12.1. IPv6 Feature

The current IP addressing standard, version 4, will eventually run out of unique addresses, so a new system is being developed. It is named IP version 6 or IPv6. You should know about the following IPv6 features:

Feature	Description
Geographic assignment of addresses	The Internet Corporation for Assigned Names and Numbers (ICANN) assigns IPv6 addresses based on the following strategy: • Public IPv6 addresses are grouped by major geographic region, such as a continent. • Inside each region, the address is further subdivided by each ISP. • Inside each ISP, the address is further subdivided for each customer or other smaller Internet registries.
Efficient route summarization	Route summarization combines blocks of addresses in a routing table as a single route. As IPv6 addresses are assigned by geographic region, then ISP, and then the customer, the route summarization of IPv6 addresses is efficient when compared to IPv4 route summarization.
No need for Network Address Translation (NAT) or Port Address Translation (PAT)	From the large amount of IP addresses afforded by IPv6, each device has a publicly registered address. Having a unique address for each device removes the need for NAT and PAT.
Native Internet Protocol Security (IPSec)	IPSec can be used to encrypt any traffic supported by the IP protocol. This includes Web, e-mail, Telnet, file transfer, and SNMP traffic as well as countless others. IPv6 has built-in support for the IPSec security protocol. Within an IPv4 environment, IPSec security

	features are available as add-ons but are required in IPv6.
Header improvements	IPv6 packet headers do not need to have their logical link address changed as the packet hops from router to router. This leads to a reduction in per-packet overhead.
Built-in Quality of Service (QoS)	Built-in support for bandwidth reservations make guaranteed data transfer rates possible. Within an IPv4 environment, Quality of Service features are available as add-ons but are not part of the native protocol.
Flow label	The *flow label* is a field in the IPv6 packet header. Packets belonging to the same stream, session, or flow share a common flow label value, making the session easily recognizable without having to open the inner packet to identify the flow.

© Sergey Gorokhod
MCT/MCSE/MCITP/MCTS/MCSA/CCSE/CCSA/CCNA/A+®
E-mail: sergey@infosec.co.il
Mob: (+972) 526848757

12.2. IPv6 Address

The IPv6 address is a 128-bit binary number. A sample IPv6 IP address looks like: 35BC:FA77:4898:DAFC:200C:FBBC:A007:8973. The following list describes the features of an IPv6 address:

- The address is made up of 32 hexadecimal numbers, organized into 8 quartets.
- The quartets are separated by colons.
- Each quartet is represented as a hexadecimal number between 0 and FFFF. Each quartet represents 16-bits of data (FFFF = 1111 1111 1111 1111).
- Leading zeros can be omitted in each section. For example, the quartet 0284 could also be represented by 284.
- Addresses with consecutive zeros can be expressed more concisely by substituting a double-colon for the group of zeros. For example:
    - FEC0:0:0:0:78CD:1283:F398:23AB
    - FEC0::78CD:1283:F398:23AB (concise form)
- If an address has more than one consecutive location where one or more quartets are all zeros, only one location can be abbreviated. For example, FEC2:0:0:0:78CA:0:0:23AB could be abbreviated as:
    - FEC2::78CA:0:0:23AB or
    - FEC2:0:0:0:78CA::23AB

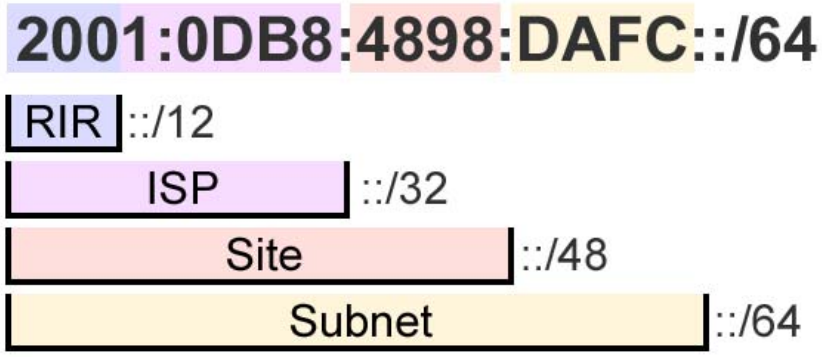
    But *not* FEC2::78CA::23AB

- The 128-bit address contains two parts:
    - The first 64-bits is known as the *prefix*. The prefix includes the network and subnet address. Because addresses are allocated based on physical location, the prefix also includes global routing information. The 64-bit prefix is often referred to as the *global routing* prefix.
    - The last 64-bits is the *interface ID*. This is the unique address assigned to an interface. ***Note****: Addresses are assigned to interfaces (network connections), not to the host. Technically, the interface ID is not a host address.*

The 64-bit prefix can be divided into various parts, with each part having a specific meaning.

