Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

**Module**

**4**

**Protocols**

* [Module Introduction](javascript://)
* **4-1**[TCP/IP Core Protocols](javascript://)
  + **4-1a**[TCP (Transmission Control Protocol)](javascript://)
  + **4-1b**[UDP (User Datagram Protocol)](javascript://)
  + **4-1c**[IP (Internet Protocol)](javascript://)
  + **4-1d**[ICMP (Internet Control Message Protocol)](javascript://)
  + **4-1e**[ARP (Address Resolution Protocol) on IPv4 Networks](javascript://)
  + **4-1f**[NDP (Neighbor Discovery Protocol)](javascript://)
  + **4-1g**[Ethernet](javascript://)
* **4-2**[Encryption Protocols](javascript://)
  + **4-2a**[Key Encryption](javascript://)
  + **4-2b**[IPsec (Internet Protocol Security)](javascript://)
  + **4-2c**[SSL (Secure Sockets Layer) and TLS (Transport Layer Security)](javascript://)
* **4-3**[Remote Access Protocols](javascript://)
  + **4-3a**[Remote File Access](javascript://)
  + **4-3b**[Terminal Emulation](javascript://)
  + **4-3c**[VPNs (Virtual Private Networks)](javascript://)
  + **4-3d**[Remote Access Policies](javascript://)
* **4-4**[Troubleshooting Network Issues](javascript://)
  + **4-4a**[Troubleshooting Tools](javascript://)
  + **4-4b**[Solving Common Network Problems](javascript://)
* **4-5**[Module Review](javascript://)
  + **4-5a**[Module Summary](javascript://)
  + **4-5b**[Key Terms](javascript://)
  + **4-5c**[Review Questions](javascript://)
  + **4-5d**[Hands-On Projects](javascript://)
  + **4-5e**[Capstone Projects](javascript://)

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# Module Introduction

### Objectives

After reading this module and completing the exercises, you should be able to:

* **1**Describe the functions of core TCP/IP protocols
* **2**Identify how each protocol’s information is formatted in a TCP/IP message
* **3**Secure network connections using encryption protocols
* **4**Configure remote access connections between devices
* **5**Employ various TCP/IP utilities for network discovery and troubleshooting

**On the Job**

Intermittent errors (those that come and go) are among the most difficult to solve, so keeping careful logs of errors is often an essential troubleshooting technique. As an independent contractor for a large telecommunications company, I served on the third and final tier of a help desk that supported an application used by internal customers (company employees) over several wide area networks. The application functioned on more than 100 dedicated circuits that all terminated to feed a large database at corporate headquarters.

Transactions managed by the application were scanned for errors before they were posted to the database. Over time we were able to identify the source of most of these errors as bugs in the application. As we requested fixes from the application developer, we happily saw drastic reductions in the number of errors. However, a few intermittent errors proved to be most difficult to troubleshoot. After eliminating application bugs as the source of the problem, we began to suspect hardware. We carefully logged each error and searched for patterns of consistency: a particular circuit, client computer, branch office, type of transaction, currency, amount of transaction, time of day, and even day of the week. After weeks of logging and searching, we could not uncover a pattern and yet still intermittent errors persisted. Finally, it occurred to us to search for patterns of no errors. We went back through our logs and identified about 15 circuits that consistently yielded no errors since we had been keeping logs.

As we worked with the hardware teams, it came to light that these 15 or so circuits all had couplers installed and none of the other circuits used couplers. We all felt we had uncovered a significant clue, but still the problem wasn’t solved. My team decided to request a network analyzer to monitor problematic circuits. Before we had the analyzer in place, the application developer was finally able to reproduce the problem in the lab by using progressively faster circuits. The application required a buffer on the receiving end, which held incoming data before it was processed by the application. Faster circuits produced a buffer overflow, resulting in corrupted transactions. The mystery was solved. The couplers had managed to slightly reduce performance of the circuits, which allowed the application buffer to keep up with these slightly slower circuits. After weeks of troubleshooting, the solution was a simple programmer fix: Increase the application buffer size.

**Jean Andrews**

**Author and Independent Contractor**

In [Module 1](javascript://), you learned that a protocol is a rule that governs how computers on a network exchange data and instructions, and then in [Module 2](javascript://), you learned about network infrastructure equipment. In [Module 3](javascript://), you learned how the data link, network, transport, and application layer protocols navigate that infrastructure with various types of addresses as they determine where to send transmitted application data and instructions. You’ve also learned about the tasks associated with each layer of the OSI model, such as formatting, addressing, and error correction. All these tasks are governed by protocols.

This module focuses on how an application’s data and instructions make the trip from one host to another at the transport, network, and data link layers. To better understand these processes, you’ll learn how protocol messages are constructed at each of these layers. You’ll then explore other kinds of protocols that offer security through encryption or network access over remote connections. You’ll round out the module learning about some more troubleshooting tools and common network problems.

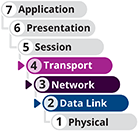
Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# 4-1TCP/IP Core Protocols



### Certification

* 1.1

Compare and contrast the Open Systems Interconnection (OSI) model layers and encapsulation concepts.

* 1.5

Explain common ports and protocols, their application, and encrypted alternatives.

* 2.3

Given a scenario, configure and deploy common Ethernet switching features.

* 5.3

Given a scenario, use the appropriate network software tools and commands.

* 5.5

Given a scenario, troubleshoot general networking issues.

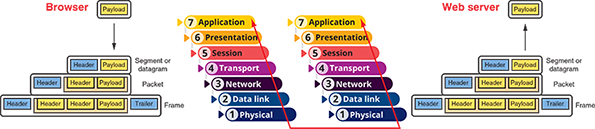
Average reading time: 54 minutes

TCP/IP is a suite of protocols, or standards, that includes TCP, IP (IPv4 and IPv6), UDP, ARP, and many others. In this part of the module, you’ll learn about message headers used at the transport layer. You’ll then work your way down the layers of the OSI model, examining each layer’s headers (and layer 2’s trailer) along the way. First, let’s summarize what you’ve learned so far about headers and trailers as illustrated in [Figure 4-1](javascript://) and described in the following list:

* **Layers 7, 6, and 5**—Data and instructions, known as the payload, are generated by an application running on the source host. For example, in [Figure 4-1](javascript://), the payload is created by the browser as data passes from the highest layer of the OSI model, down on through the next two highest layers.
* **Layer 4**—In the process of encapsulation, a transport layer protocol, usually either TCP or UDP, adds a header in front of the payload. This header includes a port to identify the receiving application on the destination host. The entire message then becomes a segment (when using TCP) or datagram (when using UDP).
* **Layer 3**—The network layer adds its own header in front of the passed-down segment or datagram. This header identifies the IP address of the destination host and the message is called a packet.
* **Layer 2**—The packet is passed to the data link layer on the NIC, which encapsulates this data with its own header and trailer, creating a frame. This layer’s frame includes a physical address used to find a node on the local network.
* **Layer 1**—The physical layer on the NIC receives the frame and places the actual transmission on the network.

**Figure 4-1**

Each layer adds its own data and addresses its transmission to the corresponding layer in the destination device



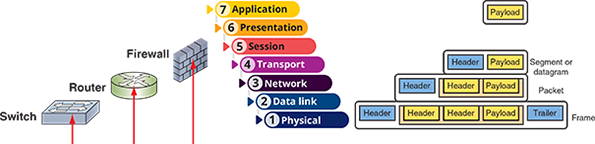
Enlarge Image

The receiving host decapsulates the message at each layer in reverse order and then presents the payload to the receiving application. As you saw in the [On the Job](javascript://) story at the beginning of this module, it’s important to understand how the various OSI layers work together when you’re troubleshooting a difficult-to-diagnose problem. What appears to be a problem at one layer might actually be caused by a process at a different layer.

In transit, the transmission might pass through any number of connectivity devices, such as switches and routers. Connectivity devices, also called networking devices, are specialized devices that allow two or more networks or multiple parts of one network to connect and exchange data. Each device is known by the innermost OSI layer header it reads and processes, as shown in [Figure 4-2](javascript://). For example, if a switch reads and processes the data link layer header but passes the message along without reading higher-layer headers, it is known as a layer 2 switch. In contrast, a router that reads and processes the network layer header and leaves alone the transport layer header is known as a layer 3 device. A layer 4 firewall will dig deep enough to read the transport layer header to check which port a message is directed to, and a layer 7 firewall might read through the entire message to check for signs of malware.

**Figure 4-2**

Connectivity devices are known by the highest OSI layer they read and process



Enlarge Image

With this quick review in hand, let’s examine the details of the core TCP/IP protocols, beginning with TCP.

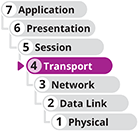
Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 4-1aTCP (Transmission Control Protocol)



Recall that TCP operates in the transport layer of the OSI model and provides reliable data delivery services. It’s helpful to compare TCP to making a phone call as you look at three characteristics of TCP in its role as a reliable delivery protocol:

* **Connection-oriented**—Before TCP transmits data, it ensures that a connection or session is established, similar to making sure someone is listening on the other end of a phone call before you start talking. TCP uses a three-step process called a [**three-way handshake**](javascript://) to establish a TCP connection. This process is described in detail later in this section. Only after TCP establishes this connection does it transmit the actual data, such as an HTTP request for a web page.
* **Sequencing and checksums**—In the analogy of a phone call, you might ask the other person if they can hear you clearly, and repeat a sentence as necessary. In the same vein, TCP sends a character string called a [**checksum**](javascript://); TCP on the destination host then generates a similar string. If the two checksums fail to match, the destination host asks the source to retransmit the data. In addition, because messages don’t always arrive in the same order they were created, TCP attaches a chronological sequence number to each segment so the destination host can, if necessary, reorder segments as they arrive.
* **Flow control**—You might slow down your talking over the phone if the other person needs a slower pace to hear every word and understand your message. Similarly, flow control is the process of gauging the appropriate rate of transmission based on how quickly the recipient can accept data. For example, suppose a receiver indicates its buffer can handle up to 4000 bytes. The sender will issue up to 4000 bytes in one or many small packets and then pause, waiting for an acknowledgment before sending more data.

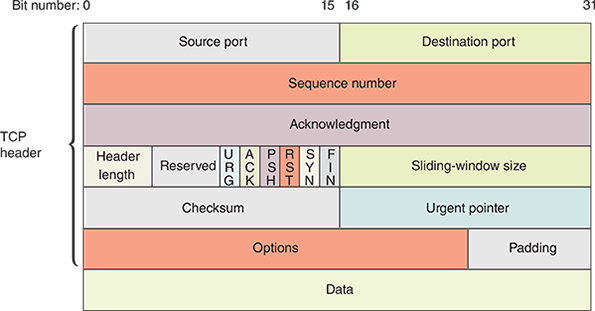
TCP manages all these elements—the three-way handshake, checksums, sequencing, and flow control—by posting data to fields in the TCP header at the beginning of a TCP segment.

### Fields in a TCP Segment

[Figure 4-3](javascript://) lists the items, called fields, included in a TCP segment. Each block shown in the figure represents a series of bits with each row representing 32 bits. If you were to string the rows alongside each other, in order from top to bottom, you would create one, long series of bits. This is a TCP segment. All the fields except the last one, the data field, are part of the TCP header. The content of the data field is the entire message passed down from the layer above the transport layer.

**Figure 4-3**

A TCP segment



Enlarge Image

**Note 4-1**

Headers are constructed in groups of 32 bits called words. Each word consists of 4 bytes, called blocks, of 8 bits each. This explains why diagrams of headers, such as the one in [Figure 4-3](javascript://), are depicted in 32-bit groups.

The fields shown in [Figure 4-3](javascript://) are defined in [Table 4-1](javascript://). Remember, the data field in the bottom row is not part of the TCP header. When the TCP segment is sent down to the network layer (layer 3), the entire segment becomes the data portion of an IP message. This payload is then encapsulated in an IP packet.

**Table 4-1**

### Fields in a TCP segment

| **Field** | **Length** | **Function** |
| --- | --- | --- |
| Source port | 16 bits | Indicates the port at the source host. Recall that a port is the number that identifies a process on a host. The port allows a process to be available for incoming or outgoing data. |
| Destination port | 16 bits | Indicates the port at the destination host. |
| Sequence number | 32 bits | Identifies the data segment’s position in the stream of data segments being sent. |
| Acknowledgment number | 32 bits | Confirms receipt of data via a return message to the sender. |
| TCP header length | 4 bits | Indicates the length of the TCP header in bytes. The header can be a minimum of 20 bytes to a maximum of 60 bytes in 4-byte increments. It’s also called the Data offset field because it indicates the offset from the beginning of the segment until the start of the data carried by the segment. |
| Reserved | 6 bits | Indicates a field reserved for later use. |
| Flags | 6 bits | Identifies a collection of six 1-bit fields or flags that signal special conditions about other fields in the header. The following flags are available to the sender:   * **URG**—If set to 1, the Urgent pointer field later in the segment contains information for the receiver. If set to 0, the receiver will ignore the Urgent pointer field. * **ACK**—If set to 1, the Acknowledgment field earlier in the segment contains information for the receiver. If set to 0, the receiver will ignore the Acknowledgment field. * **PSH**—If set to 1, data should be sent to an application without buffering. * **RST**—If set to 1, the sender is requesting that the connection be reset. * **SYN**—If set to 1, the sender is requesting a synchronization of the sequence numbers between the two nodes. This code indicates that no payload is included in the segment, and the acknowledgment number should be increased by 1 in response. * **FIN**—If set to 1, the segment is the last in a sequence and the connection should be closed. |
| Sliding-window size (or window) | 16 bits | Indicates how many bytes the sender can issue to a receiver before acknowledgment is received. This field performs flow control, preventing the receiver’s buffer from being deluged with bytes. |
| Checksum | 16 bits | Allows the receiving node to determine whether the TCP segment became corrupted during transmission. |
| Urgent pointer | 16 bits | Indicates a location in the data field where urgent data resides. |
| Options | 0–32 bits | Specifies special options, such as the maximum segment size a network can handle. |
| Padding | Variable | Contains filler bits to ensure that the size of the TCP header is a multiple of 32 bits. |
| Data | Variable | Contains data sent by the source host. The data field is not part of the TCP header—it is encapsulated by the TCP header. The size of the data field depends on how much data needs to be transmitted, the constraints on the TCP segment size imposed by the network type, and the limitation that the segment must fit within an IP packet at the next layer. |

Enlarge Table

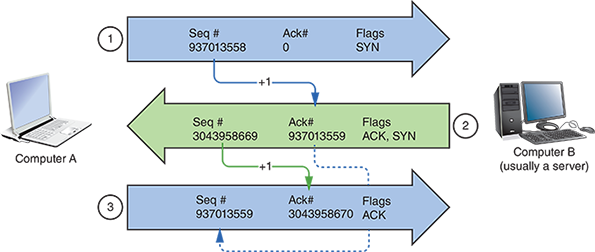
Now let’s see how the fields in the TCP header are used to perform a three-way handshake to establish a TCP session.

### TCP Three-Way Handshake

The TCP three-way handshake establishes a session before TCP transmits the actual data, such as an HTTP request for a web page. Think about how a handshake works when meeting a new acquaintance. You reach out your hand, not knowing how the other person will respond. If the person offers their hand in return, the two of you grasp hands, and you can then proceed with the conversation. [Figure 4-4](javascript://) shows the three transmissions in a TCP handshake, which are summarized in the following list:

**Figure 4-4**

The three-way handshake process establishes a TCP session



Enlarge Image

1. Step 1,

SYN (request for a connection)—Computer A issues a message to computer B with the following information:

* + In its Sequence number field, computer A selects and sends a seemingly random number that will be used to synchronize communication. In [Figure 4-4](javascript://), for example, this number is 937013558.
  + Its SYN bit is set to 1, which means the SYN flag is activated. This indicates the desire to communicate and synchronize sequence numbers. It’s as if computer A is offering a hand to computer B to see if there will be a response.
  + The ACK bit is usually set to 0 for this first transmission because there is no information yet from computer B to acknowledge.

