

# Chapter 5

## Network Applications

This chapter introduces the user to the basic utilities that are available to utilize the Internet. This will allow one to test and verify the network, as well as establish fundamental communications across the Internet for file transfer and text based email.

### Concepts Learned in this Chapter

- Changing the IP Address
- Network Address Configuration
- Address Resolution
- Hosts Table
- Remote Access
- Network Connectivity
- Network Users
- Network Messaging
- DNS Name Resolution
- User Information
- User Mail

## Table of Contents

|   |    |
|---|----|
| Network Applications.....                             | 1  |
| 5.1 Changing the IP Address.....                      | 4  |
| ifconfig interface address .....                      | 4  |
| 5.2 Address Configuration.....                        | 5  |
| 5.2.1 Static IP Address.....                          | 5  |
| 5.2.3 Configuring System.....                         | 6  |
| 5.2.3.1 Manual Editing IP Address.....                | 7  |
| 5.2.3.2 linuxconf.....                                | 7  |
| 5.2.3.3 X Windows Network Configuration Utility ..... | 7  |
| 5.2.3.4 netconfig.....                                | 7  |
| 5.4 IP Address Table .....                            | 9  |
| 5.4.1 Local Hosts Table .....                         | 9  |
| 5.4.2 Remote Name Lookup.....                         | 10 |
| 5.5 Remote Access .....                               | 11 |
| 5.5.1 Telnet .....                                    | 11 |
| 5.5.2 File Transfer Protocol.....                     | 12 |
| 5.5.3 Secure Shell.....                               | 14 |
| 5.5.4 Secure Copy.....                                | 14 |
| 5.5.5 Secure File Transfer Protocol.....              | 15 |
| 5.5.6 Remote Login and Remote Copy.....               | 15 |
| 5.5.7 Remote Modem Access.....                        | 15 |
| 5.5.7.1 Minicom.....                                  | 15 |
| 5.5.7.2 Gcomm.....                                    | 16 |
| 5.6 Network Testing .....                             | 18 |
| 5.6.1 Ping .....                                      | 18 |
| CTRL-C .....  | 19 |
| 5.6.1.1 A Little Theory on What Happens.....          | 19 |
| 5.6.2 Tracing Network Path.....                       | 21 |
| 5.6.2.1 ping -R ipaddress .....                       | 21 |
| 5.6.2.2 traceroute.....                               | 21 |
| traceroute [options] someplace.com .....              | 21 |
| 5.6.2.3 Tracepath.....                                | 23 |
| 5.6.2.4 mtr .....                                     | 24 |
| 5.6.3 Network Status.....                             | 24 |
| netstat [options] [sockets].....                      | 24 |
| 5.6.4 Address Resolution Protocol – arp .....         | 26 |
| 5.7 Host Users .....                                  | 27 |
| 5.7.1 who .....                                       | 27 |
| 5.7.2 users .....                                     | 27 |
| 5.7.3 w .....   | 28 |
| 5.7.4 last .....                                      | 28 |
| 5.7.5 lastlog .....                                   | 28 |
| 5.7.6 finger .....                                    | 29 |
| 5.7.7 whois .....                                     | 30 |
| 5.7.8 who am i .....                                  | 31 |
| 5.7.9 whoami .....                                    | 31 |

|  |    |
|--|----|
| 5.8 Network Messaging (Not Complete) .....           | 31 |
| 5.8.1 mesg .....                                     | 31 |
| 5.8.2 talk .....                                     | 31 |
| 5.8.3 wall .....                                     | 32 |
| 5.8.4 write .....                                    | 32 |
| 5.9 DNS Host Name Lookup .....                       | 32 |
| 5.9.1 nslookup .....                                 | 33 |
| 5.9.2 host .....                                     | 34 |
| 5.9.3 dig .....                                      | 34 |
| 5.10 Remote User Information .....                   | 35 |
| 5.11 User Mail.....                                  | 36 |
| 5.11.1 mail .....                                    | 36 |
| 5.11.1.1 Receiving Mail .....                        | 36 |
| 5.11.1.2 Sending Mail .....                          | 36 |
| 5.11.1.3 General Commands .....                      | 36 |
| 5.11.2 elm .....                                     | 37 |
| 5.11.3 pine .....                                    | 37 |
| 5.11.3.1 Receiving Mail using Pine.....              | 38 |
| 5.11.3.2 Sending Mail using Pine.....                | 38 |
| 5.11.3.3 Other Features.....                         | 38 |
| 5.12 Command Line Browser .....                      | 38 |
| lynx URL.....  | 38 |
| lynx yahoo.com.....                                  | 38 |
| 5.13 Downloading a Web Page Source File.....         | 39 |
| 5.14 Transferring Data to and from a Web Server..... | 40 |
| 5.15 Commands Used in this Chapter.....              | 40 |
| 5.16 Chapter Review Questions.....                   | 42 |

## 5.1 Changing the IP Address

As a Network Administrator, it is sometimes necessary to modify your IP address in order to test various systems. The process is very simple and straight forward.

The basic command to determine the IP Address on a system is found by issuing the command:

```
# ifconfig
eth0    Link encap:Ethernet HWaddr 00:60:08:3F:05:B1
        inet addr:192.168.1.10 Bcast:192.168.1.255
        Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:159347 errors:190 dropped:0 overruns:0 frame:380
        TX packets:89374 errors:0 dropped:0 overruns:0 carrier:0
        collisions:1108
        RX bytes:33190750 (31.6 Mb) TX bytes:8195375 (7.8 Mb)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        UP LOOPBACK RUNNING MTU:16436 Metric:1
        RX packets:2699 errors:0 dropped:0 overruns:0 frame:0
        TX packets:2699 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0
        RX bytes:2752123 (2.6 Mb) TX bytes:2752123 (2.6 Mb)
```

The command to establish an IP Address is the same one used to modify it. The command is:

```
ifconfig interface address
```

Where the interface will typically be **eth0**. This will either establish the address or modify it from what was previously. Typically, one does not have to specify the Subnet Mask if the address falls within the normal Class A, B, or C classes. If desired, the Subnet mask may be also specified as:

```
ifconfig interface address subnet-mask
```

This would be particularly useful if specifying a classless address.

After we have set the address, we need to terminate the service and then restart it. This is accomplished by issuing the command:

```
ifconfig interface down           and
ifconfig interface up
```

This provides a temporary change. To make the change permanent, issue the command:

```
xinetd
```

to reread the system configuration. This command does not write the new address to the correct file, this must be done manually. For example, to cycle the first Ethernet interface, one would issue the commands:

```
ifconfig eth0 down
ifconfig eth0 up
xinetd
```

As an alternative to using the ifconfig interface down / up, you may use the commands:

```
ifdown eth0
ifup eth0
```

Using this set of commands is generally preferred because the service is also restarted. Modification of the address is only effective to the present operation; the address set in the configuration files will override the modification the next time the system is booted.

## **5.2 Address Configuration**

When configuring Red Hat or Fedora Core Linux for Ethernet and the Internet, we use the configuration routines to automatically install the values for the IP Address and its appropriate settings. This procedure defines a static address to the interface.

### **5.2.1 Static IP Address**

A system may be set up with a fixed IP address, such as illustrated above. This is referred to as a static address.

When Linux boots, it calls the file **/etc/inittab**, which specifies that it should operate at the appropriate Run Level, such as 3 for the Command Line Mode or 5 for the X Windows Mode. **Inittab** then calls the file or script **/etc/rc.d/init.d/network** to activate the network interfaces. The **network** file checks another file - **/etc/sysconfig/network** to determine if the network is activated. This file contains the line:

```
NETWORKING=yes
HOSTNAME=L49.ourlab.com
GATEWAY=192.168.102.1
```

If the network is active, and **=no** if not.

Assuming that the network is active, the **/etc/rc.d/init.d/network** then executes a series of script files to activate the network interfaces. One file is **/etc/sysconfig/network-scripts/ifcfg-eth0**, which contains:

```
DEVICE=eth0
IPADDR=192.168.102.149           or your correct IP Address
NETMASK=255.255.255.0
NETWORK=192.168.102.0
BROADCAST=192.168.102.255
ONBOOT=yes
```

```
BOOTPROTO=none  
GATEWAY=192.168.102.1  
TYPE=Ethernet  
USERCTL=no  
PEERDNS=no
```

The ONBOOT=yes specifies that the interface is to be activated at bootup.

After the above information has been installed, another file, **/sbin/ifconfig** is run to provide additional configuration of the Internet card. Finally the file **/sbin/route** is run to set up the routing tables for the system.

To summarize, the configuration of the Ethernet interface follows the following process:

1.     inittab
2.     /etc/rc.d/init.d/network
- 3a.         /etc/sysconfig/network
- 3b.         /etc/sysconfig/network-scripts/ifup
- 4a.         /etc/sysconfig/network-scripts/ifup-eth0
- 4b.         /sbin/ifconfig
- 4c.         /sbin/route

### 5.2.2         Dynamic Address

Alternatively, a system may be set up to obtain an address from a server, called a Dynamic Host Configuration Protocol (DHCP) Server. In this situation, the system may have a different IP address every time it boots (although it generally will not in a business environment).

If we now look at the **/etc/sysconfig/network-scripts/ifcfg-eth0** file, we will observe something like the following setup:

```
DEVICE=eth0  
BOOTPROTO=dhcp  
NETMASK=255.255.255.0  
NETWORK=192.168.102.0  
BROADCAST=192.168.102.255  
ONBOOT=yes  
BOOTPROTO=none  
GATEWAY=192.168.102.1  
USERCTL=no  
PEERDNS=no
```

Note that even though we specify a network address, netmask, and gateway, these are overwritten by the dhcp values when the system is set up for dynamic IP Addressing.

### 5.2.3         Configuring System

There are a number of ways that you may configure the IP address of a system, from direct editing to using one of several configuration programs.

### 5.2.3.1 Manual Editing IP Address

To manually configure the IP Address of a system, you may edit the `/etc/sysconfig/network-scripts/ifcfg-eth0` file. After performing this, you must deactivate the service and restart it in order for it to take effect (`ifdown / ifup eth0`). At this time, this process is not encouraged, as there are better methods available.

### 5.2.3.2 `linuxconf`

The utility `linuxconf` is a very versatile program that allows the administrator to configure many different attributes of the system. Issuing the command:

**`linuxconf`**

from the CLI starts the program in a low-level graphics mode. If working in the X Windows mode, the command will start a “fancier” graphics screen – but it does exactly the same thing.

In this case, we need to select the Config: Networking: Host name and IP network devices. Clicking on this will open a screen to set the system name, domain name, and its IP address plus other necessary information. If your system has multiple NICs, they all may be configured from this screen by scrolling down.

Unfortunately, `linuxconf` has been depreciated in later versions of Red Hat. Great tool to do many things. (It should be available off of the Internet.)

### 5.2.3.3 X Windows Network Configuration Utility

From within X Windows, you may access the **System: Network Configuration** utility (icon displaying a small network), the Red Hat GUI configuration tool. This opens a new window with various options for configuring network interfaces. Selecting the HOSTS tab will display the IP address of the card for static configuration. From this screen you may edit the existing address, and apply the changes.

Using the Devices tab displays a card type window. Editing the existing Ethernet device allows you to specify the card name, protocol and the hardware device type. Editing the Protocol tab will show the protocol set to TCP/IP opens another window where you can specify the IP address, subnet mask and gateway or dhcp configuration, the hostname, and routing if appropriate.

Sometimes (and I do not know why), you may also observe a second Network Configuration with a “Tux” icon. This utility opens up a slightly different window that was formally available as the “`netcfg`” utility – which is no longer included with version Red Hat 7.2 . This is unfortunate, as it is a superior card configurator. (Since it is not available on the present system that is being tested against, I am not able to describe its usage.)

