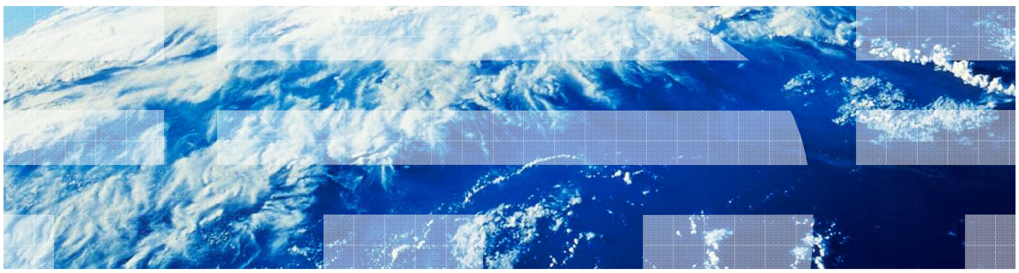
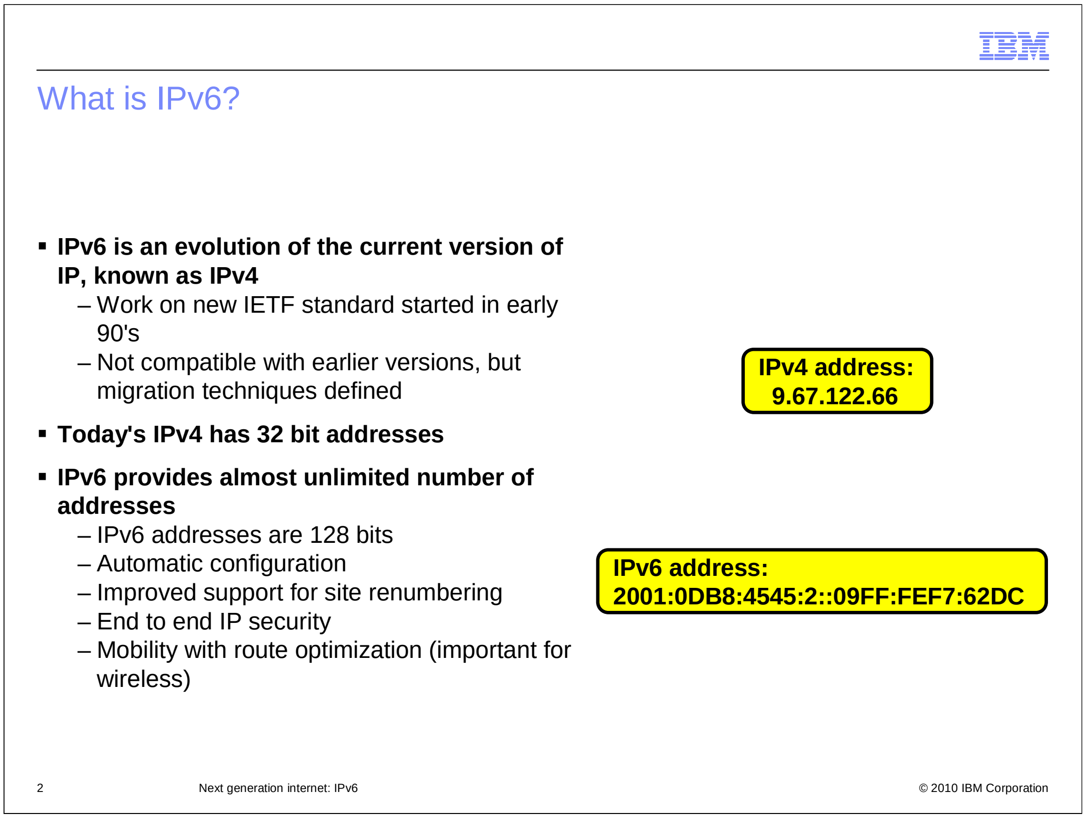
z/OS Communications Server

Next generation internet: IPv6



This presentation describes how to migrate to internet protocol version six.



The internet protocol, or IP, is the basic building block on which all internet applications are built. IP provides the mechanism by which individual data packets are sent from computer to computer, over any mixture of networks links, routers and operating systems.

Web, email, instant messaging, remote database access, voice over IP – none of these can exist without the underlying IP service and its universal addressing system.

Today's internet runs on IPv4. IPv4 uses 32-bit addresses, which in theory allow over four billion unique addresses. In practice, the usable number is less than one billion. This limited address space will eventually become exhausted, possibly as soon as 2011.

IPv6 is the latest version of the IP standard which is intended to progressively replace IPv4. IPv6 uses 128-bit addresses, allowing for enough addresses to meet the anticipated needs for the foreseeable future. To give some perspective on how many addresses this is, IPv6 supports 35 trillion interconnected networks, each the size and complexity as those used by large companies such as IBM.

