



# Windows Server® 2008

## IPv6 Transition Technologies

*Microsoft Corporation*

*Published: October 2003*

*Updated: February 2008*

---

### **Abstract**

The migration of Internet Protocol version 4 (IPv4) to Internet Protocol version 6 (IPv6) will not happen overnight. There will be a period of transition when both protocols are in use over the same infrastructure. To address this transition period, the designers of IPv6 have created technologies and address types so that IPv6 nodes can communicate with each other in a mixed environment, even if they are separated by an IPv4-only infrastructure. This white paper describes the IPv6 transition technologies that are supported by the IPv6 protocol for Microsoft® Windows Server® 2008 and Windows Vista™. This white paper is intended for network engineers and support professionals who are already familiar with basic networking concepts, TCP/IP, and IPv6.

**Microsoft**

*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 White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, 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.*

*Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.*

*© 2008 Microsoft Corporation. All rights reserved.*

*Microsoft, Windows, Windows Server, and Windows Vista 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
<b>Transition Mechanisms</b> .....	<b>3</b>
Using Both IPv4 and IPv6 .....	3
Dual IP Layer Architecture .....	3
Dual Stack Architecture.....	4
DNS Infrastructure .....	6
Address Records.....	6
Pointer Records.....	6
Address Selection Rules .....	6
IPv6 over IPv4 Tunneling.....	6
<b>Tunneling Configurations</b> .....	<b>8</b>
Router-to-Router .....	8
Host-to-Router and Router-to-Host.....	8
Host-to-Host.....	9
Types of Tunnels.....	10
Configured Tunnels .....	10
Automatic Tunnels.....	10
<b>ISATAP</b> .....	<b>12</b>
ISATAP Components.....	13
Obtaining an ISATAP Prefix.....	14
Resolving the ISATAP Name .....	14
Using the netsh interface isatap set router Command.....	16
ISATAP Addressing Example .....	16
ISATAP Routing.....	17
Configuring an ISATAP Router .....	18
ISATAP Communication Examples .....	18
ISATAP Host to ISATAP Host.....	18
ISATAP Host to IPv6 Host.....	19
<b>6to4</b> .....	<b>22</b>

6to4 Components .....	22
6to4 Addressing Example .....	23
6to4 Routing .....	24
6to4 Support in Windows Server 2008 and Windows Vista .....	25
6to4 Host/router Support .....	25
6to4 Router Support .....	26
6to4 Communication Examples .....	26
6to4 Host to 6to4 Host/router .....	26
6to4 Host to IPv6 Host .....	28
<b>Teredo .....</b>	<b>31</b>
Teredo Components .....	31
Teredo Address Format .....	33
Teredo Addressing Example .....	34
Teredo Routing .....	35
How Teredo Works .....	36
Initial Configuration .....	36
Initial Communication Between Two Teredo Clients in Different Sites .....	37
<b>PortProxy .....</b>	<b>39</b>
<b>Migrating to IPv6 .....</b>	<b>41</b>
<b>Summary .....</b>	<b>42</b>
<b>Related Links .....</b>	<b>43</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 from IPv4 to IPv6 is 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 IPv6 transition technologies is the same for Windows Server 2008, Windows Vista, Windows Server 2003, and Windows XP with Service Pack 1 (SP1) and later.

## Node Types

RFC 2893 defines the following node types:

- IPv4-only node  
A node that implements only IPv4 (and has only IPv4 addresses) and 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) and does not support IPv4. This node is only able to communicate with IPv6 nodes and applications. This type of node is not common today, but might become more prevalent as smaller devices such as cellular phones and handheld computing devices include the IPv6 protocol.
- IPv6/IPv4 node

A node that implements both IPv4 and IPv6.

- IPv4 node

A node that implements IPv4. An IPv4 node can be an IPv4-only node or an IPv6/IPv4 node.

- IPv6 node

A node that implements IPv6. 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.

---

## Transition Mechanisms

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

- Using both IPv4 and IPv6
- IPv6 over IPv4 tunneling
- DNS infrastructure

### Using Both IPv4 and IPv6

During the time that the routing infrastructure is being transitioned from IPv4-only, to IPv4 and IPv6, and finally to IPv6-only, hosts must be able to reach destinations using either IPv4 or IPv6. For example, during the transition, some server services will be reachable over IPv6. However, some services, which have not yet been updated to support both IPv4 and IPv6, are only reachable over IPv4. Therefore, hosts must be able to use both IPv4 and IPv6. To use both IPv4 and IPv6 Internet layers on the same host, IPv6/IPv4 hosts can have the following architectures:

- Dual IP layer architecture
- Dual stack architecture

#### Dual IP Layer Architecture

A dual IP layer architecture contains both IPv4 and IPv6 Internet layers with a single implementation of Transport layer protocols such as TCP and UDP. Figure 1 shows a dual IP layer architecture.

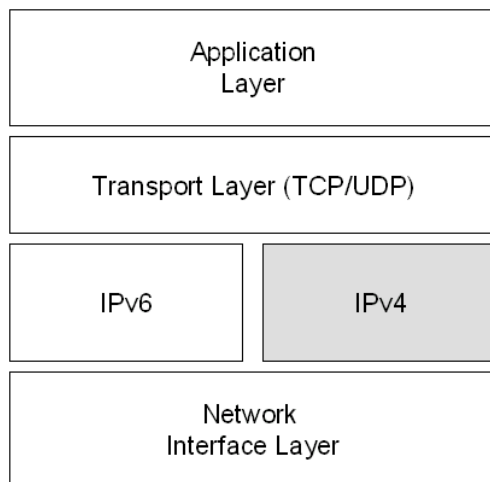


Figure 1: A Dual IP Layer Architecture

The Next Generation TCP/IP stack in Windows Server 2008 and Windows Vista is a new implementation of the TCP/IP protocol suite that includes both IPv4 and IPv6 in a dual IP layer architecture as shown in Figure 1. For more information, see [Next Generation TCP/IP Stack in Windows Server 2008 and Windows Vista](#) at <http://www.microsoft.com/technet/community/columns/cableguy/cg0905.mspx>.

With a single protocol stack that contains both IPv4 and IPv6, a host running Windows Server 2008 or Windows Vista can create the following types of packets:

- IPv4 packets
- IPv6 packets
- IPv6 over IPv4 packets

These are IPv6 packets that are encapsulated with an IPv4 header. For more information, see "IPv6 over IPv4 Tunneling" in this white paper.

Figure 2 shows the types of communication with a dual IP layer architecture.

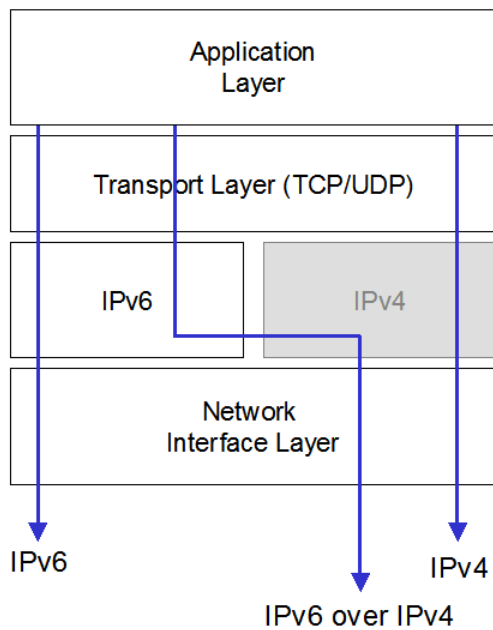


Figure 2: Communication Types with a Dual IP Layer Architecture

### Dual Stack Architecture

A dual stack architecture contains both IPv4 and IPv6 Internet layers with separate protocol stacks containing separate implementations of Transport layer protocols such as TCP and UDP. Figure 3 shows a dual stack architecture.

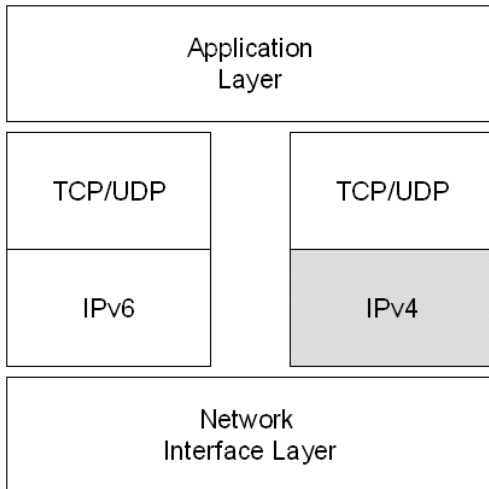


Figure 3: A Dual Stack Architecture

The IPv6 protocol for Windows Server 2003 and Windows XP is a dual stack architecture. The IPv6 protocol driver, Tcpi6.sys, contains a separate implementation of TCP and UDP.

With both IPv4 and IPv6 protocol stacks, a host running Windows Server 2003 or Windows XP can create the following types of packets:

- IPv4 packets
- IPv6 packets
- IPv6 over IPv4 packets

Figure 4 shows the types of communication with a dual stack architecture.

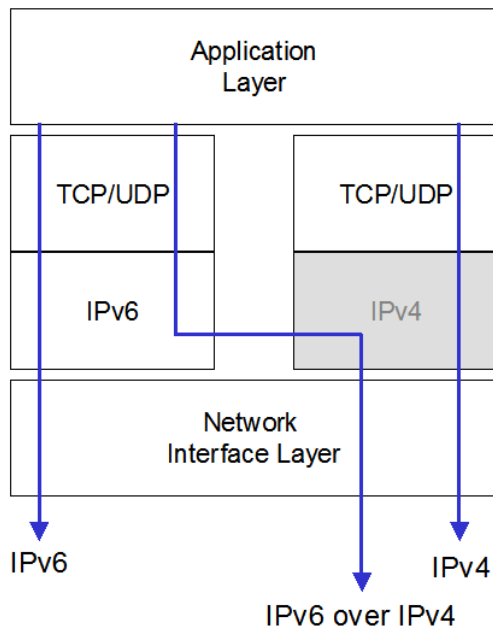


Figure 4: Communication Types with a Dual Stack Architecture

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

## 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 with DNS dynamic update) for the successful resolution of domain names to addresses:

- A records for IPv4 nodes
- AAAA records for IPv6 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 nodes
- PTR records in the IP6.ARPA domain for IPv6 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 typically 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 might contain both IPv4 and IPv6 addresses. The querying host is configured with at least one IPv4 address and (typically) multiple IPv6 addresses. The host must decide which type of address (IPv4 vs. IPv6) and the scope of the address for the source and the destination addresses when initiating communication. The host must use a set of address selection rules. Default address selection rules are described in RFC 3484. For more information, see [Source and Destination Address Selection for IPv6](http://www.microsoft.com/technet/community/columns/cableguy/cg0206.msp) at <http://www.microsoft.com/technet/community/columns/cableguy/cg0206.msp>.

You can view the default address selection rules for the IPv6 protocol for Windows 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 name query responses are preferred over IPv4 addresses.

## 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 next-hop address of the matching route for the destination and the tunneling interface.

Figure 5 shows IPv6 over IPv4 tunneling.

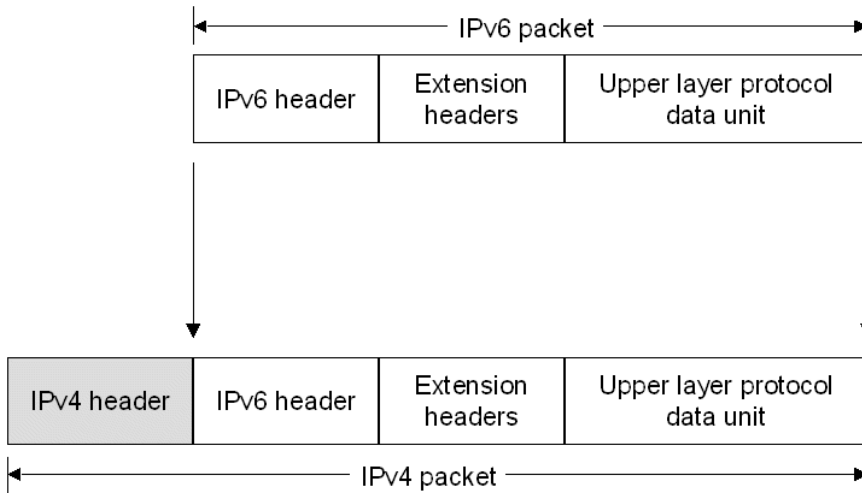


Figure 5: IPv6 over IPv4 Tunneling

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.

## 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 IPv6-capable 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 6 shows router-to-router tunneling.

