

# Module 5: Implementing IPv4

## Contents:

### Module Overview

**Lesson 1:** **Overview of TCP/IP**

**Lesson 2:** **Understanding IPv4 Addressing**

**Lesson 3:** **Subnetting and Supernetting**

**Lesson 4:** **Configuring and Troubleshooting IPv4**

**Lab:** **Implementing IPv4**

### Module Review and Takeaways

## Module Overview

IPv4 is the network protocol used on the Internet and local area networks. To ensure that you can understand and troubleshoot network communication, it is essential that you understand how IPv4 is implemented. In this module, you will see how to implement an IPv4 addressing scheme, and determine and troubleshoot network-related problems.

## Objectives

After completing this module, you should be able to:

- Describe the TCP/IP protocol suite.
- Describe IPv4 addressing.
- Determine a subnet mask necessary for subnetting or supernetting.
- Configure IPv4 and troubleshoot IPv4 communication.

## Lesson 1 : Overview of TCP/IP

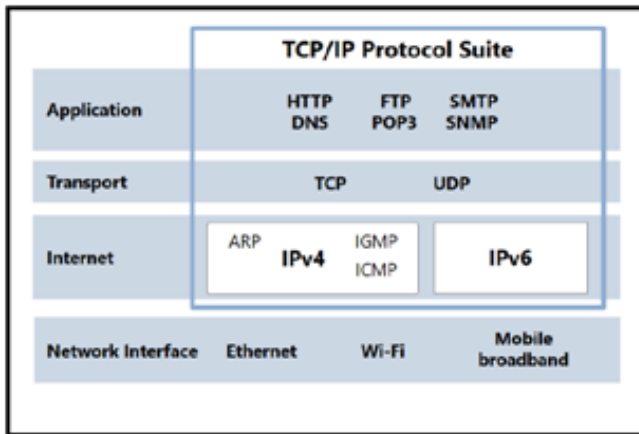
TCP/IP is an industry standard suite of protocols that provides communication in a heterogeneous network. This lesson provides an overview of IPv4, how it relates to other protocols, and how IPV4 and other protocols enable network communication. This lesson also covers sockets, which are used by network programs when communicating with programs on a remote host. Combined together, this lesson provides a foundation for understanding and troubleshooting network communication.

### Lesson Objectives

After completing this lesson, you should be able to:

- Describe the elements of the TCP/IP suite of protocols.
- Describe the individual protocols that make up the TCP/IP suite.
- Describe TCP/IP application layer protocols.
- Describe a socket, and identify port numbers for specified protocols.

### The TCP/IP Protocol Suite



The tasks performed by TCP/IP in the communication process are distributed across protocols. These protocols are organized into four distinct layers within the TCP/IP stack:

- Application layer. Programs use application layer protocols to access network resources.  
Application layer protocols include:
  - o Hypertext Transfer Protocol (HTTP)
  - o File Transfer Protocol (FTP)
  - o Simple Mail Transfer Protocol (SMTP)
  - o Domain Name System (DNS)
  - o Post Office Protocol 3 (POP3)
  - o Simple Network Management Protocol (SNMP)
- Transport layer. Transport layer protocols control data transfer reliability on the network. Transport layer protocols include:
  - o Transmission Control Protocol (TCP)
  - o User Datagram Protocol (UDP)
- Internet layer. The Internet layer protocols control packet movement between networks. Internet layer protocols include:
  - o Address Resolution Protocol (ARP)

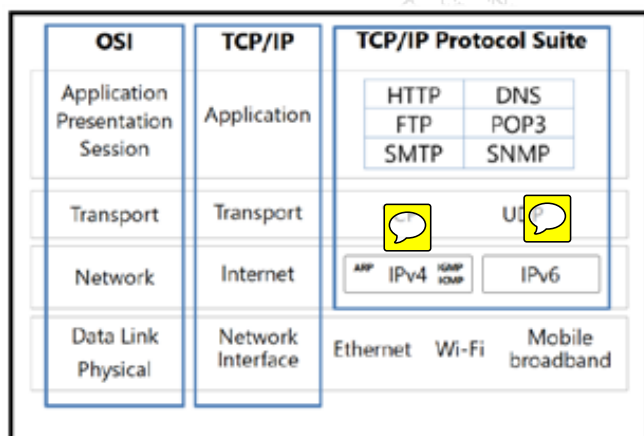
- o Internet Group Management Protocol (IGMP)
- o Internet Control Message Protocol (ICMP)
- Network interface layer. The network interface layer protocols define how datagrams from the Internet layer are transmitted on the media.

## Benefits of Architecture Layers

Rather than creating a single protocol, dividing the network functions into a stack of separate protocols provides several benefits:

- Separate protocols make it easier to support a variety of computing platforms.
- Creating or modifying protocols to support new standards does not require modification of the entire protocol stack.
- Multiple protocols that operate at the same layer enable programs to select the protocols that provide only the required level of service.
- Because the stack is split into layers, personnel who are uniquely qualified in the operations of particular layers can develop protocols simultaneously.

## Protocols in the TCP/IP Suite



The Open Systems Interconnection (OSI) model defines distinct layers

related to packaging, sending, and receiving data transmissions over a network. The layered suite of protocols that form the TCP/IP stack carry out these functions.

## Application Layer

The application layer of the TCP/IP model corresponds to the application, presentation, and session layers of the OSI model. This layer provides services and utilities that enable programs to access network resources.

## Transport Layer

The transport layer corresponds to the transport layer of the OSI model and is responsible for end-to-end communication using TCP or User Datagram Protocol (UDP). The TCP/IP protocol suite offers application programmers the choice of TCP or UDP as a transport layer protocol:

- TCP provides connection-oriented reliable communications for programs. Connection-oriented communication confirms that the destination is ready to receive data before it sends the data. To make communication reliable, TCP confirms that all packets are received. Reliable communication is desired in most cases, and is used by most programs. Web servers, File Transfer Protocol (FTP) clients, and other programs that move large amounts of data use TCP.
- UDP provides connectionless and unreliable communication. When using UDP, reliable delivery is the responsibility of the program. Programs use UDP for faster communication with less overhead than TCP. Programs such as streaming audio and video use UDP so that a single missing packet does not delay playback. UDP is also used by programs that send small amounts of data, such as Domain Name System (DNS) name lookups.

The transport layer protocol that a program uses is determined by the developer of a program, and is based on the communication requirements

of the program.

## Internet Layer

The Internet layer corresponds to the network layer of the OSI model and consists of several separate protocols, including: IP; Address Resolution Protocol (ARP); Internet Group Management Protocol (IGMP); and Internet Control Message Protocol (ICMP). The protocols at the Internet layer encapsulate transport layer data into units called *packets*, address them, and then route them to their destinations.

The Internet layer protocols are:

- IP. IP is responsible for routing and addressing. The Windows® 8 operating system and the Windows Server® 2012 operating system implement a dual-layer IP protocol stack, which includes support for both IPv4 and IPv6.
- ARP. ARP is used by IP to determine the media access control (MAC) address of local network adapters—that is, adapters installed on computers on the local network—from the IP address of a local host. ARP is broadcast-based, meaning that ARP frames cannot transit a router and are therefore localized. Some implementations of TCP/IP provide support for Reverse ARP (RARP) in which the MAC address of a network adapter is used to determine the corresponding IP address.
- IGMP. IGMP provides support for multitasking programs over routers in IPv4 networks.
- ICMP. ICMP sends error messages in an IP-based network.

## Network Interface Layer

The *network interface layer* corresponds to the data link and physical layers of the OSI model. The network interface layer is sometimes referred to as the *link layer* or *data link layer*. The network interface layer

specifies the requirements for sending and receiving packets on the network media. This layer is not typically considered part of the TCP/IP protocol suite because the tasks are performed by the combination of the network adapter driver and the network adapter.

## TCP/IP Applications

### Some common application layer protocols:

- HTTP
- HTTPS
- FTP
- RDP
- SMB
- SMTP
- POP3

Programs use application layer protocols to communicate over the network. A client and server must use the same application layer protocol to communicate. The following table lists some common application layer protocols.

