Chapter 7

Network Administration

This chapter introduces the user to several networking applications that are used to administer a system on a network. Additional applications are discussed that allow one to monitor the local network.

Concepts Learned in this Chapter

- Network applications and utilities for monitoring a system and network

Table of Contents

7.1 System Hostname

Every system is normally given a hostname. In order to establish connectivity on a MS Windows system, it is absolutely required; whereas on a Unix / Linux system it is optional, but strongly recommended. When the hostname is combined with the domain name, it is called the **Fully Qualified Host Name (FQHN)**.

The hostname is simply a name given to the computer system. It allows one to provide a more humanistic reference that may be utilized in referencing itself.

At the command prompt, one may query the hostname by simply issuing the command:

hostname

To change the hostname of a system, we issue the command:

hostname newname

The information is maintained in the **/etc/sysconfig/network** file, which contains the following information:

NETWORKING = yes	
HOSTNAME = "Hostname.domain-name"	system's hostname
GATEWAY = 192.168.102.1	your system router
GATEWAYDEV = eth0	system gateway interface

In order to have the new hostname become effective, you need to issue the command:

xinetd

7.2 System Domain Name

Every system may be given a hostname, but it is not necessary. On a Unix / Linux system it is optional, but highly recommended.

The domain name is simply the name given to the local network system. It allows one to provide a more humanistic reference that may be utilized in referencing itself.

At the command prompt, one may query the domain name by simply issuing the command:

dnsdomainname

To change the hostname of a system, we issue the command:

dnsdomainname newname

This does not permanently write the domain name to any particular file. This must be done manually.

The information is maintained in the **/etc/resolv.conf** file.

domain ourlab.com

In addition to the domain name, we also maintain the address of the DNS nameserver. In addition we may also specify what domain is to be searched. More than one nameserver may be specified, but only three will be recognized. If a request to the first nameserver fails, then the second and lastly the third will be tried. The **/etc/resolv.conf** file typically contains:

search	**isp.com**	your isp
domain	**ourlab.com**	your domain name
nameserver	**209.139.34.2**	your dns server IP Address

The command **domainname** is used to display the **Network Information Service (NIS)** domain name. This name may or may not be the same as the DNS domain name.

7.3 Alias IP Address

It is often convenient for a Network Administrator to have multiple addresses on his system so that changing the IP address is not required. This command may only be issued by the administrator.

To add an additional address to a host, we issue the command:

ifconfig interface:X IP-Address

Where the interface is typically **eth0** and **X** is a value starting at 0 or greater. It is not necessary that the first secondary address be “zero”, but it is customary.

You may create as many secondary addresses on your system as you feel are necessary. These changes are only effective for this session, they will be deleted when the system is powered down or rebooted.

For example, to configure the Ethernet NIC for the IP address:

ifconfig eth0:0 199.100.100.1

Note that the second address does not have to be on the same network as the original address. By default, the system will assume a **classful** address, in this example a Class C address. If a **classless** address is to be specified, then the subnet mask must follow the host address. Two different addresses on the same network should not be assigned, as this may cause a confusion to the networking routing – assign an address that is on a different network.

As another example, is one were to enter the following:

 # ifconfig eth0:1 192.168.103.254

Our NIC will now have three addresses, 192.168.102.{Host-Number}, 199.100.100.1 and 192.168.102.254. We are then able to display the addresses with the command:

```
#ifconfig
eth0      Link encap:Ethernet  HWaddr 00:A0:0C:C8:49:9A
          inet addr:192.168.102.149  Bcast:192.168.102.255
Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

```
        RX packets:1132 errors:0 dropped:0 overruns:0 frame:0
        TX packets:850 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:100
        RX bytes:712285 (695.5 Kb)  TX bytes:134809 (131.6 Kb)
        Interrupt:10 Base address:0x8000
eth0:0    Link encap:Ethernet  HWaddr 00:A0:0C:C8:49:9A
        inet addr:192.168.100.254  Bcast:192.168.100.255
Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:1132 errors:0 dropped:0 overruns:0 frame:0
        TX packets:850 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:100
        RX bytes:712285 (695.5 Kb)  TX bytes:134809 (131.6 Kb)
        Interrupt:10 Base address:0x8000

eth0:1    Link encap:Ethernet  HWaddr 00:A0:0C:C8:49:9A
        inet addr:192.168.103.254  Bcast:192.168.103.255
Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:1132 errors:0 dropped:0 overruns:0 frame:0
        TX packets:850 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:100
        RX bytes:712285 (695.5 Kb)  TX bytes:134809 (131.6 Kb)
        Interrupt:10 Base address:0x8000

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:76974 errors:0 dropped:0 overruns:0 frame:0
        TX packets:76974 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:5258802 (5.0 Mb)  TX bytes:5258802 (5.0 Mb)
```

7.4 Monitoring Network Performance

There are a number of utilities available for monitoring the network. A few will be discussed, but an in depth review is not provided as not all have been used. Most of the following still require testing to insure proper operation. In the past, commands that have been learned of have either proved to be inoperative or were not able to be located on the Internet. Additionally, after testing the following applications, more detail will be provided regarding the benefits, including screen shots.

There are several sites that track many of the latest security issues regarding the Internet. The premiere site is **astalavista.org**, which allows one to have limited access for information, but is subscription based. Lifetime membership is only $100, a great deal for the information that is available. It is excellent to be a member to keep track of the latest security problems. Another site is

www.ciac.org/ciac/ToolsUnixNetSec.html, which supports security for the Department of Energy. It is recommended that one maintain a continual knowledge of the latest security issues.

7.4.1 ethtool

ethtool provides the operational status of the specified network interface. Displayed are the capabilities and the present operational configuration. To display the present operational settings, issue the command:

```
# ethtool   eth0
Settings for eth0:
        Supported ports: [ TP MII ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
        Advertised auto-negotiation: Yes
        Speed: 100Mb/s
        Duplex: Full
        Port: MII
        PHYAD: 1
        Transceiver: internal
        Auto-negotiation: on
        Supports Wake-on: pumbg
        Wake-on: d
        Current message level: 0x00000001 (1)
        Link detected: yes
```

About halfway through the display, we observe the present operational speed of 100 Mbps for our specific interface, and that it is operating in a full duplex mode. The last line specifies that the link or connection to a hub or switch is active.

7.4.2 MoSSHe

MoSSHe (MOnitoring with SSH Environment) is a simple, lightweight (both in size and system requirements) server monitoring package designed for secure and in-depth monitoring of a handful of typical/critical Internet systems. It supports email alerts out of the box – and whatever you can script.

The status and logs are displayed via a web browser interface. Optimum performance is for a small system of 10 hosts, but should work well with a network of more than 100 hosts. In contrast to many other NMS (Network Management Systems) it is not possible to "overload" a MoSSHe system – the minimum checkup intervals will simply extend with each added system. For reference, for the setup of a system consisting of 29 servers and running 264 checks (vendor example), the time required per pass was 100 seconds. Each system configuration will vary significantly.

Via the web interface you can view the overall status (tactical.py), server status and service history, but you cannot modify anything - which makes it quite safe for even non-administrator multiuser use. Additional information may be obtained from **www.wyae.de/software/mosshe/**.

> display

7.4.3 ARPWATCH

Arpwatch is a utility developed at the Lawrence Berkeley National Laboratory by the Network Research Group. Two tools are available to monitor an Ethernet network activity. A database is maintained of MAC and IP address pairings, and reports certain changes via email to the designated recipient.

The library **libpcap** is required for operation; both arpwatch and libpcap must be under the same parent directory. The source code includes two utilities, **arpwatch** and **arpsnmp**, both of which monitor and report the activities, although arpsnmp interfaces with an snmp client to report the general operation. The appropriate files, **arpwatch..tar.gz** and **libpcap** may be downloaded from **http://ee.lbl.gov/** . (libpcap appears to no longer be downloadable.)

Another version is **Remote Arpwatch**, which is designed to collect ARP tables from remote devices using SNMP and checks them for changes. It is very useful for detecting problems and malicious users in networks with routers that don't support static ARP. This application is available from **freshmeat.net/projects/remarp/**.

A README file is included in the tarball download, read it for additional installation details. The utilities must be compiled for operation. Additional information regarding the use of arpsnmp is available from **ftp://ftp.net.cmu.edu/pub/snmp-dist/cmu-snmp*.tar.Z** for interfacing to a snmp system.

> display

7.4.4 NMAP

NMAP, Network Mapper, provides a mechanism for scanning networks to determine what hosts are on the network and security auditing, what services are operating, and what operating systems are providing those services. Nmap monitors raw IP packets to determine what hosts are on the network, what services (application name and version), what operating systems are running, and what types of filters / firewall exist on the network. It is a very powerful utility for performing diagnostics of a network by being able to probe different hosts.

Features include:

Port Scanning	IP Filter mapping
OS Detection	Firewalls
Version Detection	Routers

NMAP is even recommended by Microsoft, so how much better can it get? More information may be obtained from the site **www.insecure.org/nmap/** .

An example of usage would be to issue the command:

nmap -v 192.168.102.*

This provides a list of open ports on all hosts. This might be good if you are looking for security holes, but is bad if an unauthorized user is attempting to learn where they can gain access to a system. Nmap may be used for security audits, where the network administrator may inventory a network and learn of potential security holes in the system. Additionally, a network discovery may be implemented to determine which hosts are providing which well known services. This results in the creation of an "interesting ports table". Host information, such as reverse DNS names, operating system type, device types, and MAC addresses may also be obtained with the use of the proper option.

Some of the more interesting options are noted, but this list is far from complete. For one to gain the most advantage of **nmap**, practice by the user is required. Some options require the user to be logged in as the administrator.

General Options

-sS This scan is commonly used to determine incomplete session establishment, that is, an incomplete 3-way handshake during the initial session setup.

-sT This is the most basic scan, where a probe is made of open ports on a host.

-sF Stealth FIN

-sX Xmas Tree

-sN Null Scan

These three scanning techniques provide improved clandestine scanning of systems. These options often provide a probe that is not able to be detected by the remote host.

-sP A simple scan of hosts on a network by issuing a ping. A quick and simple method to determine which hosts are operational. For those hosts which block a standard ping, it is capable of also transmitting a TCP ack packet to port 80 (default) or initiating a SYN packet and waiting for a RST or SYN/ACK response.

-sV A scan to determine the version of a specified service, such as ftp, ssh, telnet, or http).

-sU Scans UDP ports that are open. Often considered unnecessary, but can be used to discover UDP ports that are open when the TCP port has been closed.

-sO Scans for IP protocol ports to determine which protocols are supported on the host.

-sI zombie-host-IP:[probeport]

An advanced stealth scan to perform a blind TCP port scan of the target. An Intrusion Detection System will detect the probe, but will identify it as originating from the "zombie-host-IP" address, which does not exist.

-sA An advanced scan to map out firewall rulesets on a host.

-PO Do not ping hosts prior to scanning them, thus allowing the scanning of networks that do not allow ICMP echo requests.
-PE Performs a true ping (ICMP echo request) packet to locate operational hosts.
-PP Performs an ICMP timestamp request (type 13) packet for listening hosts.
-O Identifies the operating system of a host.
-A Enables 'Additional' / 'Advanced' / 'Aggressive' options of -O and -sV
-v Verbose mode. Provides additional detail in the generated report.
-h Quick help display of command options.
-oN logfile
Logs the output of the scan to the file “logfile” in human readable format.
-p <port-range>
Specifies ports that are to be scanned. Format of the argument is:
-p U:X, Y, Z, T:M-O, T. In this example, UDP ports X, Y, and Z would be scanned, and TCP ports M through O, and T would be scanned.
--data_length <value>
By default, Nmap transmits minimal length packets. This options specifies the number of (random) bytes that appended to the normal data length. This can be used to test router Access Control Lists or host IP Tables.

TARGET SPECIFICATION

All options must be specified explicitly, all information following the options is considered an IP address (range). This can range from a single IP address to a range of addresses, including a wildcard. Examples might be:

192.168.1.1	Single host
192.168.1.1-50	Scan hosts .1 - .50
192.168.1.*	Scan hosts .0 - .255
192.168.*.*	Scan hosts .0-255.0.255

The target may also be specified by a Fully Qualified Domain Name.

mail.ourlab.com	Scans the DNS resolvable host

In a final note, one must make note of what hosts are being scanned. Scanning of remote systems should not be illegal (talk to your lawyer), but it can be considered an attack and thus subject one to intrusion detection and considered an attempted breach of a remote system.

7.4.5 Nagios

Nagios provides a host and service monitoring of the network, allowing an administrator to detect problems prior to the client. A monitoring daemon

operates on a host, which returns a notification to specified contacts when a problem occurs.

Features include the monitoring of:

Network Services	Environmental Factors
Host resources	Monitoring of unreachable hosts

When checking out the site, the latest version was for Fedora Core 3. Review the home site for additional features and downloading – **www.nagios.org/** .

> display

7.4.6 NLANR

NLANR, Network Performance Advisor, is a framework that provides a single application that integrates the measurement, analysis, and display of various network performance statistics. In includes a suite of tools, including ping, ifconfig, and Iperf that are used to test and gather information, which is later displayed.

The home site is **dast.nlanr.net/projects/advisor/** . There are many additional network tools in the **.../NPMT** directory.

> display

7.4.7 Netstat

Netstat is one of the few utilities that is included with nearly all Linux distributions. It is a very powerful utility that allows the administrator of a host to monitor a system's operation.

A number of different output formats are supported, providing a wide variety of information. A few of the options include:

-A	Display the address of protocol control blocks associated with socket.
-a	Display the state of all sockets.
-d	Display the number of dropped packets.
-g	Display multicast groups.
-h	Display the state of the IMP host table.
-i	Display state of auto-configured interfaces.
-p	Display statistics for the various protocols
-s	Display protocol statistics. This information is basically the same as that provided in an snmp query.
-r	Display the routing table.
-v	Displays information in verbose mode.

By default by issuing the command by itself, **netstat** will display a list of open sockets on the issuing machine. If an address (range) is specified, then those hosts will be queried and the list displayed.

To display the routing table, the command option -r is used.

