

TCP/IP Services for OpenVMS

Lab Guide

Training



COMPAQ

TCP/IP Services for OpenVMS

Lab Guide

Training



COMPAQ

Notice

© 1999 Compaq Computer Corporation

COMPAQ, the Compaq logo, and the Digital logo are registered in the U.S. Patent and Trademark Office. Other product names mentioned herein may be the trademarks of their respective companies.

Confidential computer software. Valid license from Compaq or authorized sublicensor required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Compaq Computer Corporation shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is subject to change without notice.

©1999 Compaq Computer Corporation. All rights reserved. Printed in the USA.

Aero, ALPHA, ALPHA AXP, AlphaServer, AlphaStation, Armada, BackPak, COMPAQ, Compaq Insight Manager, CompaqCare logo, Counselor, DECterm, Deskpro, DIGITAL, DIGITAL logo, DIGITAL Alpha Systems, Digital Equipment Corporation, DIGITAL UNIX, DirectPlus, FASTART, Himalaya, InfoPak, Integrity, LicensePak, Ministation, NetFlex, NonStop, OpenVMS, PaqFax, Presario, ProLiant, ProLinea, ProSignia, QuickBack, QuickFind, Qvision, RDF, RemotePak, RomPak, ServerNet, SERVICenter, SmartQ, SmartStart, SmartStation, SolutionPak, SpeedPak, StorageWorks, Systempro/LT, Tandem, TechPak, TruCluster, Tru64 UNIX, registered in United States Patent and Trademark Office.

Atalla, C-Series, Expand, FOX, Guardian, iTP, Measure, Netelligent, and PointView are trademarks of Compaq Computer Corporation.

Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation.

Intel, Pentium, and Intel Inside are registered trademarks and Xeon is a trademark of Intel Corporation.

UNIX is a trademark in the US and other countries, licensed exclusively through X-Open Company Ltd.

TCP/IP Services for OpenVMS

Lab Guide

December 1999

Module 1 — Lab Exercises

Network Overview — Module 1	L1-1
Configuration — Module 3	L1-2
TCP/IP Commands — Module 2	L1-3
Protocols — Module 4	L1-4
NFS — Module 5	L1-5
Serial Line and Static Routing — Modules 6 and 7	L1-6
Subnet Routing — Module 7	L1-7
BIND — Module 8	L1-8
Dynamic Routing — Module 7	L1-9
BOOTP and DHCP — Modules 9 and 10	L1-10
NTP — Module 11	L1-11
SNMP — Module 12	L1-12
Remote Printing — Module 13	L1-13
Troubleshooting — Module 14	L1-14
SMTP — Module 15	L1-15

Module 2 — Lab Solutions

Network Overview — Module 1	L2-1
Configuration — Module 3	L2-6
TCP/IP Commands — Module 2	L2-12
Protocols — Module 4	L2-32
NFS — Module 5	L2-39
Serial Line and Static Routing — Modules 6 and 7	L2-51
Subnet Routing — Module 7	L2-52
BIND — Module 8	L2-53
Dynamic Routing — Module 7	L2-60
BOOTP and DHCP — Modules 9 and 10	L2-64
NTP — Module 11	L2-66
SNMP — Module 12	L2-68
Remote Printing — Module 13	L2-71
Troubleshooting — Module 14	L2-76
SMTP — Module 15	L2-86

Network Overview — Module 1

1. Determine the IP address and subnet mask of your machine. If your machine is not configured yet, read the rest of this lab and then start the configuration labs.
2. Can you ping other lab machines?
3. Briefly describe the concept of a network protocol.
4. What are port numbers used for?
5. What is the difference between a MAC address and an IP address?
6. Describe the TCP/IP protocol stack.
7. Given a network address of 192.206.126, what would be a reasonable subnet mask to choose if you needed to set up five subnets? How many nodes would be configurable on each subnet?
8. Fill in the blanks:

Network Address	Subnet Mask	Configurable Subnet Count	Configurable Node Count
192.206.126	255.255.255.128	_____	_____
192.206.126	255.255.255.248	_____	_____
192.206.126		7 (minimum)	28 (maximum)
16	255.224.0.0	_____	_____
180.4	255.255.255.128	_____	_____

Fill in the blanks:

IP Address	Subnet Mask	Subnet Number	Node Number
192.206.126.140	255.255.255.128	_____	_____
150.32.0.85	255.255.255.0	_____	_____
28.231.17.6	255.224.0.0	_____	_____
192.206.126._____	255.255.255.240	11	9
150.32._____._____	255.255.255.128	503	41
16._____._____._____	255.254.0.0	99	8,321

For extra credit:

IP Address	Subnet Mask	Subnet Number	Node Number
192.206.126.185	255.255.255.144	_____	_____

Configuration — Module 3

1. If necessary, install TCP/IP Services for OpenVMS and the license PAK.
2. Configure your interface using @SYS\$MANAGER:TCPIP\$CONFIG. Use the IP addresses and subnet masks provided by the instructor.
3. Try to ping a neighboring machine using its IP address. Ping using its host name. If the host name ping fails, examine existing names in the HOSTS database and add appropriate names.
4. Be sure to configure FTP, RSH, REXEC, RLOGIN, and Telnet. Consider both server and client configuration issues for these options.
5. Start the network. Ping a neighbor. Stop the network. Ping the neighbor. Note any error messages. Start the network again.
6. Make a change to SYS\$MANAGER:SYSTARTUP_VMS.COM providing for automatic startup of the network upon boot.
7. Make a change to SYS\$MANAGER:SYSHUTDOWN.COM providing for automatic shutdown of the network.
8. Make sure that you can reach all nodes in the lab environment.

TCP/IP Commands — Module 2

1. Determine what TCPIP\$ logical names were created during the configuration labs.
2. Determine what services are available on your machine.
3. Determine if any user accounts were created in SYSUAF.DAT during the configuration.
4. Research and experiment with options on the ping command. If a UNIX system is available, try `man ping`.
5. Use Telnet to log in to a neighbor's system. Try the escape (^]) from your Telnet session. What options are available at the TELNET> prompt?
6. Make a directory called [TCPTEST]. Populate it with several files. Have a partner in class do the same. Use the RCP command to copy the files from one system to the other. Use proxies to avoid having to include a user name and password on the command line. If a UNIX system is available, set up a trust relationship using /etc/hosts.equiv.
7. Use RLOGIN to access a remote system.
8. Use RSH to get a directory listing on a remote node. If available, do this on Tru64 UNIX and OpenVMS.
9. Use FTP to copy the same files as mentioned in step 6.
10. Try the new UNIX style commands.
 - a. Display the arp table.
 - b. Display configurable interfaces.
 - c. Display the routing table.
 - d. Display network statistics.
 - e. Display configured subsystems.
 - f. Display attributes within a configured subsystem.
11. Extra credit: Prepare your system for *anonymous ftp*.

Protocols — Module 4

1. Run TCPTRACE while pinging a neighbor. Run TCPTRACE/FULL while pinging. What is the difference?

Run TCPTRACE during a Telnet session. How often is Telnet sending a packet over the wire?
2. Can you change the behavior of Telnet so that it sends one message per line?
3. Use TCPTRACE to confirm the port numbers and protocols used by FTP.
4. Does Telnet use the TOS field in the IP header? Does FTP?

NFS — Module 5

1. What are the advantages of NFS? Disadvantages?
2. With a partner, configure one node as an NFS server and the other as a client. Create a directory named [NFS_OUT] and populate it with several files.
 - a. Export the directory on the server.
 - b. Mount the served directory on the client and get access to the files.
 - c. Create a file within the served directory while logged in to the client.
 - d. Check file ownership and permissions.
 - e. Can a user logged in to the server get access to the file created while logged in to the client?
 - f. What happens if one user on the client and one user on the server (or another client) try to access the served file at the same time?
3. Explain the function of NFS proxy accounts. Set some up and use them to access the served files.

What is the difference between an NFS proxy and a communication proxy?
4. Create a CFS on an OpenVMS server system. Mount and access it from the client.
5. Turn off the PORTMAP service on the server node. What happens when the client tries to mount?
6. If your server is a UNIX system, alter some file characteristics (using SET FILE) and see how the UNIX NFS server handles it.

Serial Line and Static Routing — Modules 6 and 7

1. Select several systems to function as static routers (do not enable GATED or ROUTED for this lab). For example, if the room consists of four tables with three machines on each, choose one machine on each table to be the router node. Two of the tables need two routers — they will have serial line connections to two different tables.
2. Configure SLIP lines (no modems) connecting the routers on each table. Choose new network numbers for each table and for each serial line pair.

For example:

Table	All Nodes Ethernet Interface	Router 1 SLIP1	Router 2 SLIP2
Table 1	192.206.126	192.206.130	N.A.
Table 2	192.206.127	192.206.130	192.206.131
Table 3	192.206.128	192.206.131	192.206.132
Table 4	192.206.129	192.206.132	N.A.

As a result, each table is logically on its own Ethernet segment. (It is not necessary to do this physically; the software will think it cannot get to table 2 if it is on a different network.)

The goal of this lab is to try to get all nodes pinging (including the ability to ping the IP address of the SLIP interface).

Note that the selected router nodes now have two IP addresses. Each node with a SLIP interface configured should also be configured as a router.

This lab forces many trips to the routing tables (`TCPIP SHOW ROUTE` or `netstat -nr`) for purposes of troubleshooting.

Be prepared to do some TCPTRACE activities if questions arise. Determine the address that the sending node presents. Be aware that getting a packet to a destination is only half the battle; it has to make it back as well!

Subnet Routing — Module 7

**Note**

Do not proceed to this lab if you have not completed the previous lab.

This second routing lab is the same as the previous one except that the nodes are all using the same network ID (192.206.126), but are configured in separate subnets.

Based on your room configuration, the actual number of subnets may vary.

Given the room configuration from the previous lab, choose a reasonable subnet mask. All interfaces will use the same subnet mask.

Set up the tables as follows:

Table	All Nodes Ethernet Interface	Router 1 SLIP1	Router 2 SLIP2
Table 1	Subnet 1	Subnet 5	N.A.
Table 2	Subnet 2	Subnet 5	Subnet 6
Table 3	Subnet 3	Subnet 6	Subnet 7
Table 4	Subnet 4	Subnet 7	N.A.

Again, the goal is to be able to ping **all** interfaces on **all** nodes using both IP addresses and node names.

BIND — Module 8

1. Configure the room with one master server, two slave servers, and the rest as clients. Make sure the master server has all node name and address information that you want to serve.
2. Configure the clients to point to both the master and slave for their BIND service. Remove several host names from the client's local database. Try to ping using the removed host name.
3. Troubleshoot if necessary using NSLOOKUP.
Check Appendix B to see what commands are available within NSLOOKUP.
4. Use TCPTRACE to follow the BIND communications. Start a zone transfer while a trace is in progress.
How would you prevent attempts to contact the Internet root servers?

Dynamic Routing — Module 7

**Note**

Do not proceed to this lab if you have not completed the subnet routing lab.

1. Keep the room configured with subnetting. Configure the ROUTED daemon on all nodes (not just the routers). Have ROUTED configured on the routers so that it advertises its routes. Have ROUTED configured on the end nodes in silent mode.
2. Remove a few static routes from your routing tables. See if the dynamic routing replaces them with useful information.
3. Use TCPTRACE to see if the ROUTED daemons on the routers are actually sending out their 30-second updates.

BOOTP and DHCP — Modules 9 and 10



Note

The instructor will supply details for a BOOTP lab based on system availability and class interest.

1. Isolate a few target systems and delete their network interfaces.
2. Configure DHCP on a server and set it up with a pool of five addresses (192.206.126.80 – 192.206.126.90).
Use the GUI (dhcpgui) to set up the pool.
3. Check the contents of the NAMEPOOL file to see the result. Try to get a client to recognize its IP address presented from the DHCP server.

NTP — Module 11

1. Select and configure a system as an NTP stratum 1 server. Select and configure a second system as an NTP stratum 2 server.
2. Configure all other machines as NTP client (peer) machines.
3. Check synchronization of times.
4. Modify the time on one system so that it is out of synchronization by two minutes. Check to see if it gets resynchronized.
5. Try using `ntptrace` and `ntpq` commands.

SNMP — Module 12

1. Configure all systems as SNMP agents.
2. Review the MIB material and experiment with the `snmp_request` command.
3. Try the *set* and the *get* capabilities within the `snmp_request` command.
4. Look under SYS\$EXAMPLES for any .MY files. Use the information in the .MY files to support your experiments.

Remote Printing — Module 13

1. If a printer is available, set up the printer locally off one machine using `lprsetup`.
2. Configure that machine as a print server.
3. Configure all other machines as clients, to enable access to the remote print queue, using `lprsetup`.
4. Check the `SYSS$SPECIFIC:[TCPIP$LPD]TCPIP$PRINTCAP.DAT` file.
5. How would you prevent users on node `alfman` from accessing the LPD service?

Troubleshooting — Module 14

1. Try normal activities with `REPLY/ENABLE=NETWORK`.
2. Try using the `TCPIP ANALYZE SERVICE` command.
3. Using FTP, copy a large file. Write down the time statistics. Check all FTP logical names. Research and use logical names `TCPIP$FTP_WNDSIZE`, `TCPIP$FTP_FILE_ALQ`, and `TCPIP$FTP_FILE_DEQ`. Copy the file again using FTP and see if it makes a difference.
4. Use Telnet to confirm that FTP, SMTP, or SNMP is functioning on the remote node.
5. What columns of a `netstat -i` command might indicate network problems?
6. What columns of the `netstat -rn` output are useful in troubleshooting?

SMTP — Module 15

1. Configure SMTP and send some mail messages (using SMTP).
2. Use TCPTRACE to watch communications during mail activities.

Network Overview — Module 1

1. Determine the IP address and subnet mask of your machine. If your machine is not configured yet, read through the rest of this lab and then start the configuration labs.

