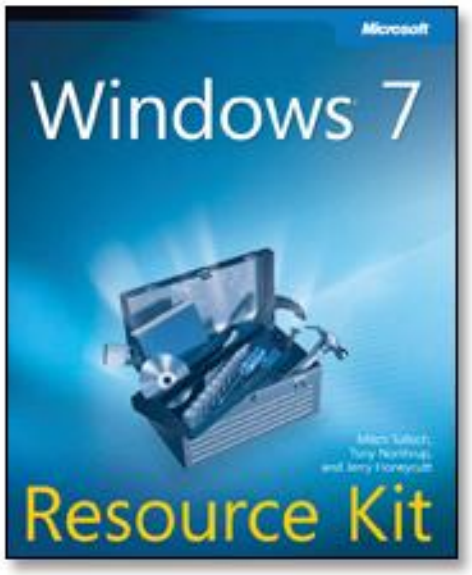

Windows® 7 Resource Kit

Mitch Tulloch,
Tony Northrup,
and Jerry Honeycutt

To learn more about this book, visit Microsoft Learning at http://www.microsoft.com/MSPress/books/

9780735627000

Microsoft® Press

Table of Contents

CHAPTER 29

Deploying IPv6

Like Windows Vista before it, Windows 7 has a new Next Generation TCP/IP stack with enhanced support for Internet Protocol version 6 (IPv6). This chapter provides you with an understanding of why IPv6 is necessary and how it works. The chapter describes the IPv6 capabilities in Windows 7, Windows Vista and Windows Server 2008 and outlines how to migrate the IPv4 network infrastructure of your enterprise to IPv6 using IPv6 transition technologies such as Intra-Site Automatic Tunnel Addressing Protocol (ISATAP). Finally, the chapter describes how to configure and manage IPv6 settings in Windows 7 and how to troubleshoot IPv6 networking problems.

Understanding IPv6

The need for migrating enterprise networks from IPv4 to IPv6 is driven by a number of different technological, business, and social factors. The most important of these are:

- The exponential growth of the Internet is rapidly exhausting the existing IPv4 public address space. A temporary solution to this problem has been found in Network Address Translation (NAT), a technology that maps multiple private (intranet) addresses to a (usually) single, public (Internet) address. Unfortunately, using NAT-enabled routers can introduce additional problems such as breaking end-to-end connectivity and security for some network applications. In addition, the rapid proliferation of mobile IP devices is accelerating the depletion of the IPv4 public address space.
- The growing use of real-time communications (RTC) on the Internet, such as Voice Over Internet Protocol (VoIP) telephony, Instant Messaging (IM), and audio/video conferencing, exposes the limited support for Quality of Service (QoS) currently provided in IPv4. These new RTC technologies need improved QoS on IP networks to ensure reliable end-to-end communications. The design of IPv4 limits possible improvements.
- The growing threats faced by hosts on IPv4 networks connected to the Internet can be mitigated considerably by deploying Internet Protocol security (IPsec), both on private intranets and on tunneled connections across the public Internet. However, IPsec was designed as an afterthought to IPv4 and is complex and difficult to implement in many scenarios.

IPv6, developed by the Internet Engineering Task Force (IETF) to solve these problems, includes the following improvements and additions:

- IPv6 increases the theoretical address space of the Internet from 4.3×10^9 addresses (based on 32-bit IPv4 addresses) to 3.4×10^{38} possible addresses (based on 128-bit IPv6 addresses), which most experts agree should be more than sufficient for the foreseeable future.
- The IPv6 address space was designed to be hierarchical rather than flat in structure, which means that routing tables for IPv6 routers can be smaller and more efficient than for IPv4 routers.
- IPv6 has enhanced support for QoS that includes a Traffic Class field in the header to specify how traffic should be handled, and a new Flow Label field in the header that enables routers to identify packets that belong to a traffic flow and handle them appropriately.
- IPv6 now requires IPsec support for standards-based, end-to-end security across the Internet. The new QoS enhancements work even when IPv6 traffic is encrypted using IPsec.

Understanding how IPv6 works is essential if you plan to benefit from IPv6 by deploying it in your enterprise. The following sections provide an overview of key IPv6 concepts, features, and terminology.

> Note For more detailed information on IP concepts, features, and terminology, see the white paper titled "Introduction to IP Version 6" at *http://www.microsoft.com/downloads/details.aspx?FamilyID=CBC0B8A3-B6A4-4952-BBE6-D976624C257C&displaylang=en*. Another good reference for learning IPv6 is the book *Understanding IPv6, Second Edition*, by Joseph Davies (Microsoft Press, 2008). See *http://www.microsoft.com/MSPress/books/11607.aspx*.

Understanding IPv6 Terminology

The following terminology is used to define IPv6 concepts and describe IPv6 features:

- Node An IPv6-enabled network device that includes both hosts and routers.
- Host An IPv6-enabled network device that cannot forward IPv6 packets that are not explicitly addressed to itself. A host is an endpoint for IPv6 communications (either the source or destination) and drops all traffic not explicitly addressed to it.
- Router An IPv6-enabled network device that can forward IPv6 packets that are not explicitly addressed to itself. IPv6 routers also typically advertise their presence to IPv6 hosts on their attached links.
- Link One or more LAN (such as Ethernet) or WAN (such as PPP) network segments

bounded by routers. Like interfaces, links may be either physical or logical.

- Neighbors Nodes that are connected to the same physical or logical link.
- Subnet One or more links having the same 64-bit IPv6 address prefix.
- Interface A representation of a node's attachment to a link. This can be a physical interface (such as a network adapter) or a logical interface (such as a tunnel interface).

Note An IPv6 address identifies an interface, not a node. A node is identified by having one or more unicast IPv6 addresses assigned to one of its interfaces.

Understanding IPv6 Addressing

IPv6 uses 128-bit (16 byte) addresses that are expressed in colon-hexadecimal form. For example, in the address 2001:DB8:3FA9:0000:0000:0000:00D3:9C5A, each block of 4-digit hexadecimal numbers represents a 16-bit digit binary number. The eight blocks of four-digit hexadecimal numbers thus equal 8 × 16 = 128 bits in total.

You can shorten hexadecimal-colon addresses by suppressing leading zeros for each block. Using this technique, the representation for the preceding address now becomes 2001:DB8:3FA9:0:0:0:D3:9C5A.

You can shorten hexadecimal-colon addresses even further by compressing contiguous 0 (hex) blocks as double colons ("::"). The address in our example thus shortens to 2001:DB8:3FA9::D3:9C5A. Note that only one double colon can be used per IPv6 address to ensure unambiguous representation.

Understanding IPv6 Prefixes

An IPv6 prefix indicates the portion of the address used for routing (a subnet or a set of subnets as a summarized route) or for identifying an address range. IPv6 prefixes are expressed in a similar fashion as the Classless Inter-Domain Routing (CIDR) notation used by IPv4. For example, 2001:DB8:3FA9::/48 might represent a route prefix in an IPv6 routing table.

In IPv4, CIDR notation can be used to represent individual unicast addresses in addition to routes and subnets. IPv6 prefixes, however, are used only to represent routes and address ranges, not unicast addresses. This is because unlike IPv4, IPv6 does not support variable length subnet identifiers, and the number of high-order bits used to identify a subnet in IPv6 is almost always 64. It is thus redundant to represent the address in our example as 2001:DB8:3FA9::D3:9C5A/64; the /64 portion of the representation is understood.

Understanding IPv6 Address Types

IPv6 supports three different address types:

- Unicast Identifies a single interface within the scope of the address. (The scope of

an IPv6 address is that portion of your network over which this address is unique.) IPv6 packets with unicast destination addresses are delivered to a single interface.

- Multicast Identifies zero or more interfaces. IPv6 packets with multicast destination addresses are delivered to all interfaces listening on the address. (Generally speaking, multicasting works the same way in IPv6 as it does in IPv4.)
- Anycast Identifies multiple interfaces. IPv6 packets with anycast destination addresses are delivered to the nearest interface (measured by routing distance) specified by the address. Currently, anycast addresses are assigned only to routers and can only represent destination addresses.

Note IPv6 address types do not include broadcast addresses as used by IPv4. In IPv6, all broadcast communications are performed using multicast addresses. See Table 29-2 for more information on multicast addresses.

Understanding Unicast Addresses

Unicast addresses are addresses that identify a single interface. IPv6 has several types of unicast addresses:

- Global Unicast Address An address that is globally routable over the IPv6-enabled portion of the Internet. Therefore, the scope of a global address is the entire Internet, and global addresses in IPv6 correspond to public (non-RFC 1918) addresses used in IPv4. The address prefix currently used for global addresses as defined in RFC 3587 is 2000::/3, and a global address has the following structure:
  - The first 48 bits of the address are the global routing prefix specifying your organization's site. (The first three bits of this prefix must be 001 in binary notation.) These 48 bits represent the public topology portion of the address, which represents the collection of large and small Internet Service Providers (ISPs) on the IPv6 Internet, and which is controlled by these ISPs through assignment by the Internet Assigned Numbers Authority (IANA).
  - The next 16 bits are the subnet ID. Your organization can use this portion to specify up to 65,536 unique subnets for routing purposes inside your organization's site. These 16 bits represent the site topology portion of the address, which your organization has control over.
  - The final 64 bits are the interface ID and specify a unique interface within each subnet.
- Link-Local Unicast Address An address that can be used by a node for communicating with neighboring nodes on the same link. Therefore, the scope of a link-local address is the local link on the network; link-local addresses are never forwarded beyond the local link by IPv6 routers. Because link-local addresses are

assigned to interfaces using IPv6 address autoconfiguration, link-local addresses in IPv6 correspond to Automatic Private IP Addressing (APIPA) addresses used in IPv4 (which are assigned from the address range 169.254.0.0/16). The address prefix used for link-local addresses is FE80::/64, and a link-local address has the following structure:

- The first 64 bits of the address are always FE80:0:0:0 (which will be shown as FE80::).
- The last 64 bits are the interface ID and specify a unique interface on the local link.

Link-local addresses can be reused—in other words, two interfaces on different links can have the same address. This makes link-local addresses ambiguous; an additional identifier called the zone ID (or scope ID) indicates to which link the address is either assigned or destined. In Windows 7, the zone ID for a link-local address corresponds to the interface index for that interface. You can view a list of interface indexes on a computer by typing **netsh interface ipv6 show interface** at a command prompt. For more information on the zone ID, see the section titled "Displaying IPv6 Address Settings" later in this chapter.

- Unique Local Unicast Address Because a site-local address prefix can represent multiple sites within an organization, it is ambiguous and not well-suited for intraorganizational routing purposes. Therefore, RFC 4193 currently proposes a new type of address called a unique local unicast address. The scope of this address is global to all sites within the organization, and using this address type simplifies the configuration of an organization's internal IPv6 routing infrastructure. A unique local address has the following structure:
  - The first seven bits of the address are always 1111 110 (binary) and the eighth bit is set to 1, indicating a unique local address. This means that the address prefix is always FD00::/8 for this type of address.
  - The next 40 bits represent the global ID, a randomly generated value that identifies a specific site within your organization.
  - The next 16 bits represent the subnet ID and can be used for further subdividing the internal network of your site for routing purposes.
  - The last 64 bits are the interface ID and specify a unique interface within each subnet.

Note Site-local addresses have been deprecated by RFC 3879 and are replaced by unique local addresses.

Identifying IPv6 Address Types

As Table 29-1 shows, you can quickly determine which type of IPv6 address you are dealing

with by looking at the beginning part of the address—that is, the high-order bits of the address. Tables 29-2 and 29-3 also show examples of common IPv6 addresses that you can recognize directly from their colon-hexadecimal representation.