1. Step 2,

SYN/ACK (response to the request)—When computer B receives this message, it responds with a segment containing the following information:

* + The ACK and SYN bits are both set to 1, essentially saying, “Yes, I’m here and I’m listening.”
  + The Acknowledgment number field contains a number that equals the sequence number computer A originally sent, plus 1. As [Figure 4-4](javascript://) illustrates, computer B sends the number 937013559. In this manner, computer B signals to computer A that it has received the request for communication and further, it expects computer A to respond again with the sequence number 937013559.
  + In its Sequence number field, computer B sends its own seemingly random number (in [Figure 4-4](javascript://), this number is 3043958669).

1. Step 3,

ACK (connection established)—Computer A issues a segment with the following information:

* + The sequence number is 937013559 because this is what computer B indicated it expects to receive.
  + The Acknowledgment number field equals the sequence number that computer B sent, plus 1. In the example, this number is 3043958670.
  + The ACK bit is set to 1.

The connection has now been established, and in the next message, computer A will begin data transmission.

**Note 4-2**

The ISN (Initial Sequence Number) of the first SYN message in the three-way handshake appears to be random, but in reality, it is calculated by a specific, clock-based algorithm, which varies by operating system. The existence of these algorithms and their predictability is actually a security loophole that hackers can use to undermine a host’s availability for connections.

Up until this point, no payload has been included in any of the three initial messages, and the sequence numbers have increased by exactly 1 in each acknowledgment. After these three transmissions, the payload or data is sent. This can be done in a single message for a small amount of data, such as a web page request, or fragmented over several messages, such as the data for the web page itself.

At this point, the sequence numbers will each be increased by the number of bits included in each received segment as confirmation that the correct length of message was received. In the example shown in [Figure 4-4](javascript://), computer A will send the next message, which will include the payload (such as an HTTP request) from a higher OSI layer. Suppose that computer A’s web page request message, the fourth message in this session, is 725 bits long. Computer B will receive this message, count the bits, and add 725 to the sequence number (937013559) of the received message. This new number, 937014284, becomes the acknowledgment number for the return message (which would be the fifth message in the session).

The two hosts continue communicating in this manner until computer A issues a segment whose FIN bit is set to 1, indicating the end of the transmission.

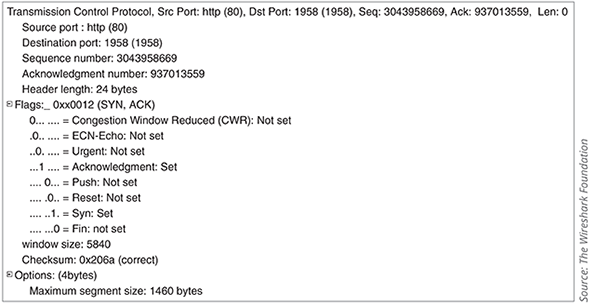
**Applying Concepts 4-1**

### Examine a Sample TCP Header

In Project 2-4, you captured and filtered to a TCP stream using Wireshark. Now that you know the function of each TCP segment field, you can interpret a segment’s header contents. Let’s practice with an example. [Figure 4-5](javascript://) shows a sample TCP header.

**Figure 4-5**

TCP segment header



Enlarge Image

Source: The Wireshark Foundation

Suppose the segment in [Figure 4-5](javascript://) was sent from computer B to computer A. [Table 4-2](javascript://) interprets the rows shown in [Figure 4-5](javascript://), beginning with the row labeled “Source port.”

**Table 4-2**

### Translation of TCP field data

| **Field name** | **TCP header data** |
| --- | --- |
| Source port | The segment was issued from computer B’s port 80, the port assigned to HTTP by default. |
| Destination port | The segment is addressed to port 1958 on computer A. |
| Sequence number | The segment is identified by sequence number 3043958669. |
| Acknowledgment number | By containing a value other than zero, this field informs computer A that its last communication was received. Computer B is indicating that the next segment it receives from computer A should have the sequence number of 937013559, which is the same as this segment’s acknowledgment number. |
| Header length | The TCP header is 24 bytes long—4 bytes larger than its minimum size, which means that some of the available options were specified or the padding space was used. |
| Flags: Congestion Window Reduced (CWR) and ECN-Echo | These optional flags can be used to help TCP react to and reduce traffic congestion. They are only available when TCP is establishing a connection. However, in this segment, they are not activated. |
| Flags: Acknowledgment and Syn | Of all the possible flags in the [Figure 4-5](javascript://) segment, only the ACK and SYN flags are set. This means that computer B is acknowledging the last segment it received from computer A, and it’s also negotiating a synchronization scheme for sequencing. |
| Window size | The window size is 5840, meaning that computer B can accept 5840 bytes of data from computer A before computer A should expect an acknowledgment. |
| Checksum | The valid outcome of the error-checking algorithm used to verify the segment’s header is 0x206a. When computer A receives this segment, it will perform the same calculation, and if the result matches, it will know the TCP header arrived without damage. |
| Maximum segment size | The maximum TCP segment size for this session is 1460 bytes. |

Enlarge Table

**Note 4-3**

A computer doesn’t “see” the TCP segment as it’s organized and formatted in [Figure 4-5](javascript://). The information in [Figure 4-5](javascript://) was generated by a [**protocol analyzer**](javascript://) (in this case, Wireshark), which is an application that collects and examines network messages. Wireshark translates each message into a user-friendly format. From the computer’s standpoint, the TCP segment arrives as a series of bits: 0s and 1s. The computer relies on TCP standards to determine how to interpret each bit in the segment based on its location and value. You’ll use the Wireshark protocol analyzer again in a later module.

TCP is not the only core protocol at the transport layer. A similar but less complex protocol, UDP, is discussed next.

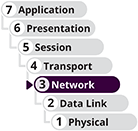
Go to pg.

[**help**](javascript://)

Application OpenedApplication Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 4-1dICMP (Internet Control Message Protocol)

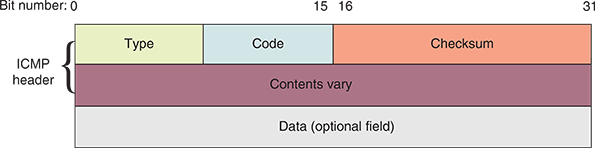


Whereas IP helps direct data to its correct destination, ICMP (Internet Control Message Protocol) is a core network layer protocol that reports on the success or failure of data delivery. It can indicate when part of a network is congested, when data fails to reach its destination, and when data has been discarded because the allotted Time to Live has expired (that is, when the data has traveled its allotted number of hops without reaching its destination). ICMP announces these transmission failures to the sender, but it does not correct errors it detects—those functions are left to higher-layer protocols, such as TCP. Instead, ICMP’s announcements provide critical information for troubleshooting network problems. ICMP messages are generated automatically by network devices, such as routers, and by utilities, such as ping.

Because it operates at layer 3 alongside IP, ICMP messages contain both an IP header and an ICMP header. [Figure 4-11](javascript://) depicts an ICMP header that is inserted after the ICMP message’s IP header. The fields are explained in [Table 4-7](javascript://). Note that the data field in the bottom row of the table does not belong to the ICMP header.

**Figure 4-11**

An ICMP packet



**Table 4-7**

### An ICMP packet

| **Field** | **Length** | **Function** |
| --- | --- | --- |
| Type | 8 bits | Indicates the type of ICMP message, such as Destination Unreachable. |
| Code | 8 bits | Indicates the subtype of the message, such as Destination host unknown. |
| Checksum | 16 bits | Allows the receiving node to determine whether the ICMP packet became corrupted during transmission. |
| Rest of header | 32 bits | Varies depending on message type and subtype. |
| Data | Variable | Usually contains the IP header and first 8 bytes of the data portion of the IP packet that triggered the ICMP message. |

IPv6 relies on ICMPv6 (Internet Control Message Protocol for use with IPv6) to perform the functions that ICMPv4 and ARP perform in IPv4 networks. This includes detecting and reporting data transmission errors, discovering other nodes on a network, and managing multicasting. To understand the different purposes of ICMPv4 and ICMPv6, let’s first take a closer look at ARP at layer 2 on IPv4 networks.

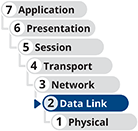
Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 4-1eARP (Address Resolution Protocol) on IPv4 Networks



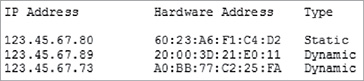
[**ARP (Address Resolution Protocol)**](javascript://) works in conjunction with IPv4 to discover the MAC address of a node on the local network and to maintain a database that maps local IPv4 addresses to MAC addresses. ARP is a layer 2 protocol that works with IPv4 in layer 3. It’s sometimes said to function at layer 2.5 because it touches information (IP addresses and MAC addresses) at both layers. However, it operates only within its local network bound by routers.

ARP relies on broadcasting, which transmits simultaneously to all nodes on a particular network segment. For example, if one node needs to know the MAC address of another node on the same network, the first node issues a broadcast message to the network, using ARP, that essentially says, “Will the computer with the IP address 1.2.3.4 please send me its MAC address?” The node with the IP address 1.2.3.4 then transmits a reply containing its physical address.

The database of IP-to-MAC address mappings is called an [**ARP table**](javascript://) or ARP cache, and it is kept on a computer’s hard drive. Each OS can use its own format for the ARP table. A sample ARP table is shown in [Figure 4-12](javascript://).

**Figure 4-12**

Sample ARP table



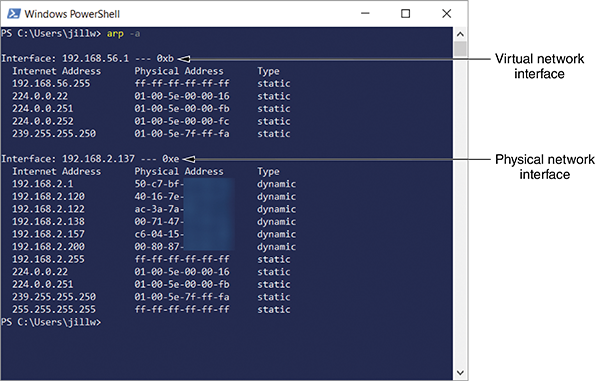
ARP tables might contain two types of entries:

* [**Dynamic ARP table entries**](javascript://) are created when a client makes an ARP request for information that could not be satisfied by data already in the ARP table; once received, the new information is recorded in the table for future reference.
* [**Static ARP table entries**](javascript://) are those that someone has entered manually using the ARP utility. This ARP utility, accessed via the arp command in both Windows and Linux, provides a way of obtaining information from and manipulating a device’s ARP table.

To view a Windows workstation’s ARP table, open a PowerShell or Command Prompt window and enter the command arp -a. [Figure 4-13](javascript://) shows sample results of this command run on a computer connected to a home network and to a virtual network in VirtualBox. The first line of each set of records contains the interface IP address, which is the local computer’s address on that network. The columns and rows below it contain the addresses of other nodes on the network, along with their physical addresses (MAC addresses) and record types.

**Figure 4-13**

The **arp -a** command lists devices on the network



The arp command can be used on IPv4 devices to diagnose and repair problems with ARP tables, as described next:

* If you notice inconsistent connectivity issues related to certain addresses, you might need to flush the ARP table on any device experiencing the problem with the command arp -d. This forces the device to repopulate its ARP table and correct any errors. You can also list a specific IP address to delete only that one record from the ARP table: arp -d 192.168.1.15. After deleting an entry, you can run a ping to repopulate the ARP table with the target device’s information.
* The command arp -s can be used to add a static entry to the ARP table. For example, the following command run with elevated permissions would add a mapping of the listed IP address with the listed MAC address: arp -s 192.168.1.15 00-11-22-AA-BB-CC

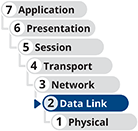
Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 4-1fNDP (Neighbor Discovery Protocol)



On IPv4 networks, neighbor discovery is managed by ARP with help from ICMP. However, IPv6 devices learn about other devices on their networks through a process called neighbor discovery. [**NDP (Neighbor Discovery Protocol)**](javascript://) information carried in ICMPv6 messages automatically detects neighboring devices and automatically adjusts when neighboring nodes fail or are removed from the network. NDP eliminates the need for ARP and some ICMP functions in IPv6 networks, and it’s much more resistant to hacking attempts than ARP.

The SLAAC process you learned about earlier is managed by NDP, as are router and network prefix discovery and neighbor discovery. NDP offers several ICMPv6 message types to perform these tasks, as follows:

* **RA (router advertisement)**—A router periodically sends an RA message out each of its configured interfaces to provide information about the network prefix, link MTU, and hop limits. The router might also advertise itself as the default router.
* **RS (router solicitation)**—To avoid waiting for the next scheduled RA message, a newly connected IPv6 host can send an RS message to request information from the router right away.
* **Redirect**—A router might send this type of message to inform hosts on the network that another router is a better gateway for a particular destination network.
* **NS (neighbor solicitation)**—IPv6 devices send NS messages to request the MAC address of a neighboring node (in IPv6, the MAC address is called the [**link-layer address**](javascript://)). These messages are used to ensure no two devices are using the same IPv6 address and to verify a neighbor’s reachability.
* **NA (neighbor advertisement)**—IPv6 devices send NA messages in response to NS messages to inform other network devices of their MAC address information.

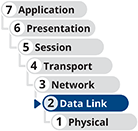
Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 4-1gEthernet



The most important data link layer standard, Ethernet, is adaptable, capable of running on a variety of network media, and offers excellent throughput at a reasonable cost. Because of its many advantages, Ethernet is, by far, the most popular network technology used on modern LANs. [**Ethernet II**](javascript://) is the current Ethernet standard and was developed by DEC, Intel, and Xerox (abbreviated as DIX) before IEEE began to standardize Ethernet.

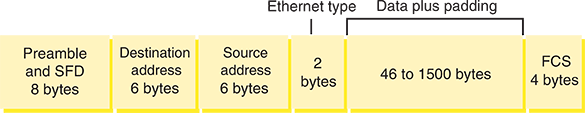
Other Ethernet standards exist, such as Ethernet 802.3. However, those standards tend to be used only for backward compatibility or to meet the needs of specific situations or vendor-specific devices. Other Ethernet standards add more fields to the header, which allows for more nuanced control of messages (even establishing a session similar to what TCP does at the transport layer). The tradeoff is that less space is allotted for data when using additional fields from other Ethernet standards. Interestingly, the data link layer can be divided into two sublayers, as follows:

* [**LLC (logical link control) sublayer**](javascript://)—The upper portion of the data link layer that identifies the type of message (the only LLC sublayer function in an Ethernet II frame) and handles multiplexing, flow and error control, and reliability (requires other types of Ethernet frames).
* [**MAC sublayer**](javascript://)—The lower portion of the data link layer that identifies the destination and source MAC addresses, includes the message, and provides the checksum in the frame’s trailer.

Unlike higher-layer protocols, Ethernet adds both a header and a trailer to the payload it inherits from the layer above it. This creates a frame around the payload. [Figure 4-14](javascript://) depicts an Ethernet II frame, and the details of the Ethernet II frame fields are listed in [Table 4-8](javascript://).

**Figure 4-14**

Ethernet II frame



**Table 4-8**

### Fields of an Ethernet II frame

| **Field name** | **Length** | **Description** |
| --- | --- | --- |
| Preamble | 7 bytes | Synchronizes the recipient’s receiver clock. |
| SFD (start frame delimiter) | 1 byte | Indicates the frame is about to begin. |
| Destination address | 6 bytes | Provides the recipient’s MAC address. |
| Source address | 6 bytes | Provides the sender’s MAC address. |
| Type field | 2 bytes | Specifies the upper-layer protocol carried in the frame. For example, an IP packet has 0x0800 in the Type field. In an Ethernet II frame, this field is the only component of the LLC sublayer. |
| Data | 46 bytes to 1500 bytes | If the data is not at least 46 bytes, padding is added to meet the minimum. |
| FCS (frame check sequence) | 4 bytes | Ensures that the data at the destination exactly matches the data issued from the source using the CRC (cyclic redundancy check) algorithm. |

Notice in [Table 4-8](javascript://) that the preamble and SFD fields are not included when calculating a frame’s size. Most protocol analyzers such as Wireshark can’t capture these first two fields (and sometimes not even the FCS), as this data is removed from incoming transmissions by the hardware before it becomes visible to any but the most sophisticated capture tools.