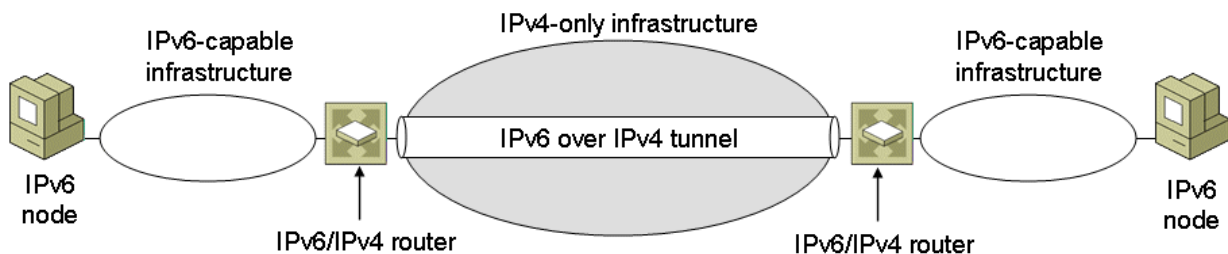


Figure 6: Router-to-Router Tunneling

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. For more information about 6to4, see "6to4" in this white paper.

### 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 destination address of the IPv6/IPv4 node.

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.

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 7 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.

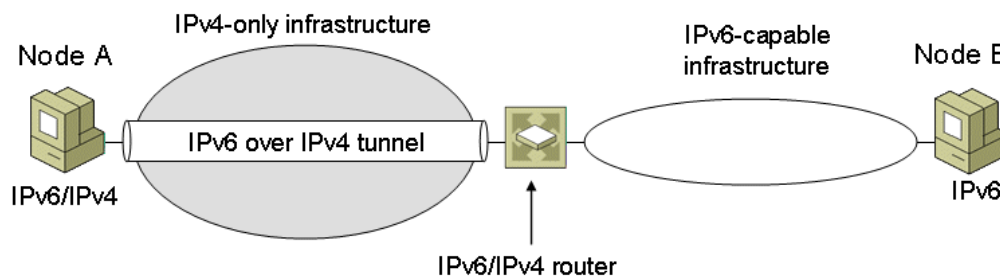


Figure 7: Host-to-Router and Router-to-Host Tunneling

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 Intra-site Automatic Tunnel Addressing Protocol (ISATAP) host that tunnels across an IPv4 network to an ISATAP router to reach the IPv6 Internet, another IPv4 network, or an IPv6-capable network. For more information about ISATAP, see "ISATAP" in this white paper.
- 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 8 shows host-to-host tunneling.

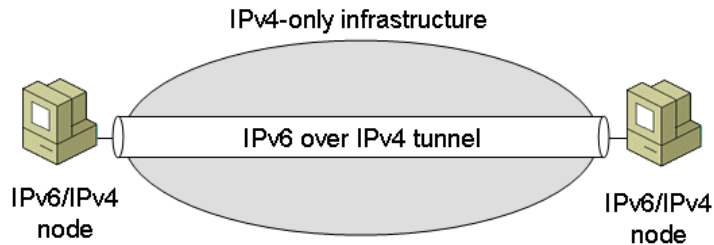


Figure 8: Host-to-Host Tunneling

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 next-hop address when sending or forwarding the packet.

Router-to-router tunneling configurations can be 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 Windows, use the **netsh interface ipv6 add v6v4tunnel** command.

### Automatic Tunnels

An automatic tunnel is a tunnel that does not require manual configuration. Tunnel endpoints for automatic tunnels are determined by the use of routes, next-hop addresses based on destination IPv6 addresses, and logical tunnel interfaces. The IPv6 protocol for Windows Server 2008 and Windows Vista supports the following automatic tunneling technologies:

- ISATAP

Used for unicast communication across an IPv4 intranet and is enabled by default. For more information, see "ISATAP" in this white paper.

- 6to4

Used for unicast communication across the IPv4 Internet and is enabled by default. For more information, see "6to4" in this white paper.

- Teredo

Used for unicast communication across the IPv4 Internet over network address translators (NATs). Teredo support is included and is disabled by default. Teredo support is included with Windows Server 2008, Windows Server 2003 Service Pack 1 and later, Windows XP with SP2 and later, and Windows XP with SP1 and the Advanced Networking Pack for Windows XP, and is disabled by default. Teredo support is also included with Windows Vista and is enabled but inactive by default. For more information, see "Teredo" in this white paper.

## 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/IPv4 hosts across an IPv4 intranet. ISATAP is described in RFC 4214. ISATAP hosts do not require any manual configuration and can create ISATAP addresses using standard address autoconfiguration mechanisms.

ISATAP addresses use the locally administered interface identifier `::0:5EFE:w.x.y.z`, in which `w.x.y.z` is a private unicast IPv4 address, or `::200:5EFE:w.x.y.z`, in which `w.x.y.z` is a public unicast IPv4 address. An ISATAP interface identifier can be combined with any 64-bit prefix that is valid for IPv6 unicast addresses, including link-local (`FE80::/64`), unique local, and global prefixes. The interface identifier portion of an ISATAP address contains an embedded IPv4 address that is used to determine the destination IPv4 address for the IPv4 header when ISATAP-addressed IPv6 traffic is tunneled across an IPv4 network.

By default, the IPv6 protocol for Windows Vista with no service packs installed, Windows Server 2003, and Windows XP automatically configures the link-local ISATAP address of `FE80::5EFE:w.x.y.z` or `FE80::200:5EFE:w.x.y.z` on an ISATAP tunneling interface for each IPv4 address that is assigned to the node. These link-local ISATAP addresses allow two hosts to communicate over an IPv4-only network. The IPv6 protocol for Windows Server 2008 and Windows Vista with Service Pack 1 does not automatically configure link-local ISATAP addresses unless the name "ISATAP" can be resolved.

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 Windows Vista with no service packs installed 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`. Figure 9 shows this configuration.

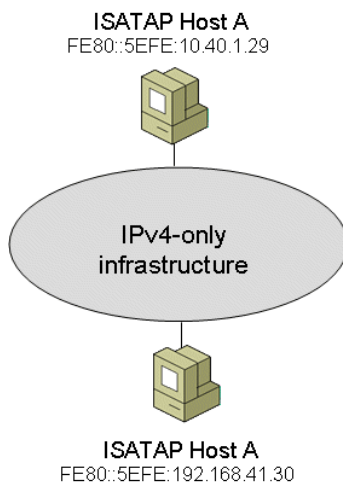


Figure 9: An Example ISATAP Configuration

When Host A sends IPv6 traffic to Host B 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 1.

**Table 1 Example IPv4 and IPv6 Addresses for Link-Local ISATAP Connectivity**

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 between ISATAP hosts, you can use the Ping tool. For example, the user on 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%7**

Because the destination of the ping command is a link-local address, you must use the *%ZoneID* as part of the destination address to specify the interface index of the interface from which traffic is sent. In this case, %7 specifies interface 7, which is the interface index assigned to the ISATAP tunneling interface on Host A. The ISATAP tunneling interface uses the link-local ISATAP address assigned to the interface as a source, and the last 32 bits in the destination IPv6 address (corresponding to the embedded IPv4 address) as the destination IPv4 address. For the source IPv4 address, IPv4 on Host A determines the best source IPv4 address to use to reach the destination IPv4 address of 192.168.41.30. In this case, Host A only has a single IPv4 address assigned, so IPv4 on Host A uses the source address of 10.40.1.29.

---

**Note** In Windows Server 2003 and Windows XP, the ISATAP tunneling interface is named “Automatic Tunneling Pseudo-Interface” and is typically assigned the interface index of 2.

---

## ISATAP Components

Figure 10 shows the components of an intranet that is using ISATAP.

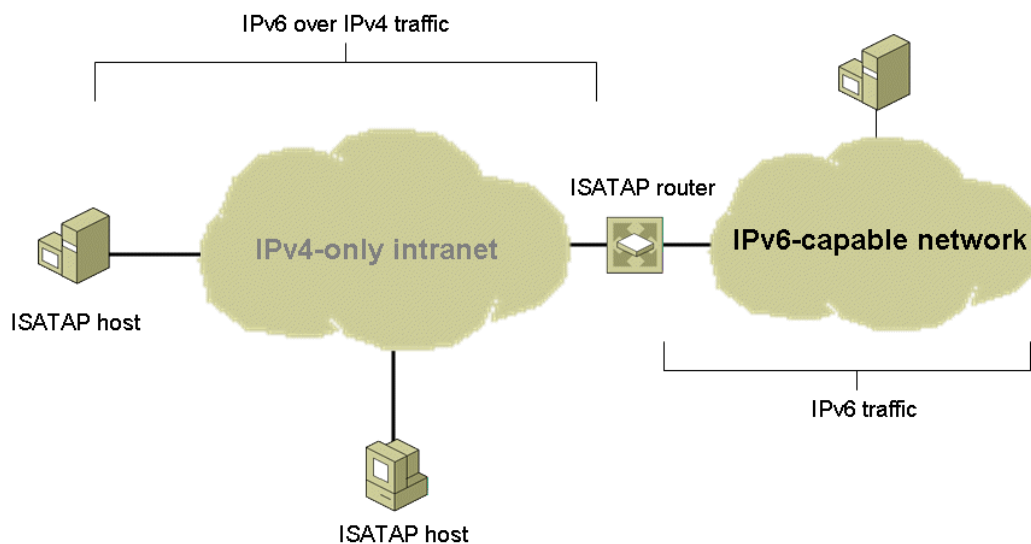


Figure 10: Components of ISATAP

ISATAP hosts have an ISATAP tunneling interface and perform their own tunneling to either other ISATAP hosts or an ISATAP router. Link-local ISATAP addresses allows ISATAP hosts on the same logical subnet (an IPv4-only intranet) to communicate with each other, but not with other IPv6 hosts on other IPv6 subnets. To communicate beyond the logical ISATAP subnet using ISATAP-based global addresses, ISATAP hosts must tunnel their packets to an ISATAP router.

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

- Advertises address prefixes to identify the logical ISATAP subnet on which ISATAP hosts are located. ISATAP hosts use the advertised address prefixes to configure unique local or global ISATAP addresses.
- Forwards packets between ISATAP hosts on the logical ISATAP subnet and IPv6 hosts on other subnets.

The other subnets can be other IPv4 networks (such as a portion of an intranet 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.

When an ISATAP host receives a suitable router advertisement from an ISATAP router, the ISATAP host adds a default route (::/0) using its ISATAP tunneling interface with next-hop address set to the link-local ISATAP address of the ISATAP router. When ISATAP hosts send packets destined to locations outside the logical ISATAP subnet, the packets are tunneled to the IPv4 address of the ISATAP router corresponding to the ISATAP router's interface on the IPv4-only network. The ISATAP router then forwards the IPv6 packet.

---

**Note** The existence of an IPv6-capable network is optional, in which case the ISATAP router is only functioning as an advertising router.

---

## Obtaining an ISATAP Prefix

For the IPv6 protocol for Windows Server 2008 and Windows Vista, the IPv4 address of the ISATAP router is obtained through one of the following methods:

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

---

**Note** For Windows Server 2003 and Windows XP, use the **netsh interface ipv6 isatap set router** command.

---

## Resolving the ISATAP Name

When the IPv6 protocol for Windows Server 2008 and Windows Vista starts, it attempts to resolve the name "ISATAP" to an IPv4 address using normal TCP/IP name resolution techniques. 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. This process is shown in Figure 11.

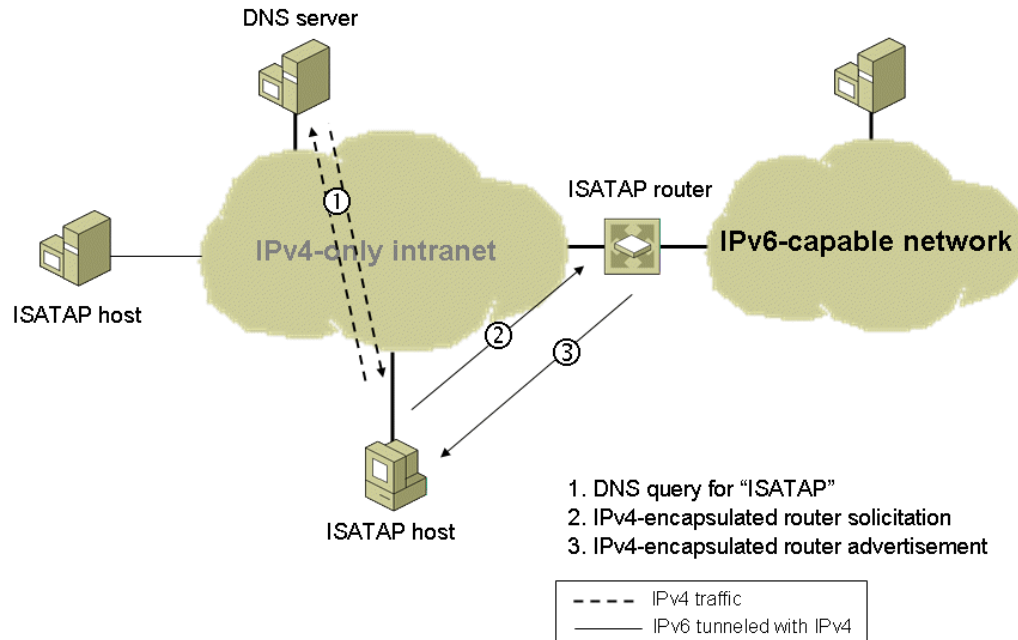


Figure 11: Obtaining an ISATAP Prefix

Normal TCP/IP name resolution techniques for resolving the name "ISATAP" include the following:

1. Checking the local host name.
2. Checking the DNS client resolver cache, which includes the entries in the Hosts file in the `SystemRoot\system32\drivers\etc` folder.
3. Forming a fully qualified domain name and sending a DNS name query. For example, if the computer running Windows Vista 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 name query to resolve the name `isatap.example.microsoft.com`.
4. Using Link-Local Multicast Name Resolution (LLMNR) to attempt to resolve the name ISATAP on the local subnet (Windows Server 2008 and Windows Vista only).
5. Converting the ISATAP name into the NetBIOS name "ISATAP <00>" and checking the NetBIOS name cache.
6. Sending a NetBIOS name query to the configured Windows Internet Name Service (WINS) servers.
7. Sending NetBIOS broadcasts.
8. Checking the `Lmhosts` file in the `SystemRoot\system32\drivers\etc` folder.