Table 29-1 Identifying IPv6 Address Types Using High-Order Bits and Address Prefix

ADDRESS TYPE	HIGH-ORDER BITS	ADDRESS PREFIX
Global unicast	001	2000::/3
Link-local unicast	1111 1110 10	FE80::/64
Unique local unicast	1111 1101	FD00::/8
Multicast	1111 1111	FF00::/8

Table 29-2 Identifying Common IPv6 Multicast Addresses

FUNCTION	SCOPE	REPRESENTATION
All-nodes multicast	Interface-local	FF01::1
All-nodes multicast	Link-local	FF02::1
All-routers multicast	Interface-local	FF01::2
All-routers multicast	Link-local	FF02::2
All-routers multicast	Site-local	FF05::2

Table 29-3 Identifying Loopback and Unspecified IPv6 Addresses

FUNCTION	REPRESENTATION
Unspecified address (no address)	::
Loopback address	::1

Note For information on IPv6 address types used by different IPv6 transition technologies, see the section titled “Planning for IPv6 Migration” later in this chapter.

Understanding Interface Identifiers

For all the types of unicast IPv6 addresses described in the preceding sections, the last 64 bits of the address represent the interface ID and are used to specify a unique interface on a local link or subnet. In previous versions of Windows, the interface ID is uniquely determined as follows:

- For link-local addresses, such as a network adapter on an Ethernet segment, the interface ID is derived from either the unique 48-bit MAC-layer (Media Access Control) address of the interface or the unique EUI-64 (Extended Unique Identifier) address of the interface as defined by the Institute of Electrical and Electronics Engineers (IEEE).
- For global address prefixes, an EIU-64–based interface ID creates a public IPv6 address.
- For global address prefixes, a temporary random interface ID creates a temporary address. This approach is described in RFC 3041; you can use it to help provide anonymity for client-based usage of the IPv6 Internet.

In Windows 7, however, the interface ID by default is randomly generated for all types of unicast IPv6 addresses assigned to LAN interfaces.

Note Windows 7 randomly generates the interface ID by default. You can also disable this behavior by typing **netsh interface ipv6 set global randomizedidentifiers=disabled** at a command prompt.

Comparing IPv6 with IPv4

Table 29-4 compares and contrasts the IPv4 and IPv6 addressing schemes.

Table 29-4 IPv4 vs. IPv6 Addressing

FEATURE	IPv4	IPv6
Number of bits (bytes)	32 (4)	128 (16)
Expressed form	Dotted-decimal	Colon-hexadecimal
Variable-length subnets	Yes	No
Public addresses	Yes	Yes (global addresses)
Private addresses	Yes (RFC 1918 addresses)	Yes (unique local addresses)

Autoconfigured addresses for the local link	Yes (APIPA)	Yes (link-local addresses)
Support for address classes	Yes, but deprecated by CIDR	No
Broadcast addresses	Yes	Multicast used instead
Subnet mask	Required	Implicit 64-bit address prefix length for addresses assigned to interfaces

Note For detailed specifications concerning IPv6 addressing, see RFC 4291 at *http://www.ietf.org/rfc/rfc4291.txt*. There are also other differences between IPv4 and IPv6, such as how the headers are structured for IPv4 versus IPv6 packets. For more information, see the white paper "Introduction to IP Version 6" at *http://www.microsoft.com/downloads/details.aspx?FamilyID=CBC0B8A3-B6A4-4952-BBE6-D976624C257C&displaylang=en*.

Understanding IPv6 Routing

Routing is the process of forwarding packets between connected network segments and is the primary function of IPv6. An IPv6 network consists of one or more network segments, also called links or subnets. These links are connected together by IPv6 routers, devices that forward IPv6 packets from one link to another. These IPv6 routers are typically third-party hardware devices, but you can also configure a multihomed Windows Server 2008 computer as an IPv6 router if needed.

How IPv6 Routing Works

The header of an IPv6 packet contains both the source address of the sending host and the destination address of the receiving host. When an IPv6 packet arrives at a host, the host uses its local IPv6 routing table to determine whether to accept the packet or forward it to another host or network.

Each IPv6 node (host or router) has its own IPv6 routing table. A routing table is a collection of routes that store information about IPv6 network prefixes and how they can be reached, either directly or indirectly. On IPv6 hosts such as computers running Windows 7, Windows Vista, or Windows Server 2008, the IPv6 routing table is generated automatically when IPv6 initializes on the system. Local administrators can use the **netsh interface ipv6** commands to manage these tables by viewing them and by manually adding or removing routes. Use of this command is discussed further below.

When an IPv6 packet arrives at a physical or logical network interface on an IPv6 host such as a multihomed Windows Server 2008 computer, the host uses the following process to determine how to forward the packet to its intended destination:

1. The host checks its destination cache to see if there is an entry that matches the destination address in the packet header. If such an entry is found, the host forwards the packet directly to address specified in the destination cache entry and the routing process ends.
2. If the destination cache does not contain an entry that matches the destination address in the packet header, the host uses its local routing table to determine how to forward the packet. Using the routing table, the host determines:
   - **Next-hop address** If the destination address is on the local link, the next-hop address is simply the destination address in the packet header. If the destination address is on a remote link, the next-hop address is the address of a router connected to the local link.
   - **Next-hop interface** This is the physical or logical network interface on the host that should be used to forward the packet to the next-hop address.
3. The host then forwards the packet to the next-hop address using the next-hop interface. The host also updates its destination cache with this information so that subsequent packets sent to the same destination address can be forwarded using the destination cache entry instead of having to use its local routing table.

IPv6 Route Determination Process

In step 2 above, the host determines the next-hop address and next-hop interface by using its local routing table. The details of this process are as follows:

1. For each routing table entry, the first N bits in the route's network prefix are compared with the same bits in the destination address in the packet header, where N is the number of bits in the route's prefix length. If these bits match, the route is determined to be a match for the destination.
2. The list of all matching routes is compiled. If only one matching route is found, this route is chosen and the route determination process is ended.
3. If multiple matching routes are found, the matching route having the largest prefix length is chosen and the route determination process is ended.
4. If multiple matching routes having the largest prefix length are found, the matching route having the lowest metric is chosen and the route determination process is ended.
5. If multiple matching routes having the largest prefix length and lowest metric are found, one of these routes is selected and the route determination process is ended.

The effective result of the above IPv6 route determination process is as follows:

1. If a route can be found that matches the entire destination address in the packet header, then the next-hop address and interface specified in this route are used to forward the packet.
2. If the above is not found, the most efficient (lowest metric) route that has the longest prefix length matching the destination address is used to forward the packet.
3. If neither of the above are found, the packet is forwarded using the default route (network prefix ::/0).

IPv6 Routing Table Structure

IPv6 routing tables can contain four different types of routing table entries (routes):

- Directly-attached network routes These typically have 64-bit prefixes and identify adjacent links (network segments connected to the local segment via one router).
- Remote network routes These have varying prefixes and identify remote links (network segments connected to the local segment via several routers).
- Host routes These have 128-bit prefixes and identify a specific IPv6 node.
- Default route This uses the prefix ::/0 and is used to forward packets when a network or host route cannot be determined.

On a Windows 7, Windows Vista, or Windows Server 2008 computer, you can use the netsh interface ipv6 show route command to display the IPv6 routing table entries. The following is a sample routing table from a domain-joined Windows 7 computer that has

a single LAN network adapter and where there are no IPv6 routers on the attached subnet and no other configured network connections:

```
Publish  Type      Met  Prefix                    Idx  Gateway/Interface Name
-------  --------  ---  ------------------------  ---  -----------------------
No       Manual    256  ::1/128                     1  Loopback Pseudo-Interface 1
No       Manual    256  fe80::/64                  15  Teredo Tunneling Pseudo-Interface
No       Manual    256  fe80::/64                  12  Local Area Connection
No       Manual    256  fe80::100:7f:fffe/128      15  Teredo Tunneling Pseudo-Interface
No       Manual    256  fe80::5efe:172.16.11.131/128   14  isatap.{9D607D7D-0703-4E67-
82ED-9A8206377C5C}
No       Manual    256  fe80::5da9:fa1d:2575:c766/128   12  Local Area Connection
No       Manual    256  ff00::/8                    1  Loopback Pseudo-Interface 1
No       Manual    256  ff00::/8                   15  Teredo Tunneling Pseudo-Interface
No       Manual    256  ff00::/8                   12  Local Area Connection
```

Each route in this table is specified using the following fields:

- Publish If Yes, the route is advertised in a routing Advertisement message; otherwise No.
- Type If Autoconf, the route was configured automatically using the IPv6 routing protocol; if Manual, the route has been configured by the operating system or an application.
- Met Indicates the metric for the route. For multiple routes having the same prefix, the lower the metric, the better the match.
- Prefix Specifies the address prefix for the route.
- Idx Specifies the index of the network interface over which packets matching the route's address prefix are reachable. To display a list of interfaces and their indices, use the **netsh interface ipv6 show interface** command.
- Gateway/Interface Name For directly-attached network routes, specifies the name of the interface; for remote network routes, specifies the next-hop address of the route.

Note For more information about IPv6 routing and routing tables, see the Cable Guy article titled ""Understanding the IPv6 Routing Table" at http://technet.microsoft.com/en-ca/library/bb878115.aspx.

Understanding ICMPv6 Messages

Internet Control Message Protocol (ICMP) for IPv4 (ICMPv4) is used in IPv4 networks to allow nodes to send and respond to error messages and informational messages. For example, when a source node uses the ping command to send ICMP Echo Request messages (ICMP type 8 messages) to a destination node, the destination node can respond with ICMP Echo

messages (ICMP type 0 messages) indicating its presence on the network.

On IPv6 networks, ICMP for IPv6 (ICMPv6) fulfills the same functions that ICMPv4 does on IPv4 networks—namely, to provide a mechanism for exchanging error messages and informational messages. ICMPv6 also provides information messages for the following:

- Neighbor Discovery (ND) The process by which hosts and routers discover each other on the network so that they can communicate at the data-link layer. (Network Discovery serves the same purpose as ARP does in IPv4 networks.)
- Multicast Listener Discovery (MLD) The process by which membership in multicast groups is determined and maintained.

Note For more information about Neighbor Discovery, see the section titled "Understanding Neighbor Discovery" later in this chapter. For more information about ICMPv6 message types and header formats, and about Multicast Listener Discovery, see the white paper "Introduction to IP Version 6" at *http://www.microsoft.com/downloads/details.aspx?FamilyID=CBC0B8A3-B6A4-4952-BBE6-D976624C257C&displaylang=en*.

Understanding Neighbor Discovery

Neighbor Discovery (ND) is the process by which nodes on an IPv6 network can communicate with each other by exchanging frames at the data-link layer. ND performs the following functions on an IPv6 network:

- Enables IPv6 nodes (IPv6 hosts and IPv6 routers) to resolve the link-layer address of a neighboring node (a node on the same physical or logical link)
- Enables IPv6 nodes to determine when the link-layer address of a neighboring node has changed
- Enables IPv6 nodes to determine whether neighboring nodes are still reachable
- Enables IPv6 routers to advertise their presence, on-link prefixes, and host configuration settings
- Enables IPv6 routers to redirect hosts to more optimal routers for a specific destination
- Enables IPv6 hosts to discover addresses, address prefixes, and other configuration settings
- Enables IPv6 hosts to discover routers attached to the local link

To understand how ND works, it helps to first compare it with the similar processes used in IPv4. In IPv4, you use three separate mechanisms to manage node-to-node communication:

- Address Resolution Protocol (ARP) A data-link layer protocol that resolves IPv4 addresses assigned to interfaces to their corresponding MAC-layer addresses. This

enables network adapters to receive frames addressed to them and send response frames to their source. For example, before a host can send a packet to a destination host whose IPv4 address is 172.16.25.3, the sending host first needs to use ARP to resolve this destination address (if the host is on the same LAN) or the IP address of the local gateway (if the host is on a different LAN) to its corresponding 48-bit MAC address (such as 00-13-20-08-A0-D1).