- The *prefix length* identifies the number of bits in the relevant portion of the prefix. To indicate the prefix length, add a slash (/) followed by the prefix length number.
- Bits past the end of the prefix length are all binary 0s. For example, the full 64-bit prefix for address 2001:0DB8:4898:DAFC:200C:FBBC:A007:8973 is 2001:0DB8:4898:DAFC:0000:0000:0000:0000**/**64.
- Full quartets with trailing 0's in the prefix address can be omitted (for example 2001:0DB8:4898:DAFC::**/**64).
- If the prefix is not on a quartet boundary (this applies to any prefix that is not a multiple of 16), any hex values listed after the boundary should written as 0's. For example, the prefix 2001:0DB8:4898:DAFC::**/**56 should be written as 35BC:FA77:4898:DA**00**::**/**56. Remember, only *leading* 0's within a quartet can be omitted.
- Be aware that the prefix length number is a binary value, while the prefix itself is a hexadecimal value.

Global routing information is identified within the 64-bit prefix by subdividing the prefix using varying prefix lengths. The following graphic is an example of how the IPv6 prefix could be divided:



This sample assignment of IPv6 addresses is explained in the following table:

Prefix	Description
Regional Internet Registry (RIR)	The Internet Corporation for Assigned Names and Numbers (ICANN) is responsible for the assignment of IPv6 addresses. ICANN assigns a range of IP addresses to Regional Internet Registry (RIR) organizations. Each current regional organization corresponds roughly to a continent. The exact size of the address range assigned to the RIR may vary, but current guidelines assign a minimum prefix of 12-bits. In the above example, the RIR has been assigned a 12-bit prefix, and is responsible for addresses in the following range: 2000::/12 to 200F:FFFF:FFFF:FFFF::/64
Internet Service Provider (ISP)	A regional organization subdivides its block of IP addresses into smaller blocks and assigns those blocks to National Internet Registries (NIR), Local Internet Registries (LIR), or Internet Service Providers (ISP). Larger organizations can further subdivide the address space to allocate to smaller ISPs. The exact size of the address range assigned by the RIR may vary, but current guidelines assign a minimum prefix of 32-bits. In the above example, the ISP has been assigned a 32-bit prefix, and is therefore responsible for addresses in the following range: 2001:0DB8::/32 to 2001:0DB8:FFFF:FFFF::/64
Site	Individual companies and other organizations request blocks of IP addresses from an ISP for use in their private networks. Each network organized by a single entity is often called a *site*, although the exact definition of the term is under debate. Although the exact size of the address range assigned to a site may vary, by convention, each site is assigned a 48-bit site ID. In the above example, the site is responsible for managing the addresses in the following range: 2001:0DB8:4898::/48 to 2001:0DB8:4898:FFFF::/64

	ISPs typically follow these guidelines for assigning address ranges to sites: • By default, all sites that represent a network, including home networks, get an address with a 48-bit prefix. • Sites that require an address space larger than this might be assigned two consecutive blocks, or might be allocated an address with a 47-bit prefix. • If the network is known to have only a single subnet, the ISP might assign a 64-bit prefix. This is typically used for mobile devices. • If the network is known to have only a single device, such as a dialup connection, the ISP might assign a 128-bit prefix.
Subnet ID	Most networks receive an address range identified with a 48-bit prefix. The remaining 16-bits in the global routing prefix are then used by the local network administrator for creating subnets. In the example above, the site has received the prefix of 2001:0DB8:4898::/48. The following list shows some of the subnets that could be created by the administrator using a 64-bit prefix: 2001:0DB8:4898:0001::/64 2001:0DB8:4898:0002::/64 2001:0DB8:4898:0003::/64 . . . 2001:0DB8:4898:FFFD::/64 2001:0DB8:4898:FFFE::/64 2001:0DB8:4898:FFFF::/64

In most cases, individual interface IDs are not assigned by ISPs, but are rather generated automatically or managed by site administrators. Interface IDs must be unique within a subnet, but can be the same if the interface is on different subnets. All addresses that identify a single interface, except those that start with 000 binaries, but use a 64-bit interface ID that follows the modified EUI-64 format. On Ethernet networks, the modified EUI-64 format interface ID can be automatically derived from the MAC address using the following process:

1. The MAC address is split into 24-bit halves.

2. The hex constant FFFE is inserted between the two halves to complete the 64-bit address. For example, 20-0C-FB-BC-A0-07 becomes: 200C:FB**FF:FE**BC:A007.
3. The seventh bit of the MAC address (reading from left to right) is set to binary 1. This bit is called the *universal/local (U/L)* bit.
    - Modifying the seventh binary bit modifies the second hex value in the address.
    - For a MAC address of 20-0C-FB-BC-A0-07, the first two hex values translate to the following binary number: 0010 0000
    - Setting the seventh bit to 1 yields 0010 0010, which translates into 22 hex.

