Chapter 8

Networking

Unix and Linux are based on the ability for one host to communicate with another, either as another user or a server. In order to accomplish this, a means was established to allow multiple hosts to communicate via an addressing system and a set of protocols that specified how the message was constructed. In this chapter, we will investigate the basic properties of the Internet and the equipment required to make it functional.

Concepts Learned in this Chapter

- Internet Fundamentals
- Internet Control Message Protocol
- Internet Addressing and Subnet Mask
- Networking Equipment
- Network Connectivity
- Network Testing
- Network Interface
- IP Version 6

Table of Contents

In order for a Unix or Linux system to communicate with another system, various protocols and hardware devices were developed interconnectivity. A basic understanding of the process is necessary in order to allow the user to understand how a system operates and is configured.

8.1 OSI Model

One of the basic concepts of how a multiple host systems communicate with one another in today's network is based on one of two models, the original Department of Defense model and the OSI model. Both function identically, but differ in the number of stages that are used to accomplish the task. The most common accepted model is the OSI, which will be discussed.

8.1.1 OSI 7 Layers

The Open Systems Interconnection (OSI) Model has been subdivided into 7 layers, where each one performs a function of the connectivity process. These 7 layers are defined as:

Layer	Name	Description
7	**Application**	Provides the user various applications, such as email, web, ftp, telnet, and many others.
6	**Presentation**	1. Converts data from the presentation format on one host to the presentation format on a second host. 2. Encrypts data if application requires.
5	**Session**	Establishes and removes connections between two applications on an on-demand requirement.
4	**Transport**	1. Selects which network interface should be utilized to optimize network transmission. 2. Detects which packets have been received correctly or those that need to be retransmitted. 3. Formats the computer data into segments for transmission between different Internet applications by using a port address assigned to each service.
3	**Network**	Formats the computer data into a packet for transmission between different networks, using an Internet Protocol address.
2	**Data Link**	Formats the computer data into a frame for transmission between different hosts, using a Media Access Control (MAC) address.

1	**Physical**	Converts the electrical signal within the computer to a format that was usable on the network media, such as Ethernet.

Information from various Internet applications flows down the layers from one host and back up the similar layers of the second computer. In each layer, an additional process is applied to the original data. In flowing down the OSI stack, information regarding the transfer is added, and on the second computer, as data flows up the stack, the transfer information is removed.

A user application's data, such as a word processor document, is forwarded to the Internet applications for transport. For example, say this chapter file, which is approximately 150 K bytes long, needs to be transported to another host via the FTP protocol. The FTP Internet application is activated and the data is handed over to it.

The Presentation Layer, for this discussion, will not be required, as neither a modification in the screen format or encryption is necessary. The data is therefore forwarded to the Session Layer.

The Session Layer needs to establish a connection between the two hosts. Although the Session Layer does not know the IP address of the remote host, it does know that the data is to be transported to another Internet host. A session is established and various attributes are negotiated between the two. The data is then passed on to the Transport Layer.

The Transport Layer first determines which is the optimum transport path, that is, which route is best to utilize to transfer the data to the remote host. Information regarding the interface is utilized to divide the data into **segments**. The size of each segment is specified as the **Maximum Transport Unit** (**MTU**), and for Ethernet is specified as 1500 bytes. Since we are transporting the data via FTP, the service number of 20 is prefixed to each segment. Additionally, a sequence number is also added in order to insure that each segment is put back into the correct order on receipt at the far end; this is necessary because each segment can take a different path across the Internet. Each segment is then passed on to the Network Layer.

The Network Layer prefixes each segment with the originating and terminating IP address. This block of data is now called a **packet**. At this time, by referring to the **routing table**, the decision is made as to which interface is used. A host may have multiple Internet interfaces in order to connect to different networks, thus a decision must be made as to which interface to use. The packet is then passed on to the specific interface Data Link Layer.

The Data Link Layer determines the maximum length of the data that may be transported across the media; thus the segment is further divided into smaller units. This is a variable length, depending upon the quality of the media, and may change while the data is being transported. The maximum length is set by the MTU for a given interface, but is typically much smaller for the initial size, typically in the range of 400 bytes. The destination and source **MAC address**, along with the data type and length, are prefixed to each unit of data, which is then called a **frame**. The frame is then passed on to the Physical Layer.

The electrical format of the frame is not proper for the media. Thus the Physical Layer converts the signal format, but not the data content, to make it transportable across the media. The proper voltage requirements, signal timing, and synchronization requirements are used to modify the signal. The frame is then sent out on the media, such as **Unshielded Twisted Pair** (**UTP**) cabling for the Ethernet interface. Other media formats may also be utilized for Ethernet or other frame protocols.

At the receiving host, the signal flows up the 7 layers, and the reverse process takes place. From the Application Layer, the data is stored onto the users host, such as on a hard drive.

Could the process have been simplified and the whole process performed in one action? Yes, it very easily could, but if modifications, or corrections to the software would have been more difficult. Additionally, this process allows different types of interface to be easily plugged into the stack. Thus interchangeability and updating of the code make the layered design much easier to work with, more code in the long run, but easier to modify.

8.2 Transport Control Protocol / Internet Protocol

The Internet functions on a large suite of protocol standards, commonly referred to as TCP / IP. These protocols are used for the various applications and for the connectivity between the 7 layers of the OSI model. The Transport Control Protocol set are responsible for the data transfer between the layers, and the Internet Protocol provides for the addressing between different hosts on different networks.

More specifically, the TCP / IP protocols are really focused on Layers 3, 4 and 5, where the specific application, addressing and transport mechanism. Layer 3, Network, utilizes the Internet Protocol portion, providing an addressing mechanism for various hosts to be interconnected. In Layer 4, the various applications are assigned a specific port number, thus allowing one host application to communicate with the same application on another host. The combination of the IP address and the **port number** is called a **socket**, and is commonly show, as an example, 192.168.1.1:80, which would be for a http (web) service from the host. Two methods are used to transport information across the network, TCP and **UDP** (**User Datagram Protocol**). Both provide the same functionality, but differ in the reliability.

TCP provides for error detection of each packet that is sent across the network. If an error should be detected, a Negative Acknowledgment (NAK) is returned back to the originating host, indicating that the pack must be retransmitted. Thus this provided higher reliability to the total communications. The terminating equipment provides error detection, as intermediate equipment does not do error detection. Thus we commonly refer to TCP as a "**Connection – Oriented**" Protocol. In a simple perspective, every time a packet is received, if it is good, then an **Acknowledgment** (**ACK**) is transmitted, or if it is bad, then a NAK is transmitted. Now in the real world it is more complicated, but the basic rule applies.