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 Windows Server 2008 or Windows Vista, 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
```

**Note** When the IPv6 protocol for Windows XP with no service packs installed starts, it attempts to resolve the name "\_ISATAP", rather than "ISATAP".

### Using the netsh interface 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 isatap set router** command. The syntax of this command is **netsh interface isatap set router AddressOrName**, in which *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 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.

### ISATAP Addressing Example

Figure 12 shows an example ISATAP configuration.

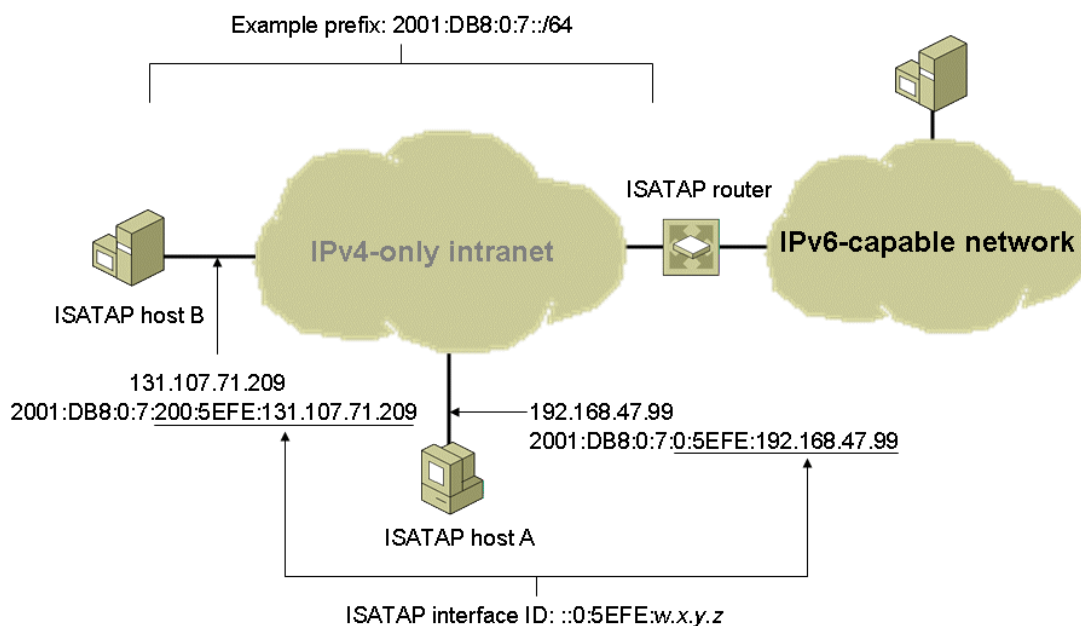


Figure 12: ISATAP Addressing Example

In this configuration, the ISATAP router is advertising the global subnet prefix 2001:DB8:0:7::/64 on the logical ISATAP subnet. ISATAP Host A, configured with the IPv4 address 192.168.47.99, uses the

subnet prefix advertised by the ISATAP router to automatically configure the global ISATAP address of 2001:DB8:0:7:0:5EFE:192.168.47.99. Similarly, ISATAP Host B uses the subnet prefix to automatically configure the global ISATAP address of 2001:DB8:0:7:200:5EFE:131.107.71.209.

## ISATAP Routing

Figure 13 shows the relevant routes for ISATAP communication in the example configuration.

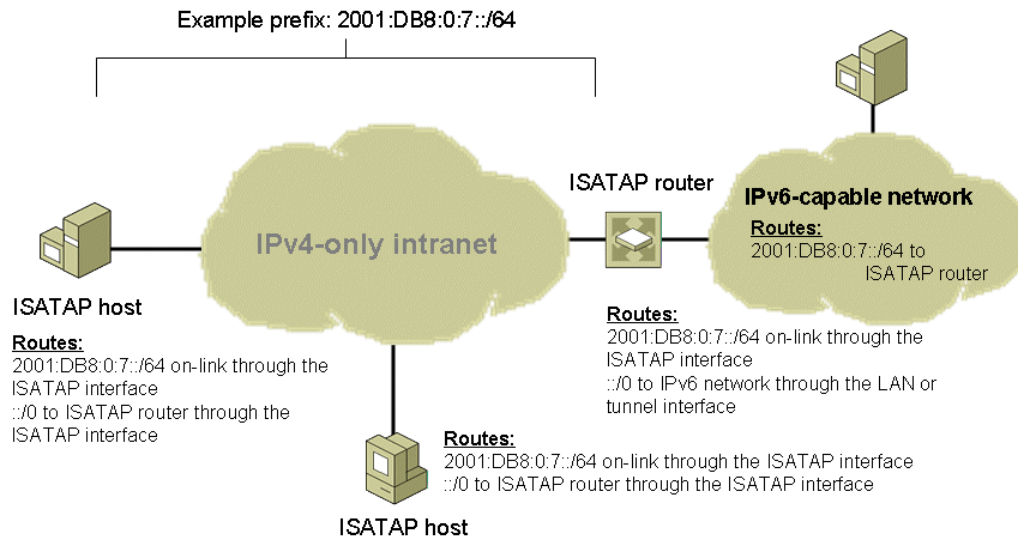


Figure 13: ISATAP Routing Example

ISATAP hosts use the following routes:

- An on-link route for the logical ISATAP subnet prefix that uses the ISATAP interface. This route allows ISATAP hosts to perform host-to-host tunneling to reach other ISATAP hosts on the same logical ISATAP subnet. In the example configuration, this is the 2001:DB8:0:7::/64 route.
- A default route that uses the ISATAP interface and has the next-hop address of the ISATAP router. This route allows ISATAP hosts to perform host-to-router tunneling to reach other IPv6 hosts on other IPv6 subnets.

An ISATAP router uses the following routes:

- An on-link route for the ISATAP subnet prefix that uses the ISATAP interface. This route allows the ISATAP router to perform router-to-host tunneling to reach other ISATAP hosts on the logical ISATAP subnet. In the example configuration, this is the 2001:DB8:0:7::/64 route.
- A default route that uses a LAN or tunneling interface and has the next-hop address of the next router on the IPv6-capable network (not shown in Figure 13). This route allows the ISATAP router to forward IPv6 traffic to destinations that are not located on the logical ISATAP subnet.

The routers of the IPv6-capable network use the following route:

- A route for the logical ISATAP subnet prefix that points back to the ISATAP router. This route allows the routers of the IPv6-capable network to forward traffic destined for ISATAP hosts to the ISATAP router.

## Configuring an ISATAP Router

A computer running Windows Server 2008 or Windows Vista 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 isatap set router AddressOrName
```

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

```
netsh interface ipv6 add route Address/PrefixLength InterfaceNameOrIndex publish=yes
```

The first command specifies the *AddressOrName* of either the IPv4 address of the ISATAP router's IPv4 intranet interface or the name of the router that resolves to the IPv4 address of the router's IPv4 intranet interface.

The second command enables forwarding and advertising on the name or interface index of the ISATAP tunneling interface.

The third command enables the advertisement of a specific subnet prefix over the ISATAP tunneling 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 to the ISATAP host.

For more information about deploying an ISATAP router, see the [Intra-site Automatic Tunnel Addressing Protocol Deployment Guide](http://www.microsoft.com/downloads/details.aspx?FamilyID=0f3a8868-e337-43d1-b271-b8c8702344cd&displaylang=en) at <http://www.microsoft.com/downloads/details.aspx?FamilyID=0f3a8868-e337-43d1-b271-b8c8702344cd&displaylang=en>.

## ISATAP Communication Examples

The following sections describe the details of how ISATAP communication works when an ISATAP host sends a packet to an ISATAP host on the same logical ISATAP subnet and when an ISATAP host sends a packet to an IPv6 host that is on another IPv6 subnet.

### ISATAP Host to ISATAP Host

Figure 14 shows how an ISATAP host communicates with another ISATAP host on the same logical ISATAP subnet.

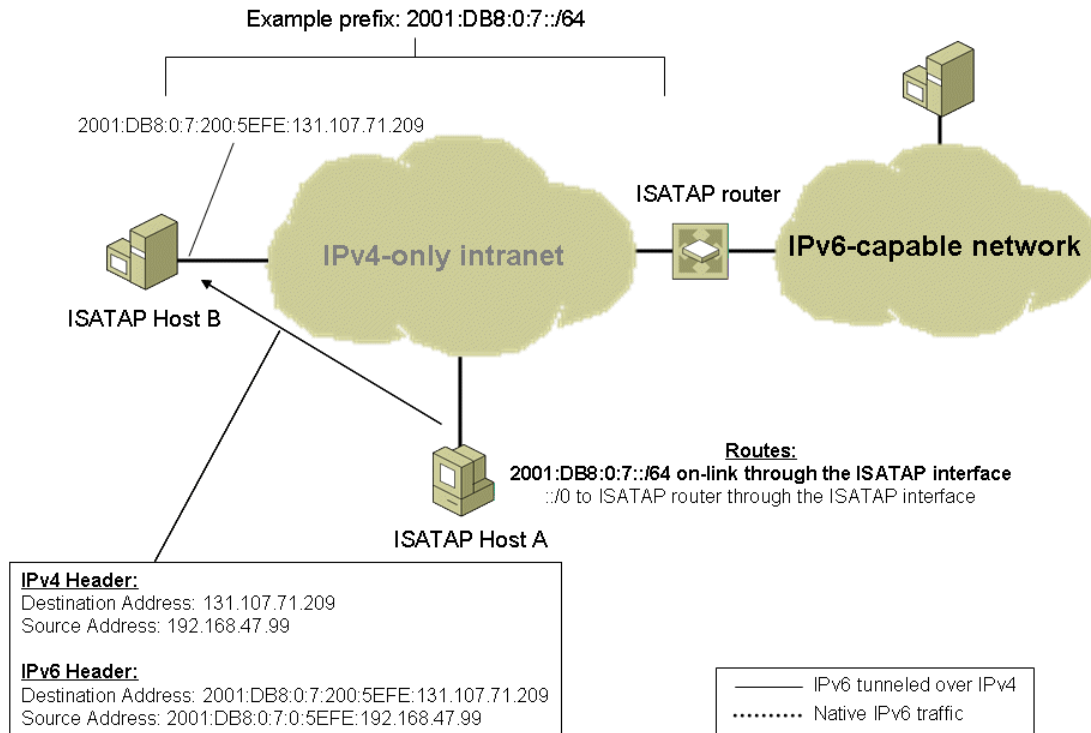


Figure 14: ISATAP Host to ISATAP Host Communication

In this example, ISATAP Host A wants to send a packet to ISATAP Host B. ISATAP Host A has resolved ISATAP Host B's IPv6 address through a DNS name query. When sending the packet, IPv6 on ISATAP Host A performs the IPv6 route determination process and finds that the closest matching route to the destination is the on-link 2001:DB8:0:7::/64 route. Because it is an on-link route, the next-hop IPv6 address is set to the destination address (2001:DB8:0:7:200:5EFE:131.107.71.209). The IPv6 packet and the next-hop address are handed to the ISATAP interface for processing.

The ISATAP interface sets the destination IPv4 address in the IPv4 header to the last 32-bits of the next-hop address, which in this case is ISATAP Host B's IPv4 address of 131.107.71.209. IPv4 on ISATAP Host A determines that the best source address to use is the IPv4 address assigned to ISATAP Host A (192.168.47.99) and then sends the packet.