### 5.2.3.4 `netconfig`

This was one of the original text based GUI type network card configuration tools. Not fancy, but it does a good job of setting the configuration of an interface.

### 5.3 Address Resolution

After a system has been configured for a network address, it must resolve a local addresses from a network addresses. Depending upon the operating system and network protocol, this address resolution may be accomplished in different ways.

Within a local network, all systems communicate between themselves using the NIC card's MAC layer address that is burned into the NIC ROM. If you specify an IP address for the remote system, some means must be established to translate the IP address to the MAC address. Alternatively, if you are using MS Windows using a NetBIOS name, then it too must be converted to the MAC address.

For IP Address to be converted into the MAC address, we must establish some sort of translation process protocol. This will be explained best with an illustrative example.

Assume            System 1 IP = 192.168.1.11            MAC = 1.2.3.4.5.6  
                          System 2 IP = 192.168.1.12            MAC = a.b.c.d.e.f  
 then

System 1 wants to talk to System 2.

1. System 1 broadcasts a message saying 'Who has IP 192.168.1.12'?
2. System 2 answers back saying 'I am 192.168.1.12  
And my MAC address is A.B.C.D.E.F'
3. System 1 stores this in its ARP table and then sends a message to
4. System 2 using its MAC address with desired information

We can retrieve the MAC layer information by issuing the command:

**arp -a**

The response will appear something like the following:

From a MS Windows system:

**Interface: 192.168.102.149 on Interface 0x2000003**

**Internet Address    Physical Address    Type**

**192.168.102.152    00-a0-0c-c8-49-99    dynamic**

or from a Unix / Linux system:

**? (192.168.102.152) at 00:a0:0c:c8:49:99 (ether) on eth0**

In order to transmit information to another network, we must transition through either a router or a gateway. In this case, lets assume:

System 3 IP = 205.205.205.205    MAC = 1.a.2.b.3.c    LAN 2

Gateway IP = 192.168.102.200    MAC = a.1.b.2.c.3    LAN 1

Gateway IP = 205.205.205.201    MAC = b.9.c.8.d.4    LAN 3

Then:

1. System 1 realizes that 205.205.205.205 is not on its local network (uses subnet mask to realize domain addresses are not the same). System 1 knows that the way out of the local network is on address. 192.168.102.200 (Gateway address stored on System 1).



2. System 1 broadcasts a message saying 'Who has IP 192.168.102.200'?
3. Gateway responds with 'I am 192.168.102.200, my MAC is a.1.b.2.c.3'.
4. System 1 sends to Gateway information with to IP of 205.205.205.205 enveloped with a to address of a.1.b.2.c.3 .
5. Gateway strips off the MAC address of a.1.b.2.c.3 and forwards the information to the appropriate location where System 3 is located.
6. Gateway broadcasts message on LAN 2 saying, 'Who has IP 205.205.205.205?'
7. System 3 responds with 'I am 205.205.205.205, my MAC is 1.a.2.b.3.c'.
8. Gateway envelops the packet with the destination IP of 205.205.205.205 and MAC address of 1.a.2.b.3.c .

The transmission of data across a network is quite complex, and requires a network analyzer to monitor it. Linux includes a utility to analyze the data on the network called **tcpdump**. This works from the CLI and displays the packets in a raw form. The user must know how to convert the code, character by character, to interpret each packet of information.

Another X Windows utility included with Red Hat is Ethereal. This utility is not only able to display the data, but is also able to translate the data into a user-friendly format. It is also easy to set up filters to observe only that information that one is interested in. (Documentation must be obtained off of the Internet – [www.ethereal.com](http://www.ethereal.com) .)

## **5.4 IP Address Table**

When we wish to communicate with another host on an IP network, we need to indicate the IP address of the addressee. This is a cumbersome process because the address is a set of four decimal numbers. We have observed that the Internet functions off of the IP address as a mechanism to deliver information between different systems. The present system uses 32 bits, which is converted into four decimal numbers; the new system that is slowly being implemented is version 6 – which has 128 bits that will be represented as 8 sets of 4 character hex digits.

It quickly becomes evident that trying to remember all of the addresses as a numeric value is quite difficult. To solve this someone in their great wisdom invented the infamous Universal Resource Locator – URL. This allows us humans the ability to issue a name to a site, and have it translated into a numeric address by the computer. This is the concept of the IP Address Table.

The name to address translation is available in two levels, local and remote.

### **5.4.1 Local Hosts Table**

As humans, it is generally easier to remember a name rather than a number. It would be much easier to address a packet of information to a name rather than a set of numbers. The solution is to set up a table that provides this

translation between a name and an IP address. Such a table exists, called the **/etc/hosts** file, and is found in **/etc** directory. This file also exists on your Windows system (in the Windows directory / folder). This table is a list of IP addresses and the specified name. When we specify a system name, the system first looks in its local hosts file to determine if there is an IP address associated with it, if found, the information is routed directly to the recipient.

For Unix and Linux, the **hosts** file is located in the **/etc** directory. The content of the file is:

| IP Address      | System Name | Alias Name |
|-----------------|-------------|------------|
| For example:    |             |            |
| 192.168.102.149 | prof        | dennis     |

In this example, the address for the station that we want to call prof is 192.168.102.149. Alternatively we could also refer to the system as “dennis”. The name does not have to be the system name, just one that we want to give it. For instance, the hostname of a system might be **jdoe415**, and has a static address of 192.168.10.155, but we know that the user of that system is named Joe Doe, so we could make an entry in our hosts table of:

|                |     |         |    |
|----------------|-----|---------|----|
| 192.168.10.155 | joe | jdoe415 | jd |
|----------------|-----|---------|----|

Any of the three names would point to the specified IP address.

What we must make sure of is that a given name does not point to two different IP addresses. Confusion is not permitted.

When you first open the hosts file, you should find the localhost address of:

|                  |                  |
|------------------|------------------|
| <b>127.0.0.1</b> | <b>localhost</b> |
|------------------|------------------|

This is an internal IP address for testing whether the OSI 7 Layer Model software interface is operational, and is referred to as the local loopback. Your additions should follow the same format.

This list can be as long as one desires, but will be applicable only to that specific system.

In addition to the basic hosts file, there are two additional files called **hosts.allow** and **hosts.deny**. These two are set up to provide special permissions or to disallow access to specific hosts. At this time, both should be empty. They are available for other server functions such as FTP and telnet.

#### 5.4.2 Remote Name Lookup

When the URL name is not located in the local hosts table, then the system looks to an outside agent to supply the name to IP address resolution. The agent is called the Domain Name System, or DNS. How the DNS system works will be covered later when we learn how to configure a DNS server. For now, when we place a name request to the DNS server, we get back the appropriate IP Address.

For now, we need to insure that the system knows where to look to obtain a remote address. If necessary, we can edit the **/etc/resolv.conf** file to specify the IP address of the DNS server that we look to for obtaining a URL to IP address conversion. The file's contents appears something like the following:

```
search ourdomain.com
```

***nameserver 123.234.135.246***  
***domain ourdomain.ext***

Thus we will initially search our local domain, then go to the remote DNS server (123.234.135.246) to translate the URL. You may have more than one nameserver IP address. They are in order of search, that is, if the first is not available, then the next will be utilized.

## **5.5 Remote Access**

There are two functions that allow one to access another Unix / Linux system that need to be understood. These are TELNET and FTP.

### **5.5.1 Telnet**

Telnet provides for the ability to log onto a remote system and run CLI commands from a command line. In general, one is not able to telnet to a MS Windows client system, but should be able to telnet into an enabled server system.

Before a system may be accessed, it must be provisioned for the Telnet Internet Service. This will be covered in a later lab. For now we will assume that the system that one is to telnet to has been optioned to support access.

To telnet to a remote system, the process is a very easy and direct process. We issue the command:

***telnet IP-Address***

If access is allowed the system will then request a username and password. The response is:

***Red Hat Linux release 7.2 (Enigma)***  
***Kernel 2.4.7-10 on an i686***  
***Login:***

Naturally, the operating system (Red Hat), release (7.2), Kernel (2.4.7-10), and system (i686) may vary by what the remote system is or has been installed with.

You are now required to enter a valid username that has been previously created. You are then prompted for the username's password, which must be entered before you may proceed. After entering this, you will be given a login message, such as:

***Last login: day date time on port (tty2)***  
***[username@hostname username]\$***

You are now at the CLI prompt and may enter any command as if you are actually logged onto the system directly. Note that you are generally not allowed to log on as the administrator, root. This is for security purposes.

Although covered in more detail later regarding security, if you should need to perform a command as the administrator, you would issue the command:

***su***

You will be prompted for the root password, which must be entered. After entering the correct password, you prompt will be:

***[root@hostname location]#***

You may now make any modifications as the system administrator.

To return to your normal username, enter the command:

**exit**

You are now back to the normal prompt. To disconnect from the remote system, enter the command:

**exit**

You will now receive the message that you have been disconnected and you are now back at your system.

You may telnet to a Unix / Linux system from a MS Windows system by opening a command window, and then proceeding with the above procedure.

### 5.5.2 File Transfer Protocol

The FTP Server acts similar to Telnet, but is specifically designed for the transfer of files between two systems. One may upload a file to a FTP Server (if allowed), or download desired files.

To log onto a remote system, issue the command:

**ftp IP-Address**

You will get the following:

***Connected to IP-Address***

***220 System-name.domain-name FTP Server (Version# wu-2.6.1-18) ready***

***user (system-name): username***

***331 Password required for username***

***Password: {enter correct password}***

***230 user username logged in.***

***ftp >***

Note in the above response, you must enter in a valid username and password. Again, the username must be a previously entered user name on the remote system. You may now enter a variety of CLI commands to navigate around the remote system, and even navigate on your own system. After you have entered a valid username, you will be in your own home directory. From this point, you may change to any other directory that you have permission to enter, that is, you can not enter a directory owned by another user (their home directory) or the administrator's home directory (/root). If you should enter in the "**anonymous**" or "**ftp**" username, then you will be located in the general ftp directory, **/var/ftp**, where you may only download files. In order to login as an anonymous user, you must give your password as your email address, well this is almost true, a fake address will also work, as the system will only check the address format. Maybe in the future for security reasons, this will be improved to verify one's real email address.

Common ftp commands include:

|                            |  |
|----------------------------|--|
| <b>ascii</b>               | Transmit a file in ASCII (7 bit), generally not recommended  |
| <b>binary</b>              | Transmit a file using binary (8 bit), recommended format     |
| <b>bye</b>                 | Terminate ftp session  |
| <b>cd</b>                  | Change directory on remote system                            |
| <b>cdup</b>                | Change to the parent directory of the remote system          |
| <b>chmod mode filename</b> | Change the attributes of a file on remote system             |
| <b>close</b>               | Synonym for bye  |
| <b>delete remote-file</b>  | Delete a file on the remote system                           |
| <b>dir</b>                 | List directory contents of remote system (ls)                |
| <b>disconnect</b>          | Synonym for bye  |
| <b>get remote-file</b>     | Retrieve (download) file from remote system                  |
| <b>help command</b>        | Obtain information for a command from the remote system      |
| <b>lcd</b>                 | Change the working directory on the local system             |
| <b>ls</b>                  | List directory contents of remote system                     |
| <b>put local-file</b>      | Copy (upload) a local file to the remote system              |
| <b>pwd</b>                 | Print the path of the working directory on the remote system |
| <b>quit</b>                | Synonym for bye  |
| <b>size file-name</b>      | Returns size of the specified file on the remote system      |
| <b>user username</b>       | Log onto a remote system if initial logon failed             |
| <b>? command</b>           | Synonym for help   |

Many other commands are available, the above is a list of those that are most likely to be used.

