

TCP/IP for Dummies Tutorial

**InterWorks '96
San Diego, CA**

Presented by:

Fred Mallett
FAME Computer Education
334 Haroldson Dr
Corpus Christi, TX 78412
512-991-3044

frederm@aol.com

Overview

- Introduction to TCP/IP Networking
- Network Model
- TCP/IP Terms
- Host Names
- Internet Addresses
- Subnet Addresses
- Internet Addresses to host name mapping
- Quiz# 1
- Routing
- TCP/IP Daemons
- TCP/IP Information files
- TCP/IP Local Information files
- ifconfig
- mkhosts
- route
- Quiz# 2
- TCP installed links
- Internet Services
- Common internet services error messages
- Trouble shooting commands
- ping
- netstat
- Trouble shooting Techniques

Introduction to TCP/IP

What is TCP/IP?

Two separate protocols, independent of each other

TCP

Transmission Control Protocol, defines the protocol for one host (host application) to talk to another. Layer four of the OSI/ISO model. TCP is connection oriented. Uses PAR (positive Ack with re-transmission). Treats data as a stream.

IP

Internet Protocol, defines the protocol that gateways use to identify networks, and paths to networks and hosts. Handles the routing of data between networks. Layer three of the OSI/ISO model. IP is a connectionless protocol Treats all data as datagrams.

UDP

User Datagram Protocol, Treats data as a message, and sends packets. Unreliable, connectionless protocol. Has less overhead than the connections of TCP

Network Model

ISO/OSI Network Model

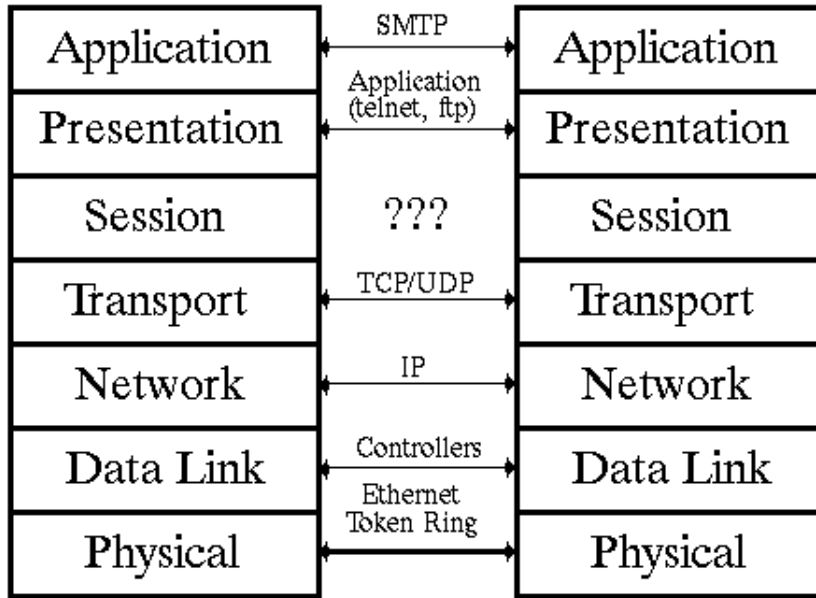
OSI

Open Systems Interconnection Reference Model
First proposed by the:

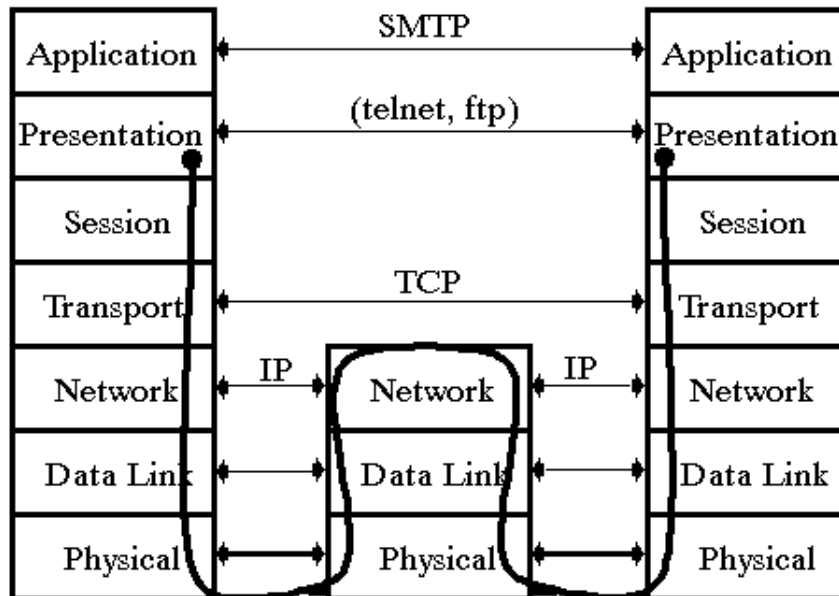
ISO

International Standards Organization

The model is a theoretical model and most network implementations do not follow it exactly, TCP/IP really has less "layers"



TCP Communications

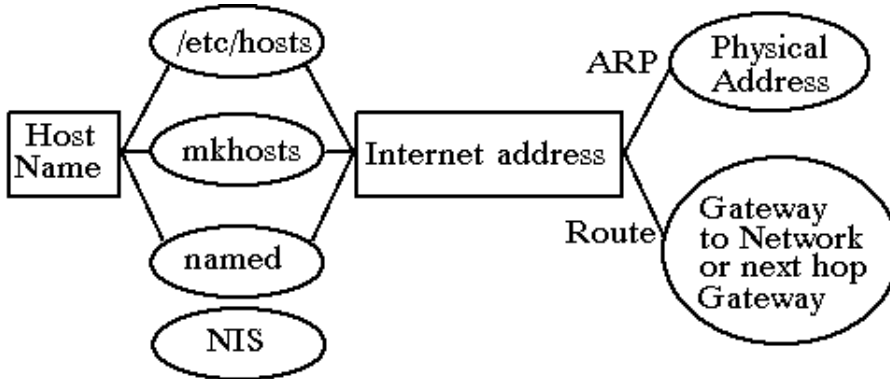


Communication requirements

For a message to be "sent" (no guarantee that it gets there), the following three steps must be

accomplished:

1. Hostname to internet address conversion
2. Local or remote network decision
3. Local - Internet to physical address translation
Remote - Next hop gateway address, then gateway internet address to physical address trans.