- ICMPv4 Router Discovery These ICMPv4 messages enable routers to advertise their presence on IPv4 networks and enable hosts to discover the presence of these routers. When Router Discovery is enabled on a router, the router periodically sends Router Advertisements to the all-hosts multicast address (224.0.0.1) to indicate to hosts on the network that the router is available. When Router Discovery is enabled on hosts, the hosts can send Router Solicitations to the all-routers multicast address (224.0.0.2) to obtain the address of the router and assign this address as the host's default gateway.
- ICMPv4 Redirect Routers use these ICMPv4 messages to inform hosts of more optimal routers to use for specific destinations. ICMPv4 Redirect messages are needed because hosts typically cannot determine the best router on their subnet to send remote traffic for a given destination.

On IPv4 networks, these three mechanisms enable nodes on a network segment to communicate on a link. On IPv6 networks, these three mechanisms are replaced by the five ICMPv6 message types shown in Table 29-5.

Table 29-5 ICMPv6 Message Types Used for Neighbor Discovery

MESSAGE TYPE	ICMPV6 TYPE	DESCRIPTION
Router Solicitation	133	Sent by IPv6 hosts to the link-local scope all-routers multicast address (FF02::2) to discover IPv6 routers present on the local link.
Router Advertisement	134	Sent periodically by IPv6 routers to the link-local scope all-nodes multicast address (FF02::1), or sent to the unicast address of a host in response to receiving a Router Solicitation message from that host. (Windows Vista and later use multicast for optimization.) Router Advertisement messages provide hosts with the information needed to determine link prefixes, link MTU, whether or not to use DHCPv6 for address autoconfiguration, and lifetime for autoconfigured addresses.

Neighbor Solicitation	135	Sent by IPv6 nodes to the solicited-node multicast address of a host to discover the link-layer address of an IPv6 node, or sent to the unicast address of the host to verify the reachability of the host.
Neighbor Advertisement	136	Sent by an IPv6 node to the unicast address of a host in response to receiving a Neighbor Solicitation message from the host, or sent to the link-local scope all-nodes multicast address (FF02::1) to inform neighboring nodes of changes to the host's link-layer addresses.
Redirect	137	Sent by an IPv6 router to the unicast address of a host to inform the host of a more optimal first-hop address for a specific destination.

Note The solicited-node multicast address, which is used as the destination address for ICMPv4 Neighbor Solicitation messages (ICMPv6 type 135 messages) when address resolution is being performed, is a special type of multicast address composed of the prefix FF02::1:FF00:0/104 followed by the last 24 bits of the IPv6 address that is being resolved. IPv6 nodes listen on their solicited-node multicast addresses. The advantage of using this multicast address for address resolution in IPv6 is that typically only the targeted host is disturbed on the local link. By contrast, the ARP messages used in IPv4 for address resolution queries are sent to the MAC-layer broadcast address, which disturbs all hosts on the local segment.

Understanding Address Autoconfiguration

On IPv4 networks, addresses can be assigned to hosts in three ways:

- Manually using static address assignment
- Automatically using DHCP, if a DHCP server is present on the subnet (or a DHCP relay agent configured on the subnet)
- Automatically using Automatic Private IP Addressing (APIPA), which randomly assigns the host an address from the range 169.254.0.0 to 169.254.255.255 with subnet mask 255.255.0.0

On IPv6 networks, static addresses are generally assigned only to routers and sometimes servers, but hardly ever to client computers. Instead, IPv6 addresses are almost always assigned automatically using a process called address autoconfiguration. Address autoconfiguration can work in three ways: stateless, stateful, or both. Stateless address autoconfiguration is based on the receipt of ICMPv6 Router Advertisement messages. Stateful

address autoconfiguration, on the other hand, uses DHCPv6 to obtain address information and other configuration settings from a DHCPv6 server.

Note The DHCP Server service of Windows Server 2008 supports DHCPv6. The DHCP Server service of Windows Server 2003 does not support DHCPv6.

All IPv6 nodes (hosts and routers) automatically assign themselves link-local addresses (addresses having the address prefix FE80::/64); this is done for every interface (both physical and logical) on the node. (6to4 interfaces are an exception—they might not have link-local addresses automatically assigned.) These autoconfigured link-local addresses can be used only to reach neighboring nodes (nodes on the same link). When specifying one of these addresses as a destination address, you might need to specify the zone ID for the destination. In addition, link-local addresses are never registered in DNS servers.

Note Manual assignment of IPv6 addresses is generally needed only for IPv6 routers and for some servers. You can configure a Windows 7 computer with multiple interfaces to be used as a router. For more information on configuring IPv6 routers, see the Cable Guy article titled "Manual Configuration for IPv6" at *http://technet.microsoft.com/en-us/library/bb878102.aspx*. For a description of the IPv6 routing table, see the Cable Guy article titled "Understanding the IPv6 Routing Table" at *http://technet.microsoft.com/en-us/library/bb878115.aspx*.

An autoconfigured IPv6 address can be in one or more of the states shown in Table 29-6.

Table 29-6 Possible States for an Autoconfigured IPv6 Address

STATE	DESCRIPTION
Tentative	The uniqueness of the address is still being verified using duplicate address detection.
Valid	The address is unique and can now send and receive unicast IPv6 traffic until the Valid Lifetime expires.
Preferred	The address can be used for unicast traffic until the Preferred Lifetime expires.
Deprecated	The address can still be used for unicast traffic during existing communication sessions, but its use is discouraged for new communication sessions.
Invalid	The Valid Lifetime for the address has expired and it can no longer be used

for unicast traffic.

Note The Valid and Preferred lifetime for stateless autoconfigured IPv6 addresses is included in the Router Solicitation message.

For detailed descriptions of how address autoconfiguration, address resolution, router discovery, redirect, duplicate address detection, and neighbor unreachability detection processes are performed, see the white paper "Introduction to IP Version 6" at *http://www.microsoft.com/downloads/details.aspx?FamilyID=CBC0B8A3-B6A4-4952-BBE6-D976624C257C&displaylang=en*.

Note To display the state for each autoconfigured IPv6 address on a Windows 7 computer, open a command prompt and type **netsh interface ipv6 show addresses** at a command prompt.

Understanding Name Resolution

The Domain Name System (DNS) is fundamental to how name resolution works on both IPv4 and IPv6 networks. On an IPv4 network, host (A) records are used by name servers (DNS servers) to resolve fully qualified domain names (FQDNs) like server1.contoso.com into their associated IP addresses in response to name lookups (name queries) from DNS clients. In addition, reverse lookups—in which IP addresses are resolved into FQDNs—are supported by using pointer (PTR) records in the in-addr.arpa domain.

Name resolution works fundamentally the same way with IPv6, with the following differences:

- Host records for IPv6 hosts are AAAA ("quad-A") records, not A records.
- The domain used for reverse lookups of IPv6 addresses is ip6.arpa, not in-addr.arpa.

Note The enhancements to the Domain Name System that make IPv6 support possible are described in the draft standard RFC 3596 at *http://www.ietf.org/rfc/rfc3596.txt*.

Understanding Name Queries

Because the dual-layer TCP/IP stack in Windows 7 means that both IPv4 and IPv6 are enabled by default, DNS name lookups by Windows 7 client computers can involve the use of both A and AAAA records. (This is true only if your name servers support IPv6, which is the case with the DNS Server role for Windows Server 2008 and Windows Server 2003.) By default, the DNS client component in Windows 7 uses the following procedure when performing a name lookup using a particular interface:

1. The client computer checks to see whether it has a non-link-local IPv6 address assigned to the interface. If it has no non-link-local addresses assigned, the client sends a single name lookup to the name server to query for A records and does not query for AAAA records. If the only non-link-local address assigned to the interface is a Teredo address, the client again does not query for AAAA records. (The Teredo client in Windows Vista and later has been explicitly built not to automatically perform AAAA lookups or register with DNS to prevent overloading of DNS servers.)
2. If the client computer has a non-link-local address assigned to the interface, the client sends a name lookup to query for A records.
    - If the client then receives a response to its query (not an error message), it follows with a second lookup to query for AAAA records.
    - If the client receives no response or receives any error message (except for Name Not Found), it does not send a second lookup to query for AAAA records.

Note Because an interface on an IPv6 host typically has multiple IPv6 addresses, the process by which source and address selection works during a name query is more complex than when DNS names are resolved by IPv4 hosts. For a detailed description of how source and address selection works for IPv6 hosts, see the Cable Guy article titled "Source and Destination Address Selection for IPv6" at *http://technet.microsoft.com/en-us/library/bb877985.aspx*. For additional information on DNS behavior in Windows 7 and Windows Vista, see "Domain Name System Client Behavior in Windows Vista" at *http://technet.microsoft.com/en-us/library/bb727035.aspx*. For information about the different types of IPv6 addresses usually assigned to an interface, see the section titled "Configuring and Troubleshooting IPv6 in Windows Vista" later in this chapter.

Note Issues have arisen with poorly configured DNS name servers on the Internet. These issues, which are described in RFC 4074 (*http://www.ietf.org/rfc/rfc4074.txt*), do not cause problems on Windows Vista or later because Microsoft has altered the DNS client behavior specifically to compensate for them. However, administrators of DNS servers should make sure these issues are fixed, because they can cause problems with DNS name resolution

for most IPv6 networking stacks, including stacks found in legacy Windows platforms such as Windows XP.

Understanding Name Registration

DNS servers running Windows Server 2003 can dynamically register both A and AAAA records for Windows 7 client computers. Dynamic registration of DNS records simplifies the job of maintaining name resolution on networks running the Active Directory directory service. When a Windows 7 client computer starts up on a network, the DNS Client service tries to register the following records for the client:

- A records for all IPv4 addresses assigned to all interfaces configured with the address of a DNS server
- AAAA records for all IPv6 addresses assigned to all interfaces configured with the address of a DNS server
- PTR records for all IPv4 addresses assigned to all interfaces configured with the address of a DNS server

Note AAAA records are not registered for link-local IPv6 addresses that have been assigned to interfaces using address autoconfiguration.

PTR Records and IPv6

Windows 7 client computers do not try to register PTR records for IPv6 addresses assigned to interfaces on the computer. If you want to enable clients to perform reverse lookups for Windows 7 computers using IPv6, you must manually create a reverse lookup zone for the ip6.arpa domain on your DNS servers and then manually add PTR records to this zone. For detailed steps on how to do this, see "IPv6 for MicrosoftWindows: Frequently Asked Questions" at *http://www.microsoft.com/technet/network/ipv6/ipv6faq.mspx*.

However, PTR records for reverse lookups using IPv6 are not often used, because the namespace for reverse queries is formed by using each hexadecimal digit in the colon-hexadecimal representation of an IPv6 address as a separate level in the reverse domain hierarchy. For example, the PTR record associated with the IPv6 address 2001:DB8::D3:00FF:FE28:9C5A, whose full representation is 2001:0DB8:0000:0000:00D3:00FF:FE28:9C5A, would be expressed as A.5.C.9.8.2.E.F.F.F.0.0.3.D.0.0.0.0.0.0.0.0.0.0.0.8.B.D.0.1.0.0.2.IP6.ARPA. The performance cost of resolving such a representation would generally be too high for most DNS server implementations.

By default, DNS servers running Windows Server 2003 do not listen for DNS traffic sent over IPv6. To enable these DNS servers to listen for IPv6 name registrations and name lookups, you must first configure the servers using the **dnscmd /config /EnableIPv6 1** command. By default, DNS servers running Windows Server 2008 listen for DNS traffic sent over IPv6. You must then configure each Windows 7 client computer with the unicast IPv6 addresses of your DNS servers using DHCPv6, the properties of the Internet Protocol Version 6 (TCP/IPv6) component in the Network Connections folder, or the **netsh interface ipv6 add dns interface=*NameOrIndex* address=*IPv6Address* index=*PreferenceLevel*** command. (DHCP servers running Windows Server 2003 do not support stateful address assignment using DHCPv6.)