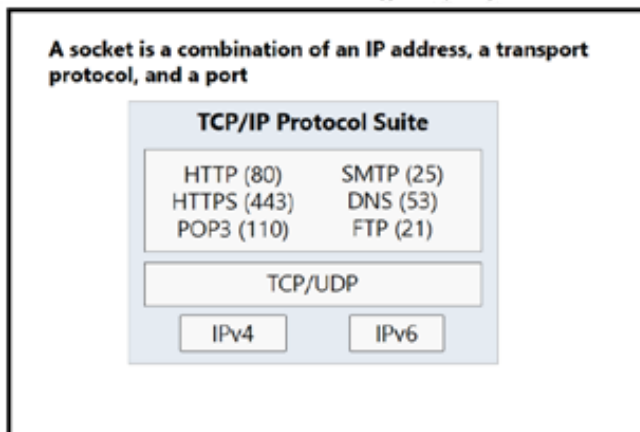
Protocol	Description
HTTP	Used for communication between web browsers and web servers.
HTTP/Secure (HTTPS)	A version of HTTP that encrypts communication between web browsers and web servers.
FTP	Used to transfer files between FTP clients and servers.
Remote Desktop Protocol (RDP)	Used to remotely control a computer that is running Windows operating systems over a network.
Server Message Block (SMB)	Used by servers and client computers for file and printer sharing.
Simple Mail Transfer Protocol (SMTP)	Used to transfer email messages over the Internet.
Post Office Protocol version 3 (POP3)	Used to retrieve messages from some email servers.



Internet Message  
Access Protocol (IMAP)

Used to retrieve messages from some email servers.

## What Is a Socket?



A socket is a combination of an IP address, a transport protocol, and a port number. When a program wants to establish communication with a program on a remote host, it creates either a TCP or a UDP socket, as appropriate. A socket requires the following information as part of the communication process:

- The transport protocol that the program uses, which could be TCP or UDP.
- The TCP or UDP port numbers that the programs are using.
- The IPv4 or IPv6 address of the source and destination hosts.

## Well-Known Ports

Programs are assigned a port number between 0 and 65,535. The first 1,024 ports are called *well-known ports* and have been assigned to specific programs. Programs listening for connections use consistent port numbers to make it easier for client programs to connect. If a program listens on a non-standard port number, then you need to specify the port number when connecting to it. Client programs typically use a random source port number above 1024. The following table identifies some of these well-known ports.



Port	Protocol	Program
80	TCP	HTTP used by a web server
443	TCP	HTTPS for a secure web server
110	TCP	POP3 used for email retrieval
143	TCP	IMAP used for email retrieval
25	TCP	SMTP used for sending email messages
53	UDP	DNS used for most name resolution requests
53	TCP	DNS used for zone transfers
20, 21	TCP	FTP used for file transfers

You need to know the port numbers that programs use so you can configure firewalls to allow communication. Most programs have a default port number for this purpose, but it can be changed when required. For example, some web-based programs run on a port other than port 80 or port 443.

 **Question:** Are there other well-known ports that you can think of?

## Lesson 2: Understanding IPv4 Addressing

Understanding IPv4 network communication is critical to ensuring that you can implement, troubleshoot, and maintain IPv4 networks. One of the core components of IPv4 is addressing. Understanding addressing, subnet masks, and default gateways allows you to identify the proper communication between hosts. To identify IPv4 communication errors, you need to understand how the communication process is designed to work.

### Lesson Objectives

After completing this lesson, you should be able to:

- Describe IPv4 Addressing.
- Identify public and private IPv4 addresses.
- Explain how dotted decimal notation relates to binary numbers.
- Describe a simple IPv4 network with classful addressing.
- Describe a more complex IPv4 network with classless addressing.

## IPv4 Addressing

- Each networked computer must be assigned a unique IPv4 address
- Network communication for a computer is directed to the IPv4 address of the computer
- Each IPv4 address contains:
  - ✓ Network ID, identifying the network
  - ✓ Host ID, identifying the computer
- The subnet mask identifies which part of the IPv4 address is the network ID (255) and which is the host ID (0)

IP address	172	16	0	10
Subnet mask	255	255	0	0
Network ID	172	16	0	0
Host ID	0	0	0	10

To configure network connectivity, you must be familiar with IPv4 addresses and how they work.

Network communication for a computer is directed to the IPv4 address of that computer.

Therefore, each networked computer must be assigned a unique IPv4 address.

Each IPv4 address is 32 bits long. To make IP addresses more readable, they are displayed in dotted decimal notation. Dotted decimal notation divides a 32-bit IPv4 address into four groups of 8 bits, which are converted to a decimal number between zero and 255. A decimal point separates the decimal numbers. Each decimal number is called an *octet*. As an example, this IP address contains of four octets: 172.16.0.10.

## Subnet Mask

Each IPv4 address is composed of a network identification (ID) and a host ID. The *network ID* identifies the network on which the computer is located. The *host ID* uniquely identifies the computer on that specific network. A *subnet mask* identifies which part of an IPv4 address is the network ID and which part is the host ID.


In the simplest scenarios, each octet in a subnet mask is either 255 or 0. A 255 represents an octet that is part of the network ID, while a 0 represents an octet that is part of the host ID. For example, a computer with an IP address of 172.16.0.10 and a subnet mask of 255.255.0.0 has a network ID of 172.16.0.0 and a host ID of 0.0.0.10.

You can present subnet masks in network prefix notation, which represents how many continuous binary numbers with the value of 1 are contained in the subnet mask. For example, the network 172.16.0.0 that has the subnet mask 255.255.0.0 can be presented as 172.16.0.0/16. The /16 represents the 16 bits that have a value of 1 when the subnet mask is represented in a binary format:

11111111.11111111.00000000.00000000. The following table represents the default subnet masks and their network prefix notation.

### Default Subnet Masks (Network Prefix Notation)

Address Class		Bits for Subnet Mask	Network Prefix
Class A	255.0.0.0	11111111 00000000 00000000 00000000	/8
Class B	255.255.0.0	11111111 11111111 00000000 00000000	/16
Class C	255.255.255.0	11111111 11111111 11111111 00000000	/24

 **Note:** The terms network, subnet, and VLAN (virtual local area network) are often used interchangeably. A large network is often subdivided into subnets, and VLANs are configured on routers or on Layer 3 switches to represent subnets.

## Default Gateway

A *default gateway* is a device, usually a router, on a TCP/IP network that forwards IP packets to other networks. The multiple internal networks in an organization can be referred to as an *intranet*.

On an intranet, any given network might have several routers that connect it to other networks, both local and remote. You must configure one of the routers as the default gateway for local hosts. This enables the local hosts to communicate with hosts on remote networks.



Before a host sends an IPv4 packet, it uses its own subnet mask to determine whether the destination host is on the same network or on a remote network. If the destination host is on the same network, the sending host transmits the packet directly to the destination host. If the destination host is on a different network, the host transmits the packet to a router for delivery.

When a host transmits a packet to a remote network, IPv4 consults the internal routing table to determine the appropriate router for the packet to reach the destination subnet. If the routing table does not contain any routing information about the destination subnet, IPv4 forwards the packet to the default gateway. The host assumes that the default gateway contains the required routing information. The default gateway is used in most cases.

Client computers usually obtain their IP addressing information from a Dynamic Host Configuration Protocol (DHCP) server. This is more straightforward than assigning a default gateway manually on each host. Most servers have a static IP configuration that is assigned manually.

? **Question:** How is network communication affected if a default gateway is configured incorrectly?

## Public and Private IPv4 Addresses

Public	Private
<ul style="list-style-type: none"><li>• Required by devices and hosts that connect directly to the Internet</li><li>• Must be globally unique</li><li>• Routable on the Internet</li><li>• Must be assigned by IANA/RIR</li></ul>	<ul style="list-style-type: none"><li>• Not routable on the Internet<ul style="list-style-type: none"><li>• 10.0.0.0/8</li><li>• 172.16.0.0/12</li><li>• 192.168.0.0/16</li></ul></li><li>• Can be assigned locally by an organization</li><li>• Must be translated to access the Internet</li></ul>
	

Devices and hosts that connect directly to the Internet require a public IPv4 address. Hosts and devices that do not connect directly to the Internet do not require a public IPv4 address.

### Public IPv4 Addresses

Public IPv4 addresses must be unique. Internet Assigned Numbers Authority (IANA) assigns public IPv4 addresses to regional Internet registries, which then assign IPv4 addresses to Internet service providers (ISPs). Usually your ISP allocates you one or more public addresses from its address pool. The number of addresses that your ISP

allocates to you depends upon how many devices and hosts that you connect to the Internet.

### Private IPv4 Addresses

Computers and devices that need to connect to the Internet must be configured with public IP addresses. However, the number of public IPv4 addresses is becoming limited. Since organizations cannot obtain public IPv4 address for every corporate computer, they use private IP addressing instead.