TCP/IP Terms

Host

An entity which communicates over a network (i.e., a node)

Ethernet address

A 48 bit address built into the hardware of the ethernet controller by the manufacturer

Host Names

Each host must have a unique name on the network. They can be just names, or use the hierarchical domain naming scheme

Internet address (IP address)

Four numbers in the range 0 - 255 separated by periods

Used to universally identify computers throughout the network, internet, or INTERNET

Router node (gateway)

A node that performs routing services between two networks of similar network protocol, often selectively

Gateway node (router)

A node that perform routing between two networks of dis-similar network protocol types

Host Names

Each host name in a network must be unique

Easy in small networks, more difficult in large networks, impossible in the INTERNET, without some control

Host names:

- can be up to 32 characters (12)
- can contain a-z, A-Z, 0-9, ., -, _
- cannot contain # or spaces
- must begin with an alphabetic character

Domain naming system

Introduced to assure unique naming, and eases the administration of the naming database. Tree structured. Requires the use of named

Usually three levels of domains, can be more

Top level, fixed, describes a category of institution:

- COM Commercial
- EDU Educational
- GOV Government
- MIL Military government
- NET Changeover paths
- ORG Non-profit
- XX Two letter country codes

Second level domains, usually represent a whole organization. Assigned (approved) by SRI-NIC

Third and below, represent organizational departments or subdivisions within an organization

tomvc@os.rd.apollo.com fredm@cbdat.mit.edu

Internet Addresses

An internet address has two fields

Left field, is the network number

Right field, is the host number

An internet address is 32 bits long

Consist of four octets (8-bit bytes), each octet defines either a network or host on that network

The network portion is assigned by NIC, the host portion is assigned locally (if you will never go on the INTERNET, you may pick a network portion)

0 in the network portion is reserved for the default route
 127 network is reserved for local loopback, or local host
 0 in the host portion is reserved for this network

255 in the host portion is reserved for broadcast packets (BSD4.3) 0 was used for broadcast packets under BSD4.2 and older

Field sizes

How many bits are used for network numbers, and how many are used for host numbers depends upon the needs of the network

This also defines the type of address

Type A N.H.H.H

1.0.0.1 to 126.254.254.254 (126 networks, 16M hosts)

Nearly impossible to get assigned a class A address

0	7 bits network	24 bits host
---	----------------	--------------

Type B N.N.H.H

128.1.0.1 to 191.255.254.254 (16K networks, 65K hosts)

Most sites with class B networks use the first host address byte for local subnets

10	14 bits network	16 bits host
----	-----------------	--------------

Type C N.N.N.H

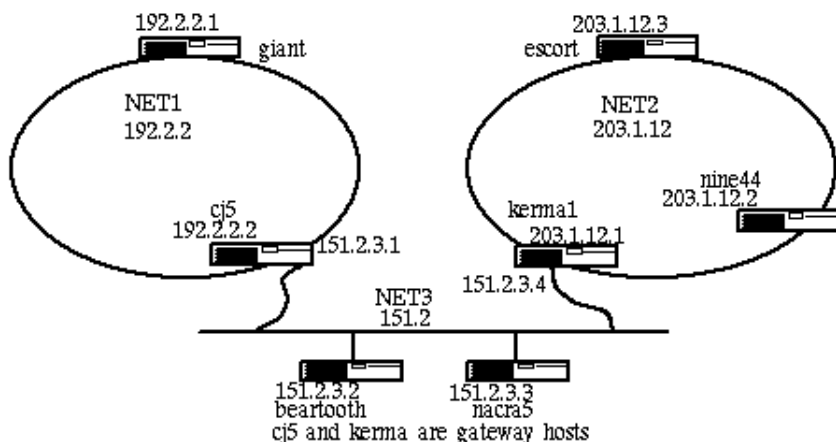
192.1.1.1 to 223.255.255.254 (2M networks, 254 hosts)

110	21 bits network	8 bits host
-----	-----------------	-------------

Type D&E N.N.N.H

224.1.1.1 to 255.255.255.254

Reserved for multicast and experimental purposes



Subnet Addresses

Subnets allow you to present a simple address to the "rest of the world", yet divide your network logically into groups of administrative, physical, or organizational control

Subnets are administered locally

The internet address doesn't change, just how tcp/ip interprets the address

Using a netmask value, you tell tcp/ip to use a portion of the host address as a subnet address instead (the netmask is an argument to the ifconfig command)

A sample type B address subnetted:

128.1.0.1 to 191.254.254.254 (16K networks, 65K hosts)

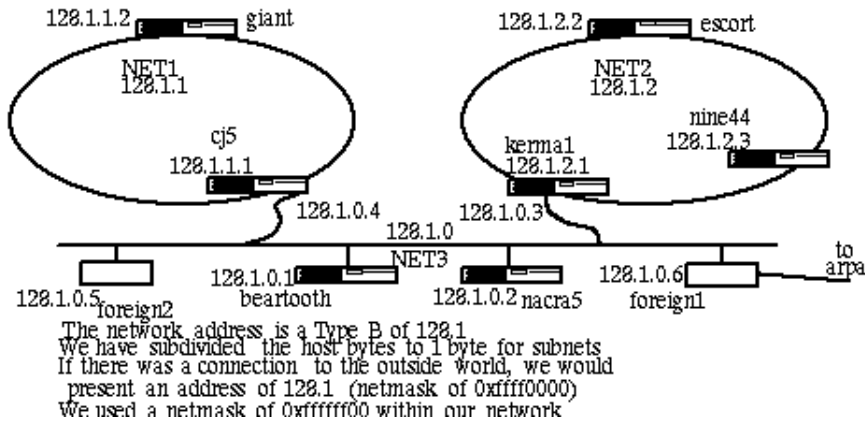
Type B: N.N.H.H

10	14 bits network	16 bits host
----	-----------------	--------------

Subnetted Type B2: N.N.S.H

10	14 bits network	8 bits subnet	8 bits host
----	-----------------	---------------	-------------

Internet Subnetted Addresses



Supernets

A supernet is a collection of smaller networks. Supernetting is a technique of using the netmask to aggregate a collection of smaller networks into a supernet. This technique is particularly useful for class C networks. A Class C network can only have 254 hosts. This can be too restrictive for some companies. For these companies, a netmask that only contains a portion of the network part can be applied to the hosts in these class C networks to form a supernet.

This supernet netmask should be applied to those interfaces that connect to the supernet using the ifconfig command. For example, a host can configure its interface to connect to a class C supernet, 192.6, by configuring an IP address of 192.6.1.1 and a netmask of 255.255.0.0 to its interface.

Internet Addresses to host name mapping

/etc/hosts file

Provides static mapping of addresses to host names. Works fine in small to medium size networks, as network size increases, the complexity of upkeep of this file increases, as does lookup time.

/etc/mkhosts (Obsolete)

Creates a hashed set of hosts files for quicker lookup by TCP if named is not used (hosts.pag and hosts.dir)

NIS (Yellow Pages)

Replaces /etc/hosts lookups

/etc/named

Replaces /etc/hosts lookups

Provides a dynamic method of mapping domain names to IP (Internet) addresses

Talks to other nameds to resolve names outside of it's area of authoritative (local) control

Each named only knows about names in it's local area of authoritative control

Recommended for large internets, internets with multiple areas of administrative control, and always when on the INTERNET (arpanet)

Routing

Routing is the process by which a packet is directed through the internet between its source and destination

Two types of routing information, static and dynamic. Static works for small fixed internets, but requires that the configuration not change, and that the administrator know the configuration of all networks this network will talk to.

