Module 3: Plan and implement IPv6

Computers and other devices that connect to computer networks mostly use the IPv4 network, although this set of protocols was built more than 40 years ago. In 1995, address management was delegated to the Internet Assigned Numbers Authority (IANA). As you learned earlier in this course, IPv4 addressing consists of private and public IPv4 addresses. While private IPv4 addresses are not routable on the Internet, and can be reused in every organization's internal network, public IPv4 addresses are routable on the Internet. Therefore, they must be unique on the Internet. With the rapid growth of the Internet that started in the late 1990s, it became evident that far more addresses would be needed to connect devices than the IPv4 address space had available. By 1998, the Internet Engineering Task Force (IETF) had formalized the successor protocol, named IPv6. In December 1998, IPv6 became a Draft Standard for the IETF, and it was ratified as an Internet Standard on 14 July 2017.

IPv4 remains widely used in enterprise and cloud networks, but the public IPv4 address space is limited. IPv6 was designed to provide a much larger address space, improve hierarchical routing, support automatic configuration, and remove the need for broadcast traffic. Windows Server 2022 and Windows Server 2025 support IPv4 and IPv6 in a dual-stack configuration, which allows organizations to introduce IPv6 while continuing to support IPv4-dependent applications and services.

In modern environments, IPv6 planning should include on-premises networks, DNS, DHCPv6, firewall policies, monitoring tools, and Azure virtual networks. Azure supports dual-stack IPv4/IPv6 virtual networks and subnets so workloads can communicate by using IPv4 and IPv6 where supported.

After you complete this module, you should have the knowledge and skills to:

- Describe an IPv6 network.
- Implement and configure IPv6.

Lesson 1: Overview of IPv6

Most devices that connect to the internet and organizational networks support IPv6, as does Microsoft Windows Server and client operating systems. Many applications support the use of both IPv4 and IPv6 addressing, and many organizations are migrating their networking solutions to work with IPv6. Therefore, it's important to understand IPv6 addressing so that you can plan, implement, and manage networks that work with IPv6.

After you complete this lesson, you should have the knowledge and skills to:

- Describe IPv6.
- Explain the advantages of using IPv6.
- Comprehend the details of IPv6 addressing.

- Describe the types of IPv6 addresses.
- Explain autoconfiguration options available for IPv6.

What is IPv6?

IPv4 uses 32-bit addresses, which limits the number of unique public addresses. As internet-connected devices, cloud services, mobile devices, and IoT deployments expanded, public IPv4 address space became scarce. IPv6 uses 128-bit addresses and provides a much larger address space. Most organizations don't replace IPv4 immediately; instead, they deploy IPv6 alongside IPv4 in a dual-stack configuration while they validate application, security, DNS, and network-device support..

IPv6 is an Internet Layer protocol for packet-switched internetworking, same as IPv4. IPv6 closely adheres to the design principles developed for IPv4 and provides end-to-end datagram transmission across multiple IP networks. The IPv6 address space provides addresses that are 128 bits, as opposed to IPv4, which provides 32-bit addresses. This makes a larger number of addresses available and also provides an address space that can be subdivided into hierarchical routing domains that reflect modern-day internet topology.

Here is a brief overview of the main issues with the current set of IPv4 protocols:

- **Rapid depletion of the address space**. This leads to the use of Network Address Translators (NATs) that map multiple private addresses to a single public IP address.
- **Lack of hierarchy support**. This results in the inability to structure the IP addresses in a way that truly maps the network topology.
- **Complex network configuration**. When using IPv4, addresses must be assigned statically or using a configuration protocol such as DHCP.
- **Security isn't automatic at the IP layer.** IPv6 supports IPsec, and Windows supports IPsec for IPv4 and IPv6, but applications and administrators must still configure appropriate security controls such as TLS, firewall rules, IPsec policies, and network segmentation..
- **Large routing tables.** Internet routing tables have continued to grow. IPv6 was designed to support hierarchical addressing and route aggregation, but effective summarization still depends on careful provider and enterprise address planning..

The IPv6 set of protocols addresses and resolves many of the issues that IPv4 has with address depletion, security, auto-configuration, extensibility, and so on. Table 1 compares the addressing features of IPv4 and IPv6.

Table 1: IPv6 vs. IPv4 addressing

IPv4	IPv6
Requires routers and the sending host to perform fragmentation.	Allows only the sending host to perform fragmentation, not routers.

Resolves IPv4 addresses to a link-layer address by using broadcast Address Resolution Protocol (ARP) request frames.	Uses multicast neighbor-solicitation messages rather than ARP request frames.
Ensures the management of membership of local subnet groups by Internet Group Management Protocol (IGMP).	Uses Multicast Listener Discovery messages rather than IGMP.
Locates the best default gateway's IPv4 address, optionally, by using Internet Control Message Protocol (ICMP) Router Discovery.	Requires ICMPv6 Router Solicitation and Router Advertisement messages to be used, which replace ICMP Router Discovery.
Maps host names to IPv4 addresses by using host (A) resource records in the Domain Name System (DNS).	Maps host names to IPv6 addresses by using IPv6 host (AAAA) resource records in DNS.
Maps IPv4 addresses to host names by using pointer (PTR) resource records in the IN-ADDR.ARPA DNS domain.	Maps IPv6 addresses to host names by using pointer (PTR) resource records in the IP6.ARPA DNS domain.
Requires support for packet sizes of 576 bytes. Those packets are potentially fragmented.	Requires support for packet sizes of 1280 bytes, with no fragmentation.

IPv6 removes broadcast and uses multicast and Neighbor Discovery Protocol instead. IPv6 also requires ICMPv6 for core network functions such as neighbor discovery, router discovery, path MTU discovery, and address autoconfiguration. Don't block all ICMPv6 traffic in firewalls; instead, allow the ICMPv6 message types required for normal IPv6 operation.

Benefits of IPv6

The main and most tangible benefit of IPv6 is the increased number of IP addresses, which resolves the issue of there not being enough IPv4 addresses that are routable on the internet. IPv4 addresses consist of 32 bits, so theoretically, there are 2^{32} IPv4 addresses available. With IPv6, there are 2^{128} available because IPv6 addresses are comprised of 128 bits. In other words, a 32-bit address space can have 4,294,967,296 possible addresses. IPv6 uses 128-bit address spaces, which can have 340,282,366,920,938,463,463,374,607,431,768,211,456 (or 3.4x10^38 or 340 undecillion) possible addresses.

Additional benefits of IPv6 addressing include:

- **Hierarchical addressing and routing.** IPv6 supports efficient hierarchical addressing and routing. Conversely, the IPv4 routing infrastructure combines flat and hierarchical routing, so there are more than 70,000 routes in backbone routers' routing tables due to how IPv4 network IDs are allocated.