If you log on as a normal user, you may traverse virtually anyplace on the remote system. This can be a security hazard. A user logged on may be able to upload a file that contains a virus, or worse, could modify or delete critical files on your system.

As noted previously, to improve security, but with some caution, two special types of users may be established. A GUEST user is typically restricted to a limited set of directories, and is allowed to upload (put) files. An ANONYMOUS user is only allowed to download (get) files from your system. Since an anonymous user does not log in under their normal username, they must supply their email address. Herein lies another problem, FTP does not verify if it is a valid address, only that it is of the correct format, hence a user could log in as an anonymous user and give an address that is totally bogus!

Again like telnet, you may not log onto a remote system as the administrator.

To exit from an ftp session, you need to issue the command:

**ftp > bye**

### 5.5.3 Secure Shell

As observed when using a telnet session, it is an insecure connection. To provide security to the connection, one may log onto a remote system using a secure connection, called **ssh**, or **Secure Shell**. All transmission across the network (Internet) is encrypted. By default, the ssh server daemon is typically active after installation.

In order to establish a secured transmission, the data must be encrypted. There are two methods that this may be accomplished by, by using a default key, or using a user generated key. At this time, only the default key is discussed, the creation of a user defined key is discussed in a later chapter.

When logging in for the first time, one will observe text similar to the following:

```
[root@fried root]# ssh 192.168.1.13
The authenticity of host '192.168.1.13 (192.168.1.13)' can't be
established.
RSA key fingerprint is
ae:9b:e3:5a:6a:d0:10:2c:99:0d:90:9e:c4:a3:0d:82.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.13' (RSA) to the list of
known hosts.
root@192.168.1.13's password:
Last login: Sat Sep 17 14:05:28 2005
[root@chinese root]#
```

By default, you will be logged in as the administrator, root. To log in as another user, use the command:

```
$ ssh -l username
```

After supplying the password (if needed), you will be in the logged in user's home directory. Note that by default, you are able to log in as the administrator, whereas telnet did not allow this. Enhanced security is provided in a later chapter.

Additional options are available.

### 5.5.4 Secure Copy

An alternative to transferring files between Unix / Linux hosts on a local network is **scp**, or **Secure Copy**. This file transfer format, like ssh, is encrypted. Again, a default key will be used, or the user generated ssh key may be used (as discussed in a later chapter).

The format of the command is:

```
scp username@host_IP : filename username@host2_IP :
filename
```

To copy a file from your host to a remote system, you would issue the command:

```
scp /path/filename username@host2_IP : filename
```

Additional options are available.

### 5.5.5 Secure File Transfer Protocol

The last secure application is **sftp**, or **Secure FTP**. Again, data transfer between two hosts is accomplished using an encrypted format. It operates identical to the ftp protocol in terms of user interaction. sftp also uses the ssh encryption key for encryption. The default key will be used unless the ssh generated key is available.

The format of the command is:

**sftp username@host\_IP**

Additional options are available.

### 5.5.6 Remote Login and Remote Copy

There are several additional commands available – **rlogin** and **rcp**. These are early commands to remotely login and remotely copy a file. Today, these commands are not secure and should not be used. Although they exist, do not use them.

### 5.5.7 Remote Modem Access

One of the oldest means to communicate between systems was via a modem. Two applications are available, first is the default Command Line application **minicom**, and the second is a much newer application that operates in the X-Windows, called **Gcomm**.

#### 5.5.7.1 Minicom

Minicom is the foundation of modem communications. Many other programs have obtained their foundation of operations from it.

The very first time that minicom is run, it requires the setup option, which provides the initial configuration. This is applied by running the command:

**minicom -s**

This brings up the initial configuration screen.

```
# minicom -s
```

```
minicom: WARNING: configuration file not found, using
defaults
```

```

[configuration]
Filenames and paths
File transfer protocols
Serial port setup
Modem and dialing
Screen and keyboard
Save setup as dfl
Save setup as..
Exit
Exit from Minicom

```

The most important of these settings is **Serial Port Setup**.

```
A - Serial Device      : /dev/ttyS1
B - Lockfile Location  : /var/lock
C - Callin Program     :
D - Callout Program    :
E - Bps/Par/Bits       : 38400 8N1
F - Hardware Flow Control : Yes
G - Software Flow Control : No

Change which setting?
```

From this point, one must specify which Serial Device (A), Bit Rate (E), and Flow Control (F and G).

Using either an internal or external modem, we must know whether we are using Comm 1 (ttyS0), Comm 2 (ttyS1), Comm 3 (ttyS2), or Comm 4 (ttyS3). From the menu, click the key “A” and edit the value for the proper ttyS value.

Next click the key “E”. This opens a new menu, allowing to select the Data Rate, Parity, Data Bits, and Stop Bits. Much more variation exists here to set up the desired service. The Data Rate must be matched to the remote end, and thus must be known. Proper Parity must be set to insure error detection. The number of bits transmitted, either 7 or 8, must be set in order to insure the proper byte length. Finally, the number of Stop Bits, either 1, 1 ½, or 2 needs to be set. This value is not really critical, as it sets the inter-character time. It is normally set to 1.

The last requirement is to set the Flow Control. Two different options exist for transmitting the flow control information, Hardware and Software. Both must be configured as to whether they are on or off. Hardware Flow Control uses a physical wire circuit in the data connector running from the modem to the computer (embedded in an internal modem), and indicates if data may be received, depending upon the buffer availability. Software Flow Control performs the exact same process, but uses an ASCII Control Character to transmit the same information. Either or both may be used, and must be configured in accordance with the remote equipment. For example, if connecting to a Cisco device for configuration to the Console port, Flow Control is not to be used.

This completes the basic setup of the minicom application. The next action is to save the configuration using the “Save Setup as dfl”. Other options may be set up, but the Serial Port must be configured properly for the application to operate properly.

Finally, after the configuration has been saved, clicking on “Exit” will close the setup and open the minicom application. The following screen now opens.

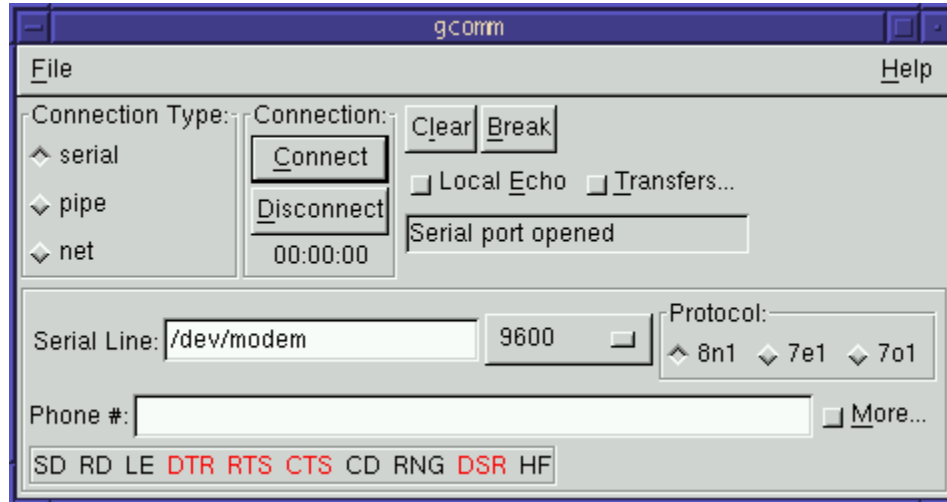
#### 5.5.7.2 Gcomm

Gcomm is the GUI equivalent to minicom, and may be used as an alternative in the X-Windows mode. It is a simple, yet very strong application that allows

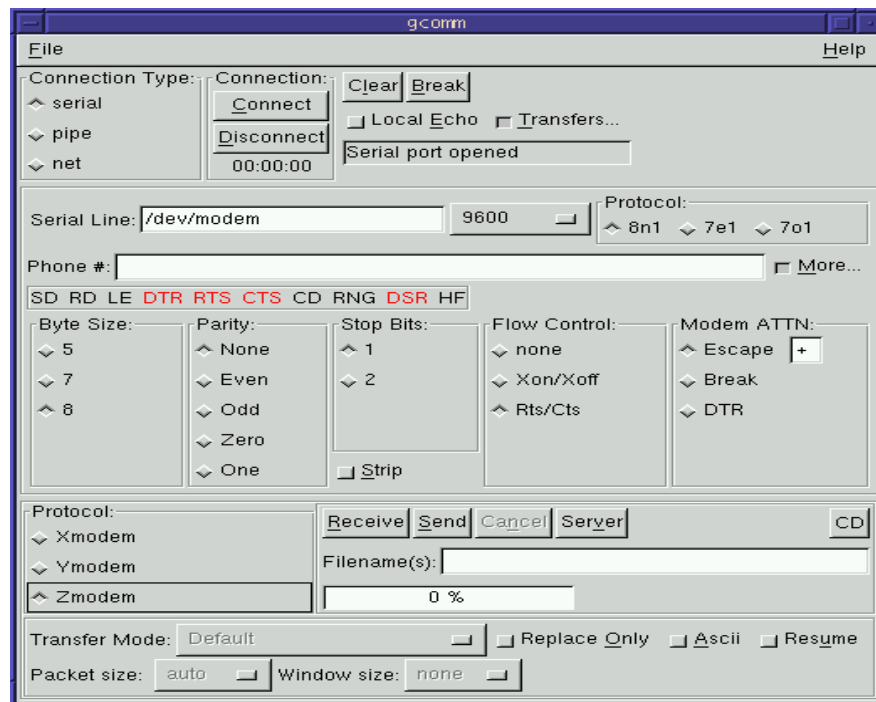


one to perform all of the basic operation for setting up a modem interface and the transmission of data between different locations.

On opening the application, the following screen appears.



By clicking the “More” button (bottom right corner), an extended feature screen opens, allowing additional configuration.



At this time, one enters the remote phone number and clicks on “Connect”. A call will then be attempted. If all is set up correctly, the call will be completed.

In addition to supporting a modem call, note that a network connection may also be made. By clicking the “net” radio button, one may enter the remote IP address.

## 5.6 Network Testing

Two commands are available to test the network operation, ping and traceroute.

### 5.6.1 Ping

One of the most powerful Internet Protocol test tools available is the **ping** command. A ping is a special packet that is transmitted to another station, which responds back with an acknowledgment packet to the originating station.

The format of the command is:

**ping IP\_Address**

Your Linux / Unix system will continue to transmit pings on a continuous basis until you terminate the process with a **CTRL-C (^C)** command. A summary statement will then be printed giving the minimum, average, maximum, and standard deviation times required for all the packets transmitted. Windows performs the same type of function, except that the Windows ping is only issued 4 times before automatically terminating.

Issuing the command produces the result:

```
[root@fried dennis]# ping 192.168.1.13
PING 192.168.1.13 (192.168.1.13) 56(84) bytes of data.
64 bytes from 192.168.1.13: icmp_seq=1 ttl=64 time=0.933 ms
64 bytes from 192.168.1.13: icmp_seq=2 ttl=64 time=0.384 ms
64 bytes from 192.168.1.13: icmp_seq=3 ttl=64 time=0.379 ms
64 bytes from 192.168.1.13: icmp_seq=4 ttl=64 time=0.404 ms
64 bytes from 192.168.1.13: icmp_seq=5 ttl=64 time=0.485 ms

--- 192.168.1.13 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 0.379/0.517/0.933/0.211 ms
```

The final line result displays the round trip time (rtt) for the minimum / average / maximum / and mean deviation for the number of issued pings.

