SESSION 10 – NETWORK CONFIGURATION AND MANAGEMENT

Course Writer: Dr. Ed. Danso Ansong, Dept of Computer Sc.
Contact Information: edansong@ug.edu.gh

UNIVERSITY OF GHANA
College of Education
School of Continuing and Distance Education

- This session shall focus on network configurations and management.

The key topics to be covered in the session are as follows:

- Sever configuration
- Network Configuration Files,
- Network Monitoring Utilities
- Troubleshooting

- Refer to the following reading material which is available on Sakai

RECOMMENDED TEXT

- Unix And Linux System Administration Handbook, 5th [Chapter 14]

Chapter Objectives

At the end of the session, the student will be able to:

- Analyse configuration files and utilities
- Perform simple network configuration
- Configure and analyse Network Monitoring tools
- Perform some basic Troubleshooting

.

Host Configuration

- **General Network Configuration Procedure**
- To connect a host to the network, the administrator needs the following information.
  - Host name for the system
  - Domain name for the system
  - IP address for the system
  - Netmask for the network (if applicable)
  - Default router (gateway) for the network
  - Name service used on the network
  - Name or address of the machine providing name service
- Once you have this information, you can configure the machine and “plug it in” to the network.

Host Configuration

- Every operating system stores network configuration information in files.
  - Fortunately, some of these files are common across multiple versions of UNIX.
  - Some of these files specify information about the host’s address and host name, or unique setup parameters.
  - Other files specify which network services the host will allow, and which other hosts on the network provide services the host may require.
  - Many operating systems also provide access to the TCP/IP network stack variables in order to allow advanced network configuration.

Host Configuration

- **Common UNIX Configuration Files**
- Network configuration information is stored in several files on the system.
  - Most of the OS installers ask for the previously listed information and configure the system for network operation.
  - Where is all of this information stored?
  - How do you make changes to the network configuration without reloading the operating system?

Host Configuration

- ***/etc/hosts File***
  - One of the most frequently used network administration files is the */etc/hosts* file (*/etc/inet/hosts* on System V machines).
  - Even Windows systems have a version of the *hosts* file, called *lmhosts*.
  - The file is a registry of IP addresses and associated host names known to a system.
  - At a minimum, it must contain the loop-back address (127.0.0.1) and the IP address for the host.
  - The *hosts* file is one of the resources consulted by applications in order to resolve a host name to an IP address when communications are requested.
  - The format of host file entries follows.

```
IP address<Tab>Fully.Qualified.Name<space>[host_alias]*
   192.168.44.55    grumpy.plc.com  grumpy  loghost
```

Host Configuration

- ***/etc/hostname.if_name File***
  - Many versions of UNIX use files in the */etc* directory to aid in the configuration of individual interfaces on the system.
    - For example, Solaris uses files with the generic name */etc/hostname.if_name* to simplify system configuration at boot time.
    - The device name of the network interface is substituted for the *if_name* portion of the file name.
    - For a host with an on-board *hme* Ethernet device, connected to a single network, the */etc/hostname.hme0* file would contain the host name to be used by that interface.
    - Machines connected to multiple networks would have multiple */etc/hostname.if_name* files.
    - Solaris also uses a file called */etc/hostname6.if_name* to configure any Ipv6 interfaces on the system.

Host Configuration

- /etc/nodename File
  - System V machines may also employ another file to maintain an “alias” for the host on the network.
    - The */etc/nodename* file contains the “alias” name from */etc/hosts*.
    - For a multi-homed host, this allows the host to respond to service requests from all connected networks by a single host name.
    - This requires users to remember only one name for the host, no matter which network interface they use to contact the host.

Host Configuration

/etc/services file

- The */etc/services* file contains a list of network ports and services that correspond to those ports.
  - For example, port 25 is defined as the SMTP port, whereas port 80 is reserved as the hypertext transport protocol daemon (*httpd*) port.
  - To add a new service to a host, the administrator must add a port number and service name pair to the */etc/services* file.

Host Configuration

- ***/etc/inetd.conf File***
  - UNIX provides two types of network service daemons:
    - persistent daemons, which are started by run control scripts and are always running, and
    - "part time" services that are only launched when required.
      - These "part-time" services are controlled by the *inet* daemon (*inetd*, or *xinetd* under Linux).
      - The persistent *inetd* daemon reads the *inetd.conf* file when it is started.
      - The *inetd.conf* file tells *inetd* the ports it should listen on. Once *inetd* is running, it monitors incoming network packets for service requests on the ports under its control.
      - When a request is received, *inetd* launches the daemon listed for this port in the *inetd.conf* file.

Host Configuration

- ***/etc/resolv.conf File***
  - Most versions of UNIX use the information in the */etc/resolv.conf* file to configure the name service client on the host.
  - The file consists of keywords and values. Some of the more common keywords follow.
    - *domain*: DNS domain of this host
    - *nameserver* (up to three allowed): IP address of the name server(s) this host should contact. The preferred name server should be listed first.
    - *search*: List of up to seven domains the system should search when trying to resolve an unqualified host name.

Host Configuration

- ***/etc/nsswitch.conf File***
  - The /etc/nsswitch.conf file, also known as the service switch file, is used to tell the system which order it should try to resolve host names.
  - Linux also uses the /etc/host.conf file for this purpose, and BSDI uses the /etc/host.conf file to determine resolution order.
  - These files consist of keywords and values. Some of the more common entries follow.
    - hosts: files dns
    - ipnodes: files dns
    - passwd: files
    - ethers: files
    - netmasks: files
    - bootparams: files
    - services: files
    - aliases: files
    - netgroups: files

Host Configuration

- **Common UNIX Configuration Commands**
  - Different versions of UNIX provide several methods for the administrator to enter system network configuration information.
  - During the installation of the operating system, the installation procedure may prompt you for network information and build the files as required for the installation.
  - Once the system is up and running, other methods and utilities are available for updating network information.

Host Configuration

- **ifconfig**
  - The *ifconfig* command is used to bind the IP address, host name, net mask, broadcast address, and other network configuration parameters to a particular network interface.
  - The *ifconfig* command is run at boot time by the startup scripts called by the *init* process.
  - Unfortunately, every vendor has added its own options to the *ifconfig* command, with the list of options growing daily.
  - This fact makes it almost impossible to tabulate all options available for every operating system environment.
  - You can use the *ifconfig* command to examine and/or modify interface parameters while the system is up and running.
  - When issued with the *–a* flag, *ifconfig* prints the configuration information for all interfaces.
  - Note that IPv4 and IPv6 information is listed separately on dual-stack hosts.
  - Because the implementation details may differ on various versions of UNIX, it is best to consult the *ifconfig* manual page on your system for more information on this command.

