

TCP/IP Services for OpenVMS

Instructor Guide

Training



COMPAQ

TCP/IP Services for OpenVMS

Instructor 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.

Aero, ALPHA, ALPHA AXP, AlphaServer, AlphaStation, Armada, BackPaq, COMPAQ, Compaq Insight Manager, CompaqCare logo, Counselor, DECterm, Deskpro, DIGITAL, DIGITAL logo, DIGITAL Alpha Systems, Digital Equipment Corporation, DIGITAL UNIX, DirectPlus, FASTART, Himalaya, InfoPaq, Integrity, LicensePaq, Ministation, NetFlex, NonStop, OpenVMS, PaqFax, Presario, ProLiant, ProLinea, ProSignia, QuickBack, QuickFind, Qvision, RDF, RemotePaq, RomPaq, ServerNet, SERVICenter, SmartQ, SmartStart, SmartStation, SolutionPaq, SpeedPaq, StorageWorks, Systempro/LT, Tandem, TechPaq, 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

Instructor Guide

December 1999

Module 1 — Course Description

Course Description	1-1
Course Length	1-1
Intended Audience	1-1
Course Content and Objectives	1-1
Prerequisites	1-1
Resources	1-2
Lab Environment	1-3
Course Outline	1-4
Course Goals	1-8
Course Map	1-9
Course Conventions	1-9
Preparing to Teach	1-10
Preparing the System	1-10

Module 2 — Course Overview

Network Overview — Module 1	2-1
Instructor Notes	2-1
Key Points	2-2
Time Required to Teach	2-2
TCP/IP Commands — Module 2	2-3
Instructor Notes	2-3
Key Points	2-4
Time Required to Teach	2-4
Installing and Configuring TCP/IP Services — Module 3	2-5
Instructor Notes	2-5
Key Points	2-7
Time Required to Teach	2-7
Protocols — Module 4	2-8
Instructor Notes	2-8
Key Points	2-9
Time Required to Teach	2-9
NFS — Module 5	2-10
Instructor Notes	2-10
Key Points	2-12
Time Required to Teach	2-12
Serial Lines — Module 6	2-13
Instructor Notes	2-13
Key Points	2-19
Time Required to Teach	2-19
Routing — Module 7	2-20
Instructor Notes	2-20
Key Points	2-22
Time Required to Teach	2-22

DNS and BIND — Module 8.....	2-23
Instructor Notes.....	2-23
Key Points.....	2-25
Time Required to Teach.....	2-25
BOOTP — Module 9.....	2-26
Instructor Notes.....	2-26
Key Points.....	2-26
Time Required to Teach.....	2-26
DHCP — Module 10.....	2-27
Instructor Notes.....	2-27
Key Points.....	2-27
Time Required to Teach.....	2-27
NTP — Module 11.....	2-28
Instructor Notes.....	2-28
Key Points.....	2-29
Time Required to Teach.....	2-29
SNMP — Module 12.....	2-30
Instructor Notes.....	2-30
Key Points.....	2-31
Time Required to Teach.....	2-31
Remote Printing — Module 13.....	2-32
Instructor Notes.....	2-32
Key Points.....	2-33
Time Required to Teach.....	2-33
Troubleshooting — Module 14.....	2-34
Instructor Notes.....	2-34
Key Points.....	2-34
Time Required to Teach.....	2-34
SMTP — Module 15.....	2-35
Instructor Notes.....	2-35
Key Points.....	2-35
Time Required to Teach.....	2-35

Course Description

Course Length

Five days.

Intended Audience

This course is designed for computer professionals interested in understanding and administering the TCP/IP environment within the OpenVMS operating system.

Course Content and Objectives

This course is designed to provide the student with the information and experience needed to function in a networked environment focused on the TCP/IP protocols. After presenting introductory information on protocols and standards, the course covers the components and services within TCP/IP. The central protocols of TCP/IP are examined with emphasis on their purpose. That is, what need do they fulfill in the big picture of node-to-node communication? The course introduces user- and administrator-level programs through hands-on lab exercises. Students also examine static and dynamic routing through a series of labs.