The originator packet is made up of:

- addressee address (packet being sent to)
- addresser address
- sequence number
- ping query statement
- crc

The responder packet is made up of-

- addressee address (packet going back to ping originator)
- addresser address
- sequence number
- ping response statement
- crc

The originating station starts a timer when the packet is transmitted and stops the timer when an acknowledgment is received (round trip time). It then

prints a one line report giving the packet sequence number and the amount of time required to send the packet and get the response. Note that if the response time is less than 1 millisecond (in the case of MS Windows), the report may print **< 10 ms** – this is a programming response error that has been corrected on Linux, giving the response time in microseconds if necessary.

If the hosts table has been set up with the host of interest, then you can use the command:

```
ping {hostname}           for example
ping prof
```

where the system will substitute the proper IP address.

Note that the IP address range of **192.168** is a special address range dedicated to private networks. If your system has an address in this range and you ping an outside address (non 192.168), the ping will probably be refused or not forwarded by the router. This is a special category of addresses set up for private use.

Additional options exist with the ping command. Some of the options include:

- c *n*    Provide a listing of *n* lines
- d        Set SO\_DEBUG option for socket
- f        Flood network with continuously transmitted packets
- i *n*     Transmit a packet every *n* seconds
- l *n*     Send *n* packets as fast as possible before normal transmission
- n        Numeric output only
- p *pad*   Specify padding bytes (up to 16) to fill out packet
- q        Quiet mode
- R        Record route
- r        Direct route (do not use routing table)
- s *n*     Specify size of packet in bytes

Again, to terminate the ping process, press the keys:

**CTRL-C**

#### 5.6.1.1      A Little Theory on What Happens

When you issue a Ping to 127.0.0. 1, the local loopback address, a message is generated by the ping application, which is part of the OSI application Layer 7. This message is structured down to the Transport Layer 4, where it is looped back up through the various Layers to the Ping application. What we have proven is that during the installation of the NIC and appropriate software drivers is that they have been successfully bound together, normally referred to as binding. Note that the local loopback ping does not reach the NIC card.

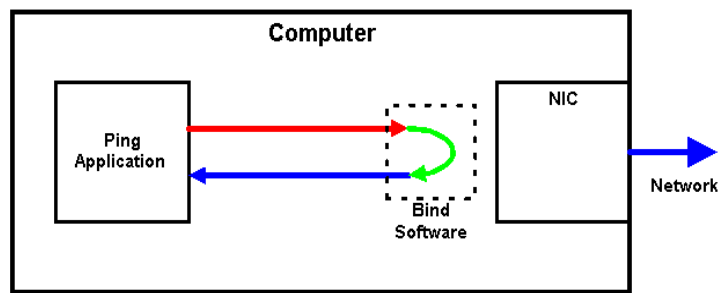


Figure 5.1: Ping to 127.0.0.1

When we transmit a Ping from one host to another, the packet message travels through the first computer through the NIC card, over the network, into the NIC card and finally to the BIND software on the second computer, where it is looped back to the first host again at the Transport Layer 4.

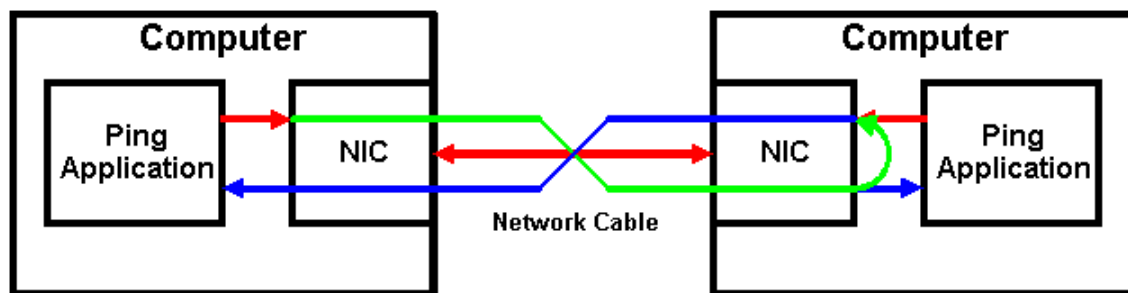


Figure 5.2: Ping to a Different Host

The Ping process uses the Internet Control Message Protocol (ICMP) that provides a test message through the Network Layer of the OSI model. A Ping does not test anything that requires OSI layers 5, 6, or 7, these are required for applications.

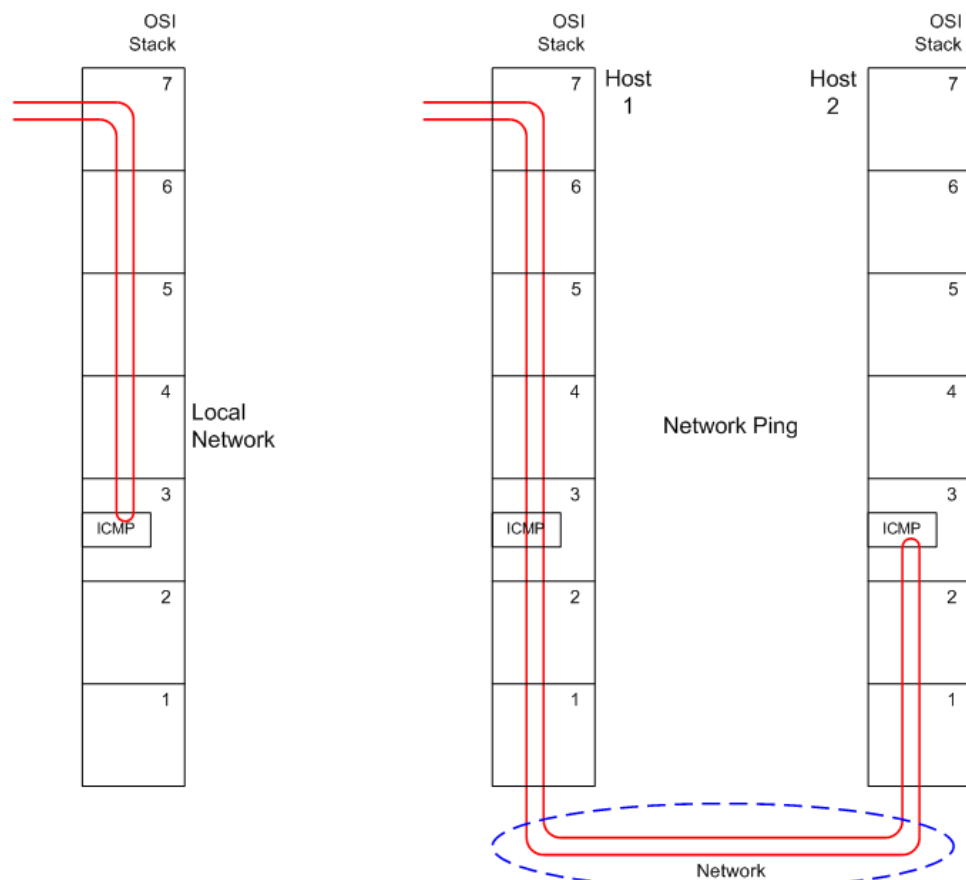


Figure 3: Ping and the OSI Stack

### 5.6.2 Tracing Network Path

It is commonly handy to learn the path that packets take through the network. This may be performed using either the **ping**, the **tracert**, **tracpath** or the **mtr** command.

#### 5.6.2.1 **ping -R ipaddress**

The **ping -R ipaddress** commands augments the basic ping command to print out the route utilized to connect to the remote host. (This is similar to the windows command **tracert IP\_address**). If you want to know the path that the packet utilized to connect to the remote system, this command will display the route and then perform a normal ping.

#### 5.6.2.2 **tracert**

It is often necessary to verify the route that packets transverse while flowing across the Internet. This is easily accomplished using the **tracert** command.

Traceroute issues a series of ICMP packets very similar to a PING packet. By use of the Time To Live (TTL) part of the packet, a transmitted packet may be sent a limited distance into the network. The following is a simple explanation of what happens.

1. The command to a remote location is issued:

**tracert [options] someplace.com**

For this example, lets assume that "someplace.com" is 4 hops away.

2. Originator sets the TTL to 1 and issues an ICMP ping.
3. The first router receives the ping with a TTL value of 1, decrements it to 0, determines that the packet cannot be forwarded, and returns a message to the originator specifying that the TTL was exceeded.
4. Originator notes the time duration of the packet, divides by 2 (one way time); and posts the value to the monitor.
5. Originator sets the TTL to 2 and issues an ICMP ping.
6. The first router receives the ping with a TTL value of 2, decrements it to 1, and forwards it to the second router.
7. The second router receives the ping with a TTL value of 1, decrements it to 0, determines that the packet cannot be forwarded, and returns a message to the originator specifying that the TTL was exceeded.
8. The originator notes the time duration of the packet, divides by 2 (one way time), and posts the value to the monitor.
9. Originator set the TTL to 3 and issues an ICMP ping.
10. The first router receives the ping with a TTL value of 3, decrements it to 2, and forwards it to the second router.
11. The second router receives the ping with a TTL value of 2, decrements it to 1, and forwards it to the third router.
12. The third router receives the ping with a TTL value of 1, decrements it to 0, determines that the packet cannot be forwarded, and returns a message to the originator specifying that the TTL was exceeded.
13. The originator notes the time duration of the packet, divides by 2 (one way time), and posts the value to the monitor.
14. Originator sets the TTL to 4 and issues an ICMP ping.
15. The second router receives the ping with a TTL value of 4, decrements it to 3, and forwards it to the second router.
16. The third router receives the ping with a TTL value of 3, decrements it to 2, and forwards it to the third routers.
17. The third router receives the ping with a TTL value of 2, decrements it to 1, and forwards it to the forth router.
18. The forth router receives the ping with a TTL value of 1, decrements it to 0, determines that the packet can not be forwarded, and returns a message to the originator specifying that the TTL was exceeded.
19. The originator notes the time duration of the packet and posts the value to the monitor. The originator also realizes that the destination has been reached, and therefore additional pings are not required, thus terminating the traceroute process.

The above has one minor variation. Instead of transmitting only one packet to each router, three packets are transmitted. This way we can display the fastest, mid, and longest time to reach each router. The maximum is typically much greater because a DNS is required to learn the routing.

The display on the monitor will show the router that returned the TTL exceeded IP Address, router URL name, and the three ping times. If you do a little reading between the lines, you may be able to determine which cities the packet passed through.

There is one case where we are not able to determine the time to a router. This is where the router has the ping response turned off. In this case, the originator will display a “\*”.

