CHAPTER

5

The TCP/IP Protocol Suite

I dwell in Possibility.
Emily Dickinson

TCP/IP is the name that refers to the group of protocols that it encompasses. This group of protocols is known as the *TCP/IP protocol suite*. It's called TCP/IP because of the two main protocols that are part of the group: TCP and IP. The TCP/IP protocol suite is also known as the *Internet protocol suite*, as TCP/IP is pretty much the backbone of the Internet (and the majority of all networks out there).

There are many good books that cover the TCP/IP protocol suite. Some of these are multivolume, so that might give you an idea of the amount of information that is covered in the standard. TCP/IP can be considered the most widely used standard of the Internet, much as Ethernet is the dominant LAN standard. In addition to multiple standards, TCP/IP also includes any applications, tools, and transmission media used in the network to pass datagrams. As a matter of fact, RFC 1180, ''A TCP/IP Tutorial,'' states that the term *internet technology* is more appropriate than *TCP/IP* when defining the purpose of the standard.

As we discussed in Chapter 1, ''Introduction to Networking,'' the processes and standards contained in the TCP/IP protocol suite are mapped to one of four layers.[1] These layers are based on the four-layer model of DARPA. Every layer within the TCP/IP reference model is cross-referenced to the seven-layer OSI reference model.

The TCP/IP protocol suite allows data communication to take place. No matter what the node is, who it was made by, which operating system software

[1] Or five layers, depending on what school of thought one follows.

is running, and where the node is located, TCP/IP makes it work. TCP/IP has kept up with the tremendous growth that the Internet (as well as networks in general) has experienced. The possibilities seem endless and may very well be. The quote we selected for this chapter really is appropriate for the TCP/IP protocol suite because anyone involved with any facet of the TCP/IP protocol suite should always dwell in the possibilities.

This chapter covers the more well-known protocols and functions that make up TCP/IP. What do these technologies and standards do? What layer of the TCP/IP reference model does each fall into and why? What are the differences among IPv4, IPv6, and IPng? These are just a few questions that will be answered in the pages to come.

5.1 The TCP/IP Layers

Developers of networking protocols adhere to a layered approach. Each layer is responsible for a different portion of the data communication that is occurring at any time. There are many protocols that are part of the TCP/IP protocol suite. Each protocol functions within a layer of the TCP/IP model, depending on its function. Figure 5-1 shows an example of the TCP/IP model, how it corresponds to the OSI model, and some of the more well-known protocols that are served at each layer.

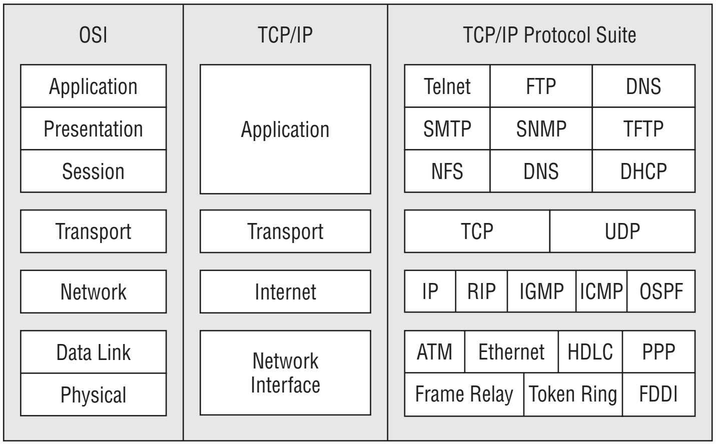

Figure 5-1 TCP/IP reference model layering

The layers in the TCP/IP reference model roughly correspond to one or more layers of the OSI reference model. Protocols of the upper layers can focus on the layer they are a member of, without concerning themselves with the functions performed by the lower levels. This is huge during the development of the protocol, as it enables developers to focus on the development at each layer, rather than worrying about an all-encompassing standard. The layers of the TCP/IP reference model and their responsibilities are as follows:

- **Network Interface layer** — The Network Interface layer corresponds to the Physical and Data Link layers of the OSI reference model. This layer is also often referred to as the *Link* layer or the *Data Link* layer. The Network Interface layer is responsible for the device drivers and hardware interfaces that connect a node to the transmission media.

- **Internet layer** — The Internet layer corresponds to the Network layer of the OSI reference model. This layer is also known as the *Network layer*. The Internet layer is responsible for the delivery of packets through a network. All routing protocols (RIP, OSPF, IP, etc.) are members of this layer. Nodes that perform functions at this layer are responsible for receiving a datagram, determining where to send it to,[2] and then forwarding it toward the destination. When a node receives a datagram that is destined for the node, this layer is responsible for determining the forwarding method for information in the packet. Finally, this layer contains protocols that will send and receive error messages and control messages as required.

RANDOM BONUS DEFINITION

uplink port — Any switch port that is designed to connect to a backbone switch or network.

- **Transport layer** — The Transport layer corresponds to the Transport layer of the OSI reference model. Two primary protocols operate at this layer: Transmission Control Protocol (TCP), and the User Datagram Protocol (UDP). This layer serves the Application layer and is responsible for data flow between two or more nodes within a network.
- **Application layer** — The Application layer corresponds to the Application, Presentation, and Session layers of the OSI reference model. Users initiate a process that will use an application to access network services. Applications work with protocols at the Transport layer in order to pass data in the form needed by the transport protocol chosen. On the receiving end, the data is received by the lower layers and passed up to the application for processing for the destination end user. This layer concerns itself with the details of the application and its process, and not so much about the movement of data. This is what separates this upper layer from the lower three layers.

[2] Based on the IP address that is assigned to the destination network or node.

The design of the TCP/IP model was based on the original Department of Defense network model. The act of layering network protocols is known as *protocol layering*. Protocol layering ensures that data sent by one layer on the source side is the same data received at that layer on the destination side. This layered principle allows focus to remain on the functions of protocols at the layer and ensure that the data matches on each end.

Most applications will use the client/server method of communication. One of the host nodes will act as the server, and the other as a client. Each layer will use a protocol or a group of protocols to transfer readable data from the source layer to the peer layer on the destination side.

Figure 5-2 shows an example of which protocols would be involved to transfer an e-mail message from a source to a destination.

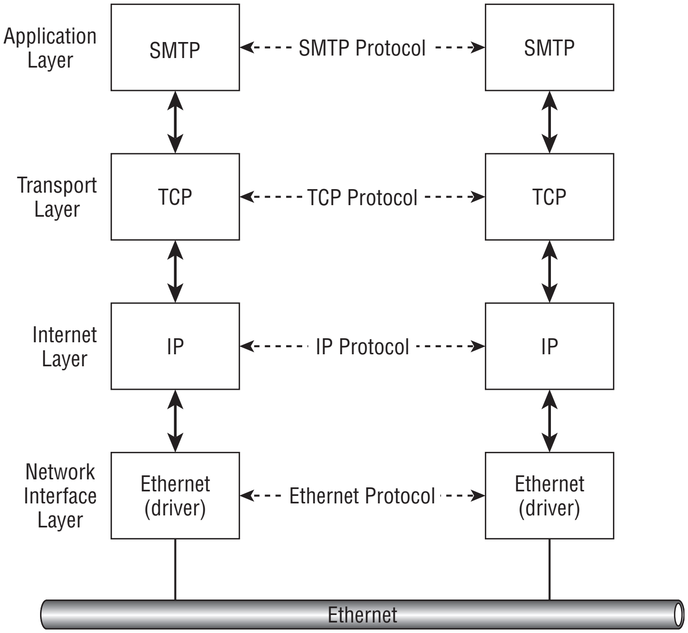

Figure 5-2 TCP/IP layering in action

As you can see, a user on one side of a communication session initiates an e-mail to be sent to the user on the destination side of the session. The Application layer protocol that is used in this process is the *Simple Mail Transfer Protocol (SMTP)*. SMTP will use TCP as the Transport layer protocol, IP as the Internet layer protocol, and then use the Ethernet interfaces at the Network Interface layer to send the data to the media for transport to the other end. This works exactly the same way when there are multiple networks in the mix (see Figure 5-3).

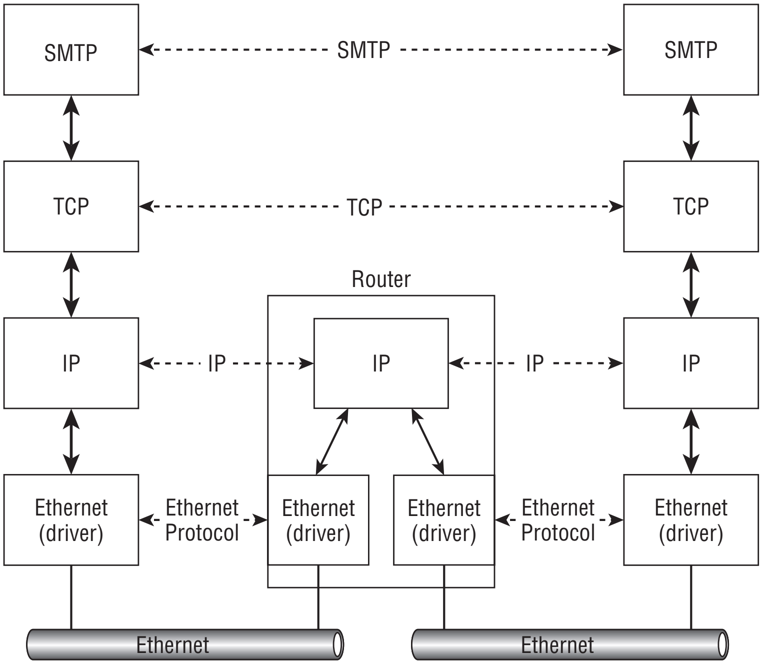

Figure 5-3 TCP/IP layering in multiple networks

In this example, a router is connecting two different networks.[3] Notice that the layers on each end, even though they are not local, are still able to recognize information from their respective peers, as though they are on the same segment. There you have it. That is how the layered model works. The next section discusses many of the protocols that make up the TCP/IP protocol suite.

5.2 Popular TCP/IP Protocols

Now that you know the principles of protocol layering and how it relates to the TCP/IP protocol suite, it's time to discuss the various protocols that operate at each layer. There are many more protocols that are part of the TCP/IP protocol suite. This section covers some of the more widely known (and used) protocols in use in many networks today.

[3] That's the really nice thing about a router. It does not care what type of network it connects to. It can be Token Ring, Ethernet, or many others. The layers don't realize any of this as long as they can talk to their peer.