\$ **tcpip**

TCPIP> **sho inter**

Interface	IP_Addr	Network mask	Packets		MTU
			Receive	Send	
WE0	192.206.126.27	255.255.255.0	95	49	1500
SL0	192.206.127.27	255.255.255.0	0	64	1006
L00	127.0.0.1	255.0.0.0	9	9	0

TCPIP>

TCPIP>

TCPIP> **sho inter we0**

Interface	IP_Addr	Network mask	Packets		MTU
			Receive	Send	
WE0	192.206.126.27	255.255.255.0	95	49	1500

TCPIP>

TCPIP>

TCPIP> **sho inter we0/full**

Interface: WE0

IP_Addr: 192.206.126.27 NETWRK: 255.255.255.0 BRDCST: 192.206.126.255

Ethernet_Addr: AA-00-04-00-01-04 MTU: 1500

Flags: None

	RECEIVE	SEND
Packets	96	49
Errors	0	0
Collisions:	0	

TCPIP>

TCPIP> **ifconfig -a**

TN0: flags=80<NOARP>

WE0: flags=c43<UP, BROADCAST, RUNNING, MULTICAST, SIMPLEX

 Hwaddr aa-00-04-00-01-04

 inet 192.206.126.27 netmask ffffffff broadcast 192.206.126.255 ipmtu 1500

SL0: flags=11<UP, POINTOPOINT>

inet 192.206.127.27 --> 0.0.0.0 netmask ffffffff ipmtu 1006

 L00: flags=100c89<UP, LOOPBACK, NOARP, MULTICAST, SIMPLEX, NOCHECKSUM>

 inet 127.0.0.1 netmask ff000000 ipmtu 4096

TCPIP>

2. Can you ping other lab machines?

If not, move on to the configuration labs.

TCPIP> **ping 192.206.126.3**

PING 192.206.126.3 (192.206.126.3): 56 data bytes

64 bytes from 192.206.126.3: icmp_seq=0 ttl=64 time=1 ms

64 bytes from 192.206.126.3: icmp_seq=1 ttl=64 time=1 ms

----192.206.126.3 PING Statistics----

2 packets transmitted, 2 packets received, 0% packet loss

round-trip (ms) min/avg/max = 1/1/1 ms

TCPIP> **ping alfman**

PING ALFMAN (192.206.126.3): 56 data bytes

64 bytes from 192.206.126.3: icmp_seq=0 ttl=64 time=1 ms

64 bytes from 192.206.126.3: icmp_seq=1 ttl=64 time=1 ms

64 bytes from 192.206.126.3: icmp_seq=2 ttl=64 time=1 ms

----ALFMAN PING Statistics----

3 packets transmitted, 3 packets received, 0% packet loss

round-trip (ms) min/avg/max = 1/1/1 ms

TCPIP>

3. Briefly describe the concept of a network protocol.

A set of rules for communication over a network.

4. What are port numbers used for?

To identify an application.

TCPIP> **sho serv**

Service	Port	Proto	Process	Address	State
BIND	53	TCP,UDP	TCPIP\$BIND	0.0.0.0	Disabled
BOOTP	67	UDP	TCPIP\$BOOTP	0.0.0.0	Enabled
ESNMP	242	UDP	ESNMP	0.0.0.0	Disabled
FINGER	79	TCP	TCPIP\$FINGER	0.0.0.0	Enabled
FTP	21	TCP	TCPIP\$FTP	0.0.0.0	Enabled
LBROKER	6570	UDP	TCPIP\$LBROKER	0.0.0.0	Enabled
LPD	515	TCP	TCPIP\$LPD	0.0.0.0	Enabled
MOUNT	10	UDP	TCPIP\$NFS_M	0.0.0.0	Enabled
NFS	2049	UDP	TCPIP\$NFS	0.0.0.0	Enabled
NTP	123	UDP	TCPIP\$NTP	0.0.0.0	Enabled
POP	110	TCP	TCPIP\$POP	0.0.0.0	Enabled
PORTMAPPER	111	TCP,UDP	TCPIP\$PORTM	0.0.0.0	Enabled
REXEC	512	TCP	TCPIP\$REXEC	0.0.0.0	Enabled
RLOGIN	513	TCP	not defined	0.0.0.0	Enabled
RSH	514	TCP	TCPIP\$RSH	0.0.0.0	Enabled
SMTP	25	TCP	TCPIP\$SMTP	0.0.0.0	Enabled
TELNET	23	TCP	not defined	0.0.0.0	Enabled
TFTP	69	UDP	TCPIP\$TFTP	0.0.0.0	Enabled

TCPIP>

5. What is the difference between a MAC address and an IP address?

MAC identifies an interface at the data link level. IP identifies an interface at the network level.

TCP/IP> **arp -a**

COMPCOM (192.206.126.80) at 00-40-d0-08-a2-b3 stale

ALFMAN (192.206.126.3) at 08-00-2b-39-80-cb

TCP/IP>

TCP/IP>

TCP/IP> **sho arp**

Address Resolution Mapping Table: 37 Buckets 16 Deep 36 Bytes

Cnt

Flags	Timer	Host	Phys Addr
1: UCS	1170	192.206.126.80	00-40-d0-08-a2-b3
2: UC	45	192.206.126.3	08-00-2b-39-80-cb

IPv6 Neighbor Cache Entries:

Cnt

Flags	Timer	Host	Phys Addr	Interface
-------	-------	------	-----------	-----------

TCP/IP>

TCP/IP>

TCP/IP>

TCP/IP> **sho arp alfman**

ALFMAN (192.206.126.3) at

08-00-2b-39-80-cb

6. Describe the TCP/IP protocol stack.

A four-layer arrangement where the software modules are “stacked” one atop the other to modularize the code. The four layers are application level, transport level, network level, and data link (or network interface) level.

7. Given a network address of 192.206.126, what would be a reasonable subnet mask to choose if you needed to set up five subnets?

255.255.255.224

How many nodes would be configurable on each subnet?

Subnetting uses 3 bits, so the remaining 5 bits can be used for node addressing (32-2=30).

8. Fill in the blanks:

Net Address	Subnet Mask	Configurable Subnet Count	Configurable Node Count
192.206.126	255.255.255.128	<u>2</u>	<u>126</u>
192.206.126	255.255.255.248	<u>32</u>	<u>6</u>
192.206.126	255.255.255.224	7 (minimum)	28 (maximum)
16	255.224.0.0	<u>8</u>	<u>2,097,150</u>
180.4	255.255.255.128	<u>512</u>	<u>126</u>

Fill in the blanks:

IP Address	Subnet Mask	Subnet Number	Node Number
192.206.126.140	255.255.255.128	<u>1</u>	<u>12</u>
150.32.0.85	255.255.255.0	<u>0</u>	<u>85</u>
28.231.17.6	255.224.0.0	<u>7</u>	<u>463,110</u>
192.206.126. <u>185</u>	255.255.255.240	11	9
150.32. <u>252.169</u>	255.255.255.128	503	41
16. <u>204.32.129</u>	255.254.0.0	99	8,321

For extra credit:

IP Address	Subnet Mask	Subnet Number	Node Number
192.206.126.185	255.255.255.144	<u>3</u>	<u>25</u>

Configuration — Module 3

1. If necessary, install TCP/IP Services for OpenVMS and the license PAK.

```
$ LICENSE LIST UCX /FULL
```

```
$ product show product tcpip/full
```

```
-----  
PRODUCT                                KIT TYPE    STATE      MAINTENANCE  
REFERENCED BY  
-----
```

```
DEC AXPVMS TCPIP V5.0-9                Full LP     Installed  
DEC AXPVMS OPENVMS V7.2
```

```
-----  
1 item found
```

2. Configure your interface using @SYSS\$MANAGER:TCPIP\$CONFIG. Use the IP addresses and subnet masks provided by the instructor.

```
$ @sys$manager:tcpip$config
```

TCP/IP Network Configuration Procedure

This procedure helps you define the parameters required to run DIGITAL TCP/IP Services for OpenVMS on this system.
Checking TCP/IP Services for OpenVMS configuration database files.

No new database files were created.

DIGITAL TCP/IP Services for OpenVMS requires a definition for at least one interface. There are no interfaces defined on this system.
Please select the Interface option from the Core Environment Menu.

DIGITAL TCP/IP Services for OpenVMS Configuration Menu

Configuration options:

- 1 - Core environment
- 2 - Client components
- 3 - Server components
- 4 - Optional components
- 5 - Shutdown DIGITAL TCP/IP Services for OpenVMS
- 6 - Startup DIGITAL TCP/IP Services for OpenVMS
- 7 - Run tests

A - Configure options 1 - 4

[E] - Exit configuration procedure

Enter configuration option: a

3. Try to ping a neighboring machine using its IP address. Ping using its host name. If the host name ping fails, examine existing names in the HOSTS database and add appropriate names.

\$ tcpip ping 192.206.126.3

```

PING 192.206.126.3 (192.206.126.3): 56 data bytes
64 bytes from 192.206.126.3: icmp_seq=0 ttl=64 time=2 ms
64 bytes from 192.206.126.3: icmp_seq=1 ttl=64 time=1 ms
----192.206.126.3 PING Statistics----
2 packets transmitted, 2 packets received, 0% packet loss

```

round-trip (ms) min/avg/max = 1/1/2 ms

4. Be sure to configure FTP, RSH, REXEC, RLOGIN, and Telnet. Consider both the server and client configuration issues for these options.

TCPIP> sho serv ftp

Service	Port	Proto	Process	Address	State
FTP	21	TCP	TCPIP\$FTP	0.0.0.0	Enabled

TCPIP> sho serv ftp/full

Service: FTP

		State:	Enabled
Port:	21	Protocol:	TCP
Inactivity:	5	User_name:	TCPIP\$FTP
Limit:	10	Active:	0
		Peak:	0

File: TCPIP\$SYSTEM:TCPIP\$FTP_RUN.COM

Flags: None

Socket Opts: Rcheck Scheck

Receive:	0	Send:	0
----------	---	-------	---

Log Opts: Acpt Actv Dactv Conn Error Exit Logi Logo Mdfy Rjct Tim0 Addr
 File: SYSS\$SYSDEVICE:[TCPIP\$FTP]TCPIP\$FTP_RUN.LOG

Security

Reject msg: not defined
 Accept host: 0.0.0.0
 Accept netw: 0.0.0.0

TCPIP>

TCPIP>

TCPIP>

TCPIP> **sho serv rsh /full**

Service: RSH

		State:	Enabled	
Port:	514	Protocol:	TCP	Address: 0.0.0.0
Inactivity:	5	User_name:	not defined	Process: TCPIP\$RSH
Limit:	3	Active:	0	Peak: 0

File: TCPIP\$SYSTEM:TCPIP\$RSH_RUN.COM

Flags: Case Listen Proxy Rexe

Socket Opts: Rcheck Scheck

Receive: 0 Send: 0

Log Opts: Acpt Actv Dactv Conn Error Exit Mdfy Rjct Tim0 Addr

File: not defined

Separators:

Port: 0 User_name: 0 Password: 0 Command: 0

Security

Reject msg: not defined
 Accept host: 0.0.0.0
 Accept netw: 0.0.0.0

TCPIP>

TCPIP> **sho serv rexec /full**

Service: REXEC

		State:	Enabled	
Port:	512	Protocol:	TCP	Address: 0.0.0.0
Inactivity:	5	User_name:	not defined	Process: TCPIP\$REXEC
Limit:	3	Active:	0	Peak: 0

File: TCPIP\$SYSTEM:TCPIP\$REXEC_RUN.COM

Flags: Listen Rexe

Socket Opts: Rcheck Scheck

Receive: 0 Send: 0

Log Opts: Acpt Actv Dactv Error Exit Mdfy Rjct Tim0 Addr

File: not defined

Separators:

Port: 0 User_name: 0 Password: 0 Command: 0

Security

Reject msg: not defined

Accept host: 0.0.0.0

TCPIP>

TCPIP>

TCPIP> **sho serv rlogin /full**

Service: RLOGIN

		State:	Enabled	
Port:	513	Protocol:	TCP	Address: 0.0.0.0
Inactivity:	1	User_name:	not defined	Process: not defined
Limit:	50	Active:	0	Peak: 0

File: not defined
 Flags: Case Listen Proxy Rlog Rtty
 Socket Opts: Keepalive Rcheck Scheck
 Receive: 3000 Send: 3000

Log Opts: Acpt Actv Dactv Conn Error Logi Logo Mdfy Rjct Tim0
 File: not defined

Security

Reject msg: TCPIP RLOGIN Connection refused
 Accept host: 0.0.0.0
 Accept netw: 0.0.0.0

TCPIP>

TCPIP>

TCPIP> **sho serv telnet /full**

Service: TELNET

		State:	Enabled	
Port:	23	Protocol:	TCP	Address: 0.0.0.0
Inactivity:	1	User_name:	not defined	Process: not defined
Limit:	50	Active:	0	Peak: 0

File: not defined
 Flags: Listen Rtty

Socket Opts: Keepalive Rcheck Scheck
 Receive: 3000 Send: 3000

Log Opts: Actv Dactv Conn Error Logi Logo Mdfy Rjct
 File: not defined

Security

Reject msg: not defined
 Accept host: 0.0.0.0
 Accept netw: 0.0.0.0

TCPIP>

5. Start the network. Ping a neighbor. Stop the network. Ping the neighbor. Note any error messages. Start the network again.

```
$ @sys$startup:tcpip$shutdown
```

```
(...)
```

```
$ tcpip ping 192.206.126.3
```

```
ping: socket: no such device or address
```

```
$
```

```
$
```

```
$ @sys$startup:tcpip$startup
```

```
(...)
```

```
$
```

```
$
```

```
$ tcpip ping 192.206.126.3
```

```
PING 192.206.126.3 (192.206.126.3): 56 data bytes
```

```
64 bytes from 192.206.126.3: icmp_seq=0 ttl=64 time=2 ms
```

```
64 bytes from 192.206.126.3: icmp_seq=1 ttl=64 time=1 ms
```

```
----192.206.126.3 PING Statistics----
```

```
2 packets transmitted, 2 packets received, 0% packet loss
```

```
round-trip (ms) min/avg/max = 1/1/2 ms
```

6. Make a change to SYSS\$MANAGER:SYSTARTUP_VMS.COM providing for automatic startup of the network upon boot.

EDIT SYSTARTUP_VMS.COM and add **@SYSS\$STARTUP:TCPIP\$STARTUP.**

7. Make a change to SYSS\$MANAGER:SYSHUTDOWN.COM providing for automatic shutdown of the network.

EDIT SYSTARTUP_VMS.COM and add **@SYSS\$STARTUP:TCPIP\$SHUTDOWN.**

8. Make sure that you can reach all nodes in the lab environment.

Ping the broadcast address (192.206.126.255, for example).