/etc/route

A command that installs a static route to a network in the nodes routing tables, by describing the host to send packets to, for a particular network. (send packets to host c to reach network b)

/etc/routed

Performs dynamic routing by contacting other hosts running routed to determine the actual network topology. A bit of a resource hog, but good when the topology changes, or the external topology is unknown

/usr/sbin/gated

Replacement for routed, used on newer UNIX boxes instead of routed.

Usually a combination of both is used in an internet, local nets static, unknown network packets are sent to a host running routed or gated.

Quiz #1

1. TCP defines the protocol for communication between _____.
2. IP defines the protocol for communication between _____.
3. TCP is a connection oriented protocol. True___ False ___
4. IP is a connection oriented protocol. True___ False ___
5. The 48-bit ethernet address is used for _____
6. The above numbers are used to: _____
7. There are _____ bits, represented by four _____ in an internet address
8. An internet address is broken down into two sections. List them. _____
9. What determines how many bits are used for each section?

-
10. What is the Domain Naming Server used for?
 11. What is the purpose of the *hosts.dir* and *hosts.pag* files?
-

TCP/IP Daemons

/etc/tcpd (Obsolete, functions now performed by kernel)

TCP/IP daemon. Initializes internal tables required by the protocols supported by TCP/IP, and enables a node's BSD socket call interface. Found on some systems, not on others. Many systems required a kernel edit to enable networking

routed

Daemon which manages the network routing tables. Normally only run continuously on gateways, on non-gateways it is invoked with the -h option to exit once routing is stable. Not used at all in small stable networks

gated

Similar to routed, but supports more protocols

named

Name server daemon that provides Domain name and host name to internet address resolution, replaces /etc/hosts lookup

inetd

A super daemon, invokes internet services when needed:

telnetd	ftpd	rexecd
rlogind	etc/rshd	tftpd
ntalkd	comsat	fingerd

TCP/IP Information files

Files that are usually the same on all hosts in the same network

/etc/hosts

Relates host names and aliases to their Internet addresses

/etc/networks

Associates Internet network numbers with network names for all accessible networks, not needed.

/etc/gateways

Resides on gateway host. Or resides on administrative host with links from all gateway hosts

Only needed in networks that contain a gateway that does not support RIP (Routing Information Protocol) and cannot run routed

Contains static routes loaded into routed's routing tables

/etc/hosts.equiv (or \$HOME/.rhosts)

Lists equivalent hosts for rlogin/telnet.... purposes (allows login without password)

/etc/resolv.conf

Contains information needed by named

TCP/IP Local Information files

Files that are often different on each host depending on network function

/etc/inetd.conf

Contains configuration information used by inetd to determine what services inetd will provide (telnet, ftp, tftp)

/etc/rc.local or /etc/*net* or /etc/rc.config.d/*net*

Startup file for local tcp/ip daemons, network data, config commands, and miscellaneous network commands

/etc/protocols

Contains valid protocols used by services on local host (ip, tcp, udp) (should not need editing)

/etc/services

Contains valid services for local host (finger, whois) (often needs editing)

All name server data base files

TCP Admin commands

/etc/arp

Address resolution display and control program.

Displays and modifies Internet to Ethernet address translation tables. Used to enter in this nodes address map, the ip address and ethernet address pair of hosts that do not support ARP (most do).

[/usr]/bin/hostname

Utility to display and modify hostname.

ifconfig

Program to assign internet addresses to a network interface.

ping spray

Utility which allows TCP to determine which hosts and gateways are still operative by eliciting an echo_response from them

[/usr/sbin|etc]/route

Program to manually manipulate, and list the network routing tables.

hostns or hosts_to_named

Program to convert /etc/hosts to format required by name server (/etc/named)

htable

Program to convert NIC standard host tables and local host tables to format required by network library routines. (/etc/hosts, /etc/gateways, /etc/networks)

gettable

Program to obtain NIC standard host tables

nshost

Program to convert name server config files to /etc/hosts format

routed

NAME

routed - network routing daemon

SYNOPSIS

routed [-g] [-s] [-q] [-t] [-n] [-f] [-h] [logfile]

DESCRIPTION

The routed daemon is invoked at boot time to manage the network routing tables. When routed is started, it uses the SIOCGIFCONF ioctl(2) to find those directly connected interfaces configured into the system and marked "up" (the software loopback interface is ignored). If multiple interfaces are present, it is assumed that the host will forward packets between networks. routed then transmits a request packet on each interface. When a request packet is received, routed formulates a reply based on the information maintained in its internal tables. The response packet generated contains a list of known routes, each marked with a "hop count" metric (a count of 16, or greater, is considered "infinite"). The metric associated with each route returned provides a metric relative to the sender.

OPTIONS

-g This flag is used on internetwork routers to offer a route to the "default" destination. This option is typically used on a gateway to the Internet, or on a gateway that uses another routing protocol whose routes are not reported to other local routers.

-s Forces routed to supply routing information whether it is acting as an internetwork router or not. This is the default if multiple network interfaces are present, or if a point-to-point link is in use.

-q This option is the opposite of the -s option. With this option, a host runs the Routing Information Protocol. It listens for broadcast updates but does not broadcast. The -q option is recommended for all non-gateway hosts.

-t If the -t option is specified, all packets sent or received are printed on the standard output. In addition, routed will not divorce itself from the controlling terminal, so that interrupts from the keyboard will kill the process.

gated

gated [-c] [-n] [-ttrace_options] [-f config_file] [trace_file]

gated is a routing daemon that handles the RIP, BGP, EGP, and HELLO routing protocols. The gated process can be configured to perform all routing protocols or any combination of the four (see WARNINGS below).

-c Parse the configuration file for syntax errors then exit. If there were no errors, leave a dump file in /usr/tmp/gated_dump. Running gated does not require super-user privilege when using the -c option, but gated may not be able to read the kernel's routing table unless it is run as super user. The -c option implies -tierk.

-n Do not modify the kernel's routing table. This option is used for testing gated configurations with actual routing data.

-ttrace_options Enable trace flags on startup. trace_options can include one or more of the following values:

A	all	P	protocol
i	internal	u	update
e	external	R	RIP
k	kernel	H	hello
r	route	C	icmp
m	mark	p	EGP
t	nostamp	B	BGP

-f config_file Use an alternate configuration file. By default, gated uses /etc/gated.conf.

trace_file Trace file in which to place trace information.

SIGHUP Re-read configuration.

SIGINT Snapshot of current state.

The current state of all gated tasks, timers, protocols, and tables are written to /usr/tmp/gated_dump.

SIGTERM Graceful shutdown.

SIGUSR1 Toggle tracing.

NINGS

At HP-UX 8.07, the shipped version of gated did not require a config file. At 9.01 and above it does, or gated dies.

Also, gated is known for expiring it's own interface, but this can be fixed with an entry in the config file.

Here is a sample of the config file options most commonly used, use man gated.conf for the complete list:

gated.conf -

gated configuration file syntax

The gated config file consists of a sequence of statements terminated by a semicolon (;).

Statements are composed of tokens separated by white space, which can be any combination of blanks, tabs and new-line characters.

