



## IPv6/IPv4 Coexistence and Migration

*Microsoft Corporation*

*Published: August 2002*

---

### **Abstract**

The migration of IPv4 to IPv6 will not happen overnight. Rather, there will be a period of transition when both protocols are in use over the same infrastructure. To address this, the designers of IPv6 have created technologies and types of addresses so that nodes can communicate with each other in a mixed environment, even if they are separated by an IPv4 infrastructure. This article describes IPv4 and IPv6 coexistence and migration technologies and how these technologies are supported by the IPv6 protocol for the Windows Server 2003 family. This article is intended for network engineers and support professionals who are already familiar with basic networking concepts, TCP/IP, and IPv6.



*This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein. The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.*

*This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AS TO THE INFORMATION IN THIS DOCUMENT.*

*Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.*

*Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.*

*© 2002 Microsoft Corporation. All rights reserved.*

*Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.*

*The names of actual companies and products mentioned herein may be the trademarks of their respective owners.*

---

## Contents

<b>Introduction</b> .....	<b>1</b>
Node Types.....	1
Compatibility Addresses .....	2
<b>Coexistence Mechanisms</b> .....	<b>4</b>
Dual IP Layer .....	4
IPv6 over IPv4 Tunneling.....	5
DNS Infrastructure .....	6
Address Records.....	6
Pointer Records.....	6
Address Selection Rules .....	6
<b>Tunneling Configurations</b> .....	<b>7</b>
Router-to-Router .....	7
Host-to-Router and Router-to-Host.....	8
Host-to-Host.....	8
Types of Tunnels.....	9
Configured Tunnels.....	9
Automatic Tunnels.....	9
<b>6to4</b> .....	<b>11</b>
6to4 Support in the Windows Server 2003 Family.....	13
<b>ISATAP</b> .....	<b>16</b>
Using an ISATAP Router .....	17
Resolving the ISATAP Name .....	18
Resolving the _ISATAP Name for Windows XP .....	19
Using the netsh interface ipv6 isatap set router Command .....	20
Setting up an ISATAP Router .....	20
ISATAP and 6to4 Example .....	20
Part 1: From ISATAP Host A to 6to4 Router A .....	23
Part 2: From 6to4 Router A to 6to4 Router B.....	23
Part 3: From 6to4 Router B to ISATAP Host B .....	24

<b>PortProxy</b> .....	<b>25</b>
<b>Migrating to IPv6</b> .....	<b>27</b>
<b>Appendix A: IPv6 Automatic Tunneling</b> .....	<b>28</b>
<b>Appendix B: 6over4</b> .....	<b>29</b>
<b>Summary</b> .....	<b>32</b>
<b>Related Links</b> .....	<b>33</b>

---

## Introduction

Protocol transitions are not easy and the transition from IPv4 to IPv6 is no exception. Protocol transitions are typically deployed by installing and configuring the new protocol on all nodes within the network and verifying that all node and router operations work successfully. Although this might be possible in a small or medium sized organization, the challenge of making a rapid protocol transition in a large organization is very difficult. Additionally, given the scope of the Internet, rapid protocol transition becomes an impossible task.

The designers of IPv6 recognize that the transition from IPv4 to IPv6 will take years and that there might be organizations or hosts within organizations that will continue to use IPv4 indefinitely. Therefore, while migration is the long-term goal, equal consideration must be given to the interim coexistence of IPv4 and IPv6 nodes.

The designers of IPv6 in the original "The Recommendation for the IP Next Generation Protocol" specification (RFC 1752) defined the following transition criteria:

- Existing IPv4 hosts can be upgraded at any time, independent of the upgrade of other hosts or routers.
- New hosts, using only IPv6, can be added at any time, without dependencies on other hosts or routing infrastructure.
- Existing IPv4 hosts, with IPv6 installed, can continue to use their IPv4 addresses and do not need additional addresses.
- Little preparation is required to either upgrade existing IPv4 nodes to IPv6 or deploy new IPv6 nodes.

The inherent lack of dependencies between IPv4 and IPv6 hosts, IPv4 routing infrastructure, and IPv6 routing infrastructure requires a number of mechanisms that allow seamless coexistence.

### Note

Except where noted, the support for coexistence and migration is the same for the IPv6 protocol for the Windows Server 2003 family and the IPv6 protocol for Windows XP or Windows XP with Service Pack 1 (SP1).

## Node Types

RFC 2893 defines the following node types:

- IPv4-only node

A node that implements only IPv4 (and has only IPv4 addresses). This node does not support IPv6. Most hosts and routers installed today are IPv4-only nodes.

- IPv6-only node

A node that implements only IPv6 (and has only IPv6 addresses). This node is only able to communicate with IPv6 nodes and applications. This type of node is not common today, but may

become more prevalent as smaller devices such as cellular phones and handheld computing devices include IPv6 stacks.

- **IPv6/IPv4 node**

A node that has both IPv4 and IPv6 implemented. This node is IPv6-enabled if it has an IPv6 interface configured.

- **IPv4 node**

A node that implements IPv4 (it can send and receive IPv4 packets). An IPv4 node can be an IPv4-only node or an IPv6/IPv4 node.

- **IPv6 node**

A node that implements IPv6 (it can send and receive IPv6 packets). An IPv6 node can be an IPv6-only node or an IPv6/IPv4 node.

For coexistence to occur, the largest number of nodes (IPv4 or IPv6 nodes) can communicate using an IPv4 infrastructure, an IPv6 infrastructure, or an infrastructure that is a combination of IPv4 and IPv6. True migration is achieved when all IPv4 nodes are converted to IPv6-only nodes. However, for the foreseeable future, practical migration is achieved when as many IPv4-only nodes as possible are converted to IPv6/IPv4 nodes. IPv4-only nodes can communicate with IPv6-only nodes only when using an IPv4-to-IPv6 proxy or translation gateway.

## Compatibility Addresses

The following addresses are defined to aid in the coexistence of IPv4 and IPv6 nodes:

- **IPv4-compatible addresses**

The IPv4-compatible address, 0:0:0:0:0:w.x.y.z or ::w.x.y.z (where w.x.y.z is the dotted decimal representation of a public IPv4 address), is used by IPv6/IPv4 nodes that are communicating with IPv6 over an IPv4 infrastructure. When the IPv4-compatible address is used as an IPv6 destination, the IPv6 traffic is automatically encapsulated with an IPv4 header and sent to the destination using the IPv4 infrastructure.

- **IPv4-mapped addresses**

The IPv4-mapped address, 0:0:0:0:FFFF:w.x.y.z or ::FFFF:w.x.y.z, is used to represent an IPv4-only node to an IPv6 node. It is used only for internal representation. The IPv4-mapped address is never used as a source or destination address of an IPv6 packet. The IPv6 protocol for the Windows Server 2003 family does not support the use of IPv4-mapped addresses. The IPv4-mapped address is used by some IPv6 implementations when acting as a translator between IPv4-only and IPv6-only nodes.

- **6over4 addresses**

6over4 addresses are comprised of a valid 64-bit unicast address prefix and the interface identifier ::WWXX:YYZZ (where WWXX:YYZZ is the colon-hexadecimal representation of w.x.y.z, a unicast IPv4 address assigned to an interface). An example of a link-local 6over4 address based on the IPv4 address of 131.107.4.92 is FE80::836B:45C. 6over4 addresses are used to represent a host when using the automatic tunneling mechanism defined in RFC 2529. For more information, see Appendix B in this article.

- **6to4 addresses**

6to4 addresses are based on the prefix 2002:WWXX:YYZZ::/48 (where WWXX:YYZZ is the colon-hexadecimal representation of w.x.y.z, a public IPv4 address assigned to an interface). 6to4 addresses are used to represent a site when using the automatic tunneling mechanism defined in RFC 3056, also known as 6to4. For more information, see "6to4" in this article.

- **ISATAP addresses**

Intra-site Automatic Tunnel Addressing Protocol (ISATAP) addresses are composed of a valid 64-bit unicast address prefix and the interface identifier ::0:5EFE:w.x.y.z (where w.x.y.z is a unicast IPv4 address assigned to an interface). An example of a link-local ISATAP address is FE80::5EFE:131.107.4.92. ISATAP addresses are used to represent a host when using the automatic tunneling mechanism defined in the Internet draft titled "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)" (draft-ietf-ngtrans-isatap-0x.txt). For more information, see "ISATAP" in this article.

---

## Coexistence Mechanisms

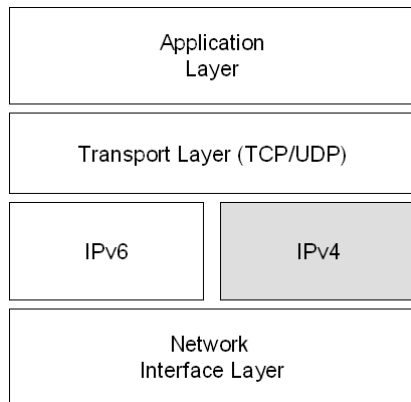
To coexist with an IPv4 infrastructure and to provide eventual migration to an IPv6-only infrastructure, the following mechanisms are used:

- Dual IP layer
- IPv6 over IPv4 tunneling
- DNS infrastructure

### Dual IP Layer

The dual IP layer is an implementation of the TCP/IP suite of protocols that includes both an IPv4 Internet layer and an IPv6 Internet layer. This is the mechanism used by IPv6/IPv4 nodes so that communication with both IPv4 and IPv6 nodes can occur. A dual IP layer contains a single implementation of Host-to-Host layer protocols such as TCP and UDP. All upper layer protocols in a dual IP layer implementation can communicate over IPv4, IPv6, or IPv6 tunneled in IPv4.

Figure 1 shows a dual IP layer architecture.



