

Introduction

Installing the Linux operating system is only the first step toward creating a fully functional departmental server or Web site. Almost all computers are now networked in some way to other devices therefore a basic understanding of networking and issues related to the topic will be essential to feeling comfortable with Linux servers.

This introductory chapter forms the foundation on which the following network configuration and troubleshooting chapters will be built. These chapters will then introduce the remaining chapters that cover Linux troubleshooting, general software installation and the configuration of many of the most popular Linux applications used in corporate departments and Small Office/Home Office (SOHO) environments.

Familiarity with the concepts explained in the following sections will help answer many of the daily questions often posed by coworkers, friends, and even yourself. It will help make the road to Linux mastery less perilous, a road that begins with an understanding of the OSI networking model and TCP/IP.

The OSI Networking Model

The Open System Interconnection (OSI) model, developed by the International Organization for Standardization, defines how the various hardware and software components involved in data communication should interact with each other.

A good analogy would be a traveler who prepares herself to return home through many dangerous kingdoms by obtaining permits to enter each country at the very beginning of the trip. At each frontier our friend has to hand over a permit to enter the country. Once inside, she asks the border guards for directions to reach the next frontier and displays the permit for that new kingdom as proof that she has a legitimate reason for wanting to go there.

In the OSI model each component along the data communications path is assigned a layer of responsibility, in other words, a kingdom over which it rules. Each layer extracts the permit, or header information, it needs from the data and uses this information to correctly forward what's left to the next layer. This layer also strips away its permit and forwards the data to the next layer, and so the cycle continues for seven layers.

The very first layer of the OSI model describes the transmission attributes of the cabling or wireless frequencies used at each "link" or step along the way. Layer 2 describes the error correction methodologies to be used on the link; layer 3 ensures that the data can hop from link to link on the way to the final destination described in its header. When the data finally arrives, the layer 4 header is used to determine which locally installed software application should receive it. The application uses the guidelines of layer 5 to keep track of the various communications sessions it has with remote computers and uses layer 6 to verify that the communication or file format is correct. Finally, layer 7 defines what the end user will see in the form of an interface, be it graphical on a screen or otherwise. A description of the functions of each layer in the model can be seen in Table 2-1.

Table 2-1: The Seven OSI Layers

Layer	Name	Description	Application
7	Application	<ul style="list-style-type: none">The user interface to the application	telnet
6	Presentation	<ul style="list-style-type: none">Converts data from one presentation format to another. For example, e-mail text entered into Outlook Express being converted into SMTP mail formatted data.	FTP sendmail
5	Session	<ul style="list-style-type: none">Manages continuing requests and responses between the applications at both ends over the various established connections.	
4	Transport	<ul style="list-style-type: none">Manages the establishment and tearing down of a connection. Ensures that unacknowledged data is retransmitted. Correctly re-sequences data packets that arrive in the wrong order.After the packet's overhead bytes have been stripped away, the resulting data is said to be a segment.	TCP UDP
3	Network	<ul style="list-style-type: none">Handles the routing of data between links that are not physically connected together.After the link's overhead bytes have been stripped away, the resulting data is said to be a packet.	IP ARP
2	Link	<ul style="list-style-type: none">Error control and timing of bits speeding down the wire between two directly connected devices.Data sent on a link is said to be structured in frames.	Ethernet ARP
1	Physical	<ul style="list-style-type: none">Defines the electrical and physical characteristics of the network cabling and interfacing hardware	Ethernet

An Introduction to TCP/IP

TCP/IP is a universal standard suite of protocols used to provide connectivity between networked devices. It is part of the larger OSI model upon which most data communications is based.

One component of TCP/IP is the Internet Protocol (IP) which is responsible for ensuring that data is transferred between two addresses without being corrupted.

For manageability, the data is usually split into multiple pieces or packets each with its own error detection bytes in the control section or header of the packet. The remote computer then receives the packets and reassembles the data and checks for errors. It then passes the data to the program that expects to receive it.

How does the computer know what program needs the data? Each IP packet also contains a piece of information in its header called the type field. This informs the computer receiving the data about the type of layer 4 transportation mechanism being used.

The two most popular transportation mechanisms used on the Internet are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

When the type of transport protocol has been determined, the TCP/UDP header is then inspected for the "port" value, which is used to determine which network application on the computer should process the data. This is explained in more detail later.

TCP Is a Connection-Oriented Protocol

TCP opens up a virtual connection between the client and server programs running on separate computers so that multiple and/or sporadic streams of data can be sent over an indefinite period of time between them. TCP keeps track of the packets sent by giving each one a sequence number with the remote server sending back acknowledgment packets confirming correct delivery. Programs that use TCP therefore have a means of detecting connection failures and requesting the retransmission of missing packets. TCP is a good example of a connection-oriented protocol.

How TCP Establishes A Connection

Any form of communication requires some form of acknowledgement for it to become meaningful. Someone knocks on the door to a house, the person inside asks "Who is it?", to which the visitor replies, "It's me!" Then the door opens. Both persons knew who was on the other side of the door before it opened and now a conversation can now begin.

TCP acts in a similar way. The server initiating the connection sends a segment with the SYN bit set in TCP header. The target replies with a segment with the SYN and ACK bits set, to which the originating server replies with a segment with the ACK bit set. This SYN, SYN-ACK, ACK mechanism is often called the "three-way handshake".

The communication then continues with a series of segment exchanges, each with the ACK bit set. When one of the servers needs to end the communication, it sends a segment to the other with the FIN and ACK bits set, to which the other server also replies with a FIN-ACK segment also. The communication terminates with a final ACK from the server that wanted to end the session.