Options for traceroute include:

- l            Display TTL value of returned packet
- m **n**        Set the maximum number of hops (TTL)
- n            Numeric display only
- p **port**     Set base UDP port number
- q **n**        Set number of probes (default = 3)
- r            Bypass normal routing table
- s **IP**        Use **IP** as source address in outgoing probe packets
- t **tos**       Set the Type of Service value
- v            Display in verbose mode
- w **n**        Set time to wait to **n** seconds for response to a probe

The following provides an example listing of each router that the pings take to reach the remote location.

```
[root@fried dennis]# traceroute eugene
traceroute to eugene (65.110.15.200), 30 hops max, 38 byte packets
 1 gateway (192.168.1.1) 1.524 ms 0.304 ms 0.250 ms
 2 10.180.192.1 (10.180.192.1) 15.864 ms 11.295 ms 10.845 ms
 3 12.244.113.33 (12.244.113.33) 13.511 ms 10.490 ms 10.986 ms
 4 12.119.124.17 (12.119.124.17) 10.987 ms 11.297 ms 16.408 ms
 5 gbr1-p100.dlstx.ip.att.net (12.123.16.234) 10.662 ms 10.674 ms 360.404 ms
 6 tbr2-p012401.dlstx.ip.att.net (12.122.12.77) 14.005 ms 12.891 ms 11.631 ms
 7 gar1-p370.dlrx.ip.att.net (12.123.196.97) 11.184 ms 13.042 ms 11.919 ms
 8 12.119.136.30 (12.119.136.30) 12.852 ms 32.183 ms 11.209 ms
 9 so-0-0-0.cr1.dfw2.us.above.net (64.125.28.209) 128.264 ms 11.247 ms 13.498 ms
10 so-4-0-0.mpr4.sjc2.us.above.net (64.125.29.53) 55.269 ms 57.245 ms 56.026 ms
11 so-6-0-0.cr2.sea1.us.above.net (64.125.28.22) 74.031 ms 74.058 ms 76.467 ms
12 209.249.11.173.data-fortress.com (209.249.11.173) 79.886 ms 78.639 ms 77.873 ms
13 a.cust.65-110-0-2.van.data-fortress.com (65.110.0.2) 79.347 ms 79.571 ms 80.425 ms
14 NET-65-110-12-220.van.data-fortress.com (65.110.12.220) 80.447 ms 80.865 ms 81.204 ms
15 eugene (65.110.15.200) 108.384 ms 82.094 ms 82.502 ms
```

### 5.6.2.3 Tracepath

Tracepath is very similar to traceroute, with only one ping per router transmitted.

```
[root@fried dennis]# tracepath eugene
1?: [LOCALHOST] pmtu 1500
1: gateway (192.168.1.1) 2.112ms
2: 10.180.192.1 (10.180.192.1) 14.043ms
3: 12.244.113.33 (12.244.113.33) 16.956ms
4: 12.119.124.17 (12.119.124.17) 15.357ms
5: gbr1-p100.dlstx.ip.att.net (12.123.16.234) 15.218ms
6: tbr2-p012401.dlstx.ip.att.net (12.122.12.77) asymm 7 18.030ms
7: gar1-p370.dlrx.ip.att.net (12.123.196.97) 15.644ms
8: 12.119.136.30 (12.119.136.30) asymm 9 16.585ms
9: so-0-0-0.cr1.dfw2.us.above.net (64.125.28.209) 17.745ms
10: so-4-0-0.mpr4.sjc2.us.above.net (64.125.29.53) 62.392ms
11: so-6-0-0.cr2.sea1.us.above.net (64.125.28.22) asymm 12 88.588ms
12: 209.249.11.173.data-fortress.com (209.249.11.173) 160.058ms
13: a.cust.65-110-0-2.van.data-fortress.com (65.110.0.2) 98.710ms
14: NET-65-110-12-220.van.data-fortress.com (65.110.12.220) 92.915ms
15: eugene (65.110.15.200) 85.727ms
reached
```

Resume: pmtu 1500 hops 15 back 15

This format is a little easier to read, but only provides the largest time to reach each router due to the lookup time of the DNS.

#### 5.6.2.4 mtr

A relatively new utility for displaying the path to a remote site is **mtr**. This utility was written by Matt (last name unknown), but on some systems may be referred to as “My Traceroute”. For more information, go to the home web page at <http://www.bitwizard.nl/mtr/>. It is very similar to **tracpath**, except that the display is dynamic, that is, it continuously transmits a ping packets with different Time To Live (TTL) values. The display then shows these values.

```

Matt's traceroute [v0.54]
steamed.ricepad.org Sun Dec 11 14:59:24 2005
Keys: D - Display mode R - Restart statistics Q - Quit
      Packets
Hostname %Loss Rcv Snt Last Best Avg Worst
1. ccgw.dearroz.net 0% 4 4 0 0 0 0
2. ???
3. 68.87.206.9 0% 3 3 8 7 9 11
4. 68.87.207.81 0% 3 3 7 7 8 8
5. 12.118.225.5 0% 3 3 7 7 8 9
6. tbr2-p013801.dlstx.ip.att.net 0% 3 3 10 9 10 10
7. 12.122.82.229 0% 3 3 8 8 11 16
8. pop1-dls-P3-0.atdn.net 0% 3 3 8 8 9 10
9. bb1-dls-P0-0.atdn.net 0% 3 3 10 9 10 11
10. bb1-hou-P6-0.atdn.net 0% 3 3 14 13 15 16
11. bb1-atm-P7-0.atdn.net 0% 3 3 28 28 29 30
12. bb2-atm-P1-0.atdn.net 0% 3 3 31 28 30 31
13. bb2-rdu-P4-0.atdn.net 0% 3 3 36 36 37 38
14. bb1-rdu-P2-0.atdn.net 0% 3 3 36 36 37 38
15. bb1-vie-P12-0.atdn.net 0% 3 3 47 44 45 47
16. pop2-vie-P0-0.atdn.net 0% 3 3 44 44 45 45
17. dar2-mtc-S0-0-0.atdn.net 0% 3 3 44 44 45 46
18. ???

```

#### 5.6.3 Network Status

Another command available to assist in determining the network status is the **netstat** utility. This command is not intended to test the network connectivity, but rather the host system network information. The command format is:

**netstat [options] [sockets]**

Options include:

- a Show all
- A fam Set address families
- c Continuous update
- e Extensive reporting
- i Show table of all networking interfaces
- l Display netlink kernel messages
- M Displays masqueraded connections
- n Do not resolve names (show IP addresses numerically)
- N Show creation / deletion messages
- o Displays timers
- r Displays routing table

Protocols include:



```

-t      TCP
-u      UDP
-w      raw
-x      UNIX
--ax25  ax25
--ddp   ddp
--ip     inet
--ipx    ipx
--netrom netrom

```

An example of the printout when issuing the netstat command when no outside services are connected would be:

```

Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address          State
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags   Type       State           I-Node Path
unix  10      [ ]     DGRAM                809      /dev/log
unix  3        [ ]     STREAM  CONNECTED      2241     /tmp/.ICE-unix/1310
unix  3        [ ]     STREAM  CONNECTED      2240
unix  3        [ ]     STREAM  CONNECTED      2238     /tmp/.ICE-unix/1336
unix  3        [ ]     STREAM  CONNECTED      2237
. . .
unix  3        [ ]     STREAM  CONNECTED      1464
unix  3        [ ]     STREAM  CONNECTED      1463
unix  3        [ ]     STREAM  CONNECTED      1384     /tmp/.font-unix/fs7100
unix  3        [ ]     STREAM  CONNECTED      1383
unix  4        [ ]     STREAM  CONNECTED      1386     /tmp/.X11-unix/X0
unix  3        [ ]     STREAM  CONNECTED      1374
unix  2        [ ]     DGRAM                1312
unix  2        [ ]     DGRAM                1227
unix  2        [ ]     DGRAM                1179
unix  2        [ ]     DGRAM                1126
unix  2        [ ]     DGRAM                1074
unix  2        [ ]     DGRAM                978
unix  2        [ ]     DGRAM                863
unix  2        [ ]     DGRAM                821
unix  2        [ ]     STREAM  CONNECTED      472
Active IPX sockets
Proto Recv-Q Send-Q Local Address           Foreign Address          State

```

Now if we initiate a telnet session, just to ourself, then the output would appear something like the following:

```

Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address          State
tcp    0  0 192.168.102.149:telnet  192.168.102.149:32778   ESTABLISHED
tcp    0  0 192.168.102.149:32778  192.168.102.149:telnet ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags   Type       State           I-Node Path
unix  11      [ ]     DGRAM                809      /dev/log
unix  3        [ ]     STREAM  CONNECTED      2241     /tmp/.ICE-unix/1310
unix  3        [ ]     STREAM  CONNECTED      2240
unix  3        [ ]     STREAM  CONNECTED      2238     /tmp/.ICE-unix/1336
unix  3        [ ]     STREAM  CONNECTED      2237
unix  3        [ ]     STREAM  CONNECTED      2235     /tmp/.X11-unix/X0
unix  3        [ ]     STREAM  CONNECTED      2234
unix  2        [ ]     DGRAM                2199
. . .
unix  3        [ ]     STREAM  CONNECTED      1383
unix  4        [ ]     STREAM  CONNECTED      1386     /tmp/.X11-unix/X0

```

```

unix 3      [ ]      STREAM  CONNECTED  1374
unix 2      [ ]      DGRAM    1312
unix 2      [ ]      DGRAM    1227
unix 2      [ ]      DGRAM    1179
unix 2      [ ]      DGRAM    1126
unix 2      [ ]      DGRAM    1074
unix 2      [ ]      DGRAM    978
unix 2      [ ]      DGRAM    863
unix 2      [ ]      DGRAM    821
unix 2      [ ]      STREAM  CONNECTED  472
Active IPX sockets
Proto Recv-Q Send-Q Local Address           Foreign Address         State

```

Notice the difference between the two, in the second listing, we have two active Internet connections. The first is from our host to the remote host – ourselves. The second is from a remote host (ourselves) to us, again ourselves. You can tell this by looking at the port number that follows the IP address – 192.168.102.149:telnet (23) [originating] to 192.168.102.149:32778. We have set up a session between our host to port 23 on the remote telnet server. Likewise, the server has is communicating back to us on port 32778 from server port 23. the port number specified by our host (32778) will be different for each new telnet session.

#### 5.6.4 Address Resolution Protocol – arp

Normally we do not track the IP address of everyone. The IP address is considered to be a variable and is utilized, but there is a fixed address attached to every host, called the **MAC Address**. To resolve this difference, the originating computer needs to determine the physical host that is attached to a specific IP.

The originating host issues a “ping like” message called an **ARP (Address Resolution Protocol)** that says “Who out there has the IP Address of *ipadd*. The host who has the specified IP address then responds back with the message “I am the host with IP address *ipadd* and my MAC address is *macadd*.”

After the originating host receives the ARP reply, it creates an entry into its ARP cache (memory) that relates the two. From that time on, packets to the local host with a desired IP Address are actually transmitted to the MAC address.

When we wish to connect to a remote host, we again issue the command with the IP address which is routed to the appropriate local router. That router, assuming it does not yet have the *ipadd* to *macadd* translation in its cache, outputs an ARP packet to locate the local host with the specified *ipadd*. The router then translates the IP Address to the desired MAC address within the packet and forwards it to the designated host.

Of course, I must add the special question – What is the network seal of approval?

**arp**

## **5.7 Host Users**

It is often necessary to determine what users are logged onto a system. There are four commands that provide information as to who is presently logged onto a system and what they are doing.

The commands available are:

|              |                |                 |
|--------------|----------------|-----------------|
| <b>who</b>   | <b>last</b>    | <b>whois</b>    |
| <b>users</b> | <b>lastlog</b> | <b>who am i</b> |
| <b>w</b>     | <b>finger</b>  | <b>whoami</b>   |

### **5.7.1 who**

Who provides a list of the users that are presently logged onto a system.

Several options include:

- i Show user idle time
- q Login names and number of users
- w Message status

With this you can display those which users are presently logged onto the system.

The who command derives its information from two files in the /var directory. These files are in a database format, and are not directly readable. They are:

**/var/run/utmp** and  
**/var/log/wtmp**

A history of all system users is maintained in the /var/log/wtmp file. To view the entries, issue the command:

**who /var/log/wtmp**

The following is an example of the who command.

```
$who
root pts/0 Jul 23 14:48
root pts/1 Aug 1 11:44
```

### **5.7.2 users**

The user command provides a simple quick list of the users presently logged onto the system. If you only need this information, this is the easiest command.

The following is an example of the users command.

```
$ users
hlul root
```

### 5.7.3 w

The w command list what users are on the system and what they are doing. If you need a command to view what the users are up to, then this is the optimum command to use.

The following is an example of the w command.

```
$w
11:55am up 13 days, 22:04, 2 users, load average: 0.08, 0.04, 0.01
USER  TTY  FROM          LOGIN@  IDLE  JCPU  PCPU  WHAT
root  pts/0  -             23Jul03 8days 0.03s 0.03s /bin/cat
root  pts/1  -             11:44am 0.00s 0.21s 0.03s w
```

### 5.7.4 last

The last command provides a list of users, access, source host name, and date / time – duration of logged on time. It also provides the date / time that the current logging session is being maintained.

