4.2.5 IPv4 and IPv6 Interoperability Facts

Transitioning to IPv6 requires time and dedication. IPv6 is not backwards compatible with IPv4; IPv4 hosts and routers do not support IPv6 traffic, and IPv6 hosts and routers do not support IPv4 traffic.

The following table lists strategies for deploying IPv6:

<table>
<tr><th>Method</th><th>Description</th></tr>
<tr><td>Dual Stack</td><td>A common method for moving from IPv4 to IPv6 is referred to as dual stack configuration. In this method, both the IPv4 and IPv6 protocol stacks run concurrently on a host. IPv4 is used to communicate with IPv4 hosts, and IPv6 is used to communicate with IPv6 hosts. Microsoft uses two methods to create a dual stack host:
Windows 2003/XP uses a dual stack implementation, where IPv4 and IPv6 are separate protocols.
Windows Vista and later, as well as Windows Server 2008 and later, uses a dual architecture protocol stack, where IPv4 and IPv6 use common transport and framing layers. By default, Windows uses IPv6 whenever possible. The dual layer architecture means you cannot uninstall either IPv4 or IPv6; however, you can disable one or the other and change their order of priority.</td></tr>
<tr><td>Tunneling</td><td>Tunneling wraps an IPv6 packet within an IPv4 packet, allowing IPv6 hosts or sites to communicate over the existing IPv4 infrastructure. Using tunneling, a device encapsulates IPv6 packets in IPv4 packets for transmission across an IPv4 network, and then the packets are de-encapsulated to their original IPv6 packets by another device at the other end.

You can configure the following tunnel types, and tunnels can be configured manually or automatically:
Router-to-router
Host-to-router or router-to-host
Host-to-host (end-to-end)
Windows Server 2008 and later and Windows clients support the tunneling solutions listed below.

<table>
<tr><td>Manually configured tunnel</td><td>With a manually configured tunnel, tunnel endpoints are configured as point-to-point connections between devices. Manual tunneling:
Is configured between routers at different sites.
Requires dual layer routers as the tunnel endpoints. Hosts can be IPv6-only hosts.
Works through NAT.
Uses a static (manual) association of an IPv6 address with the IPv4 address of the destination tunnel endpoint.
Is configured using Netsh.
Because of the time and effort required for configuration, use manually configured tunnels only when you have a few sites that need to connect through the IPv4 Internet or when you want to configure secure site-to-site associations.</td></tr>
<tr><td>Intra-site Automatic Tunnel Addressing Protocol (ISATAP)</td><td>The Intra-site Automatic Tunnel Addressing Protocol (ISATAP) is a tunneling method for use within a site to provide IPv6 communication over a private IPv4 network. ISATAP tunneling:
Is configured between individual hosts and an ISATAP router.
Requires an IPv6 router and dual layer or IPv6-only clients. Routers and hosts perform tunneling when communicating on the IPv4 network.
Does not work through NAT.
Automatically generates link-local addresses that include the IPv4 address of each host:
The prefix is the well-known link-local prefix FE80::/16.
The remaining prefix values are set to 0.
The first two quartets of the interface ID are set to 0000:5EFE.
The remaining two quartets use the IPv4 address written in either dotted-decimal or hexadecimal notation.
A host with an IPv4 address of 192.168.12.155 would have the following IPv6 address when using ISATAP: FE80::5EFE:C0A8:0C9B (also designated as FE80::5EFE:192.168.12.155).

Use ISATAP to begin a transition to IPv6 within a site.
You can start by adding a single ISATAP router and configuring each host as an ISATAP client.
Vista clients will use ISATAP automatically if they can find the ISATAP router.
Vista clients query the DNS server for a router named ISATAP. When using ISATAP, be sure to use this name for the server or create an A or CNAME record in DNS using ISATAP as the name and pointing to</td></tr>
</table>
</td></tr>
</table>

		the ISATAP router.
	6to4 tunneling	With 6to4 tunneling, tunneling endpoints are configured automatically between devices. 6to4 tunneling: ▪ Is configured between routers at different sites. ▪ Requires routers that provide dual layer support as the tunnel endpoints. Hosts can be IPv6-only hosts. ▪ Works through NAT. ▪ Uses a dynamic association of an IPv6 site prefix to the IPv4 address of the destination tunnel endpoint. ▪ Automatically generates an IPv6 address for the site using the 2002::/16 prefix followed by the public IPv4 address of the tunnel endpoint router. For example, a router with the IPv4 address of 207.142.131.202 would serve the site with the following prefix: 2002:CF8E:83CA::/48 (CF8E:83CA is the hexadecimal equivalent of 207.142.131.202). Use 6to4 tunneling to dynamically connect multiple sites through the IPv4 Internet. Because of its dynamic configuration, 6to4 tunneling is easier to administer than manual tunneling.
	Teredo tunneling	Teredo (also known as NAT traversal or NAT-T) establishes the tunnel between individual IPv6 hosts so they can communicate through a private or public IPv4 network. Teredo is a last-resort technology that is used only when there is no native IPv6, ISATAP, or 6to4 connectivity present between hosts. Teredo tunneling: ▪ Is configured between individual hosts. ▪ Has dual layer hosts that perform IPv6 tunneling to send on the IPv4 network. ▪ Works through NAT. ▪ Uses a 2001::/32 prefix followed by the IPv4 public address converted to hexadecimal. For example, the IPv4 public address of 207.142.131.202 would provide clients with the prefix 2001:0:CF8E:83CA::/64. For Windows Vista and Windows 7, the Teredo component is enabled, but inactive by default. In Windows Server 2012, Teredo is enabled by default only on non-domain networks (it is disabled by default on Windows Server 2008 and 2003 SP1). To use Teredo, a user must either install an application that needs to use Teredo or configure the advanced settings on a Windows Firewall exception to use edge traversal. Teredo behavior differs when machines are members of a domain. Teredo is disabled on XP and Server 2003 machines that belong to a domain. Teredo is enabled on Vista and 2008 machines that belong to a domain. Teredo is disabled by default on Windows 8 and Windows Server 2012 machines that are part of a domain.
PortProxy	PortProxy is a TCP proxy that allows an IPv4-only host to communicate with an IPv6-only host. PortProxy does this by transmitting TCP traffic for application-layer protocols that do not embed address or port information in the TCP segment. Thus, an application like FTP does not work across a PortProxy computer because FTP embeds addresses when using the FTP Port command. To configure PortProxy, use the **Netsh interface portproxy** command with the necessary parameters.	
IPv4 Compatible Address	An IPv4 address that is compatible with IPv6 has ten octets. The last four octets are the device's IPv4 address. The format is: 0:0:0:0:0:0:w:x:y:z	
IPv4 Mapped Address	If a device is not compatible with IPv6, you can use an IPv4 mapped address. This address is used to represent an IPv4-only node to an IPv6 node. The sixth octet contains FFFF with the last four octets as the IPv4 address of the device. The format is: 0:0:0:0:0:FFFF:w:x:y:z ::FFFF:w.x.y.z is a simplified version.	
IPv6 to IPv4 Address	An IPv6-to-IPv4 address allows IPv6 packets to travel over an IPv4 network, such as the IPv4 Internet, without additional configuration or tunneling. This type of addressing works best when an IPv6-to-IPv4 router is used. The first octet is 2002, the second octet contains the first two bytes of the IPv4 address, and the third octet contains the second two bytes of the IPv4 address. The format is: 2002:u:v::/16	