TCP/IP Commands — Module 2

1. Determine what TCPIP\$ logical names were created during the configuration labs.

```
$ sho logical *tcpip*
```

```
(LNM$PROCESS_TABLE)
```

```
(LNM$JOB_80D704C0)
```

```
(LNM$GROUP_000001)
```

```
(LNM$SYSTEM_TABLE)
```

```
"TCPIP$BIND_DOMAIN" = "bruden.com"
"TCPIP$BIND_RETRY" = "...."
"TCPIP$BIND_STATE" = "....."
"TCPIP$BIND_TIMEOUT" = "...."
"TCPIP$BIND_TRANSPORT" = "UDP"
"TCPIP$BOOTP" = "SYS$COMMON:[SYSEXE]TCPIP$BOOTP.DAT;1"
"TCPIP$CFS_ATTR_CACHE_TIMEOUT" = "5"
"TCPIP$CFS_CACHE_HIGH_LIMIT" = "8"
"TCPIP$CFS_CACHE_LOW_LIMIT" = "4"
"TCPIP$CFS_CACHE_SIZE" = "256"
"TCPIP$CFS_FATAL_MESSAGES" = "_OPA0:"
"TCPIP$CFS_KEEP_ALLOC" = "1"
"TCPIP$CFS_MODUS_OPERANDI" = "0"
"TCPIP$CFS_SHOW_VERSION" = "0"
"TCPIP$CFS_TRANSFERSIZE" = "8192"
"TCPIP$CFS_WRITESIZE" = "512"
"TCPIP$CONFIGURATION" = "SYS$COMMON:[SYSEXE]TCPIP$CONFIGURATION.DAT;1"
"TCPIP$DEVICE" = "BG0:"
"TCPIP$EXAMPLES" = "SYS$COMMON:[SYSHLP,EXAMPLES.TCPIP]"
"TCPIP$EXPORT" = "SYS$COMMON:[SYSEXE]TCPIP$EXPORT.DAT;1"
```

```
"TCPIP$FTP_EXTLOG" = "1"
"TCPIP$HOST" = "SYS$COMMON:[SYSEXE]TCPIP$HOST.DAT;1"
"TCPIP$INET_DOMAIN" = "bruden.com"
"TCPIP$INET_HOST" = "den255"
"TCPIP$INET_HOSTADDR" = "192.206.126.27"
"TCPIP$LIBRARY" = "SYS$SYSROOT:[TCPIP$LIB]"
"TCPIP$LPD_LOGFILE" = "SYS$SPECIFIC:[TCPIP$LPD]TCPIP$LPD_LOGFILE.LOG"
"TCPIP$LPD_PRINTCAP" = "SYS$SPECIFIC:[TCPIP$LPD]TCPIP$PRINTCAP.DAT"
"TCPIP$LPD_SPOOL" = "SYS$SPECIFIC:[TCPIP$LPD]"
"TCPIP$NETWORK" = "SYS$COMMON:[SYSEXE]TCPIP$NETWORK.DAT;1"
"TCPIP$NFS00000000_DEBUG" = "0"
"TCPIP$NFS00000000_ERROR" = "1"
"TCPIP$NFS00000000_GID" = "-2"
"TCPIP$NFS00000000_HOSTS" = "20"
"TCPIP$NFS00000000_INACTIVITY" = "02:00"
"TCPIP$NFS00000000_OPCOM" = "1"
"TCPIP$NFS00000000_SECURITY" = "0"
"TCPIP$NFS00000000_THREADS" = "20"
"TCPIP$NFS00000000_UID" = "-2"
"TCPIP$NFS00000000_XID" = "20"
"TCPIP$NTP_CONF" = "SYS$SPECIFIC:[TCPIP$NTP]TCPIP$NTP.CONF"
"TCPIP$NTP_ENABLE" = ".1.."
"TCPIP$NTP_RES_CHILD" = "SYS$COMMON:[SYSEXE]TCPIP$NTP_RES_CHILD.EXE"
"TCPIP$NTP_TEMPFILE" = "SYS$SPECIFIC:[TCPIP$NTP]NTPXXXXXX"
"TCPIP$PROXY" = "SYS$COMMON:[SYSEXE]TCPIP$PROXY.DAT;1"
"TCPIP$RMT_ENABLE" = ".1.."
"TCPIP$ROUTE" = "SYS$COMMON:[SYSEXE]TCPIP$ROUTE.DAT;1"
"TCPIP$RPC" = "SYS$SYSROOT:[TCPIP$LIB.RPC]"
"TCPIP$SERVICE" = "SYS$COMMON:[SYSEXE]TCPIP$SERVICE.DAT;1"
"TCPIP$SMTP_COMMON" = "SYS$SPECIFIC:[TCPIP$SMTP]"
"TCPIP$SNMP" = "SYS$SYSROOT:[TCPIP$LIB.SNMP]"
"TCPIP$SYSTEM" = "SYS$COMMON:[SYSEXE]"
"TCPIP$TELNETSYM_SCRATCH" = "TCPIP$LPD_SPOOL"
"TCPIP$TFTP_ROOT" = "SYS$SYSDEVICE:[TCPIP$TFTP_ROOT.]"
```

2. Determine what services are available on your machine.

\$ tcpip show serv

Service	Port	Proto	Process	Address	State
BIND	53	TCP,UDP	TCPIP\$BIND	0.0.0.0	Disabled
BOOTP	67	UDP	TCPIP\$BOOTP	0.0.0.0	Enabled
ESNMP	242	UDP	ESNMP	0.0.0.0	Disabled
FINGER	79	TCP	TCPIP\$FINGER	0.0.0.0	Enabled
FTP	21	TCP	TCPIP\$FTP	0.0.0.0	Enabled
LBROKER	6570	UDP	TCPIP\$LBROKER	0.0.0.0	Enabled
LPD	515	TCP	TCPIP\$LPD	0.0.0.0	Enabled
MOUNT	10	UDP	TCPIP\$NFS_M	0.0.0.0	Enabled
NFS	2049	UDP	TCPIP\$NFS	0.0.0.0	Enabled
NTP	123	UDP	TCPIP\$NTP	0.0.0.0	Enabled
POP	110	TCP	TCPIP\$POP	0.0.0.0	Enabled
PORTMAPPER	111	TCP,UDP	TCPIP\$PORTM	0.0.0.0	Enabled
REXEC	512	TCP	TCPIP\$REXEC	0.0.0.0	Enabled
RLOGIN	513	TCP	not defined	0.0.0.0	Enabled
RSH	514	TCP	TCPIP\$RSH	0.0.0.0	Enabled
SMTP	25	TCP	TCPIP\$SMTP	0.0.0.0	Enabled
TELNET	23	TCP	not defined	0.0.0.0	Enabled
TFTP	69	UDP	TCPIP\$TFTP	0.0.0.0	Enabled

- Determine if any user accounts were created in SYSUAF.DAT during the configuration.

```
$ set def sys$system
```

```
$ run authorize
```

```
UAF> sho */brief
```

Owner	Username	UIC	Account	Privs	Pri	Directory
(...)						
TCPIP\$BIND	TCPIP\$BIND	[3655,3]	TCPIP	Normal	8	SYS\$SPECIFIC:[TCPIP\$BIND]
TCPIP\$BOOTP	TCPIP\$BOOTP	[3655,20]	TCPIP	Normal	8	SYS\$SYSDEVICE:[TCPIP\$BOOTP]
TCPIP\$DHCP	TCPIP\$DHCP	[3655,17]	TCPIP	Normal	8	SYS\$SYSDEVICE:[TCPIP\$DHCP]
TCPIP\$FINGER	TCPIP\$FINGER	[3655,15]	TCPIP	Normal	8	SYS\$SYSDEVICE:[TCPIP\$FINGER]
TCPIP\$FTP	TCPIP\$FTP	[3655,1]	TCPIP	Normal	8	SYS\$SYSDEVICE:[TCPIP\$FTP]
TCPIP\$LD_BKR	TCPIP\$LD_BKR	[3655,16]	TCPIP	Normal	8	SYS\$SYSDEVICE:[TCPIP\$LD_BKR]
TCPIP\$LPD	TCPIP\$LPD	[3655,2]	TCPIP	Normal	8	SYS\$SPECIFIC:[TCPIP\$LPD]
TCPIP\$METRIC	TCPIP\$METRIC	[3655,13]	TCPIP	Normal	8	SYS\$SPECIFIC:[TCPIP\$METRIC]
TCPIP\$NFS	TCPIP\$NFS	[3655,7]	TCPIP	Normal	8	SYS\$SYSDEVICE:[TCPIP\$NFS]
TCPIP\$NOBODY	TCPIP\$NOBODY	[3375,3]	TCPIP	Normal	8	SYS\$SYSDEVICE:[TCPIP\$NOBODY]
TCPIP\$NTP	TCPIP\$NTP	[3655,12]	TCPIP	Normal	8	SYS\$SPECIFIC:[TCPIP\$NTP]
TCPIP\$POP	TCPIP\$POP	[3655,14]	TCPIP	Normal	8	SYS\$SYSDEVICE:[TCPIP\$POP]
TCPIP\$PORTM	TCPIP\$PORTM	[3655,10]	TCPIP	Normal	8	SYS\$SYSDEVICE:[TCPIP\$PORTM]
TCPIP\$REXEC	TCPIP\$REXEC	[3655,5]	TCPIP	Normal	8	SYS\$SYSDEVICE:[TCPIP\$REXEC]
TCPIP\$RSH	TCPIP\$RSH	[3655,4]	TCPIP	Normal	8	SYS\$SYSDEVICE:[TCPIP\$RSH]
TCPIP\$SMTP	TCPIP\$SMTP	[3655,6]	TCPIP	Normal	8	SYS\$SPECIFIC:[TCPIP\$SMTP]
TCPIP\$SNMP	TCPIP\$SNMP	[3655,11]	TCPIP	Normal	8	SYS\$SYSDEVICE:[TCPIP\$SNMP]
TCPIP\$TFTP	TCPIP\$TFTP	[3655,21]	TCPIP	Normal	8	SYS\$SYSDEVICE:[TCPIP\$TFTP]

```
UAF>
```

```
UAF>
```

```
UAF> exit
```

```
%UAF-I-NOMODS, no modifications made to system authorization file
```

```
%UAF-I-RDBNOMODS, no modifications made to rights database
```

```
$
```

4. Research and experiment with options on the ping command. If a UNIX system is available, try `man ping`.

\$ tcpip

TCPIP> **h ping**

PING

Sends ICMP ECHO packets to hosts to determine if they are active.

Same as the LOOP command.

Format

```
PING [ host ]
      [ /ALL ]
      [ /NUMBER_PACKETS=n ]
      [ /PACKET_SIZE=n ]
      [ /[NO]ROUTE ]
```

(...)

TCPIP> **ping "-s 512 192.206.126.3"**

PING 192.206.126.3 (192.206.126.3): 512 data bytes

520 bytes from 192.206.126.3: icmp_seq=0 ttl=64 time=3 ms

520 bytes from 192.206.126.3: icmp_seq=1 ttl=64 time=2 ms

520 bytes from 192.206.126.3: icmp_seq=2 ttl=64 time=2 ms

520 bytes from 192.206.126.3: icmp_seq=3 ttl=64 time=2 ms

520 bytes from 192.206.126.3: icmp_seq=4 ttl=64 time=2 ms

----192.206.126.3 PING Statistics----

5 packets transmitted, 5 packets received, 0% packet loss

round-trip (ms) min/avg/max = 2/2/3 ms

\$

\$ RSH alfman man ping

5. Use Telnet to log in to a neighbor's system. Try the escape (^]) from your Telnet session. What options are available at the TELNET> prompt?

```
$ telnet alfman
```

```
%TELNET-I-TRYING, Trying ... 192.206.126.3
```

```
%TELNET-I-SESSION, Session 01, host alfman, port 23
```

```
-TELNET-I-ESCAPE, Escape character is ^]
```

```
Digital UNIX (alfman) (ttyp4)
```

```
login: root
```

```
Password:
```

```
(...)
```

```
#
```

```
# whoami
```

```
root
```

```
#
```

```
#
```

```
#
```

```
^]
```

```
TELNET> help
```

```
Information available:
```

BIND_SESSION	CONNECT	CREATE_SESSION	DELETE_SESSION			
DISABLE	DISCONNECT	ENABLE	EXIT	HELP	RESUME	SEND
SET	SHOW	SPAWN	TN3270	UNBIND_SESSION		

```
(...)
```

6. Make a directory called [TCPTEST]. Populate it with several files. Have a partner in class do the same. Use the RCP command to copy the files from one system to the other.

```
# ls /tmp/tcptest
```

```
disktab hosts
```

```
#
```

```
# exit
```

```
%TELNET-S-REMCLOSED, Remote connection closed
```

```
-TELNET-I-SESSION, Session 01, host alfman, port 23
```

```
$
```

```
$ rcp alfman:/tmp/tcptest/* sys$scratch
```

```
%DCL-W-IVQUAL, unrecognized qualifier - check validity, spelling, and placement  
\TMP\
```

```
$ rcp "alfman:/tmp/tcptest/*" sys$scratch
```

```
$ dir sys$scratch:h*
```

```
Directory SYS$SYSROOT:[SYSMGR]
```

```
HOSTS.;1
```

```
(...)
```

```
Total of 6 files.
```

```
$ dir sys$scratch:di*/date
```

```
Directory SYS$SYSROOT:[SYSMGR]
```

```
DISKTAB.;1          2-JUL-1999 17:12:50.52
```

```
Total of 1 file.
```

```
$ sho time
```

```
2-JUL-1999 17:13:33
```

Use proxies to avoid having to include a user name and password on the command line.

\$ tcpip

TCPIP> show proxy

VMS User_name	Type	User_ID	Group_ID	Host_name
OBRIEN	ON	13	15	*
SYSTEM	ON	0	1	ALFMAN
SYSTEM	CD	root		cuff

TCPIP>

TCPIP>

TCPIP> remove proxy system

VMS User_name	Type	User_ID	Group_ID	Host_name
SYSTEM	ON	0	1	ALFMAN

Remove? [N]:y

VMS User_name	Type	Remote User_name	Host_name
SYSTEM	CD	root	cuff

Remove? [N]:

TCPIP> exit

\$ rcp "alfman:/tmp/tcptest/*" sys\$scratch

I removed the 'trust' line from /etc/hosts.equiv on alfman.

\$ rcp "alfman:/tmp/tcptest/*" sys\$scratch

Permission denied.

%RCP-E-CONHST, error connecting to remote host ALFMAN

If a UNIX system is available, set up a *trust* relationship using */etc/hosts.equiv*.

I added the 'trust' line back into */etc/hosts.equiv* on alfman.

```
$  
$ rcp "alfman:/tmp/tcptest/*" sys$scratch  
$  
$  
$ rsh alfman "grep den255 /etc/hosts.equiv"  
den255  
den255_sl  
$
```

7. Use RLOGIN to access a remote system.

```
$ rlogin /username=obrien alfman
```

Password:

8. Use RSH to get a directory listing on a remote node. If available, do this on Tru64 UNIX and OpenVMS.

```
$ rsh alfman "ls -lR /"
```

9. Use FTP to copy the same files as mentioned in step 6.

\$ ftp alfman

220 alfman FTP server (Digital UNIX Version 5.60) ready.

Connected to ALFMAN.