Together, the header and the FCS make up the 18-byte “frame” around the data. The data portion of an Ethernet frame may contain from 46 to 1500 bytes of information. Therefore, you can calculate the minimum and maximum frame sizes:

[**MTU (maximum transmission unit)**](javascript://) is the largest size, in bytes, that routers in a message’s path will allow at the network layer. Therefore, this defines the maximum payload size that a layer 2 frame can encapsulate. For Ethernet, the default MTU is 1500 bytes, a value that is generally considered the Internet standard. However, other layer 2 technologies might allow higher MTUs or require lower MTUs. Because of the overhead present in each frame and the time it takes for the NIC to manage a frame, the use of larger frame sizes on a network generally results in faster throughput.

There are a couple of notable exceptions to Ethernet frame size limitations:

* Ethernet frames on a VLAN (virtual LAN) can have an extra 4-byte field between the Source address field and the Type field, which is used to manage VLAN traffic. If this field exists, the maximum frame size is 1522 bytes. You’ll learn more about VLANs later.
* Some special-purpose networks use a proprietary version of Ethernet that allows for a [**jumbo frame**](javascript://), in which the MTU can be set above 9,000 bytes, depending on the type of Ethernet architecture used.

**Note 4-5**

You might have noticed that the maximum size of an IP packet is 65,535 bytes, while the maximum size of a network layer packet being transmitted over an Ethernet network is only 1500 bytes. Why the discrepancy?

Fragmentation is the process of dividing packets that are too large for a network’s hardware into smaller packets that can safely traverse the network. In an IPv4 network, routers examine incoming packets to determine if the packet size is larger than the outgoing interface’s MTU (that is, if the packet is larger than 1500 bytes) and if the packet is allowed to be fragmented. A packet that meets these two conditions will be divided into smaller packets, each with its own header that indicates its position in the series of fragments. While the IP packet handed down from the network layer can be thousands of bytes long, these longer packets will be fragmented into smaller messages for framing at the data link layer.

Fragmentation slows down network communications, so ideally, MTUs are set at a level that works for all devices along the message’s path. TCP also helps avoid fragmentation by negotiating at the beginning of a session an MSS (maximum segment size), which defines the maximum size of the transport layer message.

### Legacy Networking: Collisions and CSMA/CD

When IEEE released its first 802.3 standard in the early 1980s, it was officially called IEEE 802.3 CSMA/CD (Carrier Sense Multiple Access with Collision Detection), and was unofficially called Ethernet after the similar DIX standard that was published a few years earlier. As you’ve already learned, a CSMA/CD frame uses a slightly different layout than the Ethernet II frame layout. The IEEE frame is called an 802.3 frame, and the Ethernet II frame is called a DIX frame.

CSMA/CD networks often used a hub at the physical layer of the OSI model. All nodes connected to a hub compete for access to the network. The MAC (media access control) method used by nodes for arbitration on the network is [**CSMA/CD (Carrier Sense Multiple Access with Collision Detection)**](javascript://). Take a minute to think about the full name Carrier Sense Multiple Access with Collision Detection:

* Carrier Sense refers to an Ethernet NIC listening and waiting until no other nodes are transmitting data.
* Multiple Access refers to several nodes accessing the same network media.
* Collision Detection refers to what happens when two nodes attempt a transmission at the same time.

When the transmissions of two nodes interfere with each other, a [**collision**](javascript://) happens. After a collision, each node waits a random amount of time and then resends the transmission. A [**collision domain**](javascript://) is the portion of a network in which collisions can occur. Hubs connecting multiple computers in a star-bus topology resulted in massive collisions.

Recall that structured cabling guidelines provide detailed recommendations on the maximum distances cable segments can run between nodes. It’s interesting to note that these maximum cable lengths are partly determined by CSMA/CD. If a cable is too long, the entire message can be transmitted before a collision can be detected. In this case, the node does not know to resend the corrupted transmission.

To ensure that any collisions are detected, frames are made large enough to fill the entire cable during transmission. It might seem odd to think about a transmission “filling a cable,” but think about water going through a water hose. You can turn on the spigot and run the water for a very short time. The water runs through the hose to the other end but the hose isn’t filled all at the same time. Only if you leave the water running long enough, will water start coming out the other end while it’s still entering the hose at the spigot. With a long enough transmission, a similar thing happens on a cable—the beginning of the message starts arriving at its destination before the end of the message has been completely transmitted.

Today’s networks still use DIX and 802.3 frames. However, you’re unlikely to encounter a hub on a modern network. Instead, each connection to a switch consists of its own collision domain, and this connection can support communication in two directions at the same time. Therefore, the CSMA/CD process rarely plays a significant role because collisions are nearly impossible on today’s wired networks.

### Exam Tip

The CompTIA Network + exam expects you to understand how CSMA/CD works and how to troubleshoot problems from collisions.

**Remember This…**

* Explain the differences between TCP and UDP at the transport layer.
* Identify similarities and differences in headers for TCP, UDP, IP, ICMP, and Ethernet.
* Use arp -a to view a device’s ARP table.

**Self-Check**

1. Which protocol’s header includes the source MAC address?

Answer

* 1. Ethernet
  2. UDP
  3. IP
  4. TCP

1. Which of these protocols does not include some kind of integrity check field in its header?

Answer

* 1. TCP
  2. ICMP
  3. IPv6
  4. IPv4

1. An ARP table maps MAC addresses to what information?

Answer

* 1. IPv6 addresses
  2. Physical interfaces
  3. TCP or UDP ports
  4. IPv4 addresses

**You’re Ready**

You’re now ready to complete [Project 4-1: Install and Use WSL (Windows Subsystem for Linux)](javascript://), or you can wait until you’ve finished reading this module.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# 4-2Encryption Protocols

### Certification

* 1.5

Explain common ports and protocols, their application, and encrypted alternatives.

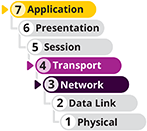
* 4.1

Explain common security concepts.

* 5.5

Given a scenario, troubleshoot general networking issues.

Average reading time: 19 minutes



So far in this module, you’ve seen how protocols package data to move it across a network. What protocols help to keep that data safe? In terms of security, data exists generally in three states:

* **At rest**—Data is most secure when it’s stored on a device that is protected by a firewall, anti-malware software, and physical security (such as being inside a locked room). However, these protections are no guarantee. Additional protections include storing portions of the data in separate locations so that no single portion is meaningful on its own.
* **In use**—For data to be used, it must be accessible, which brings inherent risk. Tightly controlling access to the data and reliable authentication of users help reduce these risks. You’ll learn more about access control and authentication methods later.
* **In motion**—This is when data is most vulnerable. Especially when data must leave your own, trusted network, it’s exposed to a multitude of potential gaps, intrusions, and weak links. Wireless transmissions, especially, are susceptible to interception. And wired transmissions also risk exposure. The number of devices, organizations, and transmission methods involved in sending a single email across the Internet highlights the need for a layer of security that travels with the data.

Encryption is the last layer of defense against data theft. In other words, if an intruder has bypassed all other methods of security, including physical security (for instance, they have broken into the data center) and network design security (for instance, they have defied a firewall’s packet-filtering techniques or removed encapsulated frames from transmissions), data in motion or at rest may still be safe if it is encrypted. [**Encryption**](javascript://) protocols use a mathematical code, called a cipher, to scramble data into a format that can be read only by reversing the cipher—that is, by deciphering, or decrypting, the data. The purpose of encryption is to keep information private. Many forms of encryption exist, with some being more secure than others. Even as new forms of encryption are developed, new ways of cracking their codes emerge, too.

To protect data at rest, in use, and in motion, encryption methods are primarily evaluated by three benchmarks:

* **Confidentiality**—Data can only be viewed by its intended recipient or at its intended destination.
* **Integrity**—Data is not modified in the time after the sender transmits it and before the receiver picks it up.
* **Availability**—Data is available and accessible to the intended recipient when needed, meaning the sender is accountable for successful delivery of the data.

Together, these three principles form the standard security model called the [**CIA (confidentiality, integrity, and availability) triad**](javascript://). Encryption can happen at various layers of the OSI model. Let’s first begin with a brief description of what key encryption is, and then you’ll learn about some of the most common encryption protocols used to protect data stored on or traveling across networks.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

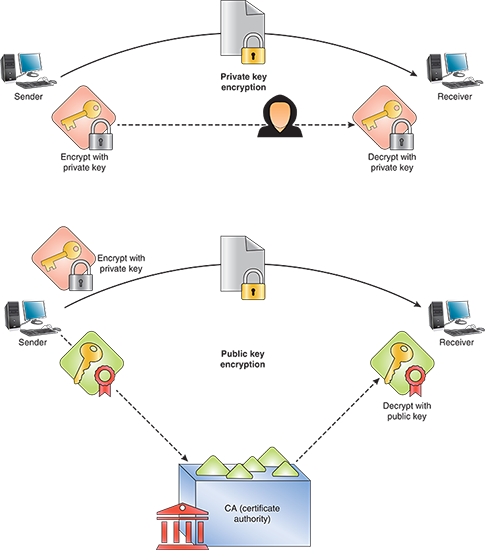
## 4-2aKey Encryption

The most popular kind of encryption encodes the original data’s bits using a [**key**](javascript://), or a random string of characters, to scramble the data—sometimes several times in different sequences—and generates a unique and consistently sized data block called ciphertext. The key is created according to a specific set of rules, or algorithms. Key encryption can be separated into two categories, as illustrated in [Figure 4-15](javascript://) and described next:

* [**Private key encryption**](javascript://)—Data is encrypted using a single key that only the sender and the receiver know. Private key encryption is also known as [**symmetric encryption**](javascript://) because the same key is used during both the encryption and decryption of the data. A potential problem with private key encryption is that the sender must somehow share the key with the recipient without it being intercepted.
* [**Public key encryption**](javascript://)—Data is encrypted with a private key known only to the user, and it’s decrypted with a mathematically related public key that can be made available through a third-party source, such as a public key server. This ensures data integrity, as the sender’s public key will only work if the data has not been tampered with. Alternatively, data can be encrypted with the public key, and then can only be decrypted with the matching private key. This ensures data confidentiality, as only the intended recipient (the owner of the keys) can decrypt the data. A public key server is a publicly accessible host, such as a server on the Internet, that freely provides a list of users’ public keys, much as a telephone book provides a list of peoples’ phone numbers. The combination of a public key and a private key is known as a key pair. Because public key encryption requires the use of two different keys, one to encrypt and the other to decrypt, it is also known as [**asymmetric encryption**](javascript://).

**Figure 4-15**

Private key encryption uses only one key, which must be securely communicated between sender and receiver, while public key encryption relies on a second, public key that can safely be obtained by anyone



Enlarge Image

With the abundance of private and public keys, not to mention the number of places where each may be kept, users need simple and secure key management. One answer to this problem is to use digital certificates. A person or a business can request a [**digital certificate**](javascript://), which is a small file containing that user’s verified identification information and the user’s public key. The digital certificate is issued, maintained, and validated by an organization called a [**CA (certificate authority)**](javascript://). The use of certificate authorities to associate public keys with certain users is known as [**PKI (public-key infrastructure)**](javascript://).

**Note 4-6**

Digital certificates are primarily used to certify and secure websites where financial and other sensitive information is exchanged, but they’re also used for other types of websites and to secure email communications, to authenticate client devices in a domain, or to authenticate users to a network. [**Authentication**](javascript://) involves a process of ensuring that a user, device, or application is who they say they are. When surfing the web, at some point you might have gotten an error that said the website’s SSL certificate was untrusted. Certificate issues like this might mean the website’s digital certificate used by the encryption protocol SSL/TLS (which is used to secure HTTP) was not signed by a trusted CA, the certificate has expired, or it wasn’t associated with a trusted root certificate. The browser alerts you that the website you’re about to visit might not be the website you think it is.

The next two sections detail specific protocols, including SSL and TLS, that are used to encrypt data as it is transmitted over a network. The first encryption protocol in the list, IPsec, operates at the network layer.

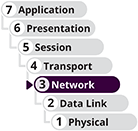
Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 4-2bIPsec (Internet Protocol Security)



[**IPsec (Internet Protocol Security)**](javascript://) is an encryption protocol suite that defines a set of rules for encryption, authentication, and key management for TCP/IP transmissions. It is an enhancement to IPv4 and is native to IPv6. IPsec works at the network layer of the OSI model—it adds security information to the headers of IP packets and encrypts the data payload. IPsec creates secure connections in five steps, as follows:

1. **IPsec initiation**—Noteworthy traffic, as defined by a security policy, triggers the initiation of the IPsec encryption process.
2. **Key management**—Through a key management process, two nodes agree on common parameters for the keys they will use. This phase primarily includes two services:
   * **IKE (Internet Key Exchange)**—Negotiates the exchange of keys, including authentication of the keys; the current version is IKEv2, which you’ll see again in the discussion on VPNs (virtual private networks) later in this module.
   * **ISAKMP (Internet Security Association and Key Management Protocol)**—Works within the IKE process to establish policies for managing the keys.
3. **Security negotiations**—IKE continues to establish security parameters and associations that will serve to protect data while in transit.
4. **Data transfer**—After parameters and encryption techniques are agreed upon, a secure channel is created, which can be used for secure transmissions until the channel is broken. Data is encrypted and then transmitted. Either [**AH (authentication header)**](javascript://) encryption or [**ESP (Encapsulating Security Payload)**](javascript://) encryption may be used. Both types of encryption provide authentication of the IP packet’s data payload through public key techniques. In addition, ESP encrypts the entire IP packet for added security.
5. **Termination**—IPsec requires regular reestablishment of a connection to minimize the opportunity for interference. To maintain communication, the connection can be renegotiated and reestablished before the current session times out.

IPsec can be used with any type of TCP/IP transmission and operates in two modes:

* **Transport mode**—Connects two hosts.
* **Tunnel mode**—Runs on routers or other connectivity devices in the context of VPNs.

Go to pg.

[**help**](javascript://)

Application OpenedApplication Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

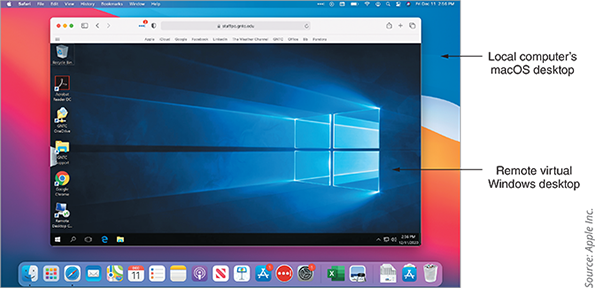
## 4-3bTerminal Emulation



A [**terminal emulator**](javascript://) is software that allows a user on one computer, called the client, to control another computer, called the host or server, across a network connection. Examples of command-line software that can provide terminal emulation include Telnet and SSH. Some GUI-based software examples are Remote Desktop for Windows, join.me, VNC (virtual network computing), and TeamViewer. You’ll explore these options further in a moment. For now, understand that a host may allow clients a variety of privileges, from merely viewing the screen to running applications and modifying data files on the host’s hard disk. After connecting, if the remote user has sufficient privileges, they can send keystrokes and mouse clicks to the host and receive screen output in return. In other words, to the remote user, it appears as if they’re working on the LAN- or WAN-connected host itself. For example, a traveling salesperson can use a laptop to “remote in” to a desktop computer at corporate headquarters. This way, they can remotely update a workbook file stored on the desktop computer using Excel, an application that is also installed on the desktop. Another example is when a student needs to use Windows-only software that is not compatible with their MacBook. Instead, the student can remote into a Windows lab computer at their school from their MacBook and work with the Windows software remotely, as shown in [Figure 4-19](javascript://).