This is the equivalent of ending a conversation by saying "I really have to go now, I have to go for lunch", to which the reply is "I think I'm finished here too, see you tomorrow..." The conversation ends with a final "bye" from the hungry person.

Here is a modified packet trace obtained from the tethereal program discussed in Chapter 4, "[Simple Network Troubleshooting](#)". You can clearly see the three way handshake to connect and disconnect the session.

```
hostA -> hostB TCP 1443 > http [SYN] Seq=9766 Ack=0 Win=5840 Len=0
hostB -> hostA TCP http > 1443 [SYN, ACK] Seq=8404 Ack=9767 Win=5792 Len=0
hostA -> hostB TCP 1443 > http [ACK] Seq=9767 Ack=8405 Win=5840 Len=0
hostA -> hostB HTTP HEAD/HTTP/1.1
hostB -> hostA TCP http > 1443 [ACK] Seq=8405 Ack=9985 Win=54 Len=0
hostB -> hostA HTTP HTTP/1.1 200 OK
hostA -> hostB TCP 1443 > http [ACK] Seq=9985 Ack=8672 Win=6432 Len=0
hostB -> hostA TCP http > 1443 [FIN, ACK] Seq=8672 Ack=9985 Win=54 Len=0
hostA -> hostB TCP 1443 > http [FIN, ACK] Seq=9985 Ack=8673 Win=6432 Len=0
hostB -> hostA TCP http > 1443 [ACK] Seq=8673 Ack=9986 Win=54
```

In this trace, the sequence number represents the serial number of the first byte of data in the segment. So in the first line, a random value of 9766 was assigned to the first byte and all subsequent bytes for the connection from this host will be sequentially tracked. This makes the second byte in the segment number 9767, the third number 9768 etc. The acknowledgment number or Ack, not to be confused with the **ACK** bit, is the byte serial number of the next segment it expects to receive from the other end, and the total number of bytes cannot exceed the **Win** or window value that follows it. If data isn't received correctly, the receiver will re-send the requesting segment asking for the information to be sent again. The TCP code keeps track of all this along with the source and destination ports and IP addresses to ensure that each unique connection is serviced correctly.

UDP, TCP's "Connectionless" Cousin

UDP is a connectionless protocol. Data is sent on a "best effort" basis with the machine that sends the data having no means of verifying whether the data was correctly received by the remote machine. UDP is usually used for applications in which the data sent is not mission-critical. It is also used when data needs to be broadcast to all available servers on a locally attached network where the creation of dozens of TCP connections for a short burst of data is considered resource-hungry.

TCP and UDP Ports

The data portion of the IP packet contains a TCP or UDP segment sandwiched inside. Only the TCP segment header contains sequence information, but both the UDP and the TCP segment headers track the port being used. The source/destination port and the source/destination IP addresses of the client & server computers are then combined to uniquely identify each data flow.

Certain programs are assigned specific ports that are internationally recognized. For example, port 80 is reserved for HTTP Web traffic, and port 25 is reserved for SMTP e-mail. Ports below 1024 are reserved for privileged system functions, and those above 1024 are generally reserved for non-system third-party applications.

Usually when a connection is made from a client computer requesting data to the server that contains the data:

- The client selects a random previously unused "source" port greater than 1024 and queries the server on the "destination" port specific to the application. If it is an HTTP request, the client will use a source port of, say, 2049 and query the server on port 80 (HTTP)
- The server recognizes the port 80 request as an HTTP request and passes on the data to be handled by the Web server software. When the Web server software replies to the client, it tells

the TCP application to respond back to port 2049 of the client using a source port of port 80.

- The client keeps track of all its requests to the server's IP address and will recognize that the reply on port 2049 isn't a request initiation for "NFS", but a response to the initial port 80 HTTP query.

The TCP/IP "Time To Live" Feature

Each IP packet has a Time to Live (TTL) section that keeps track of the number of network devices the packet has passed through to reach its destination. The server sending the packet sets the initial TTL value, and each network device that the packet passes through then reduces this value by 1. If the TTL value reaches 0, the network device will discard the packet.

This mechanism helps to ensure that bad routing on the Internet won't cause packets to aimlessly loop around the network without being removed. TTLs therefore help to reduce the clogging of data circuits with unnecessary traffic.

Remember this concept as it will be helpful in understanding the traceroute troubleshooting technique outlined in Chapter 4, "[Simple Network Troubleshooting](#)", that covers Network Troubleshooting.

The ICMP Protocol and Its Relationship to TCP/IP

There is another commonly used protocol called the Internet Control Message Protocol (ICMP). It is not strictly a TCP/IP protocol, but TCP/IP-based applications use it frequently.

ICMP provides a suite of error, control, and informational messages for use by the operating system. For example, IP packets will occasionally arrive at a server with corrupted data due to any number of reasons including a bad connection; electrical interference, or even misconfiguration. The server will usually detect this by examining the packet and correlating the contents to what it finds in the IP header's error control section. It will then issue an ICMP reject message to the original sending machine saying that the data should be re-sent because the original transmission was corrupted.

ICMP also includes echo and echo reply messages used by the Linux ping command to confirm network connectivity. ICMP TTL expired messages are also sent by network devices back to the originating server whenever the TTL in a packet is decremented to zero. More information on ICMP messages can be found in both Appendix 1, "[Miscellaneous Linux Topics](#)", and Chapter 4, "[Simple Network Troubleshooting](#)", on network troubleshooting.