A UDP connection does not provide for error detection. Thus when a packet is transmitted to a remote host, if an error occurs, or if the packet is dropped, then it is just tough luck, a NAK message is not transmitted. If the originating host does not receive a response to its query within a specified period time, then it retransmit the query. Because the connection is not guaranteed, we refer to UDP as a "**Connectionless – Oriented**" Protocol. Due to this, UDP is most commonly used on a local network rather than on the Internet, although there are Internet applications that do use it.

8.3 Service Ports

Previously mentioned was the Port. Each service that the host provides is issued a Port number, between 1 and 65,536. The first 1024 port numbers are classified as "**well known ports**", that is, they have been assigned by the Internet powers to be, known as IANA (**http://www.iana.org**), and may not be used for any other function. Example of some common port assignment include:

Port	Service
20	FTP data
21	FTP setup
22	SSH
23	Telnet
25	SMTP (Email Server)
53	Domain Name Service
80	HTTP (Web Server)
110	POP3 (Post Office Protocol – Email Collection)
220	IMAP3 (Internet Message Access Protocol – Email Collection)
520	RIP Protocol

This is just a short list of the first 1024 values (there are unassigned values). There are many values greater than 1024 that have been assigned, referred to as being "**reserved ports**", but they are not protected, in that another application may utilize any value. In Linux, the list of assigned values is maintained in the **/etc/services** file. Before an application establishes a session between itself and another host, it looks to the services file to insure that the desired value is not assigned.

The process of establishing a connection between two hosts requires several steps. This is performed to insure that a connection can be made. The process follows the following sequence; assume that host A is originating a connection to host B:

Originator	Terminator	Message
Host-A	Host-B	Request to Port 80, respond Port 32,400 (SYN)
Host-B	Host-A	Acknowledge to Port 32,400 from Port 80, OK (SYN)
Host-A	Host-B	Acknowledge Acknowledgement to Port 32,400
Host-A	Host-B	Request http Port 80 from Port 32,400

Host-B	Host-A	Acknowledge http request with data to Port 32,400
Host-B	Host-A	From Port 80 to Port 32,400, page complete
Host-A	Host-B	To Port 80 from Port 32,400, page acknowledgment
Host-B	Host-A	From Port 80 to Port 32,400, Finished
Host-A	Host-B	From Port 32,400 to Port 80, Finished
Host-B	Host-A	From Port 80 to Port 32,400, Acknowledgment

So to transmit a simple web page, at least 10 messages have flowed across the Internet. If a page includes graphics and a lot of text, then the number of packets may increase dramatically.

8.4 Internet Control Message Protocol

The Internet Control Message Protocol (ICMP) provides a very special service to the Internet. In simple ways, it is the message handler between two systems. It is one part of the Layer 3 Network OSI model. Typical messages include error, flow control, and information messages required by the operating system. Error messages include Acknowledgment (packet received correctly) and Negative Acknowledge (packet in error). Flow control is provided to notify if the terminating host is able to receive packets; if the terminating host buffers are full, then a stop message is transmitted, and will not be continued until a continue message is received.

There are two special types of information messages that are used in testing a network, **ping** and **traceroute**. Although they are similar in function but have a minor difference. For a ping, an **echo** message is transmitted to a remote host, called an **echo request**. The remote host, upon receipt of the echo message, replies with an **echo reply** message. The originating host measured the time from when it transmitted the ping to the time it received an echo reply. This is known as the **Round Trip Time** (**rtt**). This information is then displayed in the response to the ping utility.

A traceroute is basically the same, except that the value of the "**Time To Live**", or **TTL**, is modified and it utilizes a different protocol number. In a ping, the TTL is normally set to 30, or 30 hops. This means that the ping packet will die after either 30 seconds or after having passed through 30 routers, whichever occurs first. For a traceroute packet, the first packet is transmitted with a TTL set to 1; the first router decrements it to 0, notices that it cannot forward the packet, and returns an echo reply message with the information that the packet was killed. The next packet that is transmitted has a TTL of 2; when it passes through the first router, the TTL is decremented by one to the value of 1, and forwarded to the next router. The second router receives the packet, decrements to 0, and returns a message that the packet was dropped. This process continues until the maximum TTL value is transmitted, typically 30. With this data, the traceroute application displays the returning host name, IP address, and round trip time. Each remote site is pinged three times for statistical purposes.

8.5 IP Version 4 Addressing

So far we have been discussing the flow of data between various hosts, but we need to come up with some sort of addressing mechanism. Today, the primary method of addressing is the Internet Protocol, commonly referred to as "IP".

The IP address consists of 32 binary bits, segmented into 4 **octets** consisting of 8 bits each. We commonly convert each octet into a decimal number for human use, but the computer only operates with binary. In decimal, the value of an octet may range from 0 to 255, or a total of 256 values. It is represented in the following format:

Decimal Format: 0~255 . 0~255 . 0~255 . 0~255
Binary format: XXXXXXXX . XXXXXXXX . XXXXXXXX . XXXXXXXX

Knowledge of how to convert between decimal and binary is a strong plus in understanding how the Internet addressing functions is a must. The basic start for understanding the conversion starts with the following table:

Binary Position	Decimal Value
1	1
2	2
3	4
4	8
5	16
6	32
7	64
8	128

or

Bit Postion							
8	**7**	**6**	**5**	**4**	**3**	**2**	**1**
Decimal Value							
128	64	32	16	8	4	2	1

Look at the pattern; each increment in binary position causes a doubling in the decimal value. It is very important that the first 8 bit positions be memorized, and to understand how to compute larger values.

8.5.1 Decimal – Binary Conversion

In order to properly understand the creation of decimal numbers for either the address or the mask, one must have a simple understanding of the process of converting between the decimal numbers to binary numbers. The value of the bit positions is as follows:

For every bit position that has a value of ONE, we add the value of the decimal value. For example, the binary number of 00101011 would be:

```
0 x 128     = 0          most significant bit (left bit)
0 x 64      = 0
1 x 32      = 32
0 x 16      = 0
1 x 8       = 8
0 x 4       = 0
1 x 2       = 2
1 x 1       = 1          least significant bit (right bit)
decimal value = 43
```

Thus, if we have an IP address of 192.168.102.149, the binary representation will be:

11000000 . 10101000 . 01100110 . 10010101

If you need to convert a decimal number to binary you select the largest decimal value from the binary table that, when subtracted from the decimal value will not give a negative result, and subtract it from the specified decimal value. While keeping track of the binary position, keep subtracting progressively smaller values of binary numbers until one has a zero result. For example, give the decimal value of 221, this would convert into the binary value of:

```
 221
-128          Bit 8 = 1
  93
 -64          Bit 7 = 1
  29
 -16          Bit 5 = 1, but Bit 6 = 0
  13
  -8          Bit 4 = 1
   5
  -4          Bit 3 = 1
   1
  -1          Bit 1 = 1, but Bit 2 = 0
   0
```

Thus the decimal value of 221 converts to binary value of 11011101.