    In this example, the MAC address of 20-0C-FB-BC-A0-07 in modified EUI-64 format becomes: 2**2**0C:FB**FF:FE**BC:A007 (portions in red indicate modified values).

© Sergey Gorokhod
MCT/MCSE/MCITP/MCTS/MCSA/CCSE/CCSA/CCNA/A+®
E-mail: sergey@infosec.co.il
Mob: (+972) 526848757

12.3. IPv6 Address Types

In IPv6, addresses are assigned to interfaces (network connections). All interfaces are required to have some addresses, and interfaces can have more than one address. IPv6 identifies the following types of addresses:

<table>
<tr><th>Address Type</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">Unicast</td><td colspan="2">*Unicast* addresses are assigned to a single interface for the purpose of allowing that one host to send and receive data. Packets sent to a unicast address are delivered to the interface identified by that address.

Described below are three types of unicast addresses.</td></tr>
<tr><td>Link-local</td><td>*Link-local* addresses (also known as *local link* addresses) are addresses that are valid on only the current subnet.
Link-local addresses have a FE80::/10 prefix. This includes any address beginning with FE8, FE9, FEA, or FEB.
All nodes must have at least one link-local address, although each interface can have multiple addresses.
Routers never forward packets destined for local link addresses to other subnets.
Link-local addresses are used for automatic address configuration, neighbor discovery, or for subnets that have no routers.</td></tr>
<tr><td>Unique local</td><td>*Unique local* addresses are private addresses used for communication within a site or between a limited number of sites.
Unique local addresses have a FC00::/7 prefix. Currently, however, the 8th bit is always set to 1 to indicate that the address is local (and not global).</td></tr>
</table>

		Thus, addresses beginning with FC or FD are unique local addresses. • Following the prefix, the next 50-bits are used for the Global ID. The Global ID is generated randomly such that there is a high probability of uniqueness on the entire Internet. • Following the Global ID, the remaining 16-bits in the prefix are used for subnet information. • Unique local addresses are globally unique, but are not globally routable. Unique local addresses might be routed between sites by a local ISP. • Earlier IPv6 specifications defined a site-local address that was not globally unique and had a FEC0::/10 prefix. The site-local address has been replaced with the unique local address.
	Global unicast	*Global unicast* addresses are addresses that are assigned to individual interfaces that are globally unique (unique throughout the entire Internet). Global unicast addresses are any addresses that are not link-local, unique local, or multicast addresses. Currently, ISPs assign global unicast addresses with a 2000::/3 prefix (this includes any address beginning with a 2 or a 3). In the future, however, global unicast addresses might not have this restriction.
Multicast	*Multicast* addresses represent a dynamic group of hosts. Packets sent to a multicast address are sent to all interfaces identified by that address. By using a different multicast address for different functions, only the devices that need to participate in the particular function will respond to the multicast; devices that have no need to participate in the function will ignore the multicast. • All multicast addresses have a FF00::/8 prefix. • Multicast addresses that are restricted to the local link only have a FF02::/16 prefix. Packets starting with FF02 are not forwarded by routers.	

	• Multicast addresses with a FF01::/16 prefix are restricted to a single node. You should be familiar with the following well-known multicast addresses: • FF02::1 is for all nodes on the local link. This is the equivalent of the IPv4 subnet broadcast address. FF01::1 is for all interfaces on a node. • FF02::2 is for all routers on the local link. FF01::1 is for all routers on the node. • FF02::1:2 is for all DHCP servers or DHCP relay agents on the local link. DHCP relay agents forward these packets to other subnets.
Anycast	The *anycast* address is a unicast address that is assigned to more than one interface, typically belonging to different hosts. An anycast packet is routed to the nearest interface having that address (based on routing protocol decisions). • An anycast address is the same as a unicast address. Assigning the same unicast address to more than one interface makes it an anycast address. • You can have link-local, unique local, or global unicast anycast addresses. • When you assign an anycast address to an interface, you must explicitly identify the address as an anycast address (to distinguish it from a unicast address). • Anycast addresses can be used to locate the nearest server of a specific type, for example the nearest DNS or network time server.
Loopback	The local loopback address for the local host is 0:0:0:0:0:0:0:1 (also identified as ::1 or ::1/128). The local loopback address is not assigned to an interface. It can be used to verify that the TCP/IP protocol stack has been properly installed on the host.

Unspecified	The unspecified address is 0:0:0:0:0:0:0:0 (also identifies as :: or ::/128). The unspecified address is used when there is no IPv6 address. It is typically used during system startup when the host has not yet configured its address. The unspecified address should not be assigned to an interface.