*Figure 1: A Dual IP Layer Architecture*

The IPv6 protocol for the Windows Server 2003 family is not a dual IP layer. The IPv6 protocol driver, Tcpi6.sys, contains a separate implementation of TCP and UDP and is sometimes referred to as a dual-stack implementation. Figure 2 shows the dual stack architecture of the IPv6 protocol for the Windows Server 2003 family.

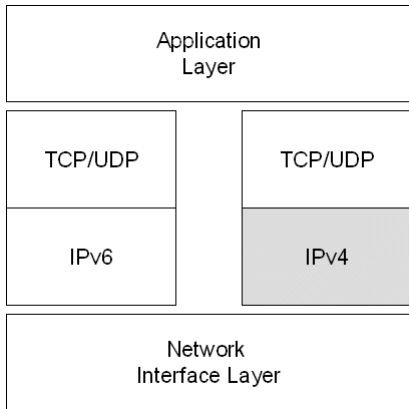


Figure 2: The Dual Stack Architecture of the IPv6 Protocol for the Windows Server 2003 Family

Although the IPv6 protocol for the Windows Server 2003 family is not a dual IP layer, it functions in the same way as a dual IP layer in terms of providing functionality for coexistence and migration.

### IPv6 over IPv4 Tunneling

IPv6 over IPv4 tunneling is the encapsulation of IPv6 packets with an IPv4 header so that IPv6 packets can be sent over an IPv4 infrastructure. Within the IPv4 header:

- The IPv4 Protocol field is set to 41 to indicate an encapsulated IPv6 packet.
- The Source and Destination fields are set to IPv4 addresses of the tunnel endpoints. The tunnel endpoints are either manually configured as part of the tunnel interface or are automatically derived from the sending interface, the next-hop address of the matching route, or the source and destination IPv6 addresses in the IPv6 header.

Figure 3 shows IPv6 over IPv4 tunneling.

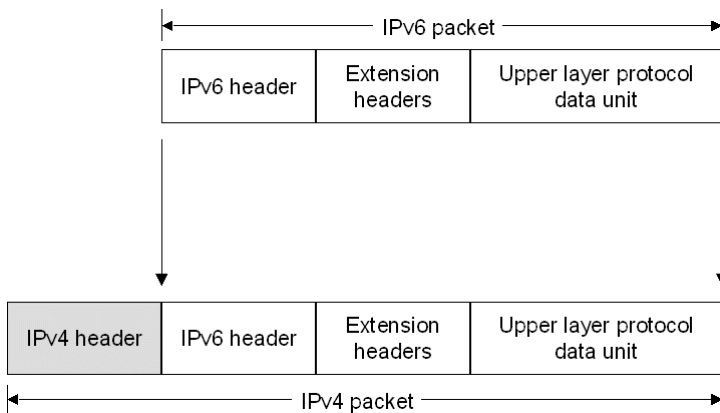


Figure 3: IPv6 over IPv4 Tunneling

For IPv6 over IPv4 tunneling, the IPv6 path maximum transmission unit (MTU) for the destination is typically 20 less than the IPv4 path MTU for the destination. However, if the IPv4 path MTU is not stored for each tunnel, there are instances where the IPv4 packet will need to be fragmented at an intermediate IPv4

router. In this case, IPv6 over IPv4 tunneled packet must be sent with the Don't Fragment flag in the IPv4 header set to 0.

## DNS Infrastructure

A Domain Name System (DNS) infrastructure is needed for successful coexistence because of the prevalent use of names (rather than addresses) to refer to network resources. Upgrading the DNS infrastructure consists of populating the DNS servers with records to support IPv6 name-to-address and address-to-name resolutions. After the addresses are obtained using a DNS name query, the sending node must select which addresses are used for communication.

### Address Records

The DNS infrastructure must contain the following resource records (populated either manually or dynamically) for the successful resolution of domain names to addresses:

- A records for IPv4-only and IPv6/IPv4 nodes
- AAAA records for IPv6-only and IPv6/IPv4 nodes

### Pointer Records

The DNS infrastructure must contain the following resource records (populated either manually or dynamically) for the successful resolution of address to domain names (reverse queries):

- PTR records in the IN-ADDR.ARPA domain for IPv4-only and IPv6/IPv4 nodes
- PTR records in the IP6.INT domain for IPv6-only and IPv6/IPv4 nodes (optional).

### Address Selection Rules

For name-to-address resolution, after the querying node obtains the set of addresses corresponding to the name, the node must determine the set of addresses to choose as source and destination for outbound packets.

This is not an issue in today's prevalent IPv4-only environment. However, in an environment in which IPv4 and IPv6 coexist, the set of addresses returned in a DNS query may contain multiple IPv4 and IPv6 addresses. The querying host is configured with at least one IPv4 address and (typically) multiple IPv6 addresses. Deciding which type of address (IPv4 vs. IPv6), and then the scope of the address (public vs. private for IPv4 and link-local vs. site-local vs. global vs. coexistence for IPv6), for both the source and the destination addresses is not an easy task.

Default address selection rules are currently under discussion and are described in the Internet draft titled "Default Address Selection for IPv6" (draft-ietf-ipv6-default-addr-select-0x.txt).

You can view the default address selection rules for the IPv6 protocol for the Windows Server 2003 family using the **netsh interface ipv6 show prefixpolicy** command to display the prefix policy table. You can modify the entries in the prefix policy table using the **netsh interface ipv6 add|set|delete prefixpolicy** commands. By default, IPv6 addresses in DNS query responses are preferred over IPv4 addresses.

## Tunneling Configurations

RFC 2893 defines the following tunneling configurations with which to tunnel IPv6 traffic between IPv6/IPv4 nodes over an IPv4 infrastructure:

- Router-to-Router
- Host-to-Router or Router-to-Host
- Host-to-Host

### Note

IPv6 over IPv4 tunneling only describes an encapsulation of IPv6 packets with an IPv4 header so that IPv6 nodes are reachable across an IPv4 infrastructure. Unlike tunneling for the Point-to-Point Tunneling Protocol (PPTP) and Layer Two Tunneling Protocol (L2TP), there is no exchange of messages for tunnel setup, maintenance, or termination. Additionally, IPv6 over IPv4 tunneling does not provide security for tunneled IPv6 packets.

### Router-to-Router

In the router-to-router tunneling configuration, two IPv6/IPv4 routers connect two IPv4 or IPv6 infrastructures over an IPv4 infrastructure. The tunnel endpoints span a logical link in the path between the source and destination. The IPv6 over IPv4 tunnel between the two routers acts as a single hop. Routes within each IPv4 or IPv6 infrastructure point to the IPv6/IPv4 router on the edge. For each IPv6/IPv4 router, there is a tunnel interface representing the IPv6 over IPv4 tunnel and routes that use the tunnel interface.

Figure 4 shows router-to-router tunneling.

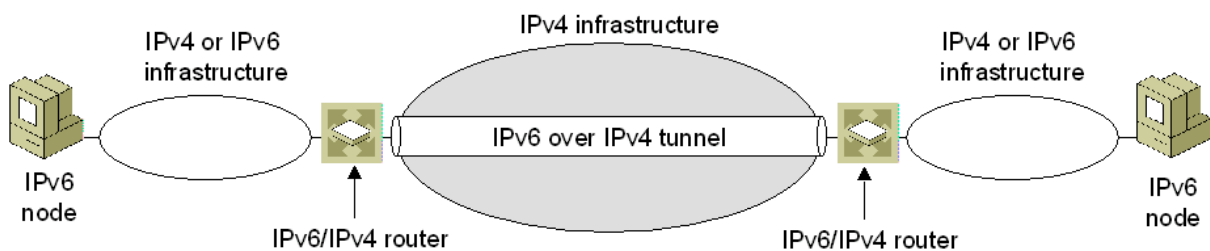


Figure 4: Router-to-Router Tunneling

Examples of this tunneling configuration are:

- An IPv6-only test lab that tunnels across an organization's IPv4 infrastructure to reach the IPv6 Internet.
- Two IPv6-only routing domains that tunnel across the IPv4 Internet.
- A 6to4 router that tunnels across the IPv4 Internet to reach another 6to4 router or a 6to4 relay router. For more information about 6to4, see "6to4" in this article.

## Host-to-Router and Router-to-Host

In the host-to-router tunneling configuration, an IPv6/IPv4 node that resides within an IPv4 infrastructure creates an IPv6 over IPv4 tunnel to reach an IPv6/IPv4 router. The tunnel endpoints span the first segment of the path between the source and destination nodes. The IPv6 over IPv4 tunnel between the IPv6/IPv4 node and the IPv6/IPv4 router acts as a single hop.

On the IPv6/IPv4 node, a tunnel interface representing the IPv6 over IPv4 tunnel is created and a route (typically a default route) is added using the tunnel interface. The IPv6/IPv4 node tunnels the IPv6 packet based on the matching route, the tunnel interface, and the next-hop address of the IPv6/IPv4 router.

In the router-to-host tunneling configuration, an IPv6/IPv4 router creates an IPv6 over IPv4 tunnel across an IPv4 infrastructure to reach an IPv6/IPv4 node. The tunnel endpoints span the last segment of the path between the source node and destination node. The IPv6 over IPv4 tunnel between the IPv6/IPv4 router and the IPv6/IPv4 node acts as a single hop.

On the IPv6/IPv4 router, a tunnel interface representing the IPv6 over IPv4 tunnel is created and a route (typically a subnet route) is added using the tunnel interface. The IPv6/IPv4 router tunnels the IPv6 packet based on the matching subnet route, the tunnel interface, and the destination address of the IPv6/IPv4 node.

Figure 5 shows host-to-router (for traffic traveling from Node A to Node B) and router-to-host (for traffic traveling from Node B to Node A) tunneling.