- **Easier configuration.** IPv6 supports autoconfiguration for devices connecting to a network by using Stateless Address Auto Configuration (SLAAC). IPv4 implementations must be configured either manually or through DHCP.
- **IP-layer security.** Private communication over a public medium like the internet requires encryption services that help protect data from being accessed or modified while in transit. There are many optional proprietary solutions to help provide security for IPv4 packets, in addition to Internet Protocol security (IPsec).
- **Support for real-time data delivery.** In IPv4, support for real-time traffic uses a User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) port to facilitate the IPv4 Type of Service (ToS) field and payload identification. However, the ToS field has different interpretations and offers limited functionality. Additionally, if an IPv4 packet payload is encrypted, payload identification using a TCP and UDP port isn't possible.
- **Windows operating system support.** Windows Server and client operating systems support IPv6, can connect to IPv6 networks, and can work with apps that support IPv6. It's important to note that even if you've yet to deploy IPv6, we don't recommend that you disable it.
- **Support for Microsoft 365 services.** Microsoft 365 supports IPv6 for many services and scenarios, but organizations shouldn't assume that all Microsoft 365 connectivity can be IPv6-only. Review the current Microsoft 365 IPv6 support guidance and Microsoft 365 endpoint documentation before changing production network egress, DNS, proxy, or firewall rules. Microsoft recommends that, for now, organizations don't turn off IPv4 when connecting to Microsoft 365 services. Some services still need both IPv6 and IPv4 for their functionality. Microsoft 365 endpoint data changes over time. Use the Microsoft 365 URLs and IP address ranges guidance as the authoritative source for current endpoints and change-management timing. You can verify which Microsoft 365 services are fully enabled for IPv6 at [IPv6 support in Microsoft 365 services](#).
- **Support for Azure virtual networking.** Azure supports IPv6 for Virtual Network in dual-stack designs. You can define IPv6 address space for a virtual network, create dual-stack subnets, assign IPv4 and IPv6 configurations to network interfaces, and expose workloads through supported Azure networking services. In Azure, IPv6 subnets must be exactly /64. Plan Azure IPv6 alongside IPv4. A dual-stack Azure design still requires IPv4 address planning for services, management paths, and workloads that don't support IPv6-only operation..
- **Better support for mobile devices and mobile workers.** As the number of mobile devices increases, a new requirement has been introduced: A device must be able to arbitrarily change locations on the IPv6 Internet while still maintaining existing connections. To provide this functionality, a mobile device is assigned a home address that can always be reached. When the mobile device is at home, it connects to the home link and uses its home address. When the mobile node is away from home, a home agent, which is usually a router, relays messages between the mobile node and the nodes with which it is communicating.

IPv6 addressing

In Module 1, you learned that IPv4 addresses consist of four decimal numbers, separated by dots, such as 192.168.1.1. Binary notation of IPv4 addresses consists of 32 bits grouped in four octets. For example, the IPv4 address in decimal notation 192.168.1.1 can be represented in binary notation as follows:

```
11000000.10101000.00000001.00000001
```

An IPv6 address has 128 bits, which means that representing an IPv6 address in binary notation would require that you write 128 digits. Therefore, it's more convenient that IPv6 addresses are represented in hexadecimal notation, because writing an IPv6 address in hexadecimal format would require you to write 32 digits. The following depicts an IPv6 address example:

```
2001:0DB8:0000:2D4C:01CC:00DD:1122:1234
```

Hexadecimal numbering system (Base 16)

In the binary numbering system, there are only two digits: 0 and 1. In the hexadecimal numbering system, there are 16 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. For example, when counting from 0 to 9, digits are the same in both decimal and hexadecimal notation. Decimal numbers from 10 to 15 are represented in hexadecimal notation by the letters A to F. Finally, number 16 in decimal notation is number 10 in hexadecimal notation. Table 2 depicts the relationship between numbers in different notations: decimal, hexadecimal, and binary. This is convenient when working on converting from hexadecimal to binary notation.

Table 2: Numbers in decimal, hexadecimal, and binary notation

Decimal notation	Hexadecimal notation	Binary notation
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001

10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

> **Note**: You can use the Calculator application that Windows Server and client operating systems include to convert between binary, decimal, and hexadecimal numbers.

Convert IPv6 addresses from hexadecimal to binary notation

Based on the data in Table 2, let's change the following IPv6 address to binary notation:

```
2001:0DB8:0000:2D4C:01CC:00DD:1122:1234
```

We will convert each block in IPv6 address to binary format, as shown in Table 3.

Table 3: Conversion from IPv6 to binary notation

Hexadecimal	**Binary**
[2][0][0][1]	[0010][0000][0000][0001]
[0][D][B][8]	[0000][1101][1011][1000]
[0][0][0][0]	[0000][0000][0000][0000]
[2][D][4][C]	[0010][1101][0100][1100]
[0][1][C][C]	[0000][0001][1100][1100]
[0][0][D][D]	[0000][0000][1101][1101]
[1][1][2][2]	[0001][0001][0010][0010]

[1][2][3][4]	[0001][0010][0011][0100]

Finally, consider the following IPv6 address that's in hexadecimal format:

```
2001:0DB8:0000:2D4C:01CC:00DD:1122:1234
```

It'll be represented in binary format as:

```
0010000000000001   0000110110111000   0000000000000000   0010110101001100
0000000111001100   0000000011011101   0001000100100010   0001001000110100
```

Convert IPv6 addresses from binary to hexadecimal notation

To convert an IPv6 address from binary to hexadecimal notation, you'll need to break the 128 bits into eight blocks of 16 bits. For example, here's an IPv6 address in binary notation:

```
0010000000000001000011011011100000000000000000000010110101001100
0000000111001100000000001101110100010001001000010 0001001000110100
```

It would be broken into eight blocks of 16 bits, as follows:

```
0010000000000001   0000110110111000   0000000000000000   0010110101001100
0000000111001100   0000000011011101   0001000100100010   0001001000110100
```

You'll then need to break each block of 16 bits into four blocks of 4 bits. For example:

```
0010 0000 0000 0001   0000 1101 1011 1000   0000 0000 0000 0000   0010
1101 0100 1100
0000 0001 1100 1100   0000 0000 1101 1101   0001 0001 0010 0010   0001
0010 0011 0100
```

Finally, using the data in Table 4 we'll convert numbers from binary notation to hexadecimal notation, convert each group of 4 bits, as shown in Table 4.