Because private IP addresses are not routable on the Internet, computers configured with private IP address cannot access the Internet.

Technologies such as network address translation (NAT) enable administrators to use a relatively small number of public IPv4 addresses and, at the same time, enable local hosts to connect to remote hosts and services on the Internet.

IANA defines the address ranges in the following table as private. Internet-based routers do not forward packets originating from, or destined to, addresses in these ranges.

Network	Range
10.0.0.0/8	10.0.0.0-10.255.255.255
172.16.0.0/12	172.16.0.0-172.31.255.255
192.168.0.0/16	192.168.0.0-192.168.255.255

## How Dotted Decimal Notation Relates to Binary Numbers

**Dotted decimal notation is based on the decimal number system, but computers use IP addresses in binary**

Within an 8-bit octet, each bit position has a decimal value

- A bit that is set to 0 always has a zero value
- A bit that is set to 1 can be converted to a decimal value
- The low-order bit represents a decimal value of 1
- The high-order bit represents a decimal value of 128

If all bits in an octet are set to 1, then the octet's decimal value is 255, the highest possible value of an octet:

$$128 + 64 + 32 + 16 + 8 + 4 + 2 + 1$$

When you assign IP addresses, you use dotted decimal notation. Dotted decimal notation is based on the decimal number system. However, in the background, computers use IP addresses in binary. To understand how to choose a subnet mask for complex networks, you must understand IP addresses in binary.

Within an 8-bit octet, each bit position has a decimal value. A bit that is



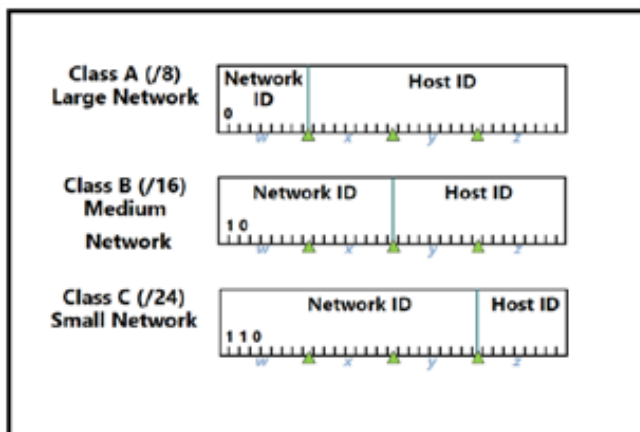
set to 0 always has a zero value. A bit that is set to 1 can be converted to a decimal value. The *low-order bit* is the rightmost bit in the octet, and it represents a

decimal value of 1. The *high-order bit* is the leftmost bit in the octet, and it represents a decimal value of 128. If all bits in an octet are set to 1, then the octet's decimal value is 255, that is:  $128 + 64 + 32 + 16 + 8 + 4 + 2 + 1$ . 255 is the highest possible value of an octet.

Most of the time, you can use a calculator to convert decimal numbers to binary and vice versa. The Windows operating systems include the Calculator app that can perform decimal-to-binary conversions, as shown in the following example.

Binary	Dotted decimal notation
10000011 01101011 00000011 00011000	131.107.3.24

## Simple IPv4 Implementations



## IPv4 Address Classes

The IANA organizes IPv4 addresses into classes.

Each class of address has a different default subnet mask that defines the number of valid hosts on the network. IANA has named the IPv4 address classes from *Class A* through *Class E*.




Classes A, B, and C are IP networks that you can assign to IP addresses on host computers.

Computers and programs use class D addresses for multicasting. The IANA reserves Class E for experimental use. An addressing process that uses an A, B or C class is called *classful addressing*. A network that uses an A, B or C class is called a *classful network*.


The following table lists the characteristics of each IP address class.

Class	First octet	Default subnet mask	Number of networks	Number of hosts per network
A	1-127	255.0.0.0	126	16,777,214
B	128-191	255.255.0.0	16,384	65,534
C	192-223	255.255.255.0	2,097,152	254

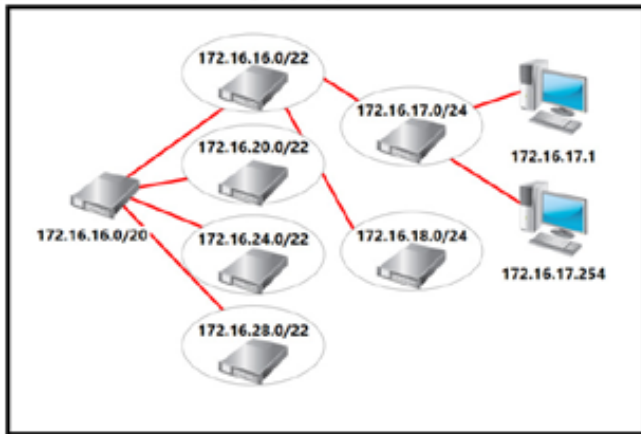
 **Note:** The Internet no longer uses routing based on the default subnet mask of IPv4 address classes.

## Simple IPv4 Networks

You can use subnetting to divide a large network into multiple smaller networks. In simple IPv4 networks, the subnet mask defines full octets as part of the network ID and host ID. A 255 represents an octet that is part of the network ID, and a 0 represents an octet that is part of the host ID. For example, you can use the 10.0.0.0 network with a subnet mask of 255.255.0.0 to create 256 smaller networks.

 **Note:** The IPv4 address 127.0.0.1 is used as a loopback address; you can use this address to test the local configuration of the IPv4 protocol stack. Consequently, the network address 127 is not permitted for configuring IPv4 hosts.

## More Complex IPv4 Implementations



In complex networks, subnet masks might not be simple combinations of 255 and 0. Rather, you might subdivide one octet with some bits that are for the network ID, and some that are for the host ID. This allows you to have the specific number of subnets and hosts that you require.

172.16.0.0 with the subnet mask 255.255.240.0 is an example of a subnet mask that can be used to divide a class B network into 16 subnets.

In many cases, rather than using a dotted decimal representation of the subnet mask, the number of bits in the network ID is specified instead. This is called *Classless Interdomain Routing* (CIDR). This is an example of CIDR notation: 172.16.0.0/20

### Variable Length Subnet Masks

Modern routers support the use of variable length subnet masks, which allow you to create subnets of different sizes when you subdivide a larger network. For example, you could subdivide a small network with 256 addresses into three smaller networks of 128 addresses, 64 addresses, and 64 addresses. This allows you to use IP addresses in a network more efficiently.

**Question:** Does your organization use simple or complex networking?

## Lesson 3: Subnetting and Supernetting

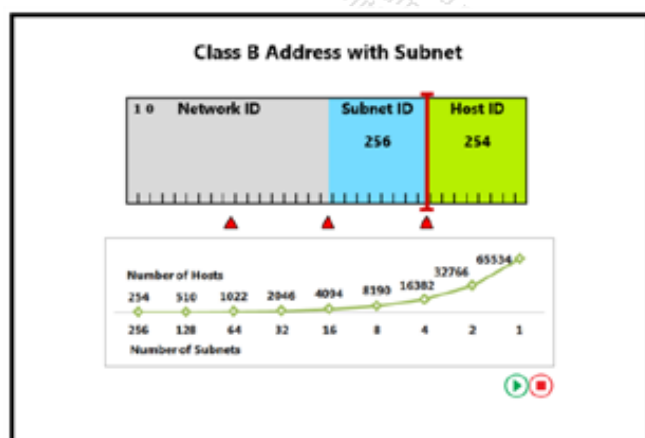
In most organizations, you need perform subnetting to divide your network into smaller subnets and allocate those subnets for specific purposes or locations. To do this, you need to understand how to select the correct number of bits to include in the subnet masks. In some cases, you may also need to combine multiple networks into a single larger network through supernetting.

## Lesson Objectives

After completing this lesson, you should be able to:

- Describe how bits are used in a subnet mask or prefix length.
- Identify when to use subnetting.
- Calculate a subnet mask that supports a specific number of subnet addresses.
- Calculate a subnet mask that supports a specific number of host addresses.
- Identify an appropriate subnet mask for a scenario.
- Describe supernetting.


## How Bits Are Used in a Subnet Mask or Prefix Length



In simple networks, subnet masks are composed of four octets, and each octet has a value of 255 or 0. If the octet is 255, that octet is part of the

network ID. If the octet is 0, that octet is part of the host ID.