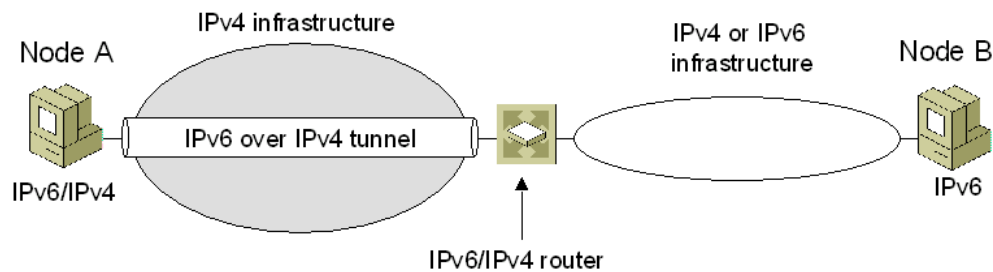


Figure 5: Host-to-Router and Router-to-Host Tunneling

Examples of host-to-router and router-to-host tunneling are:

- An IPv6/IPv4 host that tunnels across an organization's IPv4 infrastructure to reach the IPv6 Internet.
- An ISATAP host that tunnels across an IPv4 network to an ISATAP router to reach the IPv4 Internet, another IPv4 network, or an IPv6 network. For more information about ISATAP, see "ISATAP" in this article.
- An ISATAP router that tunnels across an IPv4 network to reach an ISATAP host.

## Host-to-Host

In the host-to-host tunneling configuration, an IPv6/IPv4 node that resides within an IPv4 infrastructure creates an IPv6 over IPv4 tunnel to reach another IPv6/IPv4 node that resides within the same IPv4

infrastructure. The tunnel endpoints span the entire path between the source and destination nodes. The IPv6 over IPv4 tunnel between the IPv6/IPv4 nodes acts as a single hop.

On each IPv6/IPv4 node, an interface representing the IPv6 over IPv4 tunnel is created. Routes might be present to indicate that the destination node is on the same logical subnet defined by the IPv4 infrastructure. Based on the sending interface, the optional route, and the destination address, the sending host tunnels the IPv6 traffic to the destination.

Figure 6 shows host-to-host tunneling.

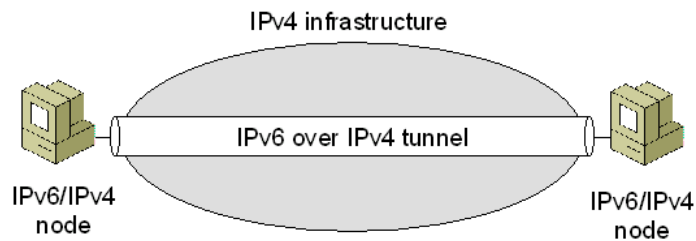


Figure 6: Host-to-Host Tunneling

Examples of this tunneling configuration are:

- IPv6/IPv4 hosts that use ISATAP addresses to tunnel across an organization's IPv4 infrastructure
- IPv6/IPv4 hosts that use IPv4-compatible addresses to tunnel across an organization's IPv4 infrastructure.

## Types of Tunnels

RFC 2893 defines the following types of tunnels:

- Configured
- Automatic

### Configured Tunnels

A configured tunnel requires manual configuration of tunnel endpoints. In a configured tunnel, the IPv4 addresses of tunnel endpoints are not derived from addresses that are encoded in the IPv6 source or destination addresses or the next-hop address of the matching route.

Typically, router-to-router tunneling configurations are manually configured. The tunnel interface configuration, consisting of the IPv4 addresses of the tunnel endpoints, must be manually specified along with static routes that use the tunnel interface.

To manually create configured tunnels for the IPv6 protocol for the Windows Server 2003 family, use the **netsh interface ipv6 add v6v4tunnel** command.

### Automatic Tunnels

An automatic tunnel is a tunnel that does not require manual configuration. Tunnel endpoints are determined by the use of logical tunnel interfaces, routes, and source and destination IPv6 addresses.

The IPv6 protocol for the Windows Server 2003 family supports the following automatic tunneling technologies:

- 6to4  
Enabled by default. For more information, see "6to4" in this article.
- ISATAP  
Enabled by default. For more information, see "ISATAP" in this article.
- IPv6 Automatic Tunneling  
Disabled by default. For more information, see Appendix A.
- 6over4  
Disabled by default. For more information, see Appendix B.

---

## 6to4

6to4 is an address assignment and router-to-router automatic tunneling technology that is used to provide unicast IPv6 connectivity between IPv6 sites and hosts across the IPv4 Internet. 6to4 uses the global address prefix:

`2002:WWXX:YYZZ::/48`

in which `WWXX:YYZZ` is the NLA ID portion of a global address and the colon-hexadecimal representation of a public IPv4 address (`w.x.y.z`) assigned to a site or host. The full 6to4 address is:

`2002:WWXX:YYZZ:[SLA ID]:[Interface ID]`

6to4 is described in RFC 3056, which defines the following terms:

- 6to4 host

Any IPv6 host that is configured with at least one 6to4 address (a global address with the `2002::/16` prefix). 6to4 hosts do not require any manual configuration and create 6to4 addresses using standard address autoconfiguration mechanisms.

- 6to4 router

An IPv6/IPv4 router that supports the use of a 6to4 tunnel interface and is typically used to forward 6to4-addressed traffic between the 6to4 hosts within a site and other 6to4 routers or 6to4 relay routers on an IPv4 internetwork, such as the Internet. 6to4 routers require additional processing logic for proper encapsulation and decapsulation and might require additional manual configuration.

- 6to4 relay router

An IPv6/IPv4 router that forwards 6to4-addressed traffic between 6to4 routers on the Internet and hosts on the IPv6 Internet.

Figure 7 shows 6to4 components.

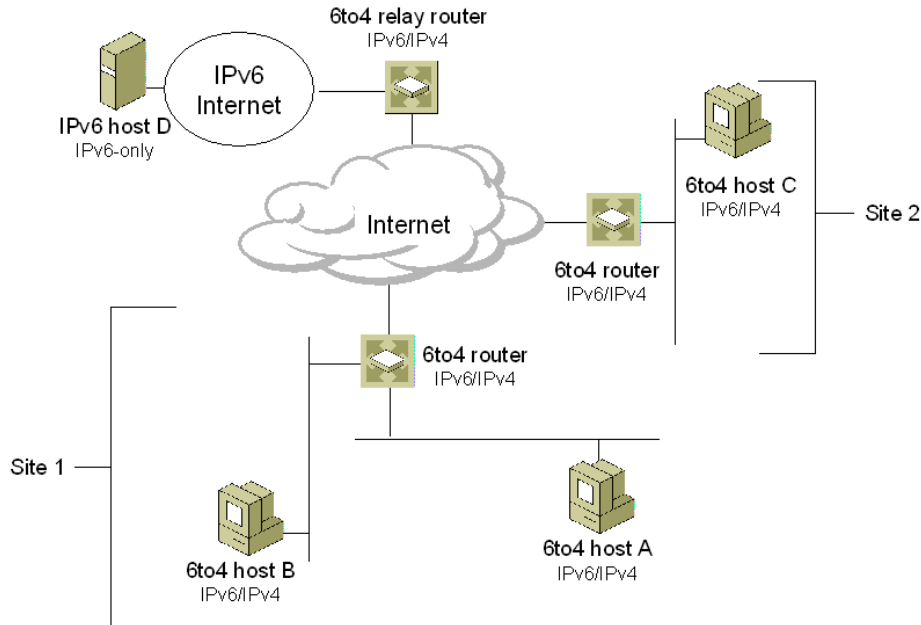


Figure 7: 6to4 Components

Within a site, local IPv6 routers advertise `2002:WWXX:YYZZ:[SLA ID]::/64` prefixes so that hosts can create an autoconfigured 6to4 address and 64-bit prefix routes are used to deliver traffic between 6to4 hosts within the site. Hosts on individual subnets are automatically configured with a 64-bit subnet route for direct delivery to neighbors and a default route with the next-hop address of the advertising router. All IPv6 traffic that does not match a 64-bit prefix used by one of the subnets within the site is forwarded to a 6to4 router on the site border.

The 6to4 router on the site border has a `2002::/16` route that is used to forward traffic to other 6to4 sites and a default route (`::/0`) that is used to forward traffic to a 6to4 relay router.

In the example network shown in Figure 7, Host A and Host B can communicate with each other because of a default route using the next-hop address of the 6to4 router in Site 1. When Host A communicates with Host C in another site, Host A sends the traffic is sent to the 6to4 router in Site 1 as IPv6 packets. The 6to4 router in Site 1, using the 6to4 tunnel interface and the `2002::/16` route in its routing table, encapsulates the packet with an IPv4 header and tunnels it to the 6to4 router in Site 2. When the 6to4 router in Site 2 receives the tunneled packet, it removes the IPv4 header and, using the 64-bit prefix route in its routing table, forwards the IPv6 packet to Host C.

In this example, Host A (with the interface ID `ID_A`) resides on subnet 1 within Site 1 that uses the public IPv4 address of `157.60.91.123`. Host C (with the interface ID `ID_C`) resides on subnet 2 within Site 2 that uses the public IPv4 address of `131.107.210.49`. When the IPv4-encapsulated IPv6 packet is sent by the 6to4 router in Site 1 to the 6to4 router in Site 2, the addresses in the IPv4 and IPv6 headers are listed in Table 1.

**Table 1 Example 6to4 Addresses**

Field	Value
IPv6 Source Address	2002:9D3C:5B7B:1:[ID_A]
IPv6 Destination Address	2002:836B:D231:2:[ID_C]
IPv4 Source Address	157.60.91.123
IPv4 Destination Address	131.107.210.49

For a more detailed example of 6to4 traffic using ISATAP-derived interface identifiers, see "ISATAP" in this article.