Host Configuration

- **route**
  - The *route* command is used to add and manage static routing tables on the host.
    - Static routes are, as the name implies, routes that should not change very often.
    - Many sites configure their systems to send all packets to the site/building/group router nearest the host’s location.
    - The routers then decide how to deliver the data to the final destination.
    - The *route* command allows the administrator to manually add a static route to this “default gateway” router.
    - The generic call to the route command is as follows.

route [-f] keyword [type] destination gateway [metric]

Host Configuration

- The *destination* field contains the IP address of the route's destination.
  - This could be a host address (for example, 172.16.33.44) or a network address (for example, 192.168.10.0), as specified by the *type* field.
- The *gateway* field specifies the address of the router that provides the gateway service for this route.
  - This field may contain the IP address of a host, the IP address of a network, or the string default. The IP address of 0.0.0.0 may also be used in place of the string default.
  - Packets bound for a destination that does not have an explicit route in the host's routing tables use the default route.
- The *metric* field gives the administrator a way to assign precedence to the routes in the routing table.
  - For example, a host with multiple network interfaces may have static routes to multiple gateway routers.
  - If one of these routes included a path with higher bandwidth, the administrator might want to force the traffic to use this route unless the link was down.
  - By assigning a low metric value (typically 1) to the preferred route, and a high metric value (something greater than 1) to the slower link, the system would use the preferred route most of the time.

Host Configuration

WARNING: Check the manual page for the **route** ***command on your system for any nonstandard behavior in that environment. For example, the BSDI kernels do not allow a metric value, but will gladly accept the value at the end of the command as a net mask. This is rarely what the administrator had in mind when configuring the host!***

- OS-specific Network Configuration Information
  - There are no specifications that force UNIX OS vendors to use common file names and content to configure the system.
    - This includes the numerous network configuration files on today’s UNIX hosts.

Host Configuration

- ***Solaris***
  - Solaris provides a plethora of files that are involved in the configuration of networking.
  - In fact, Solaris is probably the most difficult OS to deal with when configuring networking due to the number of configuration files and utilities that seem to be randomly scattered through the file systems.
- **/etc/defaultrouter**
  - This file contains the IP address of the default router (gateway) for this host.
  - A sample *defaultrouter* file might appear as follows.

```
172.16.205.97
```

  - This entry would tell the host that in the absence of a specific routing table entry for the destination host route all packets to the router at 172.16.205.97 for delivery.
  - The default router has also been called the “router of last resort,” as the host will attempt to find a specific route to the destination before it will send the packet to the default router.

Host Configuration

- **/etc/defaultdomain**
  - This file is used to specify the default NIS[+] domain for this host.
  - The NIS domain name may be different from the DNS domain name.
- /etc/netmasks
  - When a network is divided into subnets, the net mask may not be obvious.
  - The /etc/netmasks file associates a subnetwork with the net mask used to create the subnetwork.
  - This allows the system to automatically determine the broadcast address for the subnet.
  - Typical content for a *netmasks* file might resemble the following example.

```
172.16.0.0      255.255.0.0
172.16.70.64    255.255.255.224
172.16.46.32    255.255.255.240
172.16.46.48    255.255.255.240
172.16.46.64    255.255.255.224
```

Host Configuration

- **/etc/networks**
  - The *networks* file is used to associate symbolic network names with Internet protocol addresses.
  - While largely unused, some applications rely on the (sparse) information found in this file.

```
# The networks file associates Internet Protocol (IP)
# network numbers with network names. The format
# of this file is:
#    network-name  network-number nicnames . . .
# The loopback is used for intra-machine communication
loopback    127
#
# Internet networks
#
arpanet     10       arpa  # Historical
```

Host Configuration

- **/etc/init.d/network**
  - The */etc/init.d/network* file is the first of the network startup scripts invoked by *init* at boot time.
  - The services launched by the network script include the multicast routing daemon, and the DHCP client daemon.
  - The script also call *ifconfig* (to configure network interfaces).
  - The interfaces must be properly configured and available before any other network services may be started.

Host Configuration

- **/etc/init.d/inetsvc**
  - The */etc/init.d/inetsvc* file is the next network startup script invoked by *init* at boot time. This script starts many of the Solaris network daemons.
  - The *inetsvc* script starts the NIS daemons to provide NIS, the named binary to provide DNS, and the Dynamic Host Configuration Protocol (DHCP) server daemon.
  - The inetsvc script also configures a multicast route to be used by multicast services, and launches the *inetd* daemon to manage non-persistent service daemons.
  - If you want your hosts to send a log entry for every network connection request to *syslog*, change the last line of the *inetsvc* script to invoke *inetd* with the –*t* flag (in addition to the –*s* flag).
    - This enables the *trace* option of *inetd*. All incoming network connections generate a log entry in the */var/adm/messages* file

```
Feb 12 14:30:14 grumpy inetd[251]:[ID 317013 daemon.notice] ftp[13370] from 172.16.25.96
```

Host Configuration

- **/usr/sbin/ndd**
  - Solaris and HPUX provide the *ndd* command to allow the administrator to get and set driver configuration parameters.
  - For example, if you want to use a host as a packet forwarding system, the kernel must be configured to provide this service.
  - The appropriate kernel configuration may be accomplished via the *ndd* command.
  - In particular, you should use *ndd* to turn on packet forwarding at the IP protocol stack level.
  - To accomplish this task on a Solaris host, you would use the following command.

```
# /usr/sbin/ndd -set /dev/ip ip_forwarding 1
```

  - Conversely, the following command will disable packet forwarding on a Solaris system.

```
# /usr/sbin/ndd -set /dev/ip ip_forwarding 0
```

Host Configuration

- **Hosts with Multiple Interfaces**
- Quite often you will encounter systems with more than one network interface.
- This situation may be required for many reasons.
  - Sometimes the host requires more bandwidth than can be provided by one interface.
  - Other times the system needs to localize traffic to a specific network segment, or the system may need to answer to multiple “names” on the network.
  - These hosts impose some interesting problems on the system administrator.

Host Configuration

- Multi-path Capabilities
  - Systems that contain several network interfaces connected to the same network are called multi-path hosts.
    - These hosts often have very high availability or bandwidth requirements.
    - For this reason, the system architects might install multiple network interface adapters, and connect them to the same media.
    - In the case of a high-bandwidth setup, multiple network interfaces are assigned the same IP address and host name.
  - Under Solaris, this may be accomplished with the *ifconfig adapter group groupname* command.
    - This invocation of *ifconfig* allows the administrator to group interfaces into a virtual "larger" network connection.
    - This group of interfaces acts as a single link, but splits the bandwidth load across many links, thereby providing better overall bandwidth.
    - In the case of high-availability setups, the interfaces may be configured to provide automatic "fail-over," which provides a "hot spare" situation for the network interfaces.
    - This option is configured through the *ifconfig –failover* option.