5.2.1 The Application Layer

A lot of applications are supported by nodes that run TCP/IP. Many of these are commonly included with the operating system software running on the node. If they are not built into an operating system, these applications can readily be found on the Internet, often free of charge. The Application layer is not concerned with the movement of data from one point to another on a network. Its only concern is the details of the application to ensure that what goes out is what is interpreted on the other end. The following protocols are discussed in this section:

- Domain Name System
- Simple Network Management Protocol
- File Transfer Protocol
- Trivial File Transfer Protocol
- Simple Mail Transfer Protocol
- Network File System
- Telecommunications Network Protocol
- Secure Shell

5.2.1.1 Domain Name System

A domain name is simply the name assigned to a node on a network. It is also the name that is assigned as a host name for a given URL on the Internet. For example, if you want to go to the Cable News Network (CNN) website, you would open a web browser application (for example, Firefox, Internet Explorer, etc.) and initiate an HTTP session for the domain name that is assigned to CNN:[4]

```
http://www.cnn.com
```

In the example, `cnn.com` is the domain name that you want to reach because you know that is the domain name for the CNN website. So, why is DNS important? Well, instead of a direct answer to that question, let's answer it this way: What is the IP address for the CNN website? If you know

ACRONYM ALERT

AFP — AppleTalk Filing Protocol

[4] This example was probably too simple, so don't get fooled into thinking that any website you want to go to will have a domain name that matches the site. It depends whether that domain name is owned by someone else and, if it is, whether the owner is willing to sell the domain name. During the initial Internet boom, a lot of people had the foresight to buy popular domain names and later sold them for a lot of money.

that one, you really are doing well, but most likely you do not know the CNN website's IP address. If you have access to a computer that supports TCP/IP, you can find out what the address is. Open up a command-line session and initiate a `ping` to the domain name, and you will be able to see the IP address assigned to the domain name. Here is a `ping` that was run to the `cnn.com` domain name and the IP address that was returned:

```
C:\>ping cnn.com

Pinging cnn.com [64.236.16.20] with 32 bytes of data:

Reply from 64.236.16.20: bytes=32 time=88ms TTL=51
Reply from 64.236.16.20: bytes=32 time=88ms TTL=51
Reply from 64.236.16.20: bytes=32 time=87ms TTL=51
Reply from 64.236.16.20: bytes=32 time=87ms TTL=51

Ping statistics for 64.236.16.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 87ms, Maximum =  88ms, Average =  87ms
```

As you can see in the example, the IP address assigned to the CNN website is 64.236.16.20. Once you know the IP address, you can put that number where you would normally enter the URL in your web browser, and it should bring up the site.

The need for DNS is simple. Humans speak in words, whereas computers speak in numbers. Bits and bytes are all the computers understand. This is why a node has to be assigned a number.[5] Sure, humans can learn numbers and use them as well, but it would probably take a lot of conditioning to remember all the numbers in IP addresses that are assigned to nodes in networks worldwide.[6]

DNS is a database that maps host names to IP addresses. The database is referred to as a *distributed database*, as DNS information is distributed among several servers. Each server will maintain the DNS information that is assigned the server to serve to clients within its own network. DNS uses the client/server model, and the protocol itself provides the facility for the servers to share this information with authorized clients.

RANDOM BONUS DEFINITION

store-and-forward — A mode of switch operation where frames are completely received before they are forwarded onto any of the output ports of the device.

[5] Remember back when we couldn't send an e-mail to Brother Joel?

[6] Are you kidding? Jim has a hard enough time just remembering how old he is.

DNS names are organized hierarchically,[7] with an unnamed root at the top, then what are known as *top-level domain* (TLD) names next, followed by *second-level domain*, and, finally, one or more subdomains. The names assigned to nodes in the DNS hierarchical tree are often referred to as *labels*. This organized hierarchy is known as the *DNS namespace*. The DNS namespace sets the rules for how the labels are organized in the domain name. Figure 5-4 shows an example of the DNS namespace.

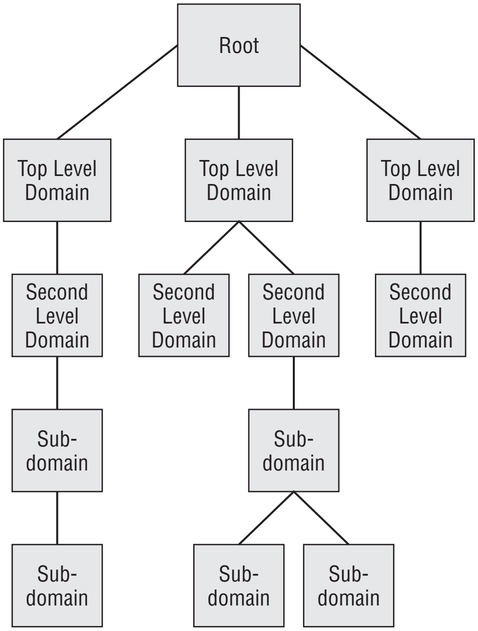

Figure 5-4 DNS namespace hierarchy

POP QUIZ

The Internet layer is also known as the ______________ layer.

The DNS namespace hierarchy requires a different administrator on each level. This ensures that the administration of a particular branch in the DNS tree does not become too cumbersome. At each level of the namespace, there is an administrative authority that provides updates to the database. The delegation of authority should ensure that no level of the namespace becomes too hard to manage.

[7] There is that word again.

Authorities at each level must ensure the DNS server is updated as required. Whenever there is a new node added into the network, the authority adds this to the database. Any removed nodes are required to be updated as well. Not keeping up with these can cause real headaches to end users as well as additional traffic on the network. DNS servers are normally installed in a redundant fashion. Updates are made to the primary server and then are synchronized with the secondary server.[8] This ensures there is not a complete failure of DNS services should the primary server fail.

So, let's see this in action, shall we? We are going to assume there is a company that sells widgets and has decided to use DNS resolution so that end users don't have to remember all of the IP addresses they have to access. DNS name syntax for this company could be:

```
widgets.co
```

In this example, `co` is the top-level domain name, and `widgets` is the second-level domain name. Notice that in between `widgets` and `co` is a period (.), which is called a *dot*. The DNS name `widgets.co` would be pronounced *widjits-dot-see-oh*. Pay attention to the dot that separates the levels within the domain name structure. In any name, the dot separates the levels. You can quickly identify the TLD when you run out of dots.

ACRONYM ALERT

μs — Microseconds

Now, let's assume there is an additional subdomain level, and an authority has been assigned to assign names to nodes within the particular department (Payroll, Production, Planning, and Sales) nodes. The namespace would be updated to reflect this (see Figure 5-5), and the name syntax for each could be[9] as follows:

```
payroll.widgets.co
production.widgets.co
planning.widgets.co
sales.widgets.co
```

The Internet Assigned Numbers Authority (IANA) is responsible for maintaining the DNS *root zone* and is the authority for domain names, IP addresses, and other parameters as well as appointing the authorities that sponsor them.

[8] The synchronization is handled by the secondary server. The secondary server will query the primary periodically to see if there are any updates and, if so, will perform the update to its record.

[9] The authority for the level can assign almost anything that he wants. Normally the name would reflect some identification that reflects the users it serves. The name must be 63 characters or less; other than that, the sky is the limit.

Sometimes the top-level domain names are specific for a particular group or organization. For instance, the top-level domain name for the country of France is `.fr`.[10]

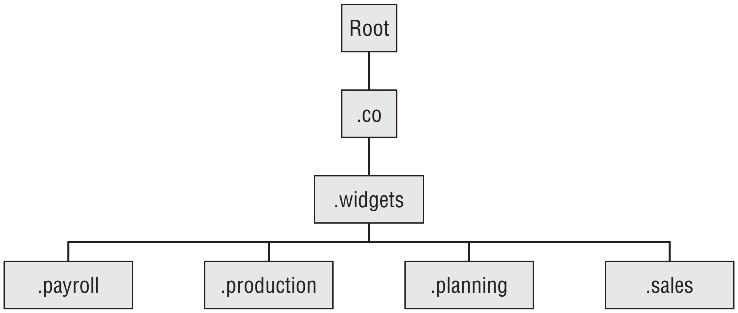

Figure 5-5 An example of the hierarchical tree structure for the widgets.co domain

Sometimes the top-level domain is not really assigned to a particular purpose and therefore is generic in nature. These types of domain names are called *generic top-level domains* (gTLD). Some of the more well-known gTLDs are

- **.biz** — restricted for use by businesses
- **.com** — intended for use by commercial organizations
- **.edu** — postsecondary educational institutions
- **.gov** — restricted for use by the United States federal, state, and local governments.
- **.jobs** — for sites related to employment
- **.mil** — the United States Military
- **.net** — miscellaneous[11]
- **.org** — miscellaneous organizations

5.2.1.2 Simple Network Management Protocol

Today's networks are no longer the shared media environments they once were. As you learned in Chapter 3, ''Network Hardware and Transmission Media,'' a lot of different nodes are deployed in the networks of today. More often than not, there is traffic sharing between nodes and multiple protocols that regulate the flow of data in the network. All this growth requires a way to keep track of what is going on within the network.

[10] Which is basically the country code.

[11] This domain was originally intended for large network infrastructure support centers.

RANDOM BONUS DEFINITION

protocol — A set of algorithms, communication formats, and processes used in the process of data transmission in a network.

Determining traffic patterns to ensure that the network keeps up with end-user demands is not an option; it is a necessity if the network is to live to its full potential. Having the ability to monitor the network[12] for any problems that may occur and getting notification when a problem has arisen is just as (if not more so) important.

Once again, the technology opened up for the development of a protocol that would do these things. That protocol is the Simple Network Management Protocol (SNMP). SNMP is a protocol that runs between an SNMP *manager* and an SNMP *client*, also known as an SNMP *managed system*, for the purpose of sharing management information pertaining to the managed system. Software that runs on the managed system used to communicate system information with the SNMP manager is known as the *SNMP agent*. The information that is shared is determined by the information (known as *managed objects*) set in the management information base (MIB).[13]

Communication between an SNMP manager and an SNMP agent is handled in two directions. The SNMP manager can query the SNMP agent for system information, or the SNMP agent can report information to the SNMP manager. There are five Protocol Data Unit (PDU) types that are exchanged between an SNMP manager and an SNMP agent.[14] These are the GetRequest, GetNextRequest, SetRequest, GetResponse, and Trap. The GetRequest, GetNextRequest, and SetRequest are all PDUs that are sent from the SNMP manager to the SNMP agent. The GetResponse and Trap are sent from the SNMP agent to the SNMP manager (see Figure 5-6).

We discuss these in more detail in Sections 5.2.1.2.1 and 5.2.1.2.3.