The following is an example of the last command.

```
$ last
root      pts/1      192.168.1.43    Sun Dec 28 18:08    still logged in
dennis    ftpd14851  192.168.1.43    Tue Dec 23 21:48 - 21:50 (00:01)
dennis    ftpd14844  192.168.1.43    Tue Dec 23 21:45 - 21:47 (00:01)
root      pts/0      -              Mon Dec 22 10:38    still logged in
root      pts/0      -              Fri Dec 19 09:38 - 10:38 (3+00:59)
root      pts/0      -              Thu Dec 18 08:39 - 09:38 (1+00:59)
root      pts/1      -              Wed Dec 17 10:27 - 11:05 (00:37)
root      pts/1      192.168.1.43    Mon Dec 8 09:49 - 11:26 (01:36)
root      pts/1      192.168.1.43    Sun Dec 7 18:03 - 00:37 (06:34)
dennis    ftpd1630   192.168.1.43    Sun Dec 7 13:34 - 13:50 (00:15)
root      pts/1      192.168.1.43    Sun Dec 7 13:13 - 15:55 (02:41)
root      pts/1      -              Sun Dec 7 13:08 - 13:08 (00:00)
root      pts/0      -              Sun Dec 7 13:08 - 08:39 (10+19:30)
reboot    system boot  2.4.7-10        Sun Dec 7 13:06      (21+07:39)
root      pts/1      -              Sun Dec 7 13:03 - down (00:00)
root      pts/1      -              Sun Dec 7 03:38 - 13:03 (09:25)
root      pts/0      -              Sat Dec 6 10:05 - down (1+02:58)
root      pts/1      -              Fri Dec 5 14:28 - 14:28 (00:00)
root      pts/0      -              Fri Dec 5 14:15 - 10:05 (19:50)
reboot    system boot  2.4.7-10        Fri Dec 5 07:53      (2+05:10)
wtmp begins Fri Dec 5 07:53:37 2003
```

### 5.7.5 lastlog

The lastlog command provides a list of all users and when they were last logged on. It is best to observe the list by piping the command to one of the display commands.

The following is an example of the lastlog command.

```
$ lastlog
Username Port From Latest
root      pts/1 192.168.1.43 Sun Dec 28 18:08:00 -0600 2003
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
**Never logged in**
```

```

operator      **Never logged in**
games         **Never logged in**
gopher        **Never logged in**
ftp           **Never logged in**
nobody        **Never logged in**
mailnull      **Never logged in**
rpm           **Never logged in**
xfs           **Never logged in**
ntp           **Never logged in**
rpc           **Never logged in**
gdm           **Never logged in**
rpcuser       **Never logged in**
nfsnobody     **Never logged in**
nscd          **Never logged in**
ident         **Never logged in**
radvd         **Never logged in**
postgres      **Never logged in**
apache        **Never logged in**
squid         **Never logged in**
named         **Never logged in**
pcap          **Never logged in**
amanda        **Never logged in**
junkbust      **Never logged in**
mailman       **Never logged in**
mysql         **Never logged in**
ldap          **Never logged in**
pvm           **Never logged in**
dennis      ftp   192.168.1.43 Tue Dec 23 21:48:44 -0600 2003
dennis2       **Never logged in**
Dennis2       **Never logged in**
brown         **Never logged in**
longgrain     **Never logged in**
mailer        **Never logged in**
utd           **Never logged in**

```

### 5.7.6 finger

The finger command is designed to display details of a specific user, and their general status.

The following is an example of the finger command.

```

$finger dennis
Login: dennis                      Name: Dennis R. Rice
Directory: /home/dennis          Shell: /bin/bash
Last login Thu Apr 24 13:21 (CDT)  on ftp from ricent
No mail.
No Plan.

Login: drice                      Name: Dennis R. Rice
Directory: /home/drice           Shell: /bin/bash
Never logged in.
No mail.
No Plan.

```

Note that there are two users on the system that match the name of dennis, hence two reports are provided.

**5.7.7 whois**

The command **whois** provides information of registered .com, .net, and .edu domains. Thus you can find out basic information about general public domain URLs.

An example of the whois command is:

```
$ whois yahoo.com
[whois.crsnic.net]
```

**Whois Server Version 1.3**

*Domain names in the .com and .net domains can now be registered with many different competing registrars. Go to <http://www.internic.net> for detailed information.*

```
YAHOO.COM.WANADOODOO.COM
YAHOO.COM.TWIXTEARS.COM
YAHOO.COM.TW
YAHOO.COM.SUPERCBCENTER.COM
YAHOO.COM.SG
YAHOO.COM.PURRFURRED.COM
YAHOO.COM.OPTIONSCORNER.COM
YAHOO.COM.IS.N0T.AS.1337.AS.SEARCH.GULLI.COM
YAHOO.COM.DALLARIVA.COM
YAHOO.COM.BR
YAHOO.COM.BERKELEYNATURALBEAUTIES.COM
YAHOO.COM.AU
YAHOO.COM
```

*To single out one record, look it up with "xxx", where xxx is one of the of the records displayed above. If the records are the same, look them up with "=xxx" to receive a full display for each record.*

```
>>> Last update of whois database: Sun, 28 Dec 2003 18:28:17 EST
<<<
```

```
...
```

*The Registry database contains ONLY .COM, .NET, .EDU domains and Registrars.*

```
$
```

The command may also be issued as:

**whois IP-Address**

This will then provide the registrar information for the URL.

### 5.7.8 who am i

The ‘who am i’ command displays basic information of the system to which one is logged on to. This would be most useful in a script that needed to learn the user details.

The following is an example of the ‘who am i’ command.

```
$who am i
friedrice.dal.devry.edu!root pts/1 Aug 1 11:44
```

### 5.7.9 whoami

You may also key in “whoami”, obtaining the results.

```
$ whoami
root
$
```

Thereby learning the username of the logged in user.

## 5.8 Network Messaging (Not Complete)

There are four commands involved in sending messages across the network to another user. They are:

1. **mesg** Specifies if a message may be received
2. **talk** Sends a message to another user logged onto your system
3. **wall** Sends a message to all other users that are currently logged on
4. **write** Establishes a terminal session between yourself and another user

### 5.8.1 mesg

The primary command is the one that sets permissions as to whether your system allows messages to be received.

Issuing the **mesg** command by itself provides a response as to the present status. Issuing the command with either a ‘y’ or ‘n’ specifies whether you wish to have another user send messages to you.

### 5.8.2 talk

The talk command allows a user to send a message to another user that is logged onto their own system. The format of the command is:

```
talk username where username is the name of the other user if a
user is logged in to the system multiple times, then
you need to use the terminal port, or “ttyname”
```

The other user will get a message specifying that you wish to talk. In order to communicate, the other user must respond back with a like command to you.

### 5.8.3 **wall**

The wall command is used to send a message to all other systems on your local network segment that are logged on and allow messages. The format of the command is:

```
wall message [EOF]
```

The message must be terminated with the End of File (EOF) character, which is a CTRL-D.

### 5.8.4 **write**

The write command sets up a terminal session between you and the remote system. It is the command line mode of AOL's Instant Messenger. The format of the command is:

```
write username (IP address)
```

You will have a session initiate from your side, but before full communications can proceed, the other person must respond with:

```
write your-username@your-hostname
```

Now you will both have a split screen, where the top is the remote user and your communications is the lower half of the screen.

When you are finished with your immediate transmission, it is normal protocol to finish your sentence with an "-o", signifying "over" (from the radio communication days). When you are finished with the session, end the session with an "-oo", for "over and out".

When complete, either party may issue an EOF character (CTRL-D) to terminate the session.

## 5.9 **DNS Host Name Lookup**

As a Network Administrator, it is commonly required to determine the IP address of specific Enterprise (Internet) servers of another domain. In normal operation of various applications, this is accomplished automatically, converting the URL that you type in to an IP address.

There are three commands that are utilized to determine the IP address of a specific server. These are:

1. **nslookup**
2. **host**
3. **dig**

All three commands perform the same basic function, to various degrees.

If we issue the command:

```
nslookup yahoo.com
```

we get back the IP address of the Name Server. Of course we know that yahoo has additional servers, namely the web and mail servers. We can often find the IP address of these servers by adding a prefix to the domain name, such as:

```
nslookup mx.yahoo.com
```



Now we will have the IP address of both the mail server and the name server. Common abbreviations for server functions are:

|             |            |
|-------------|------------|
| Name Server | <b>ns</b>  |
| Mail Server | <b>mx</b>  |
| Web Server  | <b>www</b> |
| FTP Server  | <b>ftp</b> |

Although it is not required to have the above designations, it is good practice to have them to maintain Internet constancy. For example, a company might name their mail server as mailer.company.com . In their Name Server, they should have an entry such as:

|                             |           |                                      |
|-----------------------------|-----------|--------------------------------------|
| <b>IN(ternet)</b>           | <b>MX</b> | <b>mailer.company.com</b>            |
| <b>Mailer(.company.com)</b> | <b>IN</b> | <b>A(ddress) {Server-IP-Address}</b> |

This states that the URL of mx.company.com points to the server mailer.company.com, and that the server mailer.company.com has the IP address of {server address}.

Additional information as to the configuration of the name server will be provided in one of the server labs.

### 5.9.1 nslookup

**nslookup** was the original utility to determine an IP address of a specific server on a remote network. All operating systems support this command, although it is now being replaced by the **dig** utility. (At this time, Microsoft only supports the nslookup command.)

As an example, if we issue the command:

**nslookup yahoo.com**

we get back:

**Server: 204.127.202.4**  
**Address: 204.127.202.4#53**

**Non-authoritative answer:**

**Name: yahoo.com**  
**Address: 66.218.71.198**  
**Name: yahoo.com**  
**Address: 64.58.79.230**

Thus we can see that the address of the yahoo.com domain has two addresses, 66.218.71.198 and 64.58.79.230 . The address of the name server that gave this information to us is 204.127.202.4 .

Now if we issue the command:

**nslookup ns.yahoo.com**

we get back:

**Server: 204.127.202.4**  
**Address: 204.127.202.4#53**

**Non-authoritative answer:**

**Name: ns.yahoo.com**

**Address: 204.71.177.33**

Now we see that the ns server is 204.71.177.33, but that the server that provided us the information came from the 204.127.202.4 system, from port 53 (the standard DNS server port).

### 5.9.2 host

Another utility to provide basic information without overloading the user with detail is the **host** command. Issuing the command:

**\$ host yahoo.com**

we get back:

**yahoo.com. has address 64.58.79.230**

**yahoo.com. has address 66.218.71.198**

Now we see that yahoo.com has two different addresses assigned to it, 64.58.79.230 and 66.218.71.198 . Note that the address of the name server has not been provided.

If we issue the command:

**host mx.yahoo.com**

we get back:

**mx.yahoo.com. is an alias for w1.mx.vip.sc5.yahoo.com.**

**w1.mx.vip.sc5.yahoo.com. has address 216.136.232.200**

Now we see that the mail server is an alias for the server w1.mx.vip.sc5.yahoo.com, and that the server has the IP address of 216.136.232.200 .

### 5.9.3 dig

The last command that is used is **dig**. It is to replace the **nslookup** command and is supported by both Unix and Linux. The dig utility provides the user with much more comprehensive information. Issuing the command:

**dig yahoo.com**

gives:

**; <<>> DiG 9.1.3 <<>> yahoo.com**

**;; global options: printcmd**

**;; Got answer:**

**;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8210**

**;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 5,  
ADDITIONAL: 5**

**;; QUESTION SECTION:**

**;yahoo.com. IN A**

**:: ANSWER SECTION:**

```

yahoo.com.      208   IN   A    64.58.79.230
yahoo.com.      208   IN   A    66.218.71.198

```

**:: AUTHORITY SECTION:**

```

yahoo.com.      27101 IN   NS    NS1.yahoo.com.
yahoo.com.      27101 IN   NS    NS2.yahoo.com.
yahoo.com.      27101 IN   NS    NS3.yahoo.com.
yahoo.com.      27101 IN   NS    NS4.yahoo.com.
yahoo.com.      27101 IN   NS    NS5.yahoo.com.