While the immediate benefit provided by IPv6 is the expanded address space, IPv6 also contains additional capabilities. IPv6 allows for automatic configuration of hosts in the network, using both DHCP and a new stateless auto-configuration protocol. The enhanced auto-configuration capabilities provided by IPv6 also allow for more seamless site renumbering. IPv6 provides end-to-end security with an adequate number of addresses to make this feasible. IPv6 has improved support for mobile clients. While some benefits provided by IPv6 can be retrofitted to IPv4, the lack of universal addressing in IPv4 means these solutions are cumbersome.

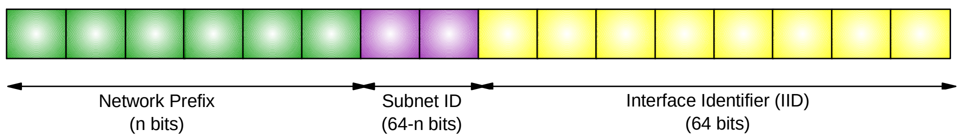
Despite these important changes, IPv6 is a conservative design. IPv6 does not change the fundamental approach to the IP routing infrastructure, DNS naming, firewall protection, or intrusion detection.

IBM

Important IPv6 technical features

- **IPv6 header and extensions header**
- **Routers no longer fragment forwarded datagrams**
- **Neighbor Discovery and Stateless Autoconfiguration**
- **IPv4/IPv6 Coexistence and Transition Mechanisms**

- **Extended IP Address**
  - Address space increased to 128 bits
    - 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 (for example, MAC address)
    - Only leftmost 64 bits available for routing and "network addressing"
    - The rightmost 64-bits identify the host on the target link



IPv6 provides many important technical improvements beyond those found in IPv4. The IPv6 header is now a fixed sized, with each option appearing in its own extension header which is daisy-chained behind the IPv6 header. Expensive, slow-path operations, such as fragmentation, have been removed from the network and instead occur only at the endpoints. Most host configuration is now automated, allowing for improved plug-and-play capabilities. IPv6 also provides many transition and coexistence mechanisms to ease the migration from IPv4 to IPv6.

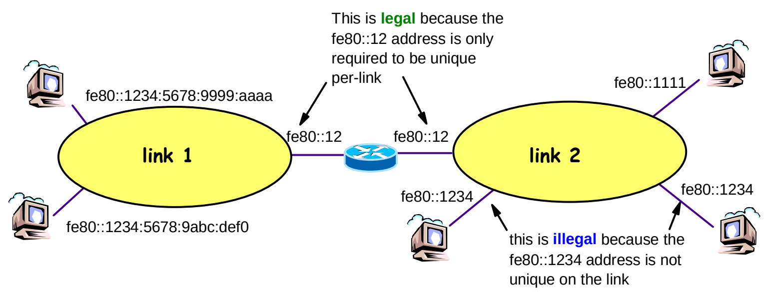
A unicast IPv6 address consists of two parts: the subnetwork prefix and the interface identifier, each of which is 64-bits in size. The subnetwork prefix is used to identify a specific link in the network, while the interface ID is used to identify a specific network interface adapter on that link.

The subnetwork prefix is further divided into two pieces: a network prefix and a subnet ID. The network prefix is used to identify a specific network which is connected to the internet, while the subnet ID is used to identify a specific link within that network. Normally, the network prefix is 48 bits in size and the subnet ID is 16 bits in size, allowing for up to 64K links in a single network. If this isn't sufficient, an enterprise might request adjacent blocks of 48-bit prefixes from their ISP. This effectively increases the size of the subnet ID to 17, 18, 19 bits, or whatever size is needed to subdivide the network.

IPv6 addresses are owned by the ISP which provides internet connectivity for the enterprise and not the enterprise itself. An ISP typically provides a 48-bit prefix to each enterprise which connects through the ISP. If an enterprise wants to change ISPs, then the new ISP will provide a different 48-bit prefix and the enterprise will need to renumber its networks as part of the change-over. IPv6 includes support to aid in this change-over, which is described later in this presentation.

IPv6 scoped unicast addressing

- **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
- **Unique local IPv6 unicast addresses for use within a site**
  – **Deprecated by the IETF**
- **Global address prefixes are provided by ISPs**



4 Next generation internet: IPv6

Every IPv6 address, other than the unspecified address, has a specific scope. A scope is a topological span within which the address can be used as a unique identifier for an interface or set of interfaces. The scope of an address is encoded as part of the address.

For unicast addresses, this presentation describes two defined scopes.

First, the link-local scope uniquely identifies interfaces attached to a single link only.

Second, the global scope uniquely identifies interfaces anywhere in the internet.

A scope zone is a connected region of topology of a given scope. For example, a specific link in a network, and the interfaces attached to that link, comprise a single zone of link-local scope. Note that a zone is a particular instance of a topological region (for example, link-1 or link-2), whereas a scope is the size of a topological region (a link or a site).