Note: *There are no broadcast addresses in IPv6. IPv6 multicast addresses are used instead of broadcast addresses.*

© Sergey Gorokhod
MCT/MCSE/MCITP/MCTS/MCSA/CCSE/CCSA/CCNA/A+®
E-mail: sergey@infosec.co.il
Mob: (+972) 526848757

12.4. IPv6 Configuration

An IPv6 address is configured by any one of the following methods:

Method	Description
Static full assignment	*Static full assignment* is where the entire 128-bit IPv6 address and all other configuration information is statically assigned to the host.
Static partial assignment	*Static partial assignment* is where the prefix is statically assigned and the interface ID uses the modified EUI-64 format derived from the MAC address.
Stateless autoconfiguration	*Stateless autoconfiguration* is where clients automatically generate the interface ID, and learn the subnet prefix and default gateway through the Neighbor Discovery Protocol (NDP). NDP uses the following messages for autoconfiguration: • *Router solicitation* (RS) is a message sent by the client to request that routers respond. • *Router advertisement* (RA) is a message sent by the router periodically and in response to RS messages to inform clients of the IPv6 subnet prefix and the default gateway address. NDP is also used by hosts to discover the address of other interfaces on the network, replacing the need for Address Resolution Protocol (ARP). ***Note:*** *Even though NDP provides enough information for the addressing of the client and for clients to learn the addresses of other clients on the network, it does not provide the client with DNS server information or other IP configuration information besides the IP address and the default gateway.*

DHCPv6	IPv6 uses an updated version of DHCP (called DHCPv6) that operates in one of two different modes: • *Stateful* DHCPv6 is when the DHCP server provides each client with the IP address, default gateway, and other IP configuration information (such as the DNS server IP address). The DHCP server tracks the status (or state) of the client. • *Stateless* DHCPv6 does not provide the client an IP address and does not track the status of each client, but rather is used to supply the client with the DNS server IP address. Stateless DHCPv6 is most useful when used in conjunction with stateless autoconfiguration.

When a host starts up, it uses the following process to configure the IPv6 address for each interface:

1. The host generates an IPv6 address using the link-local prefix (FE80::/10) and modifying the MAC address to get the interface ID. For example, if the MAC address is 20-0C-FB-BC-A0-07, the link-local address for the interface would be: FE80::220C:FBFF:FEBC:A007.
2. The host then sends a neighbor solicitation (NS) message addressed to its own link-local address to see if the address it has chosen is already in use.
    - If the address is in use, the other network host responds with a neighbor advertisement (NA) message. The process stops and manual configuration of the host is required.
    - If the address is not in use (no NA message), the process continues.
3. The host waits for a router advertisement (RA) message from a router to learn the prefix.
    - If an RA message is not received, the host sends out a router solicitation (RS) message addressed to all routers on the subnet using the multicast address FF02::2.
    - The router sends out an RA message addressed to all interfaces on the subnet using the multicast address FF02::1.
    - If no routers respond, the host attempts to use stateful DHCPv6 to receive configuration information.

4. The RA message contains information that identifies how the IPv6 address and other information is to be configured. Possible combinations are:

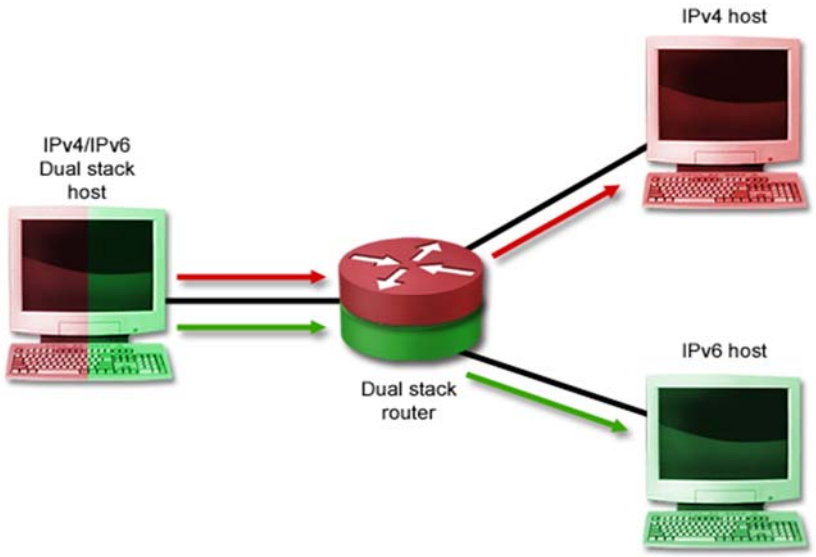
Configuration Method	Description
Use stateful autoconfiguration	Obtain the interface ID, subnet prefix, default gateway, and other configuration information from a DHCPv6 server. The host sends out a REQUEST message addressed to the multicast address FF02::1:2 to request this information from the DHCPv6 server.
Use stateless autoconfiguration	Set the interface ID automatically. Get the subnet prefix and default gateway from the RA message. Get DNS and other configuration information from a DHCPv6 server. The host sends out an INFORMATION-REQUEST message addressed to the multicast address FF02::1:2 to request this information from the DHCPv6 server.