In complex networks, you can convert the subnet mask to binary, and evaluate each bit in the subnet mask. A subnet mask is composed of contiguous 1s and 0s. The 1s start at the leftmost bit and continue uninterrupted until the bits change to all 0s.

 **Note:** Windows PowerShell® cmdlets for configuring IPv4 use a prefix length value rather than a subnet mask to define the number of network bits. The prefix length is the same number of bits used by CIDR notation.

You can identify the network ID of a subnet mask by the 1s. You can identify the host ID by the 0s. Any bits taken from the host ID and allocated to the network ID must be contiguous with the original network ID:

- Each 1 bit is part of the network ID.
- Each 0 bit is part of the host ID.

The mathematical process that is used to compare an IP address and a subnet mask is called *ANDing*.

When you use more bits for the subnet mask, you can have more subnets, but you can then have fewer hosts on each subnet. Using more bits than you need allows for subnet growth, but limits growth for hosts. Using fewer bits than you need allows for growth in the number of hosts you can have, but limits growth in subnets.

The following is a list of the bits used on the slide, and the corresponding number of subnets and hosts:

- 8 bits – 256 subnets, 254 hosts
- 7 bits – 128 subnets, 510 hosts
- 6 bits – 64 subnets, 1,022 hosts
- 5 bits – 32 subnets, 2,046 hosts
- 4 bits – 16 subnets, 4,094 hosts
- 3 bits – 8 subnets, 8,190 hosts
- 2 bits – 4 subnets, 16,382 hosts
- 1 bit – 2 subnets, 32,766 hosts
- 0 bits – 1 subnets, 65,534 hosts

## The Benefits of Using Subnetting

When you subdivide a network into subnets, you create a unique ID for each subnet that is derived from the main network ID

By using subnets, you can:

- Use a single network address across multiple locations
- Reduce network congestion by segmenting traffic
- Increase security by using firewalls
- Overcome limitations of current technologies

When you subdivide a network into subnets, you must create a unique ID for each subnet. These unique IDs are derived from the main network ID when you allocate some of the bits in the host ID to the network ID. This enables you to create more networks.

By using subnets, you can:

- Use a single, large network across multiple physical locations.
- Reduce network congestion by segmenting traffic and reducing

broadcasts on each segment.

- Increase security by dividing the network and using firewalls to control communication.
- Overcome limitations of current technologies, such as exceeding the maximum number of hosts that each segment can have.

## Calculating Subnet Addresses

When determining subnet addresses you should:

- Choose the number of subnet bits based on the number of subnets required
- Use  $2^n$  to determine the number of subnets available from  $n$  bits

For five locations, the following three subnet bits are required:

- 5 locations = 5 subnets required
- $2^2 = 4$  subnets (not enough)
- $2^3 = 8$  subnets

Before you define a subnet mask, estimate how many subnets and hosts for each subnet you may require. This enables you to use the appropriate number of bits for the subnet mask.

You can calculate the number of subnet bits that you need in the network. Use the formula  $2^n$ , where  $n$  is the number of bits. The result is the number of subnets that your network requires.

The following table indicates the number of subnets that you can create by using a specific number of bits.


Number of bits ( $n$ )	Number of subnets ( $2^n$ )
1	2
2	4
3	8

4	16
5	32
6	64

To determine the subnet addresses quickly, you can use the lowest value bit in the subnet mask. For example, if you choose to subnet the network 172.16.0.0 by using 3 bits, this means the subnet mask is 255.255.224.0. The decimal 224 is 11100000 in binary, and the lowest bit has a value of 32, so that is the increment between each subnet address.

The following table shows the subnet addresses for this example; the 3 bits that you have chosen to use to subnet the network are in bold type.

Binary network number	Decimal network number
172.16. <b>000</b> 00000.00000000	172.16.0.0
172.16. <b>001</b> 00000.00000000	172.16.32.0
172.16. <b>010</b> 00000.00000000	172.16.64.0
172.16. <b>011</b> 00000.00000000	172.16.96.0
172.16. <b>100</b> 00000.00000000	172.16.128.0
172.16. <b>101</b> 00000.00000000	172.16.160.0
172.16. <b>110</b> 00000.00000000	172.16.192.0
172.16. <b>111</b> 00000.00000000	172.16.224.0

 **Note:** You can use a subnet calculator to determine the appropriate subnets for your network, rather than calculating them manually. Subnet calculators are widely available on the Internet.

## Calculating Host Addresses



When determining host addresses you should:

- Choose the number of host bits based on the number of hosts that you require on each subnet
- Use  $2^n - 2$  to determine the number of hosts that are available on each subnet

For subnets with 100 hosts, seven host bits are required:

- $2^6 - 2 = 62$  hosts (not enough)
- $2^7 - 2 = 126$  hosts

To determine host bits in the mask, determine the required number of bits for the supporting hosts on a subnet. Calculate the number of host bits required by using the formula  $2^n - 2$ , where  $n$  is the number of bits. This result must be at least the number of hosts that you need for your network, and the maximum number of hosts that you can configure on that subnet.

On each subnet, two host IDs are allocated automatically and cannot be used by computers.

An address with the host ID of all 0s represents the network. An address with the host ID of all 1s is the broadcast address for that network.

The following table shows how many hosts a class C network has available based on the number of host bits.

Number of bits (n)	Number of hosts ( $2^n - 2$ )
1	0
2	2
3	6
4	14
5	30
6	62

You can calculate each subnet's range of host addresses by using the following process:

1. The first host is one binary digit higher than the current subnet ID.
2. The last host is two binary digits lower than the next subnet ID.


The following table shows examples of calculating host addresses.

Network	Host range
172.16.64.0/19	172.16.64.1 – 172.16.95.254
172.16.96.0/19	172.16.96.1 – 172.16.127.254
172.16.128.0/19	172.16.128.1 – 172.16.159.254

To create an appropriate addressing scheme for your organization, you must know how many subnets you need and how many hosts you need on each subnet. With that information, you can calculate an appropriate subnet mask.

## Discussion: Creating a Subnetting Scheme for a New Office

- How many subnets are required?
- How many bits are required to create that number of subnets?
- How many hosts are required on each subnet?
- How many bits are required to support that number of hosts?
- What is an appropriate subnet mask that would satisfy these requirements?



20 minutes

For this discussion, read the scenario and answer the questions on the slide.

## Scenario

You are designing an appropriate network configuration for a new campus. You have been allocated the 10.34.0.0/16 network that you can subnet as required, given these requirements:

- There are four buildings on the new campus, and each should have its own subnet to allow for routing between the buildings.
- Each building will have up to 700 users.
- Each building will have network printers that will require IP addresses.
- The typical ratio of users to printers is 50 to 1.
- You need to allocate a subnet for the server data center that will hold up to 100 servers.

## Discussion Questions

Based on this scenario, answer the following questions:

- ? **Question:** How many subnets are required?
- ? **Question:** How many bits are required to create that number of subnets?
- ? **Question:** How many hosts are required on each subnet?
- ? **Question:** How many bits are required to support that number of hosts?
- ? **Question:** What is an appropriate subnet mask that would satisfy these requirements?

## What Is Supernetting?

- Supernetting combines multiple small networks into a larger network
- The networks that you combine must be contiguous
- The following table shows an example of supernetting two class C networks

Network	Range
192.168.00010000.00000000/24	192.168.16.0 - 192.168.16.255
192.168.00010001.00000000/24	192.168.17.0 - 192.168.17.255
192.168.00010000.00000000/23	192.168.16.0 - 192.168.17.255

*Supernetting* combines multiple small networks into a single large network. This may be appropriate when you have a small network that has grown and you need to expand the address space. For example, if a branch office that is using the network 192.168.16.0/24 exhausts all of its IP addresses, you could allocate the additional network 192.168.17.0/24 to it. If you use the default subnet mask of 255.255.255.0 for these networks, then you must perform routing between them. You can use supernetting to combine them into a single network.

To perform supernetting, the networks that you are combining must be contiguous. For example, 192.168.16.0/24 and 192.168.17.0/24 can be supernetted, but you cannot supernet 192.168.16.0/24 and 192.168.54.0/24.

Supernetting is the opposite of subnetting. When you perform supernetting, you allocate bits from the network ID to the host ID. The following table shows how many networks you can combine by using a specific number of bits.

Number of bits	Number of networks combined
1	2
2	4
3	8
4	16

The following table shows an example of supernetting two class C networks. The portion of the subnet mask that you are using as part of the network ID is in bold type.