> Note For more information on enabling Windows Server 2003 DNS server support for IPv6, see Chapter 9, "Windows Support for DNS" in the online book TCP/IP Fundamentals for Microsoft Windows, which you can download from *http://www.microsoft.com/downloads/details.aspx?FamilyID=c76296fd-61c9-4079-a0bb-582bca4a846f &displaylang=en*. For further details on the DNS name query and registration behavior in Windows 7 and Windows Vista, see the article titled "Domain Name System Client Behavior in Windows Vista" on Microsoft TechNet at *http://technet.microsoft.com/en-us/library/bb727035.aspx*.

IPv6 Enhancements in Windows 7

The TCP/IP networking stack in the Windows XP and Windows Server 2003 platforms had a dual-stack architecture that used separate network and framing layers for IPv4 and IPv6 based on separate drivers: Tcpip.sys and Tcpip6.sys. Only the transport and framing layers for IPv4 were installed by default, and adding support for IPv6 involved installing an additional IPv6 protocol component through the Network Connections folder.

By contrast, in Windows 7, Windows Vista and Windows Server 2008, the TCP/IP stack has been completely redesigned and now uses a dual IP layer architecture in which both IPv4 and IPv6 share common transport and framing layers. In addition, IPv6 is installed and enabled by default in these new platforms to provide out-of-the-box support for new features such as the Windows Meeting Space application, which uses only IPv6. Finally, the dual IP layer architecture means that all of the performance enhancements of the Next Generation TCP/IP stack that apply to IPv4 also apply to IPv6. These performance enhancements include Compound TCP, Receive Window Auto-Tuning, and other enhancements that can dramatically improve performance in high-latency, high-delay, and high-loss networking environments.

Note For more information about the performance enhancements in the Next Generation TCP/IP stack, see Chapter 26, "Configuring Windows Networking."

Summary of IPv6 Enhancements in Windows 7

Windows 7 builds upon the many IPv6 enhancements introduced earlier in Windows Vista and Windows Server 2008. These earlier enhancements included:

- Dual IP layer architecture A new TCP/IP stack architecture that uses the same transport and framing layers for both IPv4 and IPv6.
- Enabled by default Both IPv4 and IPv6 are installed and enabled by default, with the stack giving preference to IPv6 when appropriate without impairing the performance of IPv4 communications on the network. For example, if a DNS name query returns both an IPv4 and IPv6 address for a host, the client will try to use IPv6 first for communicating with the host. This preference also results in better network performance for IPv6-enabled applications.
- User interface configuration support In addition to being able to configure IPv6 settings from the command line using the netsh interface ipv6 command context, you can also configure them in Windows 7 using the user interface. For more information, see the section titled "Configuring IPv6 in Windows 7 Using the User Interface" later in this chapter.
- Full IPsec support IPv6 support in previous versions of Windows offered only limited support for IPsec protection of network traffic. In Windows 7 and Windows Vista, however, IPsec support for IPv6 is the same as for IPv4, and you can configure IPsec connection security rules for IPv6 the same as for IPv4, using the Windows Firewall With Advanced Security console. For more information on configuring IPsec in Windows 7, see Chapter 27, "Configuring Windows Firewall and IPsec."
- LLMNR support The implementation of IPv6 in Windows 7 and Windows Vista supports Link-Local Multicast Name Resolution (LLMNR), a mechanism that enables IPv6 nodes on a single subnet to resolve each other's names in the absence of a DNS server. LLMNR works by having nodes send multicast DNS name queries instead of unicast queries. Windows 7 and Windows Vista computers listen by default for multicast LLMNR traffic, which eliminates the need to perform local subnet name resolution using NetBIOS over TCP/IP when no DNS server is available. LLMNR is defined in RFC 4795.
- MLDv2 support The implementation of IPv6 in Windows 7 and Windows Vista supports Multicast Listener Discovery (MLD) version 2 (MLDv2), a mechanism described in RFC 3810 that enables IPv6 hosts to register interest in source-specific multicast traffic with local multicast routers by specifying an include list (to indicate specific source addresses of interest) or an exclude list (to exclude unwanted source

addresses).

- DHCPv6 support The DHCP Client service in Windows 7 and Windows Vista supports Dynamic Host Configuration Protocol for IPv6 (DHCPv6) as defined in RFCs 3736 and 4361. This means that Windows 7 and Windows Vista computers can perform both stateful and stateless DHCPv6 configuration on a native IPv6 network.
- IPV6CP support The built-in remote access client component in Windows 7 and Windows Vista supports IPv6 Control Protocol (IPV6CP) (RFC 5072) to configure IPv6 nodes on a Point-to-Point Protocol (PPP) link. This means that native IPv6 traffic can be sent over PPP-based network connections such as dial-up connections or broadband PPP over Ethernet (PPPoE) connections to an Internet Service Provider (ISP). IPV6CP also supports Layer 2 Tunneling Protocol (L2TP), and for Windows Vista with Service Pack 1 or later, Secure Socket Tunneling Protocol (SSTP)–based Virtual Private Network (VPN) connections. For more information on IPV6CP support in Windows 7, see Chapter 28, "Connecting Remote Users and Networks."
- Random interface IDs By default, Windows 7 and Windows Vista generate random interface IDs for nontemporary autoconfigured IPv6 addresses, including both public addresses (global addresses registered in DNS) and link-local addresses. For more information, see the section titled "Disabling Random Interface IDs" later in this chapter.
- Literal IPv6 addresses in URLs Windows 7 and Windows Vista support RFC 2732–compliant literal IPv6 addresses in URLs by using the new WinINet API support in Microsoft Internet Explorer 7.0. This can be a useful feature for troubleshooting Internet connectivity with IPv6-enabled Web servers.
- New Teredo Behavior The Teredo client in Windows 7 and Windows Vista remains dormant (inactive) until it spins up (is activated by) an IPv6-enabled application that tries to use Teredo. In Windows 7 and Windows Vista, three things can bring up Teredo: an application trying to communicate using a Teredo address (the outbound instantiated scenario); a listening application that has the Edge Traversal rule enabled in Windows Firewall (any IPv6-enabled applications that need to use Teredo can easily do so by setting the Edge Traversal flag using the Windows Firewall APIs); and the *NotifyStableUnicastIpAddressTable* IP Helper API. For more information about Windows Firewall rules, see Chapter 27.

In addition to these earlier enhancements, Windows 7 and Windows Server 2008 R2 introduce the following new IPv6 enhancements:

- IP-HTTPS This stands for Internet Protocol over Secure Hypertext Transfer Protocol (IP over HTTPS), a new protocol that enables hosts located behind a proxy or firewall to establish connectivity by tunneling IP traffic inside an HTTPS tunnel. HTTPS is used instead of HTTP so that proxy servers will be prevented from looking inside the data stream and terminate the connection if traffic seems anomalous. Note that HTTPS

does not providing data security—you must use IPsec to provide data security for an IP-HTTPS connection.

In the Windows 7 implementation of DirectAccess described below, IT-HTTPS is used whenever a firewall or proxy server blocks a client computer from using 6to4 or Teredo to establish an IPv6-over-IPv4 tunnel with an IPv6-enabled DirectAccess server on the corporate intranet.

For more information about IP-HTTPS, see the *IP over HTTPS (IP-HTTPS) Tunneling Protocol Specification* on MSDN at *http://msdn.microsoft.com/en-us/library/dd358571(PROT.10).aspx*.

- DirectAccess This is a new feature of Windows 7 and Windows Server 2008 R2 that provides users with the experience of being seamlessly connected to the corporate network whenever they have Internet access. Using DirectAccess, remote users who attempt to access corporate intranet resources such as e-mail servers, shared folders, or intranet Web sites can access these resources without the need of having to connect to a virtual private network (VPN). By providing users with the same connectivity experience both inside and outside of the office, DirectAccess can increase the productivity of your mobile users. DirectAccess also enables administrators to keep the computers of mobile uses in a managed state even when they are off-site by allowing Group Policy changes to be propagated over the Internet.

DirectAccess is implemented as a client/server architecture in which remote IPv6-enabled client computers communicate with IPv6-enabled servers located on the corporate network. DirectAccess can work over existing IPv4 networks such as the public IPv4 Internet by leveraging IPv4/v6 transition technologies such as 6to4, Teredo and ISATAP. DirectAccess also supports native IPv6 connectivity for clients that have been assigned native IPv6 addresses.

DirectAccess uses IPsec tunneling to provide security for authentication and resource access. DirectAccess can be implemented in different ways ranging from providing client computers with secure access to intranet resources via an IPv6-enabled IPsec gateway to providing them with secure end-to-end connectivity with each IPv6-enabled application server located on the intranet. DirectAccess requires the use of IPv6 so that client computers can have globally-routable addresses.

For more information about DirectAccess, see *Chapter 26 Configuring Windows Networking* in this Resource Kit. Also see the *Technical Overview of DirectAccess in Windows 7 and Windows Server 2008 R2*, which can be obtained from the Microsoft Download Center at *http://www.microsoft.com/downloads/*.

How It Works: Teredo Behavior in Windows 7 and Windows Vista

Teredo is default-enabled but inactive in both workgroup and domain scenarios.

Teredo becomes active in two main scenarios:

- An application tries to communicate with a Teredo address (for example, by using a URL with a Teredo address in a Web browser). This is outbound-initiated traffic, and Teredo will go dormant again after 60 minutes of inactivity. The host firewall will allow only incoming Teredo traffic corresponding to the specific outbound request, ensuring that system security isn't compromised. This is really no different than how any outbound-initiated traffic works with the host firewall with IPv4. (In other words, all outbound traffic is allowed by default, and a state table allows responses that match the outgoing requests.)
- An application or service is authorized to use Teredo with the advanced Windows Firewall Edge Traversal flag. If an application has the Edge Traversal option, it is allowed to receive any incoming traffic over Teredo from any source (such as unsolicited traffic). Windows Meeting Space and Remote Assistance automatically set this flag for themselves, but users can do it manually for other Windows services if they prefer, such as with a web service.

Michael Surkan

Program Manager for TCP and IPv6

Configuring and Troubleshooting IPv6 in Windows 7

Although IPv6 is designed to allow IPv6-enabled nodes such as Windows 7 computers to automatically configure their interfaces with link-local addresses, these autoconfigured addresses are not registered in DNS servers and can be used only for communicating with other nodes on the local link. Alternatively, by using a DHCPv6 server, you can automatically assign global, site-local, or unique local IPv6 addresses to IPv6-enabled interfaces of link-attached nodes. This is the preferred scenario for end-to-end IPv6 connectivity in enterprises that have a native IPv6-only network infrastructure.

However, you can also use two methods to configure IPv6 settings manually on Windows 7 computers:

- Using the new IPv6 graphical user interface
- Using the **netsh interface ipv6** command context

In addition, it is important to understand the different kinds of IPv6 addresses assigned to Windows 7 computers so that you can troubleshoot IPv6 connectivity when problems arise.

