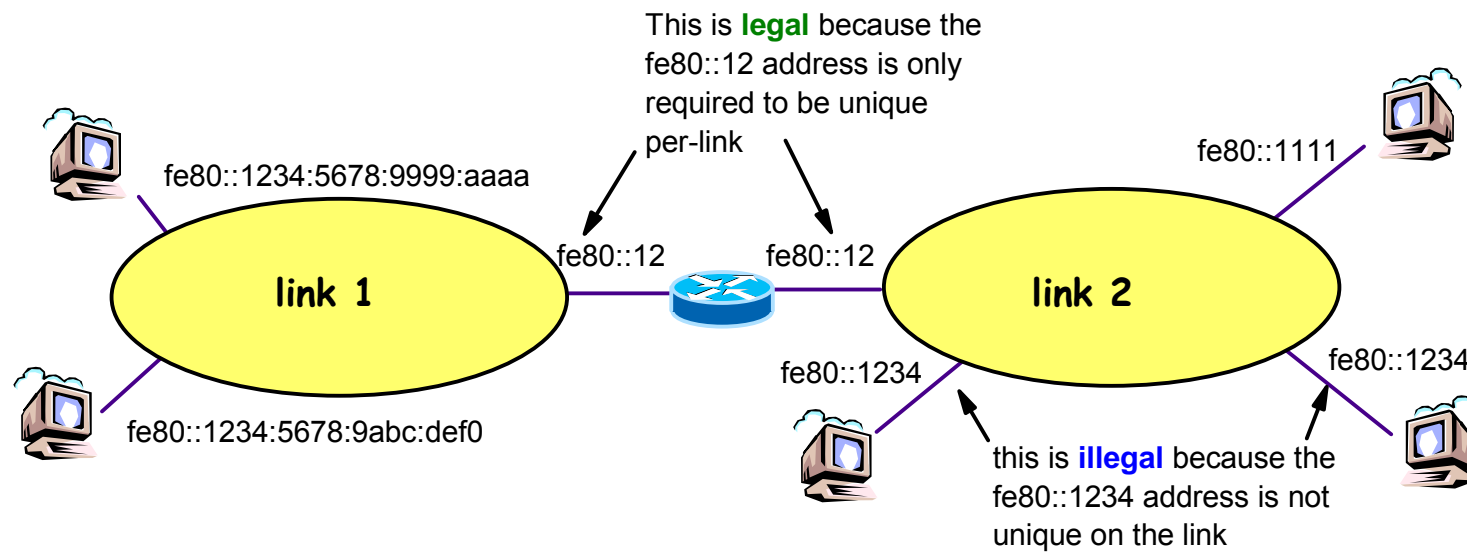
Use of DNS/hostnames no longer optional

Common IPv6 addresses and prefixes

- `::/128` – INADDR6_ANY (the unspecified address)
- `::1/128` – IPv6 loopback address
- `FF00::/8` – Multicast addresses
- `FE80::/10` – Link-local addresses
- `FC00::/7` – Unique local addresses
- `::FFFF/96` - IPv4-Mapped IPv6 Address
- Anything else – Globally unique IPv6 address

IPv6 scoped unicast addressing

- Concept of scoped unicast addresses part of architecture
- Link-local addresses for use on a single link
 - ▶ Primarily used for bootstrapping and infrastructure protocols such as Neighbor Discovery
 - ▶ Address = well-known link-local prefix plus node-generated IID
- Unique Local IPv6 Unicast addresses for use within a site
 - ▶ Like net 10 (not routable in the Internet backbone)
 - ▶ Site-local addresses
 - Part of early IPv6 standards -but introduced a lot of complexity
 - Has been deprecated by the IETF
- Global address prefixes are provided by ISPs



Neighbor Discovery

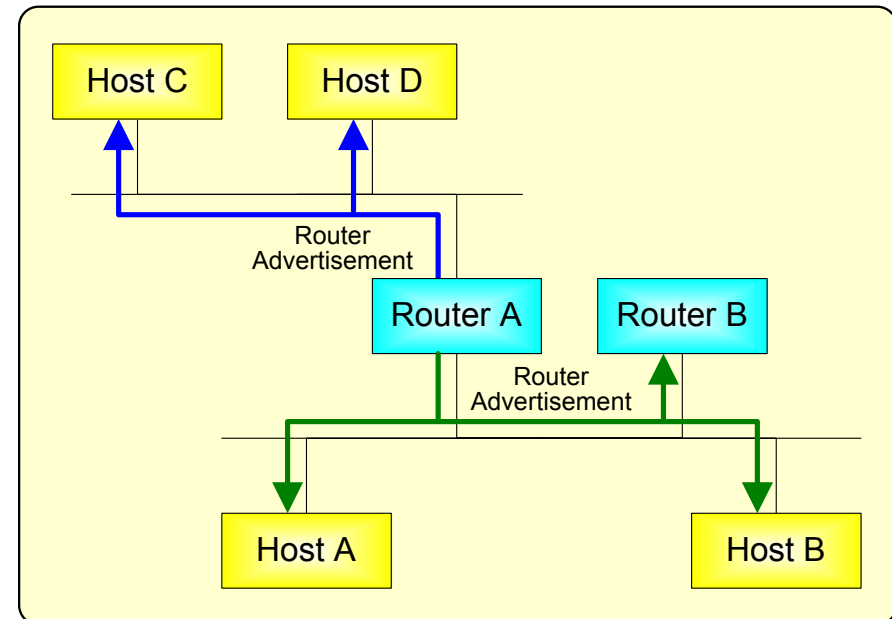
- Router Discovery
 - ▶ Router Solicitations and Router Advertisements used to find and keep track of neighboring routers
 - ▶ Includes additional information for IP stack configuration

- Address resolution
 - ▶ Neighbor Solicitations and Neighbor Advertisements perform address resolution (i.e., ARP functions)

- Neighbor Unreachability Detection (NUD)
 - ▶ Keep track of reachability of neighbors
 - ▶ If path to router fails, switch to another router before TCP timeouts

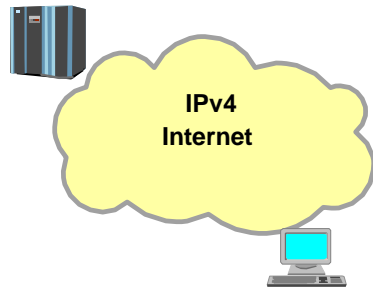
Stateless Address Autoconfiguration

- Address Configuration without separate DHCP server
 - ▶ Router is the server, advertising key address configuration information
- Address formed by combining routing prefix with Interface ID
- Link-local address configured when an interface is enabled
 - ▶ Allows immediate communication with devices on the local link
 - ▶ Primarily used for bootstrapping and discovery
 - ▶ Well-known prefix combined with locally-generated 64-bit IID
- Other addresses configured via Routing Advertisements
 - ▶ RA advertises 64-bit prefixes (e.g., on-link, form an address)
 - ▶ Public (e.g., server) addresses formed from Interface ID
- Duplicate Address Detection
 - ▶ Ensures uniqueness of configured IP address

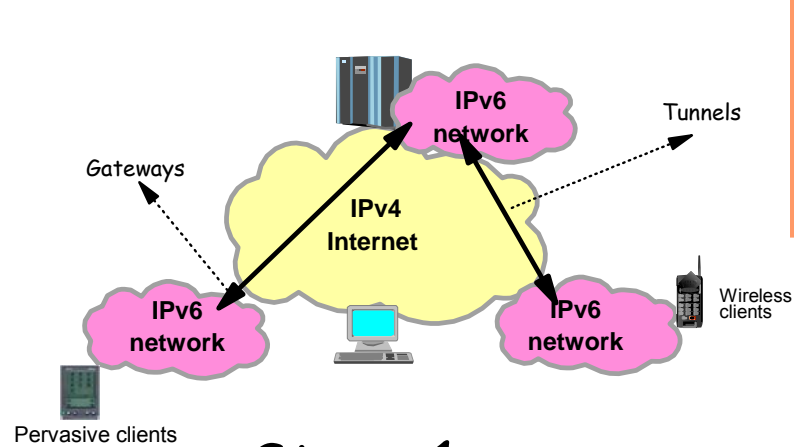


IPv4 to IPv6 Internet evolution

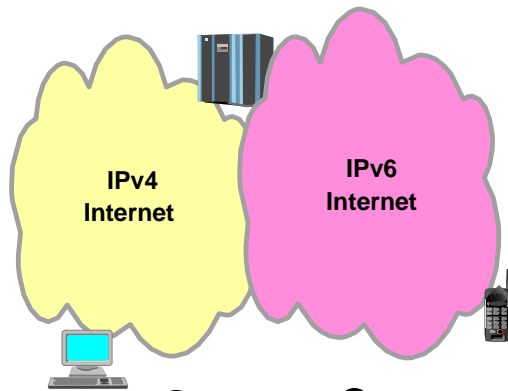
So where are we? In stage 1, preparing to move to stage 2.