Comments:

Starting with a # character and running to the end of the line.

“C” style, which starts with a /* and continues until it reaches */.

Statement Classes:

There are four (6) classes of statements. The four listed here must be specified in order:

Definition, protocol, static routes, control

/etc/gated.conf

Definition statements

These statements specify options, the autonomous system, martian networks, and interface options.

`interface interface_list interface_options ;`

Sets interface options on the specified interfaces. An interface list is all or a list of interface names

`metric metric`
Set the interface metric for this interface. This metric is used by RIP and HELLO.

Specifying the metric here overrides for internal use, but does not change the metric set by `ifconfig`.

`preference pref`

Sets the preference for routes to this interface.

`passive`

Prevents gated from deleting the route to this interface if it is believed to be down due to routing information not received.

```
martians {
  martian_list
};
```

Defines a list of martian addresses about which all routing information is ignored.

Protocol statements

These statements enable or disable protocols and set protocol options.

Routing Information Protocol (RIP):

```
rip yes|no|on|off|quiet|pointopoint|supplier [ {
  preference preference ;
  defaultmetric metric ;
  interface interface_list [noripin] [noripout] ;
  ...
  trustedgateways gateway_list ;
  sourcegateways gateway_list ;
} ] ;
```

If `yes` or `on` is specified, RIP assumes `quiet` if there is only one interface and `supplier` if there are more than two or more.

`quiet` specifies that no RIP packets are to be generated. `supplier` specifies that RIP packets are to be generated. `pointopoint` specifies that RIP packets are to be sent only to gateways listed in the `sourcegateways` clause. If the RIP clause is not specified, the default is `on`.

The default metric is 16, the default preference is 100.

HELLO Protocol:

```
hello yes|no|on|off|quiet|pointopoint|supplier [ {
  preference preference ;
  defaultmetric metric ;
  interface interface_list [nohelloin] [nohelloout] ;
  ...
  trustedgateways gateway_list ;
  sourcegateways gateway_list ;
} ] ;
```

If yes or on is specified, HELLO assumes quiet if there is only one interface and supplier if there are two or more.

quiet specifies that no HELLO packets are to be generated. supplier specifies that HELLO packets are to be generated. Pointpoint specifies that HELLO packets are to be sent only to gateways listed in the sourcegateways clause. If the HELLO clause is not specified the default is off. The default metric is 30000, the default preference is 90.

Exterior Gateway Protocol (EGP):

```
egp yes|no|on|off [ {
  preference preference ;
  defaultmetric metric ;
  packetsize maxpacketsize ;
  group          [asin autonomous_system]
                 [asout autonomous_system]
                 [maxup number]
                 [preference preference] {
    neighbor host
      [metricout metric]
      [nogendefault]
      [acceptdefault]
      [propagatedefault]
      [gateway gateway]
      [interface interface]
      [sourcenet network]
      [minhello min_hello]
      [minpoll min_poll]
    ;
  } ; ...
} ] ; ...
```

packetsize specifies the size, in bytes, of the largest EGP packet to be accepted or sent. A group lists a group of EGP peers in one autonomous system. maxup specifies the maximum number of peers to be maintained in the Up state. acceptdefault and propagatedefault tell gated to accept or propagate the default network (0.0.0.0) in updates exchanged with an EGP neighbor. If not specified, the default network is ignored when exchanging EGP updates. sourcenet specifies the network to query in EGP Poll packets, this is normally the shared network. The minimum acceptable EGP hello and poll intervals can be specified with the minhello and minpoll arguments, respectively. These are both specified as a time in seconds, minutes:seconds, or hours:minutes:seconds. Any number of group clauses can be specified containing any number of neighbor clauses. Any parameters from the neighbor clause can be specified in the group clause to provide defaults for the group.

The default metric is 255, the default preference is 200.

Border Gateway Protocol (BGP):

Note that although BGP is available with this version of gated, it is currently not supported by HP.


```

bgp yes|no|on| off [ {
    preference preference ;
    defaultmetric metric ;
    peer host
        [linktype [up|down|horizontal|internal]]
        [metricout metric]
        [asin autonomous_system]
        [asout autonomous_system]
        [nogendefault]
        [gateway gateway]
        [interface interface]
        ;
        ...
    } ] ;

```

peer specifies the address of each BGP peer. A linktype internal is assumed if the neighbor's autonomous system is the same as my autonomous system. Otherwise linktype horizontal is assumed.

The default metric is 65535 and the default preference is 150 for external BGP and 250 for internal BGP.

Redirect (ICMP):

```

redirect yes|no|on|off [ {
    preference preference ;
    interface interface_list [noicmpin] ;
    trustedgateways gateway_list ;
} ] ;

```

Controls whether gated makes routing table changes based on ICMP redirects when not functioning as a router. When functioning as a router (i.e. any interior routing protocols (RIP, HELLO) are participating in routing on any interface, ICMP redirects are disabled. When ICMP redirects are disabled, gated must actively remove the effects of redirects from the kernel as the kernel always processes ICMP redirects. The default preference is 20.

Static route statements

Static routes are defined by route statements.

```

static {
    destination gateway gateway [preference
    preference] ;
    ...
    destination interface interface [preference
    preference] ;
    ...
} ;

```

Any number of static statements can be specified, each containing any number of static route definitions. The first form defines a static route through a gateway. The second defines a static interface route which is used for primitive support of multiple networks on a single interface.

The preference for static routes defaults to 50.

Control statements

Control statements define routes that are accepted from routing peers and routes that are propagated to those peers. The clauses define which hosts to accept routes from, and propagate routes to .

Luckily there are examples provided in the HP-UX 9.XX directory:
/etc/newconfig/gated/conf

Better still, this is what you will probably use on most hosts and gateways in a local internet:

```
rip yes ;
```

As well as some static routes possibly.

Sample /etc/gated.conf

```
#      Config file gated on Achilles
#      Gated Version 2.0.1.2

traceoptions internal external route rip update ;

interface all passive ;      # don't time out my interfaces!

#rip yes ;
rip supplier {
    interface fddi0 noripout ;
        #no rip onto CTD FDDI/don't want to be a router
    trustedgateways 130.202.64.5 ; # trust sungate
} ;

# need some static routes
static {
# default is via Sungate
    default gateway 130.202.64.5 ;
# XMP via FDDI and Network Systems Corp
    146.137.5.0 gateway 146.137.5.129 ;
} ;

propagate proto rip interface ie2 {
    proto static metric 1 {
        announce default ;
    } ;
} ;

propagate proto rip interface ie0 {
    proto direct {
        announce 130.202.60 metric 1 ;
        # announce path to VISNet
    } ;
} ;
```

named

NAME

named - Internet domain name server

SYNOPSIS

```
named [ -d debuglevel ] [ -p port# ] [ -b bootfile ]
```

DESCRIPTION

Without any arguments, named will read the default boot file /etc/named.boot, read any initial data and listen for queries.

OPTIONS

-d debuglevel Print debugging information. A number after the "d" determines the level of messages printed.