```
# netstat -r
```

```
Kernel IP routing table
Destination  Gateway  Genmask        Flags  MSS Window  irtt Iface
192.168.1.0  *        255.255.255.0 U        0 0          0 eth0
default      home     0.0.0.0       UG       0 0          0 eth0
```

To specify all network interfaces, use the -i option.

```
# netstat -i
Kernel Interface table
Iface MTU Met  RX-OK RX-ERR RX-DRP RX-OVR  TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0 1500   0   8138      0      0      0   1817      0      0      0 BMRU
lo  16436   0      0      0      0      0      0      0      0      0 LRU
```

To display the multicast memberships for a system, the -g option is issued. Of course, this system is not a member of any multicast at this time.

```
# netstat -g
IPv6/IPv4 Group Memberships
Interface       RefCnt Group
--------------- ------ ---------------------
lo              1      ALL-SYSTEMS.MCAST.NET
eth0            1      ALL-SYSTEMS.MCAST.NET
lo              1      ipv6-allnodes
eth0            1      ff02::1:ffc8:499a%3221213760
eth0            1      ipv6-allnodes
```

Netsstat can display the open connections of a host, the -a option is issued. This list can be long, so in this case the it has been abbreviated.

```
# netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address   Foreign Address   State
tcp        0      0 *:sunrpc        *:*               LISTEN
tcp        0      0 *:x11           *:*               LISTEN
tcp        0      0 localhost:smtp  *:*               LISTEN
tcp        0      0 *:ssh           *:*               LISTEN
tcp        0    132 suse:ssh        longgrain:1168    ESTABLISHED
udp        0      0 *:sunrpc        *:*
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags    Type    State         I-Node Path
unix  2    [ ACC ]  STREAM  LISTENING   2491   /tmp/.X11-unix/X0
unix  2    [ ACC ]  STREAM  LISTENING   1593   /var/run/.resmgr_socket
unix  2    [ ACC ]  STREAM  LISTENING   2287   /var/run/.nscd_socket
unix  7    [ ]      DGRAM               1140   /dev/log
unix  2    [ ACC ]  STREAM  LISTENING   2640   public/cleanup
unix  2    [ ACC ]  STREAM  LISTENING   2647   private/rewrite
. . .
```

In the above connection, it can be observed that to obtain this listing, I have created an established ssh connection from another system (longgrain to suse) via an ssh connection.

Again, this is a powerful command, and it can be used inappropriately on foreign hosts. One may be subject to legal action if used inappropriately.

7.4.8 IPTraf

IPTraf uses the CLI to display various network statistics. Collected information includes TCP/UDP traffic breakdowns, interface statistics, and LAN station packet / byte counts. Features include:

- IP traffic monitor
- Byte counts
- Protocol statistics
- LAN statistics
- Packet counts

Protocols that are recognized include:

- IP
- TCP
- UDP
- ICMP
- IGMP
- IGP
- IGRP
- OSPF
- ARP
- RARP

Obtain more information and the download from **cebu.mozcom.com/riker/iptraf/**

> display

7.4.9 Ntop

Ntop is a network trafic probe, similar to the **top** utility included with the Linux distributions. Where top is a CLI interface, ntop is a GUI interface providing statistics and graphical traffic analysis through a web browser. It is based on **libpcap**, and can be run on various operating systems.

Ntop may be downloaded from **www.ntop.org/ntop.html** .

> display

7.4.10 Nessus

Nessus is a very popular commercial remote vulnerability scanner, able to audit business-critical enterprise systems and applications. It has not been tested due to the cost factor (but that does not count against it).

Features include:

- Vulnerability Database
- Local Security
- Remote Security
- Scalable architecture
- Plug-in Support
- Script Language Support
- Service Port Recognition
- Multiple Service Support
- SSL Support

Additional information and downloading of Nessus may be obtained at **www.nessus.org/** .

7.4.11 AutoScan

AutoScan is designed to explore and manage a network. Subnets can be scanned without administrator intervention. Objective of the utility is to detect all hosts connected to the network and list scanned ports.

Features include:

Multithreaded scanning	Detection of OS
Network discovery	Intrusion Alert
Simultaneous Subnet scanning	Shared Resource Detection
Equipment monitoring	Telnet client
Network Services monitoring	Nessus client

AutoScan may be obtained from **autoscan.free.fr/** . Unfortunately, the documentation (as of 9/06) is only available in French. All of the screen shots show the output in English.

> display

7.4.12 Ngrep

Ngrep is intended to provide the features of the grep utility to the network, applying them to the network layer. It recognizes IPv4, IPv6, TCP, UDP, ICMPv4/6,IGMP, PPP, SLIP, FDDI, and Token Ring.

The Ngrep application may be obtained from **ngrep.sourceforge.net/** .

> display

7.4.13 Cricket

Cricket is an application for monitoring trends in a time-series graphical mode. By using **cron**, data is collected on a periodic basis (typically every five minutes), and then graphed by using the the RRDTool. Display of the graphical information is via a web browser.

Obtain additional information and the application from **cricket.sourceforge.net/** .

> display

7.4.14 SATAN or SAINT

SATAN, or Security Administrator Tool for Analyzing Networks, is a tool for system administrators to verify activities on the network, and report its findings without exploiting them. For each discovered problem, Satan provides a short tutorial explaining the problem and provides information regarding its potential impact. Additionally, the tutorial provides a basic description of how to correct the issue.

In order to fully utilize the strengths of Satan, one must download the utility and review the documentation. It is extremely powerful and must be understood as to its potential and hazards – it can be used for both the benefit of the system administrator and by those that have other desires. This utility probes all systems that it detects, and is not limited to the local network. If one wishes to test their local network, make sure that the Internet connection has been disconnected, as it will also attempt to probe the ISP's router and other server equipment. This will probably result in your ISP disconnecting your service

within a few minutes. Realize, the ISP also has tools to detect when they are being probed running all of the time, and alarms will ring if something is not right.

The Satan utility may be downloaded from **www.porcupine.org/satan/**.

An equivalent commercial version of the Satan is SAINT®, or Security Administrator's Integrated Network Tool. The same functional tools are available with commercial support. Additional information regarding Saint is available at **www.saintcorporation.com/products/vulnerability_scan/saint/saint_scanner.html**.

> display

7.4.15 D-ITG

D-ITG (Distributed Internet Traffic Generator is the opposite of the utilities previously mentioned – it generates random traffic rather than measure it. A tool that is often required to determine if the network will function properly.

D-ITG is written in java, and may therefore be operated on both Unix / Linux and MS Windows systems. Two installations are required, one for transmitting and the second for receiving the packets. The transmitter is configured for the type and duration of transmission, whereas the receiver collect and generates various statistical tables of the received data.

D-ITG supports both IPv4 and IPv6 addressing, supporting TCP, UDP, ICMP, DNS, Telnet, and VoIP protocols at both the network, transport, and application layers. Stochastic processes for both Inter-Departure Time (IDT) and Packet Size (PS), and distribution (exponential, uniform, cauchy, normal, pareto ...) are supported. For optimum performance, one needs to review his / her knowledge in statstics.

Additional information and the software may be downloaded from **www.grid.unina.it/software/ITG/** . The program runs as a java program, thus may also be run on Microsoft Windows.

> display

7.5 Controlling Access

There are two files located in the /etc directory that are used to control access to a system, these are the **hosts.allow** and **hosts.deny** files. Options to the set up of these two files can be either simple or complex, depending upon how the user elects to utilize their powers. When displaying the manual page (**man**) for either, the HOST_ACCESS(5) page will be displayed.

For the above files, hosts.allow and hosts.deny, the following rules are applicable:

1. Each file consists of zero or more lines of text.
2. Lines are processed in the order of appearance, the search terminates when a match is found.
3. A newline character is ignored when it is preceded by a backslash (\ ), this is used to type in lines that are longer than the page / screen width and are thus easier to read and edit.
4. Lines that begin with a “#” character are ignored. The line is considered a comment or remark.

5. The remaining lines are to be formated according to the following format.

   daemon_list : client_list [: shell_command]

   daemon_list is a list of one or more daemon process names. client_list is a list of one or more host names, host addresses, patterns or wildcards. The list of items should be separated by space or comma.
6. A remote daemon or user may be designated by the format of:

   daemon@host or
   user@host
7. Various conditional options are available for matching, allowing options.
   a. An abbreviated URL name may be specified by showing only the ending portion of the URL when it is preceded by a ".". As an example, a URL of ".xyz.com" allows the full URL of "abc.xyz.com".
   b. When specifying an IP network address, the address is terminated by a ".". As an example, a network address of '150.150.' would allow all hosts on the 150.150.0.0 network.
   c. A network may be specified with an IP network address and subnet mask, such as "150.150.0.0/255.255.0.0". A classless address may also be specified, such as "192.168.32.0/255.255.240.0".
   d. A list of hosts may be specified in a file. To specify the file name, preceed the file name with a "/", or with the absolute pathname, beginning with the slash character.
   e. The characters "*" and "?" are wildcards. "*" means multiple characters, whereas "?" specifies one character.

7.5.1 /etc/hosts.deny

By default, a system is set up to deny first, then allow. Access is denied to a daemon / client pair match for an entry in the **/etc/hosts.deny** file. If an entry is not specified in this file, then access is granted to the host.

Example 1:

Deny everyone, unless specified in hosts.allow file:

```
All: All
```

Example 2:

Deny access to one user for a specific service (FTP in this example):

```
ftp : Username@domain.TLD
www : Username@ourlab.com
```

7.5.2 /etc/hosts.allow

Access is granted to a daemon / client pair match for an entry in the **/etc/hosts.allow** file. If the daemon / client pair are not denied due to a match in the /etc/hosts.deny file, then access will be granted to the host.

Example 1:

Allow access to everyone:
All : All
Example 2:
Allow access to one specific host:
All : Username@ourlab.com

7.6 Access Control Lists (ACL)

For those familiar with routers, specifically Cisco, an Access Control List is created to limit who is allowed to pass through an interface. An Access Control List on Linux may be considered just the opposite, where it provides additional permissions to specific users or groups without them being specified as part of a designated group of an application, file, or directory.

Although an ACL is applied to a file or directory, a partition of a hard drive must be configured to support it.

<Not Complete>

7.7 tcpdump Utility

Any system or host that generates a packet on a segment may be monitored to determine its contents. Using the **tcpdump** utility allows one to monitor the network in a promiscuous mode (displays all other hosts).

Although one is seeing the data, it is in a raw format. It must be manually laid out to understand what each byte / bit means. There are other programs that utilize the tcpdump utility and then present the data in a format that one can understand, such as the ethereal application.

To terminate the program operation, hit the CTRL-C keys.

20:41:50.395382 suse.dearroz.net.ssh > longgrain.street-stream: P 1136584:1136716(132) ack 5201 win 8576 (DF) [tos 0x10]
20:41:50.395574 suse.dearroz.net.ssh > longgrain.street-stream: P 1136716:1136880(164) ack 5201 win 8576 (DF) [tos 0x10]

7.8 etherape Application

The Etherape application provides a graphical display of connectivity between various stations on a network. It does not provide any significant statistics, other than representing the volume of traffic by the width of the line, and the type of data, indicated by the color of the line.

It is a nice graphical display that provides a visualization of the traffic, but does not provide much additional information. It does show connectivity to sites outside of the local network. It is a fun utility to monitor that provides some basic information.

In order to run etherape, it must be downloaded from the Internet and installed. It is available as an rpm format. It may be obtained from the website **sourceforge.net/projects/etherape**.

Unfortunately, finding a version that works may sometimes be difficult.

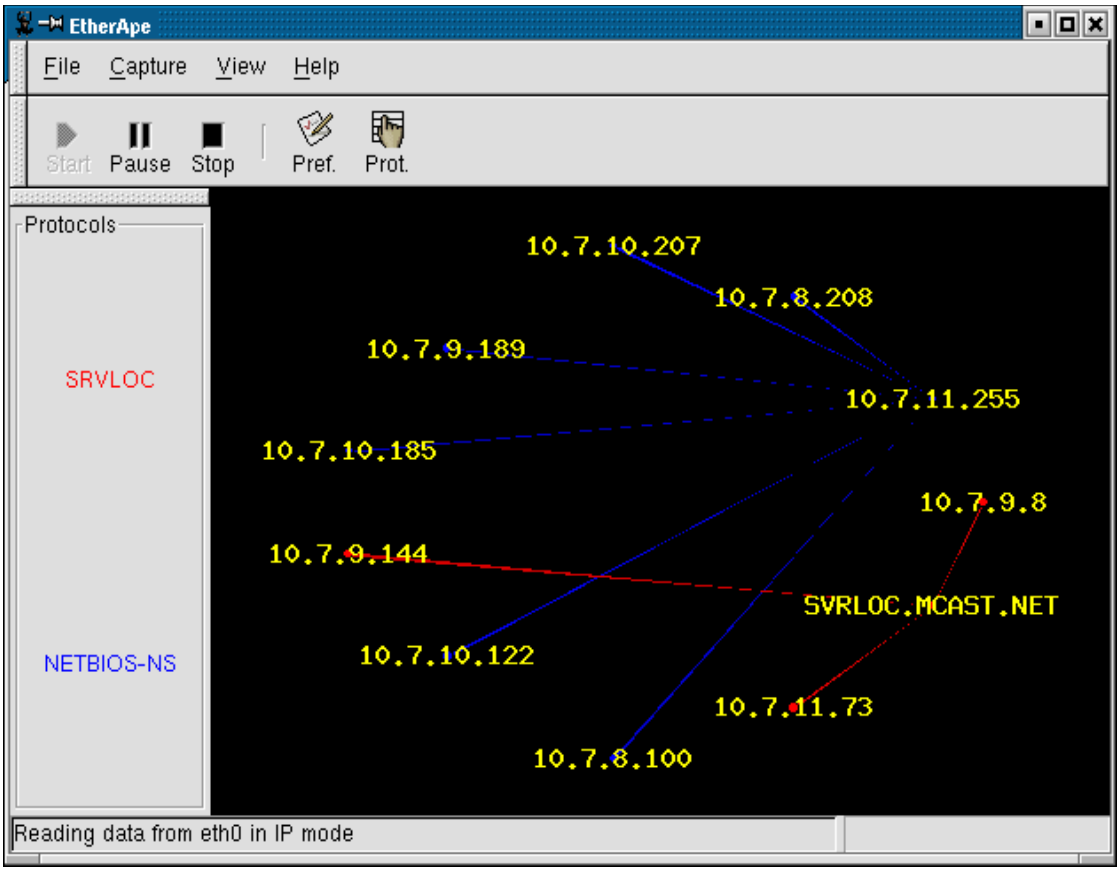

Figure 7.1: Etherape Utility

The downloaded file is in tar gzip format, and requires the gcc compiler. If gcc is not installed on your system, install it using the command:

yum install gcc*

gcc is a very large compiler, and will take some time to download and install, be patient.

7.9 WireShark / ethereal Application

Most distributions include, by default, a utility called **WireShark** or **ethereal**. The application was originally released under the application name of ethereal, but was changed in name in late 2006. This is an excellent network protocol analyzer tool – that is free! Virtually all distributions include it during the installation if desired.

Although it is not as sophisticated as Network General's Sniffer ®, it does provide a very high level of individual packet analysis. For those interested, it is also available in a version to run on MS Windows. The predominate requirement is for the user to understand the structure of the Ethernet Frame in order to be able to specify those which they desire to observe.

In the following, reference is commonly made to Ethereal, but one should now realize that the application is WireShark. Additional detail regarding WireShark may be found at http://www.wireshark.org/ .

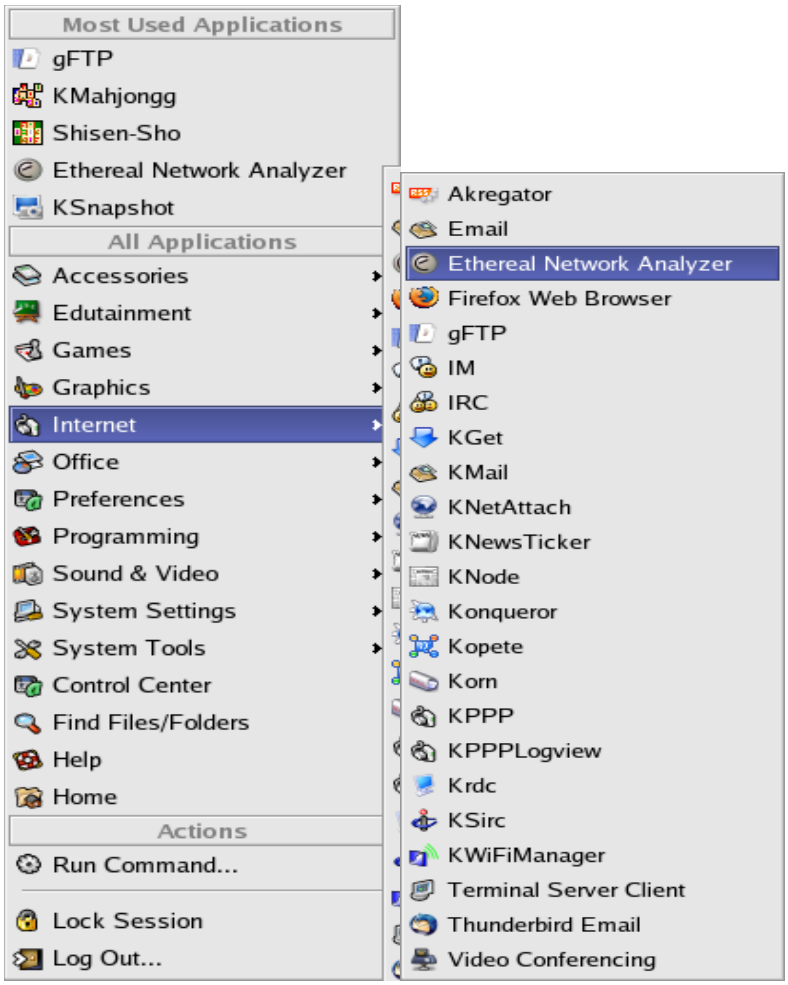

Figure 7.2: Menu to Ethereal Figure 7.3: Selecting Ethereal

The utility may be either entering the command **ethereal** in a terminal window or by clicking on **K – Programs – Internet – ethereal** . This will initialize the program.

To start a capture, select the Capture tab at the top. This will open the drop-down menu to start the process. After selecting Start, the Capture Interface must be selected.

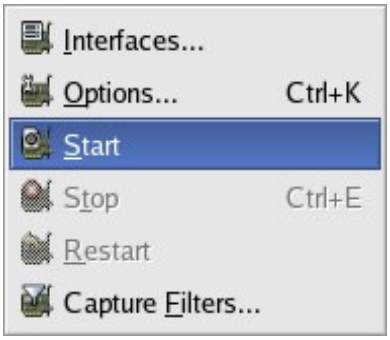

Figure 7.4: Capture Start

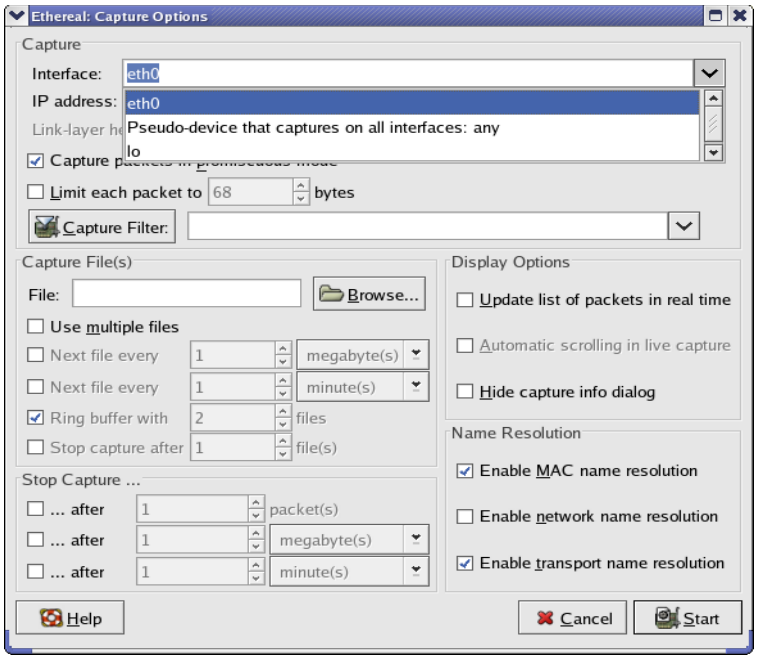

Figure 7.5: Selecting Interface

The Capture Option screen displays the options that are available, one that should be selected is “Update list of packets in real time”.

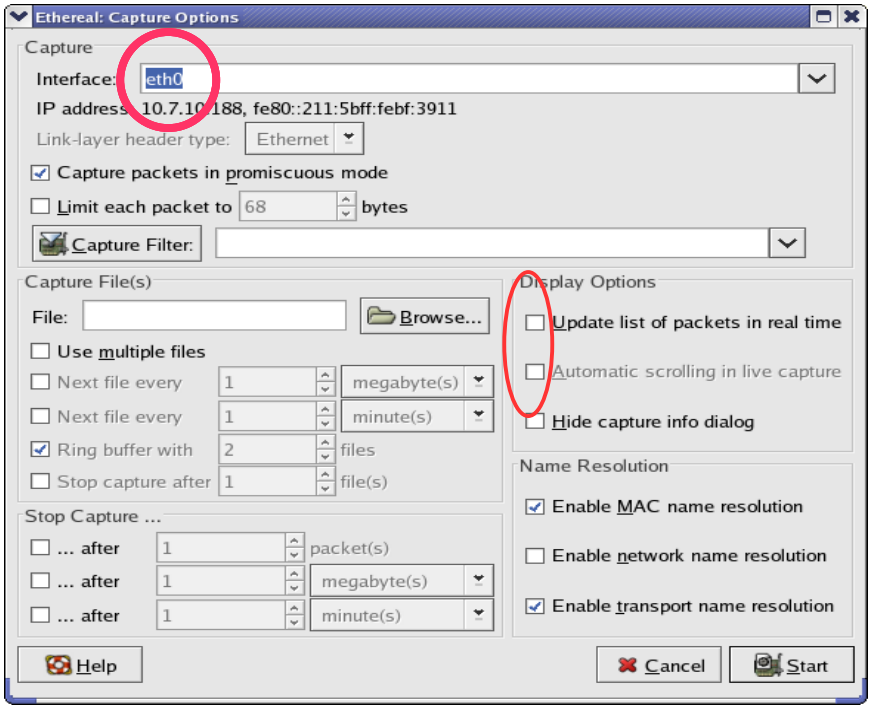

Figure 7.6: Capture Option Screen

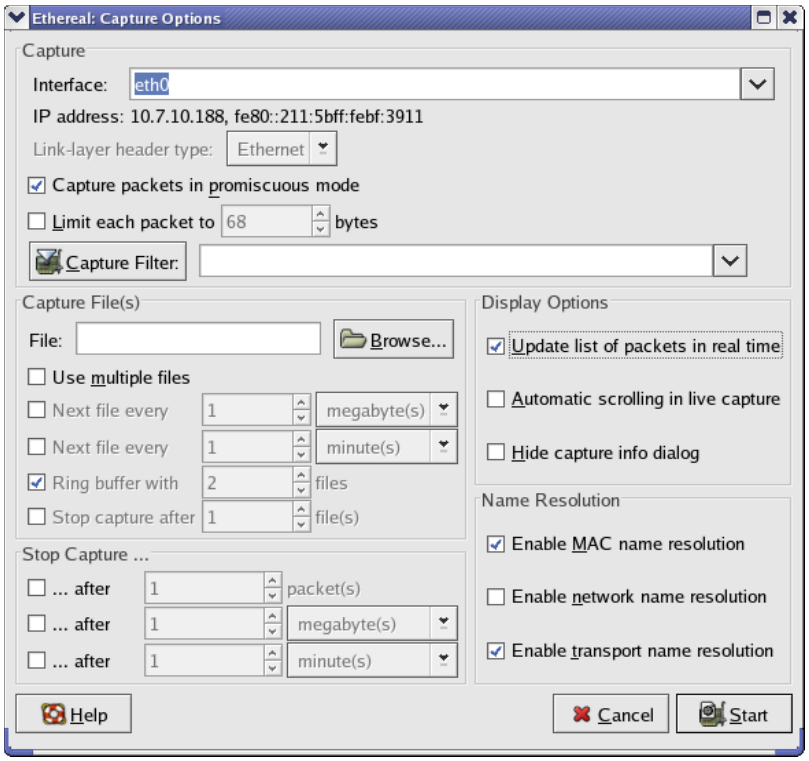

Figure 7.7: Update list of packets

7.9.1 Display

Initially the window may contain two panes, but on activation three panes are displayed. Not only will the program capture and display packets, it will also capture the data to a file. Even better, it will also read a captured file from snoop, atm snoop, LanAlyzer, Network General's Sniffer, iptrace, NetXray, Sniffer pro, Etherpeek, Lucent / Ascend, nettle and several others. (It reads the files "automatically" – you do not have to tell which type of system the data originated from.)