Table 4: Binary to hexadecimal conversion

Binary	Hexadecimal
[0010][0000][0000][0001]	[2][0][0][1]
[0000][1101][1011][1000]	[0][D][B][8]
[0000][0000][0000][0000]	[0][0][0][0]
[0010][1101][0100][1100]	[2][D][4][C]
[0000][0001][1100][1100]	[0][1][C][C]
[0000][0000][1101][1101]	[0][0][D][D]
[0001][0001][0010][0010]	[1][1][2][2]
[0001][0010][0011][0100]	[1][2][3][4]

At the end, we add colons between each group of four hexadecimal digits:

```
2001:0DB8:0000:2D4C:01CC:00DD:1122:1234
```

An alternative way to convert each four-digit binary number to a hexadecimal number is to use a mathematical operation by performing the following steps:

1. Multiply the right-most bit (first bit) by 1.

2. Multiply the second bit by 2.
3. Multiply the third bit by 4.
4. Multiply the fourth bit by 8.
5. Summarize all results.

For example, number 1110 in binary notation is calculated as follows:

0X1 + 1X2 + 1X4 + 1X8 = 14 (decimal notation) = D (hexadecimal notation)

Further, simplify an IPv6 address in hexadecimal format by removing each leading zero within a four-digit hexadecimal block. Therefore, for a block with four zeros, you'd notate it with only one zero. Here's an example of an IPv6 address:

```
2001:0DB8:0000:2D4C:01CC:00DD:1122:1234,
```

You also could notate it as:

```
2001:DB8:0:2D4C:1CC:DD:1122:1234
```

You also can use zero compression, whereby you shorten the four-digit hexadecimal block of an IPv6 address by replacing four zeros with a double colon (::). For example, here's an IPv6 address:

```
2001:0DB8:0000:2D4C:01CC:00DD:1122:1234
```

Using zero compression, you could notate it as:

```
2001:DB8::2D4C:1CC:DD:1122:1234
```


You can determine how many double colons are representing 0 bits (digits in binary notation) by:

1. Counting the blocks in the compressed address. Referring to the previous example, you'll observe that there are seven blocks.
2. Subtracting the number of blocks (seven) from eight, which is one.
3. Multiplying the result of the subtraction (one) by 16, which is 16.

The result of 16 indicates that the address in which the double colon occurs has 16 bits, 16 zeros (binary system), or four zeros (hexadecimal system).

Each address can have only one instance of zero compression only once in each address. Two or more instances of it mean that you can't illustrate how many 0 bits each double-colon instance of (::) represents.

> **Note**: Use zero compression only once in an IPv6 address. When multiple zero runs are present, compress the longest run. If runs are equal length, compress the leftmost run. This makes addresses easier to compare and document consistently.

Conversely, you'd use the reverse process to change an address into binary by:

1. Using zero compression to add zeros.

2. Adding leading zeros.
3. Changing each number into its binary equivalent.

IPv6 address structure

Each IPv6 address is 128 bits long, and the *prefix* is the part that has the bits with fixed values or the subnet prefix's bits. IPv6 addresses don't use subnet masks, and IPv6 prefixes are equivalent to a network ID for IPv4 addresses.

IPv6 uses Classless Interdomain Routing (CIDR) notations to represent subnets, prefixes, routes, and address ranges, and address/prefix-length notation indicates an IPv6 prefix. For example, 2001:DB8::/48 (a route prefix) and 2001:DB8:0:2D4C::/64 (a subnet prefix) are IPv6 address prefixes.

Most IPv6 LAN subnets use a /64 prefix. SLAAC requires a 64-bit interface identifier, and Azure IPv6 subnets must be /64. Use other prefix lengths only when a specific technology or provider design requires them.. This provides the host with a unique identifier that can be created in one of three ways, including by:

- Random assignment.
- Assignment by DHCPv6.
- Assignment based on the media access control (MAC) address of the network.

Unless DHCPv6 assigns the host bits, by default they're generated randomly.

> **Note:** A network's size dictates the prefix sizes for routes on an IPv6 router.

IPv4 special addresses and their IPv6 equivalents

Table 5 lists common IPv4 addresses and their IPv6 equivalents.

Table 5: IPv6 equivalents to IPv4 addresses

Address types	IPv4 address	IPv6 address
Unspecified	0.0.0.0	::
Loopback	127.0.0.1	::1
Autoconfigured	169.254.0.0/16	FE80::/64
Broadcast	255.255.255.255	Uses multicasts instead
Multicast	224.0.0.0/4	FF00::/8

> **Note:** Link-local IPv6 addresses are automatically configured and are used for local-link communication, neighbor discovery, and router discovery. They are similar to APIPA only in the sense that they're automatically assigned and not routed beyond the local link.

Types of IPv6 addresses

IPv6 supports three types of addresses, based on type and scope, including:

- *Unicast* addresses, which pinpoint one interface with a unicast address-type scope and are for packet delivery from a single interface to another. The unicast addresses are distinguished from the multicast addresses by the value of the high-order octet. The high-order octet for multicast addresses has the hexadecimal value of FF. Any other value for this octet identifies a unicast address.
- *Multicast* addresses, which identify multiple interfaces and are for one-to-many communications. Multicast addresses deliver to multiple interfaces or all interfaces in a set. The multicast address types supersede the IPv4 broadcast addresses.
- *Anycast* addresses, which identify multiple interfaces and are for one-of-many communications. Anycast addresses deliver to one interface in the set that the address identifies as being the nearest with respect to routing distance. A packet sent to this address is delivered to only one interface identified by the address. This is the nearest interface as identified by routing metrics. Anycast addresses are taken from the unicast address space and are not syntactically distinguishable.

> **Note:** IPv6 addresses don't identify nodes, only interfaces. A node is identified by any unicast address that's assigned to one of its interfaces.

While IPv4 includes a broadcast address, IPv6 doesn't. Instead, IPv6 uses a multicast address to send a package to all nodes, which is equivalent to broadcast traffic.

Several kinds of unicast IPv6 addresses exist, including:

- Global unicast addresses.
- Unique local addresses.
- Link-local addresses.
- Site-local addresses (Site-local addresses are deprecated and shouldn't be used in new designs. Use unique local addresses for private IPv6 addressing inside an organization)..
- Special addresses.
- Compatibility or transition addresses.

Global unicast addresses

Internet Service Providers (ISPs) provide public IPv4 addresses, and a *global unicast address* is the IPv6 equivalent of those. They're internet-routable, where ISP providers can assign a large number of IPv6 addresses to customers. The first 48 bits from the global unicast address identify the customer site, and the next 16 bits are allocated for the customer to subnet in its network.

These addresses can be used across the Internet and have the following format:
GlobalRoutingPrefix::*SubnetID*:*InterfaceID*

> **Note:** It's important to note that the network 2001:0db8::/32 isn't routable, as it's reserved for documentation.