```

**:: ADDITIONAL SECTION:**

```

NS1.yahoo.com.  131778 IN   A    66.218.71.63
NS2.yahoo.com.  41658  IN   A    209.132.1.28
NS3.yahoo.com.  126852 IN   A    217.12.4.104
NS4.yahoo.com.  128324 IN   A    63.250.206.138
NS5.yahoo.com.  120892 IN   A    64.58.77.85

```

**:: Query time: 89 msec****:: SERVER: 204.127.202.4#53(204.127.202.4)****:: WHEN: Sat Jan 11 16:12:24 2003****:: MSG SIZE rcvd: 229**

Thus we can now see in the Answer Section that yahoo.com has two addresses and five name servers, named NS1 through NS5 and we have received the IP address for each. We also observe that to answer our query, it took 89 milli-seconds on the date and time and finally, the message returned was 229 Bytes long.

A lot more information is provided to the user. In fact we can learn an extensive amount of information about a company's enterprise servers. It is important that as an administrator you treat this information properly.

### **5.10 Remote User Information**

In some cases, it is necessary to obtain information about a user on either your host or on a remote system. This is done by using the command:

```

finger username           or
finger username@hostname

```

On a local network, this might not be a real concern, but to obtain information from a remote domain can create a major information problem. By today's standards, it is not appropriate to hand out information about a user across the Internet. As such, you will typically find that the finger command does not work on a remote domain as the finger service has been disabled.

## 5.11 User Mail<sup>1</sup>

When we log onto a system as the administrator you will often observe that you have new mail. This may also occur when logged on under a regular username. This is mail that has been generated by the either as a security notice, a system event, or from another user on the your same system. Email is designed to transfer ASCII text, but with the advent of images and audio attachments, an additional protocol was added to transfer attachments using the Multimedia Internet Mail Standard, commonly known as MIME.

A user's mail is stored in the **/var/spool/mail** directory under your username. As the administrator, the file is called **root**. There are three fundamental utilities available for reading and sending the mail, called **mail**, **elm** and **pine**. Mail, elm and pine, allow one to send mail as a Mail User Agent (MUA).

There is at least one other mail reader, **mutt**. It also is a command line editors. For these editors, the user must acquaint themselves with the additional command options by referring to the man pages.

### 5.11.1 mail

The application mail is probably the oldest and most lightweight of the three mail readers. To access your mail when using the **mail** utility, issue the command:

```
mail
```

#### 5.11.1.1 Receiving Mail

When you start **mail**, you are presented with:

```
Mail version 8.1.6 6/93. Type ? for help.
```

```
"/var/spool/mail/username": 10 messages 10 new
```

```
> N 1 root@hostname date - time value originator  
:
```

You may read the first message by hitting ENTER. Hitting the ENTER key again will display the next message for reading; successive ENTERs will display the following messages. After you have read a message and terminated the session (letter **q** at the **&**), the read messages are stored in the '**mbox**' in your home directory.

#### 5.11.1.2 Sending Mail

To send an email to another user, you use the form:

```
mail username@domainname -s "mail-subject"
```

You then type in your message. To terminate the message and send it, create a blank line and enter an EOF character – CTRL-D (^D).

#### 5.11.1.3 General Commands

The following commands are available for **mail**.

- +** Move to next email
- Move to previous email
- ?** Display helpful summary of commands on the screen

<sup>1</sup> Red Hat Linux Networking and System Administration, RedHat Press (Wiley Publishing)

|     |   |
|-----|---|
| R   | Reply to sender (only)  |
| r # | Reply to sender and all others on recipient list of message # |
| d   | Deletes message   |
| h   | Show the list of emails                                       |
| n   | Go to next email and list it                                  |
| t   | List the current message                                      |
| ?   | Display available commands                                    |
| s   | Send a message  |
| #   | Read numbered message   |
| q   | Quit and save read email messages                             |
| x   | Quit and do not save read email messages                      |

### 5.11.2 elm

The second option for reading and sending mail is the **elm** application. This is a nice application that provides a simple graphical interface, utilizing a **vi** style of text entry. All operation is performed from a command entry from within the program. General commands are:

|   |                          |
|---|--------------------------|
| D | Delete message           |
| U | Undelete message         |
| M | Create a message to send |
| R | Reply to message         |
| F | Forward message          |
| Q | Quit                     |

### 5.11.3 pine

Pine is another user mail program. It is screen oriented with a limited set of functions geared towards the novice user, but has options for the power user. For a user interface (graphical) it utilizes the **pico** editor, which is part of the installed pine package. Fedora does not include this utility. Pine is no longer included with Fedora Core.

Whereas the mail utility transfers read (but not deleted) messages to the `~/mbox` file, pine does not. From the main screen, one may cycle through the received messages, reply, delete, or send messages.

To execute pine, issue the command:

**\$ pine**

Options include:

|   |  |
|---|--|
| ? | Get help using Pine                          |
| C | Compose a new message                        |
| I | View the message index in the current folder |
| L | List / Select a folder to view               |
| A | Display / Update the address book            |
| S | Configure Pine                               |
| Q | Terminate the present session                |

Using the **N** or **P** keys, the Up / Down arrows, or the specified key, you can select the desired menu option.

Note at the bottom of the screen is the familiar command list that was first observed when using pico.

### 5.11.3.1 Receiving Mail using Pine

To view received mail, click "**L**" to select the folder that you wish to view. Typically, the desired folder is the **Inbox**. Selecting the Inbox displays the received messages. One then selects the desired message by using the up / down arrows, and then clicks the ENTER button.

After reading the message, you may Reply (**R**), Save (**S**) the message, Forward (**F**) to another user, Delete (**D**) the message, or one of the other options listed at the bottom of the screen. When you terminate the Pine session, you will be asked to verify that you want to delete the deleted messages – it is done for your protection.

### 5.11.3.2 Sending Mail using Pine

To compose a new message, click "**C**" to open the composition screen. The screen is now set up as a set of prompts for generating the mail.

First you input the address of the user you wish to send the mail to. The name can be a username on your system, a name from the address book, or a fully qualified email address. Next, you are able to enter addresses for additional users as Carbon Copy (CC). Following the CC, you can add an attachment to the mail. Finally, the mail Subject is entered.

Now you can enter the text of your mail. Enter the message as in any system that you are use to. At the end of the message, when you are ready to send it, click on **CTRL-X**. Pine will request a confirmation of the send action.

### 5.11.3.3 Other Features

Other features of Pine are available, such as the Address book. The user must be investigate the many options available through the help menu.

## 5.12 Command Line Browser

Before the days of the graphical browser, one was able to look up information using the application called **lynx**.

Lynx is a fully-featured World Wide Web (WWW) client for users running cursor-addressable, character-cell display devices (e.g., vt100 terminals, vt100 emulators running on Windows 95/NT or Macintoshes, or any other "curses-oriented" display). It will display hypertext markup language (HTML) documents containing links to files residing on the local system, as well as files residing on remote systems running Gopher, HTTP, FTP, WAIS, and NNTP servers. Current versions of Lynx run on Unix, VMS, Windows 95/NT, 386DOS and OS/2 EMX.

To initiate lynx, from the command line, issue the command:

**lynx URL**

For example, issuing the command:

**lynx yahoo.com**

Gives (just a portion):

**Yahoo! (p1 of 9)****Yahoo!****[perhearts.gif] Yahoo! Personals - It's time to give fate a nudge.  
Create your profile or search for free.****Search for: \_\_\_\_\_ [on the Web \_\_\_\_\_]****Yahoo! Search o Advanced  
o Preferences****New! Top 10 Holiday Gifts - DVDs, CDs, Books, More  
Shop Auctions, Autos, Classifieds, Real Estate, Shopping, Travel  
Find HotJobs, Maps, People Search, Personals, Yellow Pages  
Connect Chat, GeoCities, Greetings, Groups, Mail, Messenger,  
Mobile  
Organize Addresses, Briefcase, Calendar, My Yahoo!, PayDirect,  
Photos  
Fun Games, Horoscopes, Kids, Movies, Music, Radio, TV  
Info Finance, Health, News, Sports, Weather More Yahoo!...****Yahoo! Health - World AIDS Day****...****5.13 Downloading a Web Page Source File**

A web page may be downloaded and saved as a separate file in a non-interactive mode by using the utility **wget**. This file may then be processed as desired. Many options are available in the use of this utility, of which only a few will be explained. To utilize, issue the command:

**wget {site-name or IP address}**

This creates a default file with the name of **index.html**. If the filename already exists, an incremental name will be created, such as **index.html.1**.

For example, if one wished to download the web page from **yahoo.com**, one would issue the command:

```
$ wget yahoo.com
--19:29:26-- http://www.yahoo.com/
=> 'index.html.1'
Resolving yahoo.com... 66.94.234.13
Connecting to www.yahoo.com [66.94.230.48]:80... connected.
HTTP request setn, awaiting response . . . 200 OK
Length: unspecified [text/html]
```

```
[ <=> ] 30,266 187.0K/s
19:29:26 ( 187.06 KB/s) - 'index.html.1' saved [30,266]
```

Here we have downloaded the source page for **yahoo.com**, and it was saved as **index.html.1**. The time of download was at 7:29 PM, the download rate was 187.06 K bytes per second, and the file size was 30,266 bytes.

Options include:

- b Operate in background mode.
- o logfile Output error messages to the the specified logfile.
- nv Run in a non-verbose mode, default operation is verbose.
- i inputfile Read a list of URLs from the specified inputfile.
- O filename Use the file “filename” rather than “index.html”.

### 5.14 Transferring Data to and from a Web Server

Data may be transferred to or from a server in a non-interactive mode, using the appropriate protocol, by using the utility **curl**. One of the most common uses is to download a file from an FTP or HTTP server. This utility may be incorporated into a script, thus allowing transfer of a file without user intervention. Such a case might be to download the latest version of a file, when you know the URL and filename.

To download a file, issue the command:

```
$ curl [option] {URL / IP Address} / filename > capture-filename
```

The desired data will be downloaded and written to the specified filename using the redirector.

For example, you might wish to download my public gpg key (covered in a later chapter) for file encryption. You would issue the command:

```
$ curl dearroz.pointclark.net / drricekey.asc.html > drricekey.asc
```

| % Total         | % Received      | % Xferd    | Average      | Speed ...     |
|-----------------|-----------------|------------|--------------|---------------|
|                 |                 |            | <b>Dload</b> | <b>Upload</b> |
| <b>100 1336</b> | <b>100 1336</b> | <b>0 0</b> | <b>1870</b>  | <b>0</b>      |

Here we observe that the file has been totally transferred (100%), the file transfer was 1336 bytes long, and the download rate was 1870 bytes per second. The output indicates that there was no uploaded file.

An immense number of options are available for using the command.

Several options include:

- a Append to the target file.
- b data The data specified is cookie information previously received from the server.
- c cookie Transmit the specified cookie.
- o filename Write the output to the filename specified, this is the same as using the redirector.
- s Operate in the silent mode, not showing the progress or errors.
- T filename Specifies the file to be uploaded.