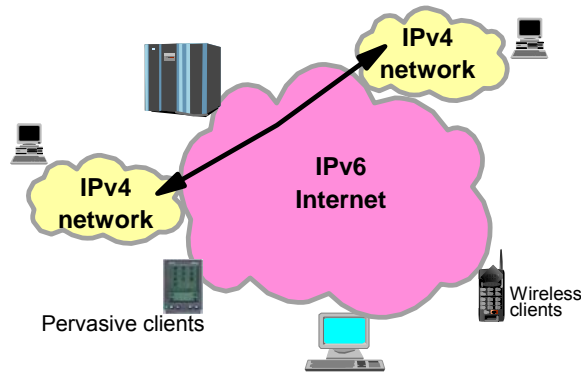
Yesterday



Stage 1



Stage 2



Stage 3

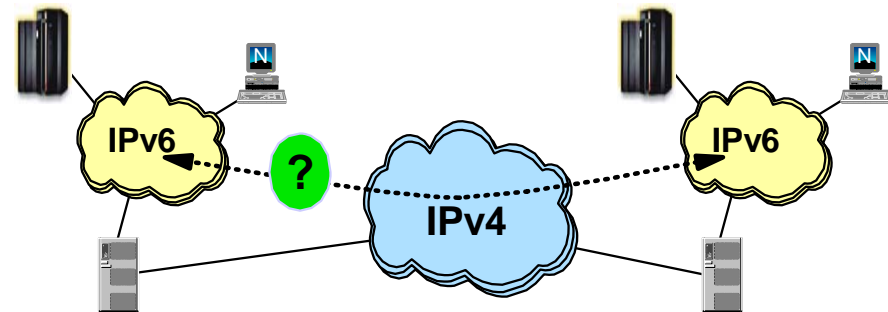
There may be a stage 4 with only IPv6, but it will take some years to get there.

General transition considerations

1

How do we share the physical network so that both IPv4 and IPv6 can be transported over one and the same physical network?

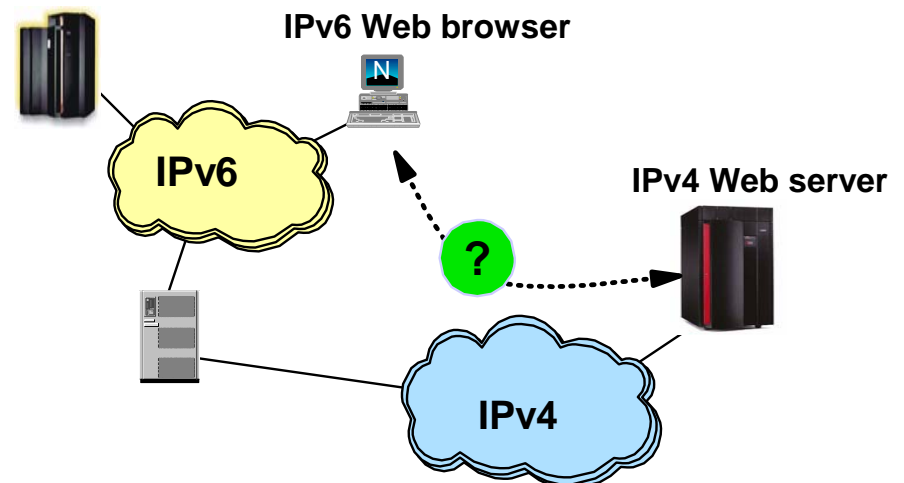
- Dual-stack
- Tunneling of IPv6 over IPv4



2

How do applications that have not yet been enhanced to support IPv6 communicate with applications that have been enhanced to support IPv6?

- Dual-stack
- Application Layer Gateways (ALG)
- Network Address Translation – Protocol Translation (NAT-PT)

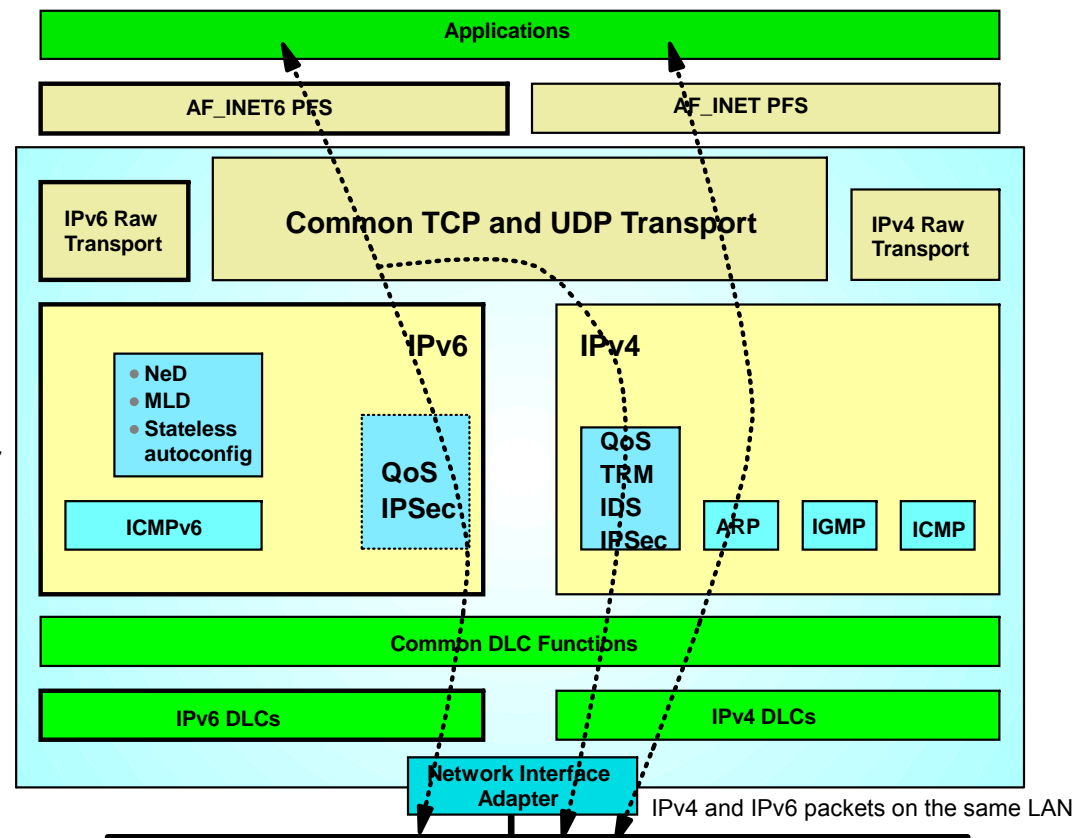


Isn't IPv6 enablement just a network engineering exercise?