The example on the slide might help make this a little more clear. The router in the middle is connected to two links. The interface on each link has the same IPv6 address, fe80::12. This is valid because a link-local address only needs to be unique on the link to which the interface it is assigned is attached. To uniquely identify the interface, you must use the combination of the link and the IPv6 address (that is, fe80::12 on link 1).

Link 2 also has two interfaces that have the same link-local address fe80::1234. This is not valid because the two interfaces to fe80::1234 are in the same link-local scope zone, and an IPv6 address must be unique within its scope zone.

IPv6 address textual representation

- An IPv6 address is represented as eight groups of four hex digits (16 bits), separated by colons
  - **2001:0DB8:0:0:240:2BFF:FE3D:71AD**
- Two colons together denote zeroes
- “Slash number” at the end is a prefix length
  - **2001:0DB8::240:2BFF:FE3D:71AD/64**
- A prefix alone is represented as if the interface ID bits are all zero
  - **2001:0DB8::/64**
- IPv4-mapped IPv6 address
  - **::FFFF:a.b.c.d**
- Obviously, this syntax might be a bit difficult for humans - use DNS

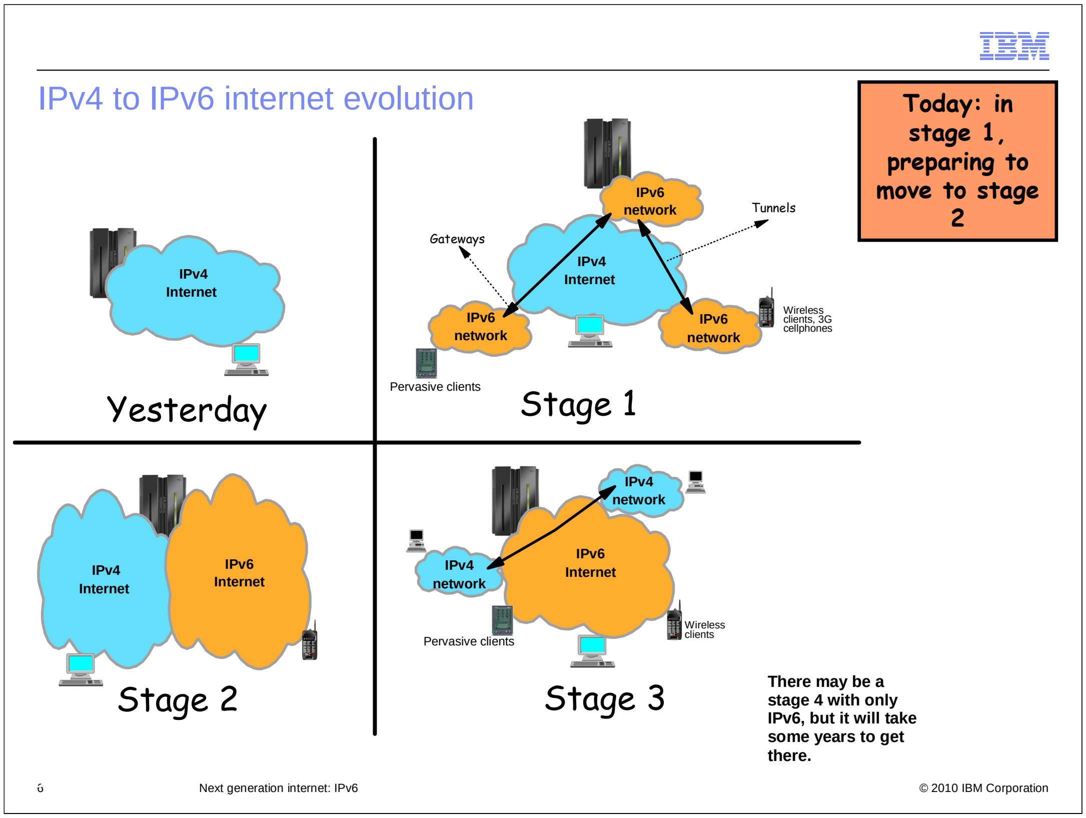
An IPv6 address is 128 bits, or a 16 byte binary number. The textual representation (the way you write them out on paper) is by taking two bytes at a time expressing their value in hexadecimal and separating each 2-byte section with a colon.

Multiple 2-byte sections with only zeroes can be represented as two colons.

The network prefix is always represented using the /prefix-length syntax (no subnet mask syntax is supported for IPv6).

An IPv4-mapped address is represented as the IPv6 address ::FFFF:a.b.c.d where a.b.c.d is the IPv4 address.

You need a name server because no one is able to remember these long addresses.



The deployment of IPv6 into an existing IP network should normally be staged over time. As an initial step, small work groups are beginning to use IPv6 to communicate among one another.

The second stage is that IPv6 networks are small islands of IPv6 connectivity in a sea of IPv4. Eventually, individuals in these isolated islands want to communicate with nodes in one of the other islands, or with devices in the IPv4 network. It is during this period of transition that most migration issues are encountered.