**Figure 4-19**

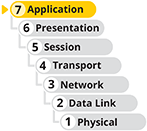
VDI connection to a Windows computer from a MacBook



Enlarge Image

Source: Apple Inc.

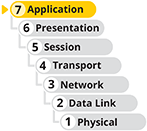
### Telnet



As you’ve learned, the Telnet protocol is a terminal emulation utility used by Telnet client/server applications that allow an administrator or other user to control a computer remotely. For example, if you were a network administrator working at one building on your school’s campus and had to modify the configuration on a router in another building, you could use Telnet to access the router and run commands to change configuration settings. However, Telnet provides little security for establishing a connection (poor authentication) and no security for transmitting data (no encryption).

While Telnet is typically installed by default in most popular OSs, often it must first be enabled. A Telnet connection is created from the client computer’s CLI using the telnet command. For example, the command telnet 192.168.2.1 will connect to a device on the network at the IP address listed (such as a router), and telnet lab-owl will connect to a computer on the network named lab-owl (such as a server). In both cases, you must enter a password to complete the connection. You could then interact with the host’s CLI using your local computer’s CLI. For example, you might want to configure the remote device’s interfaces. To close a telnet session, access the telnet prompt again with the keyboard shortcut Ctrl+] (that’s the Ctrl key with the close-bracket key) and enter the command quit, or just close the CLI window.

### SSH (Secure Shell)



SSH (Secure Shell) is a collection of protocols that performs both authentication and encryption. With SSH, you can securely log on to a host, execute commands on that host, and copy files to or from the host. SSH encrypts data exchanged throughout the session. It guards against several security threats, including the following:

* Unauthorized access to a host
* Interception of data in transit, even if the data must be transferred via intermediate nodes
* IP [**spoofing**](javascript://), in which an attacker attempts to hide their identity or impersonate another device by modifying the IP header
* DNS spoofing, in which a hacker forges name server records to falsify their host’s identity

Depending on the version, SSH may use Triple DES, AES, Blowfish, or other, less-common encryption schemes or techniques.

### Exam Tip

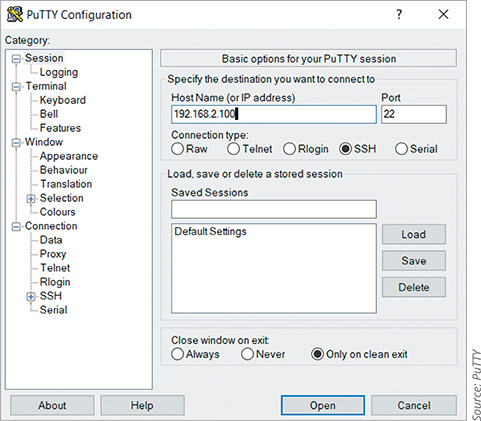
After completing your Network + certification, you might consider studying for the Security + certification. CompTIA’s Security + exam covers a broad range of foundational topics in IT security, including the encryption techniques listed here for SSH. This essential understanding of security concerns, techniques, and concepts will serve you well no matter which area of IT you choose to specialize in.

SSH was developed by SSH Communications Security, and use of their SSH implementation requires paying for a license. However, open source versions of the protocol suite, such as OpenSSH, are available for most computer platforms.

To form a secure connection, SSH must be running on both the client and server. Like Telnet, the SSH client is a utility that can be run at the shell prompt on a UNIX or Linux system or at a CLI on a Windows-based system. Other versions of the program come with a graphical interface. The SSH suite of protocols is included with all modern UNIX and Linux distributions and with macOS client operating systems. For Windows-based computers, you need to download a freeware SSH client, such as PuTTY ([putty.org](http://putty.org/" \t "_blank)). You can see in [Figure 4-20](javascript://) that PuTTY supports several connection types, including both SSH and Telnet. PuTTY can also be run from the command line. In a later module, you’ll practice using PuTTY to remotely connect to a VM in the cloud.

**Figure 4-20**

On a Windows computer, use an app like PuTTY to create a SSH connection to another computer



Source: PuTTY

SSH allows for password authentication or authentication using public and private keys. The following steps describe how to authenticate using keys:

1. Step 1

Generate a public key and a private key on your client workstation by running the ssh-keygen command (or by choosing the correct menu options in a graphical SSH program). The keys are saved in two different, encrypted files on your hard disk.

1. Step 2

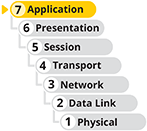
Transfer the public key to an authorization file on the host to which you want to connect.

1. Step 3

When you connect to the host via SSH, the client and host exchange public keys. If both can be authenticated, the connection is completed.

SSH listens at port 22 and is highly configurable. For example, you can choose among several encryption methods. At the end of this module in [Capstone Project 4-2](javascript://), you’ll practice using SSH in Ubuntu.

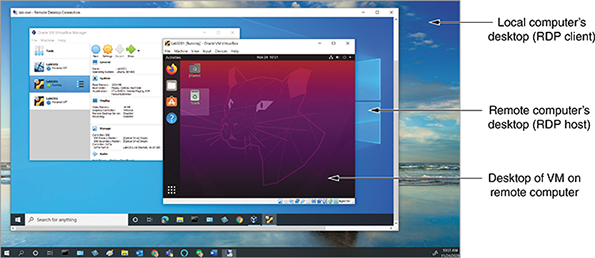
### RDP (Remote Desktop Protocol)



Recall that RDP (Remote Desktop Protocol) is a Microsoft proprietary protocol used by Windows Remote Desktop and Remote Assistance client/server utilities to connect to and control a remote computer. RDP provides a [**remote desktop connection**](javascript://) so that you see on your local computer’s screen what you would see if you were sitting in front of the remote computer instead. It’s not just a CLI like what you get with SSH and Telnet. It’s a GUI desktop with windows, icons, shortcut keys, menus, and sound. You can even run a hypervisor on the remote desktop so that, from your local computer, you can control the remote host and all its VMs. [Figure 4-21](javascript://) shows a local computer’s Windows desktop, a remote desktop connection with another Windows computer, and an Ubuntu VM running on the remote computer. At the end of this module, you’ll practice using RDP to connect two computers.

**Figure 4-21**

From one keyboard, interact with the local desktop, the remote desktop, and the VM’s desktop



Enlarge Image

For larger corporate networks, a simple RDP connection might not suffice. For example, if you have several remote workers connecting to their office desktops at the same time, you might run out of your available public IP addresses and then have to manage connections through a process called port forwarding. Instead, a [**remote desktop gateway**](javascript://) run from a single Windows server can manage all these RDP connections to the network’s computers through a single public IP address. The remote desktop gateway can also link to Active Directory’s authentication services, manage user authorization to control which users can access which network resources once logged in, and audit activity through all hosted RDP connections. Additionally, the remote desktop gateway incorporates SSL/TLS to provide secure connections to all users so there’s no need for additional encryption through a VPN.

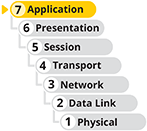
**Note 4-8**

Authentication and authorization are two different processes. Authentication is the sign-in process that allows a user to access a resource such as a network if they provide the correct credentials, such as username and password. Authorization determines what the user can do once they get into the resource. For example, any employee might be allowed to sign into the network (authentication), but only an administrator has permission to edit user accounts (authorization). A remote desktop gateway can manage both these processes independently for all network resources rather than for only a single desktop.

Suppose you have many remote users, such as salespeople for a company or students at a school, who regularly need access to a desktop managed by your company. You want to offer a consistent desktop experience with installed applications that your users commonly need. However, you don’t want to purchase dozens of physical computers for these users to connect to. Instead, you can offer virtual desktops running in VMs that the system creates, or “spins up,” only when needed. This implementation is called [**VDI (Virtual Desktop Infrastructure)**](javascript://). It differs from a traditional RDP connection in that the VDI connection targets only VMs. However, VDI can use RDP to create the connection to each VM. VDI also offers greater flexibility with options to access VMs running many different OSs or many different configurations of installed applications. VDI can also provide either persistent or non-persistent instances. With a persistent instance, when the user remotes back into the desktop, any changes they made will still be there including files they saved. With a non-persistent instance, the desktop is reset each time someone signs in.

In contrast, [**RDS (Remote Desktop Services)**](javascript://) uses RDP to allow multiple users to access the same virtual or physical Windows Server system at one time. RDS can provide access to the entire Server OS or just to a single application using RemoteApp. RDS is cheaper to support than VDI; however, RDS is more difficult to manage and more limited in customization options.

### VNC (Virtual Network Computing)



[**VNC (Virtual Network Computing)**](javascript://) is similar in concept to RDP but uses the cross-platform protocol RFB (remote frame buffer) to remotely control a workstation or server. VNC is slower than Remote Desktop and requires more network bandwidth. However, because VNC is open source, many companies have developed their own software that can

* Run OSs on client computers
* Remotely access computers, tablets, and smartphones
* Remotely control media equipment and surveillance systems

### Out-of-Band Management



Telnet, SSH, RDP, and VNC all rely on the existing network infrastructure for a network administrator to remotely control the device. Before the devices can be configured, they must already be booted up, and they must already have configuration software installed. This is called [**in-band management**](javascript://), and it inherently limits troubleshooting capabilities. [**Out-of-band management**](javascript://), however, relies on a dedicated connection (either wired or wireless) between the network administrator’s computer and each critical network device, such as routers, firewalls, servers, power supplies, applications, and security cameras. These dedicated connections allow network administrators to remotely

* Power up a device
* Change firmware settings
* Reinstall operating systems
* Monitor hardware sensors
* Troubleshoot boot problems
* Limit network users’ access to management functions
* Manage devices even when other parts of the network are down

Out-of-band management solutions come in an array of options, from basic reboot abilities to full-scale device management. A remote management card is attached to the network device’s console port, or sometimes the remote management card is built into the device. A dial-in modem—either through a wired phone line or through a cellular connection—might be attached to the device to provide backup CLI access in the event of a catastrophic network shutdown. A single device, such as a console server or console router, provides centralized management of all linked devices.

Go to pg.

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 4-2cSSL (Secure Sockets Layer) and TLS (Transport Layer Security)

SSL (Secure Sockets Layer) and TLS (Transport Layer Security) are both methods of encrypting TCP/IP transmissions—including web pages and data entered into web forms—en route between the client and server using public key encryption technology. The two protocols can work side by side and are widely known as SSL/TLS or TLS/SSL. All browsers today (for example, Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge, and Internet Explorer) support SSL/TLS to secure HTTP transmissions.

SSL was originally developed by Netscape and operates in the application layer. Since that time, the IETF (Internet Engineering Task Force), which is an organization of volunteers who help develop Internet standards, has standardized the similar TLS protocol. TLS operates in the transport layer and uses slightly different encryption algorithms than SSL, but otherwise is essentially the updated version of SSL. SSL has now been deprecated and should be disabled whenever possible, leaving the more secure TLS to provide protection. In reality, you’ll often see them both enabled for backward compatibility. You’ll also often see the terms used interchangeably—many times, even when someone says SSL, they’re referring to TLS.

As you recall, HTTP uses TCP port 80, whereas HTTPS (HTTP Secure) uses SSL/TLS encryption and TCP port 443 rather than port 80. Other protocols you’ve studied that offer SSL/TLS encrypted alternatives include SMTP TLS, LDAP over SSL, IMAP over SSL, and POP3 over SSL. Each time a client and server establish an SSL/TLS connection, they establish a unique session, which is an association between the client and server that is defined by an agreement on a specific set of encryption techniques. The session allows the client and server to continue to exchange data securely as long as the client is still connected to the server. A session is created by a handshake protocol, one of several protocols within SSL/TLS, and perhaps the most significant. As its name implies, the handshake protocol allows the client and server to introduce themselves to each other and establishes terms for how they will securely exchange data.

This handshake conversation is similar to the TCP three-way handshake you learned about earlier in this module. Given the scenario of a browser accessing a secure website, the SSL/TLS handshake works as follows:

1. Step 1

The browser, representing the client computer in this scenario, sends a client hello message to the web server, which contains information about what level of security the browser is capable of accepting and what type of encryption the browser can decipher. The client hello message also establishes a randomly generated number that uniquely identifies the client and another number that identifies the session.

1. Step 2

The server responds with a server hello message that confirms the information it received from the browser and agrees to certain terms of encryption based on the options supplied by the browser. Depending on the web server’s preferred encryption method, the server might choose to issue to the browser a public key or a digital certificate.

1. Step 3

If the server requests a certificate from the browser, the browser sends it. Any data the browser sends to the server is encrypted using the server’s public key. Session keys used only for this one session are also established. After the browser and server have agreed on the terms of encryption, the secure channel is in place and they begin exchanging data.

The certificate and key exchange process can add more steps, requiring up to 10 steps for a TLS 1.2 handshake. Recall that public key encryption is more secure than private key encryption but also takes longer. During a TLS 1.2 handshake, public key encryption is used to establish a secure session. To save time going forward, part of the handshake process is negotiating the exchange of a private key so the conversation can continue at a faster pace using private key encryption. However, this requires three round-trip messages. The newest version, TLS 1.3, reduces the number of steps and requires only one round-trip sequence. In some cases, TLS 1.3 can avoid the handshake process altogether in what’s called 0-RTT (zero round-trip time).

**Note 4-7**

Transmissions over secure connections, such as when using HTTPS websites, might be intercepted but cannot be read. For example, suppose you are using unsecured Wi-Fi at a coffee shop and log on to Twitter from your laptop browser. Without TLS protecting your logon information, anyone lounging nearby can hack into, read, and steal your unencrypted wireless transmissions.

Some online activities, however, such as online banking, should never be performed on unsecure Wi-Fi hotspots. Despite the security provided by these encryption techniques, other steps of the process can break down. One example might include browsing an insecure portion of a website (HTTP) for part of the browsing session, which provides a brief opportunity for your browser to be hijacked by a hacker and sent to what looks like the official logon page, but really is not.

**Applying Concepts 4-4**

### Browser Security

You can change the settings in your browser to make sure you’re using the latest version of TLS. On a Windows machine, changes you make to one browser for these settings will affect other browsers installed on your computer. Complete the following steps:

1. 1

In the Settings app, search for **Internet options**, and then click **Internet Options** in the search results.

1. 2

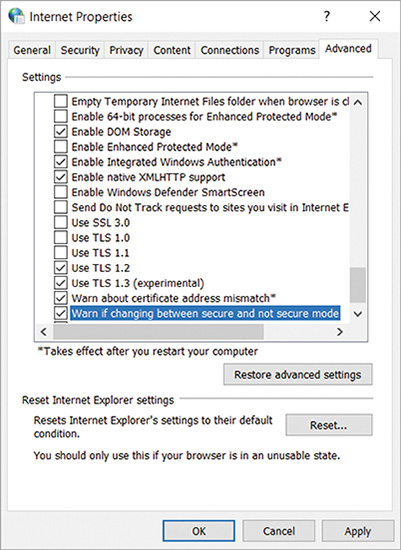
On the **Advanced** tab, scroll down to the Security section. Which SSL/TLS options are currently enabled?

1. 3

Disable **SSL 3.0**, **TLS 1.0**, and **TLS 1.1**. Make sure **TLS 1.2** and **TLS 1.3** are enabled. While TLS 1.3 might be labeled experimental, it was finalized in 2018 and provides better security at faster speeds and is already widely used by websites. Also, if you regularly use an unsecured wireless network like at a coffee shop or a restaurant, also select **Warn if changing between secure and not secure mode** so you’ll be notified when interacting with an unsecured website. See [Figure 4-16](javascript://). Click **OK**.

**Figure 4-16**

TLS 1.2 and TLS 1.3 provide the best security for surfing online



Enlarge Image

Some browsers will prevent navigation to unsecured websites when the warning option is checked as previously instructed. This is a good thing if you’re using a questionable network. But if you have trouble navigating to unsecured sites you feel comfortable with, you’ll need to go back and uncheck this option in Internet options.

### Caution

When visiting secure websites, it’s important to notice if you have a secure connection with a trusted website before entering personal information on that site. Edge, for example, shows a padlock icon when the site’s certificate has been identified and confirmed. This visual is still no guarantee, however, as scammers are now figuring out how to impersonate HTTPS websites’ credentials.