Network	Range
192.168. <b>00010000</b> .00000000/24	192.168.16.0-192.168.16.255
192.168. <b>00010001</b> .00000000/24	192.168.17.0-192.168.17.255
192.168. <b>00010000</b> .00000000/23	192.168.16.0-192.168.17.255

## Lesson 4: Configuring and Troubleshooting IPv4

An incorrect IPv4 configuration affects the availability of services that are running on a server. To ensure the availability of network services, you need to understand how to configure and troubleshoot IPv4. Windows Server 2012 introduces the ability to configure IPv4 by using Windows PowerShell.

The troubleshooting tools in Windows Server 2012 are similar to the troubleshooting tools in previous versions of Windows client operating systems and Windows Server operating systems. You may use tools, such as Microsoft® Message Analyzer, to perform detailed analysis of your network communication.

### Lesson Objectives

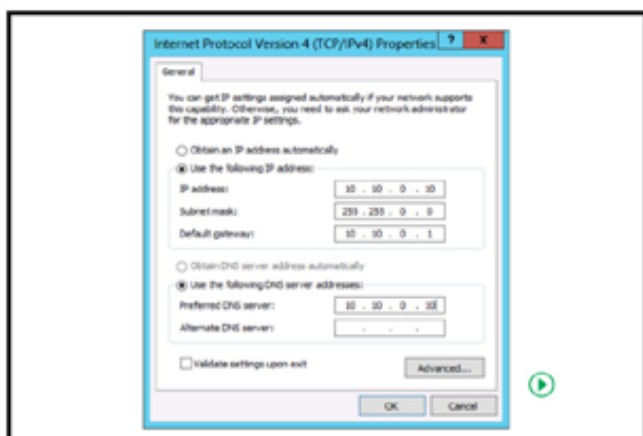
After completing this lesson, you should be able to:

- Configure IPv4 manually to provide a static configuration for a server.
- Configure a server so that it obtains an IPv4 configuration automatically.
- Explain how to use IPv4 troubleshooting tools.
- Explain how to use Windows PowerShell cmdlets for troubleshooting

## IPv4.

- Describe the troubleshooting process used to resolve fundamental IPv4 problems.
- Describe the function of Microsoft Message Analyzer.
- Use Microsoft Message Analyzer to capture and analyze network traffic.

## Configuring IPv4 Manually



You can configure IPv4 addresses manually or automatically. To configure an IPv4 address manually, enter the IPv4 address by using the Windows Server 2012 graphical interface or by using Windows PowerShell. An IPv4 address is configured automatically when a server that runs Dynamic Host Configuration Protocol – DHCP assigns an IPv4 address to the computers or network devices. Static IP addresses are usually configured on servers, routers, switches or other network devices that need to maintain persistent IP configuration that does not change over time.

To configure a static IP address for a server in an IPv4 configuration, you will need to determine the following settings:

- IPv4 address
- Subnet mask
- Default gateway

- DNS servers

Static configuration requires that you visit each computer and input the IPv4 configuration manually. This method of computer management is reasonable for servers, but it is very time consuming for client computers. Manually entering a static configuration also increases the risk of configuration mistakes.

## Configuring a Static IP Address by Using Windows PowerShell

Windows Server 2012 includes Windows PowerShell cmdlets that you can use to manage network configuration. The following table describes some of the Windows PowerShell cmdlets that are available for configuring IPv4.

Cmdlet	Description of IPv4 configuration uses
<b>New-NetIPAddress</b>	Use this command to create a new IP address and bind it to a network adapter. You cannot use this command to change an IP address.
<b>Set-NetIPAddress</b>	This command changes the configuration of an IP address.
<b>Set-NetIPInterface</b>	You can use this command to enable or disable DHCP for an interface.
<b>New-NetRoute</b>	This command creates routing table entries, including the default gateway (0.0.0.0). You cannot use this cmdlet to modify the next hop of an existing route; instead, you must remove an existing route and create a new route with the correct next hop.
<b>Set-DNSClientServerAddress</b>	Configures the DNS server that is used for an interface.

The following code is an example of the Windows PowerShell cmdlets that you can use to configure the interface Local Area Connection with the following parameters:

- Static IP address 10.10.0.10



- Subnet mask 255.255.255.0
- Default gateway 10.10.0.1

Local Area Connection is also configured to use DNS servers of 10.12.0.1 and 10.12.0.2.

```
New-NetIPAddress -InterfaceAlias "Local Area Connection" -  
IPAddress 10.10.0.10  
-PrefixLength 24 -DefaultGateway 10.10.0.1  
Set-DNSClientServerAddress -InterfaceAlias "Local Area Connection"  
-ServerAddresses  
10.12.0.1,10.12.0.2
```

## Configuring a Static IP Address by Using Netsh

You also can configure a static IP address either in the properties of the network connection or by using the netsh command-line tool. For example, the following command configures the interface Local Area Connection with the following parameters:

- Static IP address 10.10.0.10
- Subnet mask 255.255.255.0
- Default gateway 10.10.0.1

```
Netsh interface ipv4 set address name="Local Area Connection"  
source=static  
addr=10.10.0.10 mask=255.255.255.0 gateway=10.10.0.1
```

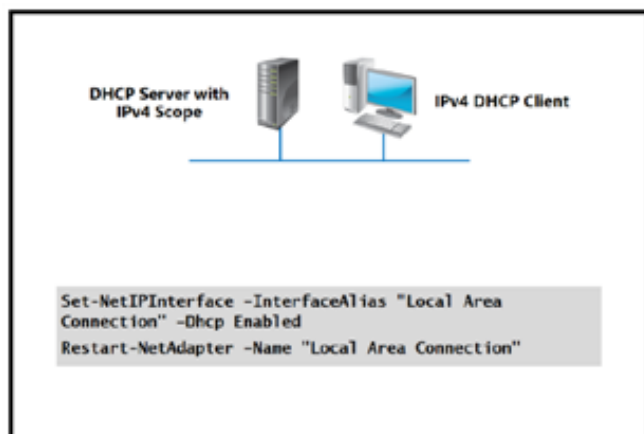


**Additional Reading:** For more information about net TCP/IP cmdlets in Windows PowerShell, go to <http://go.microsoft.com/fwlink/?LinkId=269708>.



**Question:** Do any computers or devices in your organization have static IP addresses?

## Configuring IPv4 Automatically



DHCP for IPv4 enables you to automate the process of assigning IPv4 addresses to large numbers of computers without having to assign each one individually. The DHCP service receives requests for IPv4 configuration from computers that you configure to obtain an IPv4 address automatically. It also assigns additional IPv4 settings from scopes that you define for each of your network's subnets. The DHCP service identifies the subnet from which the request originated and assigns IP configuration from the relevant scope.

DHCP helps simplify the IP configuration process; however, you must be aware that if you use DHCP to assign IPv4 information and the service is business-critical, you must do the following:

- Include resilience in your DHCP service design so that the failure of a single server does not prevent the service from functioning.
- Configure the scopes on the DHCP server carefully. If you make a mistake, it can affect the entire network and prevent communication.

When you use a laptop to connect to multiple networks, such as one at work and one at home, you should configure the IP addressing differently on each network. However, if a DHCP server exists on both networks, the DHCP server will configure the laptop IP settings automatically.

Windows operating systems also support the use of these technologies for assigning IP addresses:

- **Automatic Private IP Addressing (APIPA).** In a scenario when there is no DHCP server on the network or the DHCP server is not available, Windows uses APIPA to automatically assign itself an IP address in the address range between 169.254.0.0 and 169.254.255.255. Because APIPA does not configure the computer with DNS and default gateway settings, computers with assigned APIPA addresses have limited networking functionality. APIPA can also be used for troubleshooting DHCP. If the network administrator notices that the computer has an address from the APIPA range, it is an indication that the computer cannot communicate with the DHCP server.
- **Alternate static IP address.** If the alternate static IP address is configured on a computer network adapter and the DHCP server is not available, the computer network adapter will use the alternate static IP address.

Windows Server 2012 also has Windows PowerShell cmdlets that you can use to enable DHCP for an interface. The following table describes some of the available Windows PowerShell cmdlets that are available for configuring DHCP on an interface.