-p port# Use a different port number. The default is the standard port number as listed in /etc/services.

-b Use an alternate boot file. This is optional and allows you to specify a file with a leading dash.

Any additional argument is taken as the name of the boot file. The boot file contains information about where the name server is to get its initial data. If multiple boot files are specified, only the last is used.

EXAMPLE

The following example shows a boot file:

```
; boot file for name server
directory /usr/local/domain
; type      domain          source host/file          backup file
cache      .                  .                          root.cache
primary    Berkeley.EDU      berkeley.edu.zone
primary    32.128.IN-ADDR.ARPA    ucbhosts.rev
secondary  CC.Berkeley.EDU        128.32.137.8 128.32.137.3 cc.zone.bak
secondary  6.32.128.IN-ADDR.ARPA  128.32.137.8 128.32.137.3 cc.rev.bak
primary    0.0.127.IN-ADDR.ARPA    localhost.rev
forwarders 10.0.0.78 10.2.0.78
; slave
```

FILES

```
/etc/named.boot      name server configuration boot file
/etc/named.pid       the process id
```

```
/usr/tmp/named.run          debug output
/usr/tmp/named_dump.db     dump of the database
/usr/tmp/named.stats       name server statistics data
Configuration files read by /etc/named.boot:
/etc/named.ca
/etc/named.hosts
/etc/named.local
/etc/named.rev
```

inetd

NAME

inetd - internet "super-server"

SYNOPSIS

```
/etc/inetd [ -d ] [ configuration file ]
```

DESCRIPTION

Run at boot time by rc files. It listens for connections on certain internet sockets. When a connection is found on one of its sockets, it decides what service the socket corresponds to, and invokes a program to service the request. Essentially, inetd allows running one daemon to invoke several others, reducing load on the system.

Upon execution, inetd reads its configuration information from a configuration file which, by default, is `/etc/inetd.conf`. inetd rereads its configuration file when it receives a hangup signal, SIGHUP. (kill -hup *pid*) The fields of the configuration file are as follows:

```
service name  socket type  protocol  wait/nowait  user
              server program  server program arguments
```

Example:

```
telnet  stream  tcp  nowait  root  /etc/telnetd  telnetd
```

/etc/hosts

NAME

hosts - host name database

DESCRIPTION

The hosts file contains information regarding the known DARPA Internet hosts with which your Domain node can communicate (usually via TCP/IP). For each host, a single line should be present with the following information:

```
Internet address  official host name  aliases
```

A "#" indicates the beginning of a comment.

When using the name server named(8), this file provides a backup when named is not running. For the name server, it is suggested that only a few addresses be included in this file. These include address for the local interfaces that ifconfig(8C) needs at boot time and a few machines on the local network.

Host names may contain any printable character other than a field delimiter, newline, or comment character.

EXAMPLE

The example below is for the network shown on an earlier page in the handout

```
# /etc/hosts  Rev Date 6/9/90 by FAM
203.1.12.3      escort          lynx.crp.fame.com
203.1.12.2      nine44         speedo
203.1.12.1      kernal
151.2.3.1       cj5            trucker #gateway
151.2.3.2       beartooth
151.2.3.3       nacra5
151.2.3.4       kernal          #gateway
192.2.2.1       giant
192.2.2.2       cj5
127.0.0.1      localhost
##!!!! NOTE: Do not use any leading spaces, or blank lines.
#Do not "pretty-up" the columns with leading "0"'s
```

/etc/networks

NAME

networks - network name database

DESCRIPTION

The /etc/networks file contains information regarding the known networks which comprise the DARPA Internet. For each network a single line should be present with the following information:

```
official network name      network number      aliases
```

A "#" indicates the beginning of a comment

Network numbers may be specified in the conventional "." notation. Network names may contain any printable character other than a field delimiter, newline, or comment character.

EXAMPLE

The example below is for the network shown on an earlier page in the handout

```
#/etc/networks  Rev Date 5/10/90 by FAM
net2            203.1.12       ring2
```

```
net3          151.2          ethernet
net1          192.2.2 ring1
local        127.0.0
```

Network startup file overview

All startup files, no matter the system, must do the following:

- Set the hostname
- Start the tcpd if the system is not kernel based
- Configure the lan interface with ifconfig
- Establish the routing method, and invoke it:
routed/gated/route
- Then any other tools and commands can be started:
 - inted
 - named
 - X
 - NIS
 - NFS
 - mail
 - snmpd
 - arp
 - syslogd
 - rwhod
 - timed
 - nmconfig

Sample HP-UX netlinkrc file (9.01)

```
#!/bin/sh
# @(#)netlinkrc: $Revision: 1.6.109.7 $ $Date: 92/07/13 08:21:12 $
# $Locker: $
# net_init flag is used for Instant Ignition. If net_init is set,
# then netlinkrc return "exit 1". In order for Instant Ignition
# to work correctly, netlinkrc needs to check the STATUS variable
# after each program or scripts it calls.
net_init=0
if [ -f /etc/clusterconf ]
then
    ROOTSERVER=`/bin/cnodes -r`
    NODENAME=`/bin/cnodes -m`
    DOMAIN=`/bin/cnodes -r`
    ORGANIZATION=diskless
else
    ROOTSERVER=`hostname`
    NODENAME=$ROOTSERVER
    DOMAIN=`/bin/uname -n`
    ORGANIZATION=standalone
fi
# Start logging daemon *before* any other networking initialization.
# See nettl(1m) for more information.
/etc/nettl -start
STATUS=$?
```

```

if [ ! $STATUS -eq 0 ]
then
    net_init=1
fi
# Remove the existing /etc/netstat_data file. The first time
# netstat is executed, a new /etc/netstat_data file will be
# created.
/bin/rm -f /etc/netstat_data
# Initialize networking interfaces.
# (STEP 1)
# The "case $NODENAME" construct below allows each node in a diskless cluster
# to execute node specific calls if necessary. Add entries to
# the case construct for specific nodes in the diskless cluster only if
# needed. For example, if a specific node has more than one LAN interface,
# the node must execute separate commands for each of the interfaces.

# NOTE: If the ifconfig command line does not specify a subnet mask,
#       the subnet mask defaults to the network mask.
#       It is not necessary for both encapsulation methods to be turned on
#       for the LAN Interface. For further explanation see lanconfig(1m)
#
# The loopback interface must be explicitly configured for each address
# family of interest. The following command assumes that the hostname
# has already been set and is mapped to an IP Address in /etc/hosts.
#
# SEE ALSO: ifconfig(1m), lanconfig(1m)

case $NODENAME in
*) /etc/ifconfig lan0 inet `hostname` up
   STATUS=$?
   if [ ! $STATUS -eq 0 ]
   then
       net_init=1
   fi
   /etc/lanconfig lan0 ether
   STATUS=$?
   if [ ! $STATUS -eq 0 ]
   then
       net_init=1
   fi
   ;;
esac
/etc/ifconfig lo0 inet 127.0.0.1 up
STATUS=$?
if [ ! $STATUS -eq 0 ]
then
    net_init=1
fi
#
# Initialize network routing.
#
# (STEP 2) (OPTIONAL, FOR NETWORKS WITH GATEWAYS ONLY)
#
# The route(1m) command manipulates the network routing tables.
# The "case $NODENAME" construct below allows each node in a diskless
# cluster to execute node specific route calls if necessary. Add entries
# to the case construct for specific nodes in the diskless cluster if needed.
# The STATUS checking is for Instant Ignition.
#
#
# SEE ALSO: route(1m), routing(7)