Prerequisites

Students should be familiar with basic networking concepts and have system management level experience with OpenVMS.

Resources

- Students should receive:
 - *TCP/IP Services for OpenVMS, Student Guide*
 - *TCP/IP Services for OpenVMS, Lab Guide*
- Students should receive or have access to:
 - *DIGITAL TCP/IP Services for OpenVMS Release Notes*
 - *DIGITAL TCP/IP Services for OpenVMS Installation and Configuration*
 - *DIGITAL TCP/IP Services for OpenVMS Management*
 - *DIGITAL TCP/IP Services for OpenVMS Management Command Reference*
 - *DIGITAL TCP/IP Services for OpenVMS User's Guide*
- Students should have access to:
 - *DIGITAL TCP/IP Services for OpenVMS C Socket Programming*
 - *DIGITAL TCP/IP Services for OpenVMS eSNMP Programming and Reference*
 - *DIGITAL TCP/IP Services for OpenVMS ONC RPC Programming*
 - *TCP/IP Illustrated, Volume 1*, W. Richard Stevens, Addison-Wesley
 - *Internetworking with TCP/IP, Volume 1*, Douglas E. Comer, Prentice Hall
 - *DIGITAL UNIX Network Administration*
 - *DIGITAL UNIX System Configuration and Tuning*
- Students may benefit from:
 - *TCP/IP Network Administration*, Craig Hunt, O'Reilly
 - *DNS and BIND (Third Edition)*, Albitz and Liu, O'Reilly
 - *Troubleshooting TCP/IP*, Mark A. Miller, M&T Books
 - *Managing NFS and NIS*, Hal Stern, O'Reilly

Lab Environment

- The lab should have at least four machines, at least two of which should be running OpenVMS.
- The best setup is one machine for each student or 12 computers running OpenVMS with one Tru64 UNIX system in the mix.
- Systems should be connected through Ethernet and should not be connected to an outside network.
- Installation media should be available.
- Serial line hardware and adapters should be available but not connected.

Course Outline

- 1. Network Overview**
 - Network Concepts
 - IP Addresses
 - Subnet Masks
- 2. TCP/IP Commands**
 - User Services
 - PING
 - SHOW HOST
 - TELNET
 - RCP
 - RLOGIN
 - RSH
 - FINGER
 - FTP
- 3. Installing and Configuring TCP/IP Services**
 - Installation
 - Configuration
- 4. Protocols**
 - Data Link Protocol
 - IP Header Fields
 - TCP Header Fields
 - ICMP and ARP
 - UDP

5. Network File System (NFS)

- NFS Concepts
- NFS Implementation
- NFS Server File Systems
- Container File System
- NFS Management Commands
- Proxy Accounts
- Client Configuration
- Client Data Manipulation
- Server Configuration
- Server Data Manipulation

6. Serial Lines

- Serial Connections
- SLIP Encapsulation
- PPP Encapsulation
- Setting Up SLIP

7. Routing

- Routing Introduction
- Routing Table
- Static Routing
- Dynamic Routing
- ROUTED
- GATED
- Configuring Dynamic Routing

8. DNS and BIND

- History
- Concepts
- Resolver
- Server
- NSLOOKUP
- Load Broker and Metric Server

9. Bootstrap Protocol (BOOTP)

- BOOTP Concepts
- BOOTP Server
- BOOTP Client
- Server Configuration
- Server Data Files
- TFTP

10. Dynamic Host Configuration Protocol (DHCP)

- DHCP Concepts
- Advantages of Using DHCP
- DHCP Configuration Files
- Using the DHCP GUI
- DHCP Cluster Failover

11. Network Time Protocol (NTP)

- NTP Concepts
- NTP Internet Time Sources
- NTP Configuration Options
- NTP Support Files
- NTP Commands
- Trace of NTP Communication