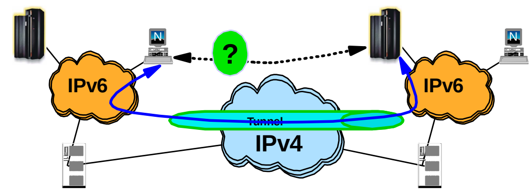
Over time, parallel IPv4 and IPv6 networks run over the same physical network equipment – the same routers, hosts and links. Eventually, as the use of IPv4 recedes, there is a reversal of the initial IPv6 deployment, with islands of IPv4 in a sea of IPv6.

General transition considerations

1

How do IPv4 and IPv6 packets share the physical network?

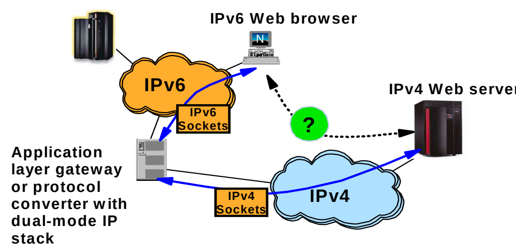
- Dual-stack
- Tunneling of IPv6 over IPv4



2

How do IPv4-only applications communicate with IPv6 applications?

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



7 Next generation internet: IPv6

The IPv6 migration issues can be broken down into two main categories. First, how do IPv6 nodes communicate with one another when the nodes do not have direct IPv6 connectivity? And second, how does an IPv4 application communicate with an IPv6 application? Both problems have several possible solutions.

Overview of IPv6 enablement

- Update the network infrastructure to support IPv6
  - Layer-3 routers
  - Firewalls
  - Intrusion Detection devices
  - Application layer gateways (ALGs)
- The physical media you use today can carry both IPv4 and IPv6 – so no new cabling
- Configure your TCP/IP stack to support IPv6 and IPv4 (dual-mode TCP/IP stack)
- Update your sockets programs
  - IPv6 requires a new sockets interface, known as AF_INET6 (Addressing Family IPv6)
  - Sockets programs that are updated to support AF_INET6 can communicate with both IPv4 and IPv6 sockets partners

8 Next generation internet: IPv6

The network infrastructure will have to be updated to support IPv6 network infrastructure functions.

These functions are neighbor discovery (an auto-addressing technology), IPv6 routing tables (OSPFv3), ICMPv6, name servers with IPv4 and IPv6 addresses, and DHCP servers for IPv6.

The infrastructure includes layer-3 routers, firewalls, intrusion detection devices, application layer gateways (ALGs), and so on.

The physical media you use today can carry both IPv4 and IPv6 – so no new cabling is needed.

However, your TCP/IP stack must be configured to support IPv6 in addition to IPv4 (known as a dual-mode TCP/IP stack).

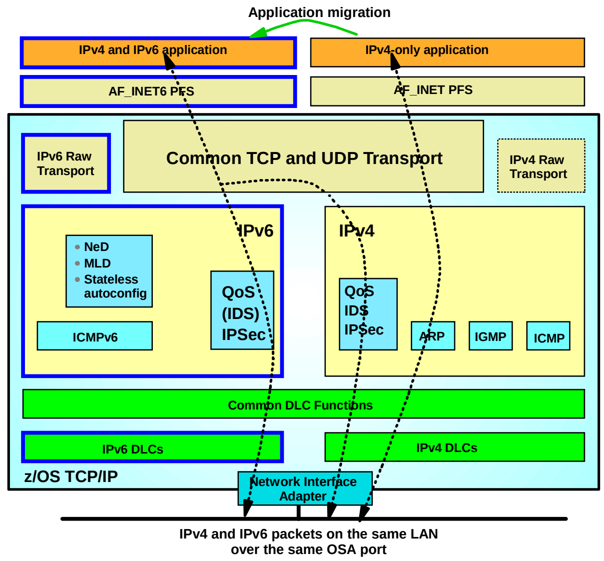
IPv6 requires a new sockets interface, known as AF_INET6 (Addressing Family IPv6). IPv4 sockets programs today use AF_INET, which is IPv4 only. An AF_INET sockets program can only communicate with an IPv4 sockets partner. Sockets programs that are updated to support AF_INET6 can communicate with both IPv4 and IPv6 sockets partners.

As of early 2010, most of the IPv6 compliance requirements address basic network infrastructure functions, and don't focus on the application layer beyond basic network infrastructure support. That will change over time as IPv6 support in applications becomes the focus of the next few years.

Most of the z/OS® Communications Server standard applications are IPv6-enabled so they will support both IPv4 and IPv6 partners. A few are still lacking. Most major subsystems on z/OS (DB2®, MQ, CICS®, IMS-Connect, and WAS) are either already IPv6-enabled or are in the process of enabling IPv6.