```

```

case $NODENAME in
    *) # add route commands for specific nodes here
        ;;
esac

#

# Initialize the network node name.
#
# (STEP 3)
#
# The nodename(1m) command assigns an NS node name to the node.
# Nodename takes an option of the form "nodename.domainname.orgname" where,
#
#   nodename           is the name of the local node
#   domainname        is the name of the domain
#   orgname            is the name of the organization
#
# Each name must start with an alphabetic character.
#
# It is strongly recommended that the string used for "nodename" above be
# identical to the string used as an argument to the hostname(1) command,
# which is typically invoked from the system initialization shell script
# file "/etc/rc".  The NS nodename used on each node in your network needs
# to be unique within that network.  The "case $NODENAME" construct below
# allows each node in a diskless cluster to execute a node specific
# nodename(1) call if necessary.  Add entries to the case construct for
# specific nodes in the diskless cluster only if needed.
#
# For example,
#
#   case $NODENAME in
#       * ) /bin/nodename ` /bin/uname -n`.mydomain.myorg
#           ;;
#   esac
#
# sets the NS nodename for all nodes (* is the wildcard) in domain
# "mydomain" and organization "myorg".
#
# The nodename command line below sets the nodename field to the system
# hostname, the domainname field to the rootserver's name, and the orgname
# field to "diskless".
#
# SEE ALSO: nodename(1)

if [ -x /bin/nodename ]
then
    case $NODENAME in
        *) /bin/nodename ` /bin/uname -n`.$DOMAIN.$ORGANIZATION
            STATUS=$?
            if [ ! $STATUS -eq 0 ]
            then
                net_init=1
            fi
            ;;
    esac
fi

#

```



```

# Start remote loop back daemon
#
if [ -f /usr/adm/rld.log ]
then
    /bin/mv /usr/adm/rld.log /usr/adm/OLDrld.log
fi
if [ -x /etc/rlbdaemon ]
then
    (/etc/rlbdaemon 2>&1 ) > /usr/adm/rld.log
    STATUS=$?
    if [ ! $STATUS -eq 0 ]
    then
        net_init=1
    fi
fi

/bin/echo "Network Link started"

#
# Start NFS. This requires installation of the NFS product.
#
if [ -x /etc/netnfsrc ]
then
    /etc/netnfsrc
    STATUS=$?
    if [ ! $STATUS -eq 0 ]
    then
        net_init=1
    fi
fi

/bin/echo "ARPA/Berkeley daemons started: \c"

#
# Start the Internet daemon.
#

[ -x /etc/inetd ] && /etc/inetd && /bin/echo "inetd \c"
STATUS=$?
if [ ! $STATUS -eq 0 ]
then
    net_init=1
fi

#
# Start ARPA/BSD networking services.
#
if [ -x /etc/netbsdsrc ]
then
    /etc/netbsdsrc
    STATUS=$?
    if [ ! $STATUS -eq 0 ]
    then
        net_init=1
    fi
fi

/bin/echo

#
# Do nfs mounts after inetd is running

```

```

#
if [ -x /etc/netnfsrc2 -a -f /etc/nfs.up ]
then
    /etc/netnfsrc2
    STATUS=$?
    if [ ! $STATUS -eq 0 ]
    then
        net_init=1
    fi
fi

#
# Start NS networking services.
#
if [ -x /etc/netnssrc ]
then
    /etc/netnssrc
    STATUS=$?
    if [ ! $STATUS -eq 0 ]
    then
        net_init=1
    fi
fi

#
# Start HP Network Management Agent
#
if [ -x /etc/netnmrc ]
then
    /etc/netnmrc
    STATUS=$?
    if [ ! $STATUS -eq 0 ]
    then
        net_init=1
    fi
fi

#
# Start HP LAN Manager/X.
#
if [ -x /etc/netlmrc ]
then
    /etc/netlmrc
    STATUS=$?
    if [ ! $STATUS -eq 0 ]
    then
        net_init=1
    fi
fi

#
# Start NCS. This requires installation of the NCS product.
# NCS must be started before any other NCS products are started.
#
if [ -x /etc/netncsrc ]
then
    /etc/netncsrc
    STATUS=$?
    if [ ! $STATUS -eq 0 ]
    then

```

```

        net_init=1
    fi
fi

#
# Start NetLS.  This requires installation of the NetLS product.
# NCS must be started before NetLS is started.
#
if [ -x /etc/netlsrc ]
then
    /etc/netlsrc
    STATUS=$?
    if [ ! $STATUS -eq 0 ]
    then
        net_init=1
    fi
fi

# return exit code for Instant Ignition
if [ $net_init -eq 0 ]
then
    exit 0
else
    exit 1
fi

```

Sample HP-UX /etc/rc.config.d/netconf

```

# netconf:  configuration values for core networking subsystems
# @(#) $Revision: 1.3.111.2 $ $Date: 95/01/25 11:35:38 $
# HOSTNAME:          Name of your system for uname -S and hostname
# OPERATING_SYSTEM: Name of operating system returned by uname -s
#                   ----- DO NOT CHANGE THIS VALUE -----
# LOOPBACK_ADDRESS: Loopback address
#                   ----- DO NOT CHANGE THIS VALUE -----
# IMPORTANT:  for 9.x-to-10.0 transition, do not put blank lines between
# the next set of statements
HOSTNAME="puxy"
OPERATING_SYSTEM=HP-UX
LOOPBACK_ADDRESS=127.0.0.1
# Internet configuration parameters.  See ifconfig(1m), lanconfig(1m)
# INTERFACE_NAME:    Network interface name (see lanscan(1m))
# IP_ADDRESS:        Hostname (in /etc/hosts) or IP address in decimal-dot
#                   notation (e.g., 192.1.2.3)
# SUBNET_MASK:       Subnetwork mask in decimal-dot notation, if different from def
# BROADCAST_ADDRESS: Broadcast address in decimal-dot notation, if different from d
# LANCONFIG_ARGS:    Link-layer encapsulation methods (e.g., ieee, ether).  See
#                   lanconfig(1m) for details.
# For each additional network interfaces, add a set of variable assignments
# like the ones below, changing the index to "[1]", "[2]" et cetera.
# IMPORTANT:  for 9.x-to-10.0 transition, do not put blank lines between
# the next set of statements

INTERFACE_NAME[0]=lan0
IP_ADDRESS[0]="179.102.3.2"
SUBNET_MASK[0]=" "
BROADCAST_ADDRESS[0]=" "
LANCONFIG_ARGS[0]="ether"
# Internet routing configuration.  See route(1m), routing(7)
# ROUTE_DESTINATION: Destination hostname (in /etc/hosts) or host or network