12. Simple Network Management Protocol (SNMP)

- SNMP Concepts
- Log Files and Command Procedures
- SNMP Configuration
- Using SNMP Utilities
- Tracing SNMP Protocol Interactions

13. Remote Printing

- Remote Printing Options
- LPR/LPD Local Configuration
- LPR/LPD Remote Configuration
- Tracing Print Packets
- TELNETSYM Overview
- PC-NFS Overview

14. Troubleshooting

- Troubleshooting Options
- Configuration Troubleshooting
- Network Troubleshooting
- Performance Tuning

15. Simple Mail Transfer Protocol (SMTP)

- SMTP Concepts
- Configuring SMTP
- Using Telnet to Test SMTP
- POP and MIME Concepts

Course Goals

Upon completion of this course, the student should be able to:

- Describe TCP/IP as implemented on OpenVMS.
- Discuss IP addresses and masks.
- Use TCP/IP Services for OpenVMS user and management commands.
- Install and configure TCP/IP Services for OpenVMS.
- Manage NFS on an OpenVMS system.
- Configure and troubleshoot static and dynamic routing.
- Manage BIND on an OpenVMS system.
- Discuss the BOOTP, DHCP, NTP, SNMP, and SMTP protocols.
- Configure remote printing.
- Troubleshoot TCP/IP Services for OpenVMS.

This course does **not** cover the following:

- UNIX network administration.
- Network internals.

Course Map

Recommended sequence:

- **Monday**
 - Lecture: Modules 1, 2, and 3.
 - Labs: Network Overview, Configuration, and TCP/IP Commands
- **Tuesday**
 - Lecture: Modules 4, 5, 6, and introduce Module 7.
 - Labs: Protocols, NFS, and Serial Line and Static Routing.
- **Wednesday**
 - Lecture: Finish Module 7, Module 8, and Appendix A (optional).
 - Labs: Subnet Routing and BIND.
- **Thursday**
 - Lecture: Modules 9, 10, 11, 12, and Appendix C (optional).
 - Labs: Dynamic Routing, BOOTP and DHCP (optional), NTP, and SNMP.
- **Friday**
 - Lecture: Modules 13, 14, 15, and review (optional).
 - Labs: All optional; Remote Printing, Troubleshooting, and SMTP.

Course Conventions

In system displays, examples, and screen captures:

- `Courier New` identifies system display.
- **Courier New Bold** identifies user input.

Preparing to Teach

You should be proficient at some implementation of TCP/IP before trying to teach this course.

Review the screen capture examples to be sure you understand the points being made.

Plan to review lab solutions on the following morning of the course.

Read through the *TCP/IP for OpenVMS Management* manual. Look through the *TCP/IP Management Commands Reference* manual as well. Not all commands are covered in the materials. If you are new to TCP/IP, be sure you understand everything in the first chapter. Selectively read the Comer and Stephens books to fill in any gaps.

An audit of this course would be extremely useful to the new instructor. Instructors experienced in other TCP/IP implementations should be able to prepare with the above materials and a week or two of system time. Be sure to actually do the labs in preparation for the course.

Preparing the System

The systems should be running the most current version of OpenVMS and should either be running TCP/IP Services or have the installation media available.

Students should have access to the system account and a nonprivileged account.

Serial line wires and all necessary adapters should be available in the classroom.

Network Overview — Module 1

Instructor Notes

This module should act as a foundation for all future discussions. Much of the content of this module may act as a review, but time spent here will minimize later confusion.

One of the goals of this module is to get all of the students to see the network for what it is: a collection of machines running some specialized software that enables applications to communicate. Make sure the students are comfortable with the overview topics presented here. They will need to draw upon this base knowledge when presented with more sophisticated network software in subsequent modules.

Do not assume that the students know this material. Students will have heard all the buzzwords, but your mission is to turn the buzzwords into useful conceptual foundations.