How IP Addresses Are Used To Access Network Devices

All TCP/IP enabled devices connected to the Internet have an Internet Protocol (IP) address. Just like a telephone number, it helps to uniquely identify a user of the system. The Internet Assigned Numbers Authority (IANA) is the organization responsible for assigning IP addresses to Internet Service Providers (ISPs) and deciding which ones should be used for the public Internet and which ones should be used on private networks.

IP addresses are in reality a string of 32 binary digits or **bits**. For ease of use, network engineers often

divide these 32 bits into four sets of 8 bits (or octets), each representing a number from 0 to 255. Each number is then separated by a period (.) to create the familiar dotted **decimal notation**. An example of an IP address that follows these rules is 97.65.25.12.

Note: Chapter 3, "[Linux Networking](#)", which covers Linux specific networking topics, explains how to configure the IP address of your Linux box.

Private IP Addresses

Some groups of IP addresses are reserved for use only in private networks and are not routed over the Internet. These are called **private IP addresses** and have the following ranges:

10.0.0.0 - 10.255.255.255
172.16.0.0 - 172.31.255.255
192.168.0.0 - 192.168.255.255

Home networking equipment/devices usually are configured in the factory with an IP address in the range 192.168.1.1 to 192.168.1.255.

You may be wondering how devices using private addresses could ever access the Internet if the use of private addresses on the Internet is illegal. The situation gets even more confusing if you consider the fact that hundreds of thousands of office and home networks use these same addresses. This must cause networking confusion. Don't worry, this problem is overcome by NAT.

The localhost IP Address

Whether or not your computer has a network interface card it will have a built-in IP address with which network-aware applications can communicate with one another. This IP address is defined as 127.0.0.1 and is frequently referred to as localhost. This concept is important to understand, and will be revisited in many later chapters.

Network Address Translation (NAT) Makes Private IPs Public

Your router/firewall will frequently be configured to give the impression to other devices on the Internet that all the servers on your home/office network have a valid **public IP address**, and not a "private" IP address. This is called **network address translation (NAT)** and is often also called **IP masquerading** in the Linux world. There are many good reasons for this, the two most commonly stated are:

- No one on the Internet knows your true IP address. NAT protects your home PCs by assigning them IP addresses from "private" IP address space that cannot be routed over the Internet. This prevents hackers from directly attacking your home systems because packets sent to the "private" IP will never pass over the Internet.
- Hundreds of PCs and servers behind a NAT device can masquerade as a single public IP address. This greatly increases the number of devices that can access the Internet without running out of "public" IP addresses.

You can configure NAT to be **one to one** in which you request your ISP to assign you a number of public IP addresses to be used by the Internet-facing interface of your firewall and then you pair each

http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_-_Ch02_-_Introduction_to_Networking

of these addresses to a corresponding server on your protected private IP network. You can also use **many to one** NAT, in which the firewall maps a single IP address to multiple servers on the network.

As a general rule, you won't be able to access the public NAT IP addresses from servers on your home network. Basic NAT testing requires you to ask a friend to try to connect to your home network from the Internet.

Examples of NAT may be found in the IP masquerade section of Chapter 14, "[Linux Firewalls Using iptables](#)", that covers the Linux iptables firewall. Some of the terms mentioned here may be unfamiliar to you but they will be explained in later sections of this chapter.

Port Forwarding with NAT Facilitates Home-Based Web sites

In a simple home network, all servers accessing the Internet will appear to have the single public IP address of the router/firewall because of many to one NAT. Because the router/firewall is located at the border crossing to the Internet, it can easily keep track of all the various outbound connections to the Internet by monitoring:

- The IP addresses and TCP ports used by each home based server and mapping it to
- The TCP ports and IP addresses of the Internet servers with which they want to communicate.

This arrangement works well with a single NAT IP trying to initiate connections to many Internet addresses. The reverse isn't true.

New connections initiated from the Internet to the public IP address of the router/firewall face a problem. The router/firewall has no way of telling which of the many home PCs behind it should receive the relayed data because the mapping mentioned earlier doesn't exist beforehand. In this case the data is usually discarded.

Port forwarding is a method of counteracting this. For example, you can configure your router/firewall to forward TCP port 80 (Web/HTTP) traffic destined to the outside NAT IP to be automatically relayed to a specific server on the inside home network

As you may have guessed, port forwarding is one of the most common methods used to host Web sites at home with DHCP DSL.

DHCP

The Dynamic Host Configuration Protocol (DHCP) is a protocol that automates the assignment of IP addresses, subnet masks default routers, and other IP parameters.

The assignment usually occurs when the DHCP configured machine boots up, or regains connectivity to the network. The DHCP client sends out a query requesting a response from a DHCP server on the locally attached network. The DHCP server then replies to the client PC with its assigned IP address, subnet mask, DNS server and default gateway information.

The assignment of the IP address usually expires after a predetermined period of time, at which point the DHCP client and server renegotiate a new IP address from the server's predefined pool of addresses. Configuring firewall rules to accommodate access from machines who receive their IP addresses via DHCP is therefore more difficult because the remote IP address will vary from time to time. You'll probably have to allow access for the entire remote DHCP subnet for a particular TCP/UDP port.

Most home router/firewalls are configured in the factory to be DHCP servers for your home network. You can also make your Linux box into a DHCP server, once it has a fixed IP address.

The most commonly used form of DSL will also assign the outside interface of your router/firewall with a single DHCP provided IP address.