5.2.1.2.1 SNMP Managers

The SNMP manager is a workstation that is running SNMP manager software. In some environments, the SNMP manager function is shared by more than one manager, so the resources of one device are not completely consumed trying to monitor the nodes in its charge. System failover is another reason why you may want to have multiple managers in your network.

[12] In a proactive manner.

[13] You will often hear people refer to the management information base as ''the MIBs.''

[14] An easy way to remember who is responsible to send what message type is to remember that the *requests* are sent by the SNMP manager to the SNMP agent, *requesting* information. That leaves only the SNMP response, which are the responses by the SNMP agent to requests that were sent by the SNMP manager, and a trap, which is notification of a problem.

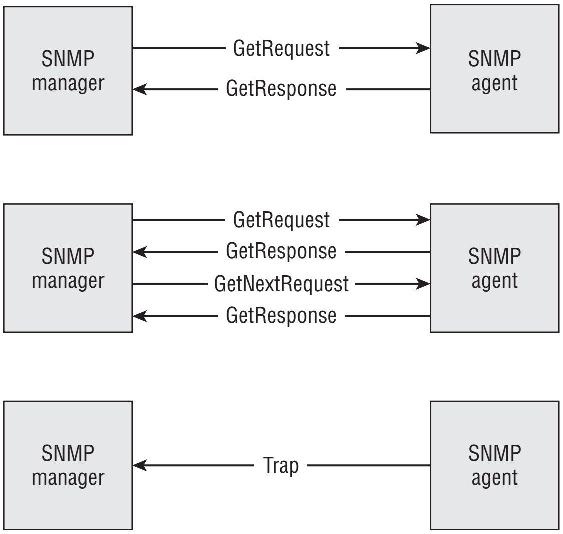

Figure 5-6 An example of SNMP's five PDUs in action

SNMP managers normally output audible alarms and also color-coded reporting in real time. SNMP managers enable you set the protocols and nodes that you want to keep an eye on.

Information that is sent from the SNMP manager to the SNMP agents can be one of three message types:

- **GetRequest** — This message type is a request by the SNMP manager for information pertaining to a variable within a particular managed object.
- **GetNextRequest** — This message type is used to retrieve information that is contained in subsequent requests for information pertaining to a managed object. This helps speed up the retrieval process as the SNMP manager does not have to send a GetRequest for each variable needed.
- **SetRequest** — This message type is used by the SNMP manager to make a change to a variable within a managed object.

5.2.1.2.2 SNMP Managed Devices

An SNMP managed device is any network node that has SNMP agent software running on it for the purposes of network management.

ACRONYM ALERT

UTP — Unshielded twisted pair

5.2.1.2.3 SNMP Agents

The SNMP agent is the software that runs on the SNMP managed device. This software is what allows the managed device to release system information to the SNMP manager. The information to be monitored is set by the SNMP manager and is known as the *managed objects*. Some of the information that can be gathered is port failure, traffic patterns, network unreachable, protocol failures, and many other things.

Information that is sent from the SNMP agent to the SNMP manager can be one of two message types:

- **GetResponse** — This message type is a response to the requests that are sent by the SNMP manager. This can be anything from a value of a variable for a managed object to an error response (for example, if there is no value or if the SNMP agent does not recognize the managed object that the SNMP manager is requesting information about).
- **Trap** — This message type is used by the SNMP agent to report a change of state for a managed object, as well as reporting errors. Some examples of errors that may be reported by the SNMP agent include
  - Link up — The link is up and operational.
  - Link down — The link is down.
  - Cold start — To start a node from the beginning (i.e., a reboot).
  - Warm start — To resume from where a process had left off.
  - OSPF neighbor state changes — In IP routing, the process of learning OSPF topology changes.
  - Authentication failures — Data that is received that cannot be authenticated or verified.
  - Hardware failures — The issue is caused by a problem with hardware.
  - Traffic bursts — The transfer of large amounts of data, without interruption, to a destination node.

5.2.1.2.4 Management Information Base

A management information base (MIB) is a database that contains manageable objects and variables of these objects pertaining to a network node, for the purpose of node management within a network. SNMP itself is not able to define details for the information it retrieves; that is what a MIB is there for.

The reason to keep MIBs and SNMP as separate standards is simple. This allows the management station to monitor multiple nodes, many with a different set of MIBs specific to the node. A MIB is configurable and can be updated. If a node is upgraded to support new and/or approved standards, the MIBs can be updated on the manager to match what is available on the agent.

> **RANDOM BONUS DEFINITION**
>
> half duplex — A communication mode where a device can either transmit or receive data across a communications channel, but not at the same time.

The formal language used by SNMP is *Abstract Syntax Notation 1* (ASN.1, pronounced *A-S-N-dot-one*). ASN.1 specifies how information can be mapped so it can be readable by humans and data nodes as well. The purpose of this encoding of data is to assign names and variables contained within a MIB to a standard so they can be precisely read and recorded by administrators as well as SNMP supported nodes. A subset of ASN.1 is the Structure Management Information (SMI) standards, which define the relationship of MIB objects.

The MIB structure is similar to the structure that is used by DNS. It is a hierarchical tree structure with an unnamed root at the top of the tree and then levels of *object identifiers* (OID). An OID is a series of sequential integers separated by dots. The OID defines the path to the sought object. Figure 5-7 shows an example of the OID for the MIB variables.

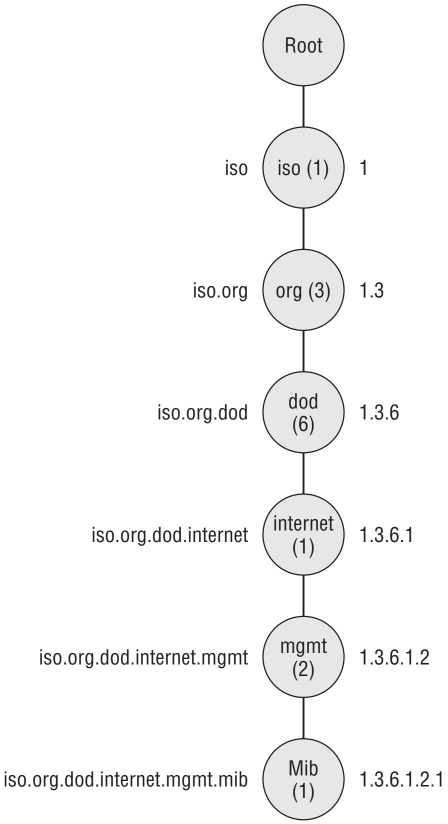

Figure 5-7 The OID structure for SNMP MIB variables

In Figure 5-7, you can see the OID string on the right side of the tree and the corresponding names for each level. All MIB variables will start with `1.3.6.1.2.1`, which is assigned the named value of `iso.org.dod.internet.mgmt.mib`.

5.2.1.2.5 SNMP version 2

The Simple Network Management Protocol version 2 (SNMPv2) introduced improvements and additions to some of the areas in the original SNMP standard. These improvements include

- Improved security
- SNMP-manager-to-SNMP-manager communication
- Improved performance
- Confidential sessions
- Additional protocol support
- Improvements in the way Trap PDUs are handled

SNMPv2 also introduced two new types of PDUs. The first one is called *GetBulkRequest*, which improved on the *GetNextRequest* PDU by giving the SNMP manager the ability to retrieve all of that consecutive data in one request instead of one request in between responses. In other words, everything is handled in one request and return response. The second PDU type that was introduced by SNMPv2 is *Inform*, which allows an SNMP manager to receive and reply to traps sent to and from another SNMP manager.

ACRONYM ALERT

TAG — Technical action group

SNMP and SNMPv2 are not completely compatible. They use different message formats as well as handle protocols differently. There are some optional configuration strategies that will help these versions coexist within the same network. One of these optional strategies is called a *bilingual network management system*, where an SNMP manager will determine what version an agent is using and then will speak with that agent in the version the agent understands. The other strategy is through the use of a *proxy agent*, where an SNMPv2 agent can act as a middleman and translate communications between an SNMPv2 manager and an SNMP agent.

5.2.1.2.6 SNMP version 3

The Simple Network Management Protocol version 3 (SNMPv3) is considered the official standard and is the one that will be developed upon if there are any updates or enhancements needed at some point in the future.

SNMPv3 introduces some very important support for securing the access to nodes in the network and also offers remote node configuration support. SNMPv3 ensures message integrity, authentication, and encryption to assist in preventing unwanted individuals from accessing important information from traffic between the managers and the agents.

5.2.1.3 File Transfer Protocol

The File Transfer Protocol (FTP) provides the ability for users to access an FTP server and transfer files to and from the server. FTP is used by network nodes as well as end users for file transfer of large amounts of data.[15] FTP is a really easy protocol to use. It provides an interactive interface for end users, authenticates and provides access controls based on the authorizations that have been given to the users, and enables the system administrator to determine the format of the stored data.

The only thing that is required for file access with the FTP protocol is a node that is running FTP server software, and the users must have some sort of a client software application running on their workstations.[16] The server needs to know the user credential information. The user needs know their user ID and password, as well as the name or IP address of the FTP server.

Nodes that participate in an FTP session can be in the same building or across the world from one another. To connect to the FTP server, all you have to do is issue an `ftp` command. The following example opens an FTP session between a workstation and the widgetsinc.com FTP server. Once connected, the FTP server will print any messages that are configured on the server and then will request the login credentials.

```
% ftp widgetsinc.co
Connected to widgetsinc.co
220-FTP server ready
230- Have a great day!
230-
230-Access to this network and the information on it are the lawful
230-property of widgets.co and its employees. If you
230-are not an authorized user then you are not authorized
230-on this server.
230-User (widgetsinc.co:(none)):
```

Previously we said that FTP provides an interactive interface for end users, provides user access control, and that the format of the data stored can be of various types. Now, let's take a look at some of these functions. For the examples in the following sections, we used a Microsoft Windows PC via

[15] Sure, you can e-mail files too, but try to e-mail a 100 MB file.

[16] If the node is TCP/IP compliant, the utility should already be available.

the cmd.exe[17] window for all command-line operations. Additionally, there is a freeware FTP server application (Cerebus FTP server) that is available for download and supported by most Windows-based PCs. This application can be downloaded at www.cerberusftp.com. We recommend that you use this application if you are interested in replicating some of the examples.

RANDOM BONUS DEFINITION

full duplex — A communication mode where a device has the ability to simultaneously transmit and receive data across a communications channel.

End users can use an FTP client application to access a node that is running the FTP server software for the purpose of either placing files on the server (with the `put` command), or getting files from the server (with the `get` command).