As another example, convert the decimal value of 98 to binary:

```
 98
-64          Bit 7        Bit 8 = 0
 34
-32          Bit 6
  2
 -2          Bit 2        Bits 5, 4, 3, and 1 = 0
  0
```

Thus the decimal value of 98 converts to the binary value of 01100010.

Note that for Internet addressing, we need to place a leading “0” in the binary value in order to fully represent all 32 bits.

8.5.2 Network and Host Address

We can separate an IP number into two groups of numbers – the left portion represents the network domain address and the right portion represents the individual host address.

Depending upon the size of a specific local host network, which may be between 4 and many thousand hosts, the host address will vary in its length. In using the binary format, the minimum size of the local network address will be the right most two bits, with the maximum being 30 bits.

Minimum Local Network:
NNNNNNNN . NNNNNNNN . NNNNNNNN . NNNNNN*XX*

Maximum Local Address:
NN*XXXXXX . XXXXXXXX . XXXXXXXX . XXXXXXXX*

8.6 Subnet Mask

When you have specified an IP address for a host, you may recall that you also needed to specify a Subnet Mask. So how do we differentiate the network address from the host address? In order to know where the Network Address ends and the Host Address begins, we need to apply a mask that separates the two. The mask duplicates the Network address in length, and for the Network Address portion each bit is set to a ONE. Most commonly, but definitely not exclusively, the Subnet Mask will be 255.255.255.0, or:

11111111 . 11111111 . 11111111 . 00000000

For example, if the Network Address is 20 bits long, with the remaining 12 bits being the Host Address, the mask will look like:

11111111 . 11111111 . 11110000 . 00000000
\---------- 20 Ones ----------/\---- 12 Zeros ----/

This is called the **Subnet Mask**. Converting this to the decimal format in this example we have:

255 . 255 . 240 . 0

In a new method of the displaying the Subnet Mask, instead of writing out the decimal value, we write down the number of ones behind a slash character. Thus, we would represent the above as “/20”. This format is called **CIDR**.

The way we use the Subnet Mask to differentiate between the network address and the local address. The network address is commonly referred to as the **domain address**, and we commonly give it a **Fully Qualified Domain Name**(**FQDN**). For example the fully qualified domain name for the IP address of 66.94.234.13 and 216.109.112.135 is yahoo.com. Unfortunately, when you query the Internet for an IP address for a FQDN, it does not return the Subnet Mask.

The portion of the Subnet Mask that comprises all “1’s” is called the NETWORK portion. The portion of the Subnet Mask that comprises all “0’s” is

called the Host portion. An imaginary line separating the two sections divides them.

```
        Network Address      |   Host Address
11111111 . 11111111 . 1111 | 0000 . 00000000
                           |
```

Whenever we specify a Host Address, we must also specify the Subnet Mask in order to differentiate between the Network address and the Host address portion. In order to determine what the network address is for a specific host address, we need to convert the host address and Subnet Mask to binary, and then perform a **Boolean AND** function.

The **ANDing** of two binary numbers results in a **true** condition only if all inputs are **true**. We commonly use a **Truth Table** to illustrate how the function works.

Input		Output
A	**B**	**C**
0	0	0
0	1	0
1	0	0
1	1	1

A & B = C

We can see that the output C is only true if both **A and B** inputs are both a **1**.

Let's see how this applies to the Subnet Mask. We take the Subnet Mask and AND it with the IP address – one bit at a time. This process requires that both IP address and the Subnet Mask be converted to their binary format value. For an example, say we have a host address of 210.56.75.33 with a Subnet Mask of 255.255.240.0; we then perform the following:

```
      210          . 56         . 75         . 33
      11010010     . 00111000   . 01001011   . 00100001   Address
      255          . 255        . 240        . 0
      11111111     . 11111111   . 11110000   . 00000000   Subnet Mask
&     11010010     . 00111000   . 01000000   . 00000000   Network Address
      210          . 56         . 64         . 0
```

Thus we observe that the first address of the local network is 210.56.64.0. Observe that the third octet one observes only the first four bits in the address – 0100. There are four values that the third octet may be:

Binary Address	**Base Address**	**Decimal Network**
0000 0000	0	
0001 0000	16	
0010 0000	32	
0011 0000	48	

0100 0000	64	Start of our address block
0101 0000	80	
0110 0000	96	
0111 0000	112	
1000 0000	128	
1001 0000	144	
1010 0000	160	
1011 0000	176	
1100 0000	192	
1101 0000	208	
1110 0000	224	
1111 0000	240	

Only the first four bits of the third octet are allowed to change, the remaining four bits must remain as “0” because they are part of the local host address.

The network address range for the example network can therefore range from 64.0 (third and fourth octet) to 80.0 MINUS 1, or 79.255 (the largest value in the fourth octet is 255, which is one less than the next higher binary value – 256, or ninth binary bit value).

Our network address range is therefore:

210.56.64.0 ~ 210.56.79.255

One might ask why the high address ends with 210.56.79.255. Note from the above table that the next network address was 210.56.80.0; the address immediately preceding this is the 79.255. Remember that the largest value in a specific octet is 255, the next larger value will increment the next octet and the present octet is 0. The first address is the **Network Address**, the last address is known as the **Broadcast Address**.

This is a large range of addresses for our example, allowing for 2048 assigned host addresses. But we do have another requirement for the addressing, two addresses in our range are reserved, the network address of 210.56.64.0 and a broadcast address of 210.56.79.255. The first is reserved for the network in general; one way to remember that it is reserve (although not correct) is to assign the actual cable an address. The cable address must be the first one available, the network address. The second address, 210.56.79.255 is used when one want to send a message to all other hosts on the local network segment; a message is broadcast to all other hosts. Another way of specifying this is that the local host address of all zero’s is reserved for the network, and an all one’s is reserved for a general broadcast.

A general broadcast of **ALL ZEROS** is used to send a message to everyone on the specific network. A general broadcast of **ALL ONES** is used to send a message to everyone on a single segment of a specific network. The difference between the two is that a specific network may consist of several routed segments, where an ALL ZEROS will hit all network Hosts and an ALL ONES will only hit the one segment that the issuing Host is on and not pass through the router. An ALL ONES address is referred to as a Broadcast, and an ALL ZEROS is referred to as a Multicast.

8.7 Reserved IP Addresses

There are four sets of addresses that are reserved for special functions. These are:

127.0.0.0 (all of the 127 address range)
10.0.0.0 (all of the 10 range)
172.16.0.0 through **172.32.0.0**
192.168.0.0 (all of the 192.168 range)

The address 127 is reserved for **loopback** purposes and testing of the OSI model stack. This address does not test the external network or the Network Interface Card, only the OSI Layers 3 through 7. The whole 127 address range is reserved for loopback, but we generally only use the address 127.0.0.1.