Host Configuration

- Multi-homed Hosts
  - Hosts with network interfaces on several networks are referred to as *multi-homed hosts*.
  - Such hosts may be configured to act as routers on the internal network.
  - Because these hosts are connected to multiple networks, it may be desirable to have them forward packets from one network to another to facilitate communications.
  - This process is sometimes referred to as *packet forwarding,* or *ip forwarding*.
- ***NOTE:* Packet forwarding will require considerable processor overhead, and is usually best left to routers designed to perform this function.**
  - The administrator can use the *route* command to set the metric for a route, or the *ndd* command to set the metric for an interface.
  - The result is the same: one route is preferred over another due to the weight (metric) assigned to each route/interface.
  - Note that configuration settings made using *ndd* or *route* do not persist across reboots.

Host Configuration

- **/etc/system**
- The */etc/system* file is a dumpster for things that don’t fit elsewhere.
  - This file is read at boot time, and is used to configure everything from network interfaces to virtual memory to executable stack limits.
  - On the networking side, this file is used to configure network interfaces.
  - Placing directives in this file will cause the configuration settings to be applied each time the system reboots.
  - One example of extraordinary network configuration is a network interface configured for “non-auto sensing” operation.
    - By default, Sun network interfaces perform auto-speed and auto-duplex sensing of the network equipment to which they are connected.
    - This situation sometimes leads to deadlocks when a host tries to auto-sense the type of connection it is using.
    - Directives in the */etc/system* file can force a Solaris system to disable the auto-sense capabilities of the interface, and force the interface to operate in a fixed speed/duplex mode.
    - Note that the same results could be obtained using the *ndd* commands (shown as comments in the following example).

Host Configuration

```
; Set interface not to negotiate speed
; same as ndd –set hme:hme_adv_autoneg_cap=0
set hme:hme_adv_autoneg_cap=0
; set interface to 100 megabit full duplex
; same as ndd –set set hme:hme_adv_100fdx_cap=1
set hme:hme_adv_100fdx_cap=1
```

- **/usr/sbin/sys-unconfig**
  - The Solaris */usr/sbin/sys-unconfig* command allows the administrator to change the network information of the host.
  - This command removes all network-specific information files, and halts the machine.
  - When the machine is rebooted, a portion of the SunInstall program is invoked to prompt the administrator for the new network connection information.
  - Once the new information is entered, the system reconfiguration is complete, and the system is ready for use.

Host Configuration

- PPP Under Solaris
  - The Solaris operating system provides a point-to-point protocol daemon called *aspppd*.
  - The *ppp* daemon is used to provide PPP service over dial-up modem links [or PPP over Ethernet (PPPOE), as used by some virtual private networks] and Network Address Translation systems.
- Red Hat Linux
  - The Linux operating system attempts to place most of the configuration files in a common location.
  - In general, under UNIX, the */etc* directory usually contains system startup scripts and configuration files.
  - But many vendors create multiple subdirectories under */etc* to segregate and contain the files/scripts required to configure the system.
  - Whereas most versions of Linux use the BSD conventions for configuration files, RedHat Linux does things a little differently.
  - Under Red Hat Linux, this subdirectory is */etc/sysconfig*.
  - The */etc/sysconfig* directory contains scripts and data files required to configure many system services, including networking.

Host Configuration

- **The linuxconf Command**
  - The */bin/linuxconf* command is the top-level Linux configuration tool.
  - This tool contains links to several second-level configuration tools.
  - One of the tools *linuxconf* can invoke is the *netconfig* command.
- WARNING: The linuxconf command is also available as an inetd service. The linuxconf command has been known to be less than bulletproof, and many administrators disable the ability to launch the linuxconf command via inetd.

Host Configuration

- The netconf Command
- The */bin/netconfig* command, is the primary network configuration GUI under Linux.
- This tool includes menus that allow the administrator to configure IP addresses, net masks, metrics, and other operational parameters of the network interfaces, as well as network protocol stacks.
- This tool often launches other utilities to perform these tasks.
- The Network Script File
- The */etc/sysconfig/network* file contains information that is “generic” to all interfaces on the host.
- The following are a few of the keywords and typical values.

```
NETWORKING=yes               # enable networking on host
FORWARD_IPV4=”no”            # yes if host routes packets
HOSTNAME=”grumpy”            # hostname for this host
DOMAINNAME=”plc.com”         # domain of this host
GATEWAY=”172.16.205.97”      # The IP gateway for this host
GATEWAYDEV=”eth0”            # default link device
```

UNIVERSITY OF GHANA

Host Configuration

- The ifcfg-ifname File
  - The */etc/sysconfig/network-scripts/ifcfg-ifname* files supply the configuration information for each network interface.
  - These script files contain a series of keywords and values parsed at boot time.
  - The values are used to configure network interfaces, net masks, and host names; set default gateways; and perform other tasks required to bring the host up on the network.

```
DEVICE="eth0"                # The interface name
BOOTPROTO="none"             # Set to DHCP to use DHCP
IPADDR="172.16.205.99"# Host’s static IP address
NETWORK=“172.16.205.96”      # The network number
BROADCAST=”172.16.205.127”   # The broadcast address
NETMASK="255.255.255.240"    # The netmask
ONBOOT="yes"                 # yes to configure at boot
```

Host Configuration

- **The ifup Script**
  - The */etc/sysconfig/network-scripts/ifup* script file is invoked at boot time by *init*.
  - The script reads the interface files and parses the entries.
  - The information from the entries is used to configure each network interface with the proper values.
- **The ifdown Script**
  - The */etc/sysconfig/network-scripts/ifdown* script file is invoked at shutdown time by *init*.
  - The script reads the interface files and parses the entries.
  - The information from the entries is used to shut down each network interface.
- **The network-functions Script**
  - The */etc/sysconfig/network-scripts/network-functions* script file contains a “library” of functions used by the *ifup* and *ifdown* scripts.
  - The functions include code for setting the net mask, host name, broadcast addresses, and gateways.
  - The functions perform operations required to configure an interface up or down.

Host Configuration

- **The /proc/sys/net Interface**
  - The */proc/sys/net* directory contains an interface to many internal features of the IP stack on the host.
  - Typically, the files in this directory appear to be empty.
  - Reading the file will give the current value of the variable.
  - The administrator can edit these files, and set a value to tune many operational parameters of the IP stack.
  - For example, by default network interfaces will accept redirect messages from network routers.
  - This allows the router to optimize the network utilization by telling two hosts how to communicate with each other without involving the router.
  - Unfortunately, this also leaves the system open to hijack attacks.
  - The administrator can check to see if the system accepts ICMP REDIRECT packets by using the following.