### 5.15 Commands Used in this Chapter

- & Forces an application to run in the background
- arp A networking tool to evaluate IP – MAC address correlation



|                      |  |
|----------------------|--|
| curl                 | Downloads a file from a URL specified site   |
| dig                  | A DNS IP Address lookup application  |
| DISPLAY              | An user environmental value for screen display   |
| ethereal             | A GUI Ethernet sniffer tool, FREE  |
| elm                  | An application to transmit and receive email across a network  |
| env                  | Displays the user's environment values   |
| export               | Writes a variable to the user's environment  |
| finger               | An application to obtain detailed information about a user   |
| ftp                  | An application to transfer files from one host to another  |
| gimp                 | An application for manipulating images   |
| grep                 | A utility to search for a specified string   |
| ifconfig             | Displays or changes a host IP Address  |
| host                 | A DNS IP Address lookup application  |
| last                 | Displays a list of users who have logged on and when they last logged on   |
| lastlog              | Displays a list of users and when they last logged on  |
| linuxconf            | A configuration tool for system network configuration  |
| mail                 | An application to transmit and receive email across a network  |
| mesg                 | Sets a configuration value as to whether a remote user may send messages to a logged on user with either talk or write |
| mtr                  | Displays a dynamic traceroute.   |
| mutt                 | An application to transmit and receive email across a network  |
| netstat              | Lists the status of port usage. Used to display outside connections  |
| Network Configurator | A Network Configuration Tool under X   |
| nslookup             | A DNS IP Address lookup application  |
| pine                 | An application to transmit and receive email across a network  |
| ping                 | An application to test connectivity to another host  |
| sendmail             | An application to transmit and receive email across a network, also a mail server daemon                               |
| ssh                  | Secure Shell, encrypted Telnet session   |
| su                   | An application to switch to a different username   |
| talk                 | An application to initiate and converse with another user  |
| tcpdump              | An Ethernet sniffer tool, very basic   |
| telnet               | An application to access a remote host   |
| tracpath             | An application to display all routers on the path to a remote host via a ping, uses only one ping                      |

|            |   |
|------------|---|
| traceroute | An application to display all routers on the path to a remote host via a ping, uses three pings   |
| tty        | A utility to display which terminal one is operating from   |
| users      | Displays a list of users presently logged onto the system   |
| w          | Displays a list of users presently logged onto the system and the application they are performing |
| wall       | An application to send a message to another user  |
| wget       | Downloads a web site source code file   |
| who        | Displays who is on the system   |
| write      | An application to send a message to another user  |
| xhost      | A utility to remotely display an X-Window's display on a remote host                              |
| xinetd     | Re-initializes the Internet Super Host – re-reads networking values                               |

### **5.16 Chapter Review Questions**

1. You desire to obtain a remote web page from the CLI. What command is issued?
  - a. Konqueror
  - b. Netscape
  - c. lynx
  - d. vi
2. You need to transfer files to and from a remote server, what Internet application would be used?
  - a. email
  - b. ftp
  - c. lynx
  - d. telnet
3. You need to test connectivity between your host and a remote system. What Internet application would be used?
  - a. ping
  - b. telnet
  - c. arp
  - d. ftp
4. As administrator, you need to verify the status and address of the interfaces. What command is issued?
  - a. ping
  - b. telnet
  - c. ifdown
  - d. ifconfig

5. Which of the DNS lookup commands displays the query, answer, and additional information if available?
  - a. nslookup
  - b. host
  - c. dig
  - d. all of the above
6. You need to download a specific file from a web site, which command would be used?
  - a. curl
  - b. dig
  - c. ftp
  - d. wget
7. You wish to download the source code file of a web site, which command would be used?
  - a. curl
  - b. ftp
  - c. http
  - d. wget
8. In order to modify a remote server, what command must be issued in order to upgrade to administrator rights?
  - a. su
  - b. sudo
  - c. telnet
  - d. ftp
9. As the administrator, you need to see who has logged onto the system in the past. What command is issued?
  - a. whois
  - b. who
  - c. users
  - d. last
10. What is the command to display the address resolution table?
  - a. host
  - b. list
  - c. netstat
  - d. arp -a
11. What keystroke is used to terminate a ping?
  - a. ^Z
  - b. ^D
  - c. ^C
  - d. ^Q
12. You need to access a remote server to make configuration changes. What Internet application would you use?
  - a. arp
  - b. telnet
  - c. ftp
  - d. su

13. You wish to not be interrupted by messages from other users. What command is issued?
  - a. kill n
  - b. mesg y
  - c. del y
  - d. mesg n
14. An interface needs to be cycled in order to activate a new IP address. What set of commands are used?
  - a. ifconfig / ping
  - b. ifdown / ifup
  - c. down / up
  - d. ethup / ethdown
15. What is the path and filename where the IP Address of the name server is stored?
  - a. /var/resolv.conf
  - b. /var/named.conf
  - c. /etc/named.conf
  - d. /etc/resolv.conf
16. As the administrator, you need to learn what users are presently logged onto a system. What commands may be used?
  - a. users, last
  - b. who, users
  - c. last, who
  - d. who, whois
17. If an interface is to be configured for dhcp, what line in the ifcfg file must be set?
  - a. DHCP
  - b. STATIC
  - c. PROTO
  - d. BOOTPROTO
18. You need to learn information about a registered domain. What command is issued?
  - a. w
  - b. who
  - c. whois
  - d. registrar
19. What is the path and filename where a user may store an IP address to name resolution?
  - a. /var/hosts
  - b. /etc/hosts
  - c. /var/resolv.conf
  - d. /etc/resolv.conf

20. As the administrator, you need to know what applications each user is running. What command is issued?
  - a. who
  - b. whois
  - c. w
  - d. run
21. What is the directory path and file where an Ethernet interface address is stored?
  - a. /etc/sysconfig/network/ifcfg-ethX
  - b. /etc/sysconfig/ifcfg-ethX
  - c. /etc/network/ifcfg-ethX
  - d. etc/sysconfig/network-scripts/ifcfg-ethX
22. As administrator, you need to learn the IP address for each interface. What command is issued?
  - a. ifup
  - b. ifdown
  - c. ifconfig | less
  - d. interface | less
23. What application is used to remotely display an X-Windows display to a remote host?
  - a. hosts
  - b. ssh
  - c. telnet
  - d. xhost

## Chapter Index

|                                       |      |                           |       |
|---------------------------------------|------|---------------------------|-------|
| <b>A</b>                              |      | finger                    | 29    |
| Address Configuration                 | 5    | FTP                       | 12    |
| Address Resolution                    | 8    | anonymous user            | 12    |
| Address Resolution Protocol – arp     | 26   | ftp user                  | 12    |
| Application                           |      | FTP Termination - bye     | 13    |
| Gcomm                                 | 16   | <b>G</b>                  |       |
| Minicom                               | 15   | Gcomm                     | 16    |
| arp                                   | 8    | <b>H</b>                  |       |
| <b>C</b>                              |      | Host User                 |       |
| Changing the IP Address               | 4    | finger                    | 29    |
| Command Line Browser                  | 38   | lastlog                   | 28    |
| Configuring System                    | 6    | who                       | 27    |
| <b>D</b>                              |      | whoami                    | 31    |
| dig                                   | 34   | whois                     | 30    |
| Directory                             |      | Host Users                | 27    |
| /etc                                  | 10   | <b>I</b>                  |       |
| /var/ftp                              | 12   | IP Address Table          | 9     |
| /var/spool/mail                       | 36   | <b>L</b>                  |       |
| DNS Host Name Lookup                  | 32   | last                      | 28    |
| DNS Lookup                            |      | lastlog                   | 28    |
| host                                  | 34   | linuxconf                 | 7     |
| DNS Name Lookup                       |      | Local Hosts Table         | 9     |
| nslookup                              | 33   | localhost                 | 10    |
| DNS Server                            |      | <b>M</b>                  |       |
| FTP ftp                               | 33   | MAC Address               | 8, 26 |
| Mail mx                               | 33   | mail                      | 36    |
| Name ns                               | 33   | General Commands          | 36    |
| Web www                               | 33   | Manual Editing IP Address | 7     |
| Dynamic Address                       | 6    | mesg                      | 31    |
| <b>E</b>                              |      | Minicom                   | 15    |
| elm                                   | 37   | <b>N</b>                  |       |
| <b>F</b>                              |      | netcfg                    | 7     |
| File                                  |      | Netstat                   | 25    |
| /etc/hosts                            | 10   | Network                   |       |
| /etc/inittab                          | 5    | IPADDR                    | 6     |
| /etc/rc.d/init.d/network              | 5    | Network Messaging         | 31    |
| /etc/resolv.conf                      | 10   | Network Ping              | 18    |
| /etc/sysconfig/network                | 5    | Network Status            | 24    |
| /etc/sysconfig/network-scripts/ifcfg- |      | <b>P</b>                  |       |
| eth0                                  | 5pp. | pico                      | 37    |
| /sbin/ifconfig                        | 6    | pine                      | 37    |
| /sbin/route                           | 6    | Commands                  | 37    |
| /var/log/wtmp                         | 27   | Receiving Mail            | 38    |
| /var/run/utmp                         | 27   | Sending Mail              | 38    |
| /var/spool/mail/root                  | 36   | Ping                      | 18    |
| ~/mbox                                | 36   | Local Loopback            | 19    |

|                                  |        |                                 |    |
|----------------------------------|--------|---------------------------------|----|
| Remote Loopback                  | 20     | ftp                             | 12 |
| Ping - A Little Theory           | 19     | host                            | 34 |
| Ping Termination - ^C            | 19     | ifconfig                        | 4  |
| R                                |        | ifconfig interface              |    |
| Receiving Mail                   | 36     | ifconfig interface up           | 4  |
| Remote Access                    | 11     | ifdown                          | 5  |
| Remote Modem Access              | 15     | last                            | 28 |
| Remote Name Lookup               | 10     | lastlog                         | 28 |
| Remote User Information          | 35     | linuxconf                       | 7  |
| resolv.conf                      |        | lynx                            | 38 |
| domain                           | 11     | mail                            | 36 |
| nameserver                       | 11     | mesg                            | 31 |
| search                           | 10     | mtr                             | 24 |
| rlogin                           | 15     | netcfg                          | 7  |
| S                                |        | netstat                         | 25 |
| Sending Mail                     | 36     | nslookup                        | 33 |
| Static IP Address                | 5      | pico                            | 37 |
| subnet mask                      |        | pine                            | 37 |
| Secure Shell (ssh)               | 14     | ping                            | 18 |
| Subnet Mask                      | 4      | Ping                            | 18 |
| ifconfig                         | 4      | ping -R ipaddress               | 21 |
| T                                |        | su                              | 11 |
| talk                             | 31     | talk                            | 31 |
| tcpdump                          | 9      | tcpdump                         | 9  |
| Telnet                           | 11     | telnet                          | 11 |
| Telnet Termination - exit        | 12     | traceroute                      | 21 |
| Terminate                        |        | w 28                            |    |
| Ping - ^C                        | 19     | wall                            | 32 |
| Termination                      |        | wget                            | 39 |
| FTP - bye                        | 13     | who                             | 27 |
| Telnet - exit                    | 12     | who am i                        | 31 |
| Trace Route                      | 21     | whoami                          | 31 |
| U                                |        | whois                           | 30 |
| URL                              |        | write                           | 32 |
| www.ethereal.com                 | 9      | xinetd                          | 4  |
| URL - Universal Resource Locator | 9      | W                               |    |
| User Mail                        | 36     | w 28                            |    |
| users                            | 27     | wall                            | 32 |
| Utility                          |        | who                             | 27 |
| arp                              | 26     | who am i                        | 31 |
| arp -a                           | 8      | whois                           | 30 |
| curl                             | 40     | write                           | 32 |
| dig                              | 34     | X                               |    |
| elm                              | 37     | X Windows Network Configuration |    |
| finger                           | 29, 35 | Utility                         | 7  |