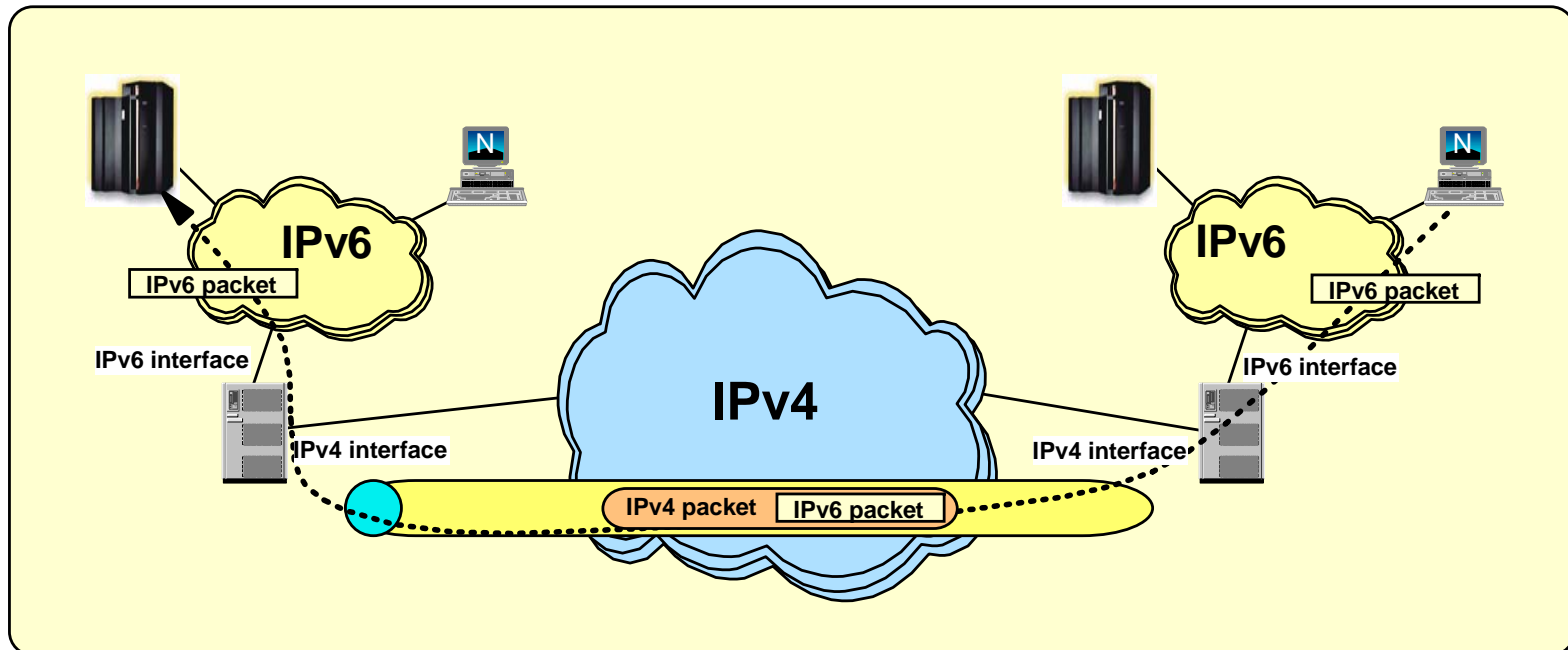
- Wish so !!
- A few facts:
 - ▶ The network infrastructure will have to be updated to support IPv6 network infrastructure functions, such as neighbor discovery (an auto-addressing technology), IPv6 routing tables (OSPFv3), ICMPv6, name servers with IPv4 and IPv6 addresses, DHCP servers for IPv6, etc.
 - Layer-3 routers
 - Firewalls
 - Intrusion Detection devices
 - ▶ The physical media you use today can carry both IPv4 and IPv6 – so no new cabling (!)
 - ▶ A TCP/IP stack must be updated to support IPv6 – alongside with IPv4 (known as dual-mode TCP/IP stack)
 - ▶ IPv6 requires a new sockets interface, known as AF_INET6 (Addressing Family IPv6)
 - IPv4 sockets programs today use AF_INET, which is IPv4 only. An AF_INET sockets program can communicate with an IPv4 sockets partner only
 - Sockets programs that are updated to support AF_INET6 can communicate with both IPv4 and IPv6 sockets partners
 -
- Sockets programs must be updated to talk IPv6 !!

z/OS TCP/IP is a dual-mode 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.

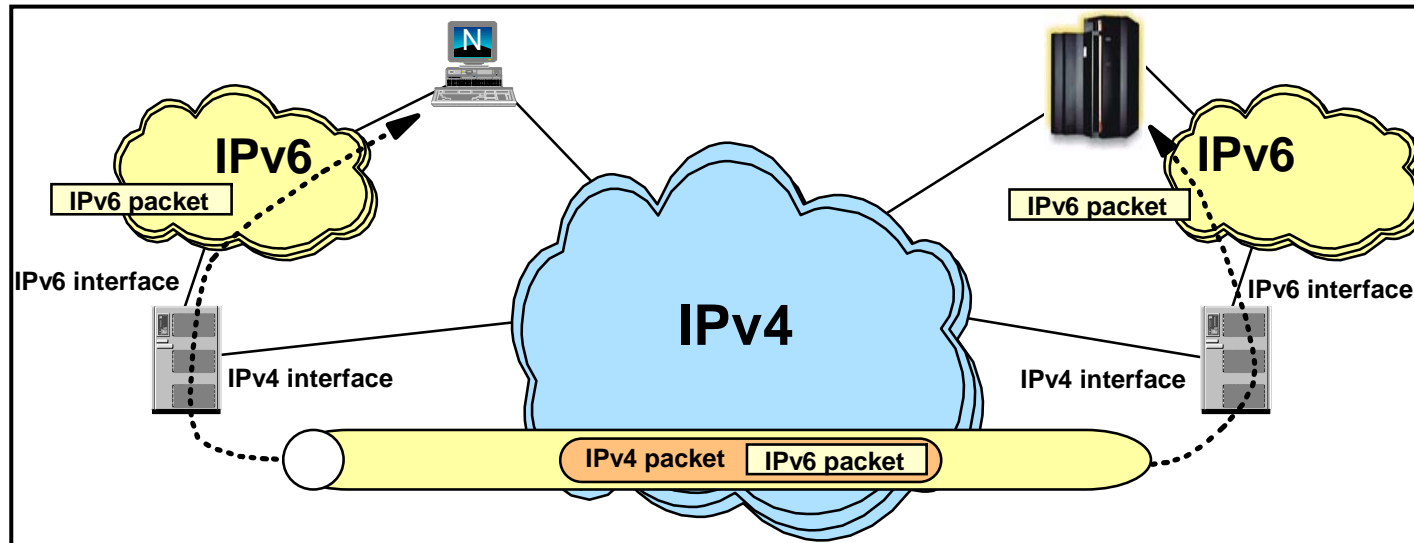


Tunneling overview



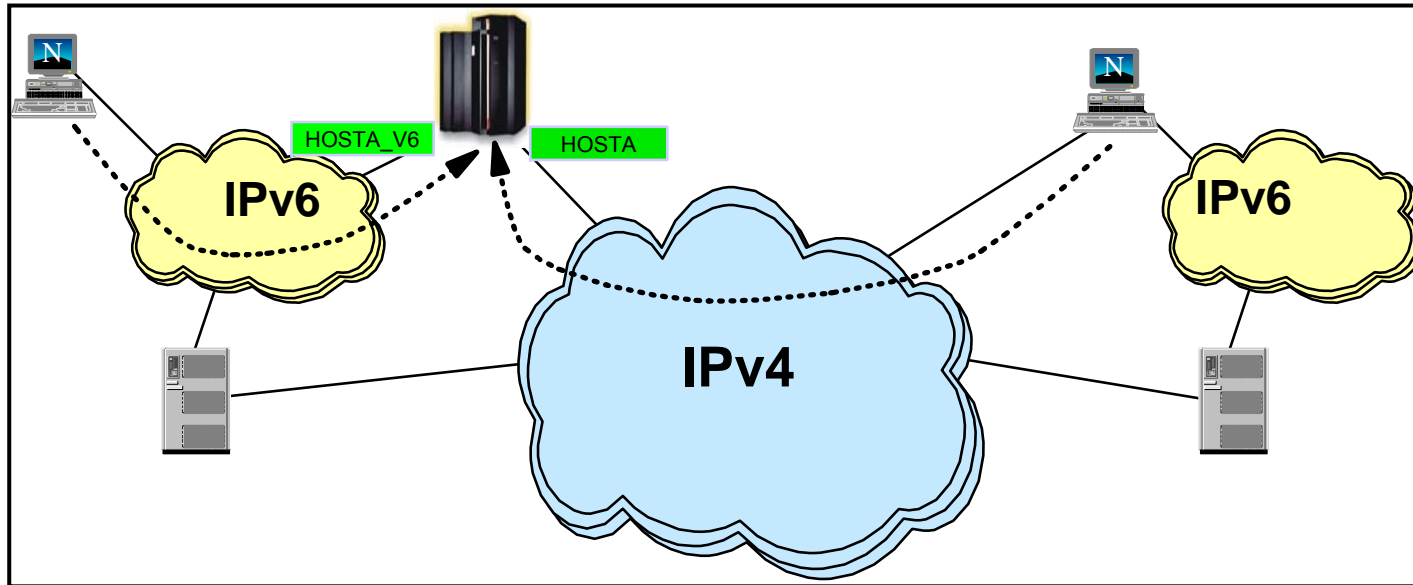
- Tunneling: encapsulating an IPv6 packet in an IPv4 packet and send the IPv4 packet to the other tunnel endpoint IPv4 address.
- Requires applications on both endpoints to use AF_INET6 sockets
- Tunnels endpoints can be in hosts or routers
 - ▶ The tunnel endpoint may be an intermediate node, the final endpoint, or a mixture of the two
- The tunnel endpoint placement depends on connectivity needs
 - ▶ Placing endpoints in routers allows entire sites to be connected over an IPv4 network
 - ▶ Placing endpoints in hosts allows access to remote IPv6 networks without requiring updates to the routing infrastructure