You can check Chapter 3, "[Linux Networking](#)", on Linux networking topics page on how to configure your Linux box to get its IP address via DHCP. You can also look at Chapter 8, "[Configuring the DHCP Server](#)", on Configuring a DHCP Server, to make your Linux box provide the DHCP addresses for the other machines on your network.

How DNS Links Your IP Address To Your Web Domain

The domain name system (DNS) is a worldwide server network used to help translate easy to remember domain names like www.linuxhomenetworking.com into an IP address that can be used behind the scenes by your computer. Here step by step description of what happens with a DNS lookup.

1. Most home computers will get the IP address of their DNS server via DHCP from their router/firewall.
2. Home router/firewall providing DHCP services often provides its own IP address as the DNS name server address for home computers.
3. The router/firewall then redirects the DNS queries from your computer to the DNS name server of your Internet service provider (ISP).
4. Your ISP's DNS server then probably redirects your query to one of the 13 "root" name servers.
5. The root server then redirects your query to one of the Internet's ".com" DNS name servers which will then redirect the query to the "[linuxhomenetworking.com](http://www.linuxhomenetworking.com)" domain's name server.
6. The [linuxhomenetworking.com](http://www.linuxhomenetworking.com) domain name server then responds with the IP address for www.linuxhomenetworking.com

As you can imagine, this process can cause a noticeable delay when you are browsing the Web. Each server in the chain will store the most frequent DNS name to IP address lookups in a memory cache which helps to speed up the response. Chapter 18, "[Configuring DNS](#)", explains how to you can make your Linux box into a caching or regular DNS server for your network or Web site if your ISP provides you with fixed IP addresses. Chapter 19, "[Dynamic DNS](#)", explains how to configure DNS for a Web site housed on a DHCP DSL circuit where the IP address constantly changes. It explains the auxiliary DNS standard called dynamic DNS (DDNS) that was created for this type of scenario.

IP Version 6 (IPv6)

Most Internet-capable networking devices use version 4 of the Internet Protocol (IPv4) which I have described here. You should also be aware that there is now a version 6 (IPv6) that has recently been developed as a replacement.

With only 32 bits, the allocation of version 4 addresses will soon be exhausted between all the world's ISPs. Version 6, which uses a much larger 128-bit address offers eighty billion, billion, billion times more IP addresses which it is hoped should last for most of the 21st century.

IPv6 packets are also labeled to provide quality-of-service information that can be used in prioritizing real-time applications, such as video and voice, over less time-sensitive ones such as regular Web surfing and chat. IPv6 also inherently supports the IPsec protocol suite used in many forms of secured

networks, such as virtual private networks (VPNs).

Most current operating systems support IPv6 even though it isn't currently being used extensively within corporate or home environments. Expect it to become an increasingly bigger part of your network planning in years to come.

How Subnet Masks Group IP Addresses into Networks

Subnet masks are used to tell which part of the IP address represents the network on which the computer is connected (network portion) and the computer's unique identifier on that network (host portion). The term **netmasks** is often used interchangeably with the term **subnet masks**, this book will use the latter term for the sake of consistency.

A simple analogy would be a phone number, such as (808) 225-2468. The (808) represents the area code, and the 225-2468 represents the telephone within that area code. Subnet masks allow you to specify how long you want the area code to be (network portion) at the expense of the number of telephones in that area code (host portion)

Most home networks use a subnet mask of 255.255.255.0. Each 255 means this octet is for the area code (network portion). So if your server has an IP address of 192.168.1.25 and a subnet mask of 255.255.255.0, the network portion would be 192.168.1 and the server or host would be device #25 on that network.

In all cases, the first IP address in a network is reserved as the network's base address and the last one is reserved for broadcast traffic that is intended to be received by all devices on the network. In our example, 192.168.1.0 would be the network address and 192.168.1.255 would be used for broadcasts. This means you can then use IP addresses from #1 to #254 on your private network.

Calculating The Number of Addresses Assigned to a Subnet

Most office and home networks use networks with 255 IP addresses or less in which the subnet mask starts with the numbers 255.255.255. This is not a pure networking text, so I'll not discuss larger networks because that can become complicated, but in cases where less than 255 IP addresses are required a few apply. There are only seven possible values for the last octet of a subnet mask. These are 0, 192, 128, 224, 240, 248 and 252. You can calculate the number of IP addresses for each of these by subtracting the value from 256.

In many cases the subnet mask isn't referred to by the dotted decimal notation, but rather by the actual number of bits in the mask. So for example a mask of 255.255.255.0 may be called a /24 (slash 24) mask instead. A list of the most commonly used masks in the office or home environment is presented in Table 2-2.

Table 2-2: The "Dotted Decimal" And "Slash" Subnet Mask Notations

Dotted Decimal Format	Slash Format	Available Addresses
-----------------------	--------------	---------------------

255.255.255.0	/24	256
255.255.255.128	/25	128
255.255.255.192	/26	64
255.255.255.224	/27	32
255.255.255.240	/28	16
255.255.255.248	/29	8
255.255.255.252	/30	4

So for example, if you have a subnet mask of 255.255.255.192, then you have 64 IP addresses in your subnet (256 - 192)

Calculating the Range of Addresses on Your Network

If someone gives you an IP address of 97.158.253.28 and a subnet mask of 255.255.255.248, how do you determine the network address and the broadcast address, in other words the boundaries, of your network? The following section outlines the steps to do this using both a manual and programmed methodology.

Manual Calculation

Take out your pencil and paper, manual calculation can be tricky. Here we go!