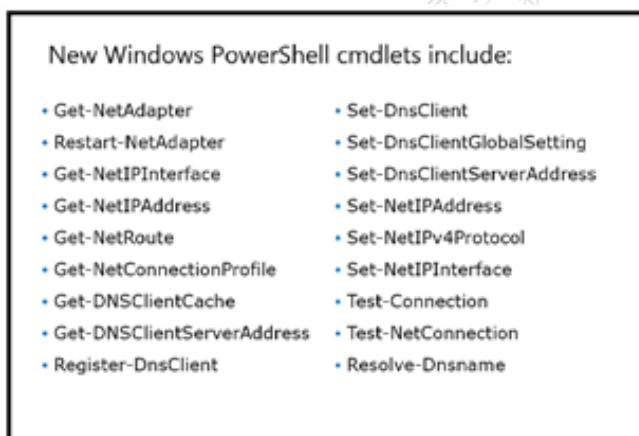
Cmdlet	Description
<b>Get-NetIPInterface</b>	Obtains a list of interfaces and their configuration. This does not include IPv4 configuration of the interface.

<b>Set-NetIPInterface</b>	Enables or disables DHCP for an interface.
<b>Get-NetAdapter</b>	Obtains a list of network adapters in a computer.
<b>Restart-NetAdapter</b>	Disables and re-enables a network adapter. This forces a DHCP client to obtain a new DHCP lease.

The following code is an example of how you can enable DHCP for the adapter Local Area Connection, and ensure that it receives an address:

```
Set-NetIPInterface -InterfaceAlias "Local Area Connection" -Dhcp
Enabled
Restart-NetAdapter -Name "Local Area Connection"
```

## Using Windows PowerShell Cmdlets to Troubleshoot IPv4



You can use command-line tools or Windows PowerShell cmdlets in Windows Server 2012 to configure and troubleshoot your network.

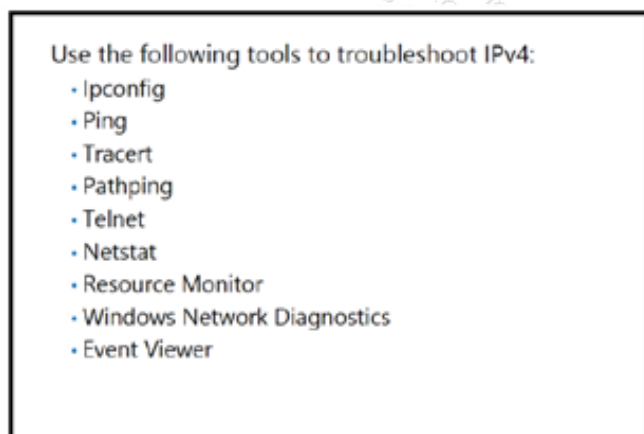
Although you could use Windows PowerShell in earlier versions of Windows Server to perform network troubleshooting and configuration, it required you to use Windows Management Instrumentation (WMI) objects, which are more difficult to use than native Windows PowerShell cmdlets.

The following table lists some of the Windows PowerShell cmdlets that you can use.

Cmdlet	Purpose
<b>Get-NetAdapter</b>	Obtains a list of network adapters in a computer.
<b>Get-NetIPv4Protocol</b>	Gets information about the IPv4 protocol configuration. Note that the <b>Get-NetIPv6Protocol</b> gets information about the IPv6 protocol configuration.
<b>Restart-NetAdapter</b>	Disables and re-enables a network adapter.
<b>Get-NetIPInterface</b>	Obtains a list of interfaces and their configuration.
<b>Get-NetIPAddress</b>	Obtains a list of IP addresses that are configured for interfaces.
<b>Get-NetRoute</b>	Obtains the list of routes in the local routing table.
<b>Get-NetConnectionProfile</b>	Obtains the type of network (public, private, domain) to which a network adapter is connected.
<b>Get-DnsClient</b>	Retrieves configuration details specific to the different network interfaces on a specified computer.
<b>Get-DNSClientCache</b>	Obtains the list of resolved DNS names that are stored in the DNS client cache.
<b>Get-DnsClientGlobalSetting</b>	Retrieves global DNS client settings such as the suffix search list.
<b>Get-DNSClientServerAddress</b>	Obtains the list of DNS servers that are used for each interface.
<b>Register-DnsClient</b>	Registers all of the IP addresses on the computer on the configured DNS server.
<b>Set-DnsClient</b>	Sets the interface-specific DNS client configurations on the computer.
<b>Set-DnsClientGlobalSetting</b>	Configures the global DNS client settings such as the suffix search list.
<b>Set-DnsClientServerAddress</b>	Configures the computer's network adapter with the IP addresses of the DNS server.
<b>Set-NetIPAddress</b>	Sets information about the IP address configuration.
<b>Set-NetIPv4Protocol</b>	Sets information about the IPv4 protocol configuration. Note that the Set-NetIPv6Protocol returns information about the IPv6 protocol configuration.
<b>Set-NetIPInterface</b>	Modifies the IP interface properties.

<b>Test-Connection</b>	Runs connectivity tests that are similar to those used by ping.
<b>Test-NetConnection</b>	Displays the following: <ul style="list-style-type: none"> <li>• Results of a DNS lookup</li> <li>• Listing of IP interfaces</li> <li>• Option to test a TCP connection</li> <li>• IPsec rules</li> <li>• Confirmation of connection establishment</li> </ul>
<b>Resolve-Dnsname</b>	Performs a DNS name query resolution for the specified name.

## IPv4 Troubleshooting Tools



Windows Server 2012 includes a number of command-line tools that can help you diagnose network problems. These tools were commonly used in earlier Windows Server editions.

### Ipconfig

Ipconfig is a command-line tool that displays the current TCP/IP network configuration.

Additionally, you can use the **ipconfig** command to refresh DHCP and DNS settings. The following table describes the command-line options for **ipconfig**.



Command	Description
<b>ipconfig /all</b>	View detailed configuration information.
<b>ipconfig /release</b>	Release the leased configuration back to the DHCP server.
<b>ipconfig /renew</b>	Renew the leased configuration.
<b>ipconfig /displaydns</b>	View the DNS resolver cache entries.
<b>ipconfig /flushdns</b>	Purge the DNS resolve cache.

## Ping

Ping is a command-line tool that verifies IP-level connectivity to another TCP/IP computer. It sends ICMP echo request messages and displays the receipt of corresponding echo reply messages. Ping is the primary TCP/IP command that you use to troubleshoot connectivity, but firewalls might block the ICMP messages.

## Tracert

Tracert is a command-line tool that identifies the path taken to a destination computer by sending a series of ICMP echo requests. Tracert then displays the list of router interfaces between a source and a destination. This tool also determines which router has failed, and what the latency, or speed, is. These results might not be accurate if the router is busy, because the ICMP packets are assigned a low priority by the router.

## Pathping

Pathping is a command-line tool that traces a route through the network in a manner similar to Tracert. However, Pathping provides more detailed statistics on the individual steps, or *hops*, through the network. Pathping can provide greater detail, because it sends 100 packets for each router, which enables it to establish trends.

## Route



Route is a command-line tool that allows you to view and modify the local routing table. You can use this to verify the default gateway, which is listed as the route 0.0.0.0. In Windows Server 2012, you can also use Windows PowerShell cmdlets to view and modify the routing table. The cmdlets for viewing and modifying the local routing table include **Get-NetRoute**, **New-NetRoute**, and **Remove-NetRoute**.

## Telnet

You can use the Telnet Client feature to verify whether a server port is listening. For example, the command **telnet 10.10.0.10 25** attempts to open a connection with the destination server, 10.10.0.10, on port 25, SMTP. If the port is active and listening, it returns a message to the Telnet client.

## Netstat

Netstat is a command-line tool that enables you to view network connections and statistics. For example, the command **netstat -ab** returns all listening ports and the executable that is listening.

## Resource Monitor

Resource Monitor is a graphical tool that allows you to monitor system resource utilization. You can use Resource Monitor to view TCP and UDP ports that are in use. You can also verify which programs are using specific ports and the amount of data that they are transferring on those ports.

## Network Diagnostics

Use Windows Network Diagnostics to diagnose and correct networking problems. In the event of a Windows Server networking problem, the **Diagnose Connection Problems** option helps you diagnose and repair the problem. Windows Network Diagnostics returns a possible description of the problem and a potential remedy. However, the solution might require manual intervention from the user.

## Event Viewer

*Event logs* are files that record significant events on a computer, such as when a process encounters an error. When these events occur, the Windows Server 2012 operating system records the event in an appropriate event log. You can use Event Viewer to read the event log. IP conflicts, which might prevent services from starting, are listed in the System event log.