Name (ALFMAN:system): **root**

331 Password required for root.

Password:

230 User root logged in.

FTP>

FTP>

FTP> **cd /tmp/tcptest**

250 CWD command successful.

FTP> **ls**

200 PORT command successful.

150 Opening ASCII mode data connection for file list (192.206.126.27,1028).

hosts

disktab

226 Transfer complete.

16 bytes received in 00:00:00.00 seconds (15.62 Kbytes/s)

FTP>

FTP>

FTP> **lcd sys\$scratch**

Local directory now SYS\$SYSROOT:[SYSMGR]

FTP> **mget ***

200 PORT command successful.

150 Opening ASCII mode data connection for disktab (192.206.126.27,1030) (31114 bytes).

226 Transfer complete.

local: SYS\$SYSROOT:[SYSMGR]DISKTAB.;4 remote: disktab

31114 bytes received in 00:00:00.05 seconds (595.77 Kbytes/s)

200 PORT command successful.

150 Opening ASCII mode data connection for hosts (192.206.126.27,1031) (2053 bytes).

226 Transfer complete.

local: SYS\$SYSROOT:[SYSMGR]HOSTS.;4 remote: hosts

2053 bytes received in 00:00:00.01 seconds (133.65 Kbytes/s)

FTP>

FTP>

FTP> **bye**

221 Goodbye.

\$

10. Try the new UNIX style commands.

a. Display the arp table.

\$ **tcpip**

TCPIP> **arp -a**

ALFMAN (192.206.126.3) at 08-00-2b-39-80-cb

TCPIP>

TCPIP> **ping compcom**

PING COMPCOM (192.206.126.80): 56 data bytes

64 bytes from 192.206.126.80: icmp_seq=0 ttl=128 time=1 ms

(...)

TCPIP> **arp -a**

COMPCOM (192.206.126.80) at 00-40-d0-08-a2-b3

ALFMAN (192.206.126.3) at 08-00-2b-39-80-cb

TCPIP>

TCPIP>

TCPIP> **show arp**

Address Resolution Mapping Table: 37 Buckets 16 Deep 36 Bytes

Cnt

Flags	Timer	Host	Phys Addr
1: UC	12	192.206.126.80	00-40-d0-08-a2-b3
2: UC	44	192.206.126.3	08-00-2b-39-80-cb

IPv6 Neighbor Cache Entries:

Cnt

Flags	Timer	Host	Phys Addr	Interface
-------	-------	------	-----------	-----------

- b. Display configurable interfaces.

TCPIP> **ifconfig -a**

TN0: flags=80<NOARP>

WE0: flags=c43<UP,BROADCAST,RUNNING,MULTICAST,SIMPLEX>

 Hwaddr aa-00-04-00-01-04 inet 192.206.126.27 netmask fffffff0
 broadcast 192.206.126.255

 ipmtu 1500

SL0: flags=11<UP,POINTOPOINT> inet 192.206.127.27 --> 0.0.0.0 netmask
 ffffff00

 ipmtu 1006

L00: flags=100c89<UP,LOOPBACK,NOARP, MULTICAST,SIMPLEX,NOCHECKSUM> inet
 127.0.0.1 netmask ff000000

 ipmtu 4096

TCPIP> **sho inter**

Interface	IP_Addr	Network mask	Packets		MTU
			Receive	Send	
WE0	192.206.126.27	255.255.255.0	783	706	1500
SL0	192.206.127.27	255.255.255.0	0	0	1006
L00	127.0.0.1	255.0.0.0	29	29	0

TCPIP>

c. Display the routing table.

TCPIP> **netstat -rn**

Routing tables

Destination	Gateway	Flags	Refs	Use	Interface
Route Tree for Protocol Family 2					
default	192.206.127.27	U	0	0	SL0
127.0.0.1	127.0.0.1	UH	0	8	LO0
192.206.126/24	192.206.126.27	U	1	697	WE0

Route Tree for Protocol Family 26

TCPIP>

TCPIP>

TCPIP> **sho route**

DYNAMIC

Type	Destination	Gateway
DN	0.0.0.0	192.206.127.27
AH	127.0.0.1	127.0.0.1
DN	192.206.126.0/24	192.206.126.27

TCPIP>

TCPIP>

TCPIP> **sho communication/route**

Routing tables

Destination	Gateway	Flags	Refs	Use	Interface
default	192.206.127.27	U	0	0	SL0
127.0.0.1	127.0.0.1	UH	0	8	LO0
192.206.126/24	192.206.126.27	U	1	697	WE0

TCPIP>

d. Display network statistics.

TCPIP> **sho comm**

Communication Parameters

Local host:	den255	Domain:	bruden.com	
		Maximum	Current	Peak
Proxies		20		
Remote Terminal				
Large buffers:	0			
UCBs:	0			
Virtual term:	disabled			

TCPIP>

TCPIP>

TCPIP> **sho comm /mem**

99 mbufs in use:

- 1 mbufs allocated to data
- 1 mbufs allocated to packet headers
- 13 mbufs allocated to socket structures
- 23 mbufs allocated to protocol control blocks
- 15 mbufs allocated to routing table entries
- 2 mbufs allocated to socket names and addresses
- 14 mbufs allocated to interface addresses
- 1 mbufs allocated to OpenVMS kernel table
- 2 mbufs allocated to OpenVMS ip multicast address
- 1 mbufs allocated to OpenVMS interface multicast address
- 1 mbufs allocated to OpenVMS BG-LAN VCIB including IVCIB
- 2 mbufs allocated to OpenVMS IFNET structure
- 3 mbufs allocated to OpenVMS LAN VCI VCIB structure
- 1 mbufs allocated to OpenVMS LAN MCAST_REQ structure
- 2 mbufs allocated to OpenVMS SELECT structure
- 1 mbufs allocated to OpenVMS ACP SLIP Structure
- 16 mbufs allocated to OpenVMS ACP allocated SERV Structure

TCPIP>

TCPIP> **netstat -m**

99 mbufs in use:

1 mbufs allocated to data

(...)

TCPIP>

TCPIP> **netstat -i**

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
TN0*	1280	<Link>	Link#2	0	0	0	0	0
WE0	1500	<Link>	aa:00:04:00:01:04	790	0	706	0	0
WE0	1500	192.206.126	den255	790	0	706	0	0
SL0	1006	<Link>	Link#4	0	0	0	0	0
SL0	1006	192.206.127	192.206.127.27	0	0	0	0	0
L00	4096	<Link>	Link#1	29	0	29	0	0
L00	4096	127	LOCALHOST	29	0	29	0	0

TCPIP>

TCPIP> **netstat -s**

ip:

791 total packets received

0 bad header checksums

0 with size smaller than minimum

0 with data size < data length

0 with header length < data size

0 with data length < header length

0 fragments received

0 fragments dropped (dup or out of space)

0 fragments dropped after timeout

0 packets forwarded

0 packets not forwardable

0 packets denied access

0 redirects sent

0 packets with unknown or unsupported protocol

791 packets consumed here

709 total packets generated here

- 0 lost packets due to resource problems
- 0 total packets reassembled ok
- 0 output packets fragmented ok
- 0 output fragments created
- 0 packets with special flags set

icmp:

- 4 calls to icmp_error
- 0 errors not generated 'cuz old ip message was too short
- 0 errors not generated 'cuz old message was icmp
- Output histogram:
 - destination unreachable: 4
- 0 messages with bad code fields
- 0 messages < minimum length
- 0 bad checksums
- 2 messages with bad length
- Input histogram:
 - echo reply: 17
 - destination unreachable: 4
- 0 message responses generated

tcp:

- 673 packets sent
 - 255 data packets (1195 bytes)
 - 0 data packets (0 bytes) retransmitted
 - 292 ack-only packets (250 delayed)
 - 0 URG only packets
 - 0 window probe packets
 - 88 window update packets
 - 38 control packets
- 628 packets received
 - 281 acks (for 1221 bytes)
 - 12 duplicate acks
 - 0 acks for unsent data
 - 537 packets (360924 bytes) received in-sequence

```

    0 packets with some dup. data (0 bytes duped)
    3 out-of-order packets (0 bytes)
    0 packets (0 bytes) of data after window
    0 window probes
    0 window update packets
    0 packets received after close
    0 discarded for bad checksums
    0 discarded for bad header offset fields
    0 discarded because packet too short
16 connection requests
5 connection accepts
21 connections established (including accepts)
33 connections closed (including 0 drops)
0 embryonic connections dropped
296 segments updated rtt (of 297 attempts)
1 retransmit timeout
0 connections dropped by rexmit timeout
0 persist timeouts
12 keepalive timeouts
    0 keepalive probes sent
    0 connections dropped by keepalive
udp:
    15 packets sent
    142 packets received
    0 incomplete headers
    0 bad data length fields
    0 bad checksums
    0 full sockets
    27 for no port (23 broadcasts, 0 multicasts)
    0 input packets missed pcb cache

```

TCPIP>

- e. Display configured subsystems.

```
TCPIP> sysconfig -s
```

```
inet: loaded and configured
```

```
socket: loaded and configured
```

```
arp: loaded and configured
```

```
TCPIP>
```

```
TCPIP>
```

- f. Display attributes within a configured subsystem.

```
TCPIP> sysconfig -q inet
```

```
inet:
```

```
inifaddr_hsize = 32
```

```
ipdefttl = 64
```

```
ipdirected_broadcast = 0
```

```
ipforwarding = 0
```

```
ipfragttl = 60
```

```
ipgateway = 0
```

```
ipport_userreserved = 5000
```

```
(...)
```

11. Extra credit: Prepare your system for *anonymous ftp*.
 - a. **Create an account** ANONYMOUS with the password GUEST.
 - b. **Set account access restrictions** NOLOCAL, NOBATCH, NOREMOTE, and NODIALUP.
 - c. **Create a welcome banner.** When an anonymous user logs in, FTP informs the user of the restrictions on the account. This information consists of a banner in the format of a HELP screen. Define the text you want to display by defining the TCPIP\$FTP_ANONYMOUS_WELCOME logical name.
 - d. **Specify a location for the log files generated by FTP sessions.** Use the TCPIP\$FTP_ANONYMOUS_LOG logical name. If you do not define TCPIP\$FTP_ANONYMOUS_LOG, FTP puts the files in SYSSYSDEVICE:[TCPIP\$FTP]TCPIP\$FTP_ANONYMOUS.LOG.
 - e. **Specify a user name for the anonymous FTP account.** Define the logical name TCPIP\$FTP_ANONYMOUS_ALIAS.

Protocols — Module 4

1. Run TCPTRACE while pinging a neighbor.

\$ tcptrace

ICMP Type = 8 ECHO, Code = 0 , Checksum = ^xDF7C

ICMP Identifier ^x1D01, Sequence Number = 2048 = ^x0800

```
1B7ECEC0  64A20140  0000895A  54000045  0000  E..TZ...@...d..~.
377D3424 | 0008011D  7CDF0008 | 037ECEC0  0010  ..~....|....$4}7
13121110  0F0E0D0C  0B0A0908  000E1359  0020  Y.....
23222120  1F1E1D1C  1B1A1918  17161514  0030  ..... !"#
33323130  2F2E2D2C  2B2A2928  27262524  0040  $%&'()*+,-./0123
                                37363534  0050  4567
```

ICMP Type = 0 ECHO_REPLY, Code = 0 , Checksum = ^xE77C

ICMP Identifier ^x1D01, Sequence Number = 2048 = ^x0800

```
037ECEC0  44E90140  0000A913  54000045  0000  E..T....@...D..~.
377D3424 | 0008011D  7CE70000 | 1B7ECEC0  0010  ..~....|....$4}7
13121110  0F0E0D0C  0B0A0908  000E1359  0020  Y.....
23222120  1F1E1D1C  1B1A1918  17161514  0030  ..... !"#
33323130  2F2E2D2C  2B2A2928  27262524  0040  $%&'()*+,-./0123
                                37363534  0050  4567
```

ICMP Type = 8 ECHO, Code = 0 , Checksum = ^xDB6D

ICMP Identifier ^x1D01, Sequence Number = 2304 = ^x0900

```
1B7ECEC0  63A20140  00008A5A  54000045  0000  E..TZ...@...c..~.
377D3425 | 0009011D  6DDB0008 | 037ECEC0  0010  ..~....m....%4}7
13121110  0F0E0D0C  0B0A0908  000E225B  0020  [".....
23222120  1F1E1D1C  1B1A1918  17161514  0030  ..... !"#
33323130  2F2E2D2C  2B2A2928  27262524  0040  $%&'()*+,-./0123
                                37363534  0050  4567
```

ICMP Type = 0 ECHO_REPLY, Code = 0 , Checksum = ^xE36D

ICMP Identifier ^x1D01, Sequence Number = 2304 = ^x0900

```

037ECEC0  43E90140  0000AA13  54000045  0000  E..T....@..C..~.
377D3425 | 0009011D  6DE30000 | 1B7ECEC0  0010  ..~....m....%4}7
13121110  0F0E0D0C  0B0A0908  000E225B  0020  [".....
23222120  1F1E1D1C  1B1A1918  17161514  0030  ..... !"#
33323130  2F2E2D2C  2B2A2928  27262524  0040  $%&'()*+,-./0123
                                37363534  0050  4567

```

\$

Run TCPTRACE/FULL while pinging.

\$ **tcptrace/full**

TCPIPtrace full display XMT packet 1 at 2-JUL-1999 17:50:38.95

```

IP Version = 4,  IHL = 5,  TOS = 00,  Total Length = 84 = ^x0054
IP Identifier  = ^x5A93,  Flags (0=0,DF=0,MF=0),
    Fragment Offset = 0 = ^x0000,  Calculated Offset = 0 = ^x0000
IP TTL = 64 = ^x40,  Protocol = 1 = ^x01,  Header Checksum = ^xA25A
IP Source Address      = 192.206.126.27
IP Destination Address = 192.206.126.3

```

```

ICMP Type = 8 ECHO,  Code = 0 ,  Checksum = ^xAC10
ICMP Identifier ^x1D01,  Sequence Number = 4608 = ^x1200

```

```

1B7ECEC0  5AA20140  0000935A  54000045  0000  E..TZ...@..Z..~.
377D342E | 0012011D  10AC0008 | 037ECEC0  0010  ..~.....4}7
13121110  0F0E0D0C  0B0A0908  000E7F78  0020  x.....
23222120  1F1E1D1C  1B1A1918  17161514  0030  ..... !"#
33323130  2F2E2D2C  2B2A2928  27262524  0040  $%&'()*+,-./0123
                                37363534  0050  4567

```

TCPIPtrace full display RCV packet 2 at 2-JUL-1999 17:50:38.95

IP Version = 4, IHL = 5, TOS = 00, Total Length = 84 = ^x0054
IP Identifier = ^x13B3, Flags (0=0,DF=0,MF=0),
Fragment Offset = 0 = ^x0000, Calculated Offset = 0 = ^x0000
IP TTL = 64 = ^x40, Protocol = 1 = ^x01, Header Checksum = ^xE93A
IP Source Address = 192.206.126.3
IP Destination Address = 192.206.126.27

ICMP Type = 0 ECHO_REPLY, Code = 0, Checksum = ^xB410
ICMP Identifier ^x1D01, Sequence Number = 4608 = ^x1200

037E	CEC0	3AE9	0140	0000	B313	5400	0045	0000	E..T....@.....~.
377D	342E	0012	011D	10B4	0000	1B7E	CEC0	0010	..~.....4}7
1312	1110	0F0E	0D0C	0B0A	0908	000E	7F78	0020	X.....
2322	2120	1F1E	1D1C	1B1A	1918	1716	1514	0030	 !"#
3332	3130	2F2E	2D2C	2B2A	2928	2726	2524	0040	\$%&'()*+,-./0123
						3736	3534	0050	4567

What is the difference?