1. Subtract the last octet of the subnet mask from 256 to give the number of IP addresses in the subnet. $(256 - 248) = 8$
2. Divide the last octet of the IP address by the result of step 1; don't bother with the remainder (for example $28 / 8 = 3$). This gives you the theoretical number of subnets of the same size that are below this IP address.
3. Multiply this result by the result of step 1 to get the network address ($8 \times 3 = 24$). Think of it as the third subnet with 8 addresses in it. The network address is therefore 97.158.253.24
4. The broadcast address is the result of step 3 plus the result of step 1 minus 1. ($24 + 8 - 1 = 31$). Think of it as the broadcast address being the network address plus the number of IP addresses in the subnet minus 1". The broadcast address is 97.158.253.31

Let's do this for 192.168.3.56 with a mask of 255.255.255.224:

1. $256 - 224 = 32$
2. $56/32 = 1$
3. $32 \times 1 = 32$. Therefore the network base address is 192.168.3.32
4. $32 + 32 - 1 = 63$. Therefore the broadcast address is 192.168.3.63

Let's do this for 10.0.0.75 with a mask of 255.255.255.240

1. $256 - 240 = 16$
2. $75/16 = 4$
3. $16 \times 4 = 64$. Therefore the network base address is 10.0.0.64
4. $64 + 16 - 1 = 79$. Therefore the broadcast address is 10.0.0.79

Note: As a rule of thumb, the last octet of your network base address must be divisible by the "256 minus the last octet of your subnet mask" and leave no remainder. If you are sub-netting a large chunk of IP addresses it's always a good idea to lay it out on a spreadsheet to make sure there are no overlapping subnets. Once again, this calculation exercise only works with subnet masks that start with "255.255.255".

Calculation Using a Script

There is a BASH script in Appendix II, "[Codes, Scripts, and Configurations](#)", that will do this for you. Here is an example of how to use it, just provide the IP address followed by the subnet mask as arguments. It will accept subnet masks in dotted decimal format or /value format

```
[root@bigboy tmp]# ./subnet-calc.sh 216.151.193.92 /28
IP Address       : 216.151.193.92
Network Base Address : 216.151.193.80
Broadcast Address  : 216.151.193.95
Subnet Mask       : 255.255.255.240
Subnet Size       : 16 IP Addresses
[root@bigboy tmp]#
```

Subnet Masks for the Typical Business DSL Line

If you purchased a DSL service from your ISP that gives you fixed IP addresses, they will most likely provide you with a subnet mask of 255.255.255.248 that defines 8 IP addresses. For example, if the ISP provides you with a public network address of 97.158.253.24, a subnet mask of 255.255.255.248, and a gateway of 97.158.253.25, then your IP addresses will be:

```
97.158.253.24 - Network base address
97.158.253.25 - Gateway
97.158.253.26 - Available
97.158.253.27 - Available
97.158.253.28 - Available
97.158.253.29 - Available
97.158.253.30 - Available
97.158.253.31 - Broadcast
```

The Physical and Link Layers

TCP/IP can be quite interesting, but a knowledge of the first two layers of the OSI model are important too, because without them, even the most basic communication would be impossible.

There are very many standards that define the physical, electrical, and error-control methodologies of data communication. One of the most popular ones in departmental networks is Ethernet, which is available in a variety of cable types and speed capabilities, but the data transmission and error correction strategy is the same in all.

Ethernet used to operate primarily in a mode where every computer on a network section shared the same Ethernet cable. Computers would wait until the line was clear before transmitting. They would then send their data while comparing what they wanted to send with what they actually sent on the cable as a means of error detection. If a mathematical comparison, or cyclic redundancy check (CRC), detected any differences between the two, the server would assume that it transmitted data simultaneously with another server on the cable. It would then wait some random time and retransmit at some later stage when the line was clear again.

Transmitting data only after first sensing whether the cable, which was strung between multiple devices, had the correct signaling levels is a methodology called **carrier sense, multiple access** or CSMA. The ability to detect garbling due to simultaneous data transmissions, also known as collisions, is called **collision detect** or CD. You will frequently see references to Ethernet being a CSMA/CD technology for this reason and similar schemes are now being used in wireless networks.

Ethernet devices are now usually connected via a dedicated cable, using more powerful hardware capable of simultaneously transmitting and receiving without interference, thereby making it more reliable and inherently faster than its predecessor versions. The original Ethernet standard has a speed of 10 Mbps; the most recent versions can handle up to 40Gbps!

The 802.11 specifications that define many wireless networking technologies are another example of commonly used layer 1 and 2 components of the OSI model. DSL, cable modem standards and, T1 circuits are all parts of these layers.

The next few sections describe many physical and link layer concepts and the operation of the devices that use them to connect the computers in our offices and homes.

Networking Equipment Terminology

Up to this point you have had only an introduction to the theory of the first two OSI layers. Now we'll cover the hardware used to implement them.

Network Interface Cards

Your network interface card is also frequently called a NIC. Currently, the most common types of NIC used in the home and office are Ethernet and wireless Ethernet cards.

The Meaning of the NIC Link Light

The link light signifies that the NIC card has successfully detected a device on the other end of the cable. This indicates that you are using the correct type of cable and that the duplex has been negotiated correctly between the devices at both ends.

Duplex Explained

Full duplex data paths have the capability of allowing the simultaneous sending and receiving of data. Half duplex data paths can transmit in both directions too, but in only one direction at a time.