```

```

#           IP address in decimal-dot notation, preceded by the word
#           "host" or "net"; or simply the word "default".
# ROUTE_MASK: Subnetwork mask in decimal-dot notation, or C language
#           hexadecimal notation. This is an optional field.
#           A IP address, subnet mask pair uniquely identifies
#           a subnet to be reached. If a subnet mask is not given,
#           then the system will assign the longest subnet mask
#           of the configured network interfaces to this route.
#           If there is no matching subnet mask, then the system
#           will assign the default network mask as the route's
#           subnet mask.
# ROUTE_GATEWAY: Gateway hostname (in /etc/hosts) or IP address in
#           decimal-dot notation. If local interface, must use the
#           same form as used for IP_ADDRESS above (hostname or
#           decimal-dot notation).
# ROUTE_COUNT: An integer that indicates whether the gateway is a
#           remote interface (one) or the local interface (zero).
# ROUTE_ARGS: Route command arguments and options. This variable
#           may contain a combination of the following arguments:
#           "-f", "-n" and "-p pmtu".

# For each additional route, add a set of variable assignments like the ones
# below, changing the index to "[1]", "[2]" et cetera.
# IMPORTANT: for 9.x-to-10.0 transition, do not put blank lines between
# the next set of statements
ROUTE_DESTINATION[0]="default"
ROUTE_MASK[0]=" "
ROUTE_GATEWAY[0]="179.102.3.2"
ROUTE_COUNT[0]="0"
ROUTE_ARGS[0]=" "

# Dynamic routing daemon configuration. See gated(1m)
# GATED: Set to 1 to start gated daemon.
# GATED_ARGS: Arguments to the gated daemon.

GATED=0
GATED_ARGS=" "

# Router Discover Protocol daemon configuration. See rdpd(1m)
# RDPD: Set to 1 to start rdpd daemon

RDPD=0

```

ifconfig

NAME

ifconfig - configure network interface parameters

SYNOPSIS

```

/usr/sbin/ifconfig or /etc/ifconfig
ifconfig interface [address_family (inet)
[address[dest_addr]][parameters]]
ifconfig interface [address_family]

```

DESCRIPTION

ifconfig is used to assign an address to a network interface and/or configure network interface

parameters. The second format shown above displays current status. `ifconfig` must be used at boot time to define the network address of each interface present on a machine.

```
ethernet on hpux                lan0, lan1, lan2
(Use lanscan to get a list of devices)
Apollo token ring on hpux      atr1
```

Addresses supplied can be names from the `hosts/networks` files or dotted decimal notation

PARAMETERS

`up` Mark an interface "up." This may be used to enable an interface after an "ifconfig down."

`down` Mark an interface "down." When an interface is marked "down," the system will not attempt to transmit messages through that interface.

`trailers` Request the use of a "trailer" link level encapsulation when sending (default). If a network interface supports trailers, the system will, when possible, encapsulate outgoing messages in a manner which minimizes the number of memory to memory copy operations performed by the receiver. On networks that support the Address Resolution Protocol (see `arp(4P)`; currently, only 10 MB ETHERNET), this flag indicates that the system should request that other systems use trailers when sending to this host. Similarly, trailer encapsulations will be sent to other hosts that have made such requests. Currently used by Internet protocols only.
NOTE: If any host on the network is running `bsd4.2`, dis-able trailers on all hosts. (No effect on HP-UX)

`-trailers` Disable the use of a "trailer"

`arp` Enable the use of the Address Resolution Protocol in mapping between network level addresses and link level addresses (default). This is currently implemented for mapping between DARPA Internet addresses and 10MB ETHERNET addresses.

`-arp` Disable the use of the Address Resolution Protocol.

`debug` Enables extra logging to the console

`-debug` Default

`metric n` Set the routing metric of the interface to `n`, default 0. Higher metrics have the effect of making a route less favorable; metrics are counted as addition hops to the destination network or host.

`debug` Enable driver dependent debugging code; usually, this turns on extra console error logging.

`-debug` Disable driver dependent debugging code.

`netmask mask (Inet only)` Specify how much of the address to reserve for subdividing networks into subnetworks. The mask can be specified as a single hexadecimal number with a leading 0x, with a dot-notation Internet address, or with a netmask name listed in the host table. The mask contains 1's for the bit positions to be used for the network and subnet

parts, and 0's for the host part. On Apollo hosts, if you use the netmask name "defaultmask" with ifconfig, and "defaultmask" is listed in hosts(5) with a subnet mask value, ifconfig succeeds and sets the subnet mask to the value found in hosts. If "defaultmask" is not found in the hosts file, ifconfig still succeeds but sets the mask to the default for the address class. For example, the default mask for a Class C network is 255.255.255.0

netmask icmp-request Specifies that the host should obtain its subnet mask from local gateways, using the ICMP Address Mask Request/Reply protocol. A node will respond to this request only if it is a gateway (that is, it has more than one network interface enabled) and if its own subnet mask was explicitly set with the "netmask mask" parameter (and it did not itself use the ICMP request protocol). If the requesting host receives no responses within 2 seconds, it assumes that subnets are not in use on the network. You also can enable this ICMP Address Mask protocol by assigning the address 255.255.255.255 to the netmask name "defaultmask" in the hosts file.

dest_addr Specify the address of the correspondent on the other end of a point to point link.

broadcast (Inet only) Specify the address to use to represent broadcasts to the network. The default broadcast address is the address with a host part of all 1's. If the host must interoperate with older TCP implementations that require the 0-form IP broadcast address, specify "broadcast 0". You cannot specify this parameter when configuring either a loopback or point-to-point interface.

The ifconfig lines used in the network depicted on page 11.

```
Gateway cj5(if interoperating with BSD4.2, use "broadcast 0 -trailers")
/etc/ifconfig atr0 192.2.2.2 up netmask 0xfffff00 trailers
/etc/ifconfig lo0 127.0.0.1 up
/etc/ifconfig lan0 151.2.3.1 up netmask 0xffff0000 trailers
```

```
Gateway kernal /etc/ifconfig atr0 203.1.12.1 up netmask 0xfffff00 trailers
/etc/ifconfig lo0 127.0.0.1 up
/etc/ifconfig lan0 151.2.3.4 up netmask 0xffff0000 trailers
```

Using ifconfig to check interface configuration

```
[135]$ /etc/ifconfig atr0
atr0: flags=43<UP,BROADCAST,RUNNING>
inet 203.1.12.2 netmask fffff00 broadcast 203.1.12.255
```

```
# ifconfig lan0
lan0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING>
inet 179.102.3.2 netmask ffff0000 broadcast 179.102.255.255
```

route

NAME

route - manually manipulate the routing tables

SYNOPSIS

```
route [ -n ] [ -f ] [ -p pmtu ] [ cmd [ net | host ] args ]
```

DESCRIPTION

route is a program used to manually manipulate the network routing tables. It normally is not needed, as the system routing table management daemon, routed or gated, should tend to this task. When external routers are used, a default route is often needed, the default route most can be this host. route accepts two commands:

```
add add a route
delete delete a route
```

Routes can be displayed with the command netstat -r

COMMAND SYNTAX

All commands have the following syntax:

```
command [net|host] destination gateway [metric]
```

destination is the destination host or network
gateway is the next-hop gateway to which packets should be addressed
metric is a count indicating the number of hops to the destination.