When you use 6to4 hosts, an IPv6 routing infrastructure within a site, a 6to4 router at the site boundary, and a 6to4 relay router, the following types of communication are possible:

- A 6to4 host can communicate with another 6to4 host within the same site.  
This type of communication is available by using the IPv6 routing infrastructure, which provides reachability to all hosts within the site. In Figure 7, this is the communication between Host A and Host B.
- A 6to4 host can communicate with 6to4 hosts in other sites across the IPv4 Internet.  
This type of communication occurs when a 6to4 host forwards IPv6 traffic that is destined to a 6to4 host in another site to the local site 6to4 router. The local site 6to4 router tunnels the IPv6 traffic to the 6to4 router at the destination site on the IPv4 Internet. The 6to4 router at the destination site removes the IPv4 header and forwards the IPv6 packet to the appropriate 6to4 host by using the IPv6 routing infrastructure of the destination site. In Figure 7, this is the communication between Host A and Host C.
- A 6to4 host can communicate with hosts on the IPv6 Internet.  
This type of communication occurs when a 6to4 host forwards IPv6 traffic that is destined for an IPv6 Internet host to the local site 6to4 router. The local site 6to4 router tunnels the IPv6 traffic to a 6to4 relay router that is connected to both the IPv4 Internet and the IPv6 Internet. The 6to4 relay router removes the IPv4 header and forwards the IPv6 packet to the appropriate IPv6 Internet host by using the IPv6 routing infrastructure of the IPv6 Internet. In Figure 7, this is the communication between Host A and Host D.

All of these types of communication use IPv6 traffic without the requirement of obtaining either a direct connection to the IPv6 Internet or an IPv6 global address prefix from an Internet service provider (ISP).

## 6to4 Support in the Windows Server 2003 Family

Support for 6to4 hosts and 6to4 routers are provided by the 6to4 component that is part of the IPv6 protocol for the Windows Server 2003 family. If there is a public IPv4 address assigned to an interface on the host and a global prefix is not received in a router advertisement, the 6to4 component:

- Automatically configures 6to4 addresses on the 6to4 Tunneling Pseudo-Interface for all public IPv4 addresses that are assigned to interfaces on the computer.
- Automatically creates a 2002::/16 route that forwards all 6to4 traffic with the 6to4 Tunneling

Pseudo-Interface (interface index 3). All traffic forwarded by this host to 6to4 destinations is encapsulated with an IPv4 header.

- Automatically performs a DNS query to obtain the IPv4 address of a 6to4 relay router on the Internet. You can also use the **netsh interface ipv6 6to4 set relay** command to specify the DNS name to query. If the query is successful, a default route is added using the 6to4 Tunneling Pseudo-Interface and the next-hop address is set to the 6to4 address of the 6to4 relay router.

The results of the 6to4 component autoconfiguration vary depending on the configuration of the host. Figure 8 shows how 6to4 is configured for different types of hosts running a member of the Windows Server 2003 family (except IPv6 host D).

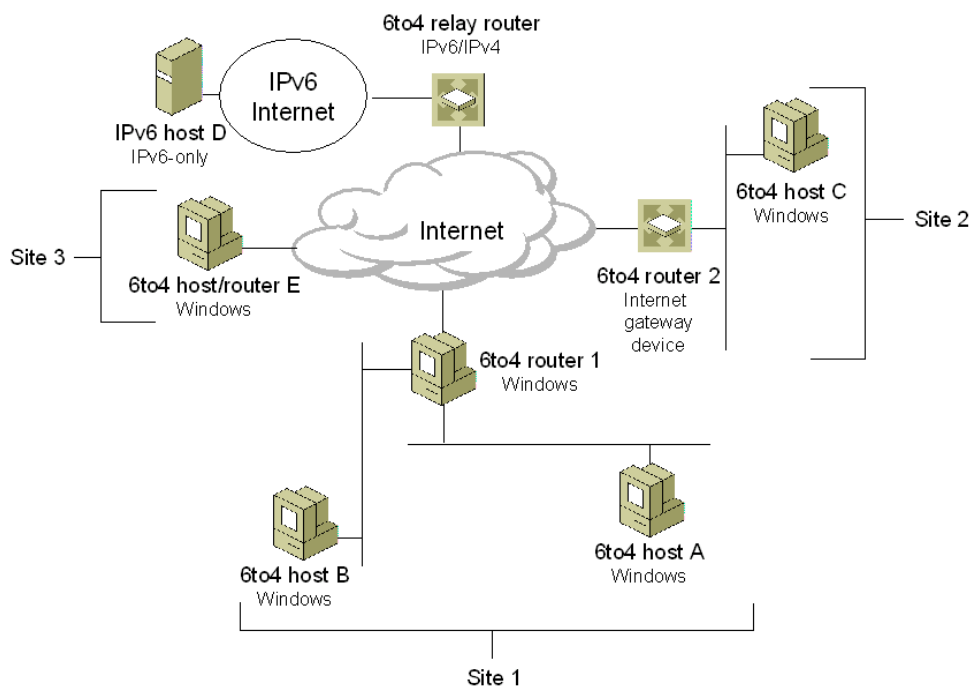


Figure 8: 6to4 for Windows Hosts

For a host that is assigned a private IPv4 address or receives a router advertisement for a global prefix, there are no 6to4 addresses assigned to the 6to4 Tunneling Pseudo-Interface. Addresses are autoconfigured based on the global prefix and a 64-bit global prefix route and default route are present in the routing table. This configuration corresponds to Host A, Host B, and Host C in Figure 8.

For a host that is assigned a public IPv4 address and does not receive a router advertisement for a global prefix, a 6to4 address of the form `2002:WWXX:YYZZ::WWXX:YYZZ` is automatically configured on the 6to4 Tunneling Pseudo-Interface. A `2002::/16` route using the 6to4 Tunneling Pseudo-Interface is added and, if the DNS query for the 6to4 relay router is successful, a default route using the 6to4 address of the 6to4 relay router as the next hop is added. This configuration corresponds to Host E in Figure 8, a host that is directly connected to the IPv4 Internet. In this case, the host is acting as its own site and its own 6to4 router.

The 6to4 component can also enable a computer running a member of the Windows Server 2003 family as a 6to4 router by utilizing the configuration of the Internet Connection Sharing (ICS) feature. This configuration corresponds to 6to4 router 1 and 6to4 router 2 in Figure 8.

If ICS is enabled on an interface that is assigned a public IPv4 address, the 6to4 component automatically:

- Enables IPv6 forwarding on both the public and private interfaces.

The public interface is connected to the Internet. The private interface is connected to a single-subnet intranet and uses private IPv4 addresses from the 192.168.0.0/24 prefix.

- Sends Router Advertisement messages on the private interface.

The router advertisements advertise the ICS computer as a default router and contain a global 6to4 address prefix that is based on the public IPv4 address assigned to the public interface. The SLA ID in the 6to4 address prefix is set to the interface index of the interface on which the advertisements are sent.

For example, for an ICS computer using the public IPv4 address of 131.107.23.89 and interface 5 as the interface index of the private interface, the advertised prefix would be 2002:836B:1759:5::/64. Private hosts receiving this router advertisement would create global addresses through normal address autoconfiguration and add a 2002:836B:1759:5::/64 route for the local subnet and a default route with a next-hop address of the link-local address of the ICS computer's private interface. Private hosts can communicate with each other on the same subnet using the 2002:836B:1759:5::/64 route. For all other destinations to other 6to4 sites or the IPv6 Internet, the IPv6 packets are forwarded to the ICS computer using the default route.

For traffic to other 6to4 sites, the ICS computer uses its 2002::/16 route and encapsulates the IPv6 traffic with an IPv4 header and sends it across the IPv4 Internet to another 6to4 router. For all other IPv6 traffic, the ICS computer uses its default route and encapsulates the IPv6 traffic with an IPv4 header and sends it across the IPv4 Internet to a 6to4 relay router.

To manually configure a 6to4 router, see [Manual Configuration for IPv6](http://www.microsoft.com/technet/columns/cableguy/cg0902.asp?frame=true) (<http://www.microsoft.com/technet/columns/cableguy/cg0902.asp?frame=true>).

#### **Note**

The 6to4 component is not performing network address translation on the IPv6 packets being forwarded. ICS is providing network address translation services on IPv4 packets being forwarded to and from private hosts. The 6to4 component uses the ICS configuration to determine the public IPv4 address and public interface.

---

## ISATAP

ISATAP is an address assignment and host-to-host, host-to-router, and router-to-host automatic tunneling technology that is used to provide unicast IPv6 connectivity between IPv6 hosts across an IPv4 intranet. ISATAP is described in the Internet draft titled "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)" (draft-ietf-ngtrans-isatap-0x.txt). ISATAP hosts do not require any manual configuration and create ISATAP addresses using standard address autoconfiguration mechanisms.

ISATAP can be used for communication between IPv6/IPv4 nodes on an IPv4 network. ISATAP addresses use the locally administered interface identifier `::0:5EFE:w.x.y.z` where:

- The `0:5EFE` portion is formed from the combination of an Organizational Unit Identifier (OUI) assigned to the Internet Assigned Numbers Authority (IANA) (`00-00-5E`), and a type that indicates an embedded IPv4 address (FE).
- The `w.x.y.z` portion is any unicast IPv4 address, which includes both public and private addresses.

The ISATAP interface identifier can be combined with any 64-bit prefix that is valid for IPv6 unicast addresses. This includes the link-local address prefix (`FE80::/64`), site-local prefixes, and global prefixes (including 6to4 prefixes).