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

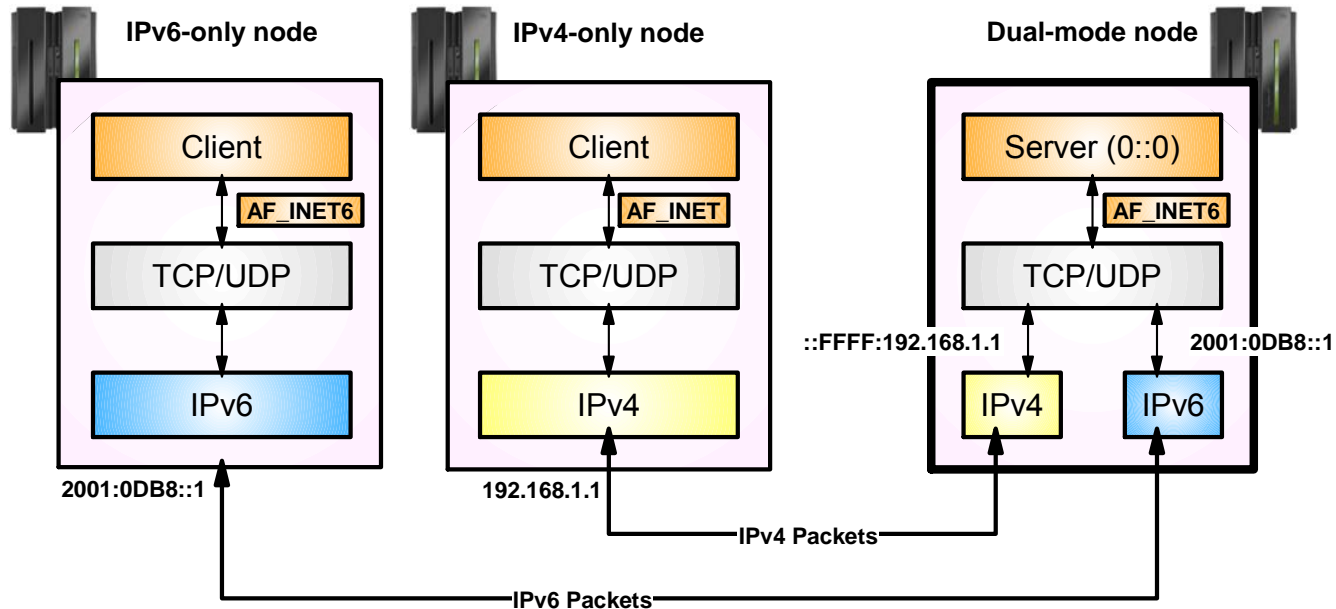
Use of distinct IPv4 and IPv6 host names



- 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

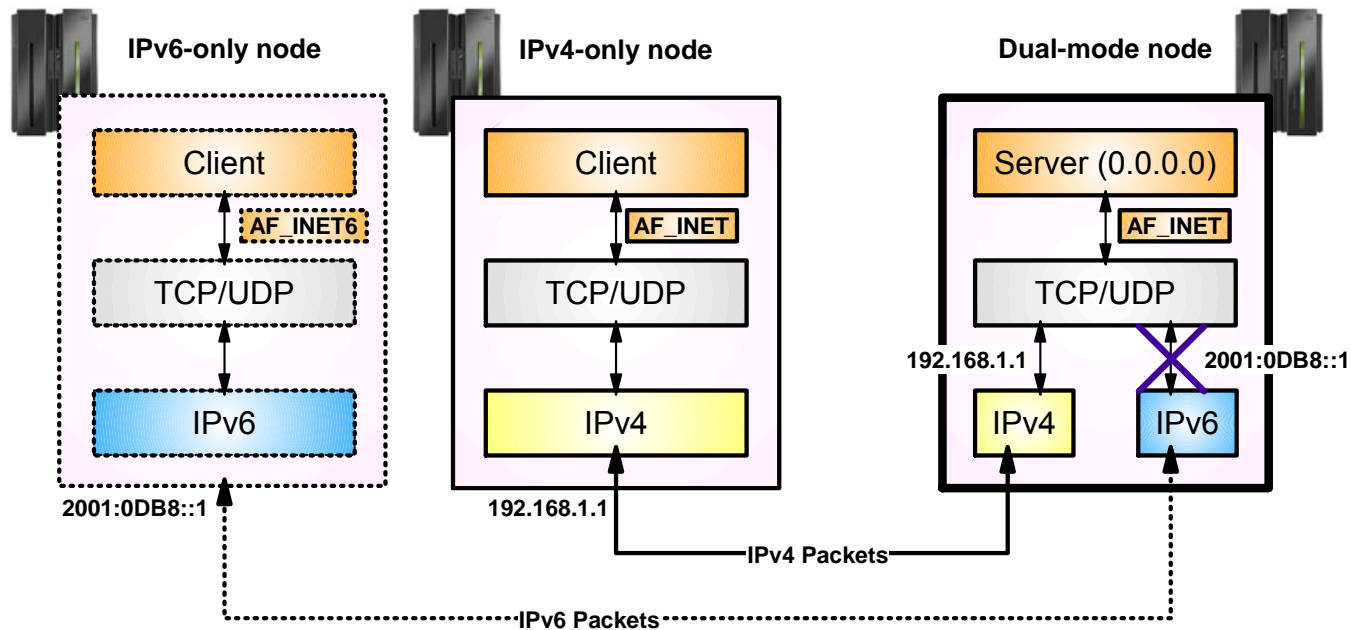
Note: Use of distinct host names is only necessary during the initial transition phases when native IPv6 connectivity does not exist

IPv6-enabled application on a dual mode stack



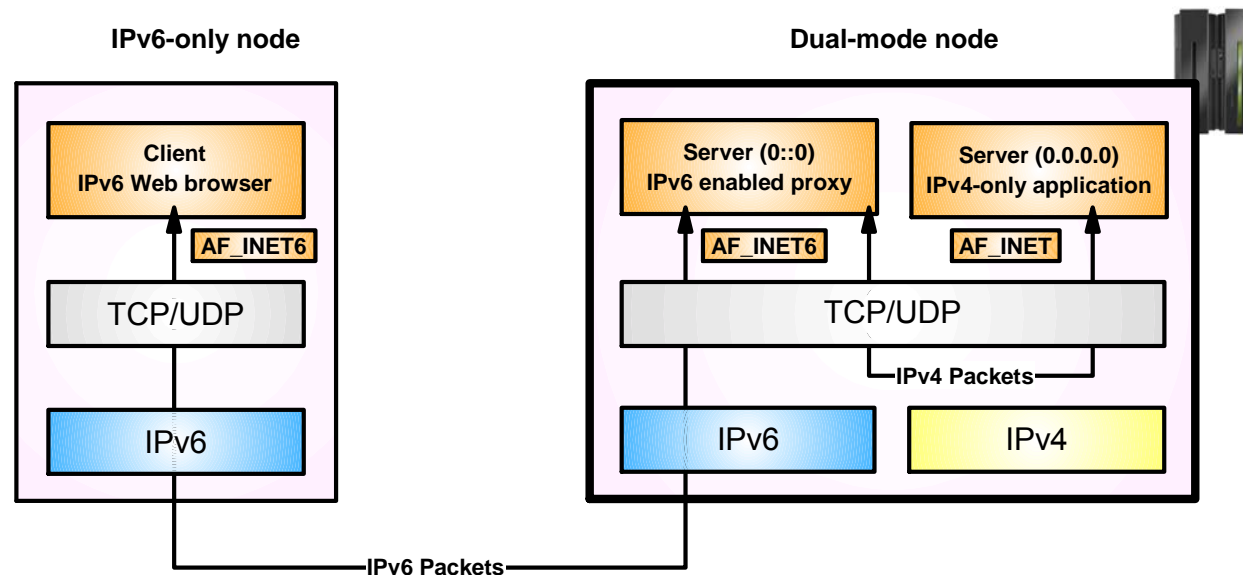
- An IPv6-enabled application can communicate over both IPv4 and IPv6 peers
 - ▶ A single socket can be used to send or receive traffic from either IPv4 or IPv6 partners
 - ▶ IPv4 packets to the IPv4 partner and IPv6 packets to the IPv6 partner
 - ▶ No changes need to be made to the partner application
- An IPv6-enabled application uses AF_INET6 sockets for both IPv4 and IPv6 partners
 - ▶ An IPv4 address is mapped to IPv6 addresses by the Transport Layer in the TCP/IP stack
 - ▶ Uses a special address format which identifies the IPv6 address as an IPv4-mapped IPv6 address
 - ▶ For example, 192.168.1.1 would be represented as ::FFFF:192.168.1.1