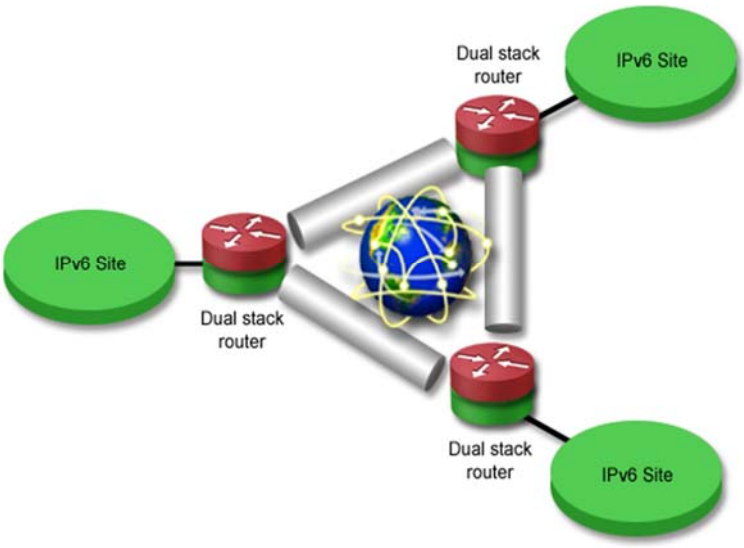
5. If a manual address or stateful autoconfiguration is used, the host sends an NS message to make sure the address is not already in use. If stateless autoconfiguration is used, the NS message at this step is unnecessary because the interface ID has already been verified in step 2.

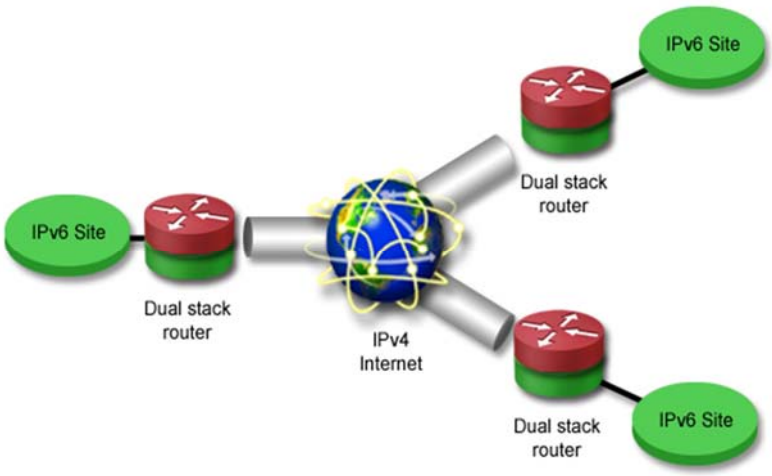
© Sergey Gorokhod
MCT/MCSE/MCITP/MCTS/MCSA/CCSE/CCSA/CCNA/A+®
E-mail: sergey@infosec.co.il
Mob: (+972) 526848757

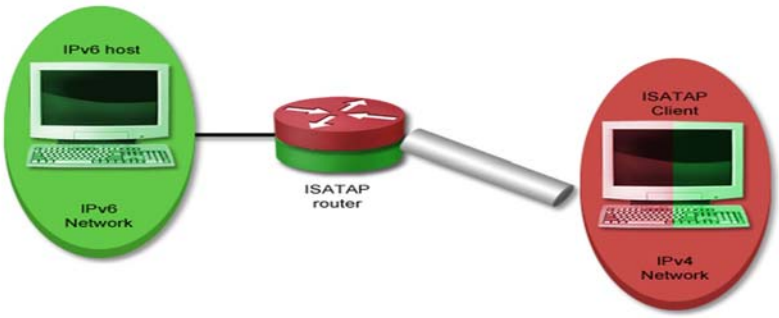
12.5. IPv6 Implementation

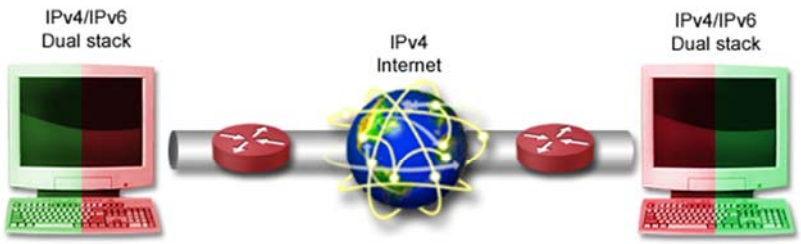
The worldwide implementation from IPv4 to IPv6 will be a long process. Although not yet widely adopted, you can implement IPv6 if your systems support it. As the implementation of IPv6 proceeds, there will be cases when compatibility with IPv4 is required. The following table lists various strategies for deploying IPv6:

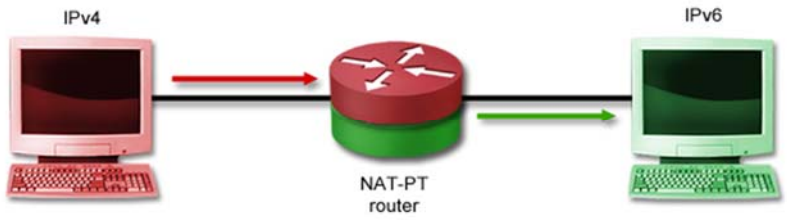
Method	Description
Dual stack	With a *dual stack* configuration, both the IPv4 and IPv6 protocol stacks run concurrently on a host. IPv4 is used to communicate with IPv4 hosts, and IPv6 is used to communicate with IPv6 hosts. When implemented on hosts, intermediate routers and switches must also run both protocol stacks.  Use a dual stack configuration to enable a host to communicate with both IPv4 and IPv6 hosts.
Tunneling	*Tunneling* wraps an IPv6 packet within an IPv4 packet, allowing IPv6 hosts or sites to communicate over the existing IPv4 infrastructure. With tunneling, a device encapsulates IPv6 packets in IPv4 packets for transmission across an IPv4 network, and then the packets are de-encapsulated to their original IPv6 packets by another device at the other end.

	Several tunneling solutions are listed below.	
	Manually configured tunnel	With a manually configured tunnel, tunnel endpoints are configured as point-to-point connections between devices. Manual tunneling: • Is configured between routers at different sites. • Requires dual-stack routers as the tunnel endpoints. Hosts can be IPv6-only hosts. • Works through NAT. • Uses a static (manual) association of an IPv6 address with the IPv4 address of the destination tunnel endpoint.  Because of the time and effort required for configuration, use manually configured tunnels only when you have a few sites that need to connect through the IPv4 Internet, or when you want to configure secure site-to-site associations.

	6-to-4 tunneling	With 6-to-4 tunneling, tunneling endpoints are configured automatically between devices. 6-to-4 tunneling: • Is configured between routers at different sites. • Requires dual-stack routers as the tunnel endpoints. Hosts can be IPv6-only hosts. • Works through NAT. • Uses a dynamic association of an IPv6 site prefix to the IPv4 address of the destination tunnel endpoint. • Automatically generates an IPv6 address for the site using the 2002::/16 prefix followed by the public IPv4 address of the tunnel endpoint router. For example, a router with the IPv4 address of 207.142.131.202 would serve the site with the following prefix: 2002:CF8E:83CA::/48 (CF8E:83CA is the hexadecimal equivalent of 207.142.131.202).  Use 6-to-4 tunneling to dynamically connect multiple sites through the IPv4 Internet. Because of its dynamic configuration, 6-to-4 tunneling is easier to administer than manual tunneling.

	Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)	The Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) is a tunneling method for use *within* a site to provide IPv6 communication over a private IPv4 network. ISATAP tunneling:
		
		• Is configured between individual hosts and an ISATAP router.
		• Requires a special dual-stack ISATAP router to perform tunneling, and dual-stack or IPv6-only clients. Dual stack routers and hosts perform tunneling when communicating on the IPv4 network.
		• Does *not* work through NAT.
		• Automatically generates link-local addresses that includes the IPv4 address of each host:
		  o The prefix is the well-known link-local prefix: FE80::/16.
		  o The remaining prefix values are set to 0.
		  o The first two quartets of the interface ID are set to 0000:5EFE.
		  o The remaining two quartets use the IPv4 address, written in either dotted-decimal or hexadecimal notation.
		
		A host with an IPv4 address of 192.168.12.155 would have the following IPv6 address when using ISATAP: FE80::5EFE:C0A8:0C9B (also designated as FE80::5EFE:192.168.12.155).
		
		 

		Use ISATAP to begin a transition to IPv6 *within* a site. You can start by adding a single ISATAP router and configuring each host as an ISATAP client.
	Teredo tunneling	Teredo tunneling establishes the tunnel between individual hosts so they can communicate through a private or public IPv4 network. Teredo tunneling: • Is configured between individual hosts. • Hosts are dual-stack hosts and perform tunneling of IPv6 to send on the IPv4 network. • Works through NAT.  Use Teredo tunneling to enable host-to-host communications between IPv6 devices through a public or private IPv4 network.
Network Address Translation-Protocol Translation (NAT-PT)	NAT-PT is a protocol that converts the IPv6 packet header into an IPv4 packet header, and vice versa. With NAT-PT, a translation table is referenced by the device, such as a Cisco router, as it converts the headers to ensure that the packet is sent to the correct host. This method is different than tunneling because the packet headers are converted between the IPv4 and IPv6, whereas tunneling wraps the IPv6 packet into an IPv4 packet. NAT-PT: • Is configured on a single router running NAT-PT.	