The structure of a global unicast address has four parts, including:

- **The three high-order (left-most) bits are set to 001.** The three left-most bits of the IPv6 address start with 001. Therefore, the fourth left-most bit's value can be 0 or 1. This means that the four left-most bits will have a value of 0010 (hexadecimal notation of 2) or 0011 (hexadecimal notation of 3). Therefore, all global unicast addresses will start with 2 or 3 when represented in hexadecimal notation. The address prefix for currently assigned global addresses is 2000::/3.
- **Global routing prefix.** The combination of the three fixed left-most bits and the next 45-bit global-routing prefix creates a 48-bit site prefix assigned to an organization, which identifies the specific organization's site.
- **Subnet ID.** This consists of the next 16 bits, identifies subnets within an organization, and enables the creation of a different amount of subnets by using those 16 bits.
- **Interface ID.** This is comprised of the remaining 64 bits and identifies a specific host's interface within the site. Can be randomly generated, assigned by DHCPv6, or created based on the network interface card's MAC address to which the IPv6 address is headed.

What's a unique local address?

A *unique local address is the IPv6 equivalent to an IPv4 private address*. They're not internet-routable, as they're reachable only within an organization's network.

Most organizations use IPv4 private addresses, so it's probable that they're using the same private range of IPv4 addresses. This could cause problems in scenarios where two or more organizations are merging. However, the IPv6 unique local address structure allocates randomly generated 40 bits to an organization identifier with a small likelihood of two randomly generated 40-bit identifiers being the same. This ensures that if organizations were to merge, two or more organizations will each have a unique IPv6 address space.

The first seven bits of the organization identifier have the fixed binary value of 1111110. Depending on the eighth bit value, hexadecimal notation will have value FC or FD. All unique local addresses have the address prefix of FC00::/7. The Local (L) flag is set to 1 to indicate a local address. An L flag value set to 0 hasn't yet been defined. Therefore, unique local addresses with the L flag set to 1 have the address prefix of FD::/8.

For unique local addresses, use a pseudo-random global ID as intended by the standard. Don't use all zeros or simple patterns, because that increases the chance of address overlap during mergers, VPN connectivity, or partner-network integration.

Link-local addresses

For local subnet communications, all IPv6 hosts have an automatically generated link-local address That's not internet-routable and is the IPv6 equivalent to an IPv4 Automatic Private IP Addressing (APIPA) address.

Link-local addresses are for communication in scenarios in which IPv4 would have used broadcast messages, such as when communicating with a DHCPv6 server. Link-local addresses are also used for neighbor discovery, which is the IPv6 equivalent of IPv4 ARP. The prefix for link-local addresses is always FE80::/64, where the second group of 64 bits are the interface identifiers.

Zone ID

IPv6 hosts that have several network interfaces use the same link-local address on all of them. A zone ID is added to the link-local address to distinguish them, using the following syntax:

Address%zone_ID

The zone ID that goes with each interface is determined by the sending hosts, and a unique interface index (an integer) gets assigned to each Windows-based host interface. Also, interfaces also include loopback and tunnel interfaces, in addition to physical network cards. Windows-based IPv6 hosts use the interface’s index as its zone ID. Here’s an example where 3 is the network-interface ID:

```
fe80::2b0:d0ff:fee9:4143%3
```

Special addresses

There are two special addresses in IPv6, including:

- The *unspecified address* (0:0:0:0:0:0:0:0 or ::), which indicates that the network interface doesn’t have an IPv6 address assigned. The unspecified IPv6 address of :: is the IPv6 equivalent to the IPv4 unspecified address of 0.0.0.0, and it’s a source address for packets that’re attempting to verify a tentative address’ uniqueness. You can’t assign an unspecified address to a network interface or use it as a destination address.
- The *loopback address* (0:0:0:0:0:0:0:1 or ::1) identifies a loopback interface, which allows nodes to send packets to themselves. It’s the IPv6 equivalent to the IPv4 loopback address of 127.0.0.1. Packets aren’t routable if addressed to the loopback address.

Compatibility or transition addresses

If you need to ensure coexistence between IPv4 and IPv6 addressing during migration, you should employ compatibility or transition addresses. Table 6 lists ranges for each IPv6 address type.

Table 6: Address ranges for IPv6 addresses

Type	Range
Global unicast	Currently allocated from 2000::/3
Unique local	FC00::/7; commonly FD00::/8 for locally assigned unique local addresses

Link-local	FE80::/10; Windows commonly displays link-local addresses beginning with fe80
Multicast	FF00::/8
Unspecified	::
Loopback	::1
Documentation	2001:DB8::/32

Autoconfiguration options

There are several options for IPv6 protocol autoconfiguration. By default, for each interface, an IPv6 host can configure a link-local IPv6 address. Also, a host can use router discovery to locate router IPv6 addresses and other configuration parameters. The Router Advertisement message includes an indication of whether to use a stateful address-configuration protocol.

> **Note:** You can perform address autoconfiguration only on multicast-capable interfaces. IPv6 stateless address autoconfiguration is defined by current IETF standards.

Types of autoconfiguration

There are three autoconfiguration types in the IPv6 protocol:

- **Stateless autoconfiguration.** This type of autoconfiguration is appropriate for smaller organizations. When using stateless autoconfiguration, each host determines its addresses from the contents of received router advertisements. With SLAAC ((Stateless Address Autoconfiguration), a host uses information in Router Advertisement messages to configure an IPv6 address. The router advertises the network prefix, and the host generates the interface identifier. Modern operating systems, including Windows, can use randomized interface identifiers instead of deriving the identifier from the network adapter MAC address..
- **Stateful autoconfiguration.** This type of autoconfiguration requires some network admin intervention as it requires a Dynamic Host Configuration Protocol for IPv6 (DHCPv6) server to install and administer the nodes. The DHCPv6 server keeps a list of the nodes to which it supplies configuration information. It also maintains state information so the server knows how long each address is in use for, and when it might be available for reassignment. In this scenario, the network adapter obtains addresses and other configuration options. An IPv6 host uses stateful address configuration in two scenarios, including when:
  - Router Advertisement messages instruct it to do so.
  - The local link has no routers.

- **Both stateless and stateful**, in which configuration is based on receipt of Router Advertisement messages and DHCPv6.

Table 7: Autoconfiguration types

Method	Address source	Other configuration source	Typical use
SLAAC only	Router Advertisement	Router Advertisement	Simple IPv6 networks where clients can self-configure
SLAAC + stateless DHCPv6	Router Advertisement	DHCPv6 options, such as DNS	Networks that want automatic addressing but centralized options
Stateful DHCPv6	DHCPv6	DHCPv6	Networks that want DHCPv6 to assign addresses and track leases

> Note: Router Advertisement flags influence whether hosts use DHCPv6 for address assignment or for additional options. DHCPv6 doesn't replace the need for routers and router advertisements in most IPv6 networks.

Stateful configuration

A DHCPv6 server assigns IPv6 addresses and other configuration options from a specific scope of IPv6 addresses when an organization implements stateful configuration. An IPv6 host uses multicast IPv6 addresses when it attempts to communicate with a DHCPv6 server.