Full duplex uses separate pairs of wires for transmitting and receiving data so that incoming data flows don't interfere with outgoing data flows.

Half duplex uses the same pairs of wires for transmitting and receiving data. Devices that want to transmit information have to wait their turn until the "coast is clear" at which point they send the data. Error-detection and data-retransmission mechanisms ensure that the data reaches the destination correctly and are specifically designed to remedy data corruption caused when multiple devices start transmitting at the same time.

A good analogy for full duplex communications is the telephone, in which both parties can speak at the same time. Half duplex on the other hand is more like a walkie-talkie in which both parties have to wait until the other is finished before they can speak.

Data transfer speeds will be low and error levels will be high if you have a device at one end of a cable set to full duplex and a device at the other end of the cable set to half duplex.

Most modern network cards can autonegotiate duplex with the device on the other end of the wire. It is for this reason that duplex settings aren't usually a problem for Linux servers.

The MAC Address

The media access control (MAC) address can be equated to the serial number of the NIC. Every IP packet is sent out of your NIC wrapped inside an Ethernet frame that uses MAC addresses to direct traffic on your locally attached network.

MAC addresses therefore have significance only on the locally attached network. As the packet hops across the Internet, its source/destination IP address stays the same, but the MAC addresses are reassigned by each router on the way using a process called ARP.

How ARP Maps the MAC Address to Your IP Address

The Address Resolution Protocol (ARP) is used to map MAC addresses to network IP addresses. When a server needs to communicate with another server it does the following steps:

1. The server first checks its routing table to see which router provides the next hop to the destination network.
2. If there is a valid router, let's say with an IP address of 192.168.1.1, the server checks its ARP table to see whether it has the MAC address of the router's NIC. You could very loosely view

this as the server trying to find the Ethernet serial number of the next hop router on the local network, thereby ensuring that the packet is sent to the correct device.

3. If there is an ARP entry, the server sends the IP packet to its NIC and tells the NIC to encapsulate the packet in a frame destined for the MAC address of the router.
4. If there is no ARP entry, the server issues an ARP request asking that router 192.168.1.1 respond with its MAC address so that the delivery can be made. When a reply is received, the packet is sent and the ARP table is subsequently updated with the new MAC address.
5. As each router in the path receives the packet, it plucks the IP packet out of the Ethernet frame, leaving the MAC information behind. It then inspects the destination IP address in the packet and use its routing table to determine the IP address of the next router on the path to this destination.
6. The router then uses the "ARP-ing" process to get the MAC address of this next hop router. It then reencapsulates the packet in an Ethernet frame with the new MAC address and sends the frame to the next hop router. This relaying process continues until the packet reaches the target computer.
7. If the target server is on the same network as the source server, a similar process occurs. The ARP table is queried. If no entry is available, an ARP request is made asking the target server for its MAC address. Once a reply is received, the packet is sent and the ARP table is subsequently updated with the new MAC address.
8. The server will not send the data to its intended destination unless it has an entry in its ARP table for the next hop. If it doesn't, the application needing to communicate will issue a timeout or time exceeded error.
9. As can be expected, the ARP table contains only the MAC addresses of devices on the locally connected network. ARP entries are not permanent and will be erased after a fixed period of time depending on the operating system used.

Chapter 3, "[Linux Networking](#)", which covers Linux network topics, shows how to see your ARP table and the MAC addresses of your server's NICs.

Common ARP Problems When Changing A NIC

You may experience connectivity problems if you change the MAC address assigned to an IP address. This can happen if you swap a bad NIC card in a server, or replace a bad server but have the new one retain the IP address of the old.

Routers typically save learned MAC to IP address map entries in a cache and won't refresh them unless a predefined period of time has elapsed. Changing the NIC, while retaining the IP address can cause problems as the router will continue to send frames onto the network with the correct target IP address but the old target MAC address. The server with the new NIC won't respond as the frame's target MAC doesn't match it's own.

This problem can be fixed in one of two ways. You can delete all the ARP entries in the router's cache. The second solution is to log into the server's console and ping it's gateway. The router will detect the MAC to IP address change and it will readjust its ARP table.

The Two Broad Types Of Networking Equipment

There are two main types of networking equipment; Data Communications Equipment (DCE) which is

intended to act as the primary communications path, and Data Terminal Equipment (DTE) which acts as the source or destination of the transmitted data.

Data Terminal Equipment

DTE devices were originally computer terminals located at remote offices or departments that were directly connected modems. The terminals would have no computing power and only functioned as a screen/keyboard combination for data processing.

Nowadays most PCs have their COM and Ethernet ports configured as if they were going to be connected to a modem or other type of purely networking-oriented equipment.

Data Communications Equipment

A DCE is also known as Data Circuit-Terminating Equipment and refers to such equipment as modems and other devices designed primarily to provide network access.

Using Straight-Through/Crossover Cables to Connect DTEs And DCEs

When a DCE is connected to a DTE, you will need a **straight-through cable**. DCEs connected to DCEs or DTEs connected to DTEs require **crossover cables**. This terminology is generally used with Ethernet cables.

The terminology can be different for cables used to connect serial ports together. When connecting a PC's COM port (DTE) to a modem (DCE) the straight-through cable is frequently called a **modem cable**. When connecting two PCs (DTE) together via their COM ports, the crossover cable is often referred to as a **null modem cable**.

Some manufacturers configure the Ethernet ports of their networking equipment to be either of the DTE or the DCE type, and other manufacturers have designed their equipment to flip automatically between the two types until it gets a good link. As you can see, confusion can arise when selecting a cable. If you fail to get a link light when connecting your Ethernet devices together, try using the other type of cable.