IPv4-only application on a dual-mode stack



- An IPv4 application running on a dual-mode stack can communicate with an IPv4 partner.
 - ▶ The source and destination addresses will be native IPv4 addresses
 - ▶ The packet which is sent will be an IPv4 packet
- If partner is IPv6 running on an IPv6 only stack, then communication fails
 - ▶ If partner was on dual-mode stack, then it would fit in previous page discussion
 - ▶ The partner only has a native IPv6 address, not an IPv4-mapped IPv6 address
 - ▶ The native IPv6 address for the partner cannot be converted into a form the AF_INET application will understand

Accessing IPv4-only applications through an IPv6 application layer gateway



- An IPv6-only client can access IPv4-only servers via an IPv6 “proxy”
 - ▶ The IPv6 proxy communicates with the IPv6-only client using IPv6, and accesses the IPv4-only server using IPv4
 - ▶ The IPv4-only server may be on the same node as the IPv6 proxy, or may reside on a different node
 - ▶ The use of a backend IPv4-only server is, in most cases, completely transparent to the IPv6 client

z/OS and IPv6 - certifications

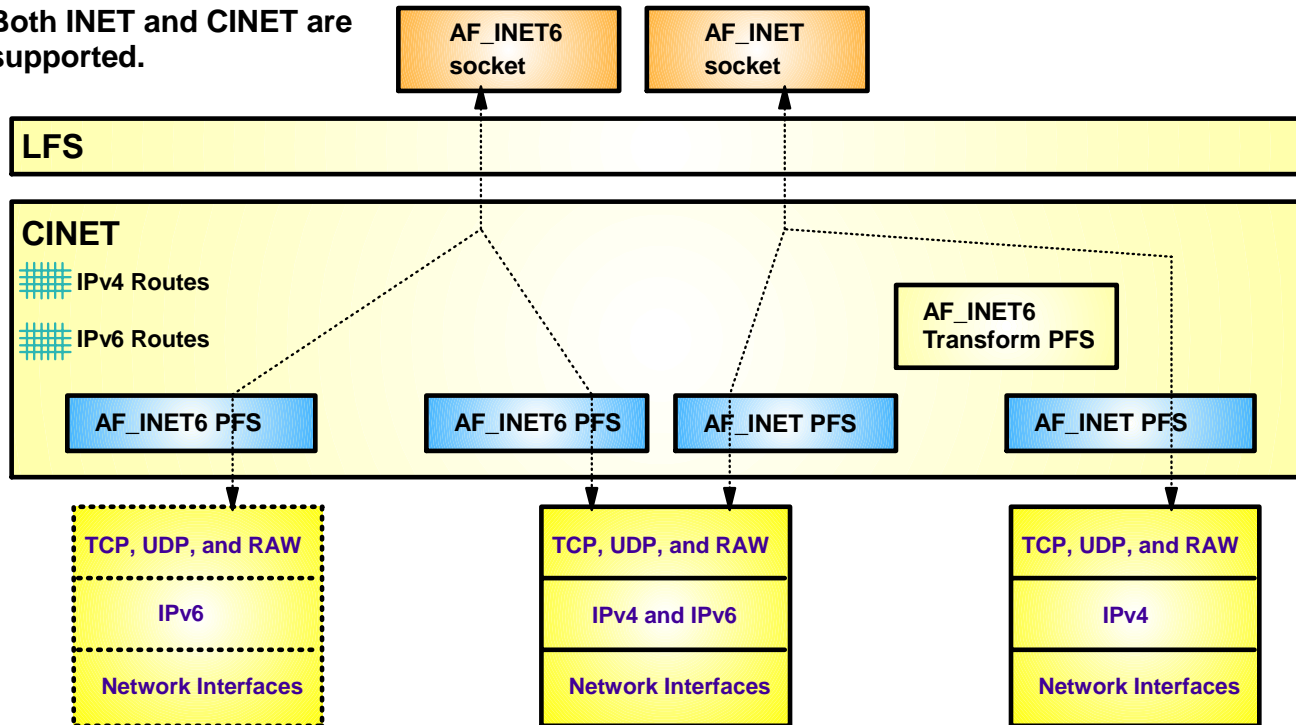
- z/OS V1R5 is IPv6 Ready Phase 1 certified by the IPv6 Forum
- z/OS V1R8 is IPv6 Ready Phase 2 certified by the IPv6 Forum
- z/OS V1R10 is IPv6 certified according to the US DoD IPv6 requirements!
 - ▶ See the “Special Interoperability Test Certification of the IBM z/OS Version 1.10 Operating System for IBM Mainframe Computer Systems for Internet Protocol Version 6 Capability”
 - ▶ From US government, Defense Information Systems Agency, Joint Interoperability Test Command

“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.”

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
A z/OS TCP/IP stack will always come up as an IPv4-only stack if AF_INET6 is not enabled in BPXPRMxx

- ▶ 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.

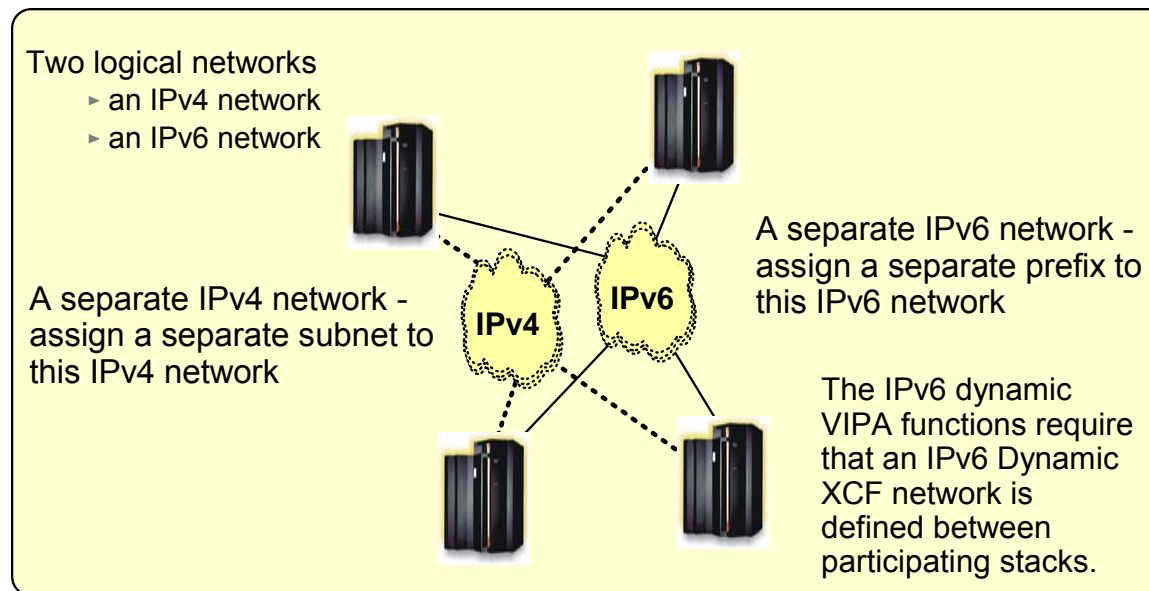
Configuring IPv6 support

- 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
```

Connecting to an IPv6 network

- Uses separate logical networks - one IPv4 and one IPv6
 - ▶ The same physical adapter and network infrastructure can be used for both, though
- IPv6 DLC support in z/OS
 - ▶ Fast Ethernet, 1GbE and 10GbE and using OSA Express in QDIO Mode
 - ▶ HiperSockets
 - ▶ IUTSAMEHOST to other stacks in same LPAR
 - ▶ XCF to other stacks in same Sysplex
 - Both static and dynamic XCF
 - ▶ ESCON (MPCPTP) to another z/OS image (not to any known Channel-attached Routers)



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 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 OSAQDIO15 DEFINE IPAQENET6 PORTNAME OSAQDIO1  
  
INTERFACE OSAQDIO25 DEFINE IPAQENET6 PORTNAME OSAQDIO2  
IPADDR FC00::9:67:115:5  
        2001:0DB8::9:67:115:5
```