The /FULL qualifier includes the packet contents.

2. Run TCPTRACE during a Telnet session.

\$ **tcptrace**

TCPIPtrace XMT packet 1 at 2-JUL-1999 17:52:01.13

	IP Address	Port	Seq #	Ack #
Source	192.206.126.27		1032	606976074 4294967005
Destination	192.206.126.3		<u>23</u>	
			Packet Length	41
TCP flags	PSH ACK			
window		4380		

			Hex	Count	Ascii
-----	-----	-----	-----	----	-----
1B7ECEC0	35A6063C	0000DE5A	29000045	0000	E..)Z...<..5..~.
DDFEFFFF	4AB82D24	17000804	037ECEC0	0010	..~.....\$-.J....
	64	00007EDD	1C111850	0020	P....~..d

TCPIPtrace RCV packet 2 at 2-JUL-1999 17:52:01.13

	IP Address	Port	Seq #	Ack #
Source	192.206.126.3		<u>23</u>	4294967005 606976075
Destination	192.206.126.27		1032	

Packet Length			41		
TCP flags	PSH ACK				
window	33580				
Hex	Count	Ascii			
-----	-----	-----	-----	-----	-----
037ECEC0	12AD063C	0040F113	29001045	0000	E..).@.<.....~.
4BB82D24	DDFEFFFF	08041700	1B7ECEC0	0010	..~.....\$-.K
0000	00000064	00006D6B	2C831850	0020	P.,km..d.....

How often is Telnet sending a packet over the wire?

One packet per character typed!

Can you change the behavior of Telnet so that it sends one message per line?

Yes. Alter the characteristic by using escape (^]) to get to the TELNET> prompt. Then enter set mode line.

TCPIPtrace XMT packet 5 at 2-JUL-1999 17:55:19.10

IP Address	Port	Seq #	Ack #		
Source	192.206.126.27	1032	606976150	4294967192	
Destination	192.206.126.3		<u>23</u>		
Packet Length			57		
TCP flags	PSH ACK				
window	4380				
Hex	Count	Ascii			
-----	-----	-----	-----	-----	-----
1B7ECEC0	E1A5063C	0000225B	39000045	0000	E..9["..<.....~.
98FFFFFF	96B82D24	17000804	037ECEC0	0010	..~.....\$-.....
2F706D74	2F206463	0000DF0F	1C111850	0020	P..... <u>cd /tmp/</u>
	0A	0D747365	74706374	0030	<u>tcptest</u> ..

3. Use TCPTRACE to confirm the port numbers and protocols used by FTP (see the next answer).
4. Does Telnet use the TOS field in the IP header?

Yes.

\$ TCPTRACE /FULL

TCPIPtrace full display RCV packet 2 at 2-JUL-1999 17:56:58.78

IP Version = 4, IHL = 5, TOS = 10, Total Length = 59 = ^x003B
 IP Identifier = ^x14B6, Flags (0=0,DF=1,MF=0),
 Fragment Offset = 0 = ^x0000, Calculated Offset = 0 = ^x0000
 IP TTL = 60 = ^x3C, Protocol = 6 = ^x06, Header Checksum = ^xAC3B
 IP Source Address = 192.206.126.3
 IP Destination Address = 192.206.126.27

TCP Source Port = 23, TCP Destination Port = 1032
 TCP Sequence Number = 4294967211 = ^xFFFFFFFFAB
 TCP Acknowledge Number = 606976184 = ^x242DB8B8
 Flags (URG=0,ACK=1,PSH=1,RST=0,SYN=0,FIN=0),
 Window = 33580 = ^x832C
 TCP Checksum = ^x7D74, Urgent Pointer = 0 = ^x0000

037ECEC0	3BAC063C	0040B614	3B001045	0000	E...@.<...;...~.
B8B82D24	ABFFFFFF	08041700	1B7ECEC0	0010	..~.....\$-..
2F706D74	2F206463	0000747D	2C831850	0020	P.,}t..cd /tmp/
	20230A	0D747365	74706374	0030	tcptest..#

Does FTP?

Yes.

TCPIPtrace full display RCV packet 4 at 2-JUL-1999 17:57:49.86

IP Version = 4, IHL = 5, TOS = 10, Total Length = 98 = ^x0062
IP Identifier = ^x14C6, Flags (0=0,DF=1,MF=0),
Fragment Offset = 0 = ^x0000, Calculated Offset = 0 = ^x0000
IP TTL = 60 = ^x3C, Protocol = 6 = ^x06, Header Checksum = ^xAC04
IP Source Address = 192.206.126.3
IP Destination Address = 192.206.126.27

TCP Source Port = 21, TCP Destination Port = 1034
TCP Sequence Number = 4294966273 = ^xFFFFFC01
TCP Acknowledge Number = 658560002 = ^x2740D402
Flags (URG=0,ACK=1,PSH=1,RST=0,SYN=0,FIN=0),
Window = 33580 = ^x832C
TCP Checksum = ^x9275, Urgent Pointer = 0 = ^x0000

037ECEC0	04AC063C	0040C614	62001045	0000	E..b..@.<.....~.
02D44027	01FCFFFF	0A041500	1B7ECEC0	0010	..~.....'@..
6D666C61	20303232	00007592	2C831850	0020	P.,.u..220 alfm
44282072	65767265	73205054	46206E61	0030	an FTP server (D
73726556	2058494E	55206C61	74696769	0040	igital UNIX Vers
2E796461	65722029	30362E35	206E6F69	0050	ion 5.60) ready.

NFS — Module 5

1. What are the advantages of NFS?
 - **Saves disk space and hardware costs**
 - **Transparent distribution of data**
 Disadvantages?
 - **Performance detriment**
 - **No file locking (on OpenVMS)**
 - **Administrative overhead**
2. With a partner, configure one node as an NFS server and the other as a client. Create a directory named [NFS_OUT] and populate it with several files.
 - a. Export the directory on the server.

```
$ rsh alfman showmount -e
```

Exports list on localhost:

```
/tmp/nfs_out                Everyone
/morerz0/obrien/nfs_dir     Everyone
```

```
$
```

- b. Mount the served directory on the client and get access to the files.

```
$ tcpip
```

```
TCPIP> mount dnfs0: /host="alfman" /path="/tmp/nfs_out"
```

```
%TCPIP$DNFSMOUNT-E-MOUNTFAIL, error mounting _DNFS1:[000000]
```

```
-SYSTEM-F-INVLOGIN, login information invalid at remote node
```

```
TCPIP> show proxy
```

VMS User_name	Type	User_ID	Group_ID	Host_name
OBRIEN	ON	13	15	*
SYSTEM	CD	root		cuff

```
TCPIP> add proxy system /uid=0 /gid=1 /nfs=(outgoing,Incoming) /host="alfman"
```

```
TCPIP>
```

```
TCPIP>
```

TCPIP> **sho proxy system**

VMS User_name	Type	User_ID	Group_ID	Host_name
SYSTEM	CD	root		cuff
SYSTEM	ON	0	1	alfman

TCPIP>

TCPIP>

TCPIP> **mount dnfs0: /host="alfman" /path="/tmp/nfs_out"**

%TCPIP\$DNFSMOUNT-S-MOUNTED, /tmp/nfs_out mounted on _DNFS2:[000000]

TCPIP> **exit**

\$

\$

\$ **set def dnfs2:[000000]**

\$ **dir/prot/own**

Directory DNFS2:[000000]

DISKTAB.;1	[SYSTEM]	(RWED,RWED,RED,RED)
------------	----------	---------------------

HOSTS.;1	[SYSTEM]	(RWD,RWD,RD,RD)
----------	----------	-----------------

Total of 2 files.

\$

\$

\$ **tcPIP**

TCPIP> **dismount dnfs2:**

%TCPIP\$DNFSDISMOUNT-S-DISMOUNTED, DNFS2:[000000] dismounted

TCPIP>

TCPIP>

TCPIP> **exit**

- c. Create a file within the served directory while logged in to the client.
- d. Check file ownership and permissions.

```
$ tcpip
TCPIP>
TCPIP>
TCPIP> mount dnfs0: /host="alfman" /path="/tmp/nfs_out"
%TCPIP$DNFSMOUNT-S-MOUNTED, /tmp/nfs_out mounted on _DNFS3:[000000]
TCPIP>
TCPIP>
TCPIP> show mount
_DNFS3:[000000]    mounted
    ALFMAN:/tmp/nfs_out
TCPIP> show mount/full
_DNFS3:[000000]    mounted
    ALFMAN:/tmp/nfs_out
    Transport                TCPIP-UDP    Writing                Enabled
    Read/write size          8192/8192    Write conversion        Enabled
    RPC timeout              0 00:00:01.00    ADF usage              USE,UPDATE,CREATE
    RPC retry limit          0           Fileids                Unique
    Attribute time           0 00:00:15.00    Server type            UNIX
    Directory time           0 00:00:30.00    Advisory Locking       Disabled
    Cache Validation         MODIFY TIME    Default user           [DEFAULT]
    Superuser                No           Default UID,GID        -2, -2
TCPIP>
TCPIP>
TCPIP>
TCPIP> exit
$
$
$ set def dnfs3:[000000]
$ dir/own/prot
Directory DNFS3:[000000]
DISKTAB.;1                [SYSTEM]                (RWED,RWED,RED,RED)
HOSTS.;1                  [SYSTEM]                (RWD,RWD,RD,RD)
Total of 2 files.
Rev. 9.41
```

\$ **dir/own/prot**

Directory DNFS3:[000000]

DISKTAB.;1	[SYSTEM]	(RWED, RWED, RED, RED)
------------	----------	------------------------

<u>FILE_BYE_ROOT.;1</u>	<u>[SYSTEM]</u>	(RWD, RWD, RD, RD)
-------------------------	-----------------	--------------------

HOSTS.;1	[SYSTEM]	(RWD, RWD, RD, RD)
----------	----------	--------------------

Total of 3 files.

\$ **dir/own/prot**

Directory DNFS3:[000000]

DISKTAB.;1	[SYSTEM]	(RWED, RWED, RED, RED)
------------	----------	------------------------

FILE_BYE_ROOT.;1	[SYSTEM]	(RWD, RWD, RD, RD)
------------------	----------	--------------------

<u>FILE_BY_OBRIEN.;1</u>	<u>[DEFAULT]</u>	(RWD, RWD, RD, RD)
--------------------------	------------------	--------------------

HOSTS.;1	[SYSTEM]	(RWD, RWD, RD, RD)
----------	----------	--------------------

Total of 4 files.

\$

\$

- e. Can a user logged in to the server get access to the file created while logged in to the client?

Yes.

- f. What happens if one user on the client and one user on the server (or another client) try to access the served file at the same time?

There is no locking (yet), so the users can accessed the simultaneously.

3. Explain the function of NFS proxy accounts. Set some up and use them to access the served files.

They set up a correlation between the UNIX UIDs and GIDs, and the user accounts on the OpenVMS system.

What is the difference between an NFS proxy and a communication proxy?

An NFS proxy is used for NFS; a communication proxy is used for everything else.

4. Create a CFS on an OpenVMS server system. Mount and access it from the client.

```
$ tcpip create container dka300:[sys0.tcpipcourse.con_tainer] /user_name=system
/root_mode=741 /host=alfman
```

```
%TCPIP-E-CFSERROR, error processing TCPIP file system request
```

```
-TCPIP-E-CCF_CFDIREXI, specified container directory already exists
```

```
$ tcpip map "/test_tainer_disk" dka300:
```

```
$ tcpip show map
```

```
Dynamic Filesystem Map
```

Pathname	Logical File System
----------	---------------------

/test_tainer_disk	DKA300:
-------------------	---------

```
$ tcpip map "/test_tainer" dka300:[sys0.tcpipcourse.con_tainer]
```

```
$ tcpip show map
```

```
Dynamic Filesystem Map
```

Pathname	Logical File System
----------	---------------------

/test_tainer	DKA300:[SYS0.TCIPICOURSE.CON_TAINER]
--------------	--------------------------------------

/test_tainer_disk	DKA300:
-------------------	---------

```
$
```

```
$
```

```
$ tcpip show export
```

File System	Host name
-------------	-----------

/tset_tainer	*
--------------	---

/vms_nfs/sys0/tcpipcourse/ob_nfs	alfman
----------------------------------	--------

```
$ tcpip add export "/test_tainer" /host=*
```

```
$
```

```
$
```

\$ tcpip show export

File System	Host name
/test_tainer	*
/tset_tainer	*
/vms_nfs/sys0/tcpipcourse/ob_nfs	alfman

\$

\$

\$ tcpip

TCPIP>

TCPIP> show cfs

CFS SERVICES		2-JUL-1999 19:05:12.99	
CFS Services		OpenVMS Services	
CLOSE	0		
CREATE_FH	0	\$ASSIGN	2 \$QIO 0
CREDIR_FH	0	\$DASSGN	0 Access 1
FREEBUFF	0		Create 0
GETATTR	0	\$DEQ	0 Deaccess 0
LINK_FH	0	\$ENQ	15
LOOKUP_FH	0		Read_attr 1
OPEN_FH	1	getVMpage	5 Write_attr 0
READ	0	freeVMpage	0
READBUFF	0		Lookup 0
READDIR_FH	0	\$CLREF	6
READLINK_FH	0	\$SETEF	6 Extend 0
REMDIR_FH	0		
REMOVE_FH	0	\$DCLAST	2 Delete 0
RENAME_FH	0	\$CLRAST	12 Enter 0
SETATTR	0	\$SETAST	36 Remove 0
STATFS	5		

```

SYMLINK_FH          0      $GETDVI          7      Read_V          2
WRITE               0
OTHER               2      $CHKPRO          0
TOTAL               8