On ISATAP Host B, IPv4 processes the IPv4 header and because the Protocol field is set to 41, it hands the IPv6 packet to the IPv6 protocol for further processing.

### ISATAP Host to IPv6 Host

When an ISATAP host sends to an IPv6 host on the IPv6-capable network, the packet's journey has two parts:

- From the ISATAP host to the ISATAP router
- From the ISATAP router to the IPv6 host

Continuing our example, when ISATAP Host A sends to a destination that is not on the logical ISATAP subnet (IPv6 Host C), IPv6 on ISATAP Host A performs the route determination process and finds that the closest matching route to the destination is the default route (::/0). The default route has a next-hop

IPv6 address of the link-local ISATAP address of the ISATAP router's interface on the IPv4-only intranet, which for our example is FE80::5EFE:10.0.0.1. The IPv6 packet and the next-hop address are handed to the ISATAP interface for processing.

The ISATAP interface sets the destination IPv4 address in the IPv4 header to the last 32-bits of the next-hop address, which in this case is the ISATAP router's IPv4 address of 10.0.0.1. IPv4 on ISATAP Host A determines that the best source address to use is the IPv4 address assigned to ISATAP Host A (192.168.47.99) and then sends the packet. Figure 15 shows the journey of the packet from ISATAP Host A to the ISATAP router.

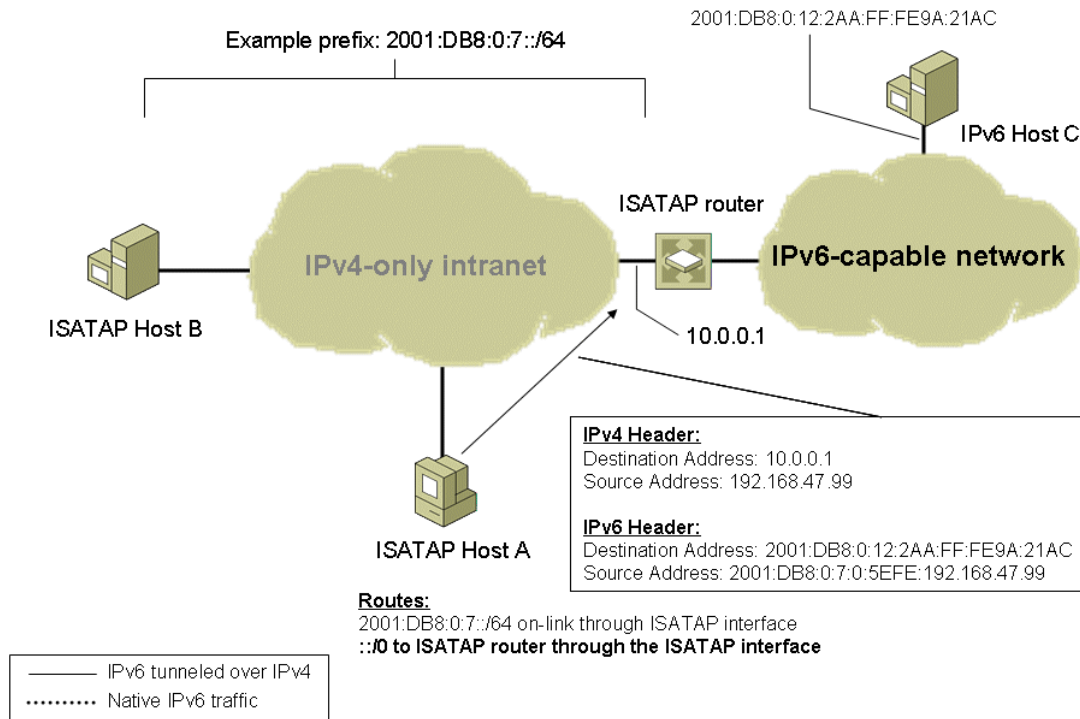


Figure 15: ISATAP Host to IPv6 Host Communication-Part 1

On the ISATAP router, IPv4 processes the IPv4 header and because the Protocol field is set to 41, it hands the IPv6 packet to IPv6 for processing. IPv6 on the ISATAP router performs the route determination process and finds that the closest matching route to the destination is the default route (::/0). The default route has a next-hop IPv6 address of the next IPv6 router on the IPv6-capable network (not shown). The IPv6 packet and the next-hop address are handed to the appropriate LAN or tunnel interface for processing. For a LAN interface, the IPv4 header is stripped off and the IPv6 router forwards the original IPv6 packet. The packet is forwarded across the IPv6-capable network to its destination. Figure 16 shows the journey of the packet from the ISATAP router to IPv6 Host C.

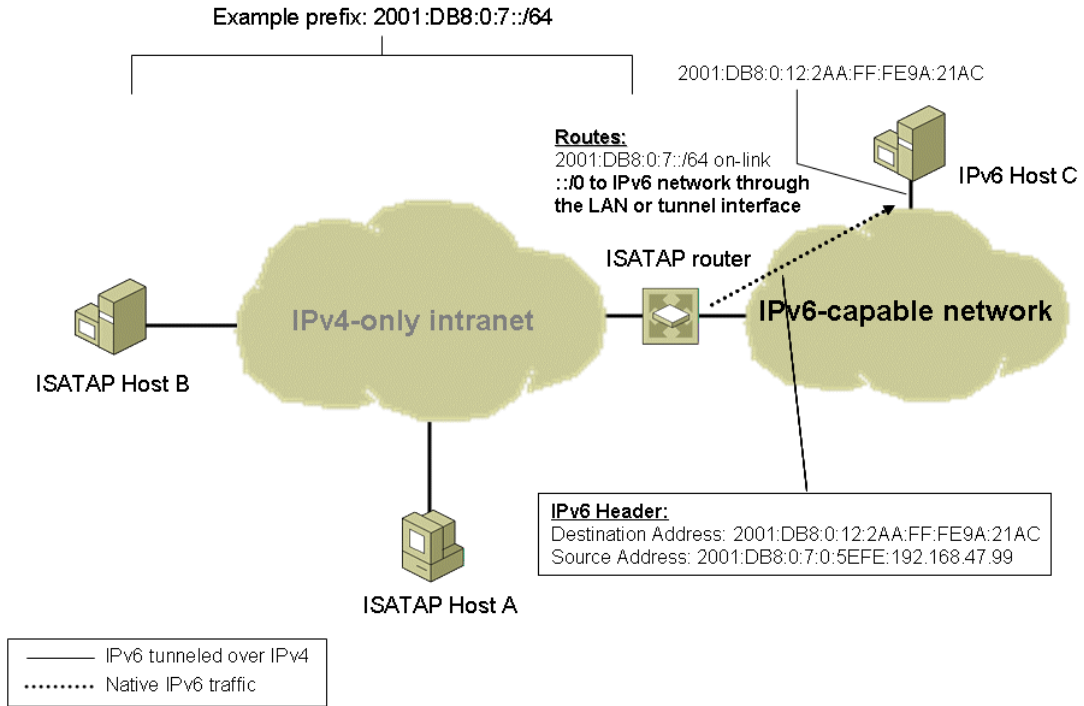


Figure 16: ISATAP Host to IPv6 Host Communication-Part 2

## 6to4

6to4 is an address assignment and router-to-router, host-to-router, and router-to-host automatic tunneling technology that is used to provide unicast IPv6 connectivity between IPv6 sites and hosts across the IPv4 Internet. 6to4 treats the entire IPv4 Internet as a single link. 6to4 is described in RFC 3056.

6to4 uses the global address prefix `2002:WWXX:YYZZ::/48`, in which `WWXX:YYZZ` is the colon-hexadecimal representation of a public IPv4 address (`w.x.y.z`) assigned to a site or host. Figure 17 shows the structure of a 6to4 address.

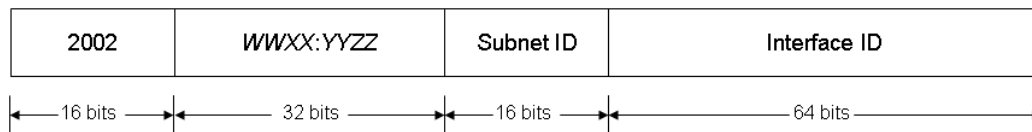


Figure 17: Structure of a 6to4 Address

6to4 allows you to assign global IPv6 addresses within your organization and to reach locations on the IPv6 Internet without requiring you to obtain a connection to the IPv6 Internet or an IPv6 global address prefix from an Internet service provider (ISP).

### 6to4 Components

Figure 18 shows the components of 6to4.

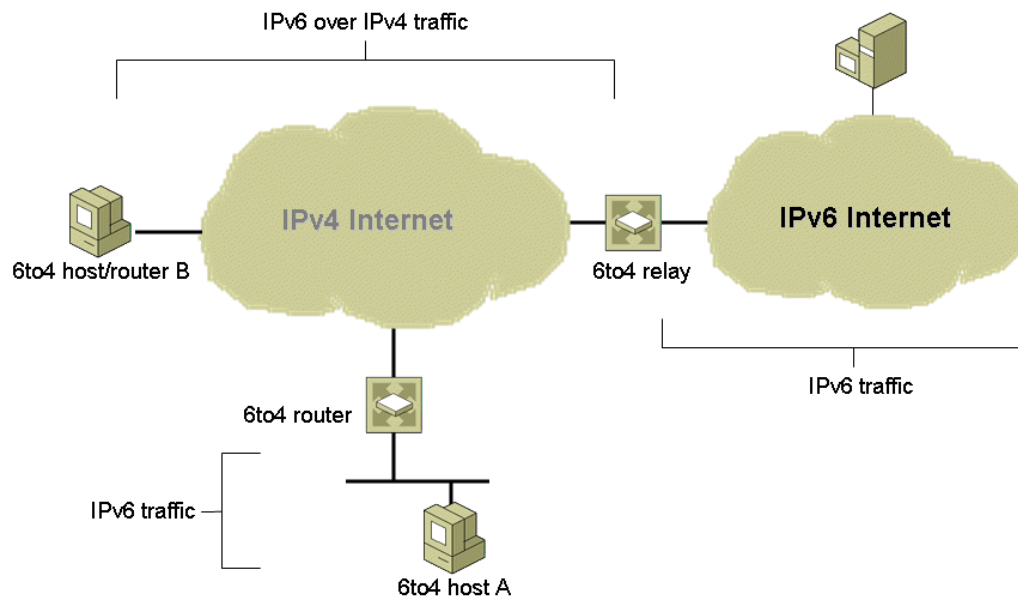


Figure 18: 6to4 Components

- 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 hosts do not perform IPv6 over IPv4 tunneling.

- 6to4 router

An IPv6/IPv4 router that uses a 6to4 tunneling interface to forward 6to4-addressed traffic between the 6to4 hosts within a site and other 6to4 routers, 6to4 host/routers, or 6to4 relays on the IPv4 Internet. 6to4 routers might require additional manual configuration.

- 6to4 host/router

An IPv6/IPv4 host that uses a 6to4 tunneling interface to exchange 6to4-addressed traffic with other 6to4 host/routers, 6to4 routers, or 6to4 relays on the IPv4 Internet. Unlike a 6to4 router, a 6to4 host/router does not forward traffic for other 6to4 hosts. An example of a 6to4 host/router is a computer running Windows Vista that is directly connected to the Internet and has been assigned a public IPv4 address.

- 6to4 relay

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

Within a site, local IPv6 routers advertise `2002:WWXX:YYZZ:SubnetID::/64` prefixes so that hosts can create an autoconfigured 6to4 address.

## 6to4 Addressing Example

Figure 19 shows an example 6to4 configuration.