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 OSAQDIO16 DEFINE IPAQENET6 PORTNAME OSAQDIO1
SOURCEVIPAIN VIPAV61

INTERFACE OSAQDIO26 DEFINE IPAQENET6 PORTNAME OSAQDIO2
SOURCEVIPAIN VIPAV62

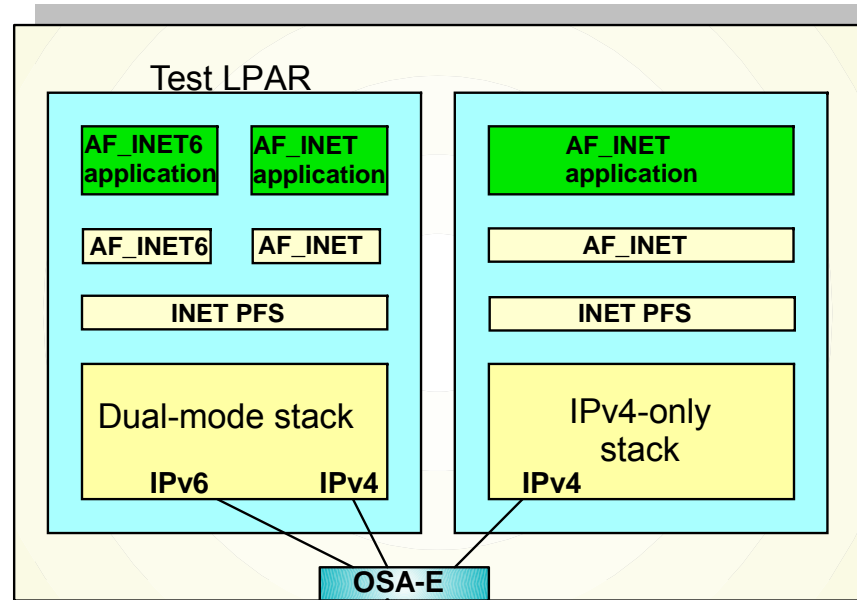
INTERFACE OSAQDIO36 DEFINE IPAQENET6 PORTNAME OSAQDIO3
```

Accessing z/OS from a remote site

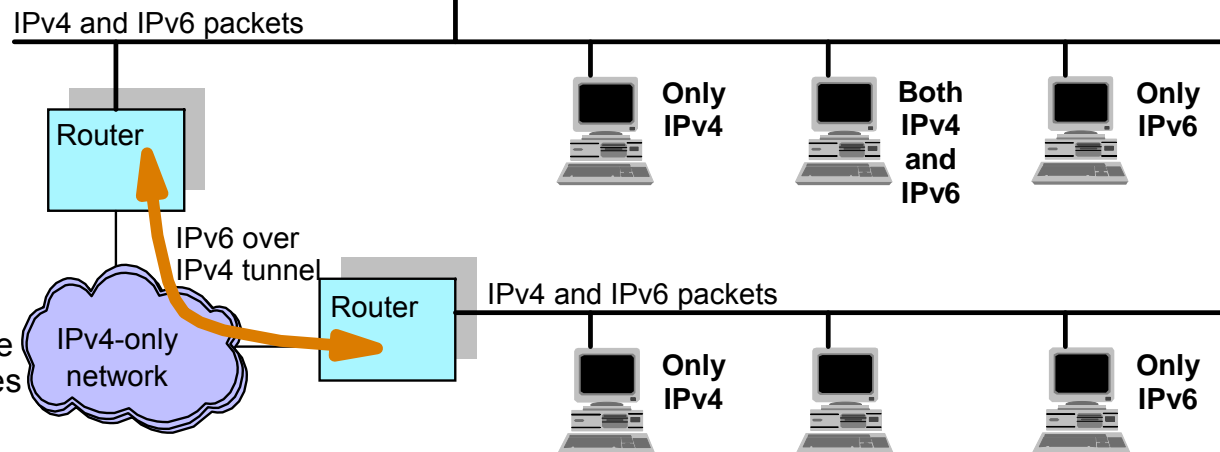
Remember: You can enable IPv6 today on z/OS without impact to your existing IPv4 users.

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

All IPv4 communication continues to work as before.



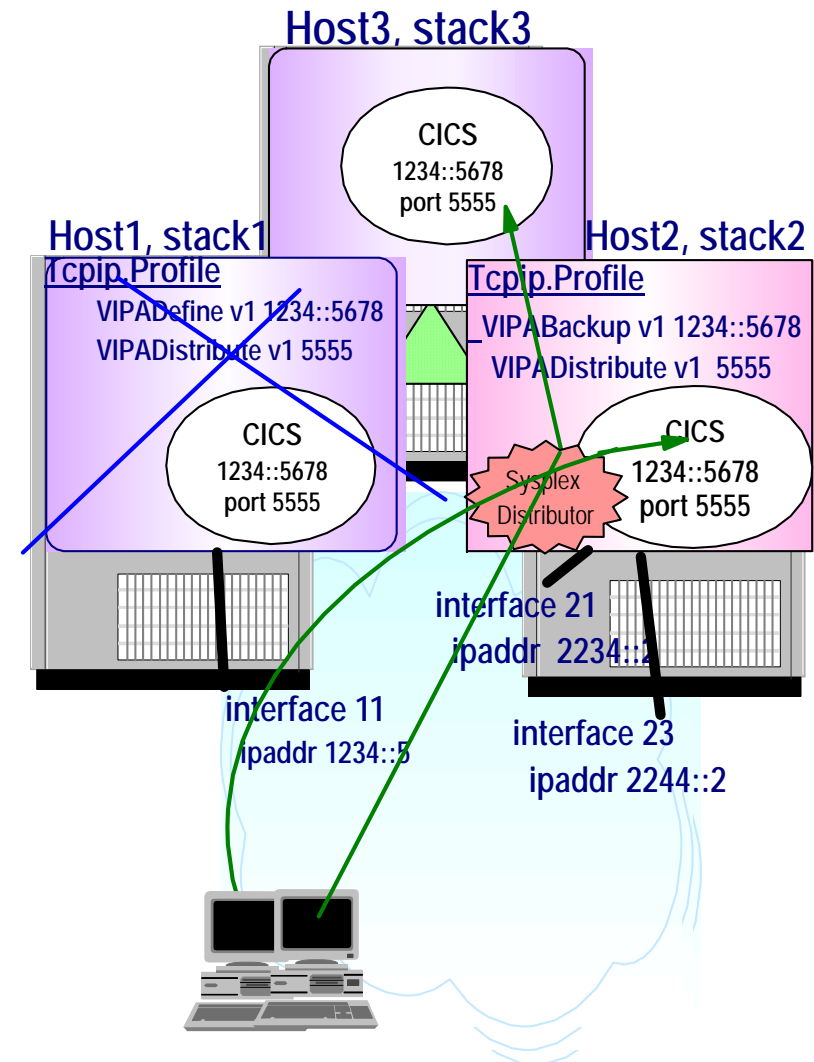
z/OS CS does not support being a tunnel endpoint, although it can route traffic through an intermediate tunnel



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

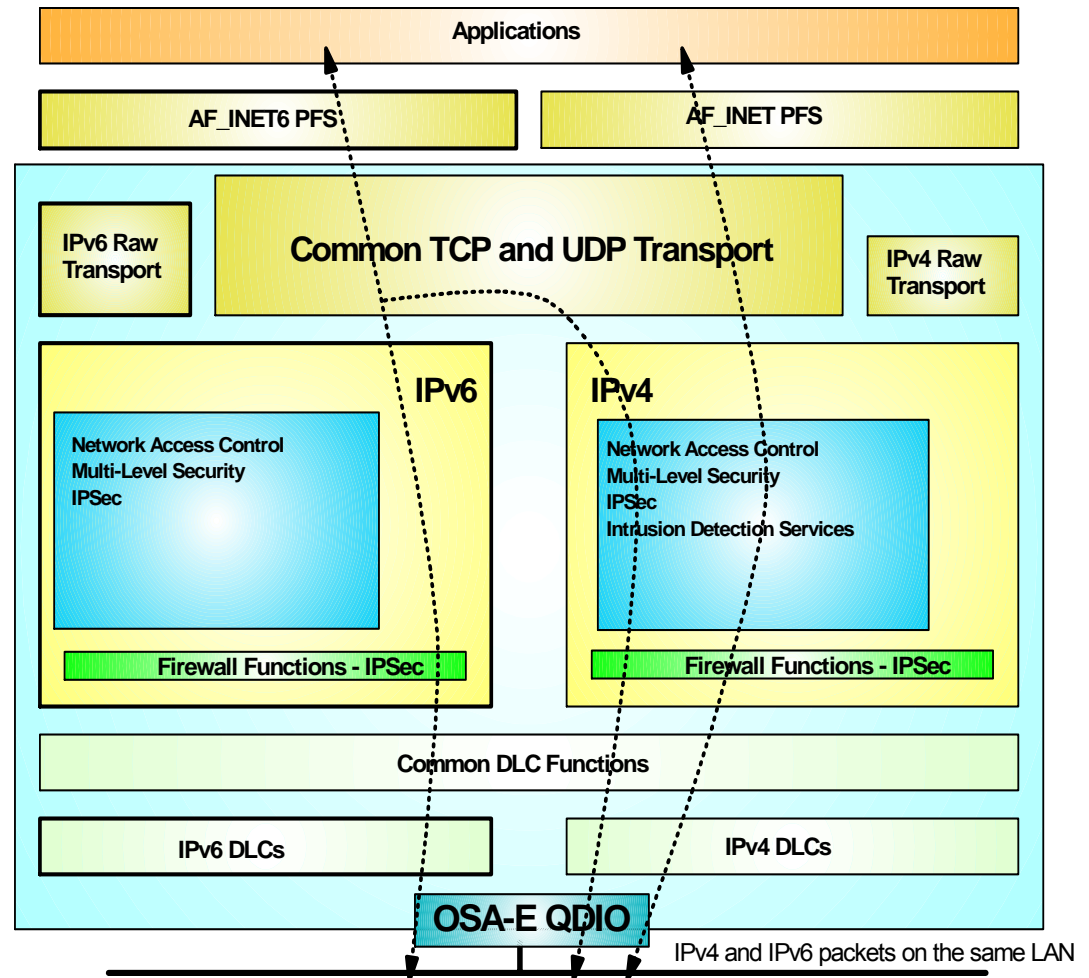
Sysplex functions that support IPv6

- 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
- A few Sysplex functions are not enabled for IPv6
 - ▶ Sysplex Wide Security Associations (SWSA)
 - ▶ Multi Node Load Balancing (MNLB)
 - Cisco does not support IPv6 for MNLB



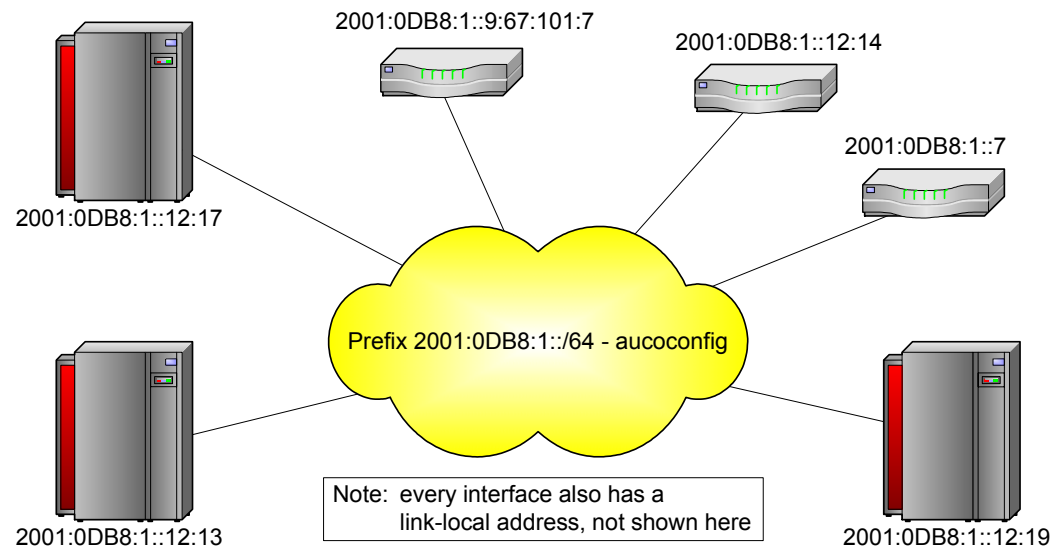
Securing your IPv6 network

- z/OS V1R5 provides the first set of security features for the IPv6 transport
 - ▶ Includes support for both Network Access Control and Multi-Level Security
- Z/OS V1R8 provides IPsec and Firewall filters support for both IPv4 and IPv6
 - ▶ IPv6 IPsec currently does not support Sysplex Wide Security Associations (SWSA)
 - ▶ IPv6 IPsec does not support NAT traversal for IPv6 IPsec
 - The IPv6 technical community approach is that NAT should not be used with IPv6
- IDS continues to be supported only for the IPv4 transport



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
 - It could be used for network protocols other than IPv6, although it isn't today
- Needed for Sysplex-related functions such as dynamic IPv6 VIPA movement



DNS in three easy steps

1. Add and modify statements in the nameserver configuration file
 - ▶ New reverse zone statements
 - ▶ IPv6-specific options (optional)
 - ▶ IPv6 information for options which can take IPv4 or IPv6 addresses (optional)
2. Add IPv6 records to forward zones with hosts that are now IPv6-capable
 - ▶ IPv6 address records: AAAA
3. Create new IPv6 reverse zone files
 - ▶ IPv6 reverse domains: ip6.arpa and ip6.int (*ip6.int has been deprecated – do not use*)
 - ▶ Use the same PTR records from IPv4, with a similar label format

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 Resolver sends queries to DNS server using IPv4
 - ▶ The IPv4 protocol is used to communicate, and does not affect what type of records are returned
 - You can still resolve host names to IPv6 addresses and vice-versa
 - ▶ **IPv6 communications can be used by the resolver starting in z/OS V1R12**
 - Prior to z/OS V1r12, you can use a local DNS name server (caching only or authoritative) if there is no IPv4 network connectivity, as the DNS name server is able to send queries via IPv4 or IPv6
- 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

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
 - The behavior is determined by the invoking application
 - ▶ 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
 - **Note:** 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
 - New statements introduced in the TCP/IP Profile in z/OS V1R12
- May want to consider using a local host file for early testing
 - ▶ using the `LOOKUP LOCAL|DNS` resolver directive

Updating the local host file - Useful for early testing

- Many platforms (z/OS, Solaris, Linux, ...) 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
 - ▶ Other platforms may use a different file for this purpose
- /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)