The directories on an FTP server can also be manipulated by the end user, provided the user had the appropriate credentials when they log in. We will talk more about user access in the next section; for now, all you need to know is that you can perform the following functions with FTP:

- Retrieve files
- Store files
- Create directories
- Remove directories
- Rename directories
- View hidden files and directories
- Issue miscellaneous commands to navigate the directory tree

ACRONYM ALERT

SA — Source address

As with any command-line structure you may come across, FTP utilizes several commands to perform tasks while in an FTP session. The command structure can vary from operating system to operating system, but the function of the command remains the same. Table 5-1 lists some of the more common FTP commands.

Keeping track of whether you are here or there is important when you are in an FTP session. Keep in mind that you will be working with files and directories on two nodes. If you are *getting* a file, you are pulling it off of the remote node and filing it away on your local node. Likewise, if you are *putting* a file, you are getting a copy of the file on your local node and saving it on the remote node.[18]

[17] cmd.exe is a command-line interpreter for most Windows-based systems that are in use today. It is the command that allows a user to communicate with the OS.

[18] This sounds straightforward, and it really is. It does get confusing at times when you have been working on an issue for a while and sleep deprivation sets in.

Table 5-1 Common FTP Commands

COMMAND	FUNCTION
`ascii`	Sets the file transfer mode to ASCII.
`binary`	Sets the file transfer mode to binary.
`cd`	Changes to another directory.
`close`	Terminates a connection.
`delete`	Removes a file.
`get`	Places a copy of a file on the remote node into a specified directory on the local node.
`hash`	Used to monitor the file transfer process. For every 1028 bytes received, a # will be placed on the screen.
`help`	Lists available FTP commands.
`?`	Gets information about commands.
`ls`	Lists the names of the files in the current directory.
`mget`	Used to copy more than one file from the remote node to the local node.
`mkdir`	Makes a new directory.
`mput`	Used to copy more than one file from the local node to the remote node.
`put`	Used to copy a file from the local node to the remote node.
`pwd`	Determine the directory path to the current directory.
`quit`	Terminates the FTP session.
`rename`	Renames a file or directory.
`rmdir`	Removes a directory and any subdirectories, if applicable.

Now it's time for a special treat. The following walks through the process of putting a file from the local node onto the remote node.

1. Once you have the name or IP address of the remote node (the FTP server), open up a session with the server, using an FTP client (in our case, we are using the command line). You should see some confirmation that you have connected, then the banner (if there is one) is printed, and you will be prompted to log in.

```
C:\>ftp 192.168.1.104
Connected to 192.168.1.104.
```

```
220-Access to this network and the information on it are the
220-lawful property of widgets.co and its employees. If you are
220-not an employee or an authorized user, then you are not
220-authorized to be on this server.
220
User (192.168.1.104:(none)):
```

2. Log in using the credentials that have been provided to you. Some users may have more rights on the server than other users. Most FTP server administrators also allow for anonymous logins. Anonymous logins are beneficial if you have customers, vendors, and partners you may want to share files with, but not give them full access, only access to the directories they have a need to connect to. Once you have logged in and provided the password, you will receive confirmation that you have been authorized on the server.

```
User (192.168.1.104:(none)): jedwards
331 User jedwards, password please
Password:
230 Password Ok, User logged in
```

3. Use the `ls` command to see what directories and files the current directory possesses. In the following example, note that there are two directories: ftproot and widgets.

```
ftp> ls
200 Port command received
150 Opening data connection
ftproot
widgets
226 Transfer complete
```

4. If you determine that you want to change to the widgets directory, use the `cd` command.

```
ftp> cd widgets
250 Change directory ok
```

ACRONYM ALERT

LLC — Logical Link Control

5. Use the `ls` command to see if there are any subdirectories; note the customers directory. Assume that you want change to that directory (with the `cd` command) and prepare to copy a file from our workstation to the remote node.[19]

```
ftp> ls
200 Port command received
150 Opening data connection
ftproot
```

[19] If you know the path name for the destination directory, you can change to that directory by listing the path (`cd widgets/customers`).

```
ftp> cd customers
250 Change directory ok
```

6. To verify your current directory, you can issue the `pwd` command.

```
ftp> pwd
257 "/widgets/customers" is the current directory
```

7. You can set the transfer mode to ASCII.[20]

```
ftp> ascii
200 Type ASCII
```

8. You can set the transfer mode to binary.

```
ftp> binary
200 Type Binary
```

9. Now put the file in the directory on the remote node. In this example, you will transfer two files: transfer.doc and transfer2.doc.

POP QUIZ

What is the function of the FTP command `ascii`?

```
ftp> put c:\transfer.doc
200 Port command received
150 Opening data connection
226 Transfer complete
ftp: 24064 bytes sent in 0.01Seconds 2406.40Kbytes/sec.

ftp> put transfer2.doc
200 Port command received
150 Opening data connection
226 Transfer complete
ftp: 24064 bytes sent in 0.00Seconds 24064000.00Kbytes/sec.
```

10. Since you transferred multiple files, you can also do this with the `mput` command. Take note that there is a confirmation required between files.

```
ftp> mput c:\trans*.*
mput c:\transfer.doc?
200 Port command received
150 Opening data connection
226 Transfer complete
ftp: 24064 bytes sent in 0.01Seconds 2406.40Kbytes/sec.

mput c:\transfer2.doc?
200 Port command received
150 Opening data connection
226 Transfer complete
ftp: 24064 bytes sent in 0.01Seconds 2406.40Kbytes/sec.
```

[20] ASCII is the default mode.

11. Finally, log out of the session with the `quit` command. This will close the session and display any messages, if configured.

```
ftp> quit
221 Have a great day
```

TIME FOR SOMETHING NICE TO KNOW

The `?` command and the `help` command do not require an FTP session to be established in order to run. If you type the command `ftp`, you initiate the FTP client. Once you have the FTP prompt, you can issue the `help` or `?` command to see a list of FTP commands. You can also connect to the remote node using the `open <destination name or IP address>` command. Here is an example of both these commands, and the output:

```
C:\>ftp
ftp> ?
Commands may be abbreviated. Commands are:

!           delete        literal      prompt       send
?           debug         ls           put          status
append      dir           mdelete      pwd          trace
ascii       disconnect    mdir         quit         type
bell        get           mget         quote        user
binary      glob          mkdir        recv         verbose
bye         hash          mls          remotehelp
cd          help          mput         rename
close       lcd           open         rmdir

ftp> open 192.168.1.104
Connected to 192.168.1.104.
```

5.2.1.4 Trivial File Transfer Protocol

Why waste time with a protocol that is so *trivial*?[21]

RANDOM BONUS DEFINITION

Session layer — Layer 5 of the seven-layer OSI model, responsible for process-to-process communication.

The Trivial File Transfer Protocol (TFTP)[22] is another popular file transfer program. Since the protocol uses UDP (see Section 5.2.2.2), there is less chatter than with the FTP protocol, which uses TCP (see Section 5.2.2.1). TFTP is mainly used

[21] Okay, it's a lame joke, but we could not resist.

[22] Note that not all nodes support TFTP. If a network is performing file transfer in a controlled environment, it is likely that TFTP is not used at all.

with the Bootstrap Protocol (see Section 5.3.4) to transfer node configuration files for nodes that do not have hard disk storage.[23] TFTP is also utilized to transfer files to and from network nodes for the purpose of troubleshooting, configuring, upgrading, and so on.

TFTP is a simple protocol that is small enough to be stored in a node's ROM. It requires a TFTP client and a TFTP server in order to function. Since UDP is a connectionless protocol, the TFTP server allocates different ports in order to support multiple TFTP clients at any given time. Security parameters are limited with the TFTP protocol. A system administrator can provide user access to only certain directories, but there is a potential for a security problem in the network if the TFTP sessions are not monitored and maintained.

TFTP does not have all the functions that are available with FTP. To understand why, keep in mind that TFTP is a simple file transfer protocol designed to transfer boot-up files for diskless nodes. You won't be able to browse directories, make directory changes, list files or directories, and you will be limited to the files you have been assigned.

TFTP commands are very similar to the FTP commands (keeping in mind that there are fewer options with TFTP). Table 5-2 contains a list of the most often used commands.

Table 5-2 Common TFTP Commands

COMMAND	FUNCTION
`connect`	Sets the remote node and/or ports for file transfer.
`get`	Places a copy of a file on the remote node onto a specified directory on the local node.
`hash`	Displays hash marks (#) to monitor file transfer progress.
`mode ascii`	Sets the file transfer mode to ASCII.
`mode binary`	Sets the file transfer mode to binary.
`put`	Copies a file from the local node to the remote node.
`quit`	Terminates the TFTP session.
`rate`	Displays the transfer rate information.
`status`	Displays relevant information about the transfer.

TFTP is connectionless. This means that a connection is not established prior to the transfer of data. When a user issues the `tftp <hostname or ip`

[23] Also known as diskless nodes or diskless systems.

address> command or the connect command, the client does not actually make a connection; rather, it buffers the information to use when it initiates the file transfer process. Following are a few TFTP command examples from a cmd.exe window:

1. To view the commands that are available in the cmd.exe command line for TFTP, you simply initiate the `tftp` command.

```
C:\>tftp

Transfers files to and from a remote computer running the TFTP service.

TFTP [-i] host [GET | PUT] source [destination]

  -i          Specifies binary image transfer mode (also called
             octet). In binary image mode the file is moved
             literally, byte by byte. Use this mode when
             transferring binary files.

  host        Specifies the local or remote host.

  GET         Transfers the file destination on the remote host
             to the file source on the local host.

  PUT         Transfers the file source on the local host to
             the file destination on the remote host.

  source      Specifies the file to transfer.

  destination  Specifies where to transfer the file.
```

2. To retrieve a file from the remote node and save a copy on the local node, use the `get` command.

```
C:\>tftp 192.168.1.104 get /widgets/Users/dns.doc
Transfer successful: 20480 bytes in 1 second, 20480 bytes/s
```

3. Finally, to place a copy of a file that is stored on a local node onto the remote node, use the `put` command.

```
C:\>tftp 192.168.1.104 put c:\dns.doc /widgets/Users/dns2.doc
Transfer successful: 6 bytes in 1 second, 6 bytes/s
```

It's as simple as that. Note that you have to know the full path for the file that you want to get and place on the remote node. This is because the TFTP protocol does not support directory path browsing. This makes it a little less simple than FTP, but if used mainly for transfer of files for diskless systems and system modification, it should easily serve the purpose of most networks.