Host Configuration

- ***HPUX***
  - HPUX achieves something that most of the other UNIX variants come close to: it manages to contain all network configuration parameters within one script.
  - The */etc/rc.config.d/netconf* script is the HPUX network configuration script. This script is invoked by *init* at system boot.
- ***NOTE:* OSF/1 is another UNIX variant that employs a single network configuration script, named */etc/rc.config*.**
- **/etc/rc.config.d/netconf**
  - To configure, or change the configuration of, network interfaces on an HPUX system you must edit the */etc/rc.config.d/network* file.
  - This file contains several variables that control the configuration of the network interface(s).
  - In the following example, the host has two interfaces. Note that any host names used in this script must be defined in */etc/hosts*, as the DNS service will not be running when this script is executed.

Host Configuration

```
NET_CARDS=2
HOSTNAME=bamboozled
INTERFACE_NAME[0]=lan0          # Ethernet link layer encapsulation
ROUTE_DESTINATION[0]=default     # could also be net or host
ROUTE_MASK[0]=255.255.255.240
ROUTE_GATEWAY[0]=172.16.46.33
ROUTE_COUNT=1
IP_ADDRESS[0]=172.16.46.40
SUBNET_MASK[0]=255.255.255.240
INTERFACE_NAME[1]=snap1          # 802.3 link layer encapsulation
ROUTE_GATEWAY[1]=172.16.70.33
IP_ADDRESS[1]=172.16.70.40
SUBNET_MASK[1]=255.255.255.240
GATED=0                     # set to 1 to enable gated
```

Host Configuration

- HPUX Network Utilities
  - In addition to the *rc.config.d/netconf* file, HPUX provides a few utilities that provide information about the system's network links.
  - ***/usr/sbin/lanscan*: shows information about network interfaces. Note that *ifconfig –a* does not work, but you can use *ifconfig* with an interface name**
  - ***/usr/sbin/lanadmin*: The *lanadmin* utility is a menu-based tool that allows the administrator to modify and monitor the network interfaces on the system.**
- ***NOTE:* HPUX will not allow an Ethernet network interface to be configured unless it is connected to the Ethernet media, and a link test is successful.**
- AIX
  - AIX uses a network configuration scheme that is more "Windows-like" than any of the other UNIX variants. Windows provides a GUI tool that allows the administrator to enter the information into a "database" called the registry.
  - AIX stores the basic network configuration information in a database located in */etc/objrepos*.
  - This database cannot be readily edited by hand and must be accessed by the AIX-specific tools *lsdev*, *chdev*, and *mkdev*.
  - A simpler alternative to AIX network configuration management is to use AIX's system management interface tool (SMIT), which acts as a front end to the database tools.

Host Configuration

- ***IRIX***
  - IRIX stores its basic network configuration information in several files in */etc* and */etc/config*. These files can be edited by hand, or via the system manager GUI interface found in the Toolbox menu or by running */usr/sysadm/bin/sysmgr*. The following are some of the more important files.
  - ***/etc/sys_id*: Contains the system's fully qualified domain name. This is used to locate the system's IP address in */etc/hosts* unless the value there is overridden by setting alternative values in */etc/config/netif.options*.**
  - ***/etc/config/ifconfig-1.options*, */etc/config-2.options*, and so on: These files are numbered for each network interface and contain any additional options needed when the corresponding interface is configured by *ifconfig* during system startup. For example, this file might include a custom net mask for a specific network interface.**
  - ***/etc/config/netif.options*: Contains values to override the name or IP address assigned to each network interface on system startup. This might be used to set unique host names for each network interface.**
  - ***/etc/config/proclaim.options*: Contains the settings for the proclaim daemon that will gather an IP address and other configuration data from a DHCP server.**
  - ***/etc/config/static-route.options*: Contains any default routes to be added at system startup.**

Host Configuration

- ***BSDI***
  - Like HPUX and Linux, the BSD kernel relies on variables set in a script file to configure the network interfaces.
  - The system ships with default values set in the file */etc/defaults/rc.conf*. Under BSD kernels, the administrator needs to edit the file */etc/rc.conf* to set values that will override the values in the default file.
  - If you need to set variables specific to a system, you can also create a file named */etc/rc.conf.local* to contain host-specific settings.
  - In this example, the host has two network interfaces (xl0 and xl1). One network interface connects to the 172.16.205.0 network, and the other connects to the 192.168.1.128 network (with a subnet mask of 255.255.255.128).

```
network_interfaces=”xl0 xl1 lo0”
hostname=”poncho”
ifconfig_xl0=”172.16.205.198”
ifconfig_xl1=”192.168.1.188”
defaultrouter=”172.16.205.254”
static_routes=”backbone private”
route_backbone=”-net 172.16.205.0”
route_private=”-net 192.168.1.128 –netmask 255.255.255.128”
```

Host Configuration

- MacOS X
  - While similar to BSD-style UNIX, MacOS X moves the host configuration information to a different file, */etc/hostconfig*.
  - Under MacOS X, the other typical BSD network data files are found in their usual places in */etc*. MacOS X also provides a GUI for managing the configuration information.
  - Note the Location pull-down menu at the top of the network control panel.
  - This allows several network configurations to be saved under different “locations,” and makes network reconfiguration on portable MacOS X systems easier.
  - The padlock icon at the bottom left-hand corner of the control panel allows the system administrator to lock the network settings such that helpful users cannot “fix” the settings for the administrator.
- Windows
  - Under Windows operating systems, many of the network configuration values are contained in the registry, in lieu of text files. This makes changing the values a matter of using the GUI-based tools provided by the OS, or for the really brave, the use of the REGEDIT.EXE utility to edit the registry.
  - **WARNING: Editing the registry is best left to Windows gurus. It is very easy to render the operating system non-bootable by making changes to the Windows registry.**

Host Configuration

- **The LMHOSTS File**
  - The \winnt\system32\drivers\etc\lmhosts file is the Windows equivalent of the UNIX */etc/hosts* file. But as with many things Windows related, the format of the file is a little bit different than its UNIX counterpart. The Windows *lmhosts* file allows options not recognized in the UNIX *hosts* file. The following are a few of these directives.
    - ***#PRE*: Preload this entry**
    - ***#DOM <DOMAIN>*: Windows domain controller for this host**
    - ***#BEGIN_ALTERNATE*: Used to group *include* statements**
    - ***#END_ALTERNATE*: Used to mark end of grouped *include* statements**
    - ***#INCLUDE <filename>*: Include the named file as part of the *hosts* file**
    - ***#MH*: Used to signify multi-homed hosts that have multiple IP addresses**
    - ***#SG*: Special group keyword maps Internet groups by NetBIOS name**
  - The entries in the Windows *lmhosts* file typically do not specify the fully qualified domain name of the computer, as shown in the following example.

```
172.16.205.101      happy     #PRE      #DOM plc
172.16.205.102      sleepy
172.16.205.103      dopey
172.16.205.104      doc
```

Host Configuration

- **Configuration Programs**
  - Windows provides a GUI-based Network Control panel that may be used to configure *hosts* network components. The Network Control panel allows the administrator to configure the interface manually or via the DHCP protocol. If manual configuration is selected, the administrator is presented with several menus of configuration options.
  - If manual configuration is selected, the administrator will need to provide the IP address, net mask value, the IP address of the gateway for this host, the name service to be used, and the name server address. Other options allow the administrator to set a security policy, and enable network filtering based on the policies set by the administrator.
- **ipconfig**
  - The *ipconfig* program displays the current settings used by the network adapter. When invoked with the */all* flag, this command will show more verbose information about the network settings. Information available includes IP addresses, host names, packet counts, subnet masks, gateways, and domain name.