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

- A dual-mode TCP/IP stack supports both IPv4 and IPv6 interfaces
- For IPv6 applications, the TCP or UDP transport layer determines per communication partner
  - If the partner is IPv4, chooses IPv4
  - If the partner is IPv6, chooses IPv6
- Raw applications choose IPv4 or IPv6 raw transport



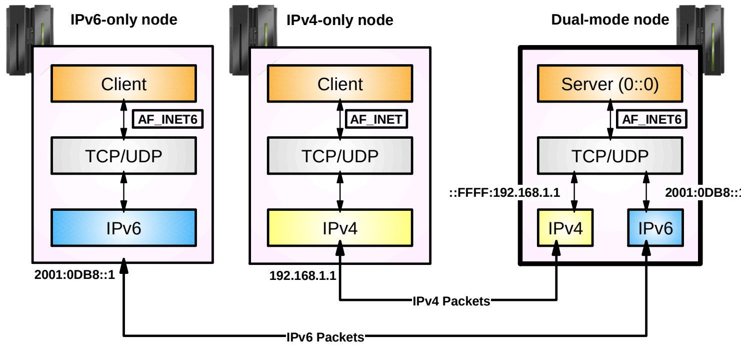
The basic building-block for IPv6 transition is the dual-mode, or dual-stack, TCP/IP node. A dual stack is able to send and receive packets using both an IPv4 network and an IPv6 network.

Existing applications which use AF_INET sockets can continue to run unmodified on a dual stack, but can only communicate with peers by way of the IPv4 network transport. In order for an application to communicate over the IPv6 network, the application must be modified to use AF_INET6 sockets. For TCP and UDP applications, a single AF_INET6 socket can be used to send packets by way of either the IPv4 or IPv6 network transport. The TCP or UDP transport selects the correct network transport protocol to use based on the destination IP address. The transport uses IPv4 if an IPv4-mapped IPv6 address is used, and IPv6 otherwise.

Note that applications which use RAW sockets select the network transport to be used based on the address family of the socket which is created: IPv4 for an AF_INET socket and IPv6 for an AF_INET6 socket. Applications which use RAW sockets are inherently protocol aware, responsible for building the entire IPv4 packet, and much of the IPv6 packet. Fortunately, few applications outside of those shipped with an operating system, such as ping and traceroute, need to use RAW sockets.

Modifying applications to be IPv6-enabled and running on a dual-mode stack' is the preferred migration path for existing applications and middleware, and the best way to implement any new applications. A single IPv6-enabled application is capable of communicating with both IPv4 and IPv6 partners, with the correct network transport protocol being chosen based on the network topology and the partner application’s capabilities.

IPv6-enabled application on a dual mode stack



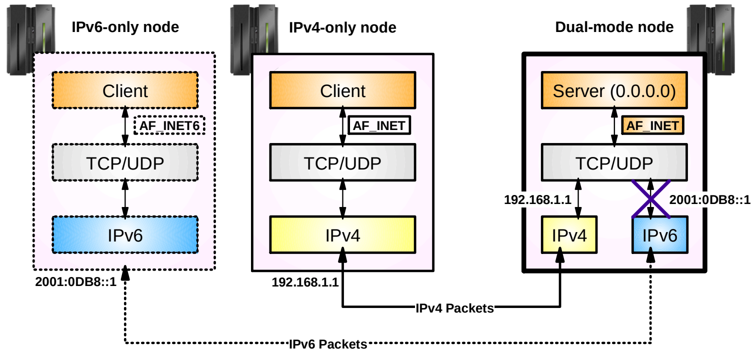
- An IPv6-enabled application can communicate with both IPv4 and IPv6 peers
  - A single socket can be used to send or receive traffic from either IPv4 or IPv6 partners
  - IPv4 packets to the IPv4 partner and IPv6 packets to the IPv6 partner
  - No changes need to be made to the partner application
- 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, 9.67.115.69 can be represented as ::FFFF:9.67.115.69

An IPv6-enabled server running on a dual-mode stack which binds to the IPv6 wildcard address, in6addr_any, is able to accept connections from both IPv4 and IPv6 clients. IPv4 packets are sent and received when communicating with an IPv4 partner, and IPv6 packets are sent and received when communicating with an IPv6 partner. A single AF_INET6 socket can be used for both IPv4 and IPv6 partners. In both cases, the application sees an IPv6 address for the partner: a native IPv6 address for IPv6 partners, and an IPv4-mapped IPv6 address for IPv4 partners.

Upgrading the server to support AF_INET6 sockets is completely transparent to the IPv4 partner and requires no changes to the IPv4 partner. The partner continues to use AF_INET sockets and continues to send and receive IPv4 packets.

The changes to an IPv6-enabled client are similar to those for the IPv6-enabled server. The IPv6-enabled client can communicate with IPv4 and IPv6 servers, and no change is required at the IPv4 server when adding IPv6 support to the client.