Autoconfigured address states

During autoconfiguration, the network interface changes states, so several autoconfiguration phases exist for an IPv6 address of a host's network interface. The host's network-interface states define the life cycle of the IPv6 address. There are several states IPv6 addresses that undergo autoconfiguration, including:

- **Tentative.** In this state, the host goes through a verification process to determine whether the IPv6 address is unique. The host uses a duplicate address-detection algorithm on the IPv6 address before assigning it to a network interface. When an address is in the tentative state, a network interface can't receive unicast traffic.
- **Valid.** In this state, the address has been verified as unique. If a network interface has a valid IPv6 address state, it can send and receive unicast traffic.
- **Preferred.** In this state, the IPv6 address enables a network interface to send and receive unicast traffic.
- **Deprecated.** In this state, the use of an IPv6 address for new communication is discouraged, although the address does remain valid.
- **Invalid.** In this state, the IPv6 address no longer allows a network interface to send or receive unicast traffic.

Lesson 2: Implement and configure IPv6

IPv4 has been used for decades, so migrating fully to IPv6 hasn't occurred for many organizations. Many use both IPv4 and IPv6 protocols. Coexistence between IPv4 and IPv6 is essential, so it's important for you to understand how IPv4 and IPv6 protocols coexist. Windows Server 2022 and Windows Server 2025 support IPv4 and IPv6 in a dual-stack configuration. In most organizations, IPv6 is introduced gradually alongside IPv4. Administrators must understand how to configure IPv6 addresses, DNS, DHCPv6, routing, firewall rules, and coexistence behavior.

After you complete this lesson, you should have the knowledge and skills to:

- Configure IPv6 settings.
- Set up Dynamic Host Configuration Protocol version 6 (DHCPv6).
- Describe the different types of IP node.
- Describe the coexistence options for IPv4 and IPv6.
- Explain what IPv6 tunneling is.
- Describe the IPv6 Router Advertisements and site prefixes.

Configure IPv6 settings

IPv6 addresses can be assigned to a network interface by:

- Manually configuring one or more IPv6 addresses on the interface.
- Using a DHCPv6 server for stateful address autoconfiguration.
- Basing stateless address autoconfiguration on the receipt of Router Advertisement messages.
- Using both stateful and stateless address autoconfiguration.

> **Note:** The applicable operating system (OS) always configures a link-local address automatically on an interface, regardless of whether you deploy stateful or stateless address autoconfiguration.

IPv6 protocol is designed for automatic configuration, which means that in many cases, you won't need to configure IPv6 addresses manually or by using a DHCPv6 server. IPv6 provides the functionality to use stateless address autoconfiguration for many of your organization's network hosts. Most network adapters on IPv6 hosts have multiple IPv6 addresses assigned to them, which means they're *multihomed*. The addresses assigned to them can include:

- A link-local address for local-link traffic. This is automatically generated.
- An additional unicast address for traffic that requires routing beyond the local link. This can be either a global or unique local address.

Configure IPv6 addressing on Windows Server and client operating systems by using:

- Graphical user interface (GUI) tools.
- Windows PowerShell commands.

- **Netsh** commands.

> **Note:** Microsoft doesn't recommend disabling IPv6 or its components unless a Microsoft support engineer or official product guidance specifically directs you to do so. Some Windows components expect IPv6 to be enabled. If you must change IPv6 behavior, use the documented registry-based method rather than clearing the IPv6 checkbox on a network adapter..

In Windows Server 2025, Windows Terminal is available for command-line administration. Use it to run PowerShell commands such as Get-NetIPAddress, Get-NetRoute, Get-NetPrefixPolicy, and Test-NetConnection during IPv6 configuration and troubleshooting.

GUI tools

To configure IPv6 settings on a network interface on a Windows Server or client operating system, you modify the interface's properties in the **Network Connections folder**. You can configure options on the **General** tab of the **Internet Protocol Version 6 (TCP/IPv6) Properties** dialog box.

General tab of the Internet Protocol Version 6 (TCP/IPv6) dialog box

In the **General** tab of the **Internet Protocol Version 6 (TCP/IPv6) Properties** dialog box, you can configure several IPv6 addressing options, including:

- **Obtain an IPv6 address automatically.** Specify that stateful or stateless address autoconfiguration is used to determine IPv6 addresses.
- **Use the following IPv6 address.** Specify that manual configuration is used for the IPv6 address and default gateway.
- **IPv6 address.** Enter an IPv6 unicast address and specify additional IPv6 addresses from the **Advanced TCP/IP Settings** dialog box.
- **Subnet prefix length.** Enter the subnet prefix length for the IPv6 address. Set this value to 64 for typical IPv6 unicast addresses.
- **Default gateway.** Enter the IPv6 unicast address for the default gateway.
- **Obtain DNS server address automatically.** Specify that stateful address autoconfiguration (DHCPv6) should automatically determine the IPv6 addresses for DNS servers.
- **Use the following DNS server addresses.** Manually configure the IPv6 addresses of the preferred and alternate DNS servers for the connection or adapter.
- **Preferred DNS server.** Specify the preferred DNS server's IPv6 unicast address.
- **Alternate DNS server.** Enter the alternate DNS server's IPv6 unicast address and then specify additional DNS servers in the **Advanced TCP/IP Settings** dialog box.

Advanced TCP/IP settings

From the **General** tab, select **Advanced** to access the **Advanced TCP/IP Settings** dialog box. It's similar to the **Advanced TCP/IP Settings** dialog box for the Internet Protocol version 4 (TCP/IPv4) component. The only differences are that there isn't a **WINS** tab, because IPv6 doesn't use NetBIOS and the Windows Internet Name Service (WINS) or an **Options** tab, because TCP/IP filtering is defined only for IPv4 traffic. For IPv6, the **Advanced TCP/IP Settings** dialog box includes the **IP Settings** and **DNS** tabs.

The IP Settings tab

You can configure the following on the **IP Settings** tab:

- **Multiple IPv6 addresses**. Specify an IPv6 address and a subnet prefix length for each unicast IPv6 address. If you select **Use the Following Ipv6 Address** on the **General** tab of the **Internet Protocol Version 6 (TCP/IPv6) Properties** dialog box, the **Add** button becomes available.
- **Multiple default gateways**. Specify the IPv6 address for each default gateway. Additionally, specify whether you want to manually select the metric for the default route associated with this default gateway or have it based on the connector or adapter's speed.
- **Route metrics**. Specify whether to use a specific metric for the routes associated with configuration of IPv6 addresses or default gateways or use a metric that the speed of the connection or adapter determines.