Host Configuration

- **Using DHCP to Assign IP Addresses**
  - A tool that simplifies IP configuration is the Dynamic Host Configuration Protocol (DHCP). DHCP automatically assigns an IP address to a host at boot time. Hosts that use the DHCP protocol to obtain their network configuration information are typically referred to as DHCP clients. Hosts that supply DHCP configuration information to other hosts are referred to as DHCP servers.
- **DHCP Client Configuration**
  - In most cases, the simplest method of configuring a host as a DHCP client is to do so when the operating system installation process is invoked. Most installers contain a section that asks how the system will receive its network configuration information. Simply selecting the DHCP option configures the host to obtain its information from a DHCP server. Because systems tend to have many uses over their lifetime, it is often handy to know which files to edit to change the host's idea of where it will get its network information. The following sections briefly describe the files/commands required to configure a host as a DHCP client.

Host Configuration

- Solaris
  - Configuring a Solaris host to be a DHCP client is relatively simple. The interface needs to be configured under the control of the *dhcpagent* program. The *dhcpagent* program is the utility that manages the network interfaces that use DHCP. From the command line, you can enter the following command.

```
ifconfig interface dhcp
```

  - Here, *interface* is *hme0*, *le0*, *eri0*, or one of the other Solaris-supported network interfaces. This command will place the interface under the control of the *dhcpagent* program. To make the system use DHCP when it boots, you need to modify/create a couple of files, as follows.
  - ***/etc/dhcp.interface*: Existence of this file causes the system to configure that interface for DHCP at boot time.**
  - ***/etc/hostname.interface*: Can still exist, but can be empty if you plan to use DHCP. If the file contains a host name, the system will come up using the static IP address referred to by the name in */etc/hostname.interface*, and then the DHCP service reconfigure the interface using a DHCP address later in the boot process.**
  - If you want to check the status of the *dhcp* client software under Solaris, use the *dhcpinfo* command. Additionally, you can examine the content of the */etc/dhcp/interface.dhc* file to view the settings for a specific interface.

Host Configuration

- ***Linux***
  - You can make a Linux system use DHCP to obtain network information by editing the */etc/sysconfig/network-scripts/ifcfg-interface* file, where the name of the interface is substituted for the word *interface*. Placing the following directive in the *ifcfg-interface* file will cause the system to use DHCP to configure the interface at boot.

```
BOOTPROTO=dhcp
```

- ***AIX***
  - Configuring AIX to use DHCP is best done via SMIT. AIX provides very complete control over the DHCP client and the configuration information it is to receive from the DHCP server
- ***IRIX***
  - IRIX uses a daemon called *proclaim* to communicate with the DHCP server to obtain an IP address and other information. *proclaim* is started during system startup if the *Primary on* or *Interface on* option is present in the */etc/config/proclaim.options* file.

Host Configuration

- ***HPUX***
  - To make a HPUX system use DHCP, you need to edit the */etc/rc.config.d/netconf* file. For each interface that will use DHCP, you need to set the *DHCP_ENABLE[N]* variable to 1, as shown in the following for interface 0 (zero).

```
DHCP_ENABLE[0]=1
```

  - Note that if there is no *DHCP_ENABLE* variable for an interface, the kernel defaults the setting to 1. When DHCP is enabled, the host will receive its IP address, gateway address, net mask, and other network configuration information from the DHCP server. Under HPUX, the */sbin/auto_params* script calls the *dhcpdb2conf* utility to modify the *netconf* file with the values provided by the *dhcp* server. Note that you can also use the SAM utility to configure the system to use DHCP. To monitor the operation of the DHCP client software under HPUX, use the */usr/sbin/dhcptools* utility provided with the system software.

Host Configuration

- ***BSDI***
  - To make a BSDI kernel use DHCP, you need to define a few values, and set a variable for each interface. This is done by editing the */etc/defaults/rc.conf* file and adding the following two directives.

```
dhcp_program=”/sbin/dhclient”
dhcp_flags=””
```

  - Once these values are set, you need to edit the */etc/rc.conf* file and add a line to cause each desired interface to use DHCP. This is illustrated for the xl0 interface in the following example.

```
ifconfig_xl0=”DHCP”
```

  - With these changes in place, the system will start the DHCP *dhclient* program at boot, and the xl0 interface will receive all network configuration information from the DHCP server. The *dhclient.conf* file allows you to customize the *dhclient* program’s operation, but such customizations are usually not required. Lease information is stored in the */etc/dhclient.leases* file.

Host Configuration

- ***MacOS X***
  - Configuring MacOS X for DHCP is a simple matter of selecting DHCP from the list of choices in the Network Control panel.
- ***Windows***
  - Setting a Windows system to be a DHCP client is accomplished via the “Network Control panel” GUI. Select the appropriate network connection, right-click on it, and then select the Internet Protocol (TCP/IP) entry. Click on the Properties button, and then click on the *Obtain an IP address automatically* button.
- **DHCP Server Configuration**
  - Configuring a system to be a DHCP server is more involved than making the system operate as a DHCP client. The server needs to be configured to know what addresses are available, which hosts may check them out, how long an IP address may be leased, and several other parameters. The following sections describe how to configure the DHCP server process under several operating system variants.

Host Configuration

- ***Solaris***
  - To configure the DHCP Server service under Solaris, the administrator should use the *dhcpconfig* command. The *dhcpconfig* command is a front end to the *dhtadm* and *pntadm* utilities that build and maintain DHCP configuration tables. The *dhcpconfig* command allows the administrator to configure DHCP and Boot Relay services (required for remote DHCP), or to unconfigure the DHCP service.
  - The *dhcpconfig* command will guide the user through several questions, collecting the information necessary to set up DHCP on the server. Once the information is collected by *dhcpconfig*, it will invoke the *dhtadm* and *pntadm* commands to create the DHCP database files, and start the DHCP service.
  - If this DHCP server were required to provide DHCP service for remote networks, the administrator would have to set up/configure the BOOTP Relay service. The DHCP registration requests are sent as broadcast packets. By default, routers will not forward broadcast packets to other networks. For the DHCP registration request packets to get to the server, a relay service has to forward them through the router.