5.2.1.5 Simple Mail Transfer Protocol

The Simple Mail Transfer Protocol (SMTP) is a protocol used for the transfers of electronic mail (e-mail) between network nodes. SMTP sets the format of e-mail from the client running on one node to a server running on another.

SMTP is not involved with the way an end user interfaces with an e-mail application or stores e-mail messages, when to check for new messages, or when to send messages, nor is it involved in determining what e-mail messages to accept or not accept on the destination node. SMTP is concerned only with how the e-mail messages are transferred across the shared medium.

SMTP works with the Post Office Protocol version 3 (POP3) and/or the Internet Message Access Protocol (IMAP), which enables e-mail messages to be stored (queued) on a server. The client periodically queries the server to check for and retrieve new messages. Without POP3 or IMAP, some messages might have a hard time reaching a destination due to the limited ability to queue data on the receiving node. In summary, POP3 and IMAP receive e-mail messages, and SMTP sends them. Many SMTP server applications include POP3 support in the same package.

Communication in SMTP is initiated by the client. The server will respond to a client query with a response code and an explanation. The server will also respond to other servers with response codes. Response codes can be used when troubleshooting e-mail transfer problems. Table 5-3 lists the server response codes and their meanings.

The client also has a set of messages that it will send to the server. There are a total of five messages used by a client to send an e-mail message. These are

- **HELO** — Used by the client to identify itself to the server
- **MAIL** — Identifies the end user sending the message
- **RCPT** — Identifies the end user the message is being sent to
- **DATA** — Identifies the contents of the message
- **QUIT** — Terminates the session

RANDOM BONUS DEFINITION

collision — When simultaneous transmission is attempted by two or more nodes on a shared Ethernet LAN

Table 5-3 SMTP Server Response Codes

SERVER RESPONSE CODE	EXPLANATION
220	Ready to receive mail from the client
221	Server is closing the session
250	Message sent from the server to the client informing the client that a requested action has been completed
251	Message sent from one server to another that it is forwarding mail for a user whom the server does not recognize
354	Message sent to a remote server in response to a query from that remote server about whether it can send mail
421	Server is unavailable
450	Message sent by the server to inform the client that a message could not be sent because the destination mailbox was not available
451	Message sent by the server when there is an error in processing a request; when this occurs, the request is terminated
452	Server has run out of storage space and cannot accept the message
500	Syntax error with a command
501	Syntax error with a function of a command
502	Server is not configured to support the request
503	Requests from the client are out of sequence and cannot be understood
550	Message cannot be delivered to the remote server or mailbox; if local, the mailbox is not available
551	The mailbox is not local, and the server cannot forward the message due to configuration constraints
552	User has run out of storage
553	SMTP address format is not correct
554	Request failed — no specification as to why

Following is a cleartext example of an SMTP session. We will assume that the client has already set up a connection request and is waiting for the response from the server (which is the response code `220` in the first line of the following example). The lines that begin with `S:` are messages from the server, and the lines that begin with a `C:` are messages from the client.

```
S: 220 smtp.widgets.com SMTP Service ready
C: HELO smtp.example.org
S: 250 Hello smtp.example.org, I am glad to meet you

C: MAIL FROM:<slick@example.org>
S: 250 Ok

C: RCPT TO:<blah@widgets.com>
S: 250 Ok
C: RCPT TO:<halb@widgets.com>
S: 550 That is not a valid user

C: DATA
S: 354 Input mail.  End data with <CR><LF>.<CR><LF>
C: From: "Slick Johnson" <slick@example.org>
C: To: Blah Blah Blah <blah@widgets.com>
C: Date: Thurs, 15 Jun 2008 08:02:11 -0500
C: Subject: Example
C:
C: Hey Blah!
C: I need 20,000 widgets.  Please send ASAP.
C: Sincerely,
C: Slick
C: <CR><LF>.<CR><LF>
S: 250 Ok

C: QUIT
S: 221 Bye
```

Notice how the five messages are organized in the SMTP transfer. Also, note that one of the intended recipients is not a valid user.

5.2.1.6 Network File System

Developed originally by Sun Microsystems, the Network File System (NFS) protocol allows end users access to files that are stored remotely as if the files were local to the end user's workstation. The original version of NFS used UDP as a transport protocol; however, with the release of NFS version 3 (NFSv3) in 1995, the protocol included transport via TCP. This made it more feasible to use NFS over a WAN, thus increasing the options available for networks that had implemented and utilized NFS.

Like all the Application layer protocols discussed so far, NFS is a client/server application. Using NFS, end users are able to view, store, update, and manage files on a remote server. All that is required is that the originating node has an *NFS client* application running and the remote node has an *NFS server* application running.

Files that are shared on the server node are *mounted*, or set as accessible, for the users in the network. Access is controlled based on the permissions or privileges that have been set for an individual user. Permissions are set based on what directories the user is authorized to access. Privileges can be read/write (user can modify the file) or read-only (user can view the file but cannot modify the file).

ACRONYM ALERT

CPU — Central processing unit

An NFS server must have some background applications running, known as *daemons*,[24] in order for the client to be able to connect to and utilize the services that are provided through the NFS protocol. Following are the daemons that need to run on the NFS server:

- **nfsd** — This is the NFS daemon, which receives and processes requests from the NFS client(s).
- **mountd** — This is the NFS mount daemon, which receives requests from `nfsd` and processes them.
- **rpcbind** — This is a daemon that provides a way for the NFS clients to see what ports the NFS server is using.

MORE UNIX DAEMONS

Here is a handy-dandy reference list of common Unix daemons and their functions.

- **dcpd — The DHCP daemon, which allows for the dynamic configuration of TCP/IP data for nodes running the appropriate client application.**
- **fingerd — The finger daemon, which provides finger protocol access to the server.**
- **ftpd — The FTP daemon, which supports and services FTP requests from a node running the client application.**
- **httpd — The HTTP daemon, which provides web server support.**

(continued)

[24] When you look at a node’s file system, you can usually tell which processes are daemons. Most of these are identified with a ‘‘d’’ at the end of the name of the process. For instance, the *http* daemon is labeled *httpd*.

MORE UNIX DAEMONS *(continued)*

- **lpd — The line printer daemon, which manages the spooling of print jobs.**
- **nfsd — The NFS daemon, which receives and processes requests from the NFS client(s).**
- **ntpd — The NTP daemon, which manages node clock synchronization.**
- **rpcbind — The RPC daemon, which takes care of remote call procedure conversions.**
- **sshd — The SSH daemon, which monitors for SSH request from an SSH client.**
- **sendmail — The SMTP daemon, which handles e-mail transport.**
- **syslogd — The system logging daemon, which logs system processes and system log messages.**
- **syncd — The synchronization daemon, which synchronizes file systems with system memory.**

NFS is more commonly used with nodes that are running a *Unix-like*[25] operating system; however, there are many other operating systems that can use and implement NFS in an environment where it is feasible to do so.

RANDOM BONUS DEFINITION

hub — A central interconnection device used in a star-wired topology

Users working in an NFS environment are able to access their home directories that are stored on the NFS server from any workstation that has access to the server. This is a huge benefit, especially for users who may migrate from workstation to workstation. Another benefit of NFS implementation is workstation resource sharing (not having to fit every workstation with the entire same storage medium and software requirements).

5.2.1.7 Telecommunications Network

The Telecommunications Network (Telnet) protocol gives a user the ability to access and manage a remote node. Almost all nodes that are running TCP/IP will support the Telnet protocol. The *Telnet client* initiates a session with a node that is running the *Telnet server* application.

[25] Unix-like is a term that is used to identify an operating system that is similar to the original Unix operating system.

POP QUIZ

What does an SMTP server response code 421 mean?

The server runs telnetd, which listens for a Telnet client request. Telnet is used mostly for system administration, management, and troubleshooting, but can also be used to check the status of other server types in the network.

To initiate a Telnet session, issue the following command:

```
telnet <ip address or dns name>
```

If you are successful, you will either be prompted with a login prompt or you will be at the user interface for the node. It depends on the settings of the remote node. Optionally, you can initiate a Telnet session in a Windows environment by issuing the `telnet` command. This will bring you to the Microsoft Telnet prompt, where you can view a list of commands. You can also initiate your session with the `open <ip address or dns name>` command. Following is the Windows Telnet client interface:

```
C:\>telnet

Microsoft (R) Windows 2000 (TM) Version 5.00 (Build 2195)
Welcome to Microsoft Telnet Client
Telnet Client Build 5.00.99206.1

Escape Character is 'CTRL+]'

Microsoft Telnet> ?

Commands may be abbreviated. Supported commands are:

close           close current connection
display         display operating parameters
open            connect to a site
quit            exit telnet
set             set options (type 'set ?' for a list)
status          print status information
unset           unset options (type 'unset ?' for a list)
?/help          print help information
```

5.2.1.7.1 Network Virtual Terminal

Because there are so many different operating systems, it's important that a client and server can participate in a Telnet session regardless of which operating system they are running. This is done through the use of a virtual node known as a *network virtual terminal* (NVT). The NVT basically provides a way for the client to provide a mapping to the interface the end user is using, and the server will map to a terminal type that it supports. Data in the NVT environment is input to a *keyboard* and then output to a *printer*. Figure 5-8 is an example of an NVT.

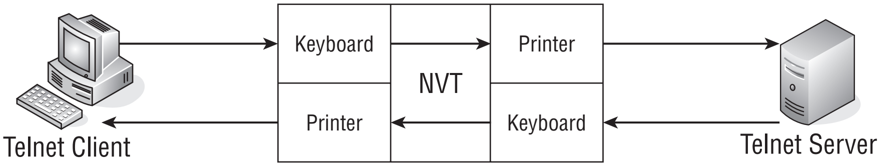

Figure 5-8 An NVT example

5.2.1.7.2 Options and Option Negotiation

If a Telnet client supports it, the client and server have the ability to negotiate the use of features known as *options* for the session. Options can be negotiated before a Telnet session is set up or at any time during the session. The following four control characters are used for option negotiation:

- **WILL** — Used when the sender wants to enable an option
- **WONT** — Used when the sender wants to disable an option
- **DO** — Used when the sender wants the receiver to enable an option
- **DON'T** — Used when the sender wants the receiver to disable an option

POP QUIZ

What does the acronym *NFS* stand for?

Table 5-4 lists some Telnet option codes.[26]

Option negotiation can be initiated by the server and the client. Some options are specifically for a client (that is, the server doesn't have a need to request), and some are for the server.