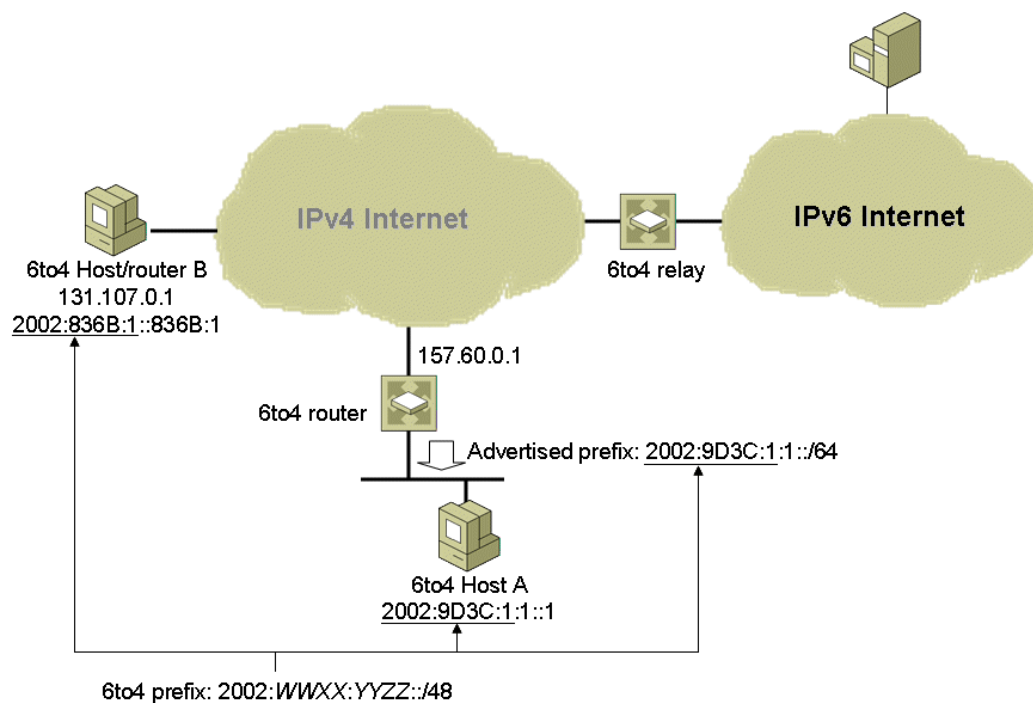


Figure 19: An Example 6to4 Configuration

The 6to4 router is directly connected to the Internet and has been assigned the public IPv4 address of 157.60.0.1. The 6to4 router creates the 48-bit prefix 2002:9D3C:1::/48, in which 9D3C:1 is the colon hexadecimal notation for 157.60.0.1. The 6to4 router advertises the 2002:9D3C:1:1::/64 prefix on the LAN interface connected to the private intranet. The *SubnetID* portion of the 64-bit prefix can be manually configured or automatically determined by the 6to4 router. IPv6 hosts on the private intranet subnet configure an IPv6 address based on the 2002:9D3C:1:1::/64 prefix using standard IPv6 stateless address autoconfiguration. In this example, 6to4 Host A automatically configures the IPv6 address 2002:9D3C:1:1::1.

6to4 Host/router B is directly connected to the Internet and has been assigned the public IPv4 address of 131.107.0.1. The IPv6 protocol for Windows Server 2008 and Windows Vista automatically derives an address of the form 2002:WWXX:YYZZ::WWXX:YYZZ. Therefore, 6to4 Host/router B assigns itself the IPv6 address 2002:836B:1::836B:1.

## 6to4 Routing

Figure 20 shows the relevant routes for 6to4 communication in the example configuration.

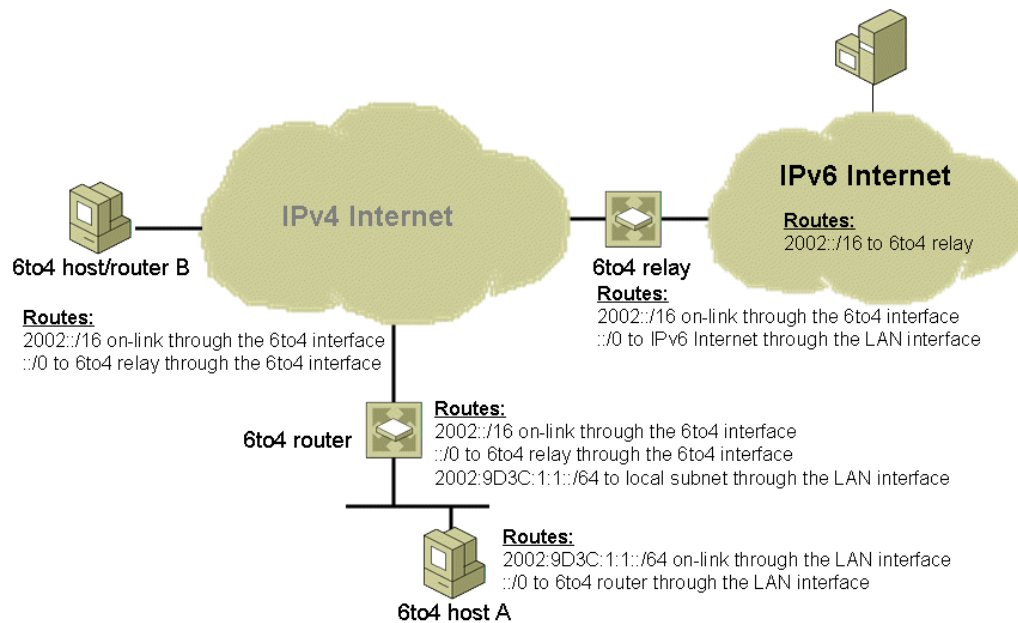


Figure 20: 6to4 Routing Example

6to4 hosts use the following routes:

- An on-link route for the intranet subnet prefix that uses the LAN interface. In the example configuration, this is the 2002:9D3C:1:1::/64 route.
- A default route that uses the LAN interface and has the next-hop address of the 6to4 router. This route allows 6to4 hosts to reach other 6to4 hosts or locations on the IPv6 Internet.

A 6to4 router uses the following routes:

- An on-link route for the intranet subnet prefix that uses the LAN interface. This route allows the 6to4 router to forward traffic to and from 6to4 hosts on the intranet subnet. In the example configuration, this is the 2002:9D3C:1:1::/64 route.

- An on-link route for the 6to4 address prefix (2002::/16) that uses the 6to4 tunneling interface. This route allows the 6to4 router to perform router-to-router tunneling to reach other 6to4 routers and the 6to4 relay and router-to-host tunneling to reach 6to4 host/routers.
- A default route that uses the 6to4 tunneling interface and has the next-hop address of the 6to4 relay. This route allows the 6to4 router to forward IPv6 traffic to IPv6 destinations that are located on the IPv6 Internet.

A 6to4 relay uses the following routes:

- An on-link route for the 6to4 address prefix (2002::/16) that uses the 6to4 tunneling interface. This route allows the 6to4 relay to perform router-to-router tunneling to reach 6to4 routers and router-to-host tunneling to reach 6to4 host/routers.
- A default route that uses a LAN (or tunnel) interface and has the next-hop address of the next router on the IPv6 Internet (not shown). This route allows the 6to4 relay to forward IPv6 traffic to IPv6 destinations that are located on the IPv6 Internet.

For 6to4 communication, the routers of the IPv6 Internet use the following route:

- A route for the 6to4 address prefix (2002::/16) that points back to the 6to4 relay. This route allows the routers of the IPv6 Internet to forward traffic destined for 6to4 hosts or 6to4 host/routers to the 6to4 relay.

## 6to4 Support in Windows Server 2008 and Windows Vista

The IPv6 protocol for Windows Server 2008 and Windows Vista provides support for 6to4 hosts/routers and 6to4 routers.

### 6to4 Host/router Support

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 IPv6 protocol does the following automatically:

- Configures 6to4 addresses of the form `2002:WWXX:YYZZ::WWXX:YYZZ` on a 6to4 tunneling interface for all public IPv4 addresses that are assigned to interfaces on the computer.
- Creates a `2002::/16` route that forwards all 6to4 traffic with the 6to4 tunneling interface. All traffic forwarded by this host to 6to4 destinations is encapsulated with an IPv4 header.
- Performs a DNS query for the name `6to4.ipv6.microsoft.com` to obtain the IPv4 address of a 6to4 relay on the IPv4 Internet. You can also use the **netsh interface 6to4 set relay** command to specify the DNS name to query. If the query is successful, the 6to4 component adds a default route that uses the 6to4 tunneling interface and sets the next-hop address to the 6to4 address of the 6to4 relay.

As a 6to4 host/router, a computer running Windows Server 2008 or Windows Vista can reach other 6to4 host/routers on the IPv4 Internet, 6to4 hosts in other 6to4 sites, and locations on the IPv6 Internet.

---

**Note** Windows Server 2003 and Windows XP always use an interface named “6to4 Tunneling Pseudo-Interface” for the 6to4 tunneling interface.

---

## 6to4 Router Support

A computer running Windows Server 2008 or Windows Vista can act as a 6to4 router by utilizing the configuration of the Internet Connection Sharing (ICS) feature. If ICS is enabled on an interface that is assigned a public IPv4 address, the 6to4 component automatically:

- Enables IPv6 forwarding on both the 6to4 tunneling and private interfaces.

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

- Determines a 64-bit IPv6 subnet prefix to advertise on the private intranet.

The 6to4 component derives the intranet subnet prefix from `2002:WWXX:YYZZ:InterfaceIndex::/64`, in which *InterfaceIndex* is the interface index of the private interface.

- Sends Router Advertisement messages on the private interface.

The router advertisements advertise the ICS computer as a default router and contain the derived 6to4 subnet prefix.

For example, for an ICS computer using the public IPv4 address of 131.107.23.89 and whose private interface is assigned the interface index 5, the advertised subnet prefix is `2002:836B:1759:5::/64`. Private hosts receiving this router advertisement 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 or 6to4 host/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 the 6to4 relay.

To manually configure a 6to4 router, see [Manual Configuration for IPv6](http://www.microsoft.com/technet/community/columns/cableguy/cg0902.mspx) at <http://www.microsoft.com/technet/community/columns/cableguy/cg0902.mspx>.

**Note** The 6to4 component of the IPv6 protocol 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 the public interface.

## 6to4 Communication Examples

The following sections describe the details of how 6to4 communication works when a 6to4 host sends a packet to a 6to4 host/router and when a 6to4 host sends a packet to an IPv6 host on the IPv6 Internet.

### 6to4 Host to 6to4 Host/router

In the example shown in Figure 21, 6to4 Host A wants to send a packet to 6to4 Host/router B. 6to4 Host A has resolved 6to4 Host B's IPv6 address through a DNS name query or other method, such as the Windows Peer-to-Peer Networking platform's Peer Name Resolution Protocol (PNRP). The journey of the packets from 6to4 Host A to 6to4 Host/router B has two parts:

- From 6to4 Host A to the 6to4 router
- From the 6to4 router to 6to4 Host/router B

In the first part of the journey, IPv6 on 6to4 Host A performs the route determination process and finds that the closest matching route to the destination is the default route. The default route has a next-hop address of the link-local address of the 6to4 router. 6to4 Host A performs normal IPv6 address resolution and sends the IPv6 packet to the 6to4 router. Figure 21 shows the delivery of the IPv6 packet to the 6to4 router.

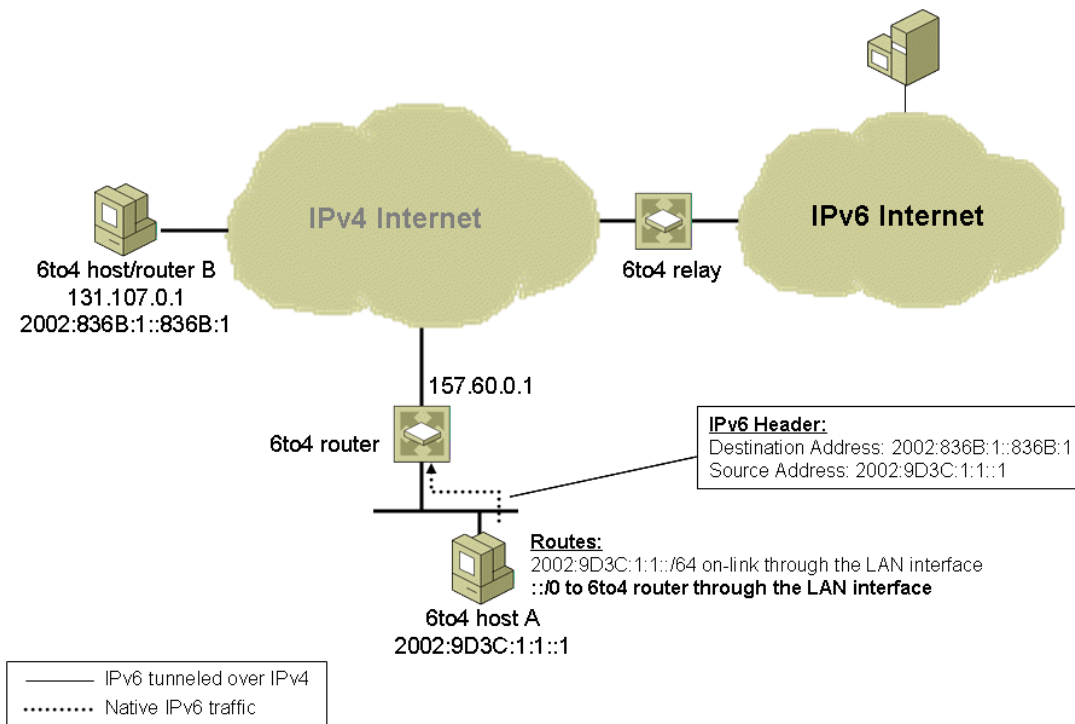


Figure 21: 6to4 Host to 6to4 Host/Router Communication-Part 1

In the second part of the journey, IPv6 on the 6to4 router performs the route determination process and finds that the closest matching route to the destination is the 2002::/16 route. Because it is an on-link route, the next-hop IPv6 address is set to the destination (2002:836B:1::836B:1). The IPv6 packet and the next-hop address are handed to the 6to4 interface for processing.