## The IPv4 Troubleshooting Process

After you identify the scope of the problem, use the following tools to troubleshoot network connectivity:

Step	Windows PowerShell	Command-line tool
Verify the network configuration is correct	Get-NetIPAddress	ipconfig
Identify the network path between hosts	Test-NetConnection -TraceRoute	tracert
See if the remote host responds	Test-NetConnection	ping
Test the service on a remote host	Test-NetConnection -Port	Telnet
See if the default gateway responds	Test-NetConnection	ping

The first step in troubleshooting a network problem is identifying the scope of the problem.

The causes of a problem that affects a single user probably differs from a problem that affects all users. If a problem affects only a single user, then the problem is likely related to the configuration of that one computer. If a problem affects all users, then the problem is likely either a server configuration issue or a network configuration issue. If a problem affects only a group of users, then you need to determine the common denominator among that group of users.

To troubleshoot network communication problems, you need to understand the overall communication process. This requires that you understand the routing and firewall configuration on your network.

The Windows Server 2012 R2 operating system introduced two new

Windows PowerShell cmdlets that you can use to help you troubleshoot network connectivity: **Get-NetIPAddress** and **Test-NetConnection**. You can run **Get-NetIPAddress** at a Windows PowerShell prompt by typing **Get-NetIPAddress** or **gnp**. Similarly, type **Test-NetConnection** or **tnc** at a Windows PowerShell prompt to run the **Test-NetConnection** cmdlet.

The following are some of the actions that you can use to identify the cause of network communication problems:

- If you know what the correct network configuration for the host should be, use one of the following to verify that it is configured correctly:

- o Windows PowerShell: **Get-NetIPAddress**

- o Command-line: **ipconfig**

If the command returns an address on the 169.254.0.0/16 network, it indicates that the host failed to obtain an IP address from DHCP.

- To help identify the routing path through your network, you can use the Windows PowerShell cmdlet **Test-NetConnection -TraceRoute**, or you can use the command-line tool **tracert**.

- To see if the remote host responds, use one of the following:

- o Windows PowerShell: **Test-NetConnection**

- o Command-line: **ping**

When you use either method to return the DNS name of the remote host, you verify both name resolution and whether the host responds. Be aware that Windows Firewall on member servers and client computers often blocks ping attempts. When this happens, the lack of a ping response might not be an indicator that the remote host is not functional, but only that the ping is being blocked. If you can ping other remote hosts on the same network, this might mean that the problem is on the remote host.

- You can use the **Test-NetConnection** cmdlet in Windows PowerShell to

test the service you are connecting to on the remote host. For example, use **Test-NetConnection -Port 80** to test connectivity to a web server. You can also use **Telnet** to connect to the port of the remote program.

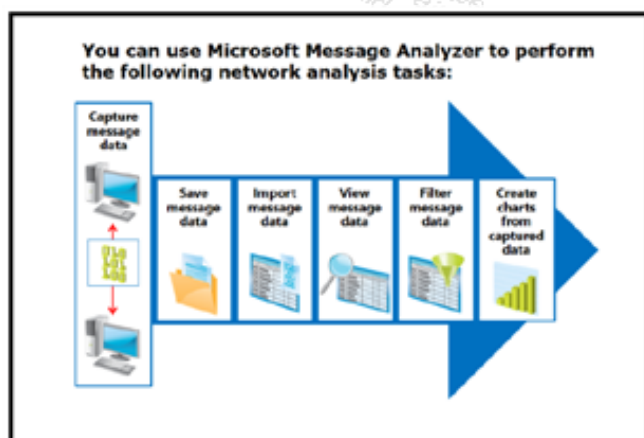
- To see if the default gateway responds, use one of the following:
  - o Windows PowerShell: **Test-NetConnection**
  - o Command-line: **ping**

Most routers respond to Test-NetConnection and ping requests. If you do not get a response when you ping the default gateway, then there is likely a configuration error on the client computer, such as an incorrect configuration of the default gateway. It is also possible that the router is experiencing errors.

**Note:** You can force **ping** to use IPv4 instead of IPv6 by using the **-4** option.

**Question:** What additional steps might you use to troubleshoot network connectivity problems?

## What Is Microsoft Message Analyzer?



*Microsoft Message Analyzer* is a tool used to capture network traffic and

then display and analyze information about that traffic. You can use Microsoft Message Analyzer to monitor live network traffic, or to import, aggregate, and analyze data from log and trace files.

You can use Microsoft Message Analyzer to perform the following network analysis tasks:


- Capture message data
- Save message data
- Import message data
- View message data
- Filter message data

Microsoft Message Analyzer uses several built-in Trace Scenarios that you can access through the Microsoft Message Analyzer console. Trace Scenarios contain specific capture settings that enable you to quickly start a trace session and then capture the information you need for your troubleshooting task. These Trace Scenarios include predefined capture configuration for Windows Firewall troubleshooting, LAN and WAN monitoring, and Web Proxy troubleshooting. You can customize Trace Scenarios to remove items that do not require monitoring.

The Microsoft Message Analyzer console contains a Charts tab that creates charts from captured data. You can customize the parameters and data that will be included in the charts, including network transactions, operations, and the network protocol. Furthermore, you can define different types of chart views, such as Timeline Chart, Pie Chart, Grid View, or Bar Chart. Charts can help you understand incoming trace data by presenting complicated traffic information visually. Often, this feature is helpful when you need to perform mathematical calculations on the trace data, such as the number of retries required for a packet being sent between hosts.

Microsoft Message Analyzer introduces remote live monitoring, which is a feature that allows administrators to monitor the network from a remote host. Administrators can connect to both remote host network adapters and virtual machine network adapters in order to capture and analyze the network traffic data.

Microsoft Message Analyzer is capable of loading data from native Microsoft Message Analyzer files, event tracing log (.etl) files, Network Monitor capture files (.cap), comma-separated values (.csv) files, and several other formats. You can download Microsoft Message Analyzer for free from the Microsoft website.

 **Reference Links:** For more information about Microsoft Message Analyzer, see the Microsoft Message Analyzer Operating Guide at <http://go.microsoft.com/fwlink/?LinkID=331073>. To download Microsoft Message Analyzer, go to <http://go.microsoft.com/fwlink/?LinkID=331072>.

## Demonstration: How to Capture and Analyze Network Traffic by Using Microsoft Message Analyzer

You can use Microsoft Message Analyzer to capture and view packets that are transmitted on a network. This allows you to view detailed information that you would not normally be able to see. This type of information can be useful for troubleshooting.

In this demonstration, you will see how to:

- Capture network traffic with Microsoft Message Analyzer.
- Analyze captured network traffic.
- Filter network traffic.

## Demonstration Steps Start a new Capture/Trace in Microsoft



## Message Analyzer

1. Sign in to LON-SVR2 as **Adatum\Administrator** with a password of **Pa\$\$w0rd**.
2. Open a Windows PowerShell prompt and run the following command:

```
ipconfig /flushdns
```

3. From the Start screen, open **Microsoft Message Analyzer**, choose **Do not update items**, and then start a new **Capture/Trace** for using the **Firewall** trace scenario.

## Capture packets from a ping request

1. In Microsoft Message Analyzer, start a packet capture.
2. At the Windows PowerShell prompt, run following cmdlet:

```
Test-NetConnection LON-DC1.adatum.com
```

3. In Microsoft Message Analyzer, stop the packet capture.

## Analyze the captured network traffic

1. In Microsoft Message Analyzer, in the results pane, under the **Module** column, select the first **ICMP** packet group.
2. Expand the **ICMP** portion of the packet to view that it includes both **Echo Request and Echo Reply packets**. This is a **ping** request that was executed when running **Test-NetConnection** cmdlet.
3. View the source and destination IP addresses for each packet.

## Filter the network traffic

1. In Microsoft Message Analyzer, enter the following filter criteria, and then apply the filter:  
  
\*DestinationAddress == 172.16.0.10
2. Verify that only packets that match the filter are displayed.
3. Close Microsoft Message Analyzer.

## Lab: Implementing IPv4

### Scenario

You have recently accepted a promotion to the server support team. One of your first assignments is configuring the infrastructure service for a new branch office.

After a security review, your manager has asked you to calculate new subnets for the branch office to support segmenting network traffic. You also need to troubleshoot a connectivity problem on a server in the branch office.

### Objectives

After completing this lab, you should be able to:

- Identify appropriate subnets for a given set of requirements.
- Troubleshoot IPv4 connectivity issues.