The addresses 10., 172.16. ~ 172.31., and 192.168. have been set aside as **private addresses**, that is, addresses that are not legitimate on the real network, but are valid on a private network. The gateway must translate them to a valid real address through a process called **Network Address Translation** (**NAT**). This exists where a small business can have a small number of registered public addresses, and the remaining stations have private addresses. By using this technique, we can significantly expand the number of actual workstations that are able to utilize the Internet, and multiple businesses can use the same set of addresses, major saving of address space to the Internet.

8.8 Classfull and Classless Subnetting

When the Internet addressing scheme was originally created, no one could imagine that it would become commercial, it was originally designed for the military – research arena, a network between various universities that were doing research and design for the government. What was implemented, as the network became commercial, was a network that was designed on the basis of the following:

Address Range			
1.0.0.0	~	127.255.255.255	this is really 126.255.255.255 because 127 is reserved
128.0.0.0	~	191.255.255.255	
192.0.0.0	~	223.255.255.255	
224.0.0.0	~	239.255.255.255	
240.0.0.0	~	255.255.255.255	

Look at the binary conversion for the start of each address class:

0.0.0.0	00000000.00000000.00000000.00000000
128.0.0.0	10000000.00000000.00000000.00000000
192.0.0.0	11000000.00000000.00000000.00000000
224.0.0.0	11100000.00000000.00000000.00000000
240.0.0.0	11110000.00000000.00000000.00000000

Look at the beginning address for each of the above addresses, namely the first octet.

Address Range	**First octet First Four Bits**
0 ~ 127	0000 ~ 0111
128 ~ 191	1000 ~ 1011
192 ~ 223	1100 ~ 1100
224 ~ 239	1110 ~ 1110
240 ~ 255	1111 ~ 1111

Specifically, notice the first four beginning bits of the octet. Do you see the progression?

0000
1000
1100
1110
1111

Initially, these addresses were given a special designation, called "**Class**".

Class A	0.0.0.0	~	127.255.255.255
Class B	128.0.0.0	~	191.255.255.255
Class C	192.0.0.0	~	123.255.255.255
Class D	224.0.0.0	~	239.255.255.255
Experimental	240.0.0.0	~	255.255.255.255

Class A addresses were reserved for the largest companies, such as AT&T, General Motors, General Electric, Boeing, HP, and others. Each of these companies had an address range of 2^{24} hosts. The Subnet Mask for a Class A address is 255.0.0.0.

Large Companies that had a host requirement greater than 254 were assigned a Class B address, allotting them 2^{16} hosts. Few companies really required than many. The Subnet Mask for a Class B address is 255.255.0.0. Many ISPs would fall into this category.

Small companies that had a host requirement of less than 254 were assigned a Class C address, allotting them 2^{8} hosts. The Subnet Mask for a Class C address is 255.255.255.0. Most small businesses fall into this category.

When transmitting to a designated set of hosts on a network, it is called a **multicast**. The address range reserved for this is a Class D. The Subnet Mask for a Class D address is 255.255.240.0. This is a special functional address range, something like a broadcast, but only designated hosts will receive the message. An example of use might be to a set of sales agents, that need to be updated with messages relevant to a those specific agents.

Finally, some space must be reserved for those that wish to tinker and experiment. The rest of the address range is allocated for that.

It became evident in the early 1990's that with the commercialization of the Internet, there would not be enough addresses available for everyone using the Class oriented address method. To overcome this problem, new addresses were allocated in what was called a **Classless** mode. This does away with the Classfull addresses and the designated Subnet Masks.

The new format allowed a floating Subnet Mask where the network side could be any number of “1’s”, ranging from 2 to 30. The companies that were previously issued a Class A address had to give up a large range of addresses that were not being used (no problem), and these were re-assigned to companies that needed the address space, with a smaller range of addresses. For example, a company that had been assigned the Class A space of 20.0.0.0 with a Subnet Mask of 255.0.0.0 might have been re-issued the address of 20.0.0.0 with a Subnet Mask of 255.255.240.0, which gave it an address range of 20.0.0.0 to 20.0.15.255. The address range from 20.0.16.0 to 20.255.255.255 may be allocated to a multitude of other companies in any reasonable requirement.

8.9 Networking Equipment

The ability for a host to network with other hosts is based both the OSI Data Link and Physical Layers, and various hardware devices.

8.9.1 Data Link Layer

The Data Link Layer is responsible for formatting the data, which is in packet format, to a format that can be transmitted across the media. The format is called a frame.

All hosts on a single segment network are addressed via the Media Access Control (MAC) address. This address consists of 48 bits, and is divided into two parts, vendor and card number. The first 24 bits designate a specific vendor, and are issued by IANA. Every vendor has at least one address, and some have many. The last 24 bits are like a serial number for the card, and must be unique to that card. Communications between different hosts on a given network segment must be addressed by using the MAC address, not the IP address.

The Data Link Layer encapsulates data into a frame; the MAC address of the destination host and the MAC address of the originating host are prefixed to the packet. The type of frame and the frame length are also inserted.

8.9.2 Physical Layer

Before the data can leave the computer, the electrical characteristics of the signal must be modified for the specific media. For instance, Ethernet requires a signal with a specific voltage range and be Manchester encoded for timing, whereas a fiber optic interface must generate a light signal in order to transmit the signal. The Physical Layer of the OSI model is strictly a hardware issue, whereas all other Layers operate as software modules.

8.9.3 Network Interface Card

On each host that is to communicate with other hosts is a Network Interface Card, commonly referred to as a “**NIC**”. On the NIC card is a combination of hardware and software. All of the Physical Layer requirements are located on the NIC card, as are parts of the software requirements of the Data Link Layer. The MAC address for the card is burned into the ROM on the card. During the computer boot process, the MAC address is copied into RAM in order to allow

the host to operate faster. This means that the MAC address can be changed with the correct utility.

8.9.4 Data Equipment

Data equipment comes in two flavors, those that originate and terminate a signal, and those where the signal passes through the equipment. What is important about the difference is the way the interfaces are wired.

8.9.4.1 Data Terminal Equipment

Data Terminal Equipment, or **DTE**, is a host that either originates or terminates a signal. A workstation or server is a DTE device. For an Ethernet interface the Transmit Out signal (“goesouta”) is on pins 1 and 2. The Receive In signal (“goesina”) is on pins 3 and 6.

8.9.4.2 Data Communications Equipment

Data Communications Equipment, or **DCE**, is a device where the information passes through it without being modified or analyzed. Such a device is a modem, hub or MAC switch. Note that a signal's electrical characteristics may be modified, but the information is not. For an DCE Ethernet interface, the Transmit IN signal (“goesina”) is on pins 1 and 2. The Receive Out signal (“goesouta”) is on pins 3 and 6, the same as that for another host.