The 6to4 interface sets the destination IPv4 address in the IPv4 header to the 32-bits corresponding to the second and third blocks of the next-hop address, which in this case is 6to4 Host/router B's IPv4 address of 131.107.0.1. IPv4 on the 6to4 router determines that the best source address to use is the public IPv4 address assigned to the 6to4 router (157.60.0.1) and then sends the packet. Figure 22 shows the delivery of the IPv4-encapsulated IPv6 packet to 6to4 Host/router B.

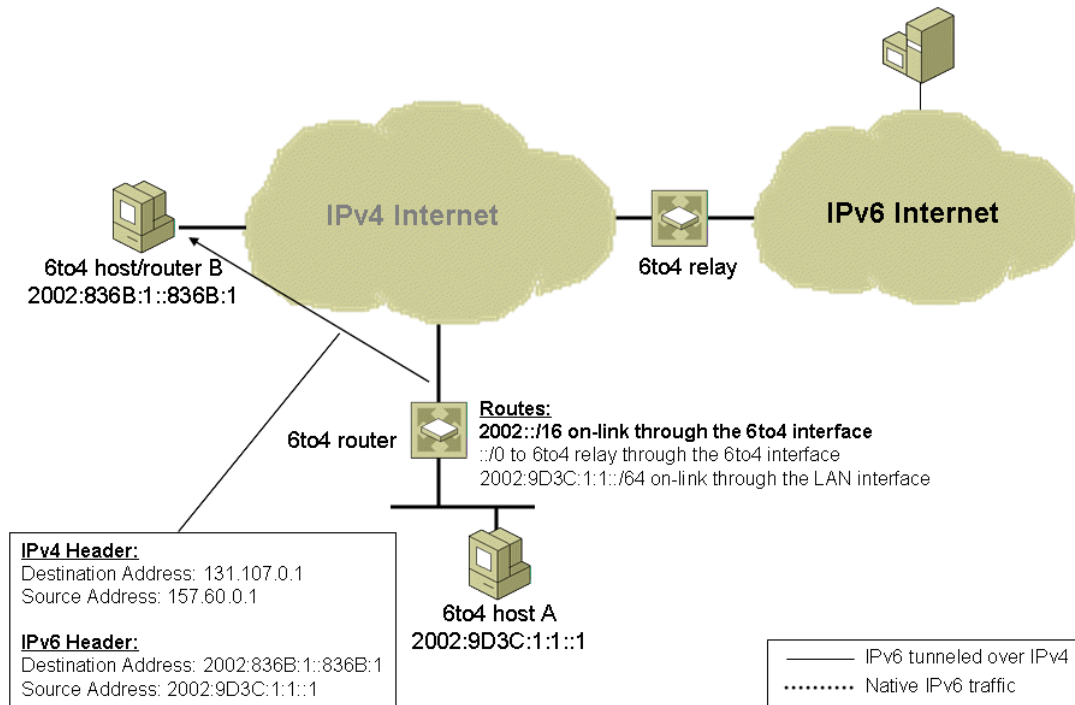


Figure 22: 6to4 Host to 6to4 Host/Router Communication-Part 2

On 6to4 Host/router B, IPv4 processes the IPv4 header and because the Protocol field is set to 41, it hands the IPv6 packet to IPv6 for further processing.

### 6to4 Host to IPv6 Host

When a 6to4 host sends to an IPv6 host on the IPv6 Internet, the packet's journey has three parts:

- From the 6to4 host to the 6to4 router
- From the 6to4 router to the 6to4 relay
- From the 6to4 relay to the IPv6 host

In the first part of the journey, 6to4 Host A resolves the IPv6 address of IPv6 Host C and sends the IPv6 packet to the 6to4 router in the manner described in the “6to4 Host to 6to4 Host/router” section of this paper.

In the second part of the journey, IPv6 on the 6to4 router performs the route determination process and finds that the closest matching route to the destination is the default route. The next-hop IPv6 address is set to the 6to4 address of the 6to4 relay (2002:C2CF:0105::1). The IPv6 packet and the next-hop address are handed to the 6to4 interface for processing.

The 6to4 interface sets the destination IPv4 address in the IPv4 header to the 32-bits corresponding to the second and third blocks of the destination 6to4 address, which in this case is 6to4 relay's IPv4 address of 194.207.1.5. IPv4 on the 6to4 router determines that the best source address to use is the public IPv4 address assigned to the 6to4 router (157.60.0.1) and then sends the packet. Figure 23 shows the delivery of the IPv4-encapsulated IPv6 packet to the 6to4 relay.

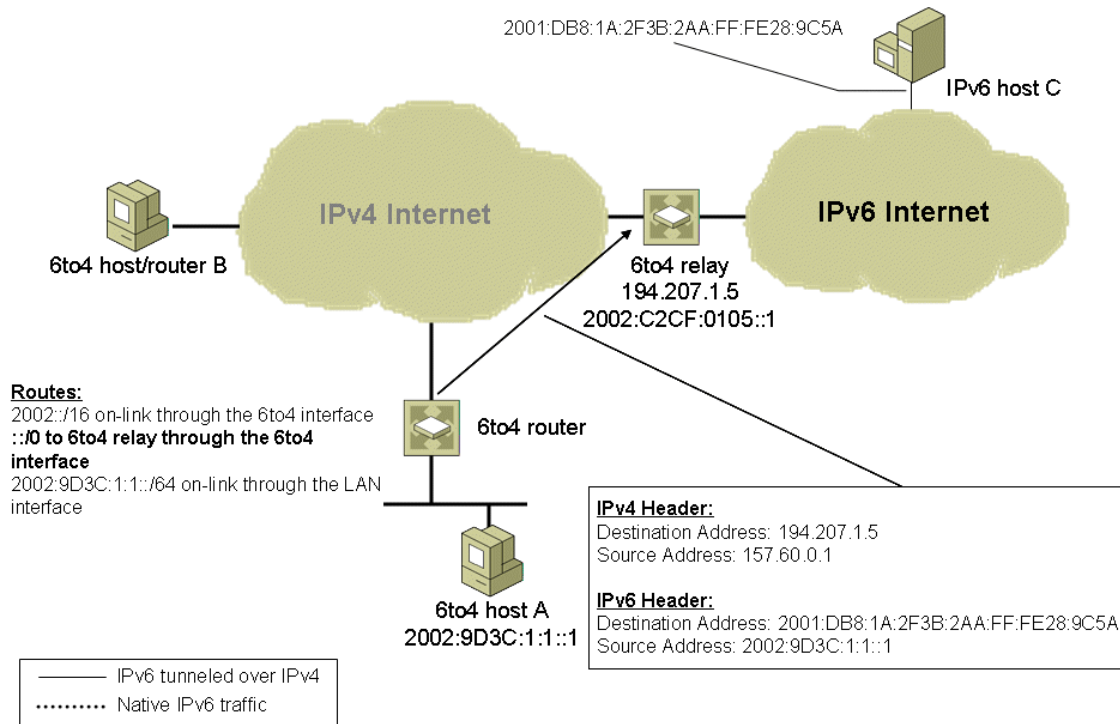


Figure 23: 6to4 Host to IPv6 Host Communication-Part 2

In the third part of the journey, IPv4 on the 6to4 relay processes the IPv4 header and because the Protocol field is set to 41, it hands the IPv6 packet to IPv6 for processing. IPv6 on the 6to4 relay performs the route determination process and finds that the closest matching route to the destination is the default route (::/0). The default route has a next-hop IPv6 address of the next IPv6 router on the IPv6 Internet (not shown in Figure 24). The IPv6 packet and the next-hop address are handed to the appropriate LAN (or tunnel) interface for processing. For a LAN interface, the IPv4 header is stripped off and the IPv6 router forwards the original IPv6 packet. The packet is forwarded across the IPv6 Internet to its destination. Figure 24 shows the journey of the IPv6 packet from the 6to4 relay to IPv6 Host C.

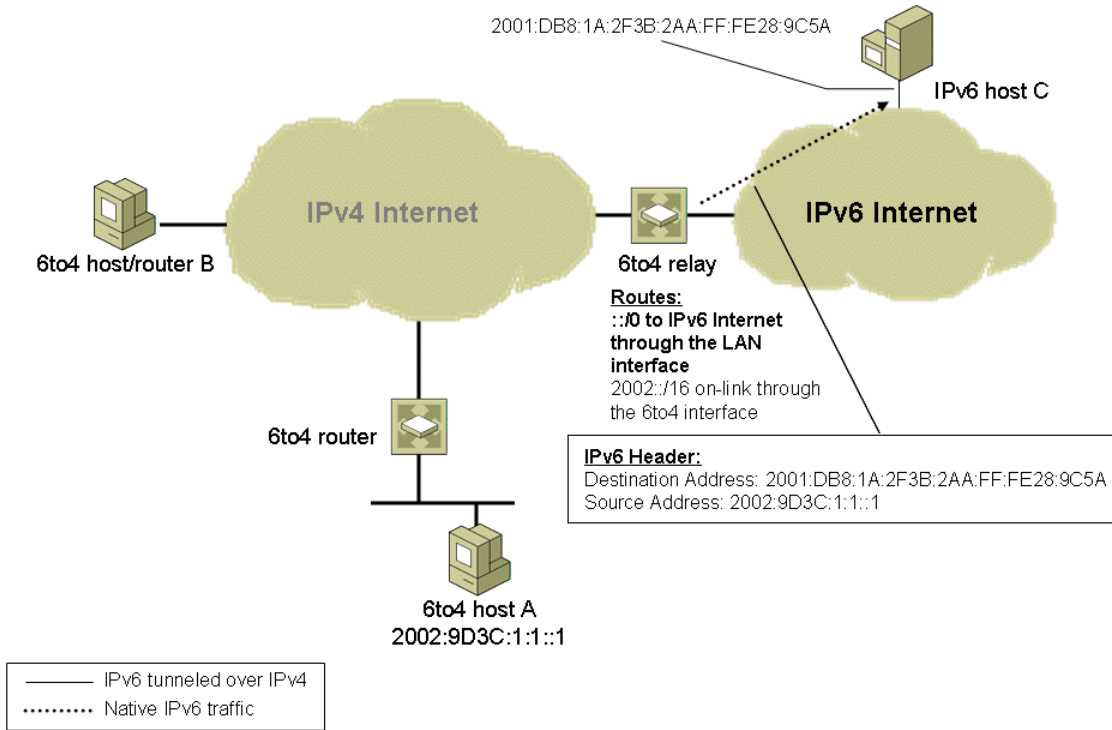


Figure 24: 6to4 Host to IPv6 Host Communication-Part 3

---

## Teredo

Teredo, also known as IPv4 network address translator (NAT) traversal (NAT-T) for IPv6, provides address assignment and host-to-host automatic tunneling for unicast IPv6 connectivity across the IPv4 Internet, even when the IPv6/IPv4 hosts are located behind one or multiple IPv4 NATs. To traverse IPv4 NATs, IPv6 packets are sent as IPv4-based User Datagram Protocol (UDP) messages. For more information about how network address translation works, see [Windows 2000 Network Address Translator \(NAT\)](http://www.microsoft.com/technet/community/columns/cableguy/cg0301.mspx) at <http://www.microsoft.com/technet/community/columns/cableguy/cg0301.mspx>.

6to4 provides a similar function as Teredo; however, 6to4 router support is required in the edge device that is connected to the Internet. 6to4 router functionality is not widely supported by IPv4 NATs. Even if the NAT were 6to4-enabled, 6to4 would still not work for configurations in which there are multiple NATs between a site and the IPv4 Internet.

Teredo resolves the issues of the lack of 6to4 functionality in modern-day NATs or multi-layered NAT configurations by tunneling IPv6 packets between the hosts within the sites. In contrast, 6to4 uses tunneling from the edge device. Tunneling from the hosts presents another issue for NATs: IPv4-encapsulated IPv6 packets are sent with the Protocol field in the IPv4 header set to 41. Most NATs only translate TCP or UDP traffic and must either be manually configured to translate other protocols or have an installed NAT editor that handles the translation. Because Protocol 41 translation is not a common feature of NATs, IPv4-encapsulated IPv6 traffic will not flow through typical NATs. Therefore, the IPv6 packet is encapsulated as an IPv4 UDP message, containing both IPv4 and UDP headers. UDP messages can be translated by most NATs and can traverse multiple layers of NATs.