### Lab Setup

Estimated Time: 45 minutes

Virtual machines	<b>20410D-LON-DC1</b> <b>20410D-LON-RTR</b> <b>20410D-LON-SVR2</b>
User name	<b>Adatum\Administrator</b>
Password	<b>Pa\$\$w0rd</b>

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. On the host computer, start **Hyper-V Manager**.
2. In Microsoft Hyper-V® Manager, click **20410D-LON-DC1**, and then, in the Actions pane, click **Start**.
3. In the Actions pane, click **Connect**. Wait until the virtual machine starts.
4. Sign in using the following credentials:
  - o User name: **Adatum\Administrator**
  - o Password: **Pa\$\$w0rd**
5. Repeat steps 2 through 4 for **20410D-LON-RTR** and **20410D-LON-SVR2**.

## Exercise 1: Identifying Appropriate Subnets

### Scenario

The new branch office is configured with a single subnet. After a security review, all branch office network configurations are being modified to place servers on a separate subnet from the client computers. You need to

calculate the new subnet mask and the default gateways for the subnets in your branch.

The current network for your branch office is 192.168.98.0/24. This network needs to be subdivided into three subnets that meet the following requirements:

- One subnet with at least 100 IP addresses for clients.
- One subnet with at least 10 IP addresses for servers.
- One subnet with at least 40 IP addresses for future expansion.

The main tasks for this exercise are as follows:

1. Calculate the bits required to support the hosts on each subnet.
2. Calculate subnet masks and network IDs.

### **Task 1: Calculate the bits required to support the hosts on each subnet**

1. How many bits are required to support 100 hosts on the client subnet?
2. How many bits are required to support 10 hosts on the server subnet?
3. How many bits are required to support 40 hosts on the future expansion subnet?
4. If all subnets are the same size, can they be accommodated?
5. Which feature allows a single network to be divided into subnets of varying sizes?
6. How many host bits will you use for each subnet? Use the simplest allocation possible, which is one large subnet and two equal-sized

smaller subnets.

## Task 2: Calculate subnet masks and network IDs

1. Given the number of host bits allocated, what is the subnet mask that you will use for the client subnet? Calculate the subnet mask in binary and decimal.
  - o The client subnet is using 7 bits for the host ID. Therefore, you can use 25 bits for the subnet mask.

Binary	Decimal

2. Given the number of host bits allocated, what is the subnet mask that you will use for the server subnet? Calculate the subnet mask in binary and decimal.
  - o The server subnet is using 6 bits for the host ID. Therefore, you will use 26 bits for the subnet mask.

Binary	Decimal

3. Given the number of host bits allocated, what is the subnet mask that you can use for the future expansion subnet? Calculate the subnet mask in binary and decimal.
  - o The future expansion subnet is using 6 bits for the host ID. Therefore, you will use 26 bits for the subnet mask.

Binary	Decimal

4. For the client subnet, define the network ID, first available host, last available host, and broadcast address. Assume that the client subnet is the first subnet allocated from the available address pool. Calculate the binary and decimal versions of each address.

Description	Binary	Decimal
Network ID		
First host		
Last host		
Broadcast		

5. For the server subnet, define the network ID, first available host, last available host, and broadcast address. Assume that the server subnet is the second subnet allocated from the available address pool. Calculate the binary and decimal versions of each address.

Description	Binary	Decimal
Network ID		
First host		
Last host		
Broadcast		

6. For the future allocation subnet, define the network ID, first available host, last available host, and broadcast address. Assume that the future allocation subnet is the third subnet allocated from the available address pool. Calculate the binary and decimal versions of each address.

Description	Binary	Decimal
Network ID		
First host		



Last host		
Broadcast		

**Results:** After completing this exercise, you should have identified a configuration of subnet that will meet the requirements of the lab scenario.

## Exercise 2: Troubleshooting IPv4

### Scenario

A server in the branch office is unable to communicate with the domain controller in the head office. You need to resolve the network connectivity problem.

The main tasks for this exercise are as follows:

1. Prepare for troubleshooting.
2. Troubleshoot IPv4 connectivity between LON-SVR2 and LON-DC1.

### Task 1: Prepare for troubleshooting

1. On LON-SVR2, open **Windows PowerShell**.
2. In the Windows PowerShell window, run the following cmdlet:

```
Test-NetConnection LON-DC1
```

3. Verify that you receive a reply that contains **PingSucceeded:True** from **LON-DC1**.

4. Open a File Explorer window, and browse to **\\LON-DC1\E\$\Labfiles\Mod05**.
  5. From File Explorer, run the **Break2.ps1** script by using Windows PowerShell.
- This script creates the problem that you will troubleshoot and repair in the next task.
6. Close File Explorer.

## Task 2: Troubleshoot IPv4 connectivity between LON-SVR2 and LON-DC1

1. Use your knowledge of IPv4 to troubleshoot and repair the connectivity problem between LON-SVR2 and LON-DC1. Consider using the following tools:
  - o **Test-NetConnection**
  - o **Test-NetConnection -TraceRoute**
  - o **Get-NetRoute**
  - o **New-NetRoute**
2. When you have repaired the problem, run the **Test-NetConnection LON-DC1** cmdlet from LON-SVR2 to confirm that the problem is resolved.

**Results:** After completing this lab, you should have resolved an IPv4 connectivity problem.

## Lab Review Questions

? **Question:** Why is variable-length subnetting required in this lab?

- ? **Question:** Which Windows PowerShell cmdlet can you use to view the local routing table of a computer instead of using **route print**?

## Prepare for the next module

After you finish the lab, revert the virtual machines back to their initial state by completing the following steps:

1. On the host computer, start **Hyper-V Manager**.
2. In Microsoft Hyper-V® Manager, in the **Virtual Machines** list, right-click **20410D-LON-DC1**, and then click **Revert**.
3. In the **Revert Virtual Machine** dialog box, click **Revert**.
4. Repeat steps 2 and 3 for **20410D-LON-RTR** and **20410D-LON-SVR2**.

## Module Review and Takeaways

### Review Questions

- ? **Question:** You have just started as a server administrator for a small organization with a single location. The organization is using the 131.107.88.0/24 address range for the internal network. Is this a concern?
- ? **Question:** You are working for an organization that provides web hosting services to other organizations. You have a single /24 network from your ISP for the web hosts. You are almost out of IPv4 addresses and have asked your ISP for an additional range of addresses. Ideally, you would like to supernet the existing network with the new network. Are there any specific requirements for supernetting?
- ? **Question:** You have installed a new web-based program that runs on

a non-standard port number. A colleague is testing access to the new web-based program, and indicates that he cannot connect to it. What are the most likely causes of his problem?

## Best Practices

When implementing IPv4, use the following best practices:

- Allow for growth when planning IPv4 subnets. This ensures that you do not need to change your IPv4 configuration scheme.
- Define purposes for specific address ranges and subnets. This enables you to both identify hosts based on their IP address easily and to use firewalls to increase security.
- Use dynamic IPv4 addresses for clients. It is much easier to manage the IPv4 configuration for client computers by using DHCP than with manual configuration.
- Use static IPv4 addresses for servers. When servers have a static IPv4 address, it is easier to identify where services are located on the network.

## Common Issues and Troubleshooting Tips

Common Issue	Troubleshooting Tip
IP conflicts	
Multiple default gateways defined	
Incorrect IPv4 configuration	

## Tools

Tool	Use for	Where to find it
Microsoft Message Analyzer	Capture and analyze network traffic.	Download from the Microsoft website
<b>Get-NetIPAddress</b>	Obtains a list of IP addresses that are configured for interfaces.	Windows PowerShell
<b>Test-NetConnection</b>	Displays the following: <ul style="list-style-type: none"> <li>Results of a DNS lookup</li> <li>Listing of IP interfaces</li> <li>Option to test a TCP connection</li> <li>Internet Protocol security (IPsec) rules</li> <li>Confirmation of connection establishment</li> </ul>	Windows PowerShell
<b>Ipconfig</b>	View network configuration.	Command prompt
<b>Ping</b>	Verify network connectivity.	Command prompt
<b>Tracert</b>	Verify network path between hosts.	Command prompt
<b>Pathping</b>	Verify network path and reliability between hosts.	Command prompt
<b>Route</b>	View and configure the local routing table.	Command prompt
<b>Telnet</b>	Test connectivity to a specific port.	Command prompt
<b>Netstat</b>	View network connectivity information.	Command prompt
Resource monitor	View network connectivity information.	Tools in Server Manager
Windows Network Diagnostics	Diagnose a problem with a network connection.	Properties of the network connection
Event Viewer	View network-related system events.	Tools in Server Manager