	• The router is a dual-stack router. Hosts run either IPv4 or IPv6.  Use NAT-PT to allow IPv4 hosts to communicate with IPv6 hosts.

© Sergey Gorokhod
MCT/MCSE/MCITP/MCTS/MCSA/CCSE/CCSA/CCNA/A+®
E-mail: sergey@infosec.co.il
Mob: (+972) 526848757

12.6. Advanced DHCP Configuration

In addition to configuring DHCP using the SDM interface, you can configure DHCP using the command line. Before discussing the configuration steps, be sure you understand the following terms:

- A *pool* is a range of IP addresses that the DHCP server can assign.
- DHCP *options* are the configuration parameters in addition to the IP address and mask that the DHCP server will deliver to hosts. Options include DNS server addresses and the default gateway address.
- An *exclusion* is a single address or a range of addresses in the pool that will not be assigned by the DHCP server.
- A *binding* is an IP address that is associated with a MAC address. Each time the specified host requests an IP address, the DHCP server will assign it the address specified in the binding.

Configuring DHCP through the command line involves the following steps:

1. Create a pool for the subnet. After creating the pool, define the following parameters for the pool:
    - The subnet address and mask.
    - DHCP options to assign (such as the default gateway, DNS server addresses, or domain name).
    - Configure the lease time.
2. Create a pool for each binding. Within the pool, configure:
    - The IP address and mask
    - The MAC address of the host
3. Configure any exclusions (addresses you don't want assigned).

***Note**: When you define the pool for the subnet, the router automatically responds to DHCP requests that come in on the interface whose IP address matches the pool you defined.*

The following table lists various commands for completing the DHCP configuration:

Use . . .	To . . .
Router(config)#ip dhcp pool WORD	Create a DHCP pool Pools are used to define a range of addresses to assign, as well as create bindings.
Router(dhcp-config)#network A.B.C.D m.m.m.m	Identify the subnet address and mask for the address pool.
Router(dhcp-config)#default-router A.B.C.D	Identify the default gateway address that will be assigned to hosts. This address should be inside the address pool. You can identify up to 8 addresses. However, most hosts can accept only a single default gateway address.
Router(dhcp-config)#dns-server A.B.C.D <A.B.C.D>	Identify DNS server addresses delivered to hosts. You can configure multiple DNS server addresses. Simply include multiple addresses separated by a space. You can specify up to 8 server addresses.
Router(dhcp-config)#domain-name WORD	Sets the domain name to be delivered to hosts.
Router(dhcp-config)#lease 0-365	Configures the IP address lease time (in days). Use the **infinite** keyword for a lease that does not expire.
Router(config)#ip dhcp excluded-address A.B.C.D <A.B.C.D>	Exclude addresses from being assigned. Identify start and ending addresses in the range, or a single address. Typically, you will exclude the DHCP server's own IP address from the range. ***Note****: This command is a global configuration command; it is not issued as part of the pool.*
Router(config)#ip dhcp pool WORD ***Router(dhcp-config)#host A.B.C.D m.m.m.m***	Create a binding. When you create a binding, you create a separate pool that is different than the pool that identifies the subnet. This pool must have a unique name.

Router(dhcp-config)#hardware-address aabb.ccdd.eeff	As part of the pool, you configure the IP address and mask that will be assigned to the host, as well as the MAC address of the host.
Switch(config)#interface vlan 1 ***Switch(config-if)#ip address dhcp***	Configure a Cisco device to get its IP address from the DHCP server. Most routers and servers have static IP addresses and do not use DHCP for obtaining an IP address. However, you could create a binding to make sure the same address is always assigned to network infrastructure devices such as servers, switches, and routers.

Example

In the following example, the router has an IP address of 172.19.1.129/25 assigned to its Fa0/1 interface. The following commands create a pool for the subnet, configures DNS and default gateway addresses to assign to hosts, sets the lease time to 10 days, excludes the router's IP address from the pool, and creates a binding for a host named Dns-Srv1 that assigns that host an address of 172.19.1.132 each time it requests an address.