Displaying IPv6 Address Settings

To display the IPv4 and IPv6 address configuration of the local computer, open a command prompt window and type **ipconfig /all** at a command prompt. The following is an example of

the information displayed by running this command on a domain-joined Windows 7 computer with a single LAN network adapter, no IPv6 routers on the attached subnet, and no other configured network connections:

```
Windows IP Configuration

   Host Name . . . . . . . . . . . . : KBERG-PC
   Primary Dns Suffix  . . . . . . . : contoso.com
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : contoso.com
   Description . . . . . . . . . . . : Broadcom NetXtreme 57xx Gigabit Controller
   Physical Address. . . . . . . . . : 00-13-D4-C2-50-F5
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::3530:6107:45a2:a92c%8(Preferred)
   IPv4 Address. . . . . . . . . . . : 172.16.11.13(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Tuesday, March 17, 2009 9:01:24 AM
   Lease Expires . . . . . . . . . . : Wednesday, March 25, 2009 9:01:29 AM
   Default Gateway . . . . . . . . . : 172.16.11.1
   DHCP Server . . . . . . . . . . . : 172.16.11.32
   DHCPv6 IAID . . . . . . . . . . . : 201331668
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-11-50-8C-A7-00-17-31-C5-D2-8E
   DNS Servers . . . . . . . . . . . : 172.16.11.32
   NetBIOS over Tcpip. . . . . . . . : Enabled

Tunnel adapter isatap.{9D607D7D-0703-4E67-82ED-9A8206377C5C}:

   Media State . . . . . . . . . . . : Enabled
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft ISATAP Adapter
   Physical Address. . . . . . . . . : 00-00-00-00-00-00-00-E0
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::5efe:172.16.11.13%9(Preferred)
   Default Gateway . . . . . . . . . :
   DNS Servers . . . . . . . . . . . : 172.16.11.32
   NetBIOS over Tcpip. . . . . . . . : Disabled

Tunnel adapter Teredo Adapter:
```

```
Media State . . . . . . . . . . . : Enabled
Connection-specific DNS Suffix  . :
Description . . . . . . . . . . . : Teredo Tunneling Pseudo-Interface
Physical Address. . . . . . . . . : 02-00-54-55-4E-01
DHCP Enabled. . . . . . . . . . . : No
Autoconfiguration Enabled . . . . : Yes
IPv6 Address. . . . . . . . . . . : 2001:0:4136:e37c:4e8:3426:7c94:fffe(Preferred)
Link-local IPv6 Address . . . . . : fe80::4e8:3426:53ef:f4f2%10(Preferred)
Default Gateway . . . . . . . . . : ::
NetBIOS over Tcpip. . . . . . . . : Disabled
```

The preceding command output displays three interfaces on this computer:

- Local Area Connection (the installed network adapter)
- ISATAP tunneling interface
- Teredo tunneling interface

The Local Area Connection interface is an Ethernet network adapter and has both an IPv4 address (172.16.11.13) assigned by DHCP and a link-local IPv6 address (fe80::3530:6107:45a2:a92c) that has been automatically assigned using IPv6 address autoconfiguration. (You can recognize the link-local address by its address prefix, FE80::/64.)

The "%8" appended to this address is the zone ID (or scope ID) that indicates which connected portion of the network the computer resides on. This zone ID corresponds with the interface index for the Local Area Connection interface. To view a list of interface indexes on a computer, type **netsh interface ipv6 show interface** at a command prompt. For the example computer, the output of this command is:

```
Idx  Met   MTU        State       Name
---  ---  -----  -----------  ------------------
  1   50  4294967295 connected    Loopback Pseudo-Interface 1
  9   25  1280       connected    isatap.{9D607D7D-0703-4E67-82ED-9A8206377C5C}
 10   10  1280       connected    Teredo Tunneling Pseudo-Interface
  8   20  1500       connected    Local Area Connection
```

Here the Idx column indicates the interface index. The zone ID might be needed when testing network connectivity with this computer from other computers using the ping and tracert commands. See the section titled "Troubleshooting IPv6 Connectivity" later in this chapter for more information.

Returning to the output of the ipconfig /all command, the state of the link-local address assigned to the LAN connection is Preferred, which indicates a valid IPv6 address you can use to send and receive unicast IPv6 traffic.

The ISATAP tunneling interface has an autoconfigured link-local address of fe80::5efe:172.16.11.13. The format for an ISATAP address is:

- The first 64 bits are a unicast prefix that can be a link-local, global, or unique local unicast IPv6 address prefix. This example uses the link-local address prefix because no ISATAP router is present on the network. This means that the resulting ISATAP address can be used only for communicating with other ISATAP hosts on the IPv4 network, and this ISATAP address is not registered in DNS servers.
- The next 32 bits are either 0:5EFE (for a private IPv4 address) or 200:5EFE (for a public IPv4 address) in an ISATAP address. (RFC 4214 also allows 100:5EFE and 300:5EFE in this portion of an ISATAP address.)
- The final 32 bits consist of the 32-bit IPv4 address of the host in dotted-decimal form (172.16.11.13 in this example).

Note For more information on ISATAP addressing, see the white paper "IPv6 Transition Technologies" at *http://www.microsoft.com/downloads/details.aspx?FamilyID=afe56282-2903-40f3-a5ba-a87bf92c096d &displaylang=en* and the white paper "Intra-site Automatic Tunnel Addressing Protocol Deployment Guide" at *http://www.microsoft.com/downloads/details.aspx?FamilyID=0f3a8868-e337-43d1-b271-b8c8702344cd &displaylang=en*. See also the section titled "Understanding ISATAP" later in this chapter.

The Teredo tunneling pseudo-interface displays the IPv6 address of the Teredo client as 2001:0:4136:e37c:4e8:3426:53ef:f4f2. The format for a Teredo client address is:

- The first 32 bits are always the Teredo prefix, which is 2001::/32.
- The next 32 bits contain the public IPv4 address of the Teredo server that helped in the configuration of this Teredo address (here 4136:E37C hexadecimal, which converts to 65.54.227.124 in dotted-decimal format). By default, the Teredo client in Windows 7, Windows Vista and Windows Server 2008 automatically tries to determine the IPv4 addresses of Teredo servers by resolving the name teredo.ipv6.microsoft.com.
- The next 16 bits are reserved for various Teredo flags.
- The next 16 bits contain an obscured version of the external UDP port number that corresponds to all Teredo traffic for this Teredo client. (The external UDP port number is obscured XORing it with 0xFFFF, and in this example 0x3426 XOR 0xFFFF = 0xCBD9 or decimal 52185, meaning UDP port 52185.)
- The final 32 bits contain an obscured version of the external IPv4 address that corresponds to all Teredo traffic for this Teredo client. (The external IPv4 address is obscured, XORing it with 0xFFFF FFFF, and in this example is 0x7C94 FFFE XOR 0xFFFF FFFF = 0x836B 0001 or dotted-decimal 131.107.0.1.)

Note IANA has allocated the IPv6 address prefix 2001::/32 for Teredo as of January 2006. (See RFC 4830 at *http://www.rfc-editor.org/rfc/rfc4380.txt* for details.) Windows XP–based clients originally used the 3FFE:831F::/32 Teredo prefix. Windows XP–based clients with the Microsoft Security Bulletin MS06-064 at *http://www.microsoft.com/technet/security/Bulletin/MS06-064.mspx* now use the 2001::/32 prefix.

Another way to display the IPv6 settings on a Windows 7 computer is to type the **netsh interface ipv6 show address** command. The results for the computer in the preceding example are:

```
Interface 1: Loopback Pseudo-Interface 1

Addr Type  DAD State   Valid Life Pref. Life Address
---------  ----------- ---------- ---------- ------------------------
Other      Preferred     infinite   infinite ::1

Interface 9: isatap.{9D607D7D-0703-4E67-82ED-9A8206377C5C}

Addr Type  DAD State   Valid Life Pref. Life Address
---------  ----------- ---------- ---------- ------------------------
Other      Preferred     infinite   infinite fe80::5efe:172.16.11.13%9

Interface 10: Teredo Tunneling Pseudo-Interface

Addr Type  DAD State   Valid Life Pref. Life Address
---------  ----------- ---------- ---------- ------------------------
Public     Preferred     infinite   infinite 2001:0:4136:e37c:1071:3426:31d2:bfce
Other      Preferred     infinite   infinite fe80::1071:3426:31d2:bfce%10

Interface 8: Local Area Connection

Addr Type  DAD State   Valid Life Pref. Life Address
---------  ----------- ---------- ---------- ------------------------
Other      Preferred     infinite   infinite fe80::3530:6107:45a2:a92c%8
```

Note An advantage of displaying IPv6 address settings using the netsh interface ipv6 show address command instead of ipconfig is that you can execute Netsh.exe commands remotely against a targeted computer by using the –r RemoteComputerName option.

For more information on how to use ipconfig, Netsh.exe, and other tools to display IPv6 configuration information, see the article "Using Windows Tools to Obtain IPv6 Configuration Information" on Microsoft TechNet at *http://technet.microsoft.com/en-us/library/bb726952.aspx*.

Direct from the Source: Explanation of Teredo States

With netsh int teredo show state, you can see the current state of Teredo, which can be one of the following:

- **Offline state** In this state, something has failed and Teredo cannot be activated (cannot be in the Qualified state) to be used by applications. Teredo enters this state in three ways:
    - When the Administrator disables it via netsh int teredo set state disabled.
    - When Teredo detects that the computer is on a managed network (detects the presence of a domain controller on the network—see the section in this sidebar titled "Teredo in Enterprise Networks" for more information), it will go offline if its type is not set to "enterpriseclient".
    - When some internal mechanism has failed in Teredo, such as suddenly being unable to reach the Teredo server or being unable to resolve teredo.ipv6.microsoft.com. In only this case, Teredo will attempt to move into the Dormant state using an exponential back-off time-out as follows: wait 5 seconds, try again; wait 10 seconds, try again; wait 20 seconds, try again; and continue until it tries every 15 minutes.
- **Dormant state** This is the state when Teredo is "enabled but not active." IPv6 traffic cannot flow over Teredo, but applications can trigger to activate Teredo. No edge traversal will occur in this state. No traffic is sent to the Teredo servers.
- **Probe state** This is the transition state from Dormant to Qualified. In this state, Teredo will try to establish communication with the Teredo server. If this succeeds, Teredo moves to the Qualified state. If this fails, Teredo will go to the Offline state.
- **Qualified state** In this state, IPv6 traffic can flow into and out of the system over Teredo and possibly traverse the edge firewall/NAT.

Teredo in Enterprise Networks

Whether a computer is domain-joined or in a workgroup doesn't matter to Teredo. Teredo looks only at the environment that the computer is in. If Teredo detects the presence of a domain controller, it will assume that the network is managed. In this case, Teredo will go offline and stay offline unless it was administratively set to "enterpriseclient" using the command **netsh interface teredo set state enterpriseclient**. Hence, Teredo will go to the Offline state on a workgroup computer that is connected to a network with a domain controller. This is to avoid traversing the

edge of a corporate network. Conversely, if you take a domain-joined laptop home, Teredo will detect that it is no longer in a managed network and will go to the Dormant state.

Note that if you disable Teredo via the DisabledComponents registry key, it will override all the Teredo netsh settings.

Kalven Wu, Software Design Engineer in Test

Windows Core Networking

Configuring IPv6 in Windows 7 Using the User Interface

To configure the IPv6 settings for a network connection in Windows 7 using the user interface, follow these steps:

1. Open the Network And Sharing Center in Control Panel.
2. Click Manage Network Connections and then double-click the connection you want to configure.
3. Click the Properties button and respond to the UAC prompt.
4. Select Internet Protocol Version 6 (TCP/IPv6) and click Properties to open the Internet Protocol Version 6 (TCP/IPv6) Properties sheet (see Figure 29-1).
5. Configure the IPv6 settings for the network connection as desired.
6. Optionally validate the new TCP/IP settings using the Windows Network Diagnostics Troubleshooter.