Host Configuration

- ***Linux, BSDI, and MacOS X***
  - Most Linux distributions come with the ISC version of DHCP. The BSDI and MacOS operating systems also use the ISC DHCP server. The following discussion covers the ISC DHCP server, not the RedHat pump server.
  - Once the DHCP package has been downloaded, built, and installed, the task of configuring the server begins. For DHCP to work, the system must be configured to provide multicast services, and there must be a route for "host" 255.255.255.255. You can check the output of the *ifconfig* command to determine if the system supports multicast. By default, most Linux/BSDI kernels support multicast operation. To provide a route for the 255.255.255.255 host, enter the following command (assuming your network device is *eth0*).

```
route add -host 255.255.255.255 dev eth0
```

  - If you get the message "255.255.255.255: Unknown host," add the following entry to the */etc/hosts* file.

```
255.255.255.255 all-ones
```

Host Configuration

- If you need to assign specific IP addresses to hosts, you can add multiple fixed address directives to the file. The format of these directives is shown in the following example. The sample code will assign the IP address 172.16.1.199 to any host that advertises that its MAC address is 08:00:20:C0:FF:EE.

```
host zircon {
hardware ethernet 08:00:20:C0:FF:EE;
fixed-address 172.16.1.199;
}
```

- When the *snmpd.conf* file is ready, create a zero-length file named */var/state/dhcp/dhcpd.leases*. Once this file exists, start the DHCP server with the command */usr/sbin/dhcpd interface*. If you leave off the interface directive, the server will start on the *eth0* interface. Remember to add the *dhcpd* server to the appropriate startup file, to ensure that the service starts at system boot.

Host Configuration

- HPUX
- The HPUX DHCP server is named *bootpd*. The first step required to set up a DHCP server under HPUX is to define the range of addresses the server will manage. This is accomplished using the *dhcptools* program, as follows.

```
dhcptools -h fip=172.16.40.50 no=30 sm=255.255.0.0 hn=hpdhcpserver##
```

- This command defines the range of addresses to be used by defining the starting address (*fip=172.16.40.50*), the number of contiguous addresses (*no=30*), the subnet mask, and the host name of the *dhcp* server (*hpdhcpserver*).
- The command will create a file */tmp/dhcphosts*, which can be incorporated into the */etc/hosts* file.

Host Configuration

- Windows
  - Windows server products offer a DHCP server package.
  - The DHCP Service package is not installed by default, and can only be installed on Windows Server-based operating systems.
  - The DHCP snap-in provides a GUI tool for configuration and management of the DHCP service.

Host Configuration

- Network Monitoring and Troubleshooting
  - As with most computer operations, networks are bound to have problems. Troubleshooting networks can be a tedious process.
  - One method of troubleshooting is to use monitoring tools that determine how the network is being used. In some cases, it may not be possible to monitor a network because physical connections may be damaged or gateways may be down.
  - Another method of monitoring the network is to watch the console error messages generated by the machines connected to the network.
- Console Error Messages
  - The error messages sent to the system console provide you with a lot of information about the health of the system and the network.
  - Unfortunately, many administrators do not pay attention to console messages (or worse yet, close the console window, or leave the console window iconized).
  - Fortunately, UNIX provides the administrator with a simple facility for capturing all console messages in a single location.
  - The UNIX kernel uses the *syslog* facility to log messages to the system console. However, *syslog* also allows these messages to be sent to remote hosts.

Host Configuration

- /etc/syslog.conf File
  - The */etc/syslog.conf* file controls the action of the *syslog* facility. The UNIX kernel defines several levels of message severity.
  - Entries in the *syslog.conf* file configure the *syslog* facility to handle these different message categories.
  - For example, a simple informational message may be ignored, whereas a system hardware error may be reported to the console with much fanfare.

Host Configuration

- Logging Messages to a Remote Host
  - In addition to controlling the local disposition of messages, the */etc/syslog.conf* file also allows the administrator to send error messages to a central log host on the network.
  - This procedure enables you to monitor one log file that contains messages from all machines on the network. It also makes life difficult for an intruder trying to cover his footsteps.
  - To enable remote *syslog* capabilities, you should add a line similar to the following to the */etc/syslog.conf* file.

```
*.alert;*.crit;*.err;*.warning;kern.notice;auth.notice @grumpy.plc.com
```

  - Once the previous directive has been added to the */etc/syslog.conf* file, you can restart the *syslogd* process on the client machine by sending a *kill -HUP* signal to the process ID of the *syslogd* process.
  - The *kill -HUP* signal will cause the *syslog* process to reread the configuration file, and restart with the new parameters.
  - Once you have set up all hosts on the network to log errors to a single location, monitoring the error log is as simple as using the *tail -f / var/adm/messages* command on the log host.

Host Configuration

- Network Monitoring
- Although kernel *syslog* messages may provide some information about the health of the network, it is sometimes necessary to more closely examine the network.
- Simple Network Management Protocol (SNMP)
- A more sophisticated method of monitoring networks is to use a network management tool based on the simple network management protocol (SNMP).
- The SNMP package allows a network administration host to constantly monitor the network. Information available to the SNMP software includes network utilization, host error counts, host packet counts, and routing information. SNMP allows you to determine normal usage of the net work and implement alarms to warn of impending problems.
- SNMP operates in a client/server mode. One machine on the network is designated as the SNMP network monitor station. It is configured to poll hosts on the local network in order to collect data into a central repository. The data may be made available to other packages in order to implement alarms, generate graphs of network utilization, and other off-line processing.
- The other hosts on the network are configured as SNMP clients. These hosts run a process that watches for polling requests from the network management station. When such a request is received, the SNMP agent code collects the data from the workstation’s kernel and forwards it to the management station.

Host Configuration

- ***Configuring SNMP***
  - Under most operating systems, the default SNMP configuration is horribly insecure. For this reason, many sites disable the SNMP client code on their systems. If you must run SNMP, there are several things you should do in an attempt to protect the security of your systems.
  - **Change the public and private community strings. By default, the public string is *public*. Private community strings vary by vendor, but are usually easy to find. Access to the private string gives an intruder the capability to reconfigure your hosts/network equipment. Change these strings and protect them as you would the root password. The SNMP community strings are just as dangerous as a compromised root password!**
  - **Block TCP and UDP ports 161, 162, and 1993 at your border routers. There should be no reason for someone outside your organization to access your site using SNMP.**
  - **Keep current with patches. Watch for SNMP security advisories, and make sure your hosts are patched.**
  - **If you generate OpenView or MRTG information for your network, do not allow it to be viewed without proper authentication. The data available from OpenView and/or MRTG packages often gives intruders important information about your network.**
  - With these rules in mind, the following sections briefly discuss how to configure the SNMP client on various operating systems.