5.2.1.7.3 Modes of Operation

Telnet servers and clients comply with one of three modes of operation:

- **Half-duplex mode** (the default) means that communication takes place in half-duplex. This in and of itself is why this mode is for the most part never used. Most nodes now support full-duplex, which means that communication cannot be handled in half-duplex

[26] Currently there are more than 50 option codes.

mode. In this mode, echoing is performed by the client, and the client will not transmit new data until the line that was sent previously is complete and has been received by the remote node.

Table 5-4 Option Codes

OPTION CODE	OPTION	EXPLANATION
0	Binary	Assumes that transmission is binary
1	Echo	Repeats information received
3	Suppress go ahead	Suppresses go ahead signaling
5	Status	Lists the Telnet status
6	Timing mark	Sets the timing mark
24	Terminal type	Sets the terminal type
31	Window size	Sets the window size
32	Terminal speed	Sets the terminal speed
33	Remote flow control	Sets the remote flow control
34	Line mode	Sets to line mode

- **Character mode** is a mode where only O-N-E_C-H-A-R-A-C-T-E-R at a time is transmitted. The server will provide an acknowledgment when it receives each character and the echoing is performed by the server. The client, in turn, will send an acknowledgment to the server as well.
- **Line mode** is the mode where full-duplex transmission occurs with data being transmitted a line at a time. In line mode, text that is entered by the user is echoed locally and only full lines of data are transmitted to the server. This greatly reduces the number of packets that are required to be transmitted across the network.

5.2.1.8 Secure Shell Protocol

The Secure Shell (SSH) Protocol provides a very important function that Telnet lacks: the ability to protect the integrity of the data being transmitted by supporting encrypted connections between network nodes.

SSH utilizes *public key cryptography*, which provides cryptographic keys to authenticate remote nodes and users. In public key cryptography, two keys are involved in the encryption/decryption process: the *public key*, which can be shared by multiple remote nodes, and a *private key*, which is a secret used to decrypt a corresponding public key.

POP QUIZ

What is the purpose of Telnet option code `32`?

Nodes that support SSH have both a public and a private key assigned to them. The private key is protected by a password, which is entered by the user. The private key corresponds with the public key, which matches the public key on the remote end. The remote node has a private key as well that will decrypt the information sent to a readable form for the remote user.

SSH is used primarily as an encrypted form of Telnet. With SSH, you can log in and be authenticated so the session is less vulnerable to attack than is the Telnet session. SSH also provides other functions, which makes it a very appeasable application to support in a network.

SSH servers listen for requests coming from an SSH client. The SSH daemon runs on the server node. There are many SSH variations in today's networks. The most popular ones are OpenSSH and Putty.[27] The most recent version of the SSH protocol itself is SSH version 2 (SSH-2), which has been submitted as a proposed Internet standard.

5.2.2 The Transport Layer

The next layer of the TCP/IP reference model is the Transport layer. It is the layer that accepts requests from the Application layer, and it sends requests to the Network layer. Transport protocols operate at the Transport layer. The two most popular of these protocols are the User Datagram Protocol (UDP) and the Transmission Control Protocol (TCP) at the Transport layer, both of which we introduce in this section. Chapter 9, ''The Transport Layer,'' will discuss these in more depth.

5.2.2.1 Transmission Control Protocol

RANDOM BONUS DEFINITION

Media Access Control — The entity or algorithm used to arbitrate for access to a shared communications channel.

We bet you are thinking to yourself that you must have heard about this protocol before. Well, you have heard of it. At the very least you have heard it mentioned in this book, and it's a good possibility that you have heard of it if you have ever configured your computer to be

[27] These and many others are open source applications, which can be downloaded from many different websites. An Internet search will point you to where you can download these.

connected to a network. You may not have known what it does, but you have heard of it.

TCP is used to transport data. It ensures that data is placed in sequence (the order that it was sent in), that data arrives at its destination (or will force a retransmission if it didn't), and it helps cut down on over-traffic in the network. To give you an idea of why TCP is important, take a look at Figure 5-9.

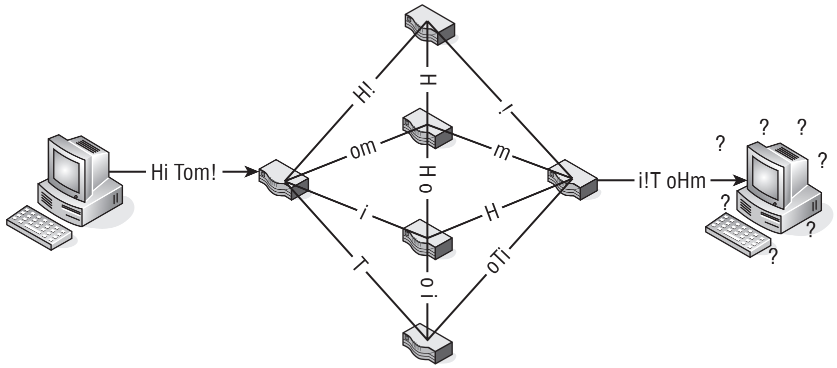

Figure 5-9 An example that proves why TCP is very helpful

In the figure, you can see that a node wants to send the message ''Hi Tom!'' to a remote node.[28] There are many different paths that data can take to get from the originating node to the remote node. Assuming that we are sending one character at a time, each character will take whatever path the routers tell it to take. Because the originating and the destination nodes do not know which path the data is taking, the destination node will have no way to put the data back together when it receives it, and therefore will most likely receive a jumbled mess. Note that the destination node receives all the data, but the message received is ''i!T oHm,'' which is nothing like the originating message.[29]

TCP is a connection-oriented protocol, which means that a TCP session must be established between a TCP server and a TCP client before any data transmission occurs. Most professionals use the analogy of a telephone when explaining the meaning of connection-oriented. When you make a phone call, you wait until someone answers the other end before you say hello, hey, how's it going, or anything else that you called to say.[30] This is exactly how TCP works. An originating node will contact a destination node to make sure they

[28] For this example, it really does not matter what application is being used to send the message. All that is important is that you understand that the information is coming from the Application layer and is being sent to the Network layer.

[29] Can you imagine what Brother Joel might think about this message?

[30] Some phrases can be uttered that we can't mention in this book.

are available to get the message. Once confirmation is received that it is okay to send data, the transmission begins.

TCP is also considered a *reliable* protocol because there are functions built into TCP that provide for various checks and balances to ensure the integrity of the data being transmitted. Some of the reliability functions are

- TCP is able to break down data that is received from the Application layer into *segments*.
- TCP places an acknowledgment timer on sent segments. When the timer expires, if the originating node does not receive confirmation from the remote node that the segment was received, the originating node will resend the segment.
- TCP maintains a checksum (within the TCP header and within the actual data payload) that is set on each end of the connection. The checksum is used to ensure that data arrives exactly as it was sent. If the receiving node notices that the checksum does not match (invalid checksum), the receiving node will throw the segment away. In throwing the segment away, the receiver does not receive the segment. This means that the receiving node does not send an acknowledgment, which causes the originator to send it again.
- TCP datagrams are not sent in order. They traverse the network over the best path possible (based on calculations made by nodes, which we discuss in several places throughout this book). TCP supports the ability for the receiving node to put all of the datagrams back into the correct order, once they have been received.
- TCP can recognize duplicate datagrams and can discard them when received.
- TCP supports what is known as flow control. Flow control is a way for each node to know how much buffer space they have available to receive data. This way no node will overwhelm the other node with more data than it can handle.

POP QUIZ

What does the acronym *SSH* stand for?

Examples of applications[31] that use TCP would be

- FTP
- Telnet

[31] Notice that some protocols use both TCP and UDP (DNS, for instance).

- SMTP
- DNS
- POP3
- HTTP
- DNS
- IMAP

5.2.2.2 User Datagram Protocol

Here is a bonus question for you. The User Datagram Protocol (UDP) is part of the Transport layer and is used to do what to data?

That's right! Just like TCP is used to transport data between nodes, UDP is also used to transport data within a network. That is about the only thing (at least functionally) that the two have in common. UDP does not guarantee that data is going to be delivered to a destination. Basically, UDP throws the data toward the destination and then moves on to its next task. This makes UDP a connectionless protocol.

UDP is usually used to send short bursts of datagrams between nodes where reliability is not a big concern. UDP can get data to a destination quicker, as it avoids all of the overhead required when all the checks and balances are occurring within TCP. Also, because UDP is connectionless, it can support *broadcasting* (sending messages to all nodes within a broadcast domain) and *multicasting* (sending messages to all nodes that are subscribed to the network).

RANDOM BONUS DEFINITION

operating system — The application software responsible for the proper operation of a given node.

UDP provides an optional checksum that can be assigned to the UDP header as well as the data payload. This ensures that if any data that is sent over UDP requires a header and data payload checksum, the destination is able to do so. If any error checking is required, it will normally be performed by the application, not via UDP.

Most voice and video applications transmit over UDP. If you have ever watched a video online that cut out or got choppy at times, this is because data was not being received. Recovery from these choppy moments can go unnoticed for the most part. If TCP were used in these instances, there would be delays that last much longer when packet loss is requiring retransmission of the data. Keep in mind that speed is the consideration when going with UDP, not reliability.[32]

[32] You can always reload that video if you want to watch it again.

Examples of protocols that use UDP include

- DNS
- BOOTP/DHCP
- TFTP
- SNMP
- RIP
- NFS

UDP accepts data (the payload) from the Application layer. It then adds a UDP header and passes the header and the payload to the Internet layer, where it is encapsulated into an IP packet and is passed on to the Network Interface layer and over the transmission medium to the destination, where it makes its way up to the Application layer on the destination end of the connection.

5.2.3 The Internet Layer

The final layer that we will be discussing in this chapter is the Internet layer. Although we will discuss this layer in detail in Chapter 10, we wanted to provide a quick overview of some Internet layer protocols.

POP QUIZ

Name the two popular transport protocols that we discussed in this chapter.

This layer is responsible for ensuring that there is a path to a destination. It receives information from the Transport layer and ensures transmission to the destination node. Some examples of protocols that operate at this layer include

- Internet Protocol (IP)
- Internet Group Multicast Protocol (IGMP)
- Internet Control Message Protocol (ICMP)
- Address Resolution Protocol (ARP)
- Routing Information Protocol (RIP)
- Open Shortest Path First (OSPF)
- Border Gateway Protocol (BGP)
- Internet Protocol Security (IPSec)

Although all layers of the TCP/IP reference model are important in their own right, the Internet layer is probably the most important one. It provides the ability to route data to a destination based on an IP address. It manages the IP addressing structure for a network, and it also defines the datagrams that are transported to a remote node.