Interestingly, a router, where the signal does pass through, but is analyzed for the IP address, is a DTE interface device. The information, that being the user data and header, is modified. Look as a router as a device that terminates the media signal format or protocol; since it must be modified and then regenerated into a new format.

8.9.5 Address Resolution

We have been discussing two different addressing methods, Internet Protocol and MAC. On the local network segment the MAC address is used, but in order to interconnect different segments, a router is used. MAC addresses are physical and are burned into the NIC card during manufacturing; IP addresses are logical and are assigned to a host by a user.

As a user, you know another host’s IP address, but you do not know the MAC address. In order to convert, a special protocol called **Address Resolution Protocol** (**ARP**) is used. An ARP packet is transmitted onto the local segment with the message something like “Who is IP address 192.168.1.1” from MAC. The host with the IP address of 192.168.1.1 responds back with “I am 192.168.1.1, and my MAC address is …”. The originating host then stores the IP and MAC address for future use.

And for the fun of networking, you can remember that the ARP is the Network Seal of Approval.

8.10 LAN Ethernet Equipment

An Ethernet Local Area Network consists of several physical devices, which are used to transmit the frame between the different hosts.

8.10.1 Host Device

A host device is any computational device. This can include a workstation, server, router, gateway, or network switch.

8.10.2 Network Interface Card

Each host utilizes a Network Interface Card (NIC) to interface the physical media, or twisted pair cabling in the case of Ethernet. (Yes, coax was used, but lets just forget it for this discussion.) As noted previously, the NIC conditions the data signal for transmission on the media.

8.10.3 Host Interconnectivity

The various hosts must be interconnected together. This is accomplished by one of two devices. Both perform the same function, but allow differences in performance.

A **Hub** is the older technology used to interconnect various hosts. Any signal received from one host is automatically rebroadcast out to all other hosts. What comes in from one goes out to all. The problem with this is that if two hosts transmit at the same time, the signals will overlap, which is termed a **collision**. When this happens, the data from each host must be retransmitted; each host initiates a random delay and retransmits, which normally insures that another collision does not occur.

Newer technology utilizes a MAC layer **Switch** to interconnect the various hosts. When a signal is received from one host, the destination MAC address is analyzed to determine which switch port the destination host is located on. The benefit of this process is that there are no collisions, but the drawback is that it takes additional time to process.

8.10.4 Inter-Network Interface

In order for one network segment to communicate with another, a routing or switching device is required. Using today's technology, the addressing protocol between segments is the **Internet Protocol**, or more commonly known as **IP**. Within each frame is encapsulated the IP packet for the destination; when the frame enters the router, the MAC address is stripped off of the frame and the IP address of the packet is analyzed. It is then routed to the proper interface for forwarding to the destination host.

Properly, a router is capable of only routing the same data that uses the same addressing / port assignment protocol from one port to another. If a translation in protocol is required, such as when translating between IP and Novell's IPX addressing format, then a gateway device is required. So what is the big difference between a router and a gateway? There are two parts to the answer. The first answer is really simple – additional software. The second is that the translation between IP and IPX is performed at the Application Layer rather than at the lower layers. By today's technology, the software for

translating from one transmission protocol to another is included in a router by default, so every router is really a gateway. The extra difference between a router is that the extra software modules that are needed for an additional variety of protocols are included with a gateway, whereas they are not included with a router. The most common protocols included in a router are Ethernet, Token Ring, HDLC, and PPP, although additional protocols may be included by some of the better vendors. The various protocols are not covered here.

8.11 IP Configuration

Naturally, with all of the equipment, before the host can work, it must be configured. In order to configure Red Hat or Fedora Core Linux (or virtually all other systems), four Internet address values are required.

1. Host IP Address
2. Host Subnet Mask
3. Gateway Address
4. DNS Address(s)

All four of these values may be easily assigned using the GUI utility, by selecting:

Menu □ System □ Network Configurator

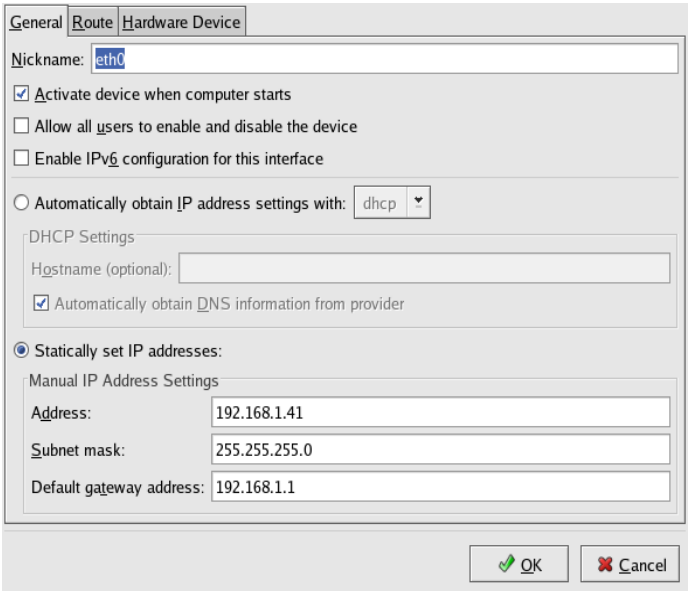

Figure 8-1: Network Configurator

Start by selecting the desired interface that is to be set. This most commonly will be the Ethernet – eth0. Click the Edit Icon. This opens the General configuration screen.

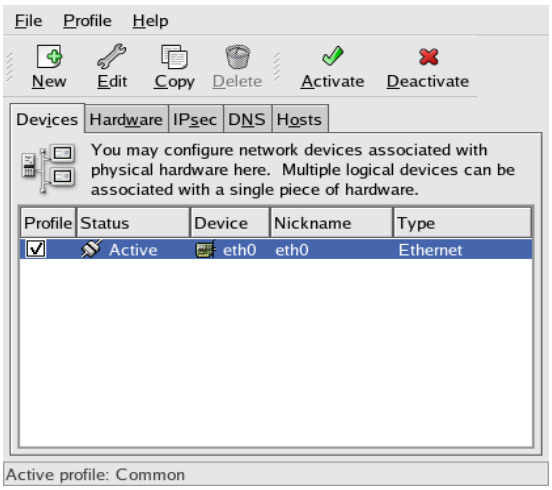

Figure 8-2: General Configuration

The settings that should be set include:

1. Activate device when computer starts.
2. Select either:
   a. Automatically obtain IP address settings with dhcp
      or
   b. Statically set IP address.
      And set:
      i. Host IP Address
      ii. Network Subnet Mask
      iii. Default gateway address

No information is required to be entered in the Route or Hardware Device tabs. Click OK. This returns one to the original window.