The DNS tab

You can configure the following on the **DNS** tab:

- **DNS server addresses, in order of use**. Enter the IPv6 addresses of DNS servers that you want to query for resolving DNS domain names. They're queried in the order you list them.
- **Append primary and connection specific DNS suffixes**. Specifies that resolution for unqualified names is limited to the domain suffixes of the primary suffix and all connection-specific suffixes. Configure the connection-specific DNS suffixes in **DNS suffix for this connection**.
- **Append these DNS suffixes (in order)**. Lists the DNS suffixes to search in the order you enter.
- **DNS suffix for this connection**. Enables you to enter a DNS suffix for the connection, unless it was configured by stateful address autoconfiguration (DHCPv6).
- **Register this connection's addresses in DNS**. Specifies that the computer should attempt to dynamically register the connection's IP addresses, using DNS and the complete computer name.
- **Use this connection's DNS suffix in DNS registration**. Select this check box if you want this registration is in addition to the full computer name's DNS registration.

Windows PowerShell commands

In Windows Server and client operating systems, you can configure IPv6 addresses, default gateways, and DNS servers with Windows PowerShell cmdlets, which Table 8 lists.

Table 8: Windows PowerShell cmdlets

Command	Description
Set-NetIPAddress	Modifies the IP address configuration properties for an existing IP address.
Set-NetIPInterface	Modifies IP interface properties.
Set-NetIPv6Protocol	Modifies information about the IPv6 Protocol configuration.
Set-NetNeighbor	Modifies a neighbor cache entry.
Set-NetRoute	Modifies one or more entries in the routing table.
Set-DnsClientServerAddress	Modifies DNS server addresses associated with an interface.
Get-NetIPAddress -AddressFamily IPv6	Shows IPv6 addresses on interfaces
New-NetIPAddress	Adds an IPv6 address
Remove-NetIPAddress	Removes an IPv6 address
Get-NetRoute -AddressFamily IPv6	Shows IPv6 routes
New-NetRoute	Adds an IPv6 route
Get-NetNeighbor -AddressFamily IPv6	Shows IPv6 neighbor cache entries
Get-NetPrefixPolicy	Shows source/destination address selection policy
Test-NetConnection	Tests connectivity and TCP ports

> **Note**: Windows Server and client operating systems also include alternative Windows PowerShell cmdlets for displaying IPv6 addresses, assigning new IPv6 addresses, and removing IPv6 addresses, such as **Get-NetIPAddress**, **New-NetIPAddress**, **Remove-NetIPAddress**.

Use the **Get-NetAdapter** cmdlet to display a list of names and indexes of the network interfaces:

```
PS C:\> Get-NetAdapter | fl Name,ifIndex
Name    : Ethernet
ifIndex : 12
```

You can also use the **Get-NetIPAddress** cmdlet to display the address information for the interface named **Ethernet**:

```
PS C:\> Get-NetIPAddress | where {$_.InterfaceAlias -eq "Ethernet"}
IPAddress         : fe80::2025:61fb:b68:c266%12
InterfaceIndex    : 12
InterfaceAlias    : Ethernet
AddressFamily     : IPv6
Type              : Unicast
PrefixLength      : 64
PrefixOrigin      : WellKnown
SuffixOrigin      : Link
AddressState      : Preferred
ValidLifetime     : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource      : False
PolicyStore       : ActiveStore
IPAddress         : 172.16.11.[illegible]
InterfaceIndex    : 12
InterfaceAlias    : Ethernet
AddressFamily     : IPv4
Type              : Unicast
PrefixLength      : 24
PrefixOrigin      : Manual
SuffixOrigin      : Manual
AddressState      : Preferred
ValidLifetime     : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource      : False
PolicyStore       : ActiveStore
```

Use the **New-NetIPAddress** cmdlet to assign a new global unicast IPv6 address with prefix length 64 and a default gateway address to the Ethernet interface:

```
PS C:\> New-NetIPAddress -InterfaceAlias "Ethernet" -IPAddress
2001:DB8:3FA9::D3:9C5A `
-PrefixLength 64 -DefaultGateway 2001:DB8:3FA9::0C01
IPAddress         : 2001:db8:3fa9::d3:9c5a
```

```
InterfaceIndex    : 12
InterfaceAlias    : Ethernet
AddressFamily     : IPv6
Type              : Unicast
PrefixLength      : 64
PrefixOrigin      : Manual
SuffixOrigin      : Manual
AddressState      : Tentative
ValidLifetime     : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource      : False
PolicyStore       : ActiveStore
IPAddress         : 2001:db8:3fa9::d3:9c5a
InterfaceIndex    : 12
InterfaceAlias    : Ethernet
AddressFamily     : IPv6
Type              : Unicast
PrefixLength      : 64
PrefixOrigin      : Manual
SuffixOrigin      : Manual
AddressState      : Invalid
ValidLifetime     : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource      : False
PolicyStore       : PersistentStore
```

To verify the result, use the **Get-NetIPAddress** cmdlet with the ***–AddressFamily*** parameter to display only IPv6 addressing information.


```
PS C:\> Get-NetIPAddress -AddressFamily IPv6 | where {$_.InterfaceAlias -
eq "Ethernet"}
IPAddress         : fe80::2025:61fb:b68:c266%12
InterfaceIndex    : 12
InterfaceAlias    : Ethernet
AddressFamily     : IPv6
Type              : Unicast
PrefixLength      : 64
PrefixOrigin      : WellKnown
SuffixOrigin      : Link
AddressState      : Preferred
ValidLifetime     : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource      : False
```

```
PolicyStore       : ActiveStore
IPAddress         : 2001:db8:3fa9::d3:9c5a
InterfaceIndex    : 12
InterfaceAlias    : Ethernet
AddressFamily     : IPv6
Type              : Unicast
PrefixLength      : 64
PrefixOrigin      : Manual
SuffixOrigin      : Manual
AddressState      : Preferred
ValidLifetime     : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource      : False
PolicyStore       : ActiveStore
```

The interface now has more than one IPv6 address assigned, so it's multihomed. It has one link-local IPv6 address and one global IPv6 address.

Netsh commands

You also can use the **Netsh.exe** tool to configure IPv6 settings, similar to how you'd use Windows PowerShell cmdlets. The netsh remains available and is useful when maintaining older scripts. For new Windows Server 2022 and Windows Server 2025 automation, prefer Windows PowerShell cmdlets from the NetTCPIP, DnsClient, and DhcpServer modules. You can configure IPv6 addresses, default gateways, and DNS servers by using commands in the **netsh interface ipv6** context.

For example, you can use the following command to configure the IPv6 unicast address **2001:db8:290c:1291::1** on the interface named **Local Area Connection** and make the address persistent:

```
netsh interface ipv6 add address "Local Area Connection"
2001:db8:290c:1291::1
```

When adding default gateways, use the following command to add a default route (::/0) that uses the interface named **Local Area Connection** with a next-hop address of **fe80::2aa:ff:fe9a:21b8**:

```
netsh interface ipv6 add route ::/0 "Local Area Connection"
fe80::2aa:ff:fe9a:21b8
```

When adding DNS servers, use the following command to add a DNS server with the IPv6 address **2001:db8:99:4acd::8** that uses the interface named **Local Area Connection**:

```
netsh interface ipv6 add dnsserver "Local Area Connection"
2001:db8:99:4acd::8
```

> **Note**: In Windows Server 2022 and Windows Server 2025, use Windows PowerShell for repeatable IPv6 configuration and troubleshooting. In Windows Server 2025, you can run these commands from Windows Terminal. GUI tools remain useful for one-time configuration and demonstrations. Use netsh primarily for legacy scripts or compatibility scenarios.