1. 4

In your browser, navigate to [paypal.com](http://paypal.com/" \t "_blank). What is the exact address shown in the address box after the page loads in the browser?

1. 5

Click the padlock icon and then click the certificate listed. What CA verified the legitimacy of the website?

Now that you understand a little about encryption and related security concerns, you’re ready to dive into remote connection protocols that require encryption for security.

**Remember This…**

* Explain how private key encryption and public key encryption work.
* Compare the roles of AH encryption and ESP encryption in IPsec.
* Describe the security provided by SSL/TLS for HTTP, SMTP, LDAP, IMAP, and POP3.

**Self-Check**

1. Which two components of the CIA triad are ensured by adequate encryption methods? Choose two.

Answer

* 1. Confidentiality
  2. Availability
  3. Accountability
  4. Integrity

1. Which IPsec encryption type encrypts the IP header?

Answer

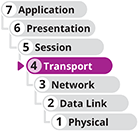
* 1. IKE
  2. ESP
  3. ISAKMP
  4. AH

Go to pg.

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 4-1bUDP (User Datagram Protocol)



UDP (User Datagram Protocol) is an unreliable, connectionless protocol. The term unreliable does not mean that UDP can’t be used reliably. Instead, it means that UDP does not guarantee delivery of data, and no connection is established by UDP before data is transmitted. By default, UDP provides no handshake to establish a connection, acknowledgment of transmissions received, error checking, sequencing, or flow control and is, therefore, more efficient and faster than TCP. Instead of conversing with someone on a phone call, this would be more like talking on a radio show where you send out your signal whether anyone is listening or not. UDP is useful when a great volume of data must be transferred quickly, such as live audio or video transmissions over the Internet. It’s also used for small requests, such as DNS, or in situations when the data changes often and speed is more important than complete accuracy, such as when gaming over a network.

In contrast to a TCP header’s 10 fields, the UDP header contains only four fields: Source port, Destination port, Length, and Checksum. Use of the UDP Checksum field is optional on IPv4 networks but required for IPv6 transmissions. [Figure 4-6](javascript://) depicts a UDP datagram. Contrast its header with the much larger TCP segment header shown earlier in [Figure 4-3](javascript://).

**Figure 4-6**

A UDP datagram



**Note 4-4**

Application layer protocols can work in conjunction with UDP to emulate some of the reliability normally provided by TCP. For example, RTP (Real-time Transport Protocol), which is used to transmit audio and video on the web, operates at the application layer of the OSI model and relies on UDP at the transport layer. RTP applies its own sequence numbers to indicate the order in which messages should be assembled at their destination. These sequence numbers also help to indicate whether messages were lost during transmission.

Now that you understand the functions of and differences between TCP and UDP at layer 4, you’re ready to step down a layer and learn more about IP (Internet Protocol) at layer 3. TCP segments and UDP datagrams are often passed down to IP for further encapsulation at the network layer.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 4-3dRemote Access Policies

A good remote access policy protects a company’s data, network, and liability, no matter what type of remote access is involved. Here are some common requirements:

* Devices used for remote access must be kept up to date with patches, anti-malware software, and a firewall.
* Device access must be controlled by a strong password and biometric measures, such as fingerprint, retina, or face recognition. The device should lock down automatically after only a few minutes of inactivity.
* Passwords must be strong and must be changed periodically. Password best practices are discussed further in later modules.
* Passwords cannot be shared, even with a family member.
* The device’s internal and external storage devices must be encrypted. Note that some countries require that encrypted storage devices be decrypted or that encryption keys be filed with authorities. Employees who travel abroad should account for this when deciding what data to transport.
* Company and customer data that is accessed, transferred, stored, or printed must be kept secure.
* The loss or theft of any devices used for remote access or to process remotely accessed data (such as a printer) must be reported to the company immediately (or within a reasonable time frame, such as 72 hours).
* Encrypted VPN software must be used to remotely access company network resources. Typically, these options are clearly defined in the policy.
* While remotely connected to the company network, the device must not be connected to the open Internet or any other network not fully owned and controlled by the employee. This restriction is usually built into full tunnel VPN solutions.
* Remote sessions must be terminated when not in use. In most cases, remote sessions should be configured to time out automatically as a precaution.

**Remember This…**

* Compare FTP, FTPS, SFTP, and TFTP.
* Given a scenario, choose the appropriate remote access tool: Telnet, SSH, RDP, VNC, or a VPN.
* Practice using Telnet, SSH, and Remote Desktop.
* Describe how a VPN works.

**Self-Check**

1. Which remote file access protocol uses port 22?

Answer

* 1. FTPS
  2. TFTP
  3. FTP
  4. SFTP

1. You need to remote into a Linux server in another building on your network. Which of the following protocols should you use?

Answer

* 1. RDP
  2. SSH
  3. SFTP
  4. VPN

1. You’re working from home and need to access a file server at the office while working in an application from your work desktop. At the same time, you often stream music in your browser. Which VPN type will be most efficient while still meeting your needs?

Answer

* 1. Full tunnel VPN
  2. Host-to-host VPN
  3. Site-to-site VPN
  4. Split tunnel VPN

**You’re Ready**

You’re now ready to complete [Project 4-2: Use Remote Desktop](javascript://), or you can wait until you’ve finished reading this module.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# 4-4Troubleshooting Network Issues

### Certification

* 5.3

Given a scenario, use the appropriate network software tools and commands.

* 5.5

Given a scenario, troubleshoot general networking issues.

Average reading time: 22 minutes

As with any type of communication, many potential points of failure exist in the TCP/IP transmission process. The number of points increases with the size of the network and the distance of the transmission. Fortunately, TCP/IP comes with a complete set of utilities that can help you track down most TCP/IP-related problems without using expensive software or hardware to analyze network traffic. You should be familiar with the purposes of the following tools and their parameters, not only because the CompTIA Network + certification exam covers them, but also because you will regularly need these tools in your work with TCP/IP networks.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 4-4aTroubleshooting Tools

You’ve already learned about eight very important TCP/IP utilities—ping, ipconfig, ip, ifconfig, hostname, nslookup, dig, telnet, and arp. The following sections present additional TCP/IP utilities that can help you discover information about nodes on your network. The module then concludes with a summary of all these utilities along with a few troubleshooting scenarios.

### netstat

The [**netstat**](javascript://) utility displays TCP/IP statistics and details about TCP/IP components and connections on a host. Information that can be obtained from the netstat command includes the following:

* The port on which a TCP/IP service is running
* Which network connections are currently established for a client
* How many messages have been handled by a network interface since it was activated
* How many data errors have occurred on a particular network interface

### Exam Tip

The netstat utility has been deprecated in most Linux distributions, although it’s still often included in the distribution by default. While the versatile ss (socket statistics) utility is designed to replace netstat, the CompTIA Network + exam still expects you to know how to use netstat. Both utilities use many of the same parameters.

To better understand what netstat can do, consider this example. Suppose you are a network administrator in charge of maintaining file, print, and web servers for an organization. You discover that your web server, which has multiple processors, sufficient hard disk space, and multiple NICs, is suddenly taking twice as long to respond to HTTP requests. Besides checking the server’s memory resources and its software for indications of problems, you can use netstat to determine the characteristics of traffic going into and out of each NIC. Perhaps you discover that one NIC is consistently handling 80 percent of the traffic instead of only half. You might run hardware diagnostics on the other NIC and discover that its onboard processor is failing, making it much slower than the first NIC.

[Table 4-9](javascript://) shows some parameters you can use with netstat in Windows. You can also use netstat on Linux machines with a different set of parameters.

**Table 4-9**

### netstat command options

| **netstat command** | **Description** |
| --- | --- |
| netstat | List all active TCP/IP connections on the local machine, including the transport layer protocol used (usually just TCP), messages sent and received, IP address, and state of those connections. |
| netstat -n | List current connections, including IP addresses and ports. |
| netstat -f | List current connections, including IP addresses, ports, and FQDNs. |
| netstat –a | List all current TCP connections and all listening TCP and UDP ports. |
| netstat -e | Display statistics about messages sent over a network interface, including errors and discards. |
| netstat -s | Display statistics about each message transmitted by a host, separated according to protocol type (TCP, UDP, IP, or ICMP). |
| netstat -r | Display routing table information. |
| netstat -o | List the PID (process identifier) for each process using a connection and information about the connection. |
| netstat -b | List the name of each process using a connection and information about the connection. Requires elevated permissions. |

**Note 4-9**

Command parameters can be combined into a single command. For example, entering the command netstat -an will display the IP addresses and ports of active TCP connections and also listening TCP and UDP ports.

### tracert or traceroute

The Windows **[tracert](javascript://)** utility uses ICMP echo requests to trace the path from one networked node to another, identifying all intermediate hops between the two nodes. Linux, UNIX, and macOS systems use UDP datagrams or, possibly, TCP SYN messages, for their [**traceroute**](javascript://) utility, but the concept is still the same.

**Note 4-10**

Traceroute can be configured to use TCP or ICMP messages. See the traceroute man pages to learn how to configure this and many other options.

Both traceroute and tracert utilities employ a trial-and-error approach to discover the nodes at each hop from the source to the destination, as described here:

* Tracert sends an ICMP echo request to the destination node and listens for an ICMP echo reply from that node.
* Traceroute sends UDP messages to a random, unused port on the destination node, and listens for an ICMP “Port Unreachable” error message in response from that node.
* Both utilities limit the TTL of these repeated trial messages, called [**probes**](javascript://), thereby triggering routers along the route to return specific information about the route being traversed. In fact, by default, they send three probes with each iteration so averages can be calculated from the three responses at each step.

Study [Figure 4-27](javascript://) to see how a trace works with traceroute. The steps are also described next:

1. Step 1

The first three UDP datagrams transmitted have their TTL set to 1. Because the TTL determines how many more network hops a datagram can make, datagrams with a TTL of 1 expire as they hit the first router. When they expire, an ICMP error message is returned to the source—in this case, the node that began the trace.

1. Step 2

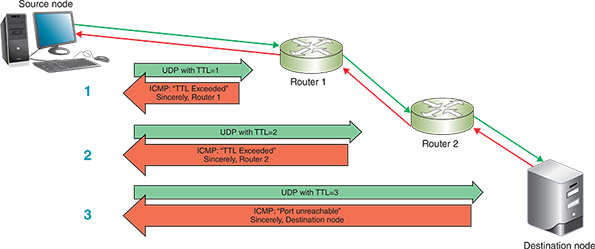
Using the return messages, the trace now knows the identity of the first router. It then transmits a series of datagrams with a TTL of 2 to determine the identity of the second router.

1. Step 3

The process continues for the next router in the path, and then the fourth, fifth, and so on, until the destination node is reached. The trace also returns the amount of time it took for the datagrams to reach each router in the path.

**Figure 4-27**

The traceroute utility uses error messages from routers to map nodes on a route



Enlarge Image

This process is identical for tracert in Windows except for two modifications. First, the probes sent from the source are ICMP echo request messages; each message is still limited by the specific TTL restrictions. Second, the final reply from the destination node is an ICMP echo reply rather than an ICMP port unreachable error message.

**Applying Concepts 4-5**

### Trace the Route to [Google.com](http://google.com/" \t "_blank)

You can perform a trace using an IP address or a host name. On a UNIX or Linux system, the command syntax would be the following:

An example of the trace route command. The command is issued thus. trace route 8 dot 8 dot 8 dot 8. The command can be issued in the following way as well. trace route google dot com.

Because tracert is installed by default on Windows, use a Windows machine for this exercise instead:

1. On a Windows system, perform a trace on one of Google’s public DNS servers with the command **tracert 8.8.8.8**. How many hops were traced? What is the IP address of the final hop?
2. Use tracert to perform a trace on Google’s web server with the command **tracert** [google.com](http://google.com/" \t "_blank). How many hops were traced this time? What is the IP address of the final hop? Why is this IP address different than the IP address of the final hop in the previous step?

The traceroute or tracert command has several available parameters. [Table 4-10](javascript://) describes some of the more popular ones.

**Table 4-10**

### traceroute and tracert command options

| **Command** | **Description** |
| --- | --- |
| traceroute –n google.com  or  tracert –d google.com | Instruct the command to not resolve IP addresses to host names. |
| traceroute –m 12 google.com  or  tracert –h 12 google.com | Specify the maximum number of hops when attempting to reach a host; this parameter must be followed by a specific number. Without this parameter, the command defaults to 30. |
| traceroute –w 2 google.com  or  tracert –w 2000 google.com | Identify a timeout period for responses; this parameter must be followed by a variable to indicate the number of seconds (in Linux) or milliseconds (in Windows) that the utility should wait for a response. The default time is usually between 3 and 5 seconds for Linux and 4000 milliseconds (4 seconds) for Windows. |
| traceroute –f 3 google.com | Set the first TTL value and must be followed by a variable to indicate the number of hops for the first probe. The default value is 1, which begins the trace at the first router on the route. Beginning at later hops in the route can more quickly narrow down the location of a network problem. tracert does not have a corresponding parameter for this function. |
| traceroute –I google.com | Instruct the command to use ICMP echo requests instead of UDP datagrams. |
| traceroute –T google.com | Instruct the command to use TCP SYN probes instead of UDP datagrams. |
| traceroute –4 google.com  or  tracert –4 google.com | Force the command to use IPv4 packets only. |
| traceroute –6 google.com  or  tracert –6 google.com | Force the command to use IPv6 packets instead of IPv4. The other parameters can be added to these IPv6 commands and function essentially the same as they do in IPv4. |

Enlarge Table

Note that a trace test might stop before reaching the destination. This usually happens for one of three reasons:

1. The device the trace is attempting to reach is down,
2. the target device is too busy to process lower-priority messages such as UDP or ICMP, or
3. a firewall blocks UDP and ICMP transmissions, especially if it receives several in a short period of time.

If you are trying to trace a route to a host situated behind a firewall, you can try using TCP in traceroute. Otherwise, your efforts might be thwarted. (Because ping uses ICMP transmissions, the same limitations exist for that utility.)

One possible work-around for firewall-imposed limitations on multiple UDP or ICMP probes in a short period of time is to add more of a delay between the probe repetitions. This can be done with the –z parameter followed by the number of seconds (up to 10) for the minimum wait time between probes. This option, like many others, is only available for traceroute, not tracert.

**Note 4-11**

Many Linux distributions, like Ubuntu, do not include the traceroute utility by default. You will have to install traceroute to use it on those systems. You might find in its place a simpler utility called tracepath. The tracepath command does not provide as many options as traceroute. However, it is based on the same principles, and it might be sufficient to save you the time of installing the traceroute package.

A trace cannot detect router configuration problems or predict variations of routes over time. Therefore, a trace is best used on a network with which you are already familiar. The traceroute or tracert utility can help you diagnose network congestion or network failures. You can then use your judgment and experience to compare the actual test results with what you anticipate the results should be.

### tcpdump

The **[tcpdump](javascript://)** utility is a free, command-line packet sniffer that runs on Linux and other UNIX operating systems. You’ve already learned about the protocol analyzer Wireshark where you captured a packet and examined the information provided at various OSI layers, and the [On the Job](javascript://) story at the beginning of this module showed how a network analyzer was used to identify the source of a difficult-to-diagnose problem with some applications. A [**packet sniffer**](javascript://) is very similar and many people use the terms interchangeably. In essence, the difference between a packet sniffer and a protocol analyzer is the level of interpretation and analysis the tool provides for the data captured from the network interface.

Like Wireshark, tcpdump captures traffic that crosses a computer’s network interface. The output can be saved to a file that you can filter or play back. Because of its robust configuration options and straightforward, command-line interface, it’s a popular tool among security professionals and hackers alike. When used on a network device, such as a router or switch, tcpdump can become a very powerful tool indeed.

You must either use the sudo command or log in as root to access tcpdump. To do this, either enter sudo before each tcpdump command, or at the shell prompt, enter sudo su root, which changes you over to the root account. [Table 4-11](javascript://) gives some tcpdump examples.