The options are comprehensive, so some detail is provided to demonstrate how to operate this utility. Because this is such a powerful tool when investigating TCP/IP operation, especially for educational purposes, it should be mastered. A simple example of how to set up a filter display is provided, which should be practiced in order to improve the information that is viewed.

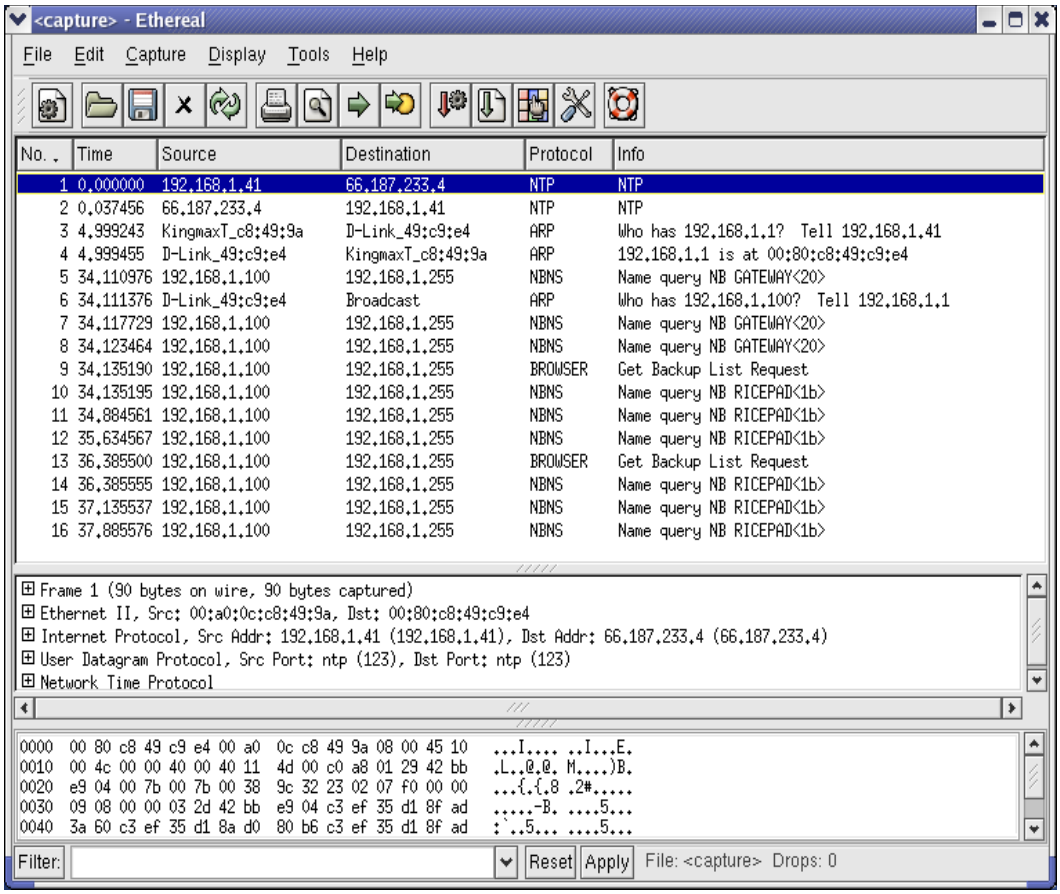

Figure 7.8: Capturing Data

The top pane contains a summary of all of the packets, specifying the sequence number, time, source, destination, protocol, and some information.

The center pain provides a breakout of the various sections of a selected packet that is highlighted from the first pane.

The bottom pain provides detail information of each section of the packet. Typically each section of the packet may be further resolved to display the individual fields. This information is displayed in both hexadecimal and ASCII. This is extremely powerful feature, allowing you to observe the specific data that in transmitted across a Local Area Network. Not only is it a fantastic protocol analyzer, it is a great learning tool.

7.9.2 Display Options

Many options are available for the capturing of data and displaying what the user wants to observe.

7.9.2.1 Data Capture

To start a capture sequence, click on the Capture top menu, then Start. This opens a dropdown window that specifies the interface. Initially, click on the following:

Capture packets in promiscuous mode
Update list of packets in real time
Automatic scrolling in live capture
Enable name resolution

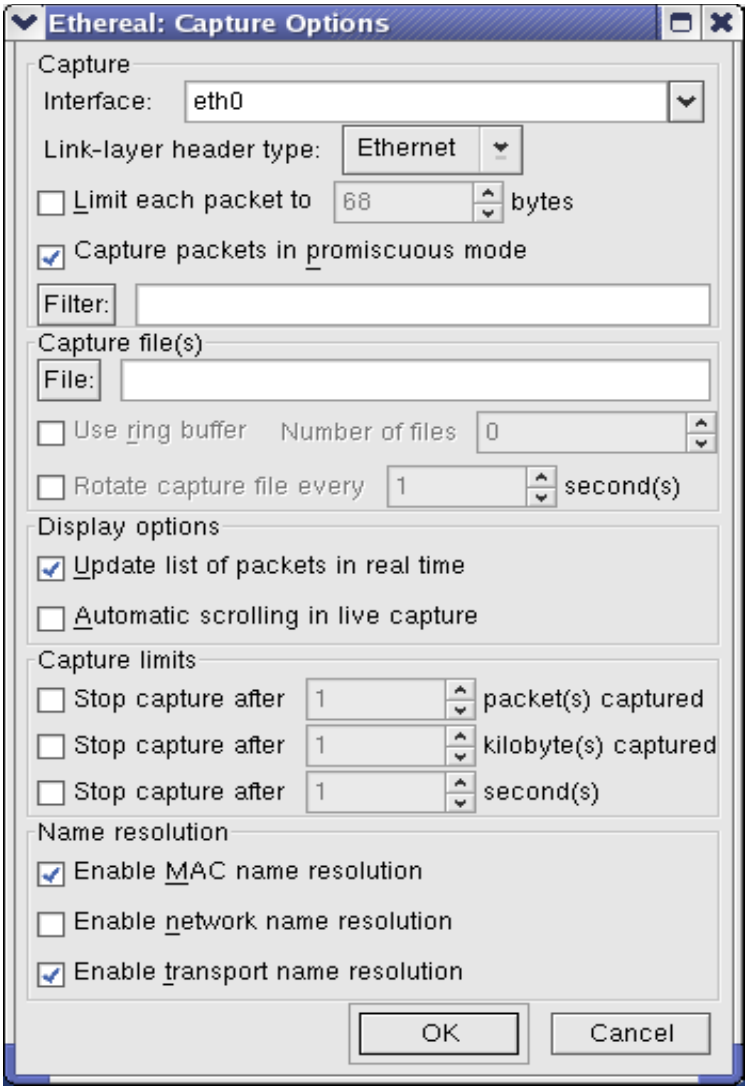

Figure 7.8: Capturing Options

This will capture all packets that transverse the network. To terminate the data collection, click on STOP.

7.9.2.2 Data Filtering

As an additional feature, you are able to filter which packets are captured. Filtering is set up under the EDIT – Display window option.

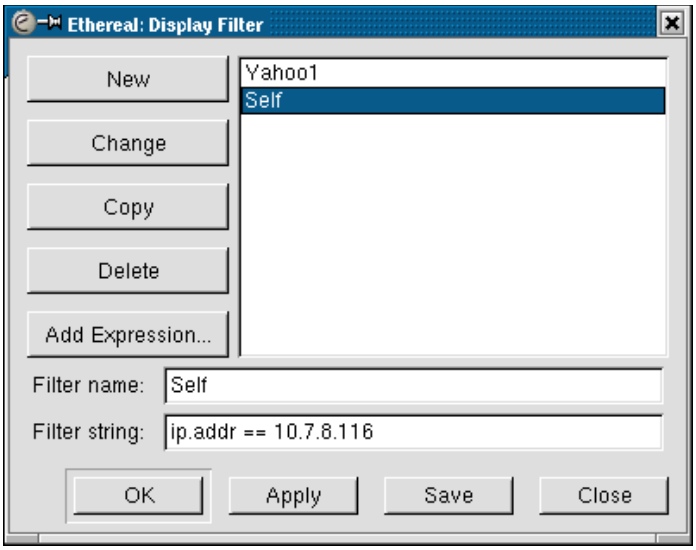

Figure 7.9: Creating a Display Filter

7.9.2.3 Saving Data to a File

Finally, you may also capture the data to a separate file for later analysis.

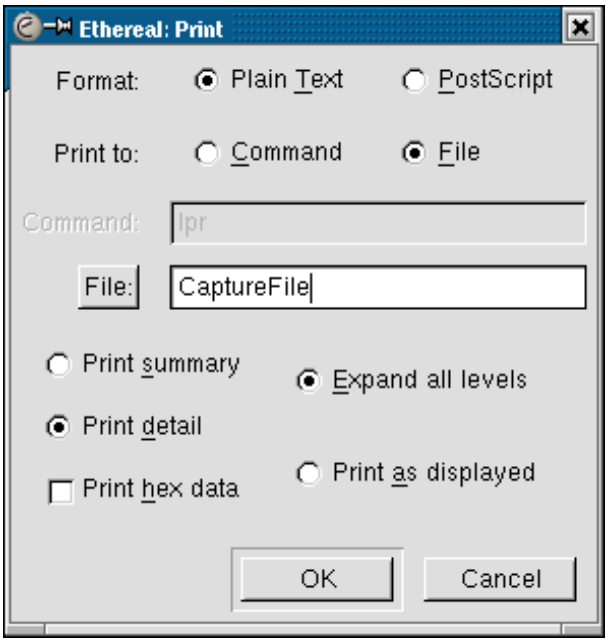

Figure 7.10: Saving Captured Data to File

7.9.2.4 Filtering

A filter may be set up with the following conditions:

Verbal	Math	Means
eq	= =	Equal to
ne	!=	Not equal
gt	>	Greater than
lt	<	Less than

ge	>=	Greater than or Equal to
le	<=	Less than or Equal to

Multiple conditions may be specified using the following:

and	&&	Logical AND
or	||	Logical OR
not	!	Logical NOT

Each field of a protocol may be appropriately specified, and then acted on in one of the following manners:

Unsigned integer
Signed integer
Boolean
Ethernet address
Byte String
IPv4 address
IPv6 address
IPX network number
String of text
Floating point number

An integer may be expressed in decimal, octal, or hexadecimal notation. In the following example, all lines are equivalent:

frame.pkt_len > 10	decimal
frame.pkt_len > 012	octal
frame.pkt_len > 0xa	hexadecimal

You can test a field against a value, resolving the value in Boolean format to being either True (1) or False (0).

IPv4 addresses may be specified by either the IP address or the URL address if address resolution is available. To filter data on addressing, one could use either:

ip.dst eq www.yahoo.com or
ip.src = = 192.168.102.149
both are equivalent in equality format.

To filter on packets that are being exclusively transmitted to or received by a specific station, one could create a filter of:

ip.addr eq 192.168.102.149

The most difficult part of filtering is knowing the name required for each packet field. The following is an abbreviated list:

Field Designator	**Meaning**
arp.dst.hw	Target hardware address
arp.dst.pln	Target Protocol size
arp.dst.proto	Target Protocol address
arp.src.hw	Sender Hardware address
arp.src.pln	Sender Protocol size

arp.src.proto	Sender Protocol address
ah.sequence	Authentication Header Sequence
ah.spi	Authentication Header SPI
cdp.checksum	Cisco Discovery Protocol Checksum
cdp.tlv.len	Cisco Discovery Protocol Length
cdp.tlv.type	Cisco Discovery Protocol Type
cdp.ttl	Cisco Discovery Protocol Time To Live
cgmp.count	Cisco Group Management Protocol Count
cgmp.gda	Cisco Group Management Protocol Group Destination Address
cgmp.type	Cisco Group Management Protocol Type
cgmp.usa	Cisco Group Management Protocol Unicast Source Address
chdlc.address	Cisco HDLC Address
chdlc.protocol	Cisco HDLC Protocol
dns.count.add_rr	DNS Additional RRs
dns.count.answers	DNS Answer RRs
dns.count.queries	DNS Questions
dns.id	DNS Transaction ID
dns.query	DNS Query
dns.response	DNS Response
eth.addr	Ethernet Address (MAC)
eth.dst	Ethernet Destination Address (MAC)
eth.len	Ethernet Length
eth.src	Ethernet Source MAC Address
eth.dst[0:3]	Ethernet destination MAC address – only end evaluated
eth.type	Ethernet Type
fddi.dst	FDDI Destination Address
frame.pkt_len	Frame Packet Total Length
ftp.request	FTP Request
ftp.response	FTP Response
http.request	HTTP Request
http.response	HTTP Response
icmp.code	ICMP Code
icmp.type	ICMP Type
ip.addr	IP Address
ip.dst	IP Destination address
ip.proto	IP Protocol
ip.src	IP Source Address
ip.tos	IP Type of Service
ip.ttl	IP Time to Live
ipx.srcnet	IPX Source network
ipx.srcnode	IPX Source node
smtp.req	Simple Mail Transfer Protocol Request
smtp.rsp	Simple Mail Transfer Protocol Response
snmpv3.flags	SNMP Version 3 Flags
snmpv3.flags.auth	SNMP Authenticated

stp.bridge.hw	Spanning Tree Protocol Bridge Identifier