Like IPv4-compatible addresses, 6over4 addresses, and 6to4 addresses, ISATAP addresses contain an embedded IPv4 address that is used to determine either the source or destination IPv4 addresses within the IPv4 header when ISATAP-addressed IPv6 traffic is tunneled across an IPv4 network.

By default, the IPv6 protocol for the Windows Server 2003 family automatically configures the link-local ISATAP address of `FE80::5EFE:w.x.y.z` on the Automatic Tunneling Pseudo-Interface (interface index 2) for each IPv4 address that is assigned to the node. This link-local ISATAP address allows two hosts to communicate over an IPv4 network by using each other's link-local ISATAP address.

For example, Host A is configured with the IPv4 address of `10.40.1.29` and Host B is configured with the IPv4 address of `192.168.41.30`. When the IPv6 protocol for the Windows Server 2003 family is started, Host A is automatically configured with the ISATAP address of `FE80::5EFE:10.40.1.29` and Host B is automatically configured with the ISATAP address of `FE80::5EFE:192.168.41.30`. This configuration is shown in Figure 9.

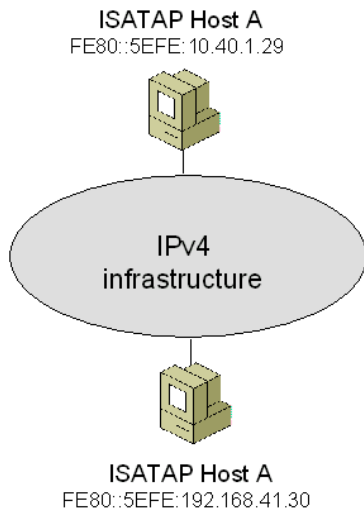


Figure 9: An Example ISATAP Configuration

When Host A sends IPv6 traffic to Host B by using Host B's link-local ISATAP address, the source and destination addresses for the IPv6 and IPv4 headers are as listed in the Table 2.

Table 2 Example Link-Local ISATAP Addresses

Field	Value
IPv6 Source Address	FE80::5EFE:10.40.1.29
IPv6 Destination Address	FE80::5EFE:192.168.41.30
IPv4 Source Address	10.40.1.29
IPv4 Destination Address	192.168.41.30

To test connectivity, use the ping command. For example, Host A would use the following command to ping Host B by using its link-local ISATAP address:

```
ping FE80::5EFE:192.168.41.30%2
```

Because the destination of the ping command is a link-local address, the *%ZoneID* portion of the command is used to specify the interface index of the interface from which traffic is sent. In this case, %2 specifies interface 2, which is the interface index assigned to the Automatic Tunneling Pseudo-Interface on Host A. The Automatic Tunneling Pseudo-Interface uses the link-local ISATAP address assigned to the interface as a source, and uses the last 32 bits in the source and destination IPv6 addresses (corresponding to the embedded IPv4 addresses) as the source and destination IPv4 addresses.

### Using an ISATAP Router

The use of link-local ISATAP addresses allows IPv6/IPv4 hosts on the same logical subnet (an IPv4 network) to communicate with each other, but not with other IPv6 addresses on other subnets. To communicate outside the logical subnet using ISATAP-derived global or site-local addresses, IPv6 hosts using ISATAP addresses must tunnel their packets to an ISATAP router. This configuration is shown in Figure 10.

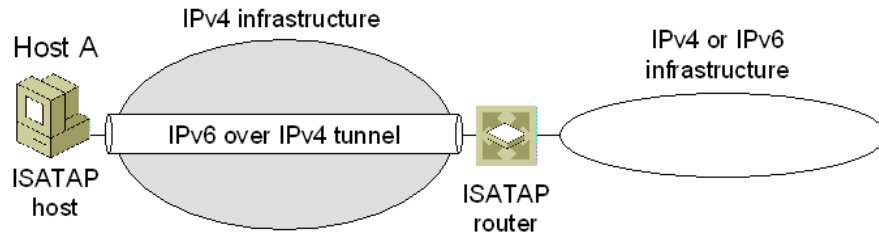


Figure 10: An ISATAP Router

An ISATAP router is an IPv6 router that performs the following:

- Forwards packets between ISATAP hosts on a logical subnet (an IPv4 network) and hosts on other subnets.

The other subnets can be other IPv4 networks (such as a portion of an organization network or the IPv4 Internet) or subnets in a native IPv6 routing domain (such as an organization's IPv6 network or the IPv6 Internet).

- Acts as a default router for ISATAP hosts.
- Advertises address prefixes to identify the logical subnet on which ISATAP hosts are located. ISATAP hosts use the advertised address prefixes to configure site-local and global ISATAP addresses.

When an ISATAP host receives a router advertisement from an ISATAP router, a default route (::/0) is added using the Automatic Tunneling Pseudo-Interface with next-hop address set to the link-local ISATAP address that corresponds to the logical subnet interface of the ISATAP router. When packets destined to locations outside the logical subnet are sent, they are tunneled to the IPv4 address of the ISATAP router corresponding to the ISATAP router's interface on the logical subnet defined by the IPv4 network containing the ISATAP router and ISATAP host. The ISATAP router then forwards the IPv6 packet.

For the IPv6 protocol for the Windows Server 2003 family, the configuration of the intranet IPv4 address of the ISATAP router is obtained through one of the following:

- The successful resolution of the name ISATAP to an IPv4 address.
- The **netsh interface ipv6 isatap set router** command.

### Resolving the ISATAP Name

When the IPv6 protocol for the Windows Server 2003 family starts, it attempts to resolve the name "ISATAP" to an IPv4 address using normal TCP/IP name resolution techniques that include the following:

1. Checking the local host name.
2. Checking the Hosts file in the *SystemRoot\system32\drivers\etc* folder.
3. Using ISATAP to form a fully qualified domain name and sending a DNS name query. For example, if the computer running a member of the Windows Server 2003 family is a member of the example.microsoft.com domain (and example.microsoft.com is the only domain name in the search list), the computer sends a DNS query to resolve the name ISATAP.example.microsoft.com.
4. Converting the ISATAP name into the NetBIOS name "ISATAP <00>" and checking the

NetBIOS name cache.

5. Sending a NetBIOS name query to a configured Windows Internet Name Service (WINS) server.
6. Sending NetBIOS broadcasts.
7. Checking the Lmhosts file in the *SystemRoot\system32\drivers\etc* folder.

If successful, the host sends an IPv4-encapsulated Router Solicitation message to the ISATAP router. The ISATAP router responds with an IPv4-encapsulated unicast Router Advertisement message containing prefixes to use for autoconfiguration of ISATAP-based addresses and, optionally, advertising itself as a default router.

To ensure that at least one of these attempts is successful, you can do one of the following:

- If the ISATAP router is a computer running a member of the Windows Server 2003 family, name the computer ISATAP and it will automatically register the appropriate records in DNS and WINS.
- Manually create an ISATAP address (A) record in the appropriate domain in DNS. For example, for the example.microsoft.com domain, create an A record for ISATAP.example.microsoft.com.
- Manually create a static WINS record in WINS for the NetBIOS name "ISATAP <00>".
- Add the following entry to the Hosts file of the computers that need to resolve the name ISATAP:

```
IPv4Address ISATAP
```

- Add the following entry to the Lmhosts file of the computers that need to resolve the name ISATAP:

```
IPv4Address ISATAP
```

### Resolving the \_ISATAP Name for Windows XP

When the IPv6 protocol for Windows XP starts, it attempts to resolve the name "\_ISATAP", rather than "ISATAP". To ensure that a computer running Windows XP can resolve the name \_ISATAP, you can do one of the following:

- Manually create \_ISATAP canonical name (CNAME) records in the appropriate domains in DNS. A CNAME record maps a name that is an alias to another name. For example, assuming that an A record already exists for the name ISATAP.example.microsoft.com, create a CNAME record that maps \_ISATAP.example.microsoft.com to ISATAP.example.microsoft.com.
- Manually create a static WINS record in WINS for the NetBIOS name "\_ISATAP <00>".
- Add the following entry to the Hosts file of the computers running Windows XP:

```
IPv4Address _ISATAP
```

- Add the following entry to the Lmhosts file of the computers running Windows XP:

```
IPv4Address _ISATAP
```

#### Note

Windows XP with SP1 attempts to resolve the name "ISATAP" to determine the IPv4 address of the ISATAP router. The methods described here are not needed if all your computers are running either a member of the Windows Server 2003 family or Windows XP with SP1.

### Using the netsh interface ipv6 isatap set router Command

Although the automatic resolution of the ISATAP name is the recommended method for configuring the IPv4 address of the ISATAP router, you can perform manual configuration with the **netsh interface ipv6 isatap set router** command. The syntax of this command is:

```
netsh interface ipv6 isatap set router AddressOrName
```

where *AddressOrName* is name or IPv4 address of the ISATAP router's intranet interface. For example, if the ISATAP router's IPv4 address is 192.168.39.1, the command is:

```
netsh interface ipv6 isatap set router 192.168.39.1
```

Once configured, the host sends an IPv4-encapsulated Router Solicitation message to the ISATAP router. The ISATAP router responds with an IPv4-encapsulated unicast Router Advertisement message containing prefixes to use for autoconfiguration of ISATAP-based addresses. This additional configuration is only needed when there is no IPv6 router on the host's subnet.