```

TCPIP>

TCPIP>

TCPIP>

TCPIP> **show cfs/summary**

CFS Service status and performance 2-JUL-1999 19:06:27.07

Service State	Cur	Max	Total	Cacheop	Cur	Hit	I/O	Inc	Status
FP-Access	0	0	0	Read	0	0	0	0	Clusize 16
FP-Attributes	0	0	0	Read-A	0	0	0	0	Limit 256
FP-Delete	0	0	0	Write	0	0	0	0	Inuse 2
FP-Dir	0	0	0	Write-A	0	0	0	0	Busy 0
FP-Rename	0	0	0	Write-D	0				Hitrate 0
FP Sub_total	0	0	0						
Buffered	0	0	0	Nameop	Cur	Hit			Status
Internal	0	0	0	-----	----	----	-----		
I/O	0	0	0	Add	0		Tabsize		0
Lock	0	0	0	Delete	0		Inuse		0
Logging	0	0	0	Lookup	0	0	Hitrate		0
Resource	0	0	0						
RMS	0	0	0	Fileop	Cur	Hit			Status
Service	0	0	0	-----	----	----	-----		
Timer	0	0	0	Find	0	0	Limit		0
Other	0	0	0	Find-A		0	Inuse	65535	
Synch	0	0	0	Find-C		0	Timeout		5
							Hitrate		0

Services count 1 2 10 ATCBs: 2 TBABs: 0 RDCBs: 7 Pages: 0

TCPIP>

TCPIP>

TCPIP> **sho serv nfs /full**

Service: NFS

		State:	Enabled	
Port:	2049	Protocol:	UDP	Address: 0.0.0.0
Inactivity:	0	User_name:	TCPIP\$NFS	Process: TCPIP\$NFS
Limit:	1	Active:	0	Peak: 0

File: TCPIP\$SYSTEM:TCPIP\$NFS_RUN.COM

Flags: None

Socket Opts: Rcheck Scheck

Receive: 64000 Send: 64000

Log Opts: Acpt Actv Dactv Conn Error Exit Logi Logo Mdfy Rjct Tim0 Addr

File: SYS\$SYSDEVICE:[TCPIP\$NFS]TCPIP\$NFS_RUN.LOG

Security

Reject msg: not defined

Accept host: 0.0.0.0

Accept netw: 0.0.0.0

TCPIP>

TCPIP>

TCPIP> **exit**

\$ set def sys\$sysdevice:[tcpip\$nf]

\$ dir

Directory SYS\$SYSDEVICE:[TCPIP\$NFS]

LOGIN.COM;1 TCPIP\$NFS_DEN255_LOGFILE.LOG;3

TCPIP\$NFS_DEN255_LOGFILE.LOG;2 TCPIP\$NFS_DEN255_LOGFILE.LOG;1

TCPIP\$NFS_RUN.LOG;3 TCPIP\$NFS_RUN.LOG;2 TCPIP\$NFS_RUN.LOG;1

Total of 7 files.

\$

\$ **type TCPIP\$NFS_DEN255_LOGFILE.LOG;3**

```
%%%%%%%%%%%% NFS$SERVER      2-JUL-1999 18:47:43.96 %%%%%%%%%%%%%%
%TCPIP-S-NFS_INISUC, NFS server is starting using log file
SYS$SYSDEVICE:[TCPIP$NFS]TCPIP$NFS_DEN255_LOGFILE.LOG;3
```

```
%%%%%%%%%%%% NFS$SERVER      2-JUL-1999 18:47:44.03 %%%%%%%%%%%%%%
%TCPIP-I-NFS_INITST, starting NFS with parameters:
writing into errlog file is enabled
default UID is -2
default GID is -2
inactivity timer is set to 02:00
```

```
%%%%%%%%%%%% NFS$SERVER      2-JUL-1999 18:47:44.28 %%%%%%%%%%%%%%
%TCPIP-S-NFS_HSTNAM, NFS server is running on host den255
```

```
%%%%%%%%%%%% NFS$SERVER      2-JUL-1999 18:50:44.12 %%%%%%%%%%%%%%
%TCPIP-S-NFS_MNTSUC, mounted file system /test_tainer
-TCP-IP-S-NFS_CLIENT, uid=0 gid=1 host_name = alfman
```

```
%%%%%%%%%%%% NFS$SERVER      2-JUL-1999 18:50:44.18 %%%%%%%%%%%%%%
%TCPIP-W-NFS_NOCMAP, cannot find client record in proxy database (TCPIP$PROXY)
-TCP-IP-S-NFS_CLIENT, uid=-2 gid=-2 host_name = alfman
```

\$

\$

\$

\$ **tcpip**

TCPIP> **add proxy system /uid=-2 /gid=-2 /nfs=(outgoing,incoming) /host="alfman"**

TCPIP>

TCPIP>

TCPIP> **show proxy**

VMS User_name	Type	User_ID	Group_ID	Host_name
OBRIEN	OND	13	15	*
SYSTEM	OND	0	1	alfman
SYSTEM	OND	-2	-2	alfman

TCPIP>

TCPIP>

TCPIP> **exit**

\$

\$

\$ **rsh alfman "df -t nfs"**

Filesystem	512-blocks	Used	Available	Capacity	Mounted on
den255:/test_tainer	4110480	1223120	2887360	30%	/test_tainer

\$

\$

5. Turn off the PORTMAP service on the server node. What happens when the client tries to mount?

It fails with an invalid RPC number unless data has been cached.

\$ **tcpip**

TCPIP>

TCPIP>

TCPIP> **sho portmap**

Program Number	Version	Protocol	Port-number	Process	Service-name
000186A0 (100000)	2	TCP	111	000000C5	PORTMAPPER
000186A0 (100000)	2	UDP	111	000000C5	PORTMAPPER
000186A3 (100003)	2	UDP	2049	0000012C	NFS
000186A5 (100005)	1	UDP	10	0000012C	MOUNT

TCPIP>

TCPIP> **set noserv portmapper**

Service	Port	Proto	Process	Address	State
PORTMAPPER	111	TCP	TCPIP\$PORTM	0.0.0.0	

Remove? [N]:y

TCPIP>

6. If your server is a UNIX system, alter some file characteristics (using SET FILE) and see how the UNIX NFS server handles it.

```
$ set file /nobackup nfs_test18.txt
```

```
$
```

```
$
```

```
$ dir n*
```

```
Directory DNFS19:[000000]
```

```
NFSCIENT3.TXT;1    NFS_TEST18.TXT;1
```

```
Total of 2 files.
```

```
$
```

```
$
```

```
$ rsh alfman ls -l /morez0/obrien/nfs_dir
```

```
total 39
```

```
-rwxr-xr--  1 nobody   system      90 Jun 11 21:20 .$ADF$nfs_test18.txt;1

-rw-r--r--  1 root     system    1104 Jun 10 15:09 denpasswd
-rwxr-xr-x  1 root     system   31114 Jun 10 15:10 disks
-rw-r--r--  1 root     system    1995 Jun 10 15:09 hosts
-rwxr-x---  1 nobody   system     24 Jun 11 21:15 nfs_test18.txt
-rwxr-x---  1 nobody   system      0 Jun 11 21:24 nfscient3.txt
-rwxr-x---  1 obrien   system     44 Jun 11 14:31 ovms_obrien_test1.txt
-rwxr-x---  1 nobody   system     38 Jun 11 14:26 ovms_test1.txt
```

Serial Line and Static Routing — Modules 6 and 7

1. Select several systems to function as static routers (do not enable GATED or ROUTED for this lab). For example, if the room consists of four tables with three machines on each, choose one machine on each table to be the router node. Two of the tables need two routers — they will have serial line connections to two different tables.
2. Configure SLIP lines (no modems) connecting the routers on each table. Choose new network numbers for each table and for each serial line pair.

For example:

Table	All Nodes Ethernet Interface	Router 1 SLIP1	Router 2 SLIP2
Table 1	192.206.126	192.206.130	N.A.
Table 2	192.206.127	192.206.130	192.206.131
Table 3	192.206.128	192.206.131	192.206.132
Table 4	192.206.129	192.206.132	N.A.

As a result, each table is logically on its own Ethernet segment. (It is not necessary to do this physically; the software will think it cannot get to table 2 if it is on a different network.)

The goal of this lab is to try to get all nodes pinging (including the ability to ping the IP address of the SLIP interface).

Note that the selected router nodes now have two IP addresses. Each node with a SLIP interface configured should also be configured as a router.

This lab forces many trips to the routing tables (`TCPIP SHOW ROUTE` or `netstat -nr`) for purposes of troubleshooting.

Be prepared to do some TCPTRACE activities if questions arise. Determine the address that the sending node presents. Be aware that getting a packet to a destination is only half the battle; it has to make it back as well!

Configure SLIP as appropriate. Use `route add` to add routes to the routing tables. Consider a reasonable default route to add to your routing table. Alter the HOSTS database as appropriate. Try the pings using addresses primarily. You can add the host names later.

Subnet Routing — Module 7



Note

Do not proceed to this lab if you have not completed the previous lab.

This second routing lab is the same as the previous one except that the nodes are all using the same network ID (192.206.126), but are configured in separate subnets.

Based on your room configuration, the actual number of subnets may vary.

Given the room configuration from the previous lab, choose a reasonable subnet mask. All interfaces will use the same subnet mask.

Set up the tables as follows:

	All Nodes Ethernet Interface	Router 1 SLIP1	Router 2 SLIP2
Table 1	Subnet 1	Subnet 5	N.A.
Table 2	Subnet 2	Subnet 5	Subnet 6
Table 3	Subnet 3	Subnet 6	Subnet 7
Table 4	Subnet 4	Subnet 7	N.A.

Again, the goal is to be able to ping **all** interfaces on **all** nodes using both IP addresses and node names.

Use the same general procedure as in the static routing lab, but you must make some decisions across the room concerning the subnet mask and subnet numbers. Draw a room map to keep track of what is happening.

BIND — Module 8

1. Configure the room with one master server, two slave servers, and the rest as clients. Make sure the master server has all node name and address information that you want to serve.

Use **TCPTP\$CONFIG** as shown in Module 8, DNS and BIND.

\$ @sys\$startup:tcpip\$config

TCP/IP Network Configuration Procedure

(...)

DIGITAL TCP/IP Services for OpenVMS Configuration Menu

Configuration options:

- 1 - Core environment
 - 2 - Client components
- (...)

Enter configuration option: **1**

DIGITAL TCP/IP Services for OpenVMS CORE ENVIRONMENT Configuration Menu

Configuration options:

- 1 - BIND Resolver
- 2 - Domain
- 3 - Routing
- 4 - Interfaces
- 5 - Time Zone

A - Configure options 1 - 5

[E] - Exit menu

Enter configuration option: **2**

DOMAIN Configuration

Enter Internet domain [bruden.com]:

DIGITAL TCP/IP Services for OpenVMS Configuration Menu

Configuration options:

- 1 - Core environment
- 2 - Client components
- 3 - Server components
- (...)

Enter configuration option: **3**

DIGITAL TCP/IP Services for OpenVMS SERVER Components Configuration Menu

Configuration options:

- 1 - BIND Disabled
- (...)
- 16 - LBROKER Enabled
- 17 - DHCP Disabled
- A - Configure options 1 - 17
- [E] - Exit menu

Enter configuration option: **1**

BIND SERVER Configuration

Service is not defined in the SYSUAF.

Service is not defined in the TCPIP\$SERVICE database.

Service is not enabled

BIND SERVER configuration options:

1 - Enable service on this node

[E] - Exit BIND configuration

Enter configuration option: **1**

The TCPIP\$AUX identifier already exists with value 3655

Creating BIND Service Entry

Checking for BIND template files in SYS\$SPECIFIC:[TCPIP\$BIND]

Creating template SYS\$SPECIFIC:[TCPIP\$BIND]TCPIP\$BIND_CONF.TEMPLATE

Creating file SYS\$SPECIFIC:[TCPIP\$BIND]127_0_0.DB

Creating file SYS\$SPECIFIC:[TCPIP\$BIND]LOCALHOST.DB

Creating file SYS\$SPECIFIC:[TCPIP\$BIND]ROOT.HINT

DIGITAL TCP/IP Services for OpenVMS SERVER Components Configuration Menu

Configuration options:

1	-	BIND	<u>Enabled</u>
2	-	BOOTP	Disabled

(...)

Shutdown and restart the network..

%TCPIP-I-ENABSERV, enabling services

%TCPIP-I-STARTNAME, starting name service

(...)

2. Configure the clients to point to both the master and slave for their BIND service. Remove several host names from the client's local database. Try to ping using the removed host name.

\$ @sys\$startup:tcpip\$config

TCP/IP Network Configuration Procedure

This procedure helps you define the parameters required to run DIGITAL TCP/IP Services for OpenVMS on this system.

Checking TCP/IP Services for OpenVMS configuration database files.
No new database files were created.

DIGITAL TCP/IP Services for OpenVMS Configuration Menu
Configuration options:

1 - Core environment
(...)

Enter configuration option: 1

DIGITAL TCP/IP Services for OpenVMS CORE ENVIRONMENT Configuration Menu

Configuration options:

1 - BIND Resolver
2 - Domain
(...)

Enter configuration option: 1

BIND RESOLVER Configuration

A BIND resolver has already been configured.

BIND Resolver Configuration

Transport: UDP
Domain:
Retry: 4
Timeout: 4
Servers: not defined
Path: No values defined

* Do you want to reconfigure BIND [NO]: y

Enter your BIND server name: **alfman** (master server)

Enter your BIND server name: (put secondary server here)

DIGITAL TCP/IP Services for OpenVMS CORE ENVIRONMENT Configuration Menu

Configuration options:

- 1 - BIND Resolver
- 2 - Domain
- (...)

Enter configuration option: 2

DOMAIN Configuration

Enter Internet domain [bruden.com]:

(...)

- 5 - Shutdown DIGITAL TCP/IP Services for OpenVMS
- 6 - Startup DIGITAL TCP/IP Services for OpenVMS
- 7 - Run tests

- A - Configure options 1 - 4
- [E] - Exit configuration procedure

Enter configuration option: **5**

Begin Shutdown...

(...)

6 - Startup DIGITAL TCP/IP Services for OpenVMS

7 - Run tests

A - Configure options 1 - 4

[E] - Exit configuration procedure

Enter configuration option: **6**

Begin Startup...

(...)

DIGITAL TCP/IP Services for OpenVMS Configuration Menu

3. Troubleshoot if necessary using NSLOOKUP.

See Module 8.

Check Appendix B to see what commands are available within NSLOOKUP.