Host Configuration

- **SNMP Under Solaris**
  - Under Solaris, the following four files are responsible for the configuration and startup of the SNMP package.
  - • ***/etc/rc2.d/S76snmpdx*: Startup script for the *snmp* daemon.**
  - • ***/etc/rc2.d/S77dmi*: Startup script for the DMI processes that are part of the *snmp* package.**
  - • ***/etc/snmp/conf/snmpdx.acl*: File containing a list of IP addresses allowed to contact the SNMP daemon on this system. The default file contains comments that explain how to add the IP addresses, and community strings for each host allowed to contact the local SNMP daemon.**
  - • ***/etc/snmp/conf/snmpd.conf*: Text configuration file for *snmpd*.**
  - The administrator needs to edit the configuration file and set the read and write community strings to something unique to the local site, as shown in the following example.

Host Configuration

- **SNMP Under Linux**
  - By default, most Linux installers no longer install the SNMP package.
  - To use SNMP, you must install the SNMP utilities from the distribution medium, or from a downloaded RPM.
  - If you do run SNMP, you need to edit the */etc/snmp.conf* file to set the runtime parameters for the SNMP utilities.
- **SNMP Under HPUX**
  - Under HPUX, the following two files are responsible for the configuration and startup of the SNMP package.
  - ***/usr/sbin/snmpd*: Startup script for the *snmp* daemon**
  - ***/etc/SnmpAgent.d/snmpd.conf*: Text configuration file for *snmpd***
  - The *snmpd.conf* file allows the administrator to define several variables that control the actions of the SNMP daemon.

Host Configuration

- **SNMP Under Windows**
  - Under Windows, the SNMP package is not installed by default.
  - To use the SNMP package under Windows the administrator must load the package from the distribution medium.
  - This may be accomplished using the Add/Remove Programs control panel.
  - Click on the Add/Remove Windows Components button, select the Management and Monitoring Tools option, and click on the Details button.
  - Next, select Simple Network Management Protocol, and click on OK.
  - Once the SNMP protocol has been installed, select Control Panel | Administrative Tools | Services GUI to select the SNMP agent.
  - This will cause a pop-up window to appear. The pop-up window contains entries for the configuration and management of the SNMP process

Host Configuration

- Remote MONitor (RMON)
- Extensions to the SNMP software packages have recently permitted more advanced monitoring of network usage. The Remote MONitor (RMON) package allows the administrator to monitor which applications are utilizing the network and which users are running these applications, giving you the ability to perform bandwidth utilization scheduling for the corporate network. RMON also provides you with the capability of monitoring disk space usage, processor utilization, memory usage, and other system monitoring functions.
- Solstice Domain Manager
- Tools such as Sun Microsystem's Solstice Domain Manager package use the data collected by the SNMP software to perform higher-level network monitoring. These packages typically include utilities that have the capacity to "discover" systems connected to a network. The discovered systems are then used to construct an interactive schematic of the network. Alarms (thresholds) on numerous network "health" statistics can be set. If a system on the network fails, an alarm for that condition would be tripped, alerting the network manager of the failure by flashing the icon of the failed system on the schematic.
- Sun's Solstice Domain Manager, shown in figure 16-10, is not bundled with Solaris. It is a separate product marketed by SunConnect, a division of Sun Microsystems.

Host Configuration

- HP OpenView
  - Hewlett-Packard's OpenView product provides capabilities similar to Domain Manager, and is available for a wide range of platforms, including UNIX and Windows.
  - OpenView employs SNMP to gather information about the network. A set of utilities included with OpenView also allows the operator to map the network, and to configure automated polling and trap handling using SNMP.
- Multi-Router Traffic Grapher (MRTG)
  - The Multi-Router Traffic Grapher (MRTG) package is another tool that employs SNMP to collect data about the network. MRTG is configured to contact routers and collect performance/throughput statistics from each interface.
  - This information is then parsed, and stored in a condensed MRTG format. The MRTG data is used to create graphs of input and output traffic for each network interface.
  - MRTG allows for short-interval (typically 15 minutes) daily graphing, and monthly graphs of network activity.
  - MRTG also allows the operator to graph other interesting network events, such as the number of incoming/outgoing messages processed by a mail server, or the number of web site visits seen by a web server.

Host Configuration

- Network Troubleshooting
  - Tracking down network problems is similar to detective work. In time, administrators develop an instinct for where to seek clues. The following discussion is by no means an exhaustive troubleshooting scenario. It will, however, provide some basic tools and techniques that all seasoned network managers use.
- Network Sniffers
  - Network sniffers are software tools that are often used by hackers to surreptitiously collect sensitive information from a network. But system administrators may also use a sniffer as a network-troubleshooting tool. These tools may allow the sysadmin to determine why a network connection is failing, or what application is consuming all of the network bandwidth.
- Vendor-supplied Network Sniffers
  - Most operating systems ship with some form of network sniffer utility. Under Solaris there is the *snoop* utility. Under Windows you will find Network Monitor. HPUX ships with the *nettl* program. Each of these tools is specific to a single operating system, which makes them somewhat cumbersome, as the sysadmin needs to learn several utilities in order to perform one function. In addition to their “proprietary” sniffer applications, most of the UNIX operating systems discussed now ship with a third-party sniffer called *tcpdump*. However, because Solaris and HPUX do not include this application, let’s examine the *snoop* (Solaris) and *nettl* (HPUX) packet sniffers.

Host Configuration

- snoop
  - Solaris ships with Sun's version of a network sniffer utility, *snoop*. The output from *snoop* is generally voluminous and somewhat cryptic. However, it is a good command to use when trying to determine whether two hosts on a network are communicating, or to identify the type of network traffic being generated. Some of the more common uses of *snoop* follow.
    - ***Monitoring traffic between two hosts.*** **For example, *snoop* can be very useful when troubleshooting two systems that cannot communicate. For example, a diskless client might issue *tftpboot* requests that are never answered. You could use *snoop* to capture packets from the diskless client to determine if the client is issuing "proper" requests. Next, you could monitor the boot server network traffic to determine if the server is replying to the boot requests, as follows.**

```
snoop -d hme0 between client_name server_name
```

- - ***Monitoring traffic using a particular protocol.*** **For example, a host on the network may not be able to resolve IP-to-MAC addresses. This typically means that the address resolution protocol (ARP) process is failing. You could use the *snoop* command to watch the network for ARP requests and the ensuing replies from the name server, as follows.**

```
snoop -d hme0 arp
```

Host Configuration

- nettl
  - HPUX ships with a network sniffer utility called *nettl*. Many consider this one of the most annoying and useless packet sniffers available. The *nettl* application is actually very functional, but configuring and learning the user interface of *nettl* are very cumbersome. The *nettl* utility is included in the HPUX Network Tracing and Logging package.
  - Once the *nettl* utility is installed, it is started (by default) at boot time. To disable this boot-time startup, you should edit the */etc/rc.config.d/nettl* file and change the value of the variable *NETTL* to 0. The *nettl* utility reads its configuration from the file */etc/nettlgen.conf*. The *nettlgen.conf* file specifies the default settings for the *nettl* (packet capture) and *netfmt* (packet display) tools.
  - Starting the *nettl* utility seems somewhat cumbersome. The example that follows shows how to start *nettl* to capture the first 1024 bytes of all input and output packets on the loop-back and Ethernet interfaces. This command captures only IP packets, due to the *ns_ls_ip* directive. This example would capture 99999 packets to the file */tmp/raw0*.