### Setting up an ISATAP Router

A computer running the IPv6 protocol for the Windows Server 2003 Family can be configured as an ISATAP router. Assuming that the router is already configured to forward IPv6 traffic on its LAN interfaces and has a default route that is configured to be published, the additional commands that need to be issued on the router are:

```
netsh interface ipv6 set interface 2 forwarding=enabled advertise=enabled
```

```
netsh interface ipv6 add route Address/PrefixLength 2 publish=yes validlifetime=ValidLife  
preferredlifetime=PreferredLife
```

The first command enables forwarding and advertising on interface index 2, the interface index assigned to the Automatic Tunneling Pseudo-Interface. The Automatic Tunneling Pseudo-Interface is the interface on which Router Solicitation messages and traffic to be forwarded is received.

The second command enables the advertisement of a specific prefix (*Address/PrefixLength*) with specified valid and preferred lifetimes (*ValidLife* and *PreferredLife*) over the Automatic Tunneling Pseudo-Interface. Use this command one or multiple times to advertise as many prefixes as required. All the prefixes configured using this command are included in the Router Advertisement message sent back to the ISATAP host.

If the router is not named ISATAP or the name ISATAP is not resolved to the IPv4 address of the router's intranet interface, you also need to issue the following command on the router:

```
netsh interface ipv6 isatap set router AddressOrName
```

in which *AddressOrName* is either the IPv4 address of the router's intranet interface or the name of the router that resolves to the IPv4 address of the router's intranet interface.

### ISATAP and 6to4 Example

Figure 11 shows two ISATAP hosts using 6to4 prefixes that are communicating across the Internet even though each site is using the 192.168.0.0/16 private address space internally.

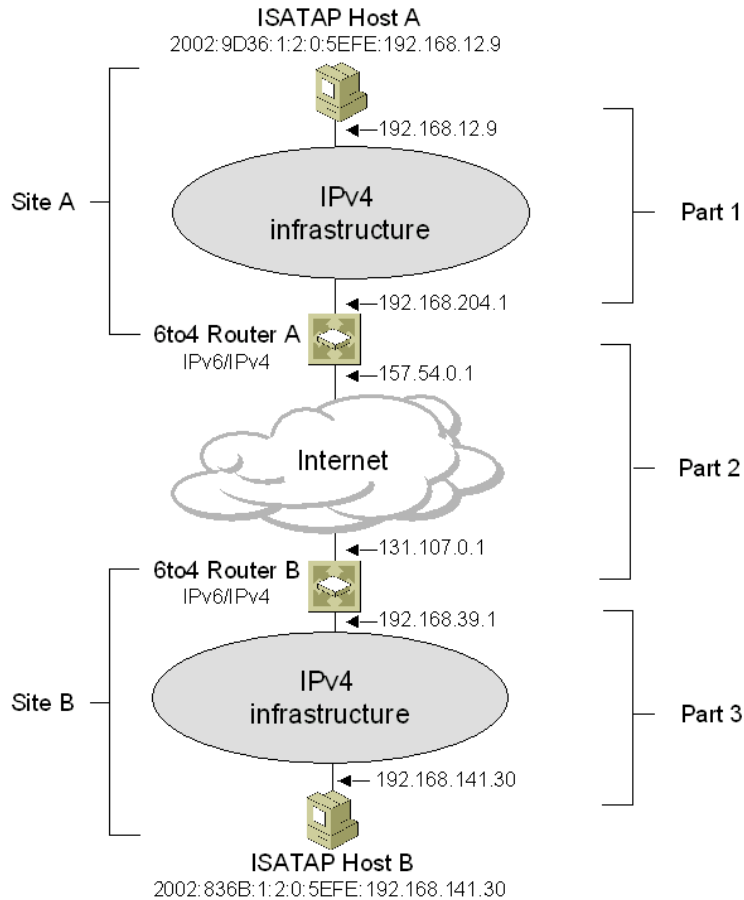


Figure 11: Communication Between ISATAP Hosts in Different 6to4 Sites

In this configuration:

- ISATAP Host A automatically configures a link-local ISATAP address of FE80::5EFE:192.168.12.9 on its Automatic Tunneling Pseudo-Interface.
- 6to4 Router A automatically configures a link-local ISATAP address of FE80::5EFE:192.168.204.1 on its Automatic Tunneling Pseudo-Interface.
- 6to4 Router B automatically configures a link-local ISATAP address of FE80::5EFE:192.168.39.1 on its Automatic Tunneling Pseudo-Interface.
- ISATAP Host B automatically configures a link-local ISATAP address of FE80::5EFE:192.168.141.30 on its Automatic Tunneling Pseudo-Interface.

ISATAP Host A can reach 6to4 Router A and all other hosts within Site A using link-local ISATAP addresses. However, ISATAP Host A cannot reach any addresses outside Site A. As a 6to4 router, 6to4 Router A constructs the global prefix 2002:9D36:1:5::/64 (9D36:1 is the colon hexadecimal notation for 157.54.0.1 and 5 is the interface index of 6to4 Router A's intranet interface) and advertises it using a router advertisement on its intranet interface. However, ISATAP Host A is not on 6to4 Router A's subnet and will never create a global address based on this 6to4 prefix.

To configure ISATAP Host A to receive the router advertisement from 6to4 Router A, the network administrator for Site A has configured 6to4 Router A as an ISATAP router and added an A record to Site A's DNS infrastructure so that the name ISATAP is resolved to the IPv4 address of 192.168.204.1. Upon startup, the IPv6 protocol on Host A resolves the ISATAP name and sends a Router Solicitation message to the addresses listed in Table 3.

**Table 3 Addresses in the Router Solicitation Message**

Field	Value
IPv6 Source Address	FE80::5EFE:192.168.12.9
IPv6 Destination Address	FF02::2
IPv4 Source Address	192.168.12.9
IPv4 Destination Address	192.168.204.1

Upon receipt of the Router Solicitation message from ISATAP Host A, 6to4 Router A sends back a unicast Router Advertisement message advertising 6to4 Router A as a default router and with a Prefix Information option to automatically configure IPv6 addresses using the prefix 2002:9D36:1:2::/64 (9D36:1 is the colon hexadecimal notation for 157.54.0.1 and 2 is the interface index of 6to4 Router A's Automatic Tunneling Pseudo-Interface).

The Router Advertisement is sent to the addresses listed in Table 4.

**Table 4 Addresses in the Router Advertisement Message**

Field	Value
IPv6 Source Address	FE80::5EFE:192.168.204.1
IPv6 Destination Address	FE80::5EFE:192.168.12.9
IPv4 Source Address	192.168.204.1
IPv4 Destination Address	192.168.12.9

Upon receipt of the Router Advertisement message, ISATAP Host A autoconfigures the address 2002:9D36:1:2:0:5EFE:192.168.12.9, a default route (::/0) using the Automatic Tunneling Pseudo-Interface (interface index 2) with the next-hop address of FE80::5EFE:192.168.204.1, and a 2002:9D36:1:2::/64 route using the Automatic Tunneling Pseudo-Interface.

Similarly, 6to4 Router B is configured as an ISATAP router and Site B has an appropriate A record in its DNS infrastructure so that ISATAP Host B autoconfigures the address 2002:836B:1:2:0:5EFE:192.168.141.30 (836B:1 is the colon hexadecimal notation for 131.107.0.1), a default route (::/0) using the Automatic Tunneling Pseudo-Interface (interface index 2) with the next-hop address of FE80::5EFE:192.168.39.1, and a 2002:836B:1:2::/64 route using the Automatic Tunneling Pseudo-Interface.

ISATAP Host A can now send a packet to ISATAP B. Let's examine the packet addressing in three parts (as shown in Figure 11) during its trip from ISATAP Host A to ISATAP Host B.

#### **Part 1: From ISATAP Host A to 6to4 Router A**

When ISATAP Host A sends the IPv6 packet, it sends it with the ::/0 route that uses the Automatic Tunneling Pseudo-Interface to the next-hop address of FE80::5EFE:192.168.204.1. By using this route, the next-hop address for this packet is set to the link-local ISATAP address of 6to4 Router A (FE80::5EFE:192.168.204.1).

Using the Automatic Tunneling Pseudo-Interface, the packet is tunneled using IPv4 from the IPv4 address assigned to its intranet interface (192.168.12.9) to the embedded IPv4 address in the ISATAP interface ID of the next-hop address (192.168.204.1). The resulting addresses are listed in Table 5.

**Table 5 Addresses in Part 1**

Field	Value
IPv6 Source Address	2002:9D36:1:2:0:5EFE:192.168.12.9
IPv6 Destination Address	2002:836B:1:2:0:5EFE:192.168.141.30
IPv4 Source Address	192.168.12.9
IPv4 Destination Address	192.168.204.1

#### **Part 2: From 6to4 Router A to 6to4 Router B**

6to4 Router A receives the IPv4 packet and removes the IPv4 header. When 6to4 Router A forwards the IPv6 packet, it forwards it with the 2002::/16 route that uses the 6to4 Tunneling Pseudo-Interface. By using

this route, the next-hop address for this packet is set to the destination address (2002:836B:1:2:0:5EFE:192.168.141.30).

Using the 6to4 Tunneling Pseudo-Interface, the packet is tunneled using IPv4 from the IPv4 address assigned to its Internet interface (157.54.0.1) to the embedded IPv4 address in the 6to4 NLA ID (836B:1) of the next-hop address (131.107.0.1). The resulting addresses are listed in Table 6.

**Table 6 Addresses in Part 2**

Field	Value
IPv6 Source Address	2002:9D36:1:2:0:5EFE:192.168.12.9
IPv6 Destination Address	2002:836B:1:2:0:5EFE:192.168.141.30
IPv4 Source Address	157.54.0.1
IPv4 Destination Address	131.107.0.1

### Part 3: From 6to4 Router B to ISATAP Host B

6to4 Router B receives the IPv4 packet and removes the IPv4 header. When 6to4 Router B forwards the IPv6 packet, it forwards it with the 2002:836B:1:2::/64 route that uses its Automatic Tunneling Pseudo-Interface. By using this route, the next-hop address for this packet is set to the destination address (2002:836B:1:2:0:5EFE:192.168.141.30).