4. Use TCPTRACE to follow the BIND communications. Start a zone transfer while a trace is in progress.

TCPIPtrace XMT packet 7 at 2-JUL-1999 21:56:28.92

IP Address Port

Source	192.206.126.27	1135			
Destination	192.206.126.3	53			
		Packet Length		63	
UDP					
			Hex	Count	Ascii
-----	-----	-----	-----	----	-----
1B7ECEC0	2187111E	0000D197	3F000045	0000	E..?......!...~.
00012B85	E8562B00	35006F04	037ECEC0	0010	..~..0.5.+V...+..
06353532	6E656406	00000000	00000100	0020	den255.
01001C	00006D6F	63036E65	64757262	0030	bruden.com.....

TCPIPtrace RCV packet 8 at 2-JUL-1999 21:56:28.92

IP Address Port

Source	192.206.126.3	53			
Destination	192.206.126.27	1135			
		Packet Length		63	
UDP					
			Hex	Count	Ascii
-----	-----	-----	-----	----	-----
037ECEC0	944C111E	00005ED2	3F000045	0000	E..?.^....L...~.
80852B85	67D22B00	6F043500	1B7ECEC0	0010	..~..5.o.+g...+..
06353532	6E656406	00000000	00000100	0020	den255.
01001C	00006D6F	63036E65	64757262	0030	bruden.com.....

How would you prevent attempts to contact the Internet root servers?

Alter the SYS\$SPECIFIC:ROOT>HINT file (formerly NAMED.CA).

Dynamic Routing — Module 7



Note

Do not proceed to this lab if you have not completed the subnet routing lab.

1. Keep the room configured with subnetting. Configure the ROUTED daemon on all nodes (not just the routers). Have ROUTED configured on the routers so that it advertises its routes. Have ROUTED configured on the end nodes in silent mode.

\$ @sys\$startup:tcpip\$config

TCP/IP Network Configuration Procedure

(...)

DIGITAL TCP/IP Services for OpenVMS Configuration Menu

Configuration options:

- 1 - Core environment

(...)

Enter configuration option: 1

DIGITAL TCP/IP Services for OpenVMS CORE ENVIRONMENT Configuration Menu

Configuration options:

- 1 - BIND Resolver
- 2 - Domain
- 3 - Routing

(...)

Enter configuration option: 3

DYNAMIC ROUTING Configuration

Dynamic routing has not been configured.

You may configure dynamic ROUTED or GATED routing.

You cannot enable both at the same time. If you want to change from one to the other, you must disable the current routing first, then enable the desired routing.

If you enable dynamic ROUTED routing, this host will use the Routing Information Protocol (RIP) - Version 1 to listen for all dynamic routing information coming from other hosts to update its internal routing tables.

It will also supply its own Internet addresses to routing requests made from remote hosts.

If you enable dynamic GATED routing, you will be able to configure this host to use any combination of the following routing protocols to exchange dynamic routing information with other hosts on the network:

- Routing Information Protocol (RIP) - Version 1 & 2
- Router Discovery Protocol (RDISC)
- Open Shortest Path First (OSPF)
- Exterior Gateway Protocol (EGP)
- Border Gateway Protocol (BGP-4)
- Static routes

* Do you want to configure dynamic ROUTED or GATED routing [NO]: y

* Do you want to enable GATED routing configuration [NO]:

ROUTED option

If you enable the 'supply' option of dynamic routing, this host will supply dynamic routing information to other hosts on the network whether it is acting as an internetwork gateway or not.

* Do you want this host to supply its dynamic routing information [NO]:

(End nodes would say no.)

There can be one and only one host on your network

which defines a default network route.

If you want this host to supply the default network route,

then all packets which cannot be routed to their

destination host or network will be routed via this host.

* Do you want this host to supply the default network route [NO]: **y**

(...)

Restart the network and ROUTED should be running.

\$

\$ **pipe sho sys | search sys\$input routed**

000000CB TCPIP\$ROUTED (...)

\$

\$ **sho proc/id=cb /all**

22-JUN-1999 17:03:03.32	User: INTERNet	Process ID: 000000CB
	Node: DEN255	Process name: "TCPIP\$ROUTED"

Terminal:

User Identifier: [SYSTEM]

Base priority: 4

Default file spec: Not available

Number of Kthreads: 1

Devices allocated: BG255:

BG257:

Process Quotas:

(...)

There are 2 processes in this job:

TCPIP\$INET_ACP

TCPIP\$ROUTED (*)

\$

2. Remove a few static routes from your routing tables. See if the dynamic routing replaces them with useful information.

Use the TCPIP SET NOROUTE... command.

3. Use TCPTRACE to see if the ROUTED daemons on the routers are actually sending out their 30-second updates.

\$ tcptrace/port=local=520/full

```
-----
TCPIPtrace full display RCV packet 5 at 22-JUN-1999 17:42:23.05
IP Version = 4,  IHL = 5,  TOS = 00,  Total Length = 72 = ^x0048
IP Identifier   = ^x2CB2,  Flags (0=0,DF=0,MF=0),
    Fragment Offset = 0 = ^x0000,  Calculated Offset = 0 = ^x0000
IP TTL = 30 = ^x1E,  Protocol = 17 = ^x11,  Header Checksum = ^xF153
IP Source Address      = 192.206.126.3
IP Destination Address = 192.206.126.255
```

UDP Source Port = 520, UDP Destination Port = 520

UDP Header and Datagram Length = 52 = ^x0034, Checksum = ^xFC2D

```
037ECEC0  53F1111E  0000B22C  48000045  0000  E..H,.....S..~.
00000102 | 2DFC3400  08020802 | FF7ECEC0  0010  ..~.....4.-....
00000000  00000000  037FCEC0  00000200  0020  .....
00000000  007ECEC0  00000200  01000000  0030  .....~.....
                01000000  00000000  0040  .....
```

BOOTP and DHCP — Modules 9 and 10



Note

The instructor will supply details for a BOOTP lab based on system availability and class interest.

\$ @sys\$startup:tcPIP\$config

TCP/IP Network Configuration Procedure

(...)

Configuration options:

- 1 - Core environment
- 2 - Client components
- 3 - Server components

(...)

Enter configuration option: **3**

DIGITAL TCP/IP Services for OpenVMS SERVER Components Configuration Menu

Configuration options:

- 1 - BIND Disabled
- 2 - BOOTP Disabled
- 3 - TFTP Disabled

(...)

Enter configuration option: **2**

BOOTP SERVER Configuration

Service is not defined in the SYSUAF.

Service is not defined in the TCPIP\$SERVICE database.

Service is not enabled

BOOTP SERVER configuration options:

- 1 - Enable service on this node

[E] - Exit BOOTP configuration

Enter configuration option: 1

The TCPIP\$AUX identifier already exists with value 3655

Found DHCP Service Entry. Disabling and Deleting it...

Creating BOOTP Service Entry

Creating SYSS\$COMMON:[SYSEXE]TCPIP\$BOOTP.DAT

DIGITAL TCP/IP Services for OpenVMS SERVER Components Configuration Menu

Configuration options:

- 1 - BIND Disabled
- 2 - BOOTP Enabled
- 3 - TFTP Disabled
- (...)

TCPIP> **set bootp cuff /hardware=address=08-00-2b-e7-71-58 /file=vmunix.;1**

1. Isolate a few target systems and delete their network interfaces.
2. Configure DHCP on a server and set it up with a pool of five addresses (192.206.126.80 –192.206.126.90).

Use the GUI (dhcpgui) to set up the pool.

3. Check the contents of the NAMEPOOL file to see the result.

\$ type SYSS\$SYSDEVICE:[TCPIP\$DHCP]NAMEPOOL.;10

Try to get a client to recognize its IP address presented from the DHCP server.

NTP — Module 11

1. Select and configure a system as an NTP stratum 1 server. Select and configure a second system as an NTP stratum 2 server.

```
$ set def sys$specific:[tcpip$ntp]
```

```
$
```

```
$ copy TCPIP$NTP.TEMPLATE;1 tcpip$ntp.conf
```

```
$
```

```
$ ! edit this file to reflect your servers.
```

```
$
```

```
$ type tcpip$ntp.conf
```

```
(...)
```

```
server 192.206.126.3 # alfman
```

```
peer 192.206.126.4 # cuff
```

```
(...)
```

```
$
```

```
$ @sys$startup:tcpip$ntp_shutdown
```

```
TCPIP$NTP server 000002B2 stopped
```

```
$
```

```
$
```

```
$ @sys$startup:tcpip$ntp_startup
```

```
$
```

```
$ tcpip show serv ntp/full
```

```
Service: NTP
```

```
State: Enabled
```

```
Port: 123
```

```
Protocol: UDP
```

```
Address: 0.0.0.0
```

```
Inactivity: 0
```

```
User_name: TCPIP$NTP
```

```
Process: TCPIP$NTP
```

```
(...)
```

```
$
```

2. Configure all other machines as NTP client (peer) machines.
3. Check synchronization of times.
\$ **SHOW TIME**
4. Modify the time on one system so that it is out of synchronization by two minutes. Check to see if it gets resynchronized.
\$ **SET TIME ...**
5. Try using the ntptrace and ntpq commands.

```
$ ntptrace := $sys$system:tcpip$ntpttrace.exe
```

```
$
```

```
$ ntptrace
```

```
LOCALHOST:
```

```
stratum 5, offset 0.000001, synch distance 0.02396
```

```
ALFMAN:
```

```
stratum 4, offset 0.004401, synch distance 0.01129
```

```
$
```

```
$ ntpq := $sys$system:tcpip$ntpq.exe
```

```
$
```

```
$ ntpq -p
```

remote	refid	st	t	when	poll	reach	delay	offset	disp
*ALFMAN	LOCAL(1)	4	u	12	64	377	2.93	16.125	5.11
CUFF	0.0.0.0	16	u	-	64	0	0.00	0.000	16000.0

```
$
```

```
$
```

SNMP — Module 12

1. Configure all systems as SNMP agents.

```
$ tcpip show config snmp /full
```

SNMP Configuration

Flags: AuthenTraps Sets

Contact: system

Location

First: Manchester, MA

Community	Type	Address_list
public	Read	0.0.0.0

\$

The TCPIP\$CONFIG procedure sets up:

- SNMP at port 161.
- SNMP master agent, at port 242 (used in the OpenVMS implementation as the master agent port number that is known to the local subagents).
- Files in the SYS\$SYSDEVICE:[TCPIP\$SNMP] directory.

You can run the SNMP startup procedure (SYS\$STARTUP:TCPIP\$SNMP_STARTUP.COM) from TCPIP\$STARTUP.COM or you can run it directly.

TCPIP\$SNMP_STARTUP.COM:

- Checks the TCP/IP Services license and enables the SNMP service.
- Installs images with the required privileges (BYPASS, PHY_IO, and WORLD).
- Runs TCPIP\$EXTENSION_MIB_STARTUP.COM (found in SYS\$SYSDEVICE:[TCPIP\$SNMP]), which handles user-written subagents.
- Sends a UDP datagram to the SNMP port. When the port receives the datagram, the auxiliary server runs the TCPIP\$SNMP_RUN.COM procedure (found in TCPIP\$SYSTEM:).

- Review the MIB material and experiment with the `snmp_request` command.

```
$ @sys$manager:tcip$define_commands
$
$ sho sym snmp_request
SNMP_REQUEST = "$SYS$SYSTEM:TCPIP$SNMP_REQUEST.EXE"
$
$ @sys$startup:TCPIP$SNMP_SHUTDOWN.COM
TCPIP$SNMP stopped
TCPIP$OS_MIBS stopped
TCPIP$HR_MIB stopped
$
$ @sys$startup:TCPIP$SNMP_STARTUP.COM
$
$ snmp_request den255 "Bruden" get 1.3.6.1.2.1.1.4.0

1.3.6.1.2.1.1.4.0 = Dennis O'Brien
$
```

- Try using the `set` and the `get` capabilities within the `snmp_request` command.

```
$ snmp_request alfman "Bruden" set 1.3.6.1.2.1.1.4.0 "-D" "Dennis O'B"

1.3.6.1.2.1.1.4.0 = Dennis O'B
```

4. Look under SYS\$EXAMPLES for any .MY files. Use the information in the .MY files to support your experiments.

```
$ dir [syshlp.examples.tcpip.snmp]
```

```
Directory SYS$COMMON:[SYSHLP.EXAMPLES.TCPIP.SNMP]
```

```
CHES$METHOD.C;1    CHES$MIB.C;1      CHES$MIB.MY;1     CHES$TBL.C;1
CHES$TBL.H;1       GAWK.EXE_ALPHA;1  GAWK.EXE_VAX;1    MIB-CONVERTER.AWK;1
RFC1213.MY;1       RFC1231.MY;1      RFC1285.MY;1      RFC1442.MY;1
SNMP-SMI.MY;1      SNMP-TC.MY;1      TCPIP$BUILD_CHES$.COM;1
TCPIP$CHES$SUBAGENT.EXE;1  TCPIP$CHES$SUBAGENT.OPT;1
V2-TC.MY;1
```

```
Total of 18 files.
```

```
$
```

Remote Printing — Module 13

1. If a printer is available, set up the printer locally off one machine using `lprsetup`.

```
$ run sys$system:tcpip$lprsetup
```

TCPIP Printer Setup Program

Command < add delete view help exit >: **a**

Adding printer entry, type '?' for help.

Enter printer name to add : ?

Enter a printer name, ie., the name by which you want to identify the printer through the `lpr` command, example: `lpr -Pprintername`.

Enter printer name to add : **oboyprint**

Enter the FULL name of one of the following printer types:

remote local : ?

You can set up 2 kinds of printers

'local' or 'remote'.

The default printer type is 'local'.

Enter the FULL name of one of the following printer types:

remote local : **local**

Enter printer synonym: **ons**

Enter printer synonym:

Enter full file specification for spool directory

SPOOLER DIRECTORY 'sd' : [SYS\$SPECIFIC:[TCPIP\$LPD.OBOYPRINT]] ?

Enter full file specification for printer log file.

printer error log file 'lf' [SYS\$SPECIFIC:[TCPIP\$LPD]OBOYPRINT.LOG] ?

Enter the name of the printcap symbol you wish to modify. Other valid entry is :

'q' to quit (no more changes)

The names of the printcap symbols are:

sd for the printer spool directory

lf for the printer error log file
lp for the name of the local printer
ps for the LPD PrintServer extensions flag
rm for the name of the remote host
rp for the name of the remote printer
fm for the printer form field
pa for the /PASSALL flag
nd for the /NODELETE flag
cr for the cr flag
p1-p8 for the /PARAMETER=(p1,...,p8) field

Enter symbol name: **p**

	Symbol	type	value
	-----	----	-----
Error log file	: lf	STR	/SYS\$SPECIFIC/TCPIP\$LPD/OBOYPRINT.LOG
Printer Queue	: lp	STR	OBOYPRINT
Spool Directory	: sd	STR	/SYS\$SPECIFIC/TCPIP\$LPD/OBOYPRINT
Parameter fields	: p1	STR	

Enter symbol name: **?**

Enter the name of the printcap symbol you wish to modify. **Other**
valid entry is :

'q' to quit (no more changes)

Enter symbol name: **lp**

Enter a new value for symbol 'lp'? [OBOYPRINT]

Enter symbol name: **q**

	Symbol	type	value
	-----	----	-----
Error log file	: lf	STR	/SYS\$SPECIFIC/TCPIP\$LPD/OBOYPRINT.LOG
Printer Queue	: lp	STR	OBOYPRINT
Spool Directory	: sd	STR	/SYS\$SPECIFIC/TCPIP\$LPD/OBOYPRINT
Parameter fields	: p1	STR	

Are these the final values for printer OBOYPRINT ? [y]

Adding comments to printcap file for new printer, type '?' for help.