The metric is required for add and addp commands; it must be zero if the destination is on a directly-attached network, and nonzero if the route utilizes one or more gateways. If adding a route with metric 0, the gateway given is the address of this host on the common network, indicating the interface to be used for transmission.

The optional keywords net and host force the destination to be interpreted as a network or a host, respectively. If the route is to a destination connected through a gateway, the metric should be greater than 0.

Add a default route as follows:

```
/etc/route add default gateway_name [non-zero metric]
```

TCP/IP software will use the default route when other routes occurring earlier in the routing table have failed, or when there are no other possible routes.

Only the super-user may modify the routing tables.

OPTIONS

-f "Flush" the routing tables of all gateway entries. Using this option in conjunction with one of the commands described above flushes the tables prior to the command's application.

-n Use dot notation instead of names.

DIAGNOSTICS

add [host | network] %s: gateway %s flags %x

This, when returned means the specified route is being added to the tables.

delete [host | network] %s: gateway %s flags %x

As above, but when deleting an entry.

%s %s done

When the -f flag is specified, each routing table entry deleted is indicated with a message of this form.

Network is unreachable

An attempt to add a route failed because the gateway listed was not on a directly-connected network. The next-hop gateway must be given.

not in table

A delete operation was attempted for an entry that wasn't present in the tables.

routing table overflow

An add operation was attempted, but the system was low on resources and was unable to allocate memory to create the new entry.

If static routing is used in the network from an earlier diagram in this handout, here are some commands that would work:

On gateway cj5

```
[458]$ /etc/route add net 151.2 192.2.2.2 0
```

```
add net 151.2: gateway 192.2.2.2
```

```
[459]$ /etc/route add net 203.1.12 151.2.3.4 1
```

```
add net 203.1.12: gateway 151.2.3.4
```

On gateway kernal

```
[123]$ /etc/route add net 151.2 203.1.12.1 0
```

```
add net 151.2: gateway 203.1.12.1
```

```
[124]$ /etc/route add net 192.2.2 151.2.3.1 1
```

```
add net 192.2.2: gateway 151.2.3.1
```

If routed or gated is not used also, all hosts would need to have a default route set to their respective gateways, in net1:

```
[458]$ /etc/route add default 192.2.2.2
```

Quiz #2

1. When is sub-netting used? _____
2. How is subnetting set-up and controlled? (What command)

3. What is the function of the route command? _____
4. What is the function of the routed daemon? _____
5. What option to routed would you use on a host node? _____
6. When would you have to edit the /etc/rc.local file?

7. When would you use the ifconfig command?

8. What service does inetd provide? _____
9. When must you use named? _____
10. When must you use the 'broadcast 0' argument of ifconfig?

11. When must you use the '-trailers' argument of ifconfig?

12. Go back to the diagram used in the first quiz and fill in the following information:
Which nodes should be running routed, and with which options
The ifconfig command lines for each node
Show any static routes that would be helpful
Show the /etc/networks file

Internet Services

NAME

rwho - who's logged in on local machines

SYNOPSIS

rwho [-a]

DESCRIPTION

The rwho command produces output similar to who, but for all machines on the local network

NAME

telnet - user interface to the TELNET protocol

SYNOPSIS

telnet [host [port]]

DESCRIPTION

telnet is used to communicate with another host using the TELNET protocol.

NAME

rlogin - remote login

SYNOPSIS

```
rlogin rhost [ -ec ] [ -8 ] [ -L ] [ -l username ]  
rhost [ -ec ] [ -8 ] [ -L ] [ -l username ]
```

DESCRIPTION

rlogin connects your terminal on the current local host system lhost to the remote host system rhost.

NAME

ftp - ARPANET file transfer program

SYNOPSIS

```
ftp [ -v ] [ -d ] [ -i ] [ -n ] [ -g ] [ host ]
```

DESCRIPTION

ftp is the user interface to the ARPANET standard File Transfer Protocol (FTP). The program allows you to transfer files to and from a remote network site.

NAME

tftp - trivial file transfer protocol

SYNOPSIS

```
tftp [ -g|g!|p|r|w ] localname host foreignname [mode]
```

DESCRIPTION

tftp is the front-end to the Trivial File Transfer Protocol. It enables you to copy files among internet hosts without remote user-level access.

NAME

rcp - remote file copy

SYNOPSIS

```
rcp [ -p ] file1 file2  
rcp [ -p ] [ -r ] file ... directory
```

DESCRIPTION

The rcp command copies files between machines. Each file or directory argument is either a remote file name of the form rhost:path or a local filename containing no ":"

NAME

remsh - remote shell

SYNOPSIS

```
rsh host [ -l username ] [ -n ] command  
host [ -l username ] [ -n ] command
```

DESCRIPTION

rsh connects to the specified host, and executes the specified command. rsh copies its standard input to the remote command, the standard output of the remote command to its standard output, and the standard error of the remote command to its standard error. Interrupt, quit and terminate signals are propagated to the

remote command; rsh normally terminates when the remote command does.

NAME

talk - talk to another user

SYNOPSIS

talk person [ttyname]

DESCRIPTION

talk is a visual communication program that copies lines from your terminal to that of another user. If you wish to talk to someone on your own machine, then person is just the person's log-in name. If you wish to talk to a user on another host, then person is of the form user@host

Common internet services error messages

All network ports in use

You either need more pseudo tty's, or the existing ones are corrupt. To see how many exist, % ls /dev/*typ*

To create, use the /etc/crpty command

Connection refused

The tcp connection to the remote host is good! The remote host was not running the desired service daemon, or inetd, or inetd does not have that service enabled

Connection reset by peer

Network timeout

Connection timed out

Check to see if remote host tcpc is running, and check routes to host

Connection dead

Check if remote host crashed, or switched out

Network unreachable

Check for entry of network, and routes to network

Unknown host

Check /etc/hosts, or named

Unknown service

Check /etc/services, and /etc/protocols files

Trouble shooting commands

ping

netstat

lanscan

landiag

T-shooting Techniques

Using ping

Ping the local host, try loopback, internet address, and host name. If the internet address works and hostname doesn't, the problem is in naming resolution.

Ping a host on this net, using internet address, then host name

Ping the gateway host

Ping farther towards the unreachable destination, the problem may lie in another networks routes, not yours

Using route

It is possible to add, delete or change the routes known to this host

Using netstat

Use -T to get all info

Use -rn to get routing + address info, shows **status** of routes

Use -ag to see active connections, and gateway used

Use -i to see condition of interfaces, could be hardware problem

Using ifconfig

Use with interface argument only to see how an interface is currently configured

Sample Net