Host Configuration

- Public Domain Sniffers
  - Although most vendors ship a network sniffer with their operating system, these tools are often geared toward capturing packets that are known to follow the rules set out for the TCP/IP protocol stack.
  - These vendor-supplied utilities often do not give the user the capability of capturing packets based on very specific criteria.
  - The vendor-supplied tools also present a variety of user interfaces that often disagree on how the menus and data should be presented to the user.
  - Luckily, there are several good public domain sniffer packages available on the net.

Host Configuration

- Ethereal
  - Ethereal, shown in figure 16-14, is a user-friendly UNIX-based program that also runs on Windows. It comes in both a read-only (protocol analyzer) version as well as a capture (sniffing) version.
  - The read-only version is useful for decoding existing packet captures, as it avoids the hassle of installing a packet capture driver on the system.
  - **Data can be captured from a live network connection, or read from a capture file.**
  - **Ethereal can read capture files from an impressive array of other sniffers, and can "uncompress" compressed data from other sniffers on the fly.**
  - **Live data can be read from Ethernet, FDDI, PPP, token-ring, X.25, or Classical IP over ATM interfaces.**
  - **Captured network data can be browsed via a GUI, or via the TTY-mode *tethereal* program.**
  - **Capture files can be programmatically edited or converted via command-line switches to the *editcap* program.**
  - **Output can be saved to disk or printed.**
  - **Data display can be refined using a display filter.**
  - **Display filters can also be used to selectively highlight and color packet summary information.**

Host Configuration

- tcpdump
  - The *tcpdump* utility is a general-purpose packet sniffer that runs on many operating systems. The *tcpdump* utility was originally developed by Van Jacobson, Craig Leres, and Steven McCanne, all of the Lawrence Berkeley National Laboratory, University of California, Berkeley. It is currently maintained as a public domain package, and is available at *http://www.tcpdump.org*
  - Although its user interface is not nearly as friendly as the Ethereal interface, *tcpdump* is one of the more popular network sniffers available. This *tcpdump* utility is also available for Windows, but the Windows version is called *windump*. The *windump/tcpdump* utility allows you to create filters to capture any type of packet you might imagine. The filter language used by *tcpdump* is easy to learn, and is often used in laboratory exercises to teach TCP/IP networking.
  - *tcpdump* does not attempt to make the output easy to read. The output is very concise, and once you get used to it, very easy to read and decipher. For example, to capture one ICMP packet, with the Ethernet frame information included and in hexadecimal output format, you could use the following.

Host Configuration

- ***netstat***
  - The *netstat* program is available under most operating systems.
  - *netstat* has a different set of command line flags under each OS, so it is usually best to consult the OS documentation for the capabilities of *netstat* on your system.
  - In general, *netstat* can display packet counts, and configuration information for your system. Some of the typical uses of *netstat* include display of input/output packet counts per interface, display of TCP/UDP sockets in use on a host, display of the ARP table for a host, and display of the routing tables for a host.

Host Configuration

- ping
  - If a host cannot be contacted on a network, the first question to answer is whether the problem is specific to the host or the network. A very useful command for making this determination is *ping*. The command name stands for “packet Internet groper,” and is used to test reachability of hosts by sending an ICMP echo request and waiting for a reply. The command name is often used as a verb, such as in “Ping pop to see if it’s up.”
  - In the example, the blues machine sent 10 packets (1,024 bytes of data, and an 8-byte ICMP header) to the *pop.plc.com* machine. The pop machine answered eight of those 10 ping packets, as shown by missing packets with sequence numbers of 1 and 7. The network appears to be fairly fast, in that other than the very first packet the average round-trip time for the packets is short.
  - By logging on to several hosts on a network and “pinging” other hosts, you can quickly determine whether the problem is network or host related.

Host Configuration

- Rebooting the noncommunicating hosts will frequently solve the problem.
- If rebooting does not work, examining the files and physical components previously listed would be the next step. In the absence of network diagnostic equipment, replacing components may be the only way to track down the failure. If, on the other hand, no hosts reply to pings, the problem is specific to the network. Some of the more common network faults follow.
  - **Missing or damaged terminators**
  - **Broken or damaged cable (may not be visible)**
  - **Faulty connectors**
  - **Damaged transceiver (may be jamming the network)**
- Once again, in the absence of network diagnostic equipment, replacing components may be the only way to precisely identify the cause of failure. If a host on one subnet is unable to contact a host located on a different subnet, and there is no difficulty pinging hosts on the local subnet, the problem could be one of the following items.
  - **Malfunction of the router connected to the subnet**
  - **Malfunction of the network on the other side of the router**
  - **Malfunction of a distant router not directly connected to the subnet**
  - **Malfunction of the remote host**
  - **Failure of a component or connection between the local host and remote hosts**
- To locate the problem, use the techniques discussed earlier in this section for each subnet in question. With subnetted networks, *ping* is often all that is required to locate (or at least narrow down) the source of a network failure.

Host Configuration

- traceroute
  - The *traceroute* utility does exactly what its name implies: it traces the route from point A to point B. This utility can be very useful when trying to determine why two hosts cannot communicate. For example, to trace the route from a specific local host to a distant host, the administrator could invoke the following command.

traceroute fully_qualified_hostname

  - The *traceroute* command prints an informational message about each "hop" along the route from the local host to the remote host. Routing loops and inoperative hosts are very easy to spot in the *traceroute* output. Note that under Windows this utility is called *tracert.exe*.

Host Configuration

- **Host Configuration**
- Unfortunately, connecting a computer to the network is not the end of the network configuration and management task.
- Once the system is on the network, the sysadmin needs to be vigilant to ensure that the system is secure.

Summary

- This chapter examined configuration files and utilities involved in the management of computer networking.
- Topics discussed include some simple network configuration commands, as well as coverage of several of the network configuration files.
- Examples of these configuration files were presented to illustrate the information required to configure a system for network operation.
- Finally, the chapter explored ways to monitor and troubleshoot local area networks.

- Unix And Linux System Administration Handbook, 5th Edition By Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley, Dan Mackin. Released September 2017. Publisher(s): Addison-Wesley Professional. ISBN: 9780134278308
- Practice Of System And Network Administration, The: Devops And Other Best Practices For Enterprise IT, Volume 1, By Thomas A. Limoncelli, Strata R. Chalup, Christina J. Hogan. Released November 2016. Publisher(s): Addison-Wesley Professional. ISBN: 9780133415087
- Essential System Administration, Third Edition bv Æleen Frisch, Published by O'Reilly Media, Inc. (20(0-596-00343-9