Teredo is designed as a last resort transition technology for IPv6 connectivity. If native IPv6, ISATAP, or 6to4 connectivity is present between communicating nodes, Teredo is not used. As more IPv4 NATs are upgraded to support 6to4 and IPv6 connectivity become ubiquitous, Teredo will be used less and less, until eventually it is not used at all.

**Note** Teredo in Windows Server 2003 Service Pack 1, Windows XP SP2, and Windows XP SP1 with the Advanced Networking Pack for Windows XP works only over cone and restricted NATs. A cone NAT stores a mapping between an internal (private) address and port number and an external (public) address and port number. After the NAT translation table entry is in place, inbound traffic to the external address and port number is allowed from any source address and port number. A restricted NAT stores a mapping between an internal address and port number and an external address and port number, for either specific external addresses or specific external addresses and port numbers. An inbound packet that does not match a NAT translation table entry for both the external destination address and port number and a specific source external address or port number is silently discarded. There is an additional type of NAT, known as a symmetric NAT, which maps the same internal address and port number to different external addresses and ports, depending on the external destination address (for outbound traffic). Teredo in Windows Server 2008 and Windows Vista can work between Teredo clients if only one Teredo client is behind one or more symmetric NATs.

For the security implications of using Teredo, see [Using IPv6 and Teredo](http://www.microsoft.com/technet/prodtechnol/winxppro/evaluate/ipv6_teredo.mspx) at [http://www.microsoft.com/technet/prodtechnol/winxppro/evaluate/ipv6\\_teredo.mspx](http://www.microsoft.com/technet/prodtechnol/winxppro/evaluate/ipv6_teredo.mspx).

For information about changes in Teredo support, see [Changes to IPv6 in Windows Server 2008 and Windows Vista](http://www.microsoft.com/technet/community/columns/cableguy/cg1005.mspx) at <http://www.microsoft.com/technet/community/columns/cableguy/cg1005.mspx>.

## Teredo Components

Figure 25 shows the set of components that enables Teredo connectivity.

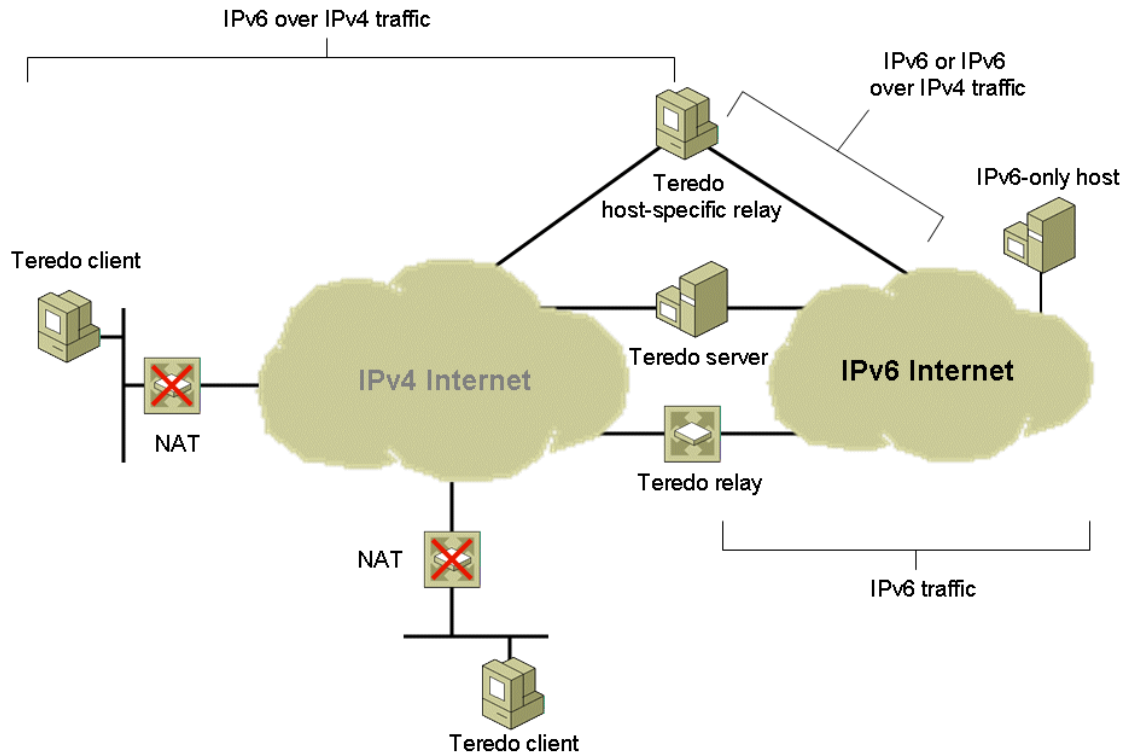


Figure 25: Teredo Components

- Teredo client

An IPv6/IPv4 node that supports a Teredo tunneling interface through which packets are tunneled to either other Teredo clients (using host-to-host tunneling) or nodes on the IPv6 Internet through a Teredo relay (using host-to-router tunneling).

- Teredo server

An IPv6/IPv4 node that is connected to both the IPv4 Internet and the IPv6 Internet. The Teredo server assists in the initial configuration of Teredo clients and facilitates the initial communication between either Teredo clients or between Teredo clients and IPv6-only hosts.

- Teredo relay

An IPv6/IPv4 router that can use host-to-router and router-to-host tunneling to forward packets between Teredo clients on the IPv4 Internet and IPv6-only hosts on the IPv6 Internet.

- Teredo host-specific relay

An IPv6/IPv4 node that has an interface and connectivity to both the IPv4 Internet and the IPv6 Internet and can communicate directly with Teredo clients over the IPv4 Internet, without the need for an intermediate Teredo relay. The connectivity to the IPv4 Internet can be through a public IPv4 address or through a private IPv4 address and an intermediate NAT. The connectivity to the IPv6 Internet can be through a direct connection to the IPv6 Internet or through an IPv6 transition technology such as 6to4.

Windows Server 2008 and Windows Vista include Teredo client and Teredo host-specific relay functionality.

## Teredo Address Format

Figure 26 shows the format of Teredo addresses.

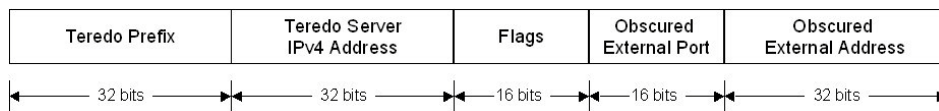


Figure 26: Teredo Addresses

A Teredo address consists of the following:

- Teredo prefix

The first 32 bits are for the Teredo prefix, which is the same for all Teredo addresses. The address space of 2001::/32 has been reserved for Teredo by IANA in RFC 4380 and is the prefix used by Teredo in Windows Server 2008 and Windows Vista. The 3FFE:831F::/32 prefix was initially used for the Windows XP implementation of Teredo. Computers running Windows XP will use the new 2001::/32 prefix when updated with Service Pack 3.

- Teredo server IPv4 address

The next 32 bits contain the IPv4 public address of the Teredo server that assisted in the configuration of this Teredo address.

- Flags

The next 16 bits are reserved for Teredo flags. The Cone flag is set when Teredo client is behind a cone NAT. For Windows Server 2008 and Windows Vista-based Teredo clients, unused bits within the Flags field provide a level of protection from address scans by malicious users. The 16 bits within the Flags field for Windows Server 2008 and Windows Vista-based Teredo clients consists of the following: CRAAAAUG AAAAAAAAA. The C bit is for the Cone flag. The R bit is reserved for future use. The U bit is for the Universal/Local flag (set to 0). The G bit is Individual/Group flag (set to 0). The A bits are set to a 12-bit randomly generated number. By using a random number for the A bits, a malicious user that has determined the rest of the Teredo address by capturing the initial configuration exchange of packets between the Teredo client and Teredo server will have to try up to 4,096 ( $2^{12}$ ) different addresses to determine a Teredo client's address during an address scan.

- Obscured external port

The next 16 bits store an obscured version of the external UDP port that corresponds to all Teredo traffic for this Teredo client. When the Teredo client sends its initial packet to a Teredo server, the source UDP port of the packet is mapped by the NAT to a different, external UDP port. All Teredo traffic for the host uses the same external, mapped UDP port.

The external port is obscured by exclusive ORing (XORing) the external port with 0xFFFF. For example, the obscured version of the external port 5000 in hexadecimal format is EC77 (5000 = 0x1388, and 0x1388 XOR 0xFFFF = 0xEC77). Obscuring the external port prevents NATs from translating the port number within the payload of the packets that are being forwarded.

- Obscured external address

The last 32 bits store an obscured version of the external IPv4 address that corresponds to all Teredo traffic for this Teredo client. Just like the external port, when the Teredo client sends its initial packet to a Teredo server, the source IP address of the packet is mapped by the NAT to a different, external address.

The external address is obscured by XORing the external address with 0xFFFFFFFF. For example, the obscured version of the public IPv4 address 131.107.0.1 in colon-hexadecimal format is 7C94:FFFE (131.107.0.1 = 0x836B0001, and 0x836B0001 XOR 0xFFFFFFFF = 0x7C94FFFE). Obscuring the external address prevents NATs from translating the address within the payload of the packets that are being forwarded.

### Teredo Addressing Example

Figure 27 shows an example Teredo configuration with two Teredo clients, one Teredo client located behind a cone NAT (Teredo Client A) and one located behind a restricted NAT (Teredo Client B).

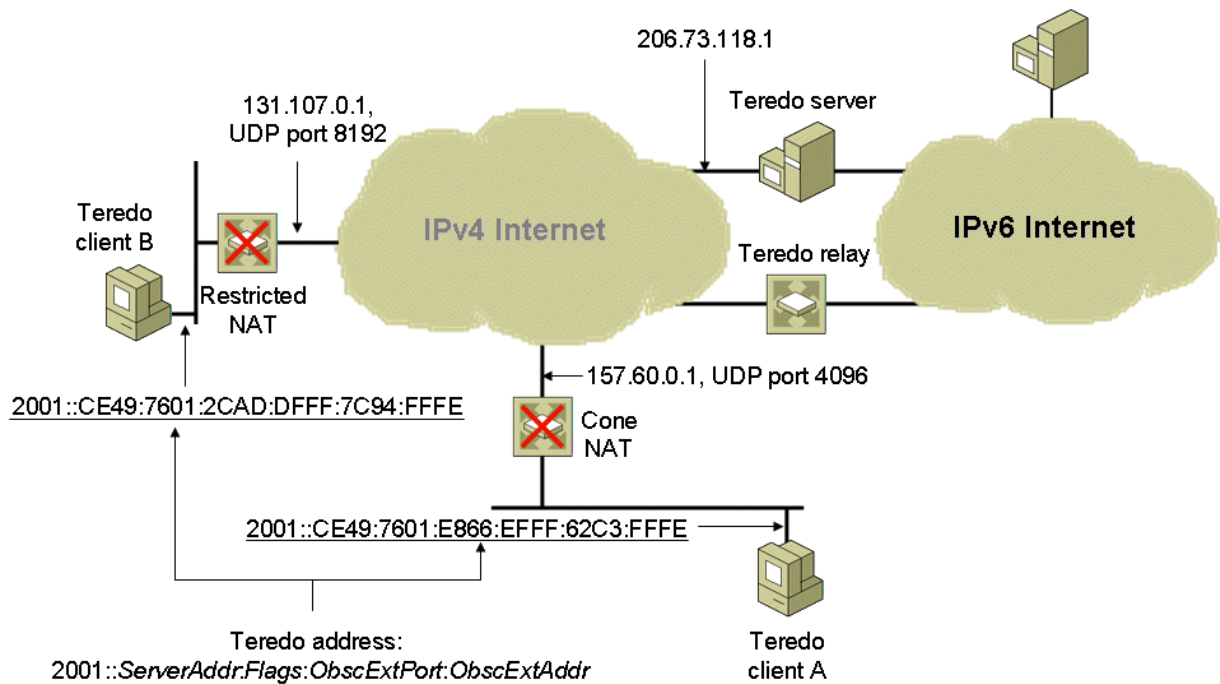


Figure 27: Teredo Addressing Example

For Teredo client A, the following are used to construct its Teredo address:

- Its Teredo server is at the public IPv4 address of 206.73.118.1
- It is behind a cone NAT
- The external address and port for its Teredo traffic is 157.60.0.1, UDP port 4096

Therefore, using the Teredo address format of 2001::ServerAddr:Flags:ObscExtPort:ObscExtAddr, a Teredo address for Teredo client A is 2001::CE49:7601:E866:FFFF:62C3:FFFE. This is based on the following:

- The 2001::/32 Teredo prefix.

- CE49:7601 is the colon-hexadecimal version of 206.73.118.1.
- E866 is the Flags field in which the Cone flag is set to 1 (indicating that Teredo Client A is located behind a cone NAT), the U and G flags are set to 0, and the remaining 12 bits are set to a random sequence to help prevent external address scans.
- EFFF is the obscured version of UDP port 4096.
- 62C3:FFFE is the obscured version of the public IPv4 address 157.60.0.1.