stp.hello	Spanning Tree Protocol Hello Time
stp.max_age	Spanning Tree Max Age
stp.type	Spanning Tree BPDU type
tcp.ack	TCP Acknowledgement Number
tcp.checksum	TCP Checksum
tcp.dstport	TCP Destination Port
tcp.port	TCP Source or Destination Port
tcp.srcport	TCP Source Port
tr.sr	Token Ring Source Route
tcp.port	Transmission Layer Port address

It is obvious that this is a very long list – and it only represents about 10% of what is available. An excellent listing is in the manual pages for ethereal. It is left to the user to properly implement all of these field options.

7.9.3 A Few Examples

You need to specify what TCP / IP feature you are interested in, and then specify what value(s) are allowed. As an example, one might set up the filter like the following:

ip.dst eq prof
ip.src = = 192.168.102.149

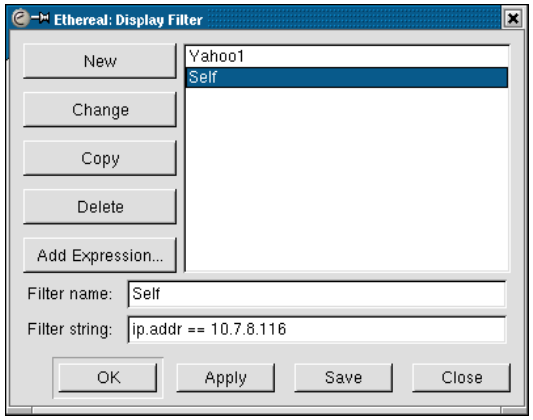

Figure 7.11: Setting the Capture Filter

This would display only packets destined for the prof (assume it is a able to be learned) or from the host 192.168.102.149, but not both.

In example 2, we have specified that packets are to be collected for both the IP destination and the IP source, where the destination is equal to prof (/etc/hosts specified) and the source is equal to the specified address. To specify both of these conditions, we would write:

ip.dst eq prof || ip.src = = 192.168.102.149

Note that we put in the OR condition. This is because we want the packet to be either from the specified station (192.168.102.149), or destined to the specified station. If we had put in an AND (&&), then only those packets that

contained both addresses would have been accepted that were going to the prof from our specified source.

We could also use the expression:

ip.addr eq prof

which would provide for either source or destination address to be equal to prof.

Of course, there is a built in routine to do the above.

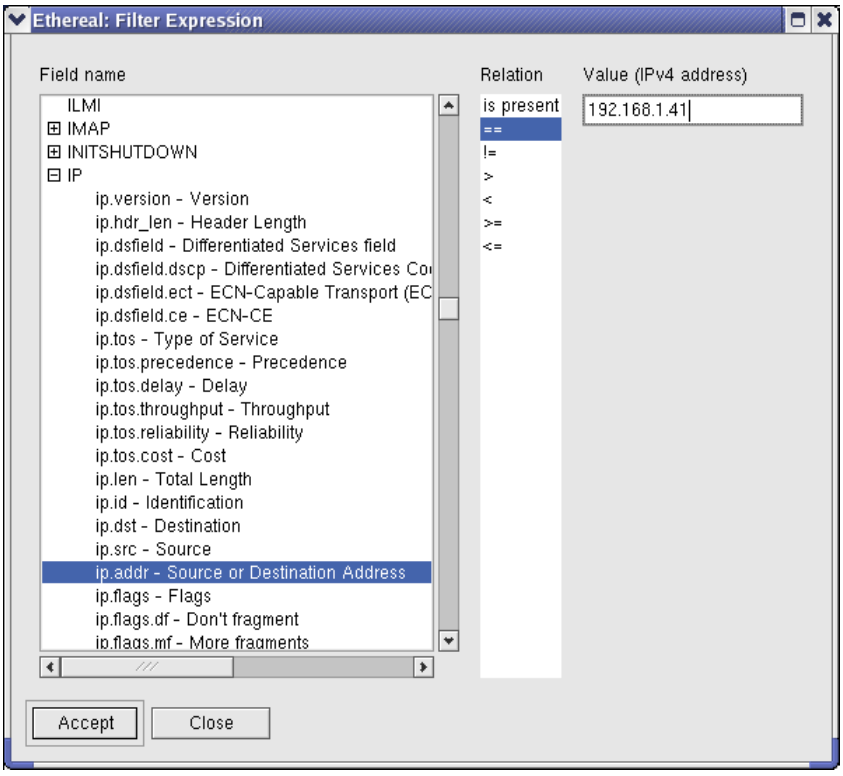

Figure 7.12: Selecting IP Source / Destination Address

If you are interested in only a specified type of packet, you need to determine which field is of interest. You may then test the field for a value, and collect only those that are equal to it – or you could collect all packets that are not equal. Such opportunities.

To set up a filter that is for a specific host and for a specific protocol, we need to set up a filter that creates two conditions that must be met – host ip && protocol.

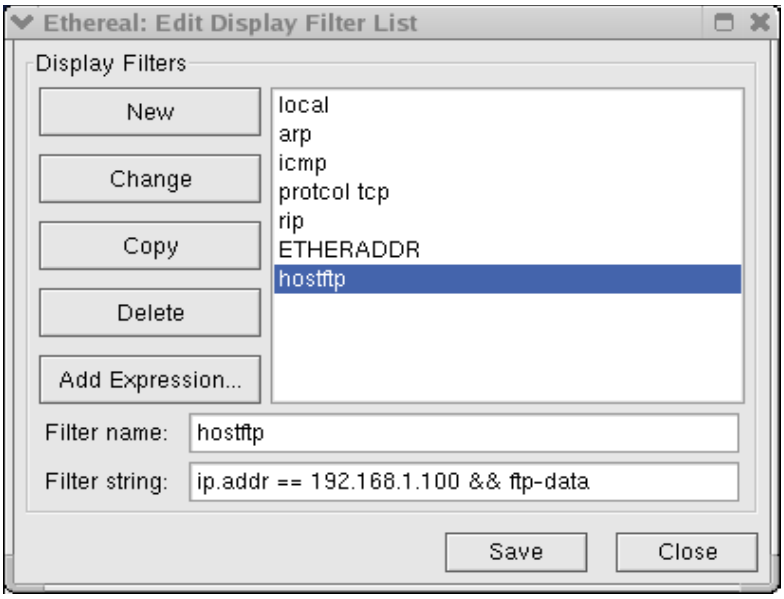

Figure 7.13: Naming the Display Filter

7.9.5 Display Filtering

In order to set up a filter, we need to open the **Edit** drop down menu and select **Display**. This opens a menu that allows one to build and save a new filter specification. This opens an Edit Display Filter List Window. First specify a Filter Name, then click on Add Expression. This opens a new window, Filter Expression. Now you can select your desired requirement by clicking on the value you desire and filling in the blanks (if appropriate). First select the Field Name, then the Relation, and finally the value in the prompt window.

Now start a Capture session and set the Interface (this must be set or you will not see any information). If you wish to observe the whole network, select Capture in Promiscuous Mode. Select Display Options to Update List in real time and to Scroll. This will display the collected data on the screen as you collect it (not necessary, but it makes you feel better). Now click OK to start the capture.

Now that we have collected the data, at the bottom of the screen, we set the Filter. Since you have previously created a filter, click the Filter button and select your desired filter option. To clear the filter, click on Reset. After you have selected the desired filter, click on the Apply button. Your display will now be only those Frames that meet the specified filter condition.

7.10 Setting up Internet Modem Dialup Access (not yet tested)

Not everyone has the luxury of having a full-time Internet access – some must rely on dialup access. In some instances, the use of a modem for connectivity to a remote location is still a requirement. In spite of it “slow” speed, it remains a fundamental of communications.

The predominant protocol used for connectivity to the Internet is PPP, or Point-to Point Protocol. It supports both dedicated and dialup connections. To

permit a PPP connection, the daemon pppd must be installed and made active. This is normally part of TCP stack installation, and only need activation.

7.10.1 Manual Configuration

A system may be manually configured with the following modifications. Manual configuration requires several files to operate.

/etc/ppp/ppp-on This file initializes a connection to your ISP. Information includes the ISP telephone number, username, password, and various ISP options.

/etc/ppp/ppp-off This file terminates a connection to your ISP.

/etc/ppp/ppp-on-dialer This file performs the dialing to the ISP in conjunction with the **chat** program. The script performs error-detection / handling and data rate negotiation.

The various scripts should be in the **/etc/ppp** directory. If not (and typically they are not), then they are in the **/usr/share/doc/ppp-*ver*** directory ("ver" is the version of the ppp daemon that is on the system). To copy the files to the **/etc/ppp** directory, issue the command:

cp –ar /usr/share/doc/ppp*/scripts/ppp-o* /etc/ppp

Then edit the file to enter the information required for your ISP service.