IPv4-only application on a dual-mode stack



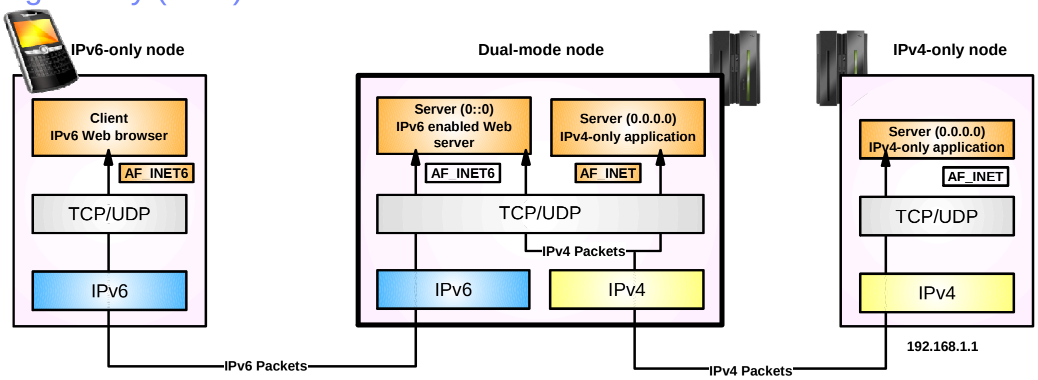
- An IPv4 application running on a dual-mode stack can communicate with an IPv4 partner.
  - The source and destination addresses are native IPv4 addresses
  - The packet which is sent is an IPv4 packet
- If partner is IPv6 running on an IPv6 only stack, then communication fails
  - If partner was on dual-mode stack, then it can 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

IPv4-only applications on a dual-mode stack continue to run as-is. However, such applications are restricted to communicating only over the IPv4 network transport. IPv4 clients running on IPv4-only stack or a dual-mode stack, or IPv6-enabled clients running on a dual-mode stack, can communicate with the IPv4-only server.

However, clients on an IPv6-only stack cannot communicate directly with the IPv4-only server. The IPv6-only client is only capable of sending and receiving IPv6 packets, and the IPv4-only server is only capable of sending and receiving IPv4 packets. Since there is no common network transport protocol over which to transmit data, the two stacks cannot communicate directly.

Note that the same restrictions apply to an IPv4-only client which tries to communicate with an IPv6-only server.

Accessing IPv4-only applications through an IPv6 application layer gateway (ALG)



- An IPv6-only client can access IPv4-only servers by way of 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 might be on the same node as the IPv6 proxy, or might reside on a different node
  - The use of a backend IPv4-only server is, in most cases, completely transparent to the IPv6 client

12 Next generation internet: IPv6

One way for an IPv6-only client to access IPv4-only servers in the network is to use an IPv6 proxy. The proxy establishes an IPv6 connection to the IPv6-only client, and establishes an IPv4 connection to the IPv4-only server. When the client wants to send data to the server, the client sends the data in an IPv6 packet. The proxy receives the data and forwards the data to the server as an IPv4 packet over the IPv4 network. Likewise, when the server wants to send data to the client, the server sends the data in an IPv4 packet to the proxy. The proxy sends the data to the client in an IPv6 packet over the IPv6 network.

Note that the IPv4-only application which the client wants to access might reside on the same server as the proxy, or might reside on a different stack which can be accessed using an IPv4 network transport.

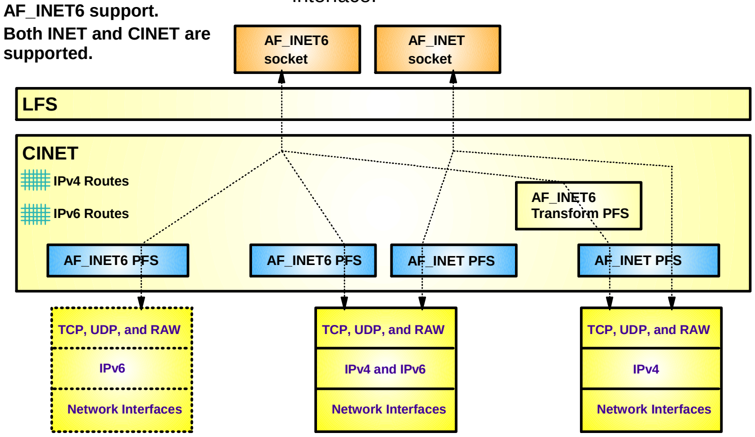
This model is expected to be the initial model for connecting new IPv6-connected devices to existing IPv4 infrastructure servers.

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 AnyNet or 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.

13 Next generation internet: IPv6

You enable IPv6 support on z/OS at the LPAR level in the BPXPRMxx PARMLIB member to enable AF_INET6 support. This can be done in either an INET or CINET environment.