No information is required to be entered in the Hardware tab.

No information is required to be entered in the IPsec tab.

Click on the DNS tab, which opens the DNS entry window.

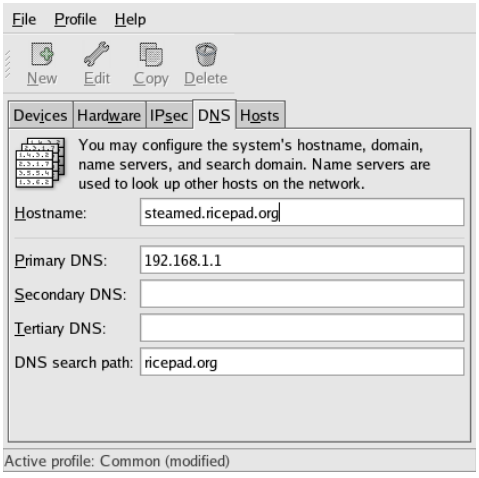

Figure 8-3: DNS Configuration

Enter the following data:

1. Fully Qualified Hostname
2. At least one IP address for your DNS server.
   The ISP must provide this address.
3. DNS search path, this is typically your domain name.

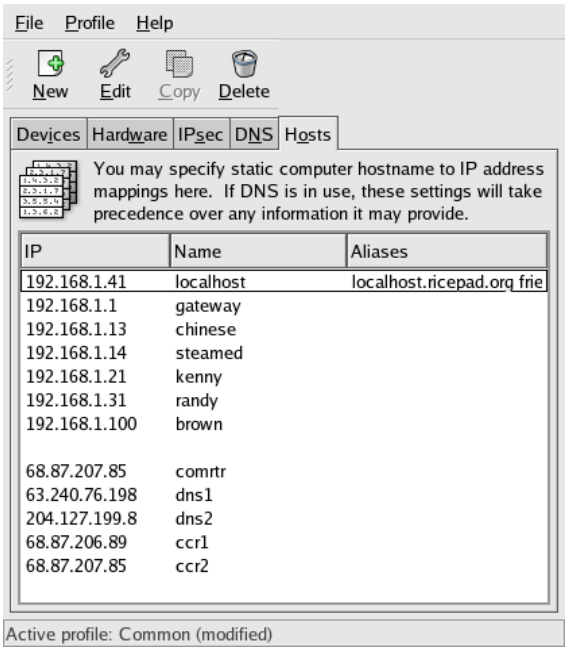

Figure 8-4: Hosts Table Contents

Optionally, the Hosts TAB may be selected, where one may make additions to the **/etc/hosts** file.

To make a new addition, click on the New Icon to open a new window and make the entry. A given entry may also be edited, by clicking on the Edit Icon.

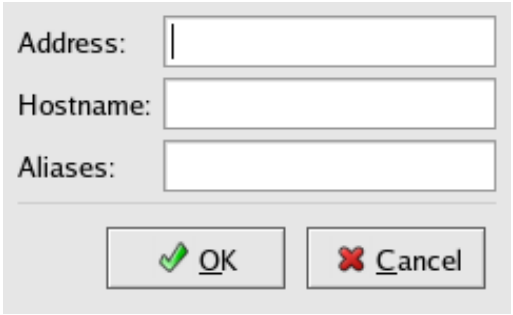

Figure 8-5: Hosts Table Entry

When complete with editing of the Hosts table, return to the Devices tab. The interface may now be bounced by clicking on the **Deactivate** and then **Activate**. The interface should now be active. When deactivating, a window will open saying that a change has been made, and requesting a confirmation that it should be saved. Click Yes. Another window will open, confirming that the

interface is being deactivated, click OK. At this point, the Activate icon must be clicked. This opens the window that shows errors, if any exist (hopefully the window closes with no information).

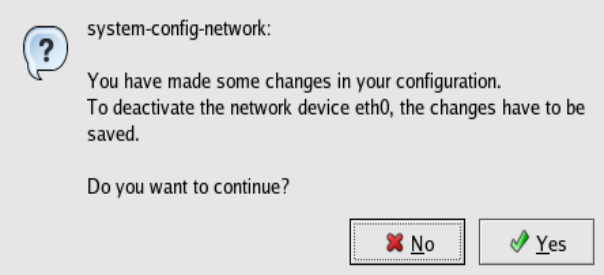

Figure 8-6: Interface Activation

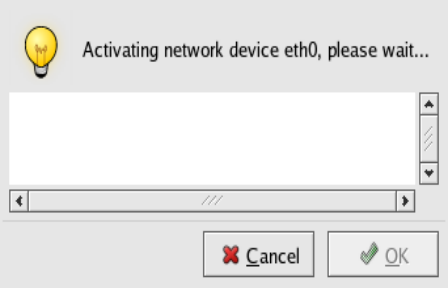

Figure 8-7: Activating Interface

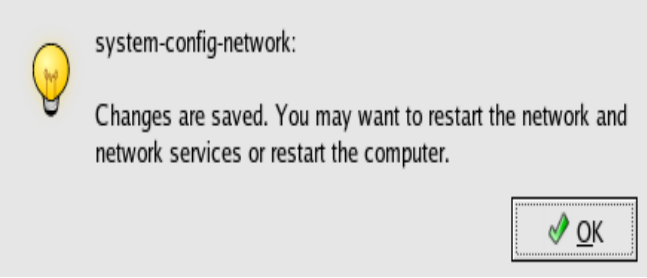

Figure 8-8: Saving Interface Changes

The interface should now be operational.

OK, all of the above was set up using the GUI interface, but what about doing it the same from the Command Line. It is not that hard; just edit the following files:

/etc/sysconfig/network-scripts/ifcfg-eth0

***[root@steamed network-scripts]#* cat ifcfg-eth0**

DEVICE=eth0
BOOTPROTO=none
BROADCAST=192.168.1.255
IPADDR=192.168.1.41
NETMASK=255.255.255.0
NETWORK=192.168.1.0

ONBOOT=yes
TYPE=Ethernet
USERCTL=no
PEERDNS=yes
GATEWAY=192.168.1.1
IPV6INIT=no

The above is for one of my computers on my local home LAN (the one configured in the GUI mode). (What, you have never heard of steamed rice?) When the file was activated, the above file was created. If one had selected to obtain an IP address from a dhcp server, then the BOOTPROTO line would have been set to "dhcp". Note that three of the four values that are required are entered into this file.