```
# less ppp-on
#!/bin/sh
#
# Script to initiate a ppp connection. This is the first part of the
# pair of scripts. This is not a secure pair of scripts as the codes
# are visible with the 'ps' command. However, it is simple.
#
# These are the parameters. Change as needed.
TELEPHONE=555-1212      # The telephone number for the ISP
ACCOUNT=username        # The account username
PASSWORD=mypassword     # Account password (note that it is in clear text)
LOCAL_IP=0.0.0.0        # Specify if a Static IP, otherwise 0.0.0.0 will mean the computer is to
                                obtain a Dynamic IP
REMOTE_IP=0.0.0.0       # Remote IP address if desired. Normally 0.0.0.0
NETMASK=255.255.255.0   # The proper netmask if needed
#
# Export them so that they will be available at 'ppp-on-dialer' time.
export TELEPHONE ACCOUNT PASSWORD
#
# This is the location of the script which dials the phone and logs
# in. Please use the absolute file name as the $PATH variable is not
# used on the connect option. (To do so on a 'root' account would be
# a security hole so don't ask.)
#
DIALER_SCRIPT=/etc/ppp/ppp-on-dialer
#
# Initiate the connection
#
# I put most of the common options on this command. Please, don't
# forget the 'lock' option or some programs such as mgetty will not
# work. The asyncmap and escape will permit the PPP link to work with
# a telnet or rlogin connection. You are welcome to make any changes
```