```
9.67.43.100      NAMESERVER
9.67.43.126      RALEIGH
9.67.43.222      HOSTNAME1       HOSTNAME1_IPV4
129.34.128.245   YORKTOWN        WATSON
1::2             HOSTNAME1       HOSTNAME1_IPV6
1:2:3:4:5:6:7:8 HOSTNAME2_IPV6
```

z/OS Communications Server Applications enabled for IPv6

- 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

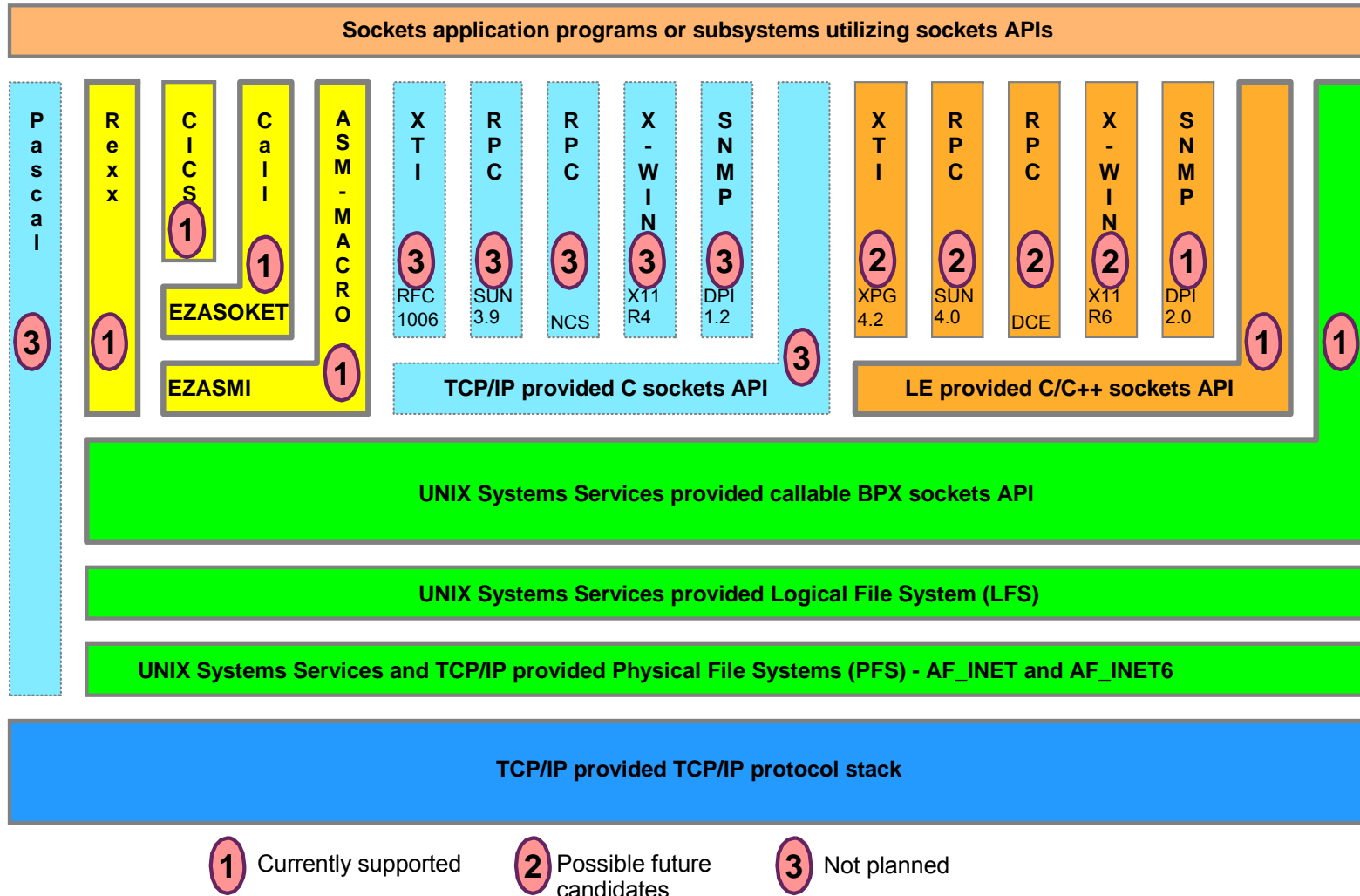
- To enable IPv6 support in the TN3270 server, activate IPv6 stack support
 - ▶ No new configuration commands are provided or needed to enable IPv6 support

- 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

Sockets-related AF_INET6 enablement



Sockets API considerations when moving to AF_INET6 (Notes)

- 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 calls replace `gethostbyname()` and `gethostbyaddr()`
 - `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 multihomed hosts, an IP address alone may be insufficient to select the interface over which to route
 - True for link-local addresses
 - Most applications will not care about this, but it is possible that some may
- 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 cache the appropriate IP addresses
- `sockaddr_in6`
 - ▶ Analogous to `sockaddr_in`, but larger
 - ▶ Holds 128-bit IPv6 address, port numbers, plus Flow Label and Interface Identifier
- `in6_addr`
 - ▶ Analogous to 32-bit `in_addr`
 - ▶ Holds a 128 bit address
- Socket calls to investigate for possible changes
 - ▶ `socket()`, `bind()`, `connect()`, `sendmsg()`, `sendto()`, `accept()`, `recvfrom()`, `recvmsg()`, `getpeername()`, `getsockname()`
- New calls
 - ▶ `inet_pton()`, `inet_ntop()`

Managing your IPv6 network

- Network management SNMP support
 - ▶ Support SNMP agent (OSNMPD)
 - ▶ 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 (as many as we can squeeze into z/OS releases!)
 - 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

Netstat impacts

- 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 IP 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!

Netstat HOME/-h

