Module 5

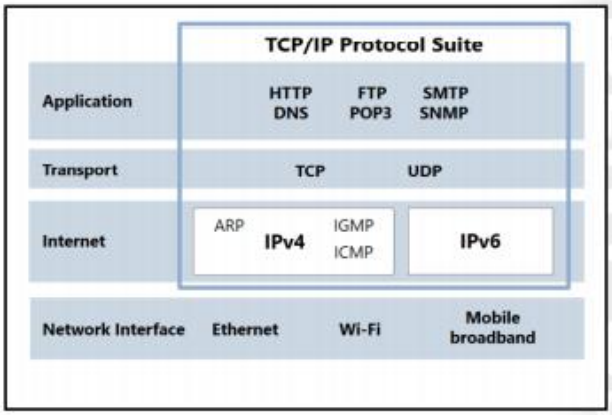
Implementing IPv4

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 and how it relates to other protocols to enable network communication. It also covers the concept of sockets, which applications use to accept network communications. Combined together, this lesson provides a foundation for understanding and troubleshooting network communication.

The TCP/IP Protocol Suite

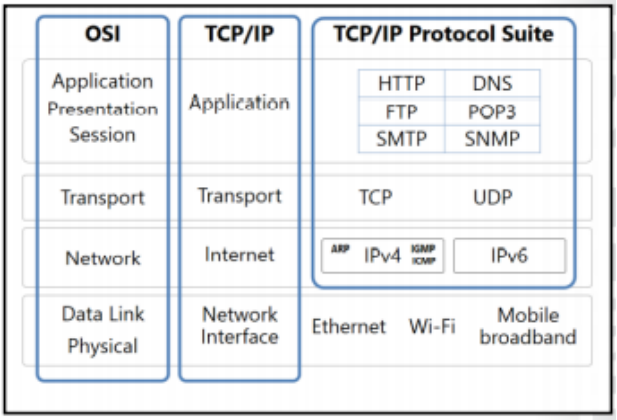
The tasks performed by TCP/IP in the communication process are distributed between protocols. These protocols are organized into four distinct layers within the TCP/IP stack:



- Application layer. Applications use the application layer protocols to access network resources.
- Transport layer. The transport layer protocols control data transfer reliability on the network.
- Internet layer. The internet layer protocols control packet movement between networks.
- Network interface layer. The network interface layer protocols define how datagrams from the Internet layer are transmitted on the media.

Protocols in the TCP/IP Suite

The Open Systems Interconnection (OSI) model defines distinct layers related to packaging, and 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 applications 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 applications. 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 applications. Web servers, File Transfer Protocol (FTP) clients, and other applications that move large amounts of data use TCP.
- UDP. Provides connectionless and unreliable communication. When using UDP, reliable delivery is the responsibility of the application. Applications use UDP for faster communication with less overhead than TCP. Applications such as streaming audio and video use UDP so that a single missing packet will not delay playback. UDP is also used by applications that send small amounts of data, such as Domain Name System (DNS) name lookups.

The transport layer protocol that an application uses is determined by the developer of an application, and is based on the communication requirements of the application.

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, including 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 applications over routers in IPv4 networks.
- ICMP. ICMP sends error messages in an IP-based network.

Network Interface Layer

The *network interface layer* (sometimes referred to as the *link layer* or *data link layer*) corresponds to the data link and physical layers of the OSI model. The network interface layer specifies the requirements for sending and receiving packets on the network media. This layer is often not formally 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

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

Some common application layer protocols:

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

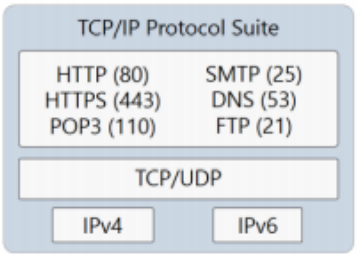
Protocol	Description
HTTP	Used for communication between web browsers and web servers.
HTTP/Secure (HTTPS)	A version of HTTP that encrypts communication between web browsers and web servers.
FTP	Used to transfer files between FTP clients and servers.
Remote Desktop Protocol (RDP)	Used to remotely control 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 Application Protocol (IMAP)	Used to retrieve messages from some email servers.

What Is a Socket?

When an application wants to establish communication with an application on a remote host, it creates a TCP or a UDP socket, as appropriate. A socket identifies the following as part of the communication process:

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

A socket is a combination of an IP address, a transport protocol, and a port



This combination of transport protocol, IP address, and port creates a socket.

Well-Known Ports

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

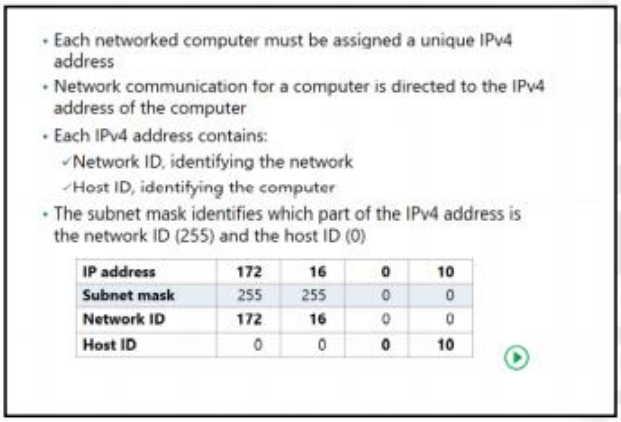
Port	Protocol	Application
80	TCP	HTTP used by a web server
443	TCP	HTTPS for a secure web server
110	TCP	POP3 used for email retrieval
143	TCP	IMAP used for email retrieval
25	TCP	SMTP used for sending email messages
53	UDP	DNS used for most name resolution requests
53	TCP	DNS used for zone transfers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

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 supposed to work.