```
# as desired. Don't use the 'defaultroute' option if you currently
# have a default route to an ethernet gateway.
#
exec /usr/sbin/pppd debug lock modem crtscts /dev/ttyS0 38400 \
      asyncmap 20A0000 escape FF kdebug 0 $LOCAL_IP:$REMOTE_IP \
      noipdefault netmask $NETMASK defaultroute connect $DIALER_SCRIPT
(END)
```

Note the bold text above, these items need to be changed for the specified username. They include the ISP telephone number, username, and user password.

One change should be made to the last "exec" line (the original is shown above). Note the statement "**/dev/ttyS0**", this should be changed to "**/dev/modem**". Finally, add a link to the **/dev** directory as follows.

ln –s /dev/ttyS0 /dev/modem

This creates a soft link to point to the tty0 (com1) interface. If a different interface is used (com2 – com4), then make the appropriate modification to the link statement.

If it is known that the modem is capable of supporting higher data rates, then the "**38400**" may also be modified, but for a start, leave it at the default.

The ppp-o* scripts need to be executable (typically they are). Issue the following command if they are not already:

chmod +x /etc/ppp/ppp-o*

For our first attempt to make the connection, it would be helpful to be able to monitor the connection process. To do this we can issue the command:

tail –f /var/log/messages

This provides a continuously updated listing of the log messages. After you know that everything is working fine, the above command does not need to be issued. The best way to observe this is to open two XTerm windows under X, one set to display the messages file, the other to initiate the dialup connection.

To make a connection to the ISP, issue the command:

/etc/ppp/ppp-on

Assuming all is working properly, the connection will be made to the ISP, the user will be logged on, and a session may be completed.

After the Internet session has been completed, terminate the connection with the command:

/etc/ppp/ppp-off

7.10.2 GUI Configurations

Two utilities are available for configuring a PPP connection, RP3 under GNOME and KPPP under KDE. Both of these utilities allow connection and a connection status (data rate and time) within their respective toolbar. Both utilities require root level access to configure, so logging in under another username and running either utility will require the root's password to configure.

The KPPP configuration will be demonstrated, but RP3 is similar. Prior to configuration, the following information will be required:

1. ISP access phone number
2. User login name
3. User password
4. ISP Gateway address
5. DNS address(es)

Both utilities provide a simple GUI tool for configuring a dial-up connection, given the administrator has the above information.

Using either of these utilities will produce several files. The following are examples of their contents.

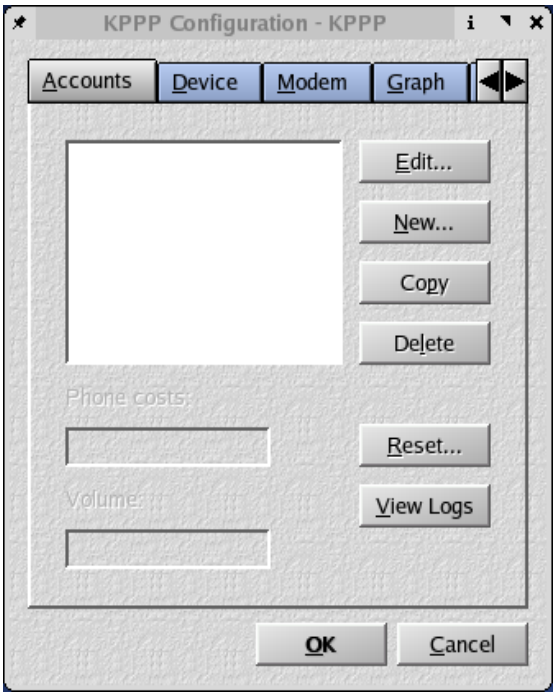

Figure 7.14: Configuring KPPP

Figure 7.15: Creating a New Account

Clicking on New will open the Create New Account window. Do not click on Wizard as it is for other countries – not including the United States. Instead click on “Dialog Setup”.

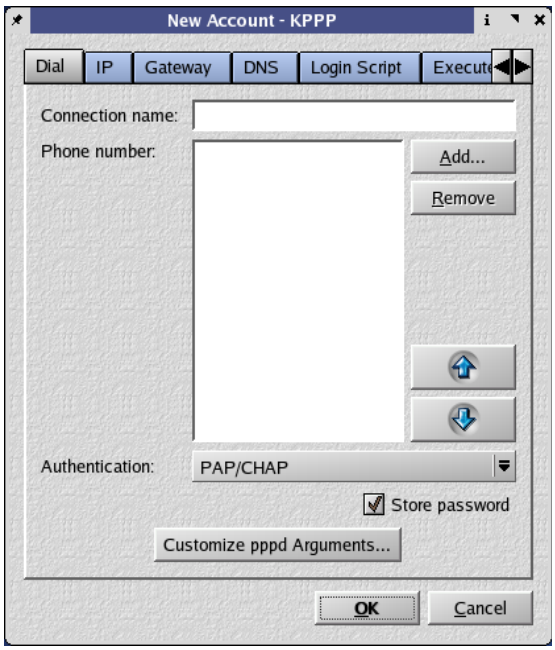

Figure 7.16: New KPPP Account

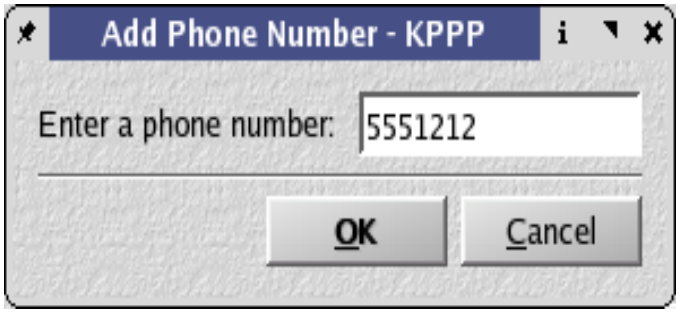

Figure 7.17: Entering a Phone Number

Click on “Add” to open the “Add Phone Number” window. Enter the ISP’s telephone number. Click “OK”.

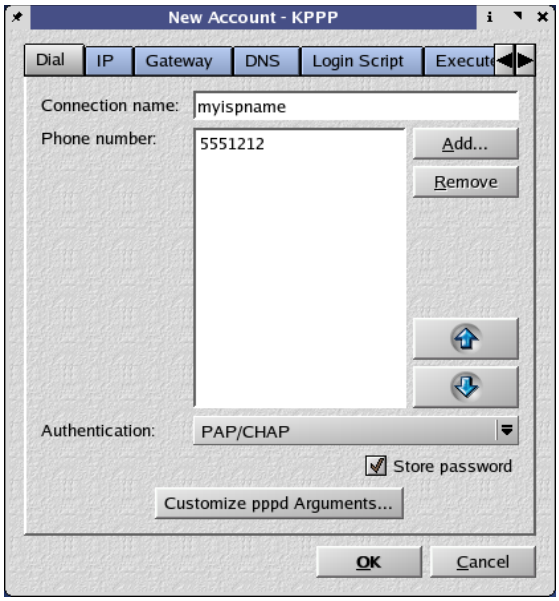

Figure 7.18: Account Entry

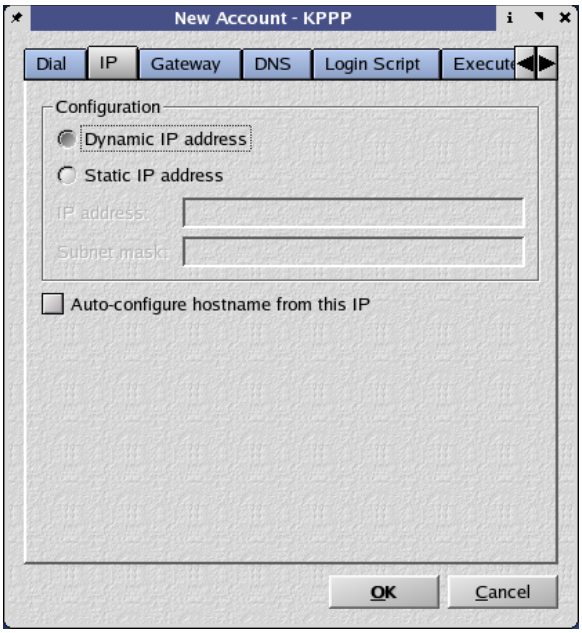

Figure 7.19: Selecting IP Address

Click on the “IP” tab and insure that the system will receive a Dynamic IP address.

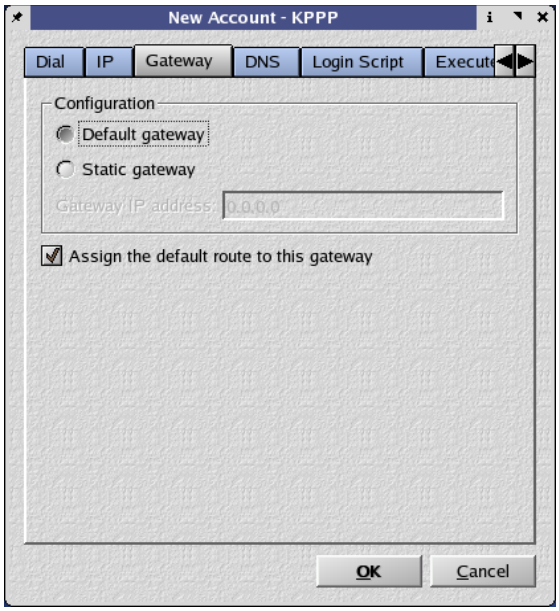

Figure 7.20: Specifying Gateway

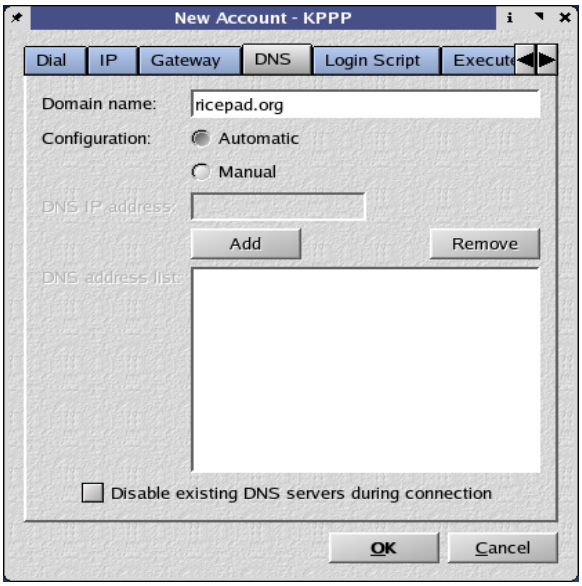

Figure 7.21: Specifying DNS

Click on the “Gateway” tab and insure that the Configuration is set to Default Gateway.

Next click on the “DNS” tab. Set the Domain name to your local system domain name and set the Configuration to Automatic.

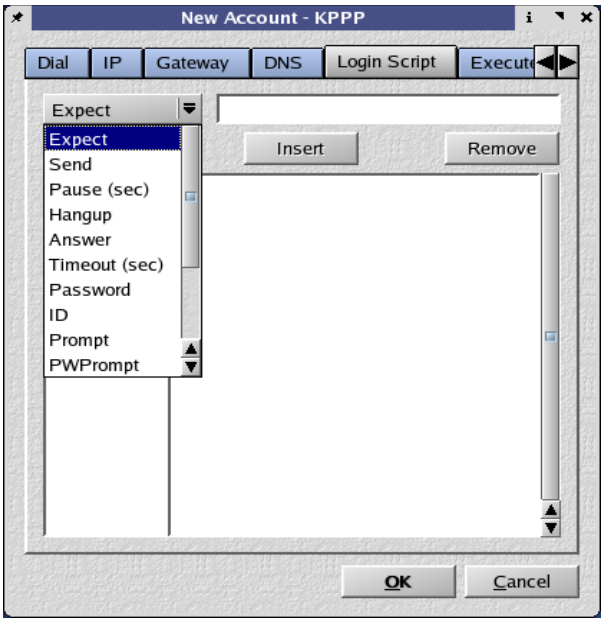

Figure 7.22: Specifying Login Script

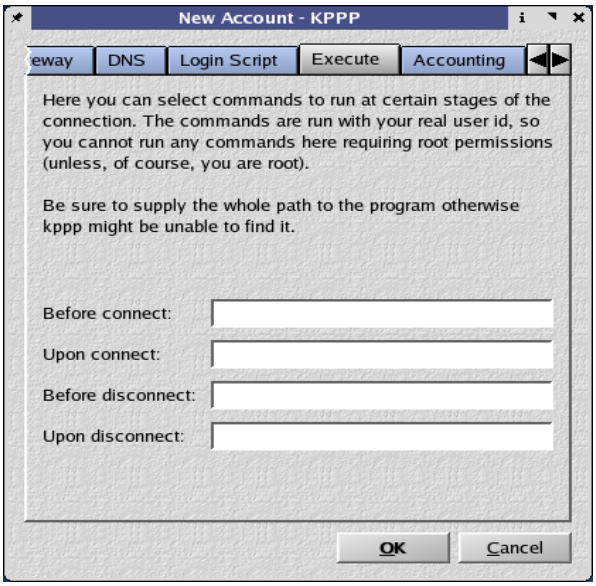

Figure 7.23: Specifying Scripts

The “Login Script” tab allows the setup of special requirements of a login script. No additions should be necessary.

The “Execute” tab provides for additional information that is not normally required.

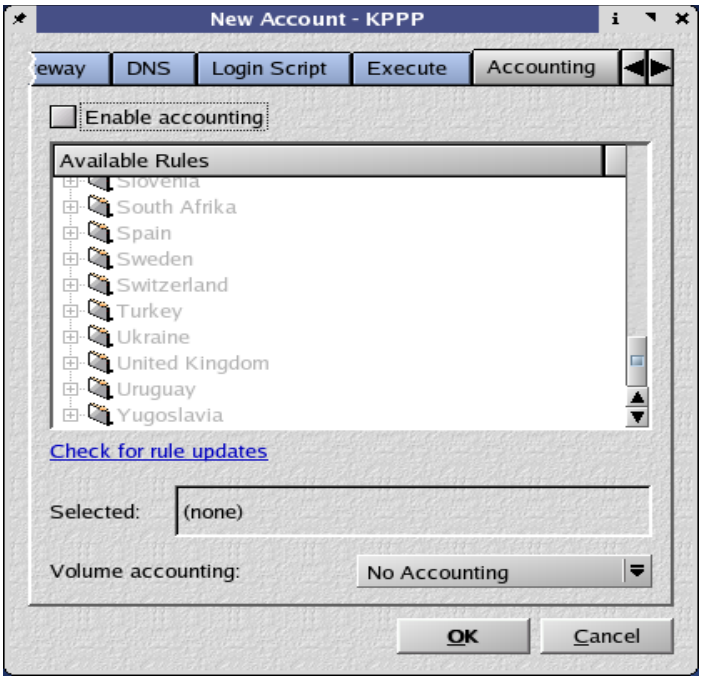

Figure 7.24: Account Dialing Rules

Figure 7.25: Creating Account

Clicking on the “Accounting” tab provides for additional features that are generally not required. Again click on the “Accounts” tab, and you will be at the starting configuration. Click on the “OK” button.

Clicking on the “View Logs” button allows one to observe the past connections. Of course none have been made yet. Click on the “Close” button.

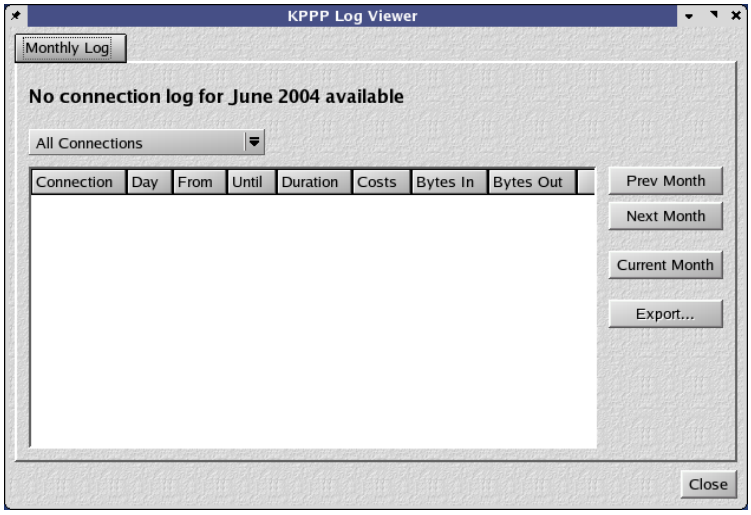

Figure 7.26: Log Viewer

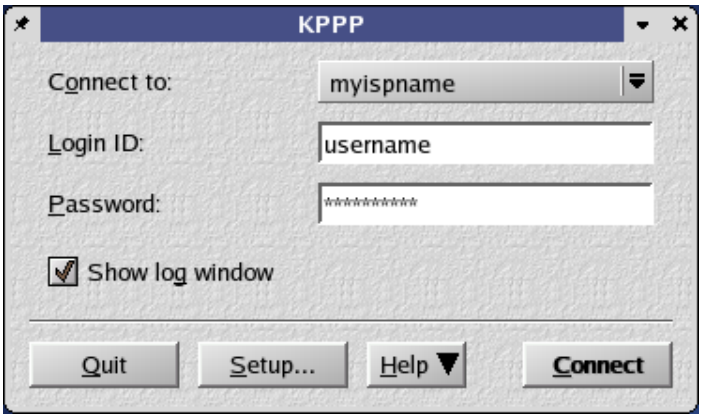

Figure 7.27: Specifying Username to Dial

Insert the username that you wish to log in under and the password that is required. Click the “Connect” button.