/etc/resolv.conf

***[root@steamed etc]#* cat resolv.conf**
search ricepad.org
nameserver 192.168.1.1

In this file, we see that the "ricepad.org" domain is to be searched first, and the DNS nameserver address is specified. In this particular case, the host is searching to the local router / firewall; the router has the Internet DNS IP addresses for additional searching. Your system may need to specify the IP address for your ISP DNS server.

8.12 Network Testing

OK, you have finally hooked everything up. If everything is working correctly, then it all works – but what if it doesn't? Where does one start looking for the problem?

8.12.1 ifconfig

The first test to make is to see if the interface is active. Issue the command:

ifconfig

***[root@steamed etc]#* ifconfig**
eth0 Link encap:Ethernet HWaddr 00:50:8D:6A:D0:F9
inet addr:192.168.1.41 Bcast:192.168.1.255
Mask:255.255.255.0
inet6 addr: fe80::250:8dff:fe6a:d0f9/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500
Metric:1
RX packets:29398 errors:0 dropped:0 overruns:0 frame:0
TX packets:28210 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:8154046 (7.7 MiB) TX bytes:3117670 (2.9 MiB)
Interrupt:11 Base address:0xc000
[remainder deleted]

Observe that the assigned IP address is 192.168.1.41 and the Subnet Mask has been assigned to 255.255.255.0. Also note that the MAC address, 00:50:8D:6A:D0:F9 is also displayed. Finally, note that the interface is active – "UP". If the NIC card is not connected, the status will be "DOWN".

8.12.2 LAN LEDs

Whenever one has a problem, the first thing that should be checked is the cabling. This is verified by looking on the LEDs that are typically on the NIC card; if the network is operational, there should be a steady light. There may also be a blinking LED, indicating that data is being either transmitted or received (that is a very good sign).

On some Hubs and MAC Switches, there may also be LEDs indicating the operational status of the interface. This LED should be either on or blinking.

If the LEDs are off, then there is a definite cabling problem. Additional testing of the cabling is required. One of two problems typically exist, either the wire is broken, or the connectors are incorrectly crimped. Cabling problems are definitely a different topic, and therefore not discussed here.

8.12.3 Network Ping

One of the most valuable test utilities is the **ping**. An ICMP packet is transmitted, and the response is recorded. There are two ping tests that should be performed, OSI stack and external host.

Issuing a ping to the local loopback, IP address 127.0.0.1, tests the OSI Layers 7 through 3. It does not test the Data Link or Physical Layers because the ICMP function is located within the Network Layer. (Recall that to terminate a ping, click CTRL-C.)

```
# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.045 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.040 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.039 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.040 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.040 ms
{click ^C}
--- 127.0.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3998ms
rtt min/avg/max/mdev = 0.039/0.040/0.045/0.008 ms, pipe 2
```

In order to test the Data Link and Physical Layers, a ping must be made to another host on the local network; the best test is to the local router. In the following example, a test is made to my local router / firewall.

```
# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=0.319 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.291 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.299 ms
```

{click ^C}
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.291/0.297/0.319/0.015 ms, pipe 2

You might observe several differences from the above to that displayed by Microsoft. First, Microsoft automatically terminates after four pings, for Unix and Linux, one must manually stop the ping (for more features, see the man page). Second, the **Round Trip Time** (**RTT**) ("time = ...) is displayed in thousands of a millisecond, whereas Microsoft displays only in values greater than 10 milliseconds. Last, the mean deviation is displayed for statistical purposes, Microsoft does not provide this information.

8.12.4 Network Traceroute

Another test that performs a similar test to a ping is the **traceroute**. This is basically the same process, but allows one to see all of the equipment that is transversed to the destination.

[root@steamed etc]**# traceroute ccr1**
traceroute to ccr1 (68.87.206.89), 30 hops max, 38 byte packets
1 gateway (192.168.1.1) 0.282 ms 0.220 ms 0.216 ms
2 * * *
3 ccr1 (68.87.206.89) 11.527 ms * 8.818 ms

Notice that since I have an IP address in the /etc/hosts file, I was able to use the name for the site rather than the IP address. To exit, click the ^C keys.

8.12.5 Active Traceroute

Once upon a time, a gentleman by the name of Mike wrote a small program to provide an updated, real time display of traceroute pings. The utility is called **mtr** (appropriately named of course). The updates will continue until you manually stop the application (^C).

```
# mtr
                                          Matt's traceroute  [v0.54]
steamed.ricepad.org                         Fri Sep 23 22:46:35 2005
Keys:  D - Display mode    R - Restart statistics    Q - Quit
                          Packets               Pings
Hostname               %Loss  Rcv  Snt  Last Best  Avg  Worst
 1. ccgw.dearroz.net     0%   24   24     0    0    0      0
 2. ???
 3. ccr1                 0%   23   23     9    7    9     27
```

8.12.6 Network Status

Various network servers ports may be displayed as to which are active or have a connection made to them. This is viewed using the **netstat** command.

This listing can be quite long, so the following is highly abbreviated. I have highlighted some values and inserted an explaination.

```
# netstat   -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:32769           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN
...
tcp        0      0 0.0.0.0:445             0.0.0.0:*               LISTEN
tcp        0      0 192.168.1.41:40810      192.168.1.1:22          ESTABLISHED
```

An SSH session is active between the host and the router.

```
tcp        0      0 :::6000                 :::*                    LISTEN
tcp        0      0 :::80                   :::*                    LISTEN
tcp        0      0 :::22                   :::*                    LISTEN
tcp        0      0 :::443                  :::*                    LISTEN
tcp        0      0 ::ffff:192.168.1.41:22  ::ffff:192.168.1.44:1047 ESTABLISHED
```

This is the reverse SSH session from the router to the host.

```
udp        0      0 0.0.0.0:32769           0.0.0.0:*
...
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node Path
unix  2      [ ACC ]     STREAM     LISTENING     7432   /tmp/.s.PGSQL.5432
unix  2      [ ACC ]     STREAM     LISTENING     13661  @/tmp/fam-root-
unix  2      [ ACC ]     STREAM     LISTENING     9678   /tmp/.font-unix/fs7100
unix  2      [ ACC ]     STREAM     LISTENING     13278  /tmp/.X11-unix/X0
...
unix  3      [ ]         STREAM     CONNECTED     6655
unix  2      [ ]         DGRAM                    6532
unix  2      [ ]         DGRAM                    6463
Active IPX sockets
Proto Recv-Q Send-Q Local Address          Foreign Address         State
```

8.13 IP Version 6

So far the discussion has been oriented towards the IP Version 4 (IPv4) Protocol. Back in the early 1990's, IANA realized that at the continuing growth of the Internet, that there would not be enough IP addresses available for everyone. Previous discussion showed one solution to the problem, to establish a "Classless" addressing method. The second method was to establish private addresses. These two methods provided a significant delay in the problem to IP addressing space, but the time is near where the number of Internet addresses will run out. To solve the problem, a new addressing method was developed, IP Version 6. So what happened to version 5, I guess it didn't work, so they just dropped it. At this time, it is estimated that the address space will be totally used up before 2015.