**Table 4-11**

### tcpdump command options

| **tcpdump command** | **Description** |
| --- | --- |
| tcpdump not port 22 or tcpdump not port 23 | Filter out SSH or Telnet packets, which is helpful when running tcpdump on a remotely accessed network device. |
| tcpdump -n | Instruct the command to not resolve IP addresses to host names. |
| tcpdump -c 50 | Limit the number of captured packets to 50. |
| tcpdump -i any | Listen to all network interfaces on a device. |
| tcpdump -D | List all interfaces available for capture. |
| tcpdump port http | Filter out all traffic except HTTP. |
| tcpdump -w capture.cap | Write the file output to a file named capture.cap. |
| tcpdump -r capture.cap | Read the file capture.cap and output the data in the terminal window. This file can also be read by applications like Wireshark. |

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

## 4-4bSolving Common Network Problems

You can use the tools presented in this module to troubleshoot and solve several common problems on your network. [Table 4-12](javascript://) gives a brief summary of all the command-line utilities you’ve covered so far in this course and how they can help you.

**Table 4-12**

### Command-Line Utilities

| **Command** | **Common uses** |
| --- | --- |
| arp | Provide a way to obtain information from and manipulate a device’s ARP table. |
| dig | Query DNS servers with more advanced options than nslookup. |
| ipconfig, ip, or ifconfig | Provide information about TCP/IP network connections and the ability to manage some of those settings. |
| netstat | Display TCP/IP statistics and details about TCP/IP components and connections on a host. |
| nmap | Detect, identify, and monitor devices on a network. |
| nslookup | Query DNS servers and provide the ability to manage the settings for accessing those servers. |
| ping | Verify connectivity between two nodes on a network. |
| ssh | Establish a secured connection with a remote host for executing commands from the remote device’s CLI. |
| telnet | Establish an unsecured connection with a remote host for executing commands from the remote device’s CLI. |
| tcpdump | Capture traffic that crosses a computer’s network interface. |
| traceroute or tracert | Trace the path from one networked node to another, identifying all intermediate routers between the two nodes. |

Using what you’ve learned, let’s explore a few common network problems and how to solve them.

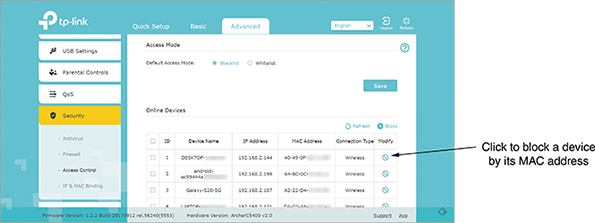
### Duplicate MAC Addresses

Devices on separate networks can have the same MAC address without causing any problems. Even if duplicate MAC addresses need to communicate with each other across separate networks, the fact that MAC addresses exist at layer 2 means the MAC addresses themselves are not transmitted outside of their local network. However, two devices on the same network with the same MAC address is a problem.

Because MAC addresses are assigned statically by the manufacturer, you might wonder how two devices could possibly have the same MAC address. Sometimes manufacturers (by accident or by neglect) reuse the same MAC address for two or more devices. Additionally, a MAC address can be impersonated, which is a security risk called spoofing. On a network where access is limited to certain devices based on their MAC address (see [Figure 4-28](javascript://)), an attacker can spoof an approved device’s MAC address and gain access to the network. This is a relatively easy attack to carry out, which is why MAC address filtering is not considered a reliable way to control access to a network.

**Figure 4-28**

This home router can block, or blacklist, a device based on its MAC address



Enlarge Image

Source: TP-Link Corporation

Most of the time, though, duplicate MAC addresses only cause intermittent connectivity issues for the computers involved in the duplication. Here’s how the situation develops:

1. Step 1

Each computer regularly broadcasts its IP address and the duplicated MAC address so devices on the network can update their ARP tables.

1. Step 2

Those other devices, in response, update their records to point toward one computer, and then the other computer, and then back to the first one, and so on, depending upon the latest transmission they received.

1. Step 3

Sometimes devices will send communications to the correct computer, and sometimes their records will be wrong.

Thankfully, duplicate MAC addresses are a relatively rare problem. It happens most often when managing multiple virtual devices on a large network, and in those cases, it’s typically due to human error. Most switches will detect the problem and produce helpful error messages of some kind. Then it’s a matter of tracking down which virtual devices have the same MAC address and updating each device’s configuration.

**Applying Concepts 4-6**

### Change a MAC Address

It only takes a few, short steps to change a Windows computer’s MAC address. Complete the following steps:

1. 1

Open Network and Sharing Center, and click **Change adapter settings**.

1. 2

Right-click any wired network adapter and click **Properties**. Then click **Configure**.

1. 3

On the Advanced tab, click **Network Address** (on a VM, this field is called Locally Administered Address). Select the **Value** radio button, and then enter a 12-digit value with no hyphens or colons, and click **OK**. Close all windows and restart the computer.

1. 4

Run **ipconfig /all** to confirm the new MAC address is active.

1. 5

To return to the default MAC address, repeat the earlier steps but select **Not Present** on the Advanced tab.

### Hardware Failure

When a router, switch, NIC, or other hardware goes down, your job as a network technician includes identifying the location of the hardware failure. Even on smaller networks, it can be a challenge to determine exactly which device is causing problems. Though you could manually check each device on your network for errors, you might be able to shorten your list with a little detective work first. Here’s how:

1. Use tracert or traceroute (depending on your OS) to track down malfunctioning routers and other devices on larger networks. Because ICMP messages are considered low priority, be sure to run the command multiple times and compare the results before drawing any conclusions.
2. Keep in mind that routers are designed to route traffic to other destinations. You might get more accurate tracert or traceroute feedback on a questionable router if you target a node on the other side of that router rather than aiming for the router itself.
3. As you home in on the troublesome device, use ping to test for network connectivity.

**Remember This…**

* Practice using netstat, tracert, traceroute, tcpdump, and Wireshark.
* Explain how to solve problems related to duplicate MAC addresses or hardware failure.

**Self-Check**

1. You need to determine which device on your network is sending excessive messages to your Ubuntu Server. Which utility will give you this information?

Answer

* 1. traceroute
  2. tcpdump
  3. netstat
  4. arp

1. What protocol must be allowed through a firewall for tracert to work correctly?

Answer

* 1. SSH
  2. NDP
  3. ICMP
  4. TLS

**You’re Ready**

You’re now ready to complete [Project 4-3: Redirect Command Output to a Text File](javascript://), or you can wait until you’ve finished the Review Questions for this module.

**You’re Ready**

You’re now ready to complete [Project 4-4: Repair a Duplicate IP Address](javascript://), or you can wait until you’ve finished the Review Questions for this module.

**You’re Ready**

After you finish the Hand-On Projects, you’re ready to complete the [Module 4 Capstone Projects](javascript://).

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# Module Review

## 4-5a**Module Summary**

### TCP/IP Core Protocols

* Data and instructions, known as the payload, are generated by an application running on the source host. In a process called encapsulation, a transport layer protocol (usually either TCP or UDP) adds a header to the payload. The network layer adds its own header to the passed-down segment or datagram. The packet is passed to the data link layer on the NIC, which encapsulates this data with its own header and trailer, creating a frame. The physical layer places the actual transmission on the network. The receiving host decapsulates the message at each layer in reverse order and then presents the payload to the receiving application.
* Each device is known by the innermost OSI layer header it reads and processes.
* Before TCP transmits data, it ensures that a connection or session is established. TCP also sends a character string called a checksum; TCP on the destination host then generates a similar string. If the two checksums fail to match, the destination host asks the source to retransmit the data. TCP also supports flow control, which is the process of gauging the appropriate rate of transmission based on how quickly the recipient can accept data.
* The TCP header includes several important fields, including Source port, Destination port, Sequence number, Acknowledgment number, TCP header length, several flags, Sliding-window size, and an Urgent pointer. TCP uses the sequence number and acknowledgment number fields to orchestrate a three-way handshake that establishes a session.
* UDP (User Datagram Protocol) is an unreliable, connectionless protocol, which means that UDP does not guarantee delivery of data, and no connection is established by UDP before data is transmitted. The UDP header contains only four fields: Source port, Destination port, Length, and Checksum.
* IP (Internet Protocol) belongs to the network layer of the OSI model and specifies where data should be delivered, identifying the data’s source and destination IP addresses. IP is the protocol that enables TCP/IP to internetwork—that is, to traverse more than one LAN segment and more than one type of network through a router. The first field of an IP header indicates the IP version. Additional information in the header relates to fragmentation, TTL (Time to Live), identification of the encapsulated protocol, and source and destination IP addresses.
* Whereas IP helps direct data to its correct destination, ICMP (Internet Control Message Protocol) is a core network layer protocol that reports on the success or failure of data delivery. It can indicate when part of a network is congested, when data fails to reach its destination, and when data has been discarded because the allotted Time to Live has expired (that is, when the data has traveled its allotted number of hops). ICMP announces these transmission failures to the sender, but it does not correct errors it detects—those functions are left to higher-layer protocols, such as TCP.
* ARP (Address Resolution Protocol) works in conjunction with IPv4 to discover the MAC address of a node on the local network and to maintain a database that maps local IP addresses to MAC addresses. ARP is a layer 2 protocol that works with IPv4 in layer 3. It’s sometimes said to function at layer 2.5 because it touches information (IP addresses and MAC addresses) at both layers. However, it operates only within its local network bound by routers. To view a Windows workstation’s ARP table, open a PowerShell or Command Prompt window and enter the command arp -a.
* The most important data link layer standard, Ethernet, is adaptable, capable of running on a variety of network media, and offers excellent throughput at a reasonable cost. Unlike higher-layer protocols, Ethernet adds both a header and a trailer to the payload it inherits from the layer above it. This creates a frame around the payload. Included in the Ethernet frame is information about the destination and source MAC addresses, the encapsulated protocol, and the FCS (frame check sequence) to ensure the data at the destination exactly matches the data issued from the source.

### Encryption Protocols

* In terms of security, data exists generally in three states: at rest, in use, and in motion. Encryption is the last layer of defense against data theft. Encryption protocols use a mathematical code, called a cipher, to scramble data into a format that can be read only by reversing the cipher—that is, by deciphering, or decrypting, the data.
* To protect data at rest, in use, and in motion, encryption methods are primarily evaluated by three benchmarks: confidentiality, integrity, and availability. These three principles form the standard security model called the CIA (confidentiality, integrity, and availability) triad.
* With private key encryption, data is encrypted using a single key that only the sender and the receiver know. A potential problem with private key encryption is that the sender must somehow share the key with the recipient without it being intercepted.
* With public key encryption, data is encrypted with a private key known only to the user, and it’s decrypted with a mathematically related public key that can be made available through a third-party source, such as a public key server. This ensures data integrity, as the sender’s public key will only work if the data has not been tampered with. Alternatively, data can be encrypted with the public key, and then it can only be decrypted with the matching private key. This ensures data confidentiality, as only the intended recipient (the owner of the keys) can decrypt the data.
* IPsec (Internet Protocol Security) is an encryption protocol suite that defines a set of rules for encryption, authentication, and key management for TCP/IP transmissions. IPsec works at the network layer of the OSI model—it adds security information to the headers of IP packets and encrypts the data payload. Either AH (authentication header) encryption or ESP (Encapsulating Security Payload) encryption may be used. Both types of encryption provide authentication of the IP packet’s data payload through public key techniques. In addition, ESP encrypts the entire IP packet for added security.
* SSL (Secure Sockets Layer) and TLS (Transport Layer Security) are both methods of encrypting TCP/IP transmissions en route between the client and server using public key encryption technology. SSL operates in the application layer. TLS operates in the transport layer and uses slightly different encryption algorithms than SSL, but otherwise is essentially the updated version of SSL.
* HTTP uses TCP port 80, whereas HTTPS (HTTP Secure) uses SSL/TLS encryption and TCP port 443. Other protocols that offer SSL/TLS encrypted alternatives include SMTP TLS, LDAP over SSL, IMAP over SSL, and POP3 over SSL. Each time a client and server establish an SSL/TLS connection, they establish a unique session, or an association between the client and server that is defined by an agreement on a specific set of encryption techniques. The session allows the client and server to continue to exchange data securely as long as the client is still connected to the server.

### Remote Access Protocols

* FTPS (FTP Security or FTP Secure) is an added layer of protection for FTP using SSL/TLS that can encrypt both the control and data channels. SFTP (Secure FTP) is a file-transfer version of SSH that includes encryption and authentication, and it’s sometimes inaccurately called FTP over SSH or SSH FTP. Note that SFTP is an extension of the SSH protocol, not of FTP. TFTP (Trivial FTP) is a simple version of FTP that includes no authentication or security for transferring files and uses UDP at the transport layer (unlike FTP, which relies on TCP at the transport layer). TFTP requires very little memory and is most often used by machines behind the scenes to transfer boot files or configuration files.
* Terminal emulation, also called remote virtual computing, allows a remote client to take over and command a host computer. Examples of terminal emulation software are Telnet, SSH, Remote Desktop, and VNC (Virtual Network Computing).
* Examples of command-line software that can provide terminal emulation include Telnet and SSH. Some GUI-based software examples are Remote Desktop for Windows, join.me, VNC (virtual network computing), and TeamViewer. A host may allow clients a variety of privileges, from merely viewing the screen to running programs and modifying data files on the host’s hard disk.
* SSH allows for password authentication or authentication using public and private keys. First, generate a public key and a private key on your client workstation by running the ssh-keygen command (or by choosing the correct menu options in a graphical SSH program). Then transfer the public key to an authorization file on the host to which you want to connect. When you connect to the host via SSH, the client and host exchange public keys. If both can be authenticated, the connection is completed.
* RDP (Remote Desktop Protocol) is a Microsoft proprietary protocol used by Windows Remote Desktop and Remote Assistance client/server utilities to connect to and control a remote computer. Similarly, VNC (Virtual Network Computing) uses the cross-platform protocol RFB (remote frame buffer) to remotely control a workstation or server. VNC is slower than Remote Desktop and requires more network bandwidth. However, because VNC is open source, many companies have developed their own software using VNC.
* Telnet, SSH, RDP, and VNC all rely on the existing network infrastructure for a network administrator to remotely control the device. Before the devices can be configured, they must already be booted up, and they must already have configuration software installed. This is called in-band management, and it inherently limits troubleshooting capabilities. Out-of-band management, however, relies on a dedicated connection (either wired or wireless) between the network administrator’s computer and each critical network device.
* A VPN (virtual private network) is a virtual connection that remotely accesses resources between a client and a network (client-to-site VPN), two networks (site-to-site VPN), or two hosts over the Internet or other types of networks (host-to-host VPN).
* To ensure a VPN can carry all types of data in a private manner over any kind of connection, special VPN protocols encapsulate higher-layer protocols in a process known as tunneling. Two common approaches to VPN tunneling either require all network traffic to traverse the VPN tunnel or only some of that traffic. While IPsec itself can be used as a tunneling protocol, it often provides encryption for another tunneling protocol. This pairing offers more flexibility and other features that IPsec can’t offer alone.

### Troubleshooting Network Issues

* The netstat utility displays TCP/IP statistics and details about TCP/IP components and connections on a host. The netstat command can be used to obtain information about the port on which a TCP/IP service is running, which network connections are currently established for a client, how many messages have been handled by a network interface since it was activated, and how many data errors have occurred on a particular network interface.
* The Windows tracert utility uses ICMP echo requests to trace the path from one networked node to another, identifying all intermediate hops between the two nodes. Linux, UNIX, and macOS systems use UDP datagrams or, possibly, TCP SYN messages, for their traceroute utility, but the concept is still the same.
* The tcpdump utility is a free, command-line packet sniffer that runs on Linux and other UNIX operating systems. Like Wireshark, tcpdump captures traffic that crosses a computer’s network interface. Because of its robust configuration options and straightforward, command-line interface, it’s a popular tool among security professionals and hackers alike. You must either use the sudo command or log in as root to access tcpdump.
* Devices on separate networks can have the same MAC address without causing any problems. However, two devices on the same network with the same MAC address is a problem. Most of the time duplicate MAC addresses only cause intermittent connectivity issues for the computers involved in the duplication. It happens most often when managing multiple virtual devices on a large network, and in those cases, it’s typically due to human error.
* When a router, switch, NIC, or other hardware goes down, a network technician’s job is to identify the location of the hardware failure. Various tools help to narrow down the possibilities, including using tracert or traceroute to track down malfunctioning routers and other devices on larger networks, targeting a node on the other side of a suspected router problem rather than aiming for the router itself, and using ping to test for network connectivity.