Router#ip dhcp pool SubnetA
Router(dhcp-config)#network 172.19.1.128 255.255.255.128
Router(dhcp-config)#default-router 172.19.1.129
Router(dhcp-config)#dns-server 172.19.1.132
Router(dhcp-config)#lease 10
Router(dhcp-config)#exit
Router(config)#ip dhcp excluded-address 172.19.1.129
Router(config)#ip dhcp pool Dns-Srv1
Router(dhcp-config)#host 172.19.1.132 255.255.255.128
Router(dhcp-config)#hardware-address 0fe8.11a7.ab89

12.7. Advanced NAT Configuration

Network Address Translation (NAT) allows you to connect a private network to the Internet without obtaining registered addresses for every host. Private addresses are translated to the public address of the NAT router. When configuring NAT, you have the following options:

Method	Description
Static	Use static translation to translate a single outside address to a single inside address. To configure static NAT, use the following general process: 1. Define a static map that associates the inside address with the outside address. 2. Identify which router interface is the inside interface, and which interface is the outside interface.
Overloaded with PAT	Use overloaded NAT with Port Address Translation (PAT) to translate multiple inside addresses to a single public address. To configure overloaded NAT, use the following general process: 1. Create an access list that allows the specified inside addresses to be translated. 2. Link the access list to the internal interface using the **overloaded** option. The IP address assigned to the outside interface will automatically be used as the outside address for all inside hosts. 3. Identify which router interface is the inside interface, and which interface is the outside interface.
Dynamic	Use dynamic translation to translate a range of outside addresses to a range of inside addresses. To configure dynamic NAT, use the following general process: 1. Define the pool of outside addresses that can be used for translation.

	2. Create an access list that allows the specified inside addresses to be translated. 3. Link the pool with the access list. 4. Identify which router interface is the inside interface, and which interface is the outside interface. ***Note**: If the number of outside addresses defined in the pool is less than the number of inside addresses allowed by the access list, the number of inside hosts that can gain outside access will be limited to the number of outside addresses in the pool. To allow a greater number of inside hosts to use a smaller number of outside addresses, add the **overloaded** parameter to step 3. This uses dynamic NAT with PAT.*

The following table lists the configuration steps and commands for each method.

Method	Configuration Task	Command Examples
Static NAT	Configure static mappings (mapping inside local addresses to outside local addresses)	***Router(config)#ip nat inside source static 192.168.1.1 203.44.55.1***
	Identify inside and outside interfaces	***Router(config)#interface ethernet0*** ***Router(config-if)#ip nat inside*** ***Router(config-if)#interface serial0*** ***Router(config-if)#ip nat outside***
Overloaded with PAT	Identify allowed translated inside local addresses	***Router(config)#access-list 1 permit 192.168.1.0 0.0.0.255***
	Associate the allowed list with the outside interface and identify the translation type as overloaded	***Router(config)#ip nat inside source list 1 interface serial0 overload***
	Identify inside and outside interfaces	***Router(config)#interface ethernet0*** ***Router(config-if)#ip nat inside*** ***Router(config-if)#interface serial0*** ***Router(config-if)#ip nat outside***

Dynamic NAT	Define an inside global address pool	***Router(config)#ip nat pool pooled_addr 203.44.55.1 203.44.55.254 netmask 255.255.255.0***
	Identify allowed translated inside local addresses	***Router(config)#access-list 1 permit 192.168.1.0 0.0.0.255***
	Associate the allowed list with the pool	***Router(config)#ip nat inside source list 1 pool pooled_addr***
	Identify inside and outside interfaces	***Router(config)#interface ethernet0*** ***Router(config-if)#ip nat inside*** ***Router(config-if)#interface serial0*** ***Router(config-if)#ip nat outside***

Example
In this example, you have been given six public addresses from your ISP (177.211.5.89 to 177.211.5.94) using a 29-bit mask. You will use one of those addresses for the router interface, and want to use the remaining 5 addresses for dynamic NAT with PAT for inside hosts. You want to configure Internet access for all inside hosts on the 10.0.1.0/24 network. The following commands create the pool, define the allowed inside addresses, link the access list to the pool, and configure the inside and outside interfaces.

Router(config)#ip nat pool public_addr 177.211.5.90 177.211.5.94 netmask 255.255.255.248
Router(config)#access-list 1 permit 10.0.1.0 0.0.0.255
Router(config)#ip nat inside source list 1 pool public_addr overloaded
Router(config)#int eth0/1
Router(config)#ip addr 10.0.1.1 255.255.255.0
Router(config-if)#ip nat inside
Router(config-if)#int ser0/1/0
Router(config-if)#ip addr 177.211.5.89 255.255.255.248
Router(config-if)#ip nat outside

***Note**: The **ip nat pool** command can use the **prefix-length** keyword instead of the **netmask** keyword as in the following example:*

ip nat pool public_addr 177.211.5.89 177.211.5.94 netmask 29

Use the following commands to monitor NAT:

Use . . .	To . . .
Router#clear ip nat translation	Clear (delete) the dynamic entries in the NAT table.
Router#show ip nat statistics	View counters for packets and NAT table entries, as well as basic configuration information.
Router#show ip nat translations	View the NAT/PAT translation table entries.

© Sergey Gorokhod
MCT/MCSE/MCITP/MCTS/MCSA/CCSE/CCSA/CCNA/A+®
E-mail: sergey@infosec.co.il
Mob: (+972) 526848757