Demonstration: Configure IPv6

For high-level demonstration steps, refer to the slide **Notes**, which is in the accompanying Microsoft PowerPoint presentation.

Implement DHCPv6

To implement DHCPv6 in your network environment, you'll need a computer running Windows Server 2022, Windows Server 2025, or a supported Azure Local version,, with a DHCP service server role installed. After the DHCP service is installed, the DHCP console displays for a node for managing automatic IPv4 address assignments and another one for managing IPv6 address assignments. The following three options are available for configuring a DHCP scope:

- Creating an IPv6 scope.
- Creating an IPv6 prefix and preference.
- Creating exclusions.
- Configuring lease time.

After creating the scope, you can also configure additional DHCP settings, which Module 2 "Implement DHCP" details.

One of the most common options when configuring DHCP is to configure DNSv6 on DHCP v6 IP address scope. The standard option for DNS servers in DHCPv6 is Option 23. In Windows Server 2022 and newer versions, DNS Server supports IPv6 by default. Client computers or DHCP servers can register host records (AAAA) in DNS server. Similar to IPv4, you can manually create AAAA records for computers in DNS. DNS for IPv6 supports the creation and configuration of reverse lookup zones.

You can also use **netsh** to set DNS parameters for DHCP scope. Use the following commands to add the DNS server addresses to your DHCP scope:

```
   netsh dhcp server v6 scope <scopeID> set optionvalue 23 IP6ADDRESS
<IPv6 DNS Server 1>
   netsh dhcp server v6 scope <scopeID> set optionvalue 23 IP6ADDRESS
<IPv6 DNS Server 2> add
```

Replace <scopeID> with your specific scope identifier and <IPv6 DNS Server 1>, <IPv6 DNS Server 2> with the actual IPv6 DNS server addresses you wish to use.

It is recommended that, in Windows Server 2025, you use PowerShell to manage DHCP scopes. See the following example:

```
Add-DhcpServerv6Scope `
```

```
    -Prefix 2001:db8:100:10:: `
    -Name "IPv6 Lab Network"

Set-DhcpServerv6Scope `
    -Prefix 2001:db8:100:10:: `
    -State Active
```

```
Set-DhcpServerv6OptionValue `
    -Prefix 2001:db8:100:10:: `
    -DnsServer 2001:db8:100:10::10,2001:db8:100:10::11
```

> Note: DHCP failover in Windows Server supports IPv4 scopes only. DHCPv6 scopes can't be failover-enabled. If DHCPv6 is critical in your design, plan availability by using redundant DHCPv6 servers, router advertisements, documented configuration, and tested recovery procedures.

Demonstration: Configure DHCP for IPv6

For high-level demonstration steps, refer to the **Slide Notes**, which is in the accompanying Microsoft PowerPoint presentation.

Node types

In a coexistence scenario where you're using both IPv4 and IPv6 protocols, there are five categories of hosts (nodes):

- **IPv4-only node**. Supports only IPv4 and is configured with an IPv4 address only. It won't support the IPv6 protocol.
- **IPv6-only node**. Supports only IPv6 protocol and is configured with an IPv6 address only. Supports an environment where all other nodes and applications use only IPv6 protocol.
- **IPv6/IPv4 node.** Supports both protocols. Windows Server and client operating systems use IPv4 and IPv6 protocols by default.
- **IPv4 node.** Node is configured with an IPv4 address, but act as an IPv4-only node or IPv6/IPv4 node.
- **IPv6 node.** Node is configured with an IPv6 address, but can act as an IPv6-only node or IPv6/IPv4 node.

In a coexistence scenario where you're using both IPv4 and IPv6 protocols, you can implement multiple technologies to ensure that IPv4-only nodes can communicate with IPv6-only nodes. In most current Windows Server environments, the practical deployment model is a dual-stack node that supports both IPv4 and IPv6. IPv6-only designs are possible, but they require validation of applications, management tools, security inspection, DNS, monitoring, and connectivity to external services.

Coexistence options for IPv4 and IPv6

Microsoft has implemented an IPv6 protocol in Windows Server and client operating systems beginning with versions Windows Server 2008 and Windows Vista to enable organizations to add IPv6 to their existing IPv4 network instead of replacing IPv4.

Dual IP-layer architecture

A dual IP-layer architecture contains both IPv4 and IPv6 internet layers with a single implementation of transport layer protocols, such as TCP and UDP. This architecture allows for easier IPv6 migration, results in fewer files to maintain to provide IPv6 connectivity, and ensures that IPv6 is available without adding any new protocols in the network-card configuration.

Dual-stack architecture

Windows Server and Windows client operating systems support both IPv6 and IPv4 in a dual-stack configuration. The dual IP stack helps reduce maintenance costs by providing:

- Shared transport and framing layer.
- Shared filtering for firewalls and IPsec.
- Consistent performance, security, and support for both IPv6 and IPv4.

When a computer connects to a new network that advertises IPv6 routability, Windows OS tests IPv6 connectivity. Based on that test, a computer will only use IPv6 if IPv6 connectivity is actually functioning. Windows also supports a functionality called address sorting. This functionality helps the Windows OS to determine which protocol to use when applications that support both IPv4 and IPv6 addresses are configured for both protocol stacks.

> Note: For most organizations, dual stack is the recommended transition model because it avoids forcing every application and dependency to become IPv6-only at the same time. Use DNS, prefix policy, routing, and firewall rules to control which protocol is used for each workload.

Requirements for the DNS infrastructure

DNS service also supports IPv6 protocol. When you add an IPv6 host to a network, you must ensure that you add the records that're necessary to support IPv6 name-to-address and address-to-name resolution. Required DNS records for coexistence are:

- Host (A) resource records for IPv4 nodes.
- IPv6 host (AAAA) resource records.
- Reverse lookup pointer (PTR) resource records that map IPv4 and IPv6 nodes to their host names.

> **Note**: In most cases, the IPv6 host (AAAA) resource records that IPv6 nodes require are registered in DNS dynamically.

Both addresses are returned to a client when a name resolves to both an IPv4 and IPv6 address. The client chooses which address to use based on prefix polices in which each prefix has a precedence level assigned to it. A higher precedence is preferable to a lower precedence. Windows uses prefix policies to help choose source and destination addresses when both IPv4 and IPv6 are available. Don't rely on a static table in courseware as the authoritative value. Use Get-NetPrefixPolicy on the target Windows Server version to view the active prefix policy.