For Teredo Client B, the following are used to construct its Teredo address:

- Its Teredo server is at the public IPv4 address of 206.73.118.1.
- It is behind a restricted NAT.
- The external address and port for its Teredo traffic is 131.107.0.1, UDP port 8192.

Therefore, a Teredo address for Teredo Client B is 2001::CE49:7601:2CAD:DFFF:7C94:FFFE. This is based on the following:

- The 2001::/32 Teredo prefix.
- CE49:7601 is the colon-hexadecimal version of 206.73.118.1.
- 2CAD is the Flags field in which the Cone flag is set to 0 (indicating that Teredo Client B is located behind a restricted NAT), the U and G flags are set to 0, and the remaining 12 bits are set to a random sequence to help prevent external address scans.
- DFFF is the obscured version of UDP port 8192.
- 7C94:FFFE is the obscured version of the public IPv4 address 131.107.0.1.

## **Teredo Routing**

Figure 28 shows the relevant routes for Teredo communication.

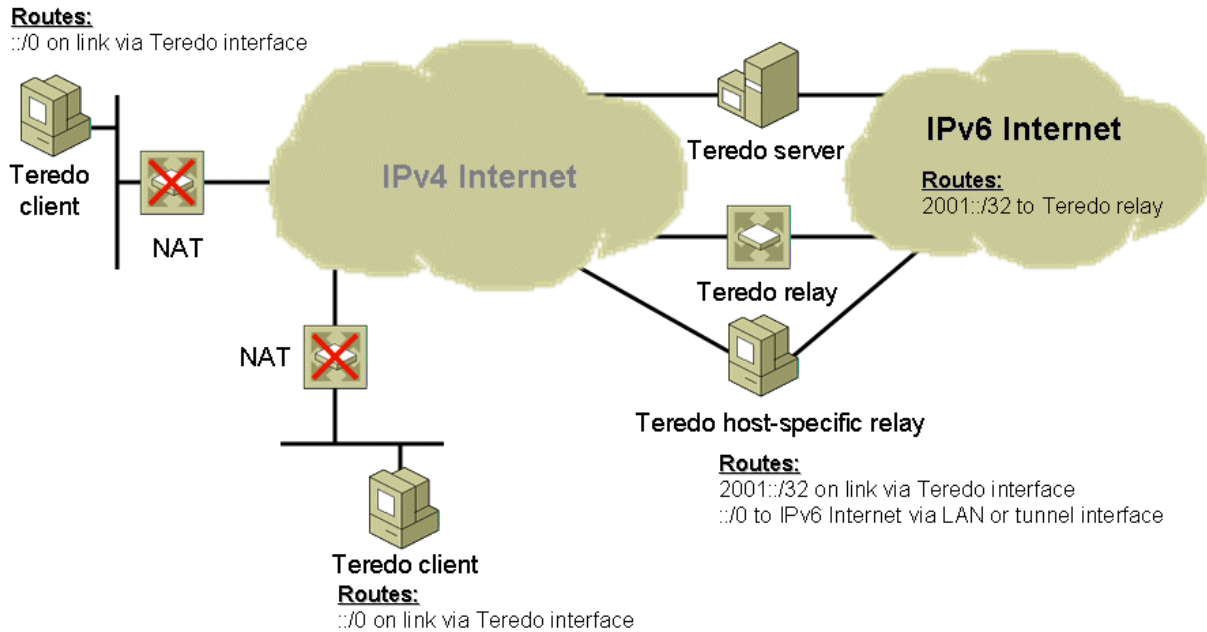


Figure 28: Teredo Routing Example

Teredo hosts use a default route that considers all IPv6 addresses as on-link and using the Teredo tunneling interface. When this default route is used, the next-hop address is set to the destination address in the IPv6 packet and the next-hop interface is set to the Teredo tunneling interface. The Teredo tunneling interface determines how to deliver the packet.

Teredo servers, Teredo relays, and Teredo host-specific relays use the following routes:

- An on-link route for the Teredo prefix that uses the Teredo interface. In the example configuration, this is the 2001::/32 route.
- A default route that uses a LAN (or tunnel) interface that is attached to the IPv6 Internet. This route allows Teredo servers, Teredo relays, and Teredo host-specific relays to reach locations on the IPv6 Internet.

Within the IPv6 Internet, the 2001::/32 route points to a Teredo relay.

## How Teredo Works

For two Windows-based Teredo clients, the most crucial Teredo processes are those used for initial configuration and communication with a Teredo client in a different site.

### Initial Configuration

Initial configuration for Teredo clients is accomplished by sending a series of Router Solicitation messages to a set Teredo of servers. The responses are used to derive a Teredo address and determine whether the client is behind a cone, restricted, or symmetric NAT. You can see what type of NAT the Teredo client has determined from the display of the **netsh interface ipv6 show teredo** command.

Based on the received Router Advertisement messages, the Teredo client constructs its Teredo address from the following:

- The first 64 bits are set to the value included in the Prefix Information option of the received router advertisement. The 64-bit prefix advertised by the Teredo server consists of the Teredo prefix (32 bits) and the public IPv4 address of the Teredo server (32 bits).
- The next 16 bits are the Flags field.
- The next 16 bits are set to the obscured external UDP port number that is included in a special Teredo header in the router advertisement.
- The last 32 bits are set to the obscured external IPv4 address that is included in a special Teredo header in the router advertisement.

### Initial Communication Between Two Teredo Clients in Different Sites

The initial communication process between Teredo clients located in different sites depends on whether those sites are behind cone NATs or restricted NATs.

When both Teredo clients are located behind cone NATs, the NAT translation table entries for Teredo traffic for each Teredo client allows traffic from any source IP address or source UDP port. Therefore, a Teredo client in one site can send packets directly to a Teredo client in another site without the use of additional packets to establish NAT translation table entries.

When the Teredo clients are located behind restricted NATs, additional NAT translation table entries must be created before unicast packets can be exchanged. Figure 29 shows the initial communication process between Teredo clients that are located in different sites when both sites are behind restricted NATs.

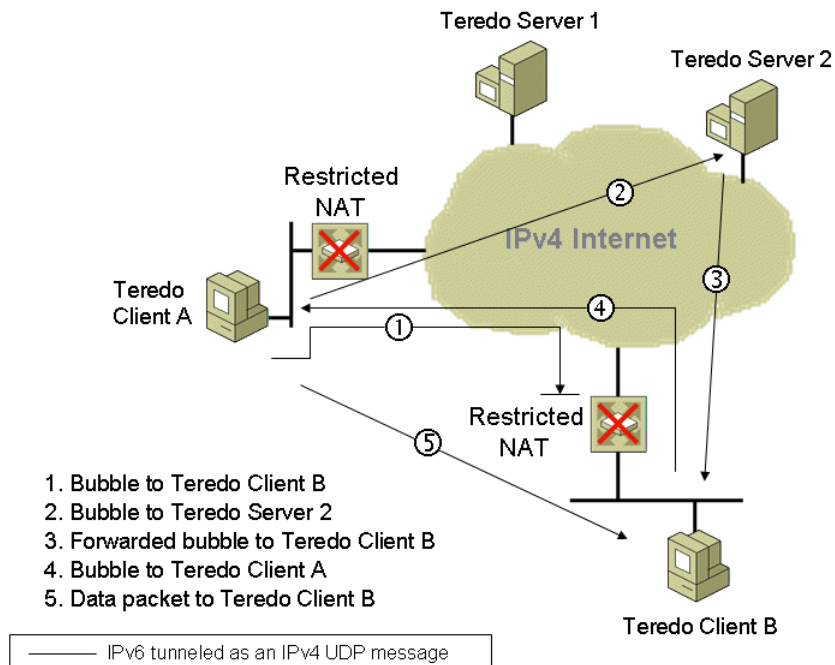


Figure 29: Initial Communications Between Teredo Clients Located Behind Restricted NATs

To send an initial communication packet from Teredo Client A to Teredo Client B, the following process is used:

1. Teredo Client A sends a bubble packet directly to Teredo Client B. A bubble packet contains no data and is used to create or maintain NAT translation mappings. Because Teredo Client B is behind a restricted NAT, Teredo traffic from an arbitrary source IPv4 address and UDP port number is not allowed unless there is a source-specific NAT translation table entry. Assuming that there is none, the restricted NAT silently discards the bubble packet. However, when the restricted NAT for Teredo Client A forwarded the bubble packet, it created a source-specific NAT translation table entry that will allow future packets sent from Teredo Client B to be forwarded to Teredo Client A.
2. Teredo Client A sends a bubble packet to Teredo Client B through Teredo Server 2 (Teredo Client B's Teredo server). The IPv4 destination address in the bubble packet is set to the IPv4 address of Teredo Server 2, which Teredo Client A determines from the third and fourth blocks of Teredo Client B's Teredo address.
3. Teredo Server 2 forwards the bubble packet to Teredo Client B. The restricted NAT for Teredo Client B forwards the packet because there is an existing source-specific mapping for Teredo traffic from Teredo Server 2 (established by the initial configuration of Teredo Client B and maintained over time).
4. Teredo Client B responds to the bubble packet received from Teredo Client A with its own bubble packet, which is sent directly to Teredo Client A. Because Teredo Client A's restricted NAT has a source-specific mapping for Teredo traffic from Teredo Client B (as established by the initial bubble packet sent from Teredo Client A in step 1), it forwards the bubble packet to Teredo Client A.
5. Upon receipt of the bubble packet from Teredo Client B, Teredo Client A determines that source-specific NAT mappings exist for both NATs. Teredo Client A sends an initial communication packet directly to Teredo Client B. Subsequent packets are sent directly between Teredo Client A and Teredo Client B.

This process occurs transparently to the user at Teredo Client A.

There are additional initial communication processes that depend on whether the destination for the initial communication is on the same link, on the IPv6 Internet, or with a Teredo host-specific relay. For more information, see [Teredo Overview](http://www.microsoft.com/technet/network/ipv6/teredo.msp) at <http://www.microsoft.com/technet/network/ipv6/teredo.msp>.

---

## 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 Windows Server 2008 and Windows Vista 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 transition is from IPv4 to IPv6 and from IPv6 to IPv4. 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 A records in 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 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 have not yet been IPv6-enabled.

To configure the PortProxy component, use the **netsh interface portproxy add|set|delete v4tov4|v4tov6|v6tov4|v6tov6** commands.

**Note** 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.

---

## Migrating to IPv6

As a general methodology, to migrate from IPv4 to IPv6, you can perform the following steps:

1. Begin upgrading your applications to be independent of IPv6 or IPv4.

For example, Windows Sockets applications might need 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 (required) and PTR records in the IP6.ARPA reverse domain (optional). Additionally, ensure that the DNS servers support DNS dynamic update for AAAA records so that IPv6 hosts can automatically register their names and IPv6 addresses.

3. Upgrade hosts to IPv6/IPv4 nodes.

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

4. Deploy ISATAP.

This optional step provides tunneled IPv6 connectivity before native IPv6 connectivity is deployed across your network.

5. Upgrade routing infrastructure for native IPv6 routing.

Routers must be upgraded and configured to support native IPv6 prefix advertisement and routing.

---

## Summary

Migrating to IPv6 involves the upgrading of applications, hosts, routers, and DNS to support IPv6. Because this migration might take years, IPv6/IPv4 nodes must be able to coexist over IPv4 infrastructures such as the Internet and private intranets. To provide automatic configuration and tunneling over an IPv4 intranet, the IPv6 protocol for Windows Server 2008 and Windows Vista supports ISATAP. To provide automatic configuration and tunneling over the IPv4 Internet, the IPv6 protocol for Windows Server 2008 and Windows Vista supports 6to4. To provide automatic configuration and tunneling over the IPv4 Internet when nodes are separated by NATs, the IPv6 protocol in Windows Server 2008 and Windows Vista supports a Teredo client and host-specific relay. To provide support between IPv4-only and IPv6-only nodes and services, Windows Server 2008 and Windows Vista supports PortProxy.

---

## Related Links

See the following resources for further information:

- ["Understanding IPv6, Second Edition" Microsoft Press book](http://www.microsoft.com/MSPress/books/11607.aspx) at <http://www.microsoft.com/MSPress/books/11607.aspx>
- [Microsoft IPv6 Web site](http://www.microsoft.com/ipv6) at <http://www.microsoft.com/ipv6>
- [IP Version 6 Working Group Web site](http://www.ietf.org/html.charters/old/ipv6-charter.html) at <http://www.ietf.org/html.charters/old/ipv6-charter.html>

For the latest information about Windows Server 2008, see the [Windows Server 2008 Web site](http://www.microsoft.com/windowsserver2008) at <http://www.microsoft.com/windowsserver2008>.