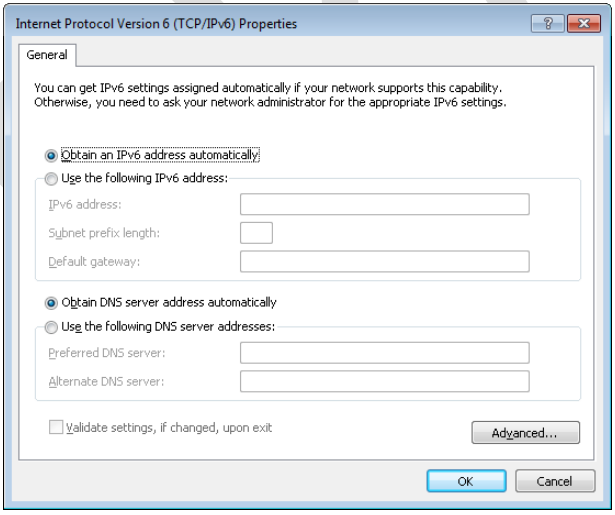

Figure 29-1 IPv6 properties of a network connection.

By default, the IPv6 settings for a network connection are configured as follows:

- Obtain An IPv6 Address Automatically This specifies that the physical or logical interface associated with this connection uses stateful or stateless address autoconfiguration to obtain its IPv6 address.
- Obtain DNS Server Address Automatically This specifies that the physical or logical interface associated with this connection uses stateful address autoconfiguration (DHCPv6) to obtain the IPv6 addresses of preferred and alternate DNS servers.

By selecting Use The Following IPv6 Address, you can manually configure the IPv6 address settings for a network connection by specifying the following:

- IPv6 Address Type the unicast IPv6 address you want to assign to the physical or logical interface associated with this connection in colon-hexadecimal form. If you need to assign additional unicast IPv6 addresses to the interface, click the Advanced button and then click the IP Settings tab.
- Subnet Prefix Length Type the subnet prefix length for the IPv6 address you assigned to the physical or logical interface associated with this connection. For unicast IPv6 addresses, the subnet prefix length should almost always be specified as 64.
- Default Gateway Type the unicast IPv6 address of the default gateway for the local IPv6 subnet in colon-hexadecimal form. If you need to specify additional default gateways, click the Advanced button and then click the IP Settings tab.

By selecting Use The Following DNS Server Addresses, you can manually specify IPv6 addresses for a preferred and an alternate DNS server to be used by your connection. If you need to specify additional alternate DNS servers, click the Advanced button and then click the DNS tab. The remaining settings on the DNS tab have similar functionality to those used for configuring IPv4 address settings.

Note The Advanced TCP/IP Settings dialog box does not have a WINS tab because IPv6 does not use NetBIOS for name resolution.

Configuring IPv6 in Windows 7 Using Netsh

To configure the IPv6 settings for a network connection in Windows 7 using the Netsh.exe command, open a command prompt window with local administrator credentials and type the appropriate Netsh.exe command from the netsh interface ipv6 context. Some examples of IPv6 configuration tasks that can be performed from this context include:

- To add the unicast IPv6 address 2001:DB8::8:800:20C4:0 to the interface named Local Area Connection as a persistent IPv6 address with infinite valid and preferred lifetimes, type the following command:

```
netsh interface ipv6 add address “Local Area Connection”
2001:DB8::8:800:20C4:0
```

- To configure a default gateway with unicast IPv6 address 2001:DB8:0:2F3B:2AA:FF:FE28:9C5A for the interface named Local Area Connection, add a default route with this address specified as a next-hop address by typing the following command:

  netsh interface ipv6 add route ::/0 "Local Area Connection" 2001:DB8:0:2F3B:2AA:FF:FE28:9C5A

- To configure a DNS server with unicast IPv6 address 2001:DB8:0:1::1 as the second (alternate) DNS server on the list of DNS servers for the interface named Local Area Connection, type the following command:

  netsh interface ipv6 add dnsserver "Local Area Connection" 2001:DB8:0:1::1 index=2

For more information on using the netsh interface ipv6 context, type **netsh interface ipv6 ?** at a command prompt.

Other IPv6 Configuration Tasks

The following section describes some additional IPv6 configuration tasks that network administrators may need to know how to perform with Windows 7 computers.

Enabling or Disabling IPv6

You cannot uninstall IPv6 in Windows 7, but you can disable IPv6 on a per-adapter basis. To do this, follow these steps:

1. Open the Network And Sharing Center in Control Panel.
2. Click Manage Network Connections and then double-click the connection you want to configure.
3. Clear the check box labeled Internet Protocol Version 6 (TCP/IPv6) and then click OK (see Figure 29-2).

Note that if you disable IPv6 on all your network connections using the user interface method as described in the preceding steps, IPv6 will still remain enabled on all tunnel interfaces and on the loopback interface.

As an alternative to using the user interface to disable IPv6 on a per-adapter basis, you can selectively disable certain features of IPv6 by creating and configuring the following DWORD registry value:

HKLM\SYSTEM\CurrentControlSet\Services\tcpip6\Parameters\DisabledComponents

Table 29-7 describes the flag values that control each IPv6 feature. By combining these flag values together into a bitmask, you can disable more than one feature at once. (By default, DisabledComponents has the value 0.)

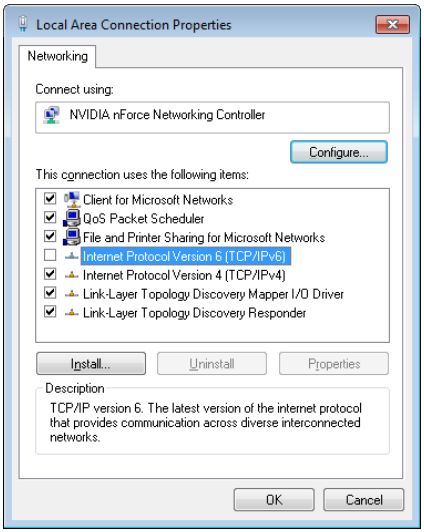

Figure 29-2 Disabling IPv6 for a network connection.

Table 29-7 Bitmask Values for Disabling IPv6 Features in Windows 7

FLAG LOW-ORDER BIT	RESULT OF SETTING THIS BIT TO A VALUE OF 1
0	Disables all IPv6 tunnel interfaces, including ISATAP, 6to4, and Teredo tunnels
1	Disables all 6to4-based interfaces
2	Disables all ISATAP-based interfaces
3	Disables all Teredo-based interfaces
4	Disables IPv6 over all non-tunnel interfaces, including LAN and Point-to-Point Protocol (PPP) interfaces
5	Modifies the default prefix policy table1 to prefer IPv4 over IPv6 when attempting connections

[1]For more information concerning the IPv6 prefix policy table, see the Cable Guy article "Source and Destination Address Selection" for IPv6 at *http://technet.microsoft.com/en-us/library/bb877985.aspx*.

For example, by setting the value of *DisabledComponents* to 0xFF, you can simultaneously disable IPv6 on all your network connections and tunnel interfaces. If you do this, IPv6 still remains enabled on the loopback interface, however.

Note For some examples of common flag combinations that can be used to enable or disable different aspects of IPv6 functionality in Windows 7 and Windows Vista, see the Cable Guy article "Configuring IPv6 with Windows Vista" at *http://technet.microsoft.com/en-us/library/bb878057.aspx*.

Depending on your scenario, there are other ways of effectively disabling IPv6 on Windows 7 computers, including:

- Disable the IP Helper service This service must be running for IPv6 transition technologies such as ISATAP, Teredo, and 6to4 to function on the computer. This service provides automatic IPv6 connectivity over an IPv4 network, and if the service is stopped, the computer will have only IPv6 connectivity if it is connected to a native IPv6 network. Therefore, if your network is not native IPv6, disabling this service on Windows 7 computers effectively disables IPv6 on them. You can use Group Policy to disable this service on targeted Windows 7 computers.
- Use netsh to disable all IPv6 interfaces For example, the following commands will disable all IPv6 transition technologies (Teredo, 6to4, and ISATAP):

  netsh interface teredo set state disabled

  netsh interface ipv6 6to4 set state state=disabled undoonstop=disabled

  netsh interface ipv6 isatap set state state=disabled

  You can include these commands in a script and send them inside an SCCM package to disable transition technologies on targeted computers.
- Configure Windows Firewall to block IPv6 traffic You could block incoming and outgoing IPv6 protocol 41 (for ISATAP and 6to4) and UDP 3544 (for Teredo) traffic using the Windows Firewall, and you can use Group Policy to push this out to targeted computers. Businesses that implement perimeter firewalls may want to do this as a best practice for safeguarding their networks.

Disabling Random Interface IDs

You can disable the default behavior of generating random interface IDs for nontemporary autoconfigured public addresses (global addresses registered in DNS) and link-local addresses by using the following command:

netsh interface ipv6 set global randomizeidentifiers=disabled

To re-enable the generating of random interface IDs, use the following command:

netsh interface ipv6 set global randomizeidentifiers=enabled

Note Disabling random interface IDs causes link-local addresses to revert to using 48-bit MAC-layer (or 64-bit EUI) addresses for generating the interface ID portion of the address. In Windows, this happens immediately and does not require a reboot.

Resetting IPv6 Configuration

To remove all user-configured IPv6 settings and restore the IPv6 configuration of a computer to its default state, type the following command:

netsh interface ipv6 reset

You must reboot the computer for this command to take effect.

Displaying Teredo Client Status

To verify the current state of the Teredo client on your computer, open a command prompt window using local administrator credentials, and then type the following command:

netsh interface teredo show state

For a Windows 7 computer on which Teredo is currently inactive, the typical output for this command looks like this:

```
Teredo Parameters
---------------------------------------------
Type                    : default
Server Name             : teredo.ipv6.microsoft.com.
Client Refresh Interval : 30 seconds
Client Port             : unspecified
State                   : dormant
Client Type             : teredo client
Network                 : managed
NAT                     : none (global connectivity)
```

Note If your command output doesn't contain all of the preceding information, you probably started your command prompt session using standard credentials instead of administrator credentials.

If you now start an IPv6-enabled application that uses Teredo, such as Windows Meeting Space or Windows Remote Assistance, and then type the same Netsh command, the command output typically now looks like this:

```
Teredo Parameters
```

```
-----------------------------------------
Type                    : default
Server Name             : teredo.ipv6.microsoft.com.
Client Refresh Interval : 30 seconds
Client Port             : unspecified
State                   : qualified
Client Type             : teredo client
Network                 : managed
NAT                     : restricted
```

Comparing these two command outputs shows that starting an application that uses Teredo changes the Teredo client state from Dormant (inactive) to Qualified (active).

Note The output of the **netsh interface teredo show state** command also tells you the type of NAT your computer is behind (if any). In the preceding example, the computer is behind a restricted NAT. Teredo works well behind restricted and cone NATs and can even work behind symmetric NATs, but communication between certain types of NATs doesn't work. If you plan to purchase a SOHO router for broadband Internet connectivity, the best choice is a router that supports 6to4. For more information on how Teredo works and on the different types of NATs, see "Teredo Overview" at *http://www.microsoft.com/technet/network/ipv6/teredo.mspx*.

Troubleshooting IPv6 Connectivity

The standard approach for troubleshooting TCP/IP network connectivity issues on IPv4 networks is to follow these steps:

1. Type **ipconfig /all** at a command prompt to verify the IPv4 configuration of the computer that is experiencing the problem.
2. If verifying the computer's IPv4 configuration doesn't resolve the issue, try using the ping command to test for network connectivity, beginning with the local computer and working outward until the cause of the problem is determined. Specifically, follow these steps in order listed:
   a. Ping the IPv4 loopback address 127.0.0.1 to verify that TCP/IP is installed and configured properly on the computer.
   b. Ping the IPv4 address of the local computer.
   c. Ping the IPv4 address of the default gateway.
   d. Ping the IPv4 address of an IPv4 host on a remote subnet.