Do you want to add comments to the printcap file [n] ? :

```

*****
* TCPIP$LPD_STARTUP.COM      TCPIP$LPD_PRINTCAP*
*   and TCPIP$LPD_SHUTDOWN.COM      *
*   have been updated for this printer      *
*                                           *
* Set up activity is complete for this printer*
*****

```

Command < add delete view help exit >: **ex**

\$

2. Configure that machine as a print server.

(...)

DIGITAL TCP/IP Services for OpenVMS SERVER Components Configuration Menu

Configuration options:

1	-	BIND	Disabled
2	-	BOOTP	Enabled
3	-	TFTP	Enabled
4	-	FTP	Enabled
5	-	<u>LPR/LPD</u>	<u>Enabled</u>
6	-	NFS	Enabled
7	-	PC-NFS	Disabled

(...)

3. Configure all other machines as clients to enable access to the remote print queue using `lprsetup`.

```
$ run sys$system:tcpip$lprsetup
```

TCPIP Printer Setup Program

Command < add delete view help exit >: **a**

Adding printer entry, type '?' for help.

Enter printer name to add : ?

Enter a printer name, ie., the name by which you want to identify the printer through the `lpr` command, example: `lpr -Pprintername`.

Enter printer name to add : **alfp**

Enter the FULL name of one of the following printer types:

remote local : **remote**

Enter printer synonym:

Enter full file specification for spool director

SPOOLER DIRECTORY 'sd' : [SYS\$SPECIFIC:[TCPIP\$LPD.ALPD]] ?

Set LPD PrintServer extensions flag 'ps' [] ?

Set remote system name 'rm' [] ? **alfman**

Set remote system printer name 'rp' [] ? ?

The 'rp' parameter specifies the remote printer name argument.

The name specified must be one of the recognized names for the printer on the remote machine.

Enter a new value, or press RETURN to use the default.

Enter a value for 'rp'? [] ? **alfprint**

(...)

4. Check the SYSS\$SPECIFIC:[TCPIP\$LPD]TCPIP\$PRINTCAP.DAT file.

```
$ type sys$specific:[tcpip$lpd]tcpip$printcap.dat
```

```
#
```

```
# LOCAL PRINTERS
```

```
#
```

```
TCPIP$LPD_QUEUE:\
```

```
    :lp=TCPIP$LPD_QUEUE:\
```

```
    :sd=TCPIP$LPD_SPOOL:
```

```
#
```

```
OLDIE|oldie:\
```

```
    :lf=/SYSS$SPECIFIC/TCPIP$LPD/OLDIE.LOG:\
```

```
    :lp=OLDIE:\
```

```
    :sd=/SYSS$SPECIFIC/TCPIP$LPD/OLDIE:
```

```
#
```

```
ALFP|alfp:\
```

```
    :lf=/SYSS$SPECIFIC/TCPIP$LPD/ALFP.LOG:\
```

```
    :lp=ALFP:\
```

```
    :rm=alfman:\
```

```
    :rp=alfprint:\
```

```
    :sd=/SYSS$SPECIFIC/TCPIP$LPD/ALFP:
```

```
#
```

```
$
```

5. How would you prevent users on node alfman from accessing the LPD service?

```
TCPIP> SET SERVICE LPD /REJECT=HOSTS=alfman
```

Troubleshooting — Module 14

1. Try normal network activities with `REPLY/ENABLE=NETWORK`.

\$ reply/enable=network

```
%%%%%%%%%% OPCOM 2-JUL-1999 20:21:20.42 %%%%%%%%%%%  
Operator _DEN255$RTA1: has been enabled, username SYSTEM
```

```
%%%%%%%%%% OPCOM 2-JUL-1999 20:21:20.42 %%%%%%%%%%%  
Operator status for operator _DEN255$RTA1:  
NETWORK
```

```
%%%%%%%%%% OPCOM 2-JUL-1999 20:21:28.41 %%%%%%%%%%%  
Message from user TCPIP$NFS on DEN255  
%TCPIP-S-NFS_MNTSUC, mounted file system /test_tainer  
-TCPIP-S-NFS_CLIENT, uid=0 gid=1 host_name = alfman
```

```
%%%%%%%%%% OPCOM 2-JUL-1999 20:21:44.38 %%%%%%%%%%%  
Message from user INTERnet on DEN255  
INTERnet ACP Process creation success
```

```
%%%%%%%%%% OPCOM 2-JUL-1999 20:21:44.38 %%%%%%%%%%%  
Message from user INTERnet on DEN255  
INTERnet ACP FTP Accept Request from Host: 192.206.126.3 Port: 1073
```

```
%%%%%%%%%% OPCOM 2-JUL-1999 20:22:19.62 %%%%%%%%%%%  
Message from user INTERnet on DEN255  
TELNET Login Request from Remote Host: 192.206.126.3 Port: 1075
```

```
%%%%%%%%%% OPCOM 2-JUL-1999 20:23:01.74 %%%%%%%%%%%  
Message from user INTERnet on DEN255  
INTERnet ACP RSH Connection Request from Host: 192.206.126.27 Port: 1023
```

%%%%%%%%%% OPCOM 2-JUL-1999 20:23:01.75 %%%%%%%%%%%

Message from user INTERnet on DEN255

INTERnet ACP RSH Reject Request - proxy cache empty - from Host: 192.206.126.3
Port: 1023

\$ reply/disable=network

%%%%%%%%%% OPCOM 2-JUL-1999 20:23:55.10 %%%%%%%%%%%

Operator _DEN255\$RTA1: has been disabled, username SYSTEM

\$

2. Try using the TCPIP ANALYZE SERVICE command.

This command shows information on user network applications.

3. Using FTP, copy a large file. Write down the time statistics. Check all FTP logical names. Research and use logical names TCPIP\$FTP_WNDSIZE, TCPIP\$FTP_FILE_ALQ, and TCPIP\$FTP_FILE_DEQ. Copy the file again using FTP and see if it makes a difference.

\$

\$ **ftp alfman**

220 alfman FTP server (Digital UNIX Version 5.60) ready.

Connected to ALFMAN.

Name (ALFMAN:system): **root**

331 Password required for root.

Password:

230 User root logged in.

FTP>

FTP>

FTP> **cd /**

250 CWD command successful.

FTP>

FTP>

FTP> **lcd sys\$scratch**

Local directory now SYS\$SYSROOT:[SYSMGR]

FTP>

FTP>

FTP> **get vmunix**

200 PORT command successful.

150 Opening ASCII mode data connection for vmunix (192.206.126.27,1036) (9305104 bytes).

%TCPIP-E-FTP_OUTPROCF, error processing output file
SYS\$SYSROOT:[SYSMGR]VMUNIX.;1

-RMS-W-RTB, 35825 byte record too large for user's buffer

421 Service not available, Remote server has closed the connection

```
$ dir sys$scratch:vm* /size
```

```
Directory SYS$SYSROOT:[SYSMGR]
```

```
VMSIMAGES.DAT;1          35
VMUNIX.;1                14523
```

```
Total of 2 files, 14558 blocks.
```

```
(...)
```

```
Grand total of 2 directories, 5 files, 14615 blocks.
```

```
$ del sys$scratch:vmunix.;1
```

```
$
```

```
$
```

```
$ ftp alfman
```

```
220 alfman FTP server (Digital UNIX Version 5.60) ready.
```

```
Connected to ALFMAN.
```

```
Name (ALFMAN:system): root
```

```
331 Password required for root.
```

```
Password:
```

```
230 User root logged in.
```

```
FTP> binary
```

```
200 Type set to I.
```

```
FTP> cd /
```

```
250 CWD command successful.
```

```
FTP> lcd sys$scratch
```

```
Local directory now SYS$SYSROOT:[SYSMGR]
```

```
FTP> get vmunix
```

```
200 PORT command successful.
```

```
150 Opening BINARY mode data connection for vmunix (192.206.126.27,1038)
    (9305104 bytes).
```

```
226 Transfer complete.
```

```
local: SYS$SYSROOT:[SYSMGR]VMUNIX.;1 remote: vmunix
```

```
9305104 bytes received in 00:00:12.24 seconds (742.16 Kbytes/s)
```

```
FTP>
```

FTP>

FTP> **bye**

221 Goodbye.

\$

\$ **dir sys\$scratch:vmu* /size**

Directory SYS\$SYSROOT:[SYSMGR]

VMUNIX.;1 18175

Total of 1 file, 18175 blocks.

\$

\$

\$ **del sys\$scratch:vmunix.;1**

\$

\$ **define /system/exec tcpip\$ftp_wndsize 16384**

\$

\$ **ftp alfman**

220 alfman FTP server (Digital UNIX Version 5.60) ready.

Connected to ALFMAN.

Name (ALFMAN:system): **root**

331 Password required for root.

Password:

230 User root logged in.

FTP>

FTP>

FTP> **cd /**

250 CWD command successful.

FTP>

FTP>

FTP> **lcd sys\$scratch**

Local directory now SYS\$SYSROOT:[SYSMGR]

FTP>

FTP>

FTP> **binary**

200 Type set to I.

FTP>

FTP>

FTP> **get vmunix**

200 PORT command successful.

150 Opening BINARY mode data connection for vmunix (192.206.126.27,1040)
(9305104 bytes).

226 Transfer complete.

local: SYS\$SYSROOT:[SYSMGR]VMUNIX.;1 remote: vmunix

9305104 bytes received in 00:00:12.00 seconds (757.12 Kbytes/s)

FTP>

FTP>

FTP> **bye**

421 Timeout (900 seconds): closing control connection.

\$

\$

\$ **dir sys\$scratch:vmu* /size**

Directory SYS\$SYSROOT:[SYSMGR]

VMUNIX.;1	<u>18175</u>
-----------	--------------

Total of 1 file, 18175 blocks.

\$ **del sys\$scratch:vmunix.;1**

\$

\$ **show rms_default**

	MULTI- BLOCK COUNT	Indexed	Relative	MULTIBUFFER COUNTS				Sequential	NETWORK BLOCK COUNT
				Disk	Magtape	Unit	Record		
Process	0	0	0	0	0	0	0	0	
System	16	0	0	0	0	0	0	8	

	Prolog	Extend	Quantity	VCC_DFW
Process	0		0	0
System	0		0	0

\$

\$

\$

\$ **define /system/exec tcpip\$ftp_file_alq 1000**

\$

\$ **define /system/exec tcpip\$ftp_file_deq 500**

\$

\$ **ftp alfman**

220 alfman FTP server (Digital UNIX Version 5.60) ready.

Connected to ALFMAN.

Name (ALFMAN:system): **root**

331 Password required for root.

Password:

230 User root logged in.

FTP> **cd /**

250 CWD command successful.

FTP> **lcd sys\$scratch**

Local directory now SYS\$SYSROOT:[SYSMGR]

FTP> **binary**

200 Type set to I.

FTP>

FTP>

FTP> **get vmunix**

200 PORT command successful.

150 Opening BINARY mode data connection for vmunix (192.206.126.27,1042)
(9305104 bytes).

226 Transfer complete.

local: SYS\$SYSROOT:[SYSMGR]VMUNIX.;1 remote: vmunix

9305104 bytes received in 00:00:08.93 seconds (1017.58 Kbytes/s)

FTP>

FTP>

FTP> **bye**

221 Goodbye.

\$

\$

\$ **dir sys\$scratch:vmu* /size**

Directory SYS\$SYSROOT:[SYSMGR]

VMUNIX.;1 18175

Total of 1 file, 18175 blocks.

\$

\$

\$ **del sys\$scratch:vmunix.;1**

\$

\$

4. Use Telnet to confirm that FTP, SMTP, or SNMP is functioning on the remote node.

\$ telnet alfman /port=21

%TELNET-I-TRYING, Trying ... 192.206.126.3

%TELNET-I-SESSION, Session 01, host alfman, port 21

-TELNET-I-ESCAPE, Escape character is ^]

220 alfman FTP server (Digital UNIX Version 5.60) ready.

quit

221 Goodbye.

%TELNET-S-REMCLOSED, Remote connection closed

-TELNET-I-SESSION, Session 01, host alfman, port 21

\$

\$

\$ telnet alfman /port=25

%TELNET-I-TRYING, Trying ... 192.206.126.3

%TELNET-I-SESSION, Session 01, host alfman, port 25

-TELNET-I-ESCAPE, Escape character is ^]

220 alfman ESMTP Sendmail 8.8.8 (1.1.22.3/09Jul99-0512PM) Tue, 27 Jul 1999
21:29:08 -0400 (EDT)

QUIT

221 alfman closing connection

%TELNET-S-REMCLOSED, Remote connection closed

-TELNET-I-SESSION, Session 01, host alfman, port 25

\$

\$

\$ telnet alfman /port=161

%TELNET-I-TRYING, Trying ... 192.206.126.3

%TELNET-E-CONNFAIL, Failed to connect to remote host

-SYSTEM-F-REJECT, connect to network object rejected

\$

5. What columns of a `netstat -i` command might indicate network problems?

```
$ tcpip "netstat -i"
```

```
Name Mtu Network Address Ipkts Ierrs Opkts Oerrs Coll
```

6. What columns of the `netstat -rn` output are useful in troubleshooting?

```
$ tcpip "netstat -rn"
```

Routing tables

```
Destination Gateway Flags Refs Use Interface
```

```
$
```

```
$ tcpip "netstat -rnv"
```

Routing tables

```
Destination Gateway Flags Refs Use Interface MTU
HopCnt Expire
```

```
RPipe SPipe Thresh RTT RTTvar Prec
```

SMTP — Module 15

1. Configure SMTP and send some mail messages (using SMTP).

Start sender with:

```
$ @SYSS$MANAGER:TCPIP$SMTP_STARTUP
```

Enable receiver with:

```
$ TCPIP ENABLE SERVICE SMTP
```

```
$ mail
```

```
MAIL> send
```

```
To: SMTP%"obrien@alfman"
```

2. Use TCPTRACE to watch communications during mail activities.

```
$ TCPTRACE /PORT=25
```