5.2.3.1 Internet Protocol

The Internet Protocol (IP) is the most important protocol that exists within the Internet layer. IP receives data from one of the Transport layer protocols, packages it into a datagram, and then transports it to and from a given set of nodes. IP is a connectionless protocol, which means it does not establish a line of communication prior to transmitting.[33] IP is also responsible for the IP addressing for network nodes.

The network node that is responsible for getting data between different networks is a router. The router is responsible for receiving a datagram known as a *packet* and pointing the packet in the direction it needs to go, based on the IP address the packet is looking for. IP addresses are learned by the router based on information from another router or information that it has discovered as it was passing packets to and fro. The information received for the purpose of routing packets is determined, calculated, and provided for by a routing protocol. IP addresses can also be configured and set statically (hard coded), but this is a tedious task to maintain. The dynamic option is a preferred method.[34]

ACRONYM ALERT

FCS — Frame Check Sequence

Since IP is connectionless, the upper layers are responsible for any error checking. The most IP will do is drop a packet and then send a message to the source IP address telling them that the packet didn't make it to where it was supposed to go. There are many protocols that work with IP and are placed into an IP packet for transmission. Some of these include

- TCP
- UDP
- ICMP

POP QUIZ

Which layer of the TCP/IP reference model is probably the most important one?

There are a few versions of IP in use today. IP version 4 (IPv4) is the most commonly used version, but a proposed standard, IP version 6 (IPv6), is in use and

[33] Here is more of that repetition that we mentioned in the front matter of this book.

[34] You will find that there are times when static routes make the most sense. They can also help you get a route back up when you are troubleshooting an issue. Static routes can be your friend.

is intended to eventually be the successor to IPv4. The main difference between IPv4 and IPv6 is the addressing. IPv6 allows for more addressing flexibility, as there is room for a larger address space. Both versions will probably be around for a long time, and there are ways to ensure that they can coexist, but eventually you will probably see a migration to IPv6.

Have you ever heard of *IP Next Generation (IPng)?* IPng is nothing more than the unofficial name for IPv6. The name was coined early and replaced when the proposed standard was submitted.

5.2.3.2 Internet Group Multicast Protocol

The Internet Group Multicast Protocol (IGMP) is a protocol that provides support for IP multicasting. IGMP provides a way for messages to be sent to multiple nodes. Nodes are grouped into multicast groups, so when a multicast message is destined for a group, only that group will receive the message.

IGMP messages are transmitted in an IP datagram. Multicast routers (that is, routers that can support multicasting) use IGMP messages to keep track of what groups are connected to what interfaces on the router. When the operating system of the originating node initiates a program process that requires IGMP support, the node will send a report out of an interface in which the process joins the group. Processes can join groups over multiple interfaces. When there are no other processes running in a group, the node will no longer report the group.

IGMP queries are sent out by a multicast router periodically to see if anyone has a process that might belong to a multicast group. This query is sent out of every router interface. When a remote node receives an IGMP query, it will respond with one report for each group that it recognizes as having a running process.

There may be many remote nodes running processes that are tied to a multicast group. Each node is responsible for reporting process and group information. The times that these reports are sent are staggered so there are not too many nodes responding at the same time. For a router to acknowledge a multicast group, there must be at least one node that is a member of the group.

5.2.3.3 Internet Control Message Protocol

The Internet Control Message Protocol (ICMP) is responsible for reporting conditions that need attention. When something goes wrong with IP, TCP, or UDP transmission, ICMP is there to let you know about it. Like ICMP, TCP, and UDP, ICMP messages are transmitted within an IP datagram.

Two versions of ICMP are in use today: *ICMP version 4* (ICMPv4)[35] and *ICMP version 6* (ICMPv6). ICMPv4 was developed to work with IPv4, so with the release of IPv6 updates were required and ICMPv6 was born.[36]

The functions of each version are basically the same. ICMPv*whatever* is there to pass messages. Following are the main reporting functions performed by ICMP:

- Error reporting
- Testing and troubleshooting
- Informational reporting

IP and ICMP work very well together. As a matter of fact, you can consider ICMP the ''right-hand man'' of IP. While IP is busy packing up data and routing that data to a destination, ICMP is taking care of all the busywork. ICMP passes messages that help ensure IP can perform its job well.

Many consider ICMP one of the simplest protocols there is. If you think about it, this is true. ICMP doesn't have to give a lot of thought or calculation to do its job. All it has to do is pass messages.

5.2.3.4 Routing Information Protocol

The Routing Information Protocol (RIP) is a dynamic routing protocol that is used in many networks. It is a *distance-vector* protocol, which means that each router will advertise the destinations it is aware of and the distance to each destination to neighboring routers.

POP QUIZ

What is the difference between *IPng* and *IPv6*?

Many different implementations of RIP were in place when the protocol became an Internet standard. Although there were a few differences between RIP implementations in different networks, the differences didn't cause many interoperability issues in production. A second version of RIP (RIPv2, or RIP2) was introduced and offered a few improvements over the original version of RIP. The most notable of these improvements was the support of variable length subnet masking (VLSM)[37] and support for authentication.

[35] ICMPv4 wasn't always called that. It was called simply ICMP since its inception. The v4 was added later to separate it from ICMPv6.

[36] ICMPng in and of itself is a pretty cool acronym. Not too many adopted the term, but at least one of the authors of this book would have adopted it (yes, we are talking about that author who thinks *catenet* is a cool term).

[37] VLSM increases the efficiency of the utilization of IP addresses in a given network by allowing different subnet masks to be used for each subnet. This will be discussed in Chapter 10, ''The Internet Layer.''

RIP determines distances to a destination based on what is known as a *hop count*, which is the number of devices a packet must pass through on the way to a destination. The hop count increases each time a packet reaches a node along the path to its destination. The link taken by the packet from one node to another node is the actual hop. Figure 5-10 shows an example of hops[38] in a network.

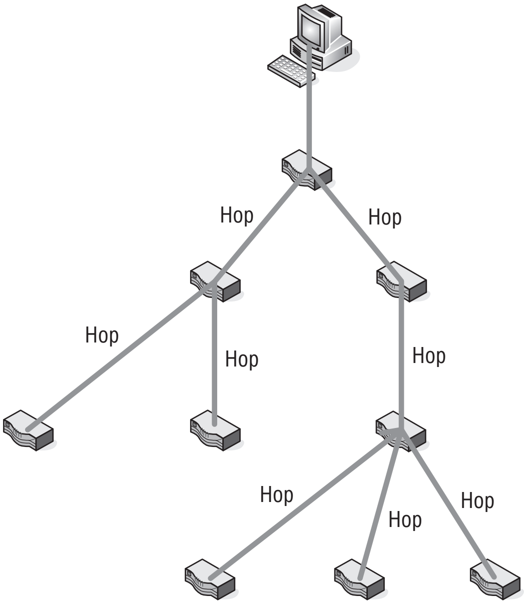

Figure 5-10 Hops in a RIP-routed environment

Now, let's quickly review the operation of RIP. When a router first boots up, one of the first things it will do (once connectivity is established) is send a packet out of each interface requesting routing tables from each of the neighboring routers. In turn, each router will send the routing table to the router that requested it. As the router receives the routing table from the neighboring routers, it will send a response telling the neighbors it has received the requested routing table. The neighbors will respond with any updates they may have since they last sent the routing table. If there are no updates, the neighbors will validate that they know of the originating router.

Once the preliminary routing table updates are performed, the routing table of each router will be broadcast to all other neighbor routers. This update occurs every 30 seconds. Updates known as *triggered updates* will occur whenever

[38] This is not to be confused with the flower hops, which is a key ingredient in beer. There is a shortage of hops at the time of this writing, which makes the hobby of home brewing a bit more expensive than in years past.

there is a change with the hop count to a destination. When triggered updates occur, only the information that has changed is sent.

5.2.3.5 Open Shortest Path First

The Open Shortest Path First (OSPF) protocol is a dynamic routing protocol that uses the *link state* between nodes to determine routing paths for packets. The link state is simply the state of the link to the next router (the neighbor). Routers in an OSPF environment do not check the distance from one point to another in a network. Instead, the routers monitor the state of a link to each of its neighbor routers (the router next door). The link states are logged into the *link state database* (LSDB), which is then shared with all the neighbors. LSDB information that is received is used to build the routing table for the router and then the information is shared with its neighbors.

Although an OSPF system can be a single autonomous system, most often OSPF routers are assigned as members of OSPF areas. Each area is identified by a 32-bit identifier, much like an IP address. Routers in the OSPF environment are also assigned tasks they need to perform to ensure that the routing domain runs smoothly. Following are a few important terms you will need to know:

- **Backbone area** — The core of the entire OSPF network. The identifier that is assigned to the backbone area is 0.0.0.0. All areas are connected to the backbone area.
- **Stub area** — An autonomous system that only receives LSDB updates from routers within the same area. The stub area only receives external routes through the default route.
- **Not so stubby area (NSSA) —** A stub area that contains no external routes. The NSSA can retrieve external updates and send them to the backbone.
- **Internal router** — Any router that only shares information with routers in the same area.
- **Backbone router** — Any router that participates in the backbone area. Most backbone routers are ABRs as they share information between areas. There may be some routers in the backbone that are not ABRs, but these are still backbone routers as they are in the backbone area.
- **Area border router (ABR)** — Any router that is a member of more than one area.
- **Autonomous system boundary router (ASBR)** — Any router that shares link state with a router in another area is called an ASBR. Note that any router within the area can be an ASBR; this includes area border routers, backbone routers, and internal routers.

- **Designated router (DR)** — Any router that handles advertisements on multi-access networks. The DR is elected by a process among other routers. It is responsible for being the representative for the multi-access network to the rest of the network. It is also in place to ensure that data is not flooded due to the multi-access environment.
- **Backup designated router (BDR)** — Any router that takes over the responsibilities of the DR if the DR should fail.

5.2.3.6 Border Gateway Protocol

The Border Gateway Protocol (BGP) provides for IP data communication between routers that are in different autonomous systems (AS). BGP routers share information with one another, providing paths that can be used to reach an AS. To prevent routing loops, BGP routers make a determination of the best path and any possible loops are pruned from the decision tree.

An AS can be classified much as areas are in OSPF, including

- **Multihomed AS** — An AS that connects to more than one other AS. A multihomed AS does not participate in transit traffic.
- **Stub AS** — An AS that connects to only one other AS. A stub AS does not participate in transit traffic.
- **Transit AS** — An AS that connects to more than one other AS. A transit AS participates in local and transit traffic.

Data traffic within an AS is either *transit* traffic (just passing through) or *local* traffic (traffic that starts or ends[39] within the AS).