If IPv6 is enabled, you'll always have an IPv6 loopback address; any other IPv6 addresses will need to be defined to the TCP/IP profile. (This is similar to IPv4.) Existing AF_INET programs continue to work just fine.

New AF_INET6 socket programs can communicate with IPv4 partners but they will also be able to communicate with IPv6 partners.

As of z/OS V1R12, TCP/IP does not support an IPv6-only stack. If AF_INET6 is enabled, the stack is dual mode.

Netstat output format LONG or SHORT

- When IPv6 is enabled, most netstat reports look different
  - Without IPv6 enabled, Netstat uses the SHORT report format
    - Might have both local and remote IP address in one 80-character line
    - Override the SHORT format by coding IPCONFIG FORMAT LONG
  - With IPv6 enabled, Netstat uses the LONG report format
    - Each IPv6 address might potentially be up to 45 characters long
- Update any netstat screen-scraping programs

```
MVS TCP/IP NETSTAT CS V1R11       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 Establsh
  Local Socket:   ::ffff:9.42.105.45..23
  Foreign Socket: ::ffff:9.50.52.109..58646
  Application Data:  EZBTNSRV TCPABC81 TSO10001 ET B
```

14 Next generation internet: IPv6

When IPv6 is enabled, netstat reports will look different because the IP addresses are longer. So if you use any automation that examines netstat reports, the programs might need adjustment.

Even before you enable IPv6, you can enable the long netstat report format by coding FORMAT LONG on your IPCONFIG statement in the TCP/IP profile. This option will instruct netstat to use the long report format by default even when the stack is not yet IPv6-enabled.

This will allow you time to change any netstat screen-scraping programs you have developed.

Many of the reports are much more readable and self-explanatory in the long format.

Steps for moving to an IPv6 Environment (1 of 3)

- **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 statements for any IPv6 interfaces
- **IPv6 address selection**
  - Obtain an address block from your ISP, or use an IPv4 address to create a 6to4 prefix
  - Use site-local IPv6 addresses (for test purposes only)
  - Assign IPv6 addresses to the IPv6 interfaces and static VIPAs
  - Manually configure addresses on the INTERFACE statement in the TCP/IP Profile or auto-configure using Neighbor Discovery Stateless Auto-configuration
    - VIPA addresses must be manually configured

15 Next generation internet: IPv6

There are six major steps to move to an IPv6 environment.

First, look into network access.

A local area network can carry both IPv4 and IPv6 packets physically over the same media.

An OSA express port can be used for both IPv4 and IPv6.

You'll need to update your TCP/IP profile to add INTERFACE statements for any new IPv6 interfaces.

If you are communicating between LPARs, you have two choices. You can either use QDIO in a shared LAN or share an OSA, or you can use MPCPTP6 interfaces using either XCF in the same sysplex or ESCON® CTC links if not. If you are running z/9 or z/10, you can use IPv6 hipersockets.

Step two is to think about your IPv6 address selection.

Get a block of addresses from your ISP or use an IPv4 address to create a 6to4 prefix.

For testing, you can use site-local addresses, but avoid them in production. Consider using emerging "Unique Local IPv6 Unicast Addresses" instead of site-local address.

IPv6 addresses can be assigned to the IPv6 interfaces and static VIPAs.

Addresses can be manually configured on the INTERFACE statement or automatically configured using neighbor discovery auto-configuration.

Note that VIPAs must be manually configured.

Steps for moving to an IPv6 Environment (2 of 3)

- **DNS setup**
  - Use DNS BIND 9 Name Server for both IPv4 and IPv6 resources
  - 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, configure a third host name to be used only for IPv6
  - If using stateless auto-configuration to define IPv6 addresses, store static VIPA addresses in DNS since the auto-configured addresses will change over time
- **INET or Common INET**
  - Both are supported for IPv6, but INET is much simpler
  - Do not run IPv4 and dual-mode stacks under CINET
  - AF_INET6 NETWORK statement must be coded in BPXPRMxx before starting IPv6-enabled stacks

Step three is to set up your Domain Name System (DNS).

You can use a DNS BIND 9 name server for both IPv6 and IPv4 addresses. Use an existing host name for IPv4 connectivity to avoid possible disruption in network connectivity and IPv4-only applications on an IPv6-enabled stack. Use separate host names for IPv4 only, IPv6 only, and dual applications.

If you use stateless auto-configuration to define IPv6 addresses, use a static VIPA in your DNS rather than the automatically configured address because that can change.

Step four is to decide whether to use INET or COMMON INET (CINET). INET allows you to have only one TCP/IP stack per system while CINET allows up to eight TCP/IP stacks on one system. You can use either INET or CINET but you should not run a combination of IPv4 and dual mode stacks under CINET. Instead, run dual-mode stacks in a separate LPAR from IPv4 only stacks.