IPv4 Addressing

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. The decimal numbers are separated by a period (dot). Each decimal number is called an *octet*, for example, the following IP address: 172.16.0.10.

Subnet Mask

Each IPv4 address is composed of a network 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.

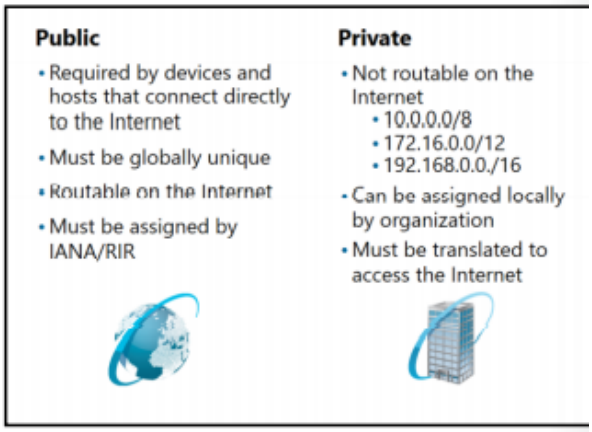
Default Gateway

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.

Public and Private IPv4 Addresses

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 (RIRs). RIRs 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 have to connect to the Internet.

Private IPv4 Addresses

The pool of IPv4 addresses is becoming smaller, so RIRs are reluctant to allocate superfluous IPv4 addresses. 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, 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

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.

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

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*—the rightmost bit in the octet—represents a decimal value of 1. The *high-order bit*—the leftmost bit in the octet—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). That is the highest possible value of an octet.

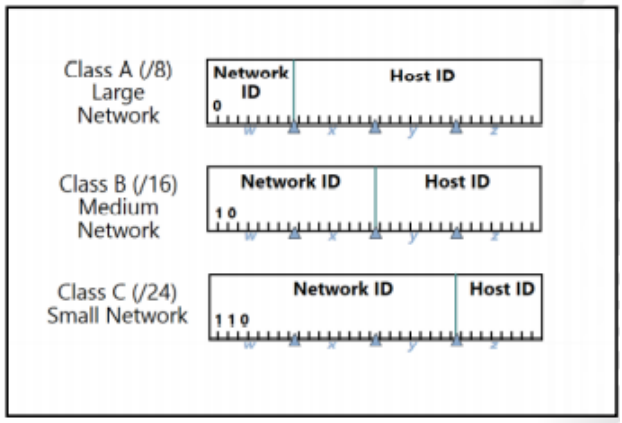
Most of the time, you can use a calculator to convert decimal numbers to binary and vice versa. The Calculator application included in Windows operating systems can perform decimal-to-binary conversions, as shown in the following example.

Binary	Dotted decimal notation
10000011 01101011 00000011 00011000	131.107.3.24

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. Class D addresses are used by computers and applications for multicasting. The IANA reserves Class E for experimental use. 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

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.

Lab Setup

Estimated Time: 45 minutes

Virtual machines	20410C-LON-DC1 20410C-LON-RTR 20410C-LON-SVR2
User name	Adatum\Administrator
Password:	Pa$$w0rd

Module 5

▶ Task 1: Prepare for troubleshooting

1. On LON-SVR2, on the taskbar, click the **Windows PowerShell** icon.
2. At the Windows PowerShell prompt, type the following cmdlet, and then press Enter:

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

3. Verify that you receive a reply that contains **PingSucceded:True** from **LON-DC1**.
4. Open a **File Explorer** window, and browse to **\\LON-DC1\E$\Labfiles\Mod05**.
5. Right-click **Break2.ps1**, and then click **Run with PowerShell**.

Note: 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. On LON-SVR2, at the Windows PowerShell prompt, type the following, and then press Enter:

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

2. Verify that you receive a reply that contains **PingSucceded:False** from **LON-DC1**.
3. At the Windows PowerShell Prompt, type the following, and then press Enter:

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

Notice that the host is unable to find the default gateway, and that the following warning message appears: "**Name resolution of lon-dc1 failed – Status: HostNotFound**."

4. At the Windows PowerShell Prompt, type the following, and then press Enter:

```
Get-NetRoute
```

Notice that the default route and the default gateway information is missing in the routing table.

Note: You should not be able to locate **DestinationPrefix 0.0.0.0/0** and **NextHop 10.10.0.1**.

5. At the Windows PowerShell Prompt, type the following, and then press Enter:

```
Test-NetConnection 10.10.0.1
```

6. Notice that the default gateway is responding by verifying that you receive a reply that contains **PingSucceded:True** from **10.10.0.1**.
7. At the Windows PowerShell Prompt, type the following, and then press Enter:

```
New-NetRoute –InterfaceAlias “Ethernet” –DestinationPrefix 0.0.0.0/0 –NextHop
10.10.0.1
```

Note: The **New-NetRoute** cmdlet will create the default route and the default gateway information that was missing.

8. At the Windows PowerShell Prompt, type the following, and then press Enter:

```
Get-NetRoute
```

9. Notice that the default route and the default gateway information is present in the routing table by locating **DestinationPrefix 0.0.0.0/0** and **NextHop 10.10.0.1**.
10. At the Windows PowerShell prompt, type the following, and then press Enter:

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

11. Verify that you receive a reply that contains **PingSucceded:True** from LON-DC1.

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

▶ Prepare for the next module

After you finish the lab, revert the virtual machines back to their initial state. To do this, complete the following steps.

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