Like RIP, BGP is a distance-vector protocol. However, instead of counting hops to a destination, BGP counts the number of autonomous systems it takes to get to a destination. BGP also supports policy-based routing. In other words, policy specifications are set by the system administrator and are used to allow BGP to determine the best route to a destination, ensuring all policies are strictly enforced. This means that even though there may be a quicker path to take to a destination, policies may prevent a datagram from going on that path.

BGP sends what are known as *keepalive messages* to its neighbors to ensure that the neighbors are reachable. If they are not reachable, BGP will recognize this as a link failure.

5.2.3.7 Internet Protocol Security

Internet Protocol Security (IPSec) is a suite of protocols that allow for security and encryption for IP datagrams. IPSec is designed to provide endpoint to

[39] The alpha and omega of BGP traffic types.

endpoint datagram security (*transport mode*) for nodes that do not support security protocols.[40] IPSec is also used in VPN environments (*tunnel mode*), which allows the gateway to the network to provide security and authentication services for the users and networks the node supports.

IPSec provides several types of security for networks and the users of the networks. One of the biggest functions that came from IPSec is the ability to encrypt datagrams so that only the destination can read and understand them.[41] IPSec also provides checks of datgrams to ensure that they have not been tampered with in transit. Finally, IPSec provides for the authentication of users, to ensure that anyone that should not have access doesn't.

AN UNRELATED MOMENT OF PAUSE

Three friends were out driving one day. One was a network sales engineer, one was a network hardware engineer, and one was a network software engineer. All of the sudden the right rear tire blew out, and the car rolled to a stop. Since the car was full of problem solvers, the three friends jumped out of the car to survey the situation.

The network sales engineer proclaimed, "The car just won't do anymore; it is time to buy a new one!"

The network hardware engineer gave it some thought and then said, "We need to try swapping the left tires with the right tires. If that does not fix it, then we need to swap the front tires with the rear tires. If we are still having problems at that point, we will have to replace the tires."

The network software engineer then piped in, "You guys are just wasting time. We need to get back in the car and drive some more to see if the problem will just work itself out."

5.3 End of Chapter Hodgepodge

We hope that you now have a better understanding of the TCP/IP reference model, some of the protocols that operate in each layer, and how each layer interfaces with each of the other layers. As you continue through the pages of this book, we will be revisiting a lot of these protocols and discussing some of the details that make each one tick.

In this section, we will discuss some of the other processes that operate in a TCP/IP environment. Like many of the other functions and specifications that we have discussed in this chapter, we will be revisiting some of these in upcoming chapters.

[40] These nodes may support security, but not at the level that a network needs the node to.

[41] Remember when we were talking about key exchange?

5.3.1 There Is Hope for Diskless Nodes

The Bootstrap Protocol (BOOTP) manages IP parameters on a given network. It assigns IP addresses for a *pool* of users. Not only that, it also provides for operating system initiation for remote diskless nodes.

BOOTP is a network protocol that uses UDP for transport. When a node is booting up, there is a bootstrap process that initiates the execution of the node's operating system. If a node is running a *BOOTP client*, the node will send a request to a *BOOTP server* for assignment of an IP address, along with any other startup assistance that the client node requires (and the BOOTP server supports). BOOTP is normally integrated into the node's motherboard or NIC card.

The *Dynamic Host Configuration Protocol* (DHCP) evolved from BOOTP. Several enhancements were provided with DHCP, although BOOTP is simpler to implement and maintain. A single *DHCP server* can provide IP addresses, subnet masks, gateway information, and more. When a node connects to the network, the DHCP client will broadcast a request for information from the DHCP server. The server will then send the requested information so the node can connect and operate in the network.

BOOTP and DHCP are called *communication management protocols*. They can work separately or together (together is the most often implemented). DHCP can serve the requests that come from a BOOTP client.

5.3.2 A Little More Information on Routing

Just when you thought we had finished with our discussion about routers, here we are back on the subject.[42] Following are a few terms that we wanted to quickly touch on. Why not? We have to discuss them somewhere.

- **Routing protocol** — The protocol that performs functions that allow the routing of packets between routers. RIP, OSPF, and BGP are examples of routing protocols. Sometimes confused with a *routed protocol*, which is not the same thing.
- **Routed protocol** — A protocol that participates in transmitting data between nodes within a network. Telnet, SNMP, and IP are all examples of a routed protocol. Routed protocols are sometimes incorrectly termed *routing protocols*.
- **Gateway** — The entry point for an entity. A computer that provides access to a network area is a gateway. A network that provides access to another network is a gateway. Many applications have gateways that allow information sharing. The node that connects the LAN to the Internet (or any other network type) is a gateway.

[42] We are far from finished with our discussion on routers.

- **Interior Gateway Protocol (IGP)** — A routing protocol that operates within an AS. RIP and OSPF are IGPs.
- **Exterior Gateway Protocol (EGP)** — BGP is often called an EGP, although the EGP protocol was the predecessor to BGP for IP routing between autonomous systems.
- **Static routing** — IP routing information that is manually configured on a node by a system administrator.
- **Dynamic routing** — IP routing information that is learned by the node through a routing protocol, such as RIP.

POP QUIZ

What are the two IGPs that we discussed in this chapter?

This concludes our discussion of routers for this chapter.

5.3.3 Sockets and Ports Are Not the Same Thing

A couple of important terms that often get confused are *socket*[43] and *port*. Note that we are referring to TCP and/or UDP ports, not to the physical interface of the node. A TCP or UDP port is a number assigned to the datagram header that is mapped to a particular process or application on a given node. A socket is the end-point of data communication flow on a network.

TCP and UDP ports are basically an extension of addressing used by TCP/IP to ensure that data communication is tied to the correct running process. Each packet header that is transported over TCP or UDP has a source and destination port logged in it. The port number can range from 0 to 65535. Port numbers are divided into three sections. These are *well-known ports* (0 through 1023), registered ports (1024 through 49151), and dynamic and/or private ports (49152 through 65535).

TCP/UDP WELL-KNOWN PORT NUMBERS

Following is an example list of many popular well-known TCP and UDP port numbers. TCP well-known port numbers are identified by an assignment of 0 through 1023. This list is only an example to provide the port numbers for many of the protocols we have covered, along with a few that are just darn interesting.

(continued)

[43] Sockets are also often called TCP or UDP sockets (depending on the transport protocol), Internet sockets, or network sockets.

TCP/UDP WELL-KNOWN PORT NUMBERS *(continued)*

For a complete and current list, go to `www.iana.org/assignments/port-numbers`.

Port Number	Description	Applicable Protocol
0	Reserved	TCP and UDP
1	TCP port service multiplexer	TCP and UDP
5	Remote job entry	TCP and UDP
7	Echo	TCP and UDP
20	FTP – data	TCP
21	FTP – control	TCP
22	SSH	TCP and UDP
23	Telnet	TCP and UDP
25	SMTP	TCP and UDP
53	DNS	TCP and UDP
67	BOOTP/DHCP – server	TCP and UDP
68	BOOTP/DHCP - client	TCP and UDP
69	TFTP	TCP and UDP
80	HTTP	TCP and UDP
101	NIC host name server	TCP and UDP
107	Remote Telnet service	TCP and UDP
109	POP2	TCP and UDP
110	POP3	TCP and UDP
115	SFTP	TCP and UDP
118	SQL	TCP and UDP
123	NTP	TCP and UDP
135	DCE endpoint	TCP and UDP
143	IMAP	TCP and UDP
161	SNMP	TCP and UDP
162	SNMP trap	TCP and UDP
166	Sirius	TCP and UDP
179	BGP	TCP and UDP
213	IPX	TCP and UDP
220	IMAPv3	TCP and UDP

(continued)

TCP/UDP WELL-KNOWN PORT NUMBERS *(continued)*

Port Number	Description	Applicable Protocol
389	LDAP	TCP and UDP
401	UPS	TCP and UDP
500	ISAKMP	UDP
513	Login	TCP
513	Who	UDP
515	Lpd	TCP
520	RIP	UDP
546	DHCPv6 client	TCP and UDP
547	DHCPv6 server	TCP and UDP
647	DHCP failover	TCP
666	Doom (video game)	UDP
989	FTP data over TLS/SSL	TCP and UDP
990	FTP control over TLS/SSL	TCP and UDP
992	Telnet over TLS/SSL	TCP and UDP
1023	Reserved	TCP and UDP

Any application that provides a common and well-known service (SMTP, FTP, Telnet, etc.) will monitor for incoming requests on the well-known ports. Firewalls can be configured to allow or deny specific ports, thus enhancing network security. If a request comes in with a port that is not defined, the server will assign a port number for the duration of the application process.

ACRONYM ALERT

CRC — Cyclic redundancy check

The socket is the combination of an IP address or node name and a port number. The syntax of a socket would be

```
<ip address> :< port number>
```

An example of this would be the Telnet protocol, which uses port number 23 (for both TCP and UDP). If the host that is running the Telnet server has an IP of 10.10.10.10, the Telnet client would send a request to that IP for port number 23. The syntax would look like this:

```
10.10.10.10:23
```

Any given port can have a single passive socket, which monitors for incoming requests, but can serve multiple active sockets, each serving a request from a different client.

5.4 Chapter Exercises

1. What are the four layers of the TCP/IP reference model?

   ____

   ____

   ____

   ____

2. Name four Application layer protocols that we discussed in this chapter.

   ____

   ____

   ____

   ____

3. Explain the structure of the DNS hierarchy.
4. What are the five PDU types that are used by SNMP?

   ____

   ____

   ____

   ____

5. What is the purpose of FTP?
6. Why does TFTP not perform many of the functions that FTP does?
7. What is a daemon?
8. What are the four control characters used by Telnet for option negotiation and their meanings?

   ____

   ____

   ____

   ____

9. TCP is a ____________-oriented protocol, whereas UDP is a ____________ protocol

10. What are the three main reporting functions that we said are performed by ICMP?

    ____

    ____

    ____

5.5 Pop Quiz Answers

1. The Internet layer is also known as the *Network* layer.
2. What is the function of the FTP command `ascii`?

   Sets the file transfer mode to ASCII.
3. What does an SMTP server response code `421` mean?

   Server is unavailable.
4. What does the acronym NFS stand for?

   Network File System
5. What is the purpose of Telnet option code `32`?

   Used to set the terminal speed.
6. What does the acronym SSH stand for?

   Secure Shell
7. Name the two popular transport protocols that we discussed in this chapter.

   TCP and UDP
8. Which layer of the TCP/IP reference model is probably the most important one?

   The Internet layer
9. What is the difference between IPng and IPv6?

   None. Other than the names, they are the same protocol.
10. What are the two IGPs that we discussed in this chapter?

    RIP and OSPF