Other TCP/IP troubleshooting steps you can use on IPv4 networks include:

- Use the **route print** command to verify the configuration of the local computer's routing table.
- Use **tracert** to verify that intermediate routers are configured properly.
- Use the **pathping** command to identify packet loss over multi-hop paths.
- Clear the ARP cache by typing **netsh interface ip delete arpcache** at a command prompt.
- Verify the computer's DNS configuration, clear the DNS client resolver cache, and verify DNS name resolution.

Note For more information on how to systematically troubleshoot IPv4 connectivity problems, read Chapter 32, "Troubleshooting Network Issues."

Troubleshooting IPv6 network connectivity issues requires many of the same tools you use when troubleshooting IPv4. However, you use some of these tools in a different way because of the nature of IPv6 addressing and the way IPv6 is implemented in Windows 7 and Windows Vista. The differences include:

- You might need to specify a zone ID when attempting to verify IPv6 network connectivity with a target host using the ping command. The syntax for using ping with IPv6 is **ping *IPv6Address%ZoneID*,** where *ZoneID* is the zone ID (or scope ID) of the sending interface. For example, if the target host has the link-local unicast IPv6 address FE80::D3:00FF:FE28:9C5A, and the sending interface has a zone ID of 12, to verify IPv6 connectivity with this host you would type **ping**

FE80::D3:00FF:FE28:9C5A%12 at a command prompt. To determine the zone ID for an interface, you can either use the **ipconfig /all** command or type **netsh interface ipv6 show interface** at a command prompt. Note that since the zone ID is locally defined, a sending host and a receiving host on the same link may have different zone IDs. (Global and unique local unicast IPv6 addresses do not need a zone ID.)

- You should view and clear the neighbor cache on your computer before attempting to use ping to verify IPv6 network connectivity. The neighbor cache contains recently resolved link-layer IPv6 addresses; you can view it by typing **netsh interface ipv6 show neighbors** and flush it by typing **netsh interface ipv6 delete neighbors** at an elevated command prompt.
- You should also view and clear the destination cache on your computer before attempting to verify IPv6 network connectivity using ping. The destination cache contains next-hop IPv6 addresses for destinations. You can view the cache by typing **netsh interface ipv6 show destinationcache**; you can flush it by typing **netsh interface ipv6 delete destinationcache** at an elevated command prompt.
- You should use the **–d** option when attempting to trace the route to a remote IPv6 host using tracert, or the **–n** option when using pathping. These options prevent these commands from performing DNS reverse queries on every near-side router interface along the routing path. Using these options can help speed up the display of the routing path.

Note For more help on troubleshooting IPv6 network connectivity issues, see the Cable Guy article "Troubleshooting IPv6" at *http://technet.microsoft.com/en-us/library/bb878005.aspx*. See also Chapter 12, "Troubleshooting TCP/IP," in the online book TCP/IP Fundamentals for Microsoft Windows, which you can download from *http://www.microsoft.com/downloads/details.aspx?FamilyID=c76296fd-61c9-4079-a0bb-582bca4a846f&displaylang=en*.

Note Disabling IPv4 can also be a useful troubleshooting technique for developers who need to verify that their applications are IPv6-capable.

Planning for IPv6 Migration

Migrating your existing IPv4-based network infrastructure to IPv6 requires an understanding of different IPv6 transition technologies that you can use achieve your goal. Windows 7, Windows Vista and Windows Server 2008 support three transition technologies in particular:

- ISATAP Stands for Intra-Site Automatic Tunnel Addressing Protocol, an address assignment and automatic tunneling technology defined in RFC 4214 that you can

use to provide unicast IPv6 connectivity between IPv6/IPv4 hosts (hosts that support both IPv6 and IPv4) across an IPv4-based intranet (a private network whose infrastructure hardware, such as routers, only supports IPv4 and not IPv6).

- 6to4 An address assignment and automatic tunneling technology defined in RFC 3056 that you can use to provide unicast IPv6 connectivity between IPv6/IPv4 hosts and sites across the IPv4-based public Internet. 6to4 enables you to assign global IPv6 addresses within your private network so that your hosts can reach locations on the IPv6 Internet without needing a direct connection to the IPv6 Internet or an IPv6 global address prefix obtained from an IPv6-supporting ISP. (Communication between a 6to4 site and a node on the IPv6 Internet requires the use of a 6to4 relay, however.)
- Teredo An address assignment and automatic tunneling technology defined in RFC 4380 that you can use to provide unicast IPv6 connectivity between IPv6/IPv4 hosts across the IPv4 public Internet, even when the IPv6/IPv4 hosts are located behind zero or more NATs. Teredo provides similar functionality to 6to4 but without needing edge devices that support 6to4 tunneling.

Note For more information on IPv4/v6 transition technologies, see the white paper "IPv6 Transition Technologies" at *http://www.microsoft.com/downloads/details.aspx?FamilyID=afe56282-2903-40f3-a5ba-a87bf92c096d&DisplayLang=en*.

These three IPv6 transition technologies are supported by Windows 7, Windows Vista, Windows Server 2008, Windows XP Service Pack 2, and Windows Server 2003 Service Pack 1. Of the three, ISATAP is the primary transition technology that you should use for migrating an existing IPv4-based intranet to IPv6; it is discussed further in the following sections. Teredo is primarily useful in Small Office/Home Office (SOHO) networking environments, where NAT-enabled broadband routers provide Internet connectivity for users. (Think of Teredo as a transition technology of last resort, because as IPv6 connectivity becomes ubiquitous, the need for NAT traversal will decline until Teredo is no longer needed.)

How It Works: Blocking Teredo

Teredo is intended to be a consumer technology and has generally not been recommended for enterprises. This is because Teredo requires the edge device to allow all outbound UDP traffic. For example, because of security reasons, many enterprise administrators do not want client computers on the corporate network to be directly accessible from the Internet, and in that case turning off Teredo is a good idea.

If administrators want to disable Teredo on their client computers or simply prevent it

from working, they can do so in one of three ways:

- Block all outbound UDP traffic by default. (This is the only reliable "external" method.)
- Block name resolution of the Teredo DNS host name, which by default on Windows 7 computers is teredo.ipv6.microsoft.com. (This method, however, leaves an easy workaround, because the user can hard-code IP addresses.)
- Use Group Policy or a script to create the following DWORD registry value, which turns off Teredo on targeted Windows 7 computers. (This registry setting is not exposed by default in Group Policy but can be pushed down using a custom ADMX file.)

  HKLM\SYSTEM\CurrentControlSet\Services\Tcpip6\Parameters\DisabledComponents

  You can specify the following settings for this value:

  - **0x10** Setting this value will disable Teredo only on the computer.
  - **0x01** Setting this value will disable all tunnel interfaces on the computer.

If administrators want to support only native IPv6 in their networks, or if they don't want to support any IPv6 traffic until they deploy native IPv6, they can choose to turn off all tunneling technologies using the second choice in the preceding list.

Understanding ISATAP

By default, the IPv6 protocol in Windows 7 automatically configures a link-local unicast IPv6 address of the form FE80::5EFE:w.x.y.z (for private IPv4 addresses) or FE80::200:5EFE:w.x.y.z (for public IPv4 addresses). This address is a link-local ISATAP address, and it is assigned to the ISATAP tunneling interface. Using their link-local ISATAP addresses, two ISATAP hosts (such as Windows 7 computers) can communicate using IPv6 by tunneling across an IPv4-only network infrastructure (such as a network whose routers forward only IPv4 packets and not IPv6 packets).

Note In Windows 7 and in Windows Vista with Service Pack 1 or later, link-local ISATAP addresses are automatically configured only if the name "ISATAP" (the ISATAP router name) can be resolved. Otherwise, the ISATAP interface will be media-disconnected. However, if you administratively enable ISATAP by using the **netsh interface isatap set state enabled** command, the link-local address will be configured regardless of whether the ISATAP router name can be resolved.

With the addition of one or more ISATAP routers (IPv6-enabled routers that advertise address prefixes, forward packets between ISATAP hosts and other ISATAP routers, and act as default routers for ISATAP hosts) a variety of transition topologies become possible, including:

- Connecting ISATAP hosts on an IPv4-only intranet to an IPv6-capable network
- Connecting multiple "islands" of ISATAP hosts through an IPv6-capable backbone

These configurations are possible because ISATAP routers advertise address prefixes that enable ISATAP hosts (such as Windows 7 computers) to autoconfigure global or unique local unicast IPv6 addresses.

Note Without the presence of an ISATAP router, ISATAP hosts running Windows Vista RTM could only autoconfigure link-local unicast IPv6 addresses, which limited IPv6 communications to between hosts on the IPv4-only intranet. This was changed in Windows Vista SP1 so that without an ISATAP router, the interface will show media-disconnected. In other words, Windows Vista SP1 won't configure a link-local ISATAP address when no ISATAP router is configured. The behavior in Windows 7 is the same as in Windows Vista SP1.

Note For more information on how ISATAP works, see the white paper "IPv6 Transition Technologies" at *http://www.microsoft.com/downloads/details.aspx?FamilyID=afe56282-2903-40f3-a5ba-a87bf92c096d &displaylang=en*.

Direct from the Source: ISATAP Interface Name

The ISATAP interface name is based on the DNS setting of the primary IPv4 interface of this ISATAP interface. For example, if the DNS suffix assigned to the primary IPv4 interface of this ISATAP interface is contoso.com, the ISATAP interface name will be isatap.contoso.com.

An alternate form of the ISATAP interface name is isatap.{GUID} where GUID is a globally unique identifier. However, this GUID form is used to name the ISATAP interface only if there is no DNS suffix setting on the primary IPv4 interface.

Xinyan Zan, Technical Lead

IPv6 Transition Technology

Migrating an Intranet to IPv6

Best practices for migrating existing IPv4-based network infrastructures to IPv6 are still evolving. Therefore, this section presents a general outline of how to migrate an intranet to IPv6 and provides references to more detailed information on the subject for interested readers.

The ultimate goal of IPv4 to IPv6 migration is to achieve an IPv6-only network infrastructure that has IPv6-only hosts. From a practical standpoint, however, the lesser goal of achieving a network infrastructure that supports both IPv6 and IPv4—and where hosts also support both IPv6 and IPv4 but use mainly IPv6—is a more reasonable goal to aim for. Achieving this goal is a lengthy process that involves seven main steps:

1. Upgrading your applications and services
2. Preparing your DNS infrastructure
3. Upgrading your hosts
4. Migrating from IPv4-only to ISATAP
5. Upgrading your routing infrastructure
6. Upgrading your DHCP infrastructure
7. Migrating from ISATAP to native IPv6

Step 1: Upgrading Your Applications and Services

To prepare your applications and services for migration, you will need to upgrade existing applications and services to support IPv6 in addition to IPv4. This may require upgrades from ISVs and third-party vendors or custom coding on your part. Although the ultimate goal is for all your applications and services to run native IPv6, a more appropriate target is to ensure that they work with both IPv4 and IPv6.

For further guidance, see the MSDN topic "IPv6 Guide for Windows Sockets Applications" at *http://msdn2.microsoft.com/en-us/library/ms738649.aspx*.

Step 2: Preparing Your DNS Infrastructure

You must prepare your DNS infrastructure to support the AAAA records used to resolve DNS names to IPv6 addresses. This might require upgrading your existing DNS servers. The DNS Server service of Windows Server 2008 and Windows Server 2003 supports dynamic registration of AAAA records for unicast IPv6 addresses (excluding link-local addresses).

For more information on configuring Windows Server 2003 DNS servers to support IPv6 hosts, see Chapter 9, "Windows Support for DNS," in the online book *TCP/IP Fundamentals for Microsoft Windows*, which can be found at *http://technet.microsoft.com/en-us/library/bb727009.aspx*.