Because the Automatic Tunneling Pseudo-Interface is used to forward the packet, the packet is tunneled using IPv4 from the IPv4 address assigned to its intranet interface (192.168.39.1) to the embedded IPv4 address in the ISATAP interface ID of the next-hop IPv6 address (192.168.141.30). 6to4 Router B sets the addresses in the forwarded packet as listed in Table 7.

**Table 7 Addresses in Part 3**

Field	Value
IPv6 Source Address	2002:9D36:1:2:0:5EFE:192.168.12.9
IPv6 Destination Address	2002:836B:1:2:0:5EFE:192.168.141.30
IPv4 Source Address	192.168.39.1
IPv4 Destination Address	192.168.141.30

---

## PortProxy

To facilitate the communication between nodes or applications that cannot connect using a common Internet layer protocol (IPv4 or IPv6), the IPv6 protocol for the Windows Server 2003 family provides PortProxy, a component that allows the proxying of the following traffic:

- IPv4 to IPv4  
TCP traffic to an IPv4 address is proxied to TCP traffic to another IPv4 address.
- IPv4 to IPv6  
TCP traffic to an IPv4 address is proxied to TCP traffic to an IPv6 address.
- IPv6 to IPv6  
TCP traffic to an IPv6 address is proxied to TCP traffic to another IPv6 address.
- IPv6 to IPv4  
TCP traffic to an IPv6 address is proxied to TCP traffic to an IPv4 address.

The most interesting and useful proxying for IPv6/IPv4 coexistence and migration is from IPv4 to IPv6 and from IPv6 to IPv4. For coexistence and migration, PortProxy enables the following scenarios:

- An IPv4-only node can access an IPv6-only node.  
  
In the IPv4 DNS infrastructure of the IPv4-only node, the name of the IPv6-only node resolves to an IPv4 address assigned to an interface of the PortProxy computer. (This might require manual configuration of an A record in the DNS.) The PortProxy computer is configured to proxy IPv4 to IPv6. All TCP traffic sent by the IPv4-only node is proxied in a manner similar to Internet proxy servers: the IPv4-only node establishes a TCP connection with the PortProxy computer and the PortProxy computer establishes a separate TCP connection with the IPv6-only node. The TCP connection data is transferred between the IPv4-only node and the IPv6-only node by the PortProxy component.
- An IPv6-only node can access an IPv4-only node.  
  
In the IPv6 DNS infrastructure of the IPv6-only node, the name of the IPv4-only node resolves to an IPv6 address assigned to an interface of the PortProxy computer. (This might require manual configuration of AAAA records in the DNS.) The PortProxy computer is configured to proxy IPv6 to IPv4. TCP traffic sent by the IPv6-only node to the PortProxy computer is proxied to the IPv4-only node.
- An IPv6 node can access an IPv4-only service running on PortProxy computer.  
  
In the IPv6 DNS infrastructure of the IPv6-only node, the name of the IPv6/IPv4 node resolves to an IPv6 address assigned to an interface of the PortProxy computer. The PortProxy computer is configured to proxy from IPv6 to IPv4 on the PortProxy computer. TCP traffic sent by the IPv6 node to the PortProxy computer is proxied to an IPv4-only service or application running on the PortProxy computer.

The last scenario allows IPv6 nodes to access services running a server that has not yet been IPv6-enabled. For example, the Windows Server 2003 Family includes an IPv6-enabled Telnet client but not an

IPv6-enabled Telnet server. However, you can use PortProxy on the Telnet server computer to allow IPv6 nodes to access the Telnet service.

To configure the PortProxy component, use the **netsh interface portproxy add|set|delete v4tov4|v4tov6|v6tov4|v6tov6** commands. For example, to configure a computer running a member of the Windows Server 2003 family to IPv6-enable the Telnet service (using TCP port 23), use the **netsh interface portproxy add v6tov4 23** command.

Notice that the default DNS behavior of the Telnet server (an IPv6/IPv4 node) is to dynamically register both its IPv6 and IPv4 addresses in the DNS. The default behavior of a computer running a member of the Windows Server 2003 family is to query the DNS for all record types, preferring the use of IPv6 addresses to IPv4 addresses. When the Telnet client is a computer running a member of the Windows Server 2003 family, it attempts to connect using IPv6 first. With PortProxy properly configured on the Telnet server computer, the first attempt to connect using an IPv6 address of the Telnet server should be successful, without manual configuration of DNS records.

### Notes

The PortProxy component is only provided with the IPv6 protocol for the Windows Server 2003 family.

The PortProxy component works only for TCP traffic and for application-layer protocols that do not embed address or port information inside the upper-layer PDU. PortProxy has no facilities to check for and change embedded address or port information in upper layer PDUs that are being proxied. For example, PortProxy cannot be used to IPv6-enable the FTP server service because the FTP PORT command embeds IPv4 address information inside the FTP PDU.

---

## Migrating to IPv6

To be sure, the migration of IPv4 to IPv6 will be a long process and some details of migration have yet to be determined. As a general methodology, to migrate from IPv4 to IPv6, you must perform the following steps:

1. Upgrade your applications to be independent of IPv6 or IPv4.

Applications must be changed to use new Windows Sockets application programming interfaces (APIs) so that name resolution, socket creation, and other functions are independent of whether IPv4 or IPv6 is being used.

2. Update the DNS infrastructure to support IPv6 address and PTR records.

The DNS infrastructure might need to be upgraded to support the new AAAA records and PTR records in the IP6.INT reverse domain (optional).

3. Upgrade hosts to IPv6/IPv4 nodes.

Hosts must be upgraded to use a dual IP layer or dual IP stack. DNS resolver support must also be added to process DNS query results that contain both IPv4 and IPv6 addresses.

4. Upgrade routing infrastructure for native IPv6 routing.

Routers must be upgraded to support native IPv6 routing and IPv6 routing protocols.

5. Convert IPv6/IPv4 nodes to IPv6-only nodes.

IPv6/IPv4 nodes can be upgraded to be IPv6-only nodes. This should be a long-term goal because it will take years for all current IPv4-only network devices to be upgraded to IPv6-only. For those IPv4-only nodes that cannot be upgraded to IPv6/IPv4 or IPv6-only, employ translation gateways as appropriate so that IPv4-only nodes can communicate with IPv6-only nodes.

---

## Appendix A: IPv6 Automatic Tunneling

As defined in RFC 2893, IPv6 Automatic Tunneling is the tunneling that occurs when IPv4-compatible addresses (::w.x.y.z where w.x.y.z is a public IPv4 address) are used. IPv6 Automatic Tunneling is a host-to-host tunnel between two IPv6/IPv4 hosts using IPv4-compatible addresses.

For example, when Host1 (with the public IPv4 address of 157.60.91.123 and corresponding IPv4-compatible address of ::157.60.91.123) sends traffic to Host2 (with the IPv4 address of 131.107.210.49 and corresponding IPv4-compatible address of ::131.107.210.49), the addresses in the IPv4 and IPv6 headers are as listed in Table 8.

**Table 8 Example IPv6 Automatic Tunneling Addresses**

Field	Value
IPv6 Source Address	::157.60.91.123
IPv6 Destination Address	::131.107.210.49
IPv4 Source Address	157.60.91.123
IPv4 Destination Address	131.107.210.49

To test connectivity, use the ping command. For example, Host A would use the following command to ping Host B by using its IPv4-compatible address:

**ping ::131.107.210.49**

The IPv6 protocol for the Windows Server 2003 family does not use IPv4-compatible addresses by default. To enable IPv4-compatible addresses, use the **netsh interface ipv6 set state v4compat=enabled** command. When enabled for the IPv6 protocol for the Windows Server 2003 family, communication to IPv4-compatible addresses is facilitated by a ::/96 route in the IPv6 routing table that uses the Automatic Tunneling Pseudo-Interface (interface index 2). This route indicates that all addresses with the first 96 bits set to 0 are forwarded to their destination addresses using the Automatic Tunneling Pseudo-Interface. The Automatic Tunneling Pseudo-Interface uses the last 32 bits in the source and destination IPv6 addresses (corresponding to the embedded IPv4 addresses) as the source and destination IPv4 addresses for the outgoing IPv4 packet.

### Notes

In this article, the term "IPv6 Automatic Tunneling" refers to the use of IPv4-compatible addresses. The term "automatic tunneling" is tunneling that occurs without manual configuration independent of the type of addressing being used.

IPv4-compatible addresses are not widely used because they are only defined for public IPv4 addresses and their functionality has been replaced with ISATAP for IPv4 intranets and 6to4 for the IPv4 Internet. For more information, see "ISATAP" and "6to4" in this article.

## Appendix B: 6over4

6over4, also known as IPv4 multicast tunneling, is a host-to-host, host-to-router, and router-to-host automatic tunneling technology that is used to provide unicast and multicast IPv6 connectivity between IPv6 nodes across an IPv4 intranet. 6over4 is described in RFC 2529. 6over4 hosts use a valid 64-bit prefix for unicast addresses and the interface identifier `::WWXX:YYZZ`, where `WWXX:YYZZ` is the colon-hexadecimal representation of the IPv4 address (`w.x.y.z`) assigned to the host. By default, 6over4 hosts automatically configure the link-local address `FE80::WWXX:YYZZ` on each 6over4 interface.

6over4 treats an IPv4 infrastructure as a single link with multicast capabilities. This means that Neighbor Discovery processes (such as address resolution and router discovery) work as they do over a physical link with multicast capabilities. To emulate a multicast-capable link, the IPv4 infrastructure must be IPv4 multicast-enabled. Figure 12 shows a 6over4 configuration.