A straight-through Ethernet cable is easy to identify. Hold the connectors side by side, pointing in the same direction with the clips facing away from you. The color of the wire in position #1 on connector #1 should be the same as that of position #1 on connector #2. The same would go for positions #2 through #8, that is, the same color for corresponding wires on each end. A crossover cable has them mixed up. Table 2-3 provides some good rules of thumb.

Table 2-3: Cabling Rules of Thumb

Scenario	Likely Cable Type
----------	-------------------

PC to PC	Crossover
Hub to hub	Crossover
Switch to switch	Crossover
PC to modem	Straight-Through
PC to hub	Straight-Through
PC to switch	Straight-Through

Connectivity Using Hubs

A hub is a device into which you can connect all devices on a network so that they can talk together. Hubs physically cross-connect all their ports with one another which causes all traffic sent from a server to the hub to be blurted out to all other servers connected to that hub whether they are the intended recipient or not.

Hubs have no, or very little, electronics inside and therefore do not regulate traffic. It is possible for multiple servers to speak at once with all of them receiving garbled messages. When this happens the servers try again, after a random time interval, until the message gets through correctly.

It is for these reasons that Ethernet devices that plug into hubs should be set to half duplex.

Note: Hubs can add a lot of delays to your network because of the message garbling collisions and retransmissions. A switch is a much more reliable and predictable alternative, and ones made for the home often cost only a few dollars more.

Using Switches as a Faster Alternative to Hubs

A switch is also a device into which you can connect all devices on a home network so that they can talk together. Unlike a hub, traffic sent from Server A to Server B will be received only by Server B. The only exception is broadcast traffic which is blurted out to all the servers simultaneously.

Switches regulate traffic, thereby eliminating the possibility of message garbling and providing a more efficient traffic flow.

Devices that plug into switches should be set to full duplex to take full advantage of the dedicated bandwidth coming from each switch port.

Local Area Networks

A local area network (LAN) is a grouping of ports on a hub, switch or tied to a wireless access point (WAP) that can communicate only with each other.

It is possible to connect multiple switches and/or hubs in a chain formation to create a LAN with more ports. This is often called **daisy chaining**.

Switches and hubs provide no access control between servers connected to the same LAN. This is why network administrators group trusted servers having similar roles on the same LAN.

Servers use their IP address and subnet mask and the IP address of the remote server to determine whether they are both on the same network. If not, they attempt to communicate with each other via routers that interconnect their LANs. Routers are also capable of filtering traffic passing between the two LANs therefore providing additional security.

Larger, more expensive switches can be configured to assign only certain ports to prespecified virtual LANs or (VLANs) chosen by the network administrator. In this case, the switch houses ports on multiple LANs. A router still needs to be connected to each VLAN for internetwork communication.

How Routers Interconnect LANs

As stated before, switches and hubs usually have only servers connected to them that have been configured as being part of the same network. By connecting its NIC cards to multiple LANs, a correctly configured router is capable of relaying traffic between networks.

Routers can also be configured to deny communication between specific servers on different networks. They can also filter traffic based on the TCP port section of each packet. For example, it is possible to deny communication between two servers on different networks that intend to communicate on TCP port 80, and allow all other traffic between them. Routers therefore direct and regulate traffic between separate networks, much like a traffic policeman.

If you intend to route between networks, you must reserve an IP address for a router for each network and make sure that the router is directly connected to the LAN associated with that network. The network engineer responsible for the router will also have to specify which locally connected networks can be advertised to the router's neighbors and whether this information can be relayed to all the routers in an administrative zone, or routing domain.

In this example, we can see that the router is aware of many different networks (represented by the slash notation for subnet masks). The routing table also shows the best serial and VLAN type interfaces to use to get to these destinations and the IP address of the neighboring router through which traffic needs to be relayed to get there.

```
router>show ip route
...
...
Gateway of last resort is 10.1.1.1.1 to network 0.0.0.0

    172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
S       172.16.0.0/16 is directly connected, Null0
S       172.16.6.0/24 is directly connected, Null0
    172.17.0.0/24 is subnetted, 2 subnets
S       172.17.2.0 [1/0] via 10.1.1.1.1
S       192.168.200.0/24 is directly connected, Null0
    10.0.0.0/8 is variably subnetted, 64 subnets, 8 masks
O E1    10.2.0.0/16 [110/22] via 10.89.0.2, 3w1d, Serial1/0/1
C       10.119.3.0/24 is directly connected, Vlan3
O       10.2.2.0/24 [110/3] via 10.89.0.2, 3w1d, Serial1/0/1
O       10.132.10.0/24 [110/3] via 10.119.2.2, 3w1d, Vlan2
```

```
O      10.119.0.20/30 [110/3] via 10.89.0.26, 7w0d, Serial1/0/0
S      10.253.72.0/21 [1/0] via 10.1.1.13
C      10.10.192.0/24 is directly connected, Vlan114
O      10.230.232.0/22 [110/4] via 10.89.0.26, 7w0d, Serial1/0/0
S*    0.0.0.0/0 [1/0] via 10.1.1.1.1
router>
```

In home networks, routers usually have only two interfaces that provide connectivity to the Internet via network address translation or NAT. In other words routers act as gateways to the wider world and it won't be surprising to learn that routers are frequently referred to as "gateways".