IPv6 has numerous benefits, and many organizations plan to migrate their computers and devices to it from IPv4. Typically, you'd discontinue IPv4 use and replace it with IPv6 use only, which is called a *native IPv6 environment*. However, to do this, you have to ensure that all computers, network devices, and applications support a native IPv6 environment. Some can't operate in a native IPv6 environment, even if they support working with both IPv4 and IPv6.

When planning for a native IPv6 environment, you should carefully consider your:

- **Operating-system support**. Native IPv6 is fully supported in Windows Vista, Windows Server 2008, and newer Windows Server and client operating systems.
- **Router and firewall support**. You should ensure that the routers and firewalls in your IT infrastructure support IPv6. It's especially crucial that firewall devices or software can detect any type of threat over the IPv6 protocol.
- **Network device support**. Other network devices, such as network printers, network scanners, and network cameras, should support IPv6.
- **Application product support**. Some application products support the IPv6 protocol, but they can't work in a native IPv6 environment. However, they can work in an IPv4 and IPv6 coexistence environment.
- **Custom application support**. The design of custom applications that organizations develop should support a native IPv6 environment.

Overview of IPv6 over IPv4 tunneling

Organizations that want to introduce IPv6 protocol in their network environment but have IPv4-only hosts can use *IPv6 over IPv4 tunneling*, in which IPv6 packets are encapsulated with an IPv4 header and transmitted over an IPv4-only infrastructure.

IPv6-over-IPv4 tunneling was useful when IPv6 had to cross IPv4-only infrastructure. Today, most new designs should prefer native IPv6 or dual-stack connectivity. Use tunneling only when required by a specific transition plan, and validate security inspection, routing, MTU, and operational support.

IPv6 over IPv4 tunneling includes the following changes to the IPv4 header:

- The **IPv4 Protocol** field of the IPv4 header is set to 41 to indicate an encapsulated IPv6 packet.
- The **Source and Destination** fields of the IPv4 header are set to the tunnel endpoints' IPv4 addresses. You can configure tunnel endpoints manually as part of the tunnel interface or configure them to be derived automatically.

During IPv6 over IPv4 tunneling, no message exchange occurs for tunnel setup, maintenance, or termination. Additionally, tunneled IPv6 packets aren't secured. This means that IPv6 tunneling doesn't need to establish a protected connection first.

IPv6 router advertisements and site prefixes

When you deploy an IPv6-based network, routers periodically multicast *router advertisement* messages. With these messages, routers announce their availability and convey information to neighboring nodes that enable them to be automatically configured on the network.

The content of IPv6 router advertisements is automatically derived from the published routes in the routing table of routing devices. Nonpublished routes are used for routing, but are ignored when constructing router advertisements.

IPv6 routers use Router Advertisement messages to announce prefixes, default-router information, link MTU, and flags that influence client autoconfiguration. Hosts use this information for SLAAC, default-router discovery, and on-link prefix discovery. In most enterprise networks, Router Advertisement behavior is configured on routers, Layer 3 switches, firewalls, or network virtual appliances rather than on member servers.

When planning Router Advertisements, verify:

- The advertised prefix and prefix length.
- Whether the prefix allows autonomous address configuration.
- Whether clients should use DHCPv6 for address assignment or only for additional options.
- The router lifetime and route preference.
- Firewall rules that allow required ICMPv6 traffic.
- DNS behavior for AAAA records and reverse lookup zones.

Router advertisements for IPv6 always contain a source link-layer address option and a Maximum Transmission Unit (MTU) option. The MTU is the size of the largest data packet unit that can be transmitted in a single network layer transaction. The value for the MTU option is taken from the current link MTU of the sending interface.

> **Note**: The router advertisement only has a nonzero router lifetime if there is a published default route. A default route is a route for the zero-length prefix.

Published on-link routes result in prefix information options in router advertisements. If the on-link prefix has 64 bits, the prefix information option has both the L and A bits set and hosts receiving it will autoconfigure addresses.

An interface that sends router advertisements will also autoconfigure addresses for itself based on the prefix information options that it sends.

A finite nonaging lifetime on all published routes (for example, 30 minutes) is recommended. If you decide to retract a route, you can change the route to have an aging lifetime. The route will age over the course of several router advertisements and then disappear from both the router and any hosts receiving the router advertisements.

Routes that hosts find through router advertisements age and are not published. Addresses autoconfigured from router advertisements age as well.

The site prefix is used to derive IPv6 addresses for all the nodes in your IPv6 network. The site prefix of an IPv6 address occupies up to 48 of the leftmost bits of the IPv6 address.

IPv6 in Azure virtual networks

Azure Virtual Network supports IPv6 by using a dual-stack model. A virtual network can have IPv4 and IPv6 address spaces, and subnets can support IPv4 and IPv6 prefixes. Dual-stack virtual machines can have IPv4 and IPv6 configurations on their network interfaces, which allows supported workloads to communicate with IPv4 and IPv6 clients. IPv6 subnets in Azure must be exactly /64. Use Azure portal, Azure CLI, Azure PowerShell, or infrastructure-as-code templates to create dual-stack networks and virtual machines. Microsoft provides current procedures for creating new dual-stack VMs and for adding dual-stack networking to existing VMs.

In Azure hub-spoke environments, plan IPv6 as a transition project rather than a per-VM setting. Review hub virtual networks, spoke virtual networks, firewalls, route tables, VPN or Virtual WAN connectivity, public IP addresses, load balancers, DNS, and network security groups before enabling IPv6 for workloads.

When planning IPv6 in Azure, verify:

- The Azure service supports IPv6 for the required scenario.
- The virtual network and subnet are dual-stack.
- IPv6 subnet prefixes are /64.
- Network interfaces have IPv6 configurations where required.
- NSG rules include IPv6 source and destination prefixes.
- Route tables and network virtual appliances support IPv6.
- Load balancers, public IP addresses, DNS records, and monitoring tools are configured for IPv6.
- Hybrid connectivity requirements are validated before production deployment.

Lab 3: Configure IPv6 in Windows Server

Please refer to our online lab to supplement your learning experience with exercises.

Knowledge check

Check your knowledge by answering these questions:

1. Is there a way to dynamically provide a DNS server to an IPv6 host?
2. You need to implement IPv6 internally, and you've determined unique local IPv6 addresses as the correct type to use for private networking. You must choose a 40-bit identifier that's part of the network so that you can use unique local IPv6 addresses, and a coworker said to use all zeros for the 40 bits. Can you explain why you shouldn't do this?
3. When setting up an IPv6 node, how many addresses should you configure?

> **Note:** To find the answers, refer to the **Knowledge check** slides in the PowerPoint presentation.

Learn more

For more information, refer to:

- Internet Protocol version 6 (IPv6) overview
- Configure IPv6 for advanced users
- Install and configure IP version 6
- Examine IPv6 addressing