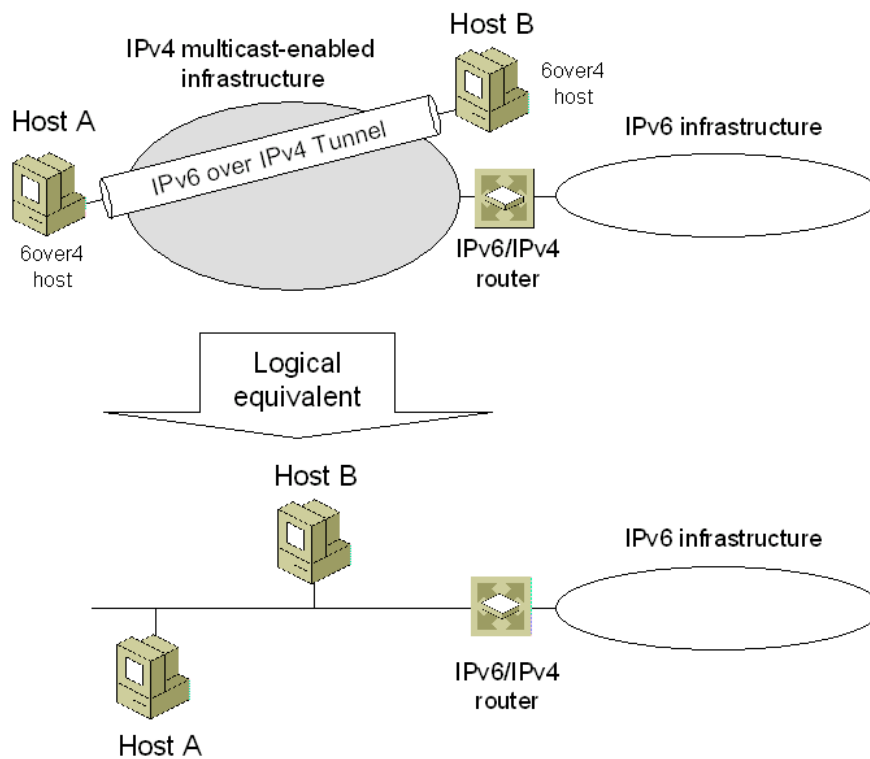


Figure 12: A 6over4 Configuration

To facilitate IPv6 multicast communications over an IPv4 multicast-enabled infrastructure, RFC 2529 defines the following mapping to translate an IPv6 multicast address to an IPv4 multicast address:

$$239.192.[\textit{second to last byte of IPv6 address}].[last byte of IPv6 address]$$

The following are example mappings for IPv6 multicast addresses:

- `FF02::1` (link-local scope all-hosts multicast address) is mapped to `239.192.0.1`
- `FF02::2` (link-local scope all-routers multicast address) is mapped to `239.192.0.2`

- FF02::1:FF28:9C5A (example solicited-node multicast address) is mapped to 239.192.156.90.

When 6over4 is enabled, the IPv4 layer uses Internet Group Membership Protocol (IGMP) messages to inform local IPv4 routers of their interest in receiving IPv4 multicast traffic that is sent to the mapped IPv4 multicast addresses. 6over4-enabled hosts also register additional multicast MAC addresses with their network adapters that correspond to the mapped IPv4 multicast addresses. For example, for an Ethernet adapter:

- The corresponding multicast MAC address for 239.192.0.1 is 01-00-5E-40-00-01.
- The corresponding multicast MAC address for 239.192.0.2 is 01-00-5E-40-00-02.
- The corresponding multicast MAC address for 239.192.156.90 is 01-00-5E-40-9C-5A.

Because the IPv4 infrastructure acts as a multicast capable link, hosts can use Neighbor Solicitation and Neighbor Advertisement messages to resolve each other's link-layer addresses. The 6over4 link-layer addresses are the tunnel endpoints. Hosts and routers can use Router Solicitation and Router Advertisement messages for router, prefix, and parameter discovery. To facilitate ND messages, RFC 2529 defines the format for the Source and Target Link-Layer Address options as shown in Figure 13.

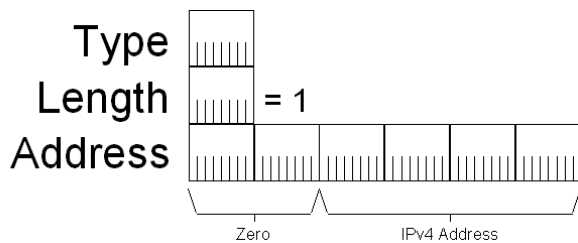


Figure 13: Source and Target Link-Layer Address options for 6over4

For example, when Host1 (with the public IPv4 address of 157.60.91.123 and corresponding link-local 6over4 address of FE80::9D3C:5B7B) sends traffic to Host2 (with the public IPv4 address of 131.107.210.49 and corresponding link-local 6over4 address of FE80::836B:D231), the addresses in the IPv4 and IPv6 headers are as listed in Table 9.

Table 9 Example 6over4 Addresses

Field	Value
IPv6 Source Address	FE80::9D3C:5B7B
IPv6 Destination Address	FE80::836B:D231
IPv4 Source Address	157.60.91.123
IPv4 Destination Address	131.107.210.49

The use of 6over4 for IPv6 protocol for the Windows Server 2003 family is disabled by default. To enable the use of 6over4, use the **netsh interface ipv6 set state 6over4=enabled**. This command creates a 6over4 tunneling interface for each IPv4 address assigned to the computer. If router advertisements are received over any of these interfaces (via the multicast mapping mechanism described earlier), appropriate addresses for this interface and routes using this interface are automatically configured.

Communication to 6over4 addresses is facilitated by routes and the 6over4 tunneling interface. For example, the interface index for the 6over4 tunneling interface of a host is 5 (the actual interface index for the 6over4 tunneling interface varies depending on the configuration of the computer). A router advertisement from a router with the link-local 6over4 address of FE80::C0A8:1501 is received. The router advertisement is advertising the router as a default router and contains the auto-configuration prefix FEC0:0:0:21A8::/64. The host configures a default route with the next-hop address of FE80::C0A8:1501 and a subnet route for prefix FEC0:0:0:21A8::/64 that uses interface index 5.

When packets are sent using the default or FEC0:0:0:21A8::/64 routes, the sending node uses the appropriate locally assigned 6over4 address as a source, and uses the last 32 bits in the source and destination IPv6 addresses (corresponding to the embedded IPv4 addresses) as the source and destination IPv4 addresses for the outgoing IPv4 packet.

A packet to a destination matching the prefix FEC0:0:0:21A8::/64 is sent to the next-hop address of the destination using the 6over4 tunneling interface. The 6over4 tunneling interface uses address resolution for the destination address to determine the source and destination link-layer addresses (and corresponding IPv4 addresses) to use when sending the IPv4-encapsulated IPv6 packet.

A packet to a destination matching the default route is sent to the next-hop address of FE80::C0A8:1501 using the 6over4 tunneling interface. The 6over4 tunneling interface uses address resolution for the next-hop address to determine the source and destination link-layer addresses (and corresponding IPv4 addresses) to use when sending the IPv4-encapsulated IPv6 packet.

To test connectivity using 6over4 addresses, use the ping command. Using the example in Table 9, Host A would use the following command to ping Host B by using its link-local 6over4 address:

```
ping FE80::836B:D231%5
```

Because the destination of the ping command is a link-local address, the *%ZoneID* portion of the command is used to specify the interface index of the interface from which traffic is sent. In this case, *%5* specifies interface 5, which is the interface index assigned to the 6over4 tunneling interface in this example.

## Notes

Because 6over4 requires an IPv4 multicast infrastructure to work properly, it is not widely used.

The difference between using ISATAP versus 6over4 on an IPv4 intranet is that 6over4 supports IPv6 multicast, and ISATAP currently does not.

When you enable 6over4 with the **netsh interface ipv6 set state 6over4=enabled** command, it creates a 6over4 tunneling interface with a globally unique identifier (GUID)-based name. To create a 6over4 tunneling interface with a friendlier name, use the **netsh interface ipv6 add 6over4tunnel** command instead of the **netsh interface ipv6 set state 6over4=enabled** command.

## Summary

Migrating to IPv6 involves the upgrading of applications, hosts, routers, and DNS to support IPv6, and then converting IPv6/IPv4 nodes to IPv6-only nodes. Because this migration might take years, IPv4/IPv6 nodes must be able to coexist over IPv4 infrastructures such as the Internet and private intranets. To provide automatic configuration and tunneling over the IPv4 Internet, the IPv6 protocol for the Windows Server 2003 family supports 6to4. To provide automatic configuration and tunneling over an IPv4 intranet, the IPv6 protocol for the Windows Server 2003 family supports ISATAP. 6to4 and ISATAP can be used together to provide IPv6 connectivity between hosts in different sites that only have IPv4 infrastructures across the IPv4 Internet. To provide support between IPv4-only to IPv6-only nodes and services, the IPv6 protocol for the Windows Server 2003 family supports PortProxy.

## Related Links

For the latest information about Microsoft's support for IPv6, see the [Microsoft Windows IPv6 Web site](http://www.microsoft.com/ipv6) at <http://www.microsoft.com/ipv6>.

For the latest information about Windows Server 2003, see the [Windows Server 2003 Web site](http://www.microsoft.com/windowsserver2003) at <http://www.microsoft.com/windowsserver2003>.

For the latest set of RFCs and Internet drafts describing IPv6/IPv4 coexistence and migration technologies, see the [Next Generation Transition \(ngtrans\) Working Group Web site](http://www.ietf.org/html.charters/ngtrans-charter.html) at <http://www.ietf.org/html.charters/ngtrans-charter.html>.