Note: The term **gateway** specifically refers to a device that routes traffic between dissimilar network protocols (IP to Appletalk) or access methods (Ethernet to DSL). Routers transfer traffic where both the protocols and communications medium are the same. The terms are frequently used interchangeably, especially if only one network protocol is being used. Therefore a home DSL router that provides IP Internet access to an Ethernet network is technically both a gateway and a router. The distinction can be important in complicated networking environments where newer technologies need to talk with older ones using incompatible communications protocols.

How Simple Routing Works

In the broader networking sense, a "route" refers to the path data takes to traverse from its source to its destination. Each router along the way may also be referred to as a hop.

Usually when we speak about a route on a Linux box, we are referring to the IP address of the first hop needed to reach the desired destination network. It is assumed that this first hop will know how to automatically relay the packet.

As explained previously, routers are designed to exchange routing information dynamically, and can therefore intelligently redirect traffic to bypass failed network links. Home Linux boxes frequently don't run a dynamic routing protocol and therefore rely on "static" routes issued by the system administrator at the command line or in configuration files to determine the next hop to all desired networks.

Chapter 3, "[Linux Networking](#)", which covers Linux network topics, shows how to add static routes to your Linux box and also how you can convert it into a simple router.

Default Gateways, The Routers Of Last Resort

A default gateway is the router that is used when no alternative devices can be found to relay the traffic. They are often called "routers of last resort".

Say for example you have two routers R1 and R2. R1 is connected to both your SOHO home network and the Internet. R2 is connected to SOHO home network and is capable of relaying data to other corporate networks with addresses starting with 10.X.X.X via another NIC card.

You could put a route on your SOHO servers that states:

- Go to network 10.0.0.0 255.0.0.0 via router R2
- Go to everything else via router R1. R1 therefore would be considered your default gateway

For most home networks, your default gateway would be the router/firewall connected to the Internet. Chapter 3, "[Linux Networking](#)", which covers Linux network topics, shows how to configure the default gateway on your Linux box.

Firewalls Help Provide a Secure Routing Environment

Firewalls can be viewed as routers with more enhanced abilities to restrict traffic, not just by port and IP address as routers do. Specifically, firewalls can detect malicious attempts to subvert the TCP/IP protocol. A short list of capabilities includes:

- Throttling traffic to a server when too many unfulfilled connections are made to it
- Restricting traffic being sent to obviously bogus IP addresses
- Providing network address translation or NAT

Routers are designed to make packets flow as quickly as possible with the minimum amount of inspection. Firewalls are used as close to the source or target of data communication as possible to try to ensure that the data hasn't been subverted.

Firewalls can often create an encrypted data path between two Private networks across the Internet providing secure communication with a greatly reduced chance of eavesdropping. These communication channels are called Virtual Private Networks or VPNs and are frequently used to connect branch offices to the corporate headquarters and also to allow sales representatives to get access to sensitive pricing information when traveling from town to town.

Additional Introductory Topics

The last few topics of this chapter may not appear to be directly related to networking, but they cover Linux help methods that you'll use extensively and the File Transfer Protocol (FTP) package, which enables you to download all the software you need to get your Linux server operational as quickly as possible.

The File Transfer Protocol

FTP is one of the most popular applications used to copy files between computers via a network connection. Knowledge of FTP is especially important and is a primary method of downloading software for Linux systems.

There are a number of commercially available GUI based clients you can load on your PC to do this, such as WSFTP and CuteFTP. You can also use FTP from the command line as shown in Chapter 6, "[Installing RPM Software](#)", on RPM software installation.

From the remote user's perspective, there are two types of FTP. The first is **regular FTP** which is used primarily to allow specific users to download files to their systems. The remote FTP server prompts you for a specific username and password to gain access to the data.

The second method, **anonymous FTP** is used primarily to allow any remote user to download files to their systems. The remote FTP server prompts you for a username, at which point the user types

anonymous or ftp with the password being any valid e-mail address.

From the systems administrator's perspective, there are another two categories. These are "active" and "passive" FTP which is covered in more detail in Chapter 15, "[Linux FTP Server Setup](#)".

It is good to remember that FTP isn't very secure as usernames, passwords and data are sent across the network unencrypted. More secure forms such as SFTP (Secure FTP) and SCP (Secure Copy) are available as a part of the Secure Shell package (covered in Chapter 17, "[Secure Remote Logins and File Copying](#)") that is normally installed by default with Fedora.

Linux Help

Linux help files are accessed using the man or manual pages. From the command line you issue the man command followed by the Linux command or file about which you want to get information. If you want to get information on the ssh command, then you'd use the command man ssh.

```
[root@bigboy tmp]# man ssh
```

If you want to search all the man pages for a keyword, then use the man command with the -k switch, for example, man -k ssh which will give a list of all the man pages that contain the word ssh.

```
[root@bigboy tmp]# man -k ssh
```

```
...
...
ssh (1) - OpenSSH SSH client (remote login program)
ssh [slogin] (1) - OpenSSH SSH client (remote login program)
ssh-agent (1) - authentication agent
ssh-keyscan (1) - gather ssh public keys
ssh_config (5) - OpenSSH SSH client configuration files
sshd (8) - OpenSSH SSH daemon
sshd_config (5) - OpenSSH SSH daemon configuration file
...
...
[root@bigboy tmp]#
```

This book is targeted at proficient Linux beginners and above so I'll be using a wide variety of commands in this book without detailed explanations to help keep the flow brisk. If you need more help on a command, use its man page to get more details on what it does and the syntax it needs. Linux help can sometimes be cryptic, but with a little practice the man pages can become your friend.