```
MVS TCP/IP NETSTAT CS V1R10      TCPIP Name: TCPCS      15:49:35
Home address list:
LinkName:  OSAQDIOLINK
  Address:  9.67.115.5
  Flags:    Primary
LinkName:  LOOPBACK
  Address:  127.0.0.1
  Flags:
IntfName:  VIPAV6
  Address:  2001::a:9:67:115:5
  Type:    Global
  Flags:
  Address:  50c9:c2d4:0:a:9:67:115:5
  Type:    Global
  Flags:   Deprecated
IntfName:  OSAQDIO46
  Address:  2001::9:67:115:5
  Type:    Global
  Flags:
  Address:  fe80::6:2900:20dc:217c
  Type:    Link_Local
  Flags:   Autoconfigured
IntfName:  LOOPBACK6
  Address:  ::1
  Type:    Loopback
  Flags:
Unavailable IPv6 Home addresses:
IntfName:  OSAQDIO26
  Address:  2001::9:67:115:66
  Type:    Global
  Reason:  Duplicate address detection pending start of interface
IntfName:  OSAQDIO66
  Address:  2001::/64
  Type:    Global
  Reason:  Interface ID not yet known
```

Netstat DEVLINKS/-d

```

MVS TCP/IP onetstat CS V1R10          TCPIP Name: TCPCS          12:55:20
DevName: OSAQDIO4                    DevType: MPCIPA
DevStatus: Ready
LnkName: OSAQDIOLINK                  LnkType: IPAQENET   LnkStatus: Ready
  NetNum: 0   QueSize: 0   Speed: 0000000100
  IpBroadcastCapability: No
  CfgRouter: Non                      ActRouter: Non
  ArpOffload: Yes                      ArpOffloadInfo: Yes
  ActMtu: 1492
  VLANid: 1260                         VLANpriority: Enabled
  ReadStorage: GLOBAL (8064K)         InbPerf: Balanced
  ChecksumOffload: Yes
BSD Routing Parameters:
  MTU Size: 00000                      Metric: 00
  DestAddr: 0.0.0.0                    SubnetMask: 255.255.255.192
Multicast Specific:
  Multicast Capability: Yes
  Group                               RefCnt
  -----
  224.0.0.1                           0000000001
Link Statistics:
  BytesIn                               = 11476
  Inbound Packets                       = 10
  Inbound Packets In Error               = 0
  Inbound Packets Discarded              = 0
  Inbound Packets With No Protocol       = 0
  BytesOut                               = 6707
  Outbound Packets                       = 10
  Outbound Packets In Error              = 0
  Outbound Packets Discarded             = 0

```

Netstat DEVLINKS/-d (continued)

```

IntfName: OSAQDIO46      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

```

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

Steps for moving to an IPv6 Environment

1. 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)

2. IPv6 address selection

- ▶ Obtain an address block from your ISP
- ▶ For test purposes, local-unicast IPv6 addresses is sufficient (but avoid using them in production)
- ▶ IPv6 addresses can be manually configured on the INTERFACE statement in the TCP/IP Profile or autoconfigured using Neighbor Discovery Stateless Autoconfiguration
 - VIPA addresses must be manually configured

Steps for moving to an IPv6 Environment

3. DNS setup

- ▶ 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 autoconfiguration to define IPv6 addresses, static VIPA addresses should be stored in DNS since the autoconfigured addresses will change over time

4. INET or Common INET

- ▶ Both are supported for IPv6, but INET is much simpler
- ▶ Running IPv4 and dual-mode stacks under CINET is not recommended - run dual-mode stacks in a separate LPAR from IPv4 only stacks
- ▶ AF_INET6 NETWORK statement must be coded in BPXPRMxx before starting IPv6-enabled stacks

Steps for moving to an IPv6 Environment



5. Selection and placement of IPv6 to IPv4 protocol converter 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

6. 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

What can you do today? Start 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
 - ▶ 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
- ❑ **Understand your ISPs IPv6 plans**
- ❑ **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
- ❑ **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
- ❑ **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
- ❑ **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**

For more information

URL		Content
http://www.twitter.com/IBM_Commserver		IBM Communications Server Twitter Feed
http://www.facebook.com/IBMCommserver		IBM Communications Server Facebook Fan Page
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.ibm.com/software/network/commserver/z_lin/		IBM Communications Server for Linux on System z
http://www.ibm.com/software/network/ccl/		IBM Communication Controller for Linux on System z
http://www.ibm.com/software/network/commserver/library/		IBM Communications Server library
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

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.0.0.0.0.0.0.0.0.0.0.0.0.0.0.c.e.f.ip6.int" {
type master;
file "db.ipv6.reverse";
};

zone "A.0.0.0.0.0.0.0.0.0.0.0.0.0.0.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.0.0.0.0.0.0.0.0.0.0.0.0.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...
```