Step 3: Upgrading Your Hosts

You may need to upgrade some of your hosts until all your hosts support both IPv6 and IPv4. Windows platforms from Windows XP Service Pack 2 onward support both IPv4 and IPv6, though full support for IPv6 functionality for built-in programs and services is only provided in Windows Vista and later.

Step 4: Migrating from IPv4-only to ISATAP

After you've prepared your applications, services, hosts, and DNS/DHCP infrastructure, you can begin deploying ISATAP routers to create islands of IPv6 connectivity within your IPv4-based intranet. You will need to add A records to the appropriate DNS zones so that your ISATAP hosts can determine the IPv4 addresses of your ISATAP routers.

You may decide to deploy zero or more ISATAP routers for inter-ISATAP subnet routing within your intranet, depending on the size of your intranet and the geographical distribution of its sites. You may decide to deploy redundant ISATAP routers to provide consistent availability of IPv6 address prefixes and other configuration settings for your ISATAP hosts. You will also likely deploy one or more ISATAP routers to provide IPv6 connectivity between your IPv4-based network infrastructure and the public IPv6 Internet as this evolves.

For more information on deploying ISATAP routers using different migration scenarios, see the white paper "Intra-site Automatic Tunnel Addressing Protocol Deployment Guide" at *http://www.microsoft.com/downloads/details.aspx?FamilyID=0f3a8868-e337-43d1-b271-b8c8702344cd &displaylang=en*.

Step 5: Upgrading Your Routing Infrastructure

After you have deployed ISATAP to enable IPv6 hosts to communicate over your IPv4 network infrastructure, you should begin upgrading your network infrastructure (including routers, gateways, and other access devices) to support IPv6. Rather than upgrading your infrastructure to support only IPv6, a more reasonable upgrade goal is dual IPv4/IPv6 support. In many cases, actual replacement of router hardware is not necessary. Because many modern hardware routers support both IPv4 and IPv6 routing, the task of upgrading of your routing infrastructure to support IPv6 becomes configuration, not replacement. As you enable IPv6 routing support for a subnet, also enable the DHCPv6 relay agent for the subnet.

Typically, you will begin upgrading your routing infrastructure early in your ISATAP deployment by upgrading the core routers on your network backbone to support IPv6. This will create islands of ISATAP hosts that connect to this backbone to communicate with other IPv6 hosts anywhere in your intranet.

Step 6: Upgrading Your DHCP Infrastructure

You can optionally upgrade your routing and DHCP infrastructure to support DHCPv6 for automatic assignment of global or unique local unicast IPv6 addresses or configuration

settings for IPv4/IPv6 nodes on your network. By using DHCPv6, an IPv6 host can obtain subnet prefixes and other IPv6 configuration settings. A common use of DHCPv6 is to configure Windows 7–based client computers with the IPv6 addresses of DNS servers on the network. (DNS servers are not configured through IPv6 router discovery.)

The DHCP Server service in Windows Server 2003 does not support stateful address autoconfiguration or the DHCPv6 protocol. The DHCP Server role in Windows Server 2008, however, supports both stateful and stateless IPv6 address autoconfiguration using DHCPv6. The DHCP Client service in Windows 7, Windows Vista and Windows Server 2008 supports address autoconfiguration using DHCPv6.

Just as with DHCP with IPv4, you also need to deploy and configure DHCPv6 relay agents for each subnet containing Windows 7 clients. Many hardware routers already support a DHCPv6 relay agent. You must configure relay agents with the IPv6 addresses of the DHCPv6 servers on your network. Relay agents can be configured but should not be enabled until you deploy IPv6 routing on your subnets.

When you are ready to enable DHCPv6 on subnets, configure your IPv6 routers to set the Managed Address Configuration and Other Stateful Configuration flags to the appropriate values for stateful or stateless DHCPv6 operation. For more information, see the Cable Guy article titled "The DHCPv6 Protocol" at *http://www.microsoft.com/technet/technetmag/issues/2007/03/CableGuy/default.aspx*.

Step 7: Migrating from ISATAP to Native IPv6

Finally, when all your network infrastructure devices support IPv6, you can begin to decommission your ISATAP routers, because you no longer need them. Whether you will also migrate your infrastructure and hosts to support only pure-IPv6 is a decision best left for the distant future.

Direct from the Source: Tips and Tricks for Transitioning from IPv4 to IPv6

When transitioning a network from IPv4-only to dual stack, there are several areas that need to be paid special attention.

Addressing

This is actually one area that gets easier with IPv6, due to the huge address space that it offers. In general, you will want to add to each individual network segment a single IPv6 /64 prefix, even in cases where you have more than one IPv4 subnet assigned to the same network (e.g., by using the secondary keyword on Cisco routers). You should not need to use Unique Local Addresses, even for lab networks. One exception might be that you do not want to use a routable /64 prefix for a segment that is not connected to your organization's globally routable space (i.e., it is physically separate).

Firewalls

Deploying IPv6 can present issues for an organization's security team. Because IPSec services are available in all IPv6 stacks, it is more common to see end-to-end security implemented with IPv6-enabled desktops. When faced with end-to-end encryption, a firewall administrator has one of two choices: either deny the traffic and drop it at the perimeter, or allow it through unchecked, thus bypassing the ACLs and other security enabled on the firewall. Note that this problem exists even with IPv6-enabled firewalls.

Tunneling Technologies

Many transition technologies, such as ISATAP, 6to4, and manually configured IPv6-in-IPv4 tunnels, encapsulate IPv6 packets inside IPv4 in order to transport them across an IPv4-only part of your network. These packets are identified by the use of IP protocol 41 in the encapsulating packet. If firewalls, ACLs, or other devices in your network are not configured to forward these packets, then communications using these technologies will break. Many home routers, for example, are configured by default to only forward UDP and TCP protocols.

Here's a real life example: After configuring a router to provide IPv6 services at an IANA meeting in Florida, IPv6 connectivity was not working. After some troubleshooting with the service provider, I determined that their router was dropping IP protocol 41, thus preventing IPv6 connectivity across the service provider's IPv4-only network.

Network Applications

When deciding to IPv6-enable an existing workflow or application, make sure to consider all parts of the process. For example, while upgrading a web front end to support IPv6, don't forget to also enable the separate file store and backend database servers as well, otherwise the workflow may appear to support IPv6 from the front end, but will not be completed tested.

DNS

Many DNS products today support the AAAA records which are used to store name-to-address mappings for IPv6 end systems. However, that does not mean that they support IPv6 lookups against the database—in some cases, this functionality must be enabled through a configuration setting or an upgrade to the product itself. This is another part of an end-to-end IPv6 workflow that needs to be considered.

Address Management

A simple way to enable IPv6 auto-configuration on your hosts is to configure your edge routers to advertise an IPv6 prefix via Router Advertisements. This enables IPv6-enabled operating systems, including Vista, Windows Server 2008, and Windows 7, to configure itself with an IPv6 address. This method of configuration is

considered stateless, as the router will not track which IPv6 addresses are configured on which end system. When performing address auditing against these systems (to investigate a security incident, for example), it is impossible to determine which host was assigned a given IPv6 address at a particular time. At best, if you are lucky, the router's ARP tables will contain the necessary information, but more often than not, you will be unable to track a specific IPv6 address to the host on which it was configured.

Here's a real life example: At a previous job, I was contacted by the local office of the United States Secret Service to investigate a threat made against a government official. I was able to track the IPv4 address they provided me to a high school in the school district where I worked, and to a specific classroom at a certain time, based on DHCP logs and switch CAM tables. A student was subsequently identified as being in the classroom alone at the time and admitted to sending the messages, which turned out to be a hoax. Tracking down auto-configured IPv6 addresses at this level of detail is nearly impossible.

Mike Owen, Network Engineer

Data and Storage Platform Division

Summary

This chapter described the features of IPv6 in Windows 7, provided an overview of how IPv6 works, and outlined best practices for migrating an existing IPv4-only network to IPv6. An IPv6 migration requires careful planning and a thorough understanding of how IPv6 works, and Windows 7, together with Windows Server 2008 R2, provide the features and tools you need to migrate your network successfully.

Additional Resources

The following resources contain additional information and tools related to this chapter.

Related Information

- *Understanding IPv6, Second Edition* by Joseph Davies (Microsoft Press, 2008). See *http://www.microsoft.com/MSPress/books/11607.aspx*.
- The IPv6 home page on Microsoft TechNet at *http://technet.microsoft.com/en-us/network/bb530961.aspx*.
- The IPv6 blog of Sean Siler, IPv6 Program Manager, *http://blogs.technet.com/ipv6*.
- "IPv6 for Microsoft Windows: Frequently Asked Questions" at *http://www.microsoft.com/technet/network/ipv6/ipv6faq.mspx*.
- The white paper "Introduction to IP Version 6" at *http://www.microsoft.com/downloads/details.aspx?FamilyID=CBC0B8A3-B6A4-4952-BBE6-D976624C257C&displaylang=en*.
- The white paper "IPv6 Transition Technologies" at *http://www.microsoft.com/downloads/details.aspx?FamilyID=afe56282-2903-40f3-a5ba-a87bf92c096d &displaylang=en*.
- The white paper "Intra-site Automatic Tunnel Addressing Protocol Deployment" at *http://www.microsoft.com/downloads/details.aspx?FamilyID=0f3a8868-e337-43d1-b271-b8c8702344cd &displaylang=en*.
- The Cable Guy article "Changes to IPv6 in Windows Vista and Windows Server 'Longhorn'" at *http://technet.microsoft.com/en-us/library/bb878121.aspx*.
- The Cable Guy article "Performance Enhancements in the Next Generation TCP/IP Stack" at *http://technet.microsoft.com/en-us/library/bb878127.aspx*.
- The Cable Guy article "Understanding the IPv6 Routing Table" at *http://technet.microsoft.com/en-us/library/bb878115.aspx*.
- The Cable Guy article "Manual Configuration for IPv6" at *http://technet.microsoft.com/en-us/library/bb878102.aspx*.
- "Using Windows Tools to Obtain IPv6 Configuration Information" on Microsoft TechNet at *http://www.microsoft.com/technet/itsolutions/network/ipv6/ipv6config.mspx*.
- The Cable Guy article "Troubleshooting IPv6" at *http://technet.microsoft.com/en-us/library/bb878005.aspx*.
- The Cable Guy article "Source and Destination Address Selection for IPv6" found on

Microsoft TechNet at *http://technet.microsoft.com/en-us/library/bb877985.aspx*.

- "Domain Name System Client Behavior in Windows Vista" on Microsoft TechNet at *http://technet.microsoft.com/en-us/library/bb727035.aspx*.
- KB 929852, "How to disable certain Internet Protocol version 6 (IPv6) components in Windows Vista," at *http://support.microsoft.com/kb/929852*.
- KB 932134, "An outdated network router may not function correctly when you use it together with new networking features in Windows Vista," at *http://support.microsoft.com/kb/932134*.
- KB 944007, "Unable to access shares via IPv6 address due to the ':' character," at *http://support.microsoft.com/kb/944007*.
- KB 934640, "In Windows, Event Viewer incorrectly displays IPv6 addresses in event descriptions," at *http://support.microsoft.com/kb/934640*.
- KB 946784, "How to obtain the IPv6 Ready Logo for Windows Vista and for Windows Server 2008 from IPv6 Forum," at *http://support.microsoft.com/kb/946784*.
- KB 929851, "The default dynamic port range for TCP/IP has changed in Windows Vista and in Windows Server 2008," at *http://support.microsoft.com/kb/929851*.
- Chapter 9, "Windows Support for DNS," and Chapter 12, "Troubleshooting TCP/IP," in the online book TCP/IP Fundamentals for Microsoft Windows, which you can download from *http://www.microsoft.com/downloads/details.aspx?FamilyID=c76296fd-61c9-4079-a0bb-582bca4a846f &displaylang=en*.