Noted earlier, IPv4 contains 32 bits. IPv6 contains 128 bits. Now that is a big increase! So how does one handle this number?

For a starter, IPv6 is not displayed in decimal format, but rather in hexadecimal format. That means you need to become familiar with Hex codes (but that is not covered here). Each Hex character represents 4 bits, so there are 32 hex digits in the IPv6 address. There are two versions of the IPv6 address to accommodate the transition between the two. The first allows for the present IPv4 address to be incorporated as the last 32 bits of the address, so one only needs to prefix the IPv4 address with the new network IPv6 address. In the future, the last 24 bits of the IPv6 address will be the last six hex digits of the MAC address.

If you look back to the listing for the **ifconfig** command, you will find the line:

inet6 addr: fe80::250:8dff:fe6a:d0f9/64

This is the IPv6 address that has been assigned to the interface. Notice that it is not the total of 32 hex values, but only 19. Writing down a IPv6 address may be abbreviated when a set of 4 characters is all zeros, thus the true address of the interface would be:

fe80:0000:0000:0000:**0250:8dff:fe6a:d0f9**

Notice that the last 6 characters are the same as the last six digits of the MAC address, 6A:D0:F9.

Thus Linux has already implemented IPv6 for future requirements for the mode that accepts the MAC address. IPv6 has already been implemented around the world, except for the United States, where there have been determined to be a few problems that still need to be resolved. The FCC has dictated that it is to be implemented by 2007. It does work, but several factors may cause problems.

You may have also noticed the "/64" value after the IPv6 address, this is the CIDR format of the subnet mask. It means that the first 64 bits of the subnet mask are all 1s. Because the standard format is cumbersome to write out, the CIDR format is now much easier to write. Even though it is easier to write, it still requires one to do the binary computation.

8.14 Commands Used in this Chapter

cat	Displays a file's contents
ifconfig	Displays the configuration of a network interface
less	Displays a file's contents
mtr	Displays the network connection path between two hosts, providing an active refresh of the path
netstat	Displays the network service port status
ping	Test the network connection between two hosts
traceroute	Displays the network connection path between two hosts

8.15 Chapter Review Questions

1. A classless IP Address uses what term to refer to it?
   a. CIDR
   b. Classfull
   c. Netfull
   d. Version 4

2. FTP data is transmitted on which Service Port?
   a. 20
   b. 21
   c. 25
   d. 80
3. A segment is created in which OSI Layer?
   a. Data Link
   b. Network
   c. Session
   d. Transport
4. For a normal ping, what specifies the number of allowable hops?
   a. Average Time
   b. RTT
   c. Timeout
   d. TTL
5. What CLI command is used to set an IP Address on a NIC?
   a. ifconfig
   b. ipconfig
   c. iwconfig
   d. Network Configurator
6. What is a device which passes data through it without modification?
   a. DCE
   b. DTE
   c. Host
   d. Router
7. Which OSI Layer provides a connection between two hosts?
   a. 1
   b. 3
   c. 4
   d. 5
8. What determines the selection of an interface to transmit data?
   a. Application Layer
   b. NIC interface
   c. IP Address
   d. Routing Table
9. An IP Address whose first octet is 221 belongs to which class?
   a. A
   b. B
   c. C
   d. D
10. Which protocol provides no Acknowledgment or error detection?
    a. IP
    b. Socket
    c. TCP
    d. UDP

11. Which OSI Layer is used for the IP Address?
    a. 1
    b. 2
    c. 3
    d. 4
12. IP version 4 contains how many bits?
    a. 16
    b. 32
    c. 64
    d. 96
13. How is a specific Internet service designated?
    a. IP Address
    b. MAC Address
    c. Port Number
    d. Socket Number
14. Which OSI Layer adds the Media Access Control address?
    a. 1
    b. 2
    c. 3
    d. 7
15. A Subnet Mask is used to specify what value of the IP Address?
    a. Host Address
    b. Network Address
    c. Port Address
    d. Socket Address
16. A Frame is created in which OSI Layer?
    a. Data Link
    b. Networking
    c. Physical
    d. Transport
17. What GUI interface provides for the NIC configuration?
    a. ifconfig
    b. ipconfig
    c. Network Configurator
    d. SysConfig
18. What is the Maximum Transport Unit value for Ethernet?
    a. 1000
    b. 1500
    c. 4000
    d. 64,000
19. An IP Address version 4 contains how many octets?
    a. 4
    b. 8
    c. 32
    d. 64

20. IP version 6 contains how many bits?
    a. 32
    b. 64
    c. 128
    d. 1024
21. Which OSI Layer converts the signal format for the connected media?
    a. 1
    b. 2
    c. 4
    d. 6
22. A router interface is which type of networking equipment?
    a. DCE
    b. DTE
    c. Routing
    d. Switching
23. When using the ping command, what specifies the time to for the response?
    a. Hop Count
    b. Deviation
    c. RTT
    d. TTL
24. An MTU of 1500 bytes is used on which media?
    a. Ethernet
    b. Serial
    c. Parallel
    d. Token Ring
25. What utility is used to verify network connectivity?
    a. ping
    b. route
    c. traceroute
    d. tracert
26. Which OSI Layer provides a connection between two hosts?
    a. 1
    b. 3
    c. 4
    d. 5
27. Service Port 25 is used for which Internet service? a
    a. Email
    b. HTTP
    c. POP3
    d. RIP

28. What is the broadcast address for the network address of 170.10.0.0?
    a. 170.10.0.255
    b. 170.10.255.255
    c. 170.10.10.0
    d. 170.255.255.255
29. What protocol is used to relate the IP and MAC addresses?
    a. ARP
    b. IP
    c. MAC
    d. TCP
30. ICMP is part of which OSI Layer?
    a. Network
    b. Presentation
    c. Session
    d. Transport
31. A Socket specifies what entities?
    a. Host name
    b. Internet service
    c. IP Address and Port Number
    d. MAC Address and IP Address
32. A host is which type of Networking Equipment device?
    a. DCE
    b. DTE
    c. Port
    d. Socket
33. An IP Address is a function of which protocol?
    a. IP
    b. Network
    c. TCP
    d. UDP
34. Which protocol is used to specify the Internet service?
    a. ARP
    b. IP
    c. Socket
    d. TCP
35. An IP Address is composed of what values?
    a. Network and Host addresses
    b. Network and Port address
    c. Octet values
    d. Port and Socket values
36. A packet is created in which OSI Layer?
    a. Data Link
    b. Network
    c. Physical
    d. Transport

Chapter Index