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# Module Review

## 4-5b**Key Terms**

* [**AH (authentication header)**](javascript://)
* [**ARP (Address Resolution Protocol)**](javascript://)
* [**ARP table**](javascript://)
* [**asymmetric encryption**](javascript://)
* [**authentication**](javascript://)
* [**CA (certificate authority)**](javascript://)
* [**checksum**](javascript://)
* [**CIA (confidentiality, integrity, and availability) triad**](javascript://)
* [**clientless VPN**](javascript://)
* [**client-to-site VPN**](javascript://)
* [**collision**](javascript://)
* [**collision domain**](javascript://)
* [**CSMA/CD (Carrier Sense Multiple Access with Collision Detection)**](javascript://)
* [**digital certificate**](javascript://)
* [**DMVPN (Dynamic Multipoint VPN)**](javascript://)
* [**dynamic ARP table entry**](javascript://)
* [**encryption**](javascript://)
* [**ESP (Encapsulating Security Payload)**](javascript://)
* [**Ethernet II**](javascript://)
* [**FTPS (FTP Secure or FTP over SSL)**](javascript://)
* [**full tunnel VPN**](javascript://)
* [**GRE (Generic Routing Encapsulation)**](javascript://)
* [**hop**](javascript://)
* [**in-band management**](javascript://)
* [**IPsec (Internet Protocol Security)**](javascript://)
* [**jumbo frame**](javascript://)
* [**key**](javascript://)
* [**link-layer address**](javascript://)
* [**LLC (logical link control) sublayer**](javascript://)
* [**MAC sublayer**](javascript://)
* [**mGRE (multipoint GRE)**](javascript://)
* [**MTU (maximum transmission unit)**](javascript://)
* [**NDP (Neighbor Discovery Protocol)**](javascript://)
* [**netstat**](javascript://)
* [**out-of-band management**](javascript://)
* [**packet sniffer**](javascript://)
* [**PKI (public-key infrastructure)**](javascript://)
* [**private key encryption**](javascript://)
* [**probe**](javascript://)
* [**protocol analyzer**](javascript://)
* [**public key encryption**](javascript://)
* [**RAS (remote access server)**](javascript://)
* [**RDS (Remote Desktop Services)**](javascript://)
* [**remote access**](javascript://)
* [**remote desktop connection**](javascript://)
* [**remote desktop gateway**](javascript://)
* [**site-to-site VPN**](javascript://)
* [**split tunnel VPN**](javascript://)
* [**spoofing**](javascript://)
* [**static ARP table entry**](javascript://)
* [**symmetric encryption**](javascript://)
* [**tcpdump**](javascript://)
* [**terminal emulator**](javascript://)
* [**TFTP server**](javascript://)
* [**three-way handshake**](javascript://)
* [**traceroute**](javascript://)
* [**tracert**](javascript://)
* [**VDI (Virtual Desktop Infrastructure)**](javascript://)
* [**VNC (Virtual Network Computing)**](javascript://)
* [**VPN (virtual private network)**](javascript://)
* [**VPN headend**](javascript://)

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?#header)

# Module Review

## 4-5c**Review Questions**

1. Which protocol’s header would a layer 4 device read and process?
   1. IP
   2. TCP
   3. ARP
   4. HTTP
2. What field in a TCP segment is used to determine if an arriving data unit exactly matches the data unit sent by the source?
   1. Source port
   2. Acknowledgment number
   3. DiffServ
   4. Checksum
3. At which OSI layer does IP operate?
   1. Application layer
   2. Transport layer
   3. Network layer
   4. Data link layer
4. What is the Internet standard MTU?
   1. 65,535 bytes
   2. 1,522 bytes
   3. 1,500 bytes
   4. 9,198 bytes
5. Which two protocols manage neighbor discovery processes on IPv4 networks?
   1. ICMP and ARP
   2. IPv4 and IPv6
   3. TCP and UDP
   4. NDP and Ethernet
6. You’re getting a duplicate IP address error on your computer and need to figure out what other device on your network is using the IP address 192.168.1.56. What command will show you which MAC address is mapped to that IP address?
   1. telnet 192.168.1.56
   2. tracert 192.168.1.56
   3. arp -a
   4. netstat -n
7. What is one advantage offered by VDI over RDS and VNC?
   1. Offers access to multiple OSs in VMs
   2. Supports remote access to mobile devices
   3. Allows multiple users to sign in at once
   4. Provides open source flexibility
8. Which encryption protocol does GRE use to increase the security of its transmissions?
   1. SSL
   2. SFTP
   3. IPsec
   4. SSH
9. Which encryption benchmark ensures data is not modified after it’s transmitted and before it’s received?
   1. Confidentiality
   2. Integrity
   3. Availability
   4. Symmetric
10. Which remote file access protocol is an extension of SSH?
    1. SFTP
    2. TFTP
    3. FTPS
    4. HTTPS
11. What three characteristics about TCP distinguish it from UDP?
12. What process is used to establish a TCP connection?
13. What is the difference between dynamic ARP table entries and static ARP table entries?
14. Which two fields in an Ethernet frame help synchronize device communications but are not counted toward the frame’s size?
15. Explain the key difference between how symmetric encryption works and how asymmetric encryption works.
16. Which secured tunneling protocol might be able to cross firewalls where IPsec is blocked?
17. When surfing online, you get some strange data on an apparently secure website, and you realize you need to check the legitimacy of the site. What kind of organization issues digital certificates for websites?
18. What tcpdump command will capture data on the eth0 interface and redirect output to a text file named checkme.txt for further analysis?
19. Which terminal emulation protocol is similar to RDP but is open source?
20. Which port must be open for RDP traffic to cross a firewall?

Go to pg.

[**help**](javascript://)

Application Opened

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/fullbook.html?" \l "header)

# Module Review

## 4-5d**Hands-On Projects**

**Note 4-12**

Websites and applications change often. While the instructions given in these projects were accurate at the time of writing, you might need to adjust the steps or options according to later changes.

**Note to Instructors and Students:** A rubric is provided for evaluating student performance on these projects. Please see Appendix D.

**Project 4-1**

### Install and Use WSL (Windows Subsystem for Linux)

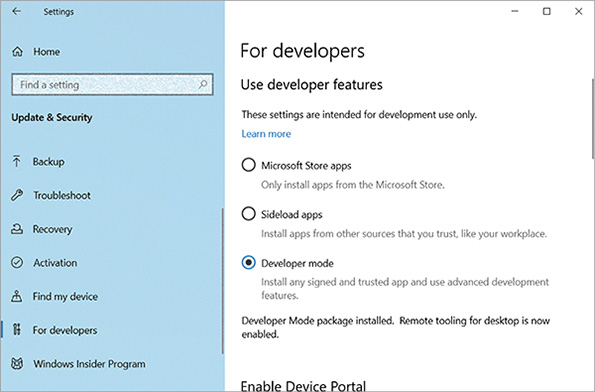
* **Estimated Time:** 45 minutes
* **Objective:** Given a scenario, use the appropriate network software tools and commands. (Obj. 5.3)
* **Resources:**
  + Windows 10 computer with administrative access
  + Internet access
* **Context:** WSL (Windows Subsystem for Linux) is a Linux shell for Windows that allows users to interact with underlying Windows functions and system files. It’s not a VM, and it’s not a fully separate operating system. It runs on any 64-bit Windows 10 system with the Anniversary Update (version 1607) or later. To use it, you must first turn on Developer Mode, and then enable the Windows Subsystem for Linux feature. To enable Windows Subsystem for Linux and install an Ubuntu Terminal on a Windows 10 system, complete the following steps:
  + 1

First, turn on Developer Mode.

* + 1. Open the **Settings** app and click **Update & Security**. In the left pane, scroll down and click **For developers**.
    2. Select **Developer mode**, as shown in [Figure 4-29](javascript://). Click **Yes** to turn on Developer Mode and close the Settings app.

**Figure 4-29**

Turn on Developer Mode from the Settings app



Enlarge Image

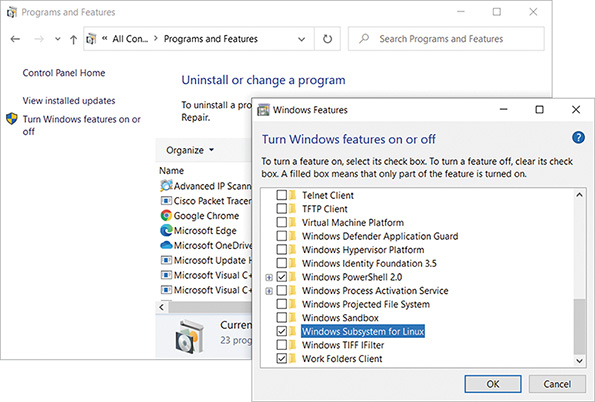
* + 2

Enable Windows Subsystem for Linux.

* + 1. Open **Control Panel** and click **Programs and Features**. In the left pane, click **Turn Windows features on or off**.
    2. Scroll down and click **Windows Subsystem for Linux**, as shown in [Figure 4-30](javascript://). Click **OK**.

**Figure 4-30**

Turn on the Windows Subsystem for Linux feature



Enlarge Image

**Note 4-13**

To open Turn Windows features on or off directly, you can also click Start, begin typing **turn Windows**, then click **Turn Windows features on or off**.

* + 1. Restart the computer when the changes are complete to finish enabling Windows Subsystem for Linux.

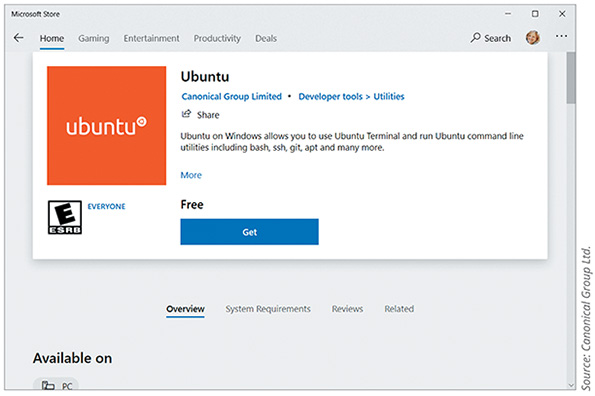
Now that you have enabled Windows Subsystem for Linux, you can install a distribution of Linux designed to run on Windows. To install and run Ubuntu on Windows, do the following:

* + 3

Open the **Microsoft Store** app and search for **Ubuntu**. Install the latest, free Ubuntu on Windows app, as shown in [Figure 4-31](javascript://).

**Figure 4-31**

Install the Ubuntu on Windows app from the Microsoft Store



Enlarge Image

Source: Canonical Group Ltd.

* + 4

After the installation is complete, launch the app. Enter a new UNIX username at the prompt. This username can be different from your Windows username.

* + 5

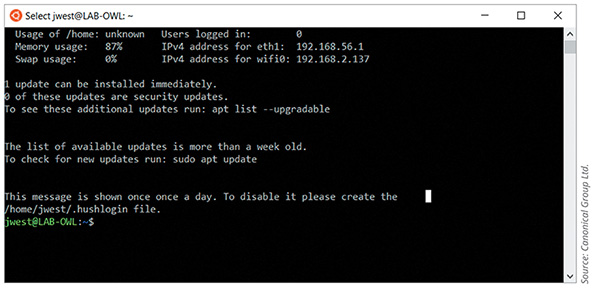
Enter a password at the next prompt. The cursor will not move as you type the password. Re-enter the password at the next prompt. Add this information as a Secure Note in your LastPass vault.

* + 6

After the installation is complete, you’ll see the Ubuntu Terminal, as shown in [Figure 4-32](javascript://). What is the Ubuntu prompt on your computer? Include all symbols in your answer.

**Figure 4-32**

Ubuntu on Windows is installed and provides an Ubuntu Terminal



Enlarge Image

Source: Canonical Group Ltd.

At this point, many of the Linux commands you have become familiar with will work as usual at the Ubuntu shell prompt. The commands interact with the underlying Windows system files, and changes to those files can be monitored through other Windows tools.

* + 7

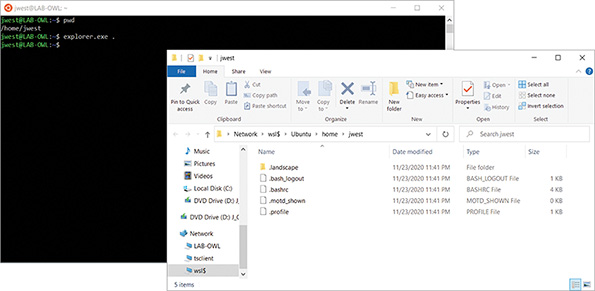
Enter the commend **pwd** to show your current working directory (recall the Linux calls folders directories). What is the current directory?

* + 8

To open a File Explorer window showing this directory, enter the command **explorer.exe.** (Notice the extra space and period after the explorer.exe portion.) [Figure 4-33](javascript://) shows the command entered in the Ubuntu Terminal window and the File Explorer window showing Ubuntu’s home directory.

**Figure 4-33**

You can see Ubuntu’s home directory in File Explorer



Enlarge Image

* + 9

To create a new directory, enter the command **mkdir mydir**. Refresh the File Explorer window. Do you see your new directory listed?

* + 10

To navigate to that directory in Ubuntu, enter the command **cd mydir**. To create a new file in that directory, enter the command **touch myfile.txt**. In File Explorer, open the **mydir** folder. What items do you see listed here?

* + 11

Choose three other Linux commands and practice using them. For each one, **take a screenshot** of the Ubuntu Terminal window showing the command and its output. Submit this visual with your answers to this project’s questions.

* + 12

In your wiki, add a new page titled **Applications:WSL-Ubuntu**. Indicate the module and project number for this installation, the computer you used for this project, a brief description of what you learned, and any other information you might find helpful when using Ubuntu on Windows later.

**Project 4-2**

### Use Remote Desktop

* **Estimated Time:** 45 minutes
* **Objective:** Compare and contrast remote access methods and security implications. (Obj. 4.4)
* **Group Work:** This project can be completed by an individual working alone or, if desired, in cooperation with a team of two or three classmates. In some cases, working as a group could provide beneficial insights when troubleshooting challenging steps and brainstorming solutions. Check with your instructor for details specific to your class.
* **Resources:**
  + Two Windows computers (with administrative access) on the same network
  + These two computers can be both physical, both virtual, or one of each.
  + One of these systems must have Windows 10 Professional, Education, or Enterprise installed, and the other can have any edition of Windows 7, 8, 8.1, or 10.
  + If you’re using a VM as one computer and the VM’s physical host as the second computer, the VM must serve as the RDP host and must have Windows 10 Professional, Education, or Enterprise installed. You might need to create a new Windows VM to meet these requirements if you previously installed Windows 10 Home on your VM. If you do create a new VM, be sure to record credentials in your LastPass vault. The physical computer will be the RDP client.
  + If you’re using a VirtualBox VM as one computer and a physical machine as the second computer, the VM’s network adapter must be configured in the bridged mode. Before starting the VM, open the VM’s **Settings** window, click **Network**, and change the Attached to field to **Bridged Adapter**. Click **OK**. You can now start the VM.
  + Internet access or a Windows ISO file if a new VM is needed
* **Context:** The host or server computer is the remote computer that serves up Remote Desktop to your local client computer. Note that a Windows Home computer cannot serve as an RDP host, although it can connect as a client to another RDP host. To prepare your host computer, you need to get its name and configure the Remote Desktop service. Complete the following steps on a Windows 10 (Professional, Education, or Enterprise) machine:
  + 1

Right-click the **Start** button and click **System**. Under Device specifications, find the device’s name and copy it to a text file using Notepad for future reference in this project. What is the RDP host’s device name?