Assuming all has been set up correctly, the connection to your ISP will be made and you will be logged in.

7.11 User Quotas

To be Completed

On a multi-user system, it is often important to limit the amount of disk space available to each user. This is important because one does not wish to allow any one user to use all of the disk space and thus limiting the other users to a small amount of disk usage. To limit the amount of disk space on the drive for a given user the administrator implements a user quota system.

Before a quota may be set the system must be configured to accept the quota system. Several actions must be taken.

1. Although not a necessity, it is normally preferable when installing a system to set up a separate partition for the users. Thus all users home directory is limited to the specified partition.
2. The **/etc/fstab** file must be modified for the partition that is to have a quota applied. As an example, if a specific partition for the **/home** directory is to be set up for quotas, the it must be modified with the following:
   >>
3. A file for each user must be created that specifies the user's hard limit, soft limit, and the user grace period. The hard limit specifies a never to be exceeded size, whereas a soft limit warns the user and allows the user to exceed the size for the specified grace period. The files that specify the quotas are maintained in the root (/) directory. The file has the following form:
   >>
4. After the files have been set up, each individual user that is to have a quota implemented, the administrator must issue the following command:

??? Setting up files

edquota –u username

7.12 Commands Used in this Chapter

Arpwatch	Network monitoring utility
AutoScan	Network monitoring utility
chmod	Modify the attributes of a file
cp	Create a copy of another file
Cricket	Network monitoring utility
D-ITG	Internet Traffic Generator
dnsdomainname	Displays the system domain name
domainname	Displays or sets the NIS system domain name
edquota	Utility to enable quota limits to a user
etherape	GUI graphical display of network traffic
ethereal	GUI interface for the tcpdump command for protocol analysis
ethtool	Utility to display the operation configuration of an interface
hostname	Displays or sets the system hostname
ifconfig	Displays or sets an interface IP address
IPTraf	Network monitoring utility
kppp	KDE GUI interface to create an Internet modem configuration
ln	Create a link between two files
MoSShe	Network monitoring utility
nano	Simple text editor that replaces pico
Negios	Network monitoring utility
Nessus	Network monitoring utility

Netstat	Displays network performance
NGrep	Network monitoring utility
NMap	Network monitoring utility
NTop	Network monitoring utility
pico	Simple text editor
ppp-on-dialer	Initiate a ppp modem connection to an ISP
ppp-off	Terminate a ppp modem connection to an ISP
rp3	GNOME GUI interface to create an Interface modem configuration
Saint	Network monitoring utility
Satan	Network monitoring utility
tail	View the last 10 lines of a file
tcpdump	Displays Ethernet packets that are observed on the local network in raw format
xinetd	Re-reads the system server perameters

7.13 Chapter Review Questions

1. In order to limit what packets are observed running ethereal, what must be set up?
   a. list
   b. packet
   c. display filter
   d. scope
2. What is the path / filename where the system hostname and gateway configuration are stored?
   a. /proc/sysconfig/network
   b. /etc/sysconfig/network-scripts/network
   c. /var/network
   d. /etc/sysconfig/network
3. It is necessary to monitor data packets on a network using a GUI graphical display. What application is used?
   a. ethereal
   b. etherape
   c. tcpdump
   d. sniffer
4. You need to monitor the local networks performance. What command is issued?
   a. monitor
   b. tcpdump
   c. etherape
   d. netstat

5. It is desired to have a graphical display showing what hosts are to where and to have a proportional display of the bandwidth. What application is available?
   a. ethereal
   b. etherape
   c. tcpdump
   d. sniffer
6. What command provides a specific listing of open sockets?
   a. socket
   b. soclist
   c. netstat
   d. ethereal
7. What does a hostname and domain name specify?
   a. FQDN
   b. IP Address
   c. MAC Address
   d. URL
8. In which file is the domain name recorded?
   a. /etc/domain.conf
   b. /etc/resolv.conf
   c. /etc/sysconfig/network
   d. /etc/sysconfig/network-scripts/resolv.conf
9. You wish to create an alias IP address on Ethernet interface 0. What command is issued?
   a. ifconfig eth0 192.168.102.249
   b. ifconfig eth0:1 192.168.102.249
   c. ifconfig eth0-1 192.168.102.249
   d. network eth0:1 192.168.102.249
10. When utilizing KDE, which utility would you use to set up a dial-up modem connection to the Internet?
    a. dial-ppp
    b. gppp
    c. kppp
    d. ppp
11. You need to know the operational status of an ethernet interface. What command is issued:
    a. ethtool eth0
    b. etherape eth0
    c. ethereal eth0
    d. etherstat eth0

Chapter Index