Code the BPXPRMxx PARMLIB member for AF_INET6 NETWORK before starting the IPv6 enabled stacks. All z/OS TCP/IP stacks in an LPAR are either IPv4-only or dual-mode, based on your BPXPRMxx definitions. The only case where this can 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 has been enabled for IPv6 in the BPXPRMxx PARMLIB member.

Steps for moving to an IPv6 Environment (3 of 3)

- **Selection and placement of IPv6 to IPv4 protocol converter or application gateway**
  - z/OS does not implement any functions that will allow IPv6-only stacks to communicate with z/OS-resident AF_INET applications
    - might need an outboard protocol converter or application-layer gateway component
    - This component will only be needed if the test configuration includes IPv6-only platforms
    - Various technologies are being made available by various vendors; SOCKS64 seems the simplest technology right now
- **Connectivity to non-local IPv6 locations**
  - Tunneling might be needed between a router connected to the LAN that z/OS is connected to
  - Might need a router at another location where IPv6 test equipment is located

Step five is to decide if you need an IPv6 to IPv4 protocol converter or application gateway.

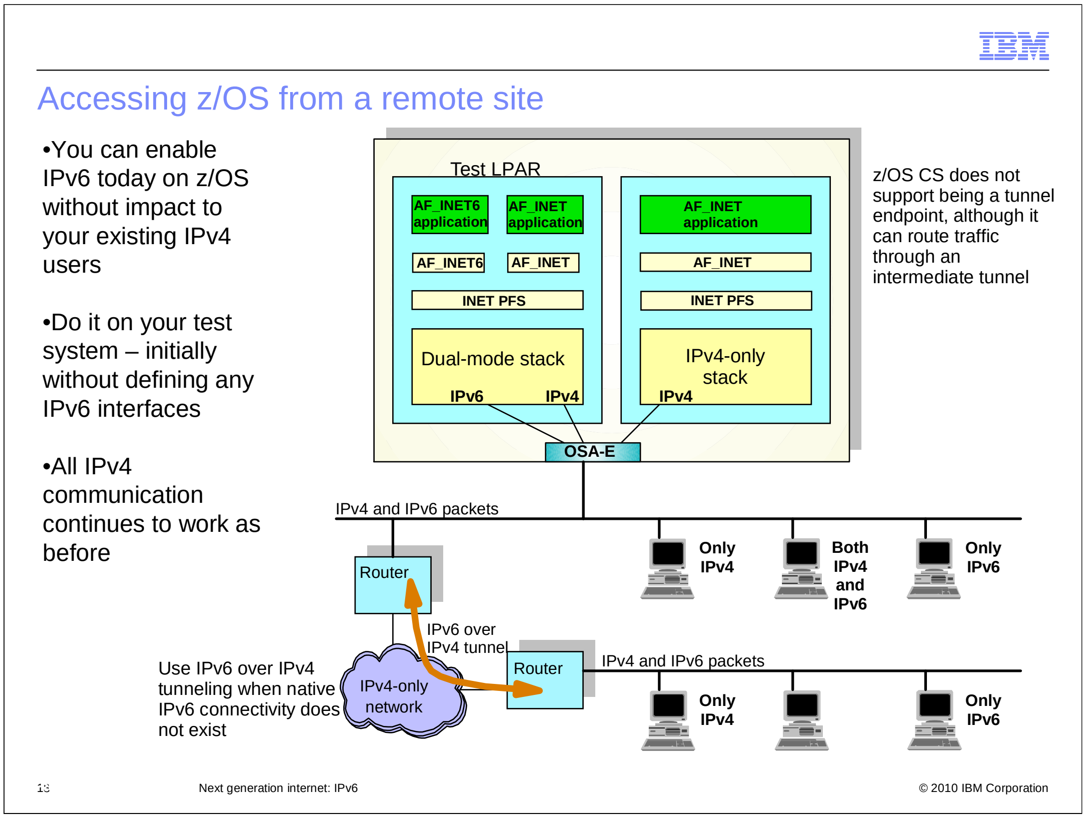
z/OS doesn’t have a function to allow IPv6 only stacks to communicate with AF_INET applications, so if this is a consideration for you, then an outboard protocol converter or application layer gateway component might be needed.

This is only a consideration if there are any IPv6-only platforms.

There are various implementations to do this such as SOCKS64.

Step six is to decide if you need connectivity to any non-local IPv6 locations.

Consider if tunneling is needed if going IPv6-IPv6 over IPv4 in the network anywhere.



Now, putting it all together, here is how you can access z/OS from a remote site.

The IPv4 host can communicate with the dual stack and the IPv4 only stack.

To communicate with the IPv4 only stack, the client application must be IPv4.

The IPv6 only stack can only communicate with the dual mode stack and with applications what are AF_INET6 enabled. Since packets must travel over an IPv4 only network, tunneling must be used. Note that z/OS V1R12 Communications Server does not support being a tunnel endpoint but can route traffic to a tunnel.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_IPv6how.ppt

This module is also available in PDF format at: ../IPv6how.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, disclaimer, and copyright information