* + 2

To enable the Remote Desktop service on the host computer, in the left pane of the Settings window, scroll down and click **Remote Desktop**. Click the slider to **Enable Remote Desktop**. In the warning box, click **Confirm**.

* + 3

Make sure you know the sign-in credentials for a user on this system. Users who have administrative privileges are allowed to use Remote Desktop by default, but other users need to be added. If you need to add a user, click **Select users that can remotely access this PC** and follow the directions on-screen.

* + 4

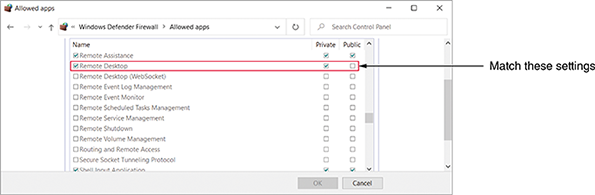
Verify that Windows Firewall is set to allow Remote Desktop activity to this computer. To do this, in the Settings app, search for **firewall**, click **Firewall & network protection** and then click **Allow an app through firewall**.

* + 5

The Allowed apps window appears. Scroll down to Remote Desktop. If the changes don’t match the settings in [Figure 4-34](javascript://), click **Change settings** and make any needed adjustments. Click **OK** to apply any changes. Close the Windows Defender Firewall and Settings windows. You will learn more about Windows Defender Firewall later.

**Figure 4-34**

Allow Remote Desktop communication on a private network through Windows Defender Firewall



Enlarge Image

* You are now ready to test Remote Desktop by accessing the host computer (physical or virtual) from another computer (physical or virtual) on your local network. Note that any edition of Windows 7, 8.1, or 10 can serve as a client computer (the computer viewing the host computer’s desktop) for a Remote Desktop connection. The following steps are written specifically for Windows 10.
* Follow these steps on the client computer (any edition of Windows 7, 8, 8.1, or 10) to create a Remote Desktop connection to the host computer:
  + 6

First, confirm the two computers can communicate on the network. In a PowerShell or Command Prompt window, ping the RDP host computer from the RDP client computer. If the ping works, continue to [Step 7](javascript://). If it does not work, you’ll need to do some troubleshooting. Here are some possible solutions:

* + 1. Make sure both computers are connected to the same network and subnet. In most cases, the first three octets of each computer’s IP address should be identical to each other.
    2. Make sure both computers can communicate successfully with other resources on the network. For example, try pinging the default gateway.
    3. Make sure both computers are connected to the network in Private mode. To check this, click the network connection’s icon on the right side of the taskbar near the clock, and then click **Network & Internet settings**. The connection should be labeled “Private network” on both computers. If it is not, click **Change connection properties** and choose the **Private** option. Note that if you’re working on a physical machine connected to a Hyper-V virtual switch, the network status will show vEthernet and you will not have the option to change between Public and Private network modes because the computer should already be in Private network mode.
    4. If the ping still won’t work, enabling File and Printer Sharing on Windows Defender Firewall sometimes solves the problem. To do this, open Network and Sharing Center and click **Change advanced sharing settings**. Select **Turn on file and printer sharing** for private networks on both computers.
  + 7

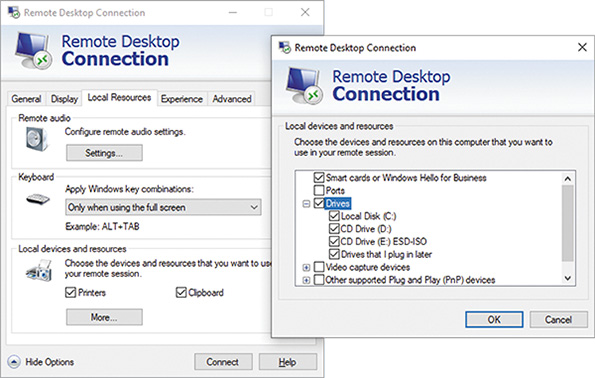
Press **Win+R**, type **mstsc** in the search box, and press **Enter**. This is easier to remember if you know that Remote Desktop Services was formerly called Microsoft Terminal Services; mstsc (Microsoft Terminal Services Client) is the client portion. Alternately, you can click **Start**, scroll down and click **Windows Accessories**, and then click **Remote Desktop Connection**.

* + 8

Enter the host name of the computer to which you want to connect. To be able to transfer files from one computer to the other, click **Show Options** and then click the **Local Resources** tab, as shown in the left side of [Figure 4-35](javascript://). Under Local devices and resources, click **More**. The dialog box on the right side of [Figure 4-35](javascript://) appears.

**Figure 4-35**

Allow drives to be shared using the Remote Desktop Connection



Enlarge Image

* + 9

Check **Drives**, click **OK**, and then click **Connect** to make the connection. If a warning box appears, check the box for Don’t ask me again for connections to this computer and then click **Connect** again.

* + 10

Enter a password for the remote computer. If you need to sign in with a different account than the one you’re using on the client computer, click **More choices**, click **Use a different account**, and enter the account credentials. If you’re using a client computer that you own, you can check the box for Remember me, and then click **OK**. If a warning box appears saying the identity of the remote computer cannot be verified, you can check the box for Don’t ask me again for connections to this computer. Click **Yes** to continue with the connection.

**Note 4-14**

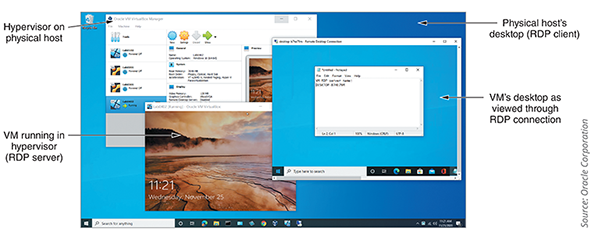
Even though Windows normally allows more than one user to be logged on at the same time, Remote Desktop does not. When a Remote Desktop session is opened, all local users on the host computer are logged off.

* + 11

The desktop of the remote computer appears in a maximized window that covers your entire screen. Float your cursor at the top of your screen to find the RDP controls. From here, you can reduce the size of the RDP window so you can see both your local computer’s desktop and the remote computer’s desktop, as shown in [Figure 4-36](javascript://). When you click inside the RDP window, you can work with the remote computer just as if you were sitting in front of it, except the response time will be slower. To move files back and forth between computers, use File Explorer on the remote computer. Files on your local computer and on the remote computer will appear in File Explorer on the remote computer’s screen in the This PC group.

**Figure 4-36**

Physical host (RDP client) connected by RDP to VM (RDP server)



Enlarge Image

Source: Oracle Corporation

* + 12

Position File Explorer on the remote computer’s desktop so that you can see both the server’s and the client’s hard drives listed in the left pane. **Take a screenshot**; submit this visual with your answers to this project’s questions.

* + 13

To close the connection to the remote computer, shut down or sign out of the remote computer or close the Remote Desktop Connection window.

* + 14

In your wiki, add a new page titled **Applications:RDP**. Indicate the module and project number for this activity, the computers you used for this project, a brief description of what you learned, and any other information you might find helpful when using RDP later.

* + 15

If you created a new VM for this project, add the new VM’s information to your VMclients page in your wiki. Include the module number, hypervisor used, VM computer name, and VM operating system. Also note any additional information that you might find helpful when you return to this VM in the future.

**Project 4-3**

### Redirect Command Output to a Text File

* **Estimated Time:** 15 minutes
* **Objective:** Given a scenario, use the appropriate network software tools and commands. (Obj. 5.3)
* **Resources:**
  + Internet access
* **Context:** Sometimes when you’re using a command such as tcpdump, the sheer volume of output can be daunting to work with. There’s no way to search through the output for specific information, and you can only expand the PowerShell or Command Prompt window so far. One solution to this problem is to redirect the command output to a text file where you can search the text, copy and paste text, and save the output for future reference. To accomplish this feat, you’ll need to add a redirection operator to the command whose output you want to export to a text file. Complete the following steps:
  + 1

First, try this simple command in PowerShell or Command Prompt:

The following command is executed in a Power Shell window. i p config right single angle bracket i p config test dot t x t. When this command runs, the i p config command runs and the output of the command is sent to a text file with the name i p config test dot t x t.

This runs the ipconfig command and redirects the output to a text file named ipconfigtest.txt. By default, the file is saved to the current default folder, for example, C:\Users\JillWest. Use File Explorer to find the file. **Take a screenshot** showing the file and its file path; submit this visual with your answers to this project’s questions.

* + 2

To specify the location of the file when you create it, add the path to the file in the command line. For example, to save the file to the desktop, use the following command (substitute the correct file path to your desktop). What command and file path did you enter?

The i p config command is executed and the output of the command needs to be saved in a specific location. The command to do this follows. i p config right single angle bracket C colon back slash Users back slash Username backslash Desktop back slash i p config test dot t x t.

**Note 4-15**

If you’re not sure what the file path is to your Desktop, you can find it in File Explorer. In the navigation pane on the left, right-click the **Desktop** link and click **Properties**. The file path is shown in the Location field. Note that your desktop might be showing your OneDrive Desktop, which would be located at the following path:

C:\Users\Username\OneDrive\Desktop

In this case, you might not be able to send command output to your OneDrive desktop. Save your files for this project to a folder on your hard drive instead.

* + 3

If you already have a file on the desktop by that name, the file will be overwritten with the new data. What if you would rather append data to an existing file? In this case, use the >> operator. Enter this command (substitute the correct file path to your desktop):

The i p config command is executed and the output of the command needs to be appended to a file that already exists in a specific location. The command to do this follows. i p config right single angle bracket right single angle bracket C colon back slash Users back slash Username backslash Desktop back slash i p config test dot t x t.

Now the new output will appear at the end of the existing file, and all the data is preserved within this single file. This option is useful when collecting data from repeated tests or from multiple computers, where you want all the data to converge into a single file for future analysis.

**Note 4-16**

When reusing an earlier command or portions of an earlier command, you can press the up arrow on your keyboard to recall earlier commands. Then use the side arrows to place your cursor to make edits.

* + 4

Where do command parameters fit when redirecting output? Let’s use the netstat command to show the IP address and port of each TCP and UDP connection on the computer. In the following command, substitute the correct file path to your desktop to output the data to a new file. Notice that any parameters you want to use should be inserted after the command itself and before the redirection operator.

The net stat command to display the I P address and port of each T C P and U D P connection on the computer is listed further. The command also redirects the output to a text file. The command is as follows. net stat hyphen a n right single angle bracket C colon back slash Users back slash Username backslash Desktop back slash connections dot t x t.

Open the file you just created. How many TCP ports are in the ESTABLISHED state?

* + 5

Open your browser and visit two or three websites in different tabs. With your browser still open, run the netstat command again and append the new data to your existing file. What command did you run? How many TCP ports are in the ESTABLISHED state now?

* + 6

You can include a space in the filename by putting quotation marks around the entire filename and location. Enter the following command:

When redirecting the output of a command to a file, if the file name contains spaces then the path of the file needs to be included in double quotes for the command to work properly. A ping command that directs the output to a text file with a name that has spaces is listed further. ping 8 dot 8 dot 8 dot 8, right single angle bracket, left double quotation mark, C colon, Users, back slash, Username, back slash, Desktop, back slash, find google dot t x t, right double quotation mark.

* + 7

**Take a screenshot** of your File Explorer window showing the files you created in this project from [Step 2](javascript://), [Step 3](javascript://), [Step 4](javascript://), [Step 5](javascript://), and [Step 6](javascript://); submit this visual with your answers to this project’s questions.

**Project 4-4**

### Repair a Duplicate IP Address

* **Estimated Time:** 15 minutes (+5 minutes for group work, if assigned)
* **Objective:** Given a scenario, troubleshoot general networking issues. (Obj. 5.5)
* **Group Work:** This project includes enhancements when assigned as a group project.
* **Resources:**
  + Windows 10 computer with administrative access
* **Context:** ARP can be a valuable troubleshooting tool for discovering the identity of a machine whose IP address you know, or for identifying two machines assigned the same IP address. Let’s see what happens when two devices on the network are assigned the same IP address. First you change the IP address of a local Windows machine to match an IP address of another device—in other words, you “break” the computer. Then you see how the arp command helps you diagnose the problem. Complete the following steps:
  + 1

Open a PowerShell or Command Prompt window and enter the command **arp –a**. Your device’s IP address is listed as the Interface address at the top of the list. Write down this IP address and the address of another device on the network.

* + 2

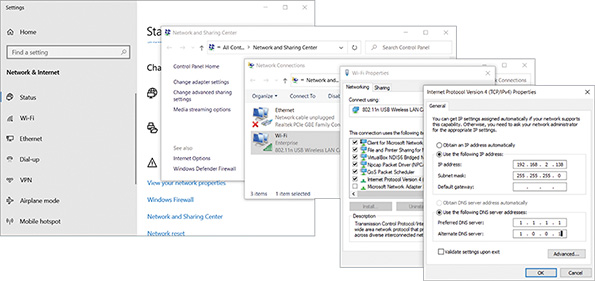
Open the Network and Sharing Center, click **Change adapter settings**, right-click the active network connection, and click **Properties**. If necessary, enter an administrator password in the UAC box and click **Yes**.

* + 3

Select Internet Protocol Version 4 (TCP/IPv4) and click **Properties**. Set the IP address to match the other device’s IP address that you wrote down in [Step 1](javascript://). The system automatically assigns the Subnet mask, as shown in [Figure 4-37](javascript://). Also assign the Cloudflare public DNS servers: 1.1.1.1 and 1.0.0.1. Click **OK** and then click **Close**.

**Figure 4-37**

The subnet mask is assigned automatically



Enlarge Image

* + 4

Back at the CLI, enter **ipconfig /all**.

* + 5

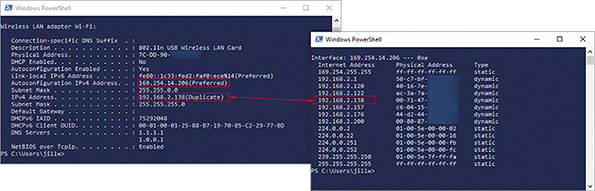
Find the appropriate network connection and identify your computer’s current IPv4 address. Answer the following questions:

* + 1. Has your computer identified the duplicate IP address problem yet? How do you know?
    2. Your computer might also have autoconfigured another IP address. If so, what address did your computer resort to?
    3. **Take a screenshot** of your TCP/IP configuration information; submit this visual with your answers to this project’s questions.
  + 6

In the window on the left side of [Figure 4-38](javascript://), you can see a warning that the IP address is a duplicate. The system also shows a preferred IPv4 address of 169.254.143.79, which is an APIPA address. How can you tell this is an APIPA address?

**Figure 4-38**

The computer automatically configured an APIPA address



Enlarge Image

* + 7

To confirm the duplication of IP addresses, enter the command **arp –a**. You can see in [Figure 4-38](javascript://) that the local computer’s IPv4 address listed on the left matches another IP address in the ARP table on the right, and again you see the APIPA address assigned to the local interface. What are two ways to solve this problem?

* + 8

**For group assignments:** Run the **arp -a** command in your CLI window and answer the following questions:

* + 1. How many other APIPA addresses appear in the output?
    2. Which ones belong to your group members?
    3. How many digits in these APIPA addresses are the same for all group members?
  + 9

Open the Internet Protocol Version 4 (TCP/IPv4) Properties dialog box again and select the options **Obtain an IP address automatically** and **Obtain DNS server address automatically** and then click **OK.** Close all active windows except your CLI.

* + 10

Run the **ipconfig** command or the **arp –a** command to confirm that a unique IP address has been assigned to your local device’s active network interface. What is the new IP address?

* + 11

Close the PowerShell or Command Prompt window.

Go to pg.

[**help**](javascript://)