The module slowly builds the notion of a protocol stack. Allow the students a few minutes to absorb each diagram and concept before moving to the next page. If your audience seems sophisticated, quiz them. Toss out questions like “What is a protocol?” and “What is the function of a subnet mask?”

The IP address and subnet mask part of the module will have been experienced by most of the audience. However, you will find few of them have a complete grasp of the concept of a subnet mask. (Many of them will thank you privately for time you spend on the material at the end of the first module.)

Make sure the students realize that this is not UCX, nor is it TCP/IP Services V4; it is TCP/IP Services for OpenVMS V5 and is based on the Compaq Tru64 UNIX implementation of TCP/IP. Therefore, they will see many UNIX style commands throughout the course (page 1–2).

Pages 1–3 through 1–20 introduce many critical concepts. Be sure the students are comfortable with the following ideas before moving to the second module: broadcasting compared to point-to-point, protocols, network device drivers, ports, MAC addresses, basic routing concepts, INETACP, TCP compared to UDP, and sockets.

The last 13 pages of the module present IP address classes and subnet masks. Be sure to point out the material in Appendix A, which supplements the last part of the module. It also helps those students who are less at ease with decimal to binary conversions.

Be prepared to make more lab exercises for the students to practice their subnet skills. Give them a repeat of lab number 8 (with different numbers) to work on each night. Review the answers in the morning. Ask questions like “Given a network address of 194.34.45 and a subnet mask of 255.255.255.224, what is the IP address of node 6 on subnet 3?” (Answer is 192.206.126.102.)

Key Points

Be sure the students understand the following:

- The difference between broadcasting media and point-to-point media.
- The meaning of the term *protocol*.
- Port numbers and sockets.
- The TCP/IP protocol stack and the purpose for each layer.
- How most “well-known” network services are started (through INETACP).
- The components of an IP address: network number, subnet number, and node number.
- The difference between the various IP address classes.
- How to manipulate and describe the meaning of subnet masks.

Time Required to Teach

This module should take approximately 55 to 75 minutes to cover.

TCP/IP Commands — Module 2

Instructor Notes

This module is an introduction to the commands common to users and administrators in the TCP/IP Services for OpenVMS environment. Take the students quickly through the command examples. Ask for questions after each example. Most students will be familiar with the commands.

Point out any port numbers, IP addresses, MAC addresses, or other items from the first module that appear in the output. Most of the examples show communication between **den255** (the node running TCP/IP Services for OpenVMS V5) and **alfman** (a node running Tru64 UNIX V4.0e).

The `/etc/hosts.equiv` file mentioned on page 2–7 allows systems running TCP/IP protocol stacks to enable a trust relationship with another system (or systems). Any node whose name appears in the file is considered trusted. This means that a user can log in (or use any of the “R” commands: `rlogin`, `rsh`, `rcp`) to the trusting system without having to supply a password, providing there exists an account on the trusting system that matches the name of the account on the trusted system. The equivalent capability on OpenVMS is enabled through proxy accounts.

If available, use screen displays to present these commands. Show the commands in action.

The end of the module introduces some UNIX style commands. If your audience has not been using UNIX at all, these commands will seem strange. Tell them that the function of the various commands can be achieved by a more traditional UCX style command. It will be their choice.

If the audience is customer support personnel within Compaq, they should make the effort to get familiar with both styles.

On page 2–17, explain the way the ARP table gets populated. Tell the students that if the first command they enter when they come in tomorrow morning is `arp -a`, they will see nothing in the table. After entering a few `ping` commands, they will see information in the table. So, basically, the IP software checks to see if there is information in the arp table to translate an IP address into a MAC address. If not, broadcast the request to all nodes within the subnet. The one containing the appropriate IP address will be the only one to respond (hopefully). The system then caches the information in the ARP cache for approximately 20 minutes since its most recent use.

On the display screen, enter the `arp -a` command (page 2–17). Then `ping` a system not shown in the `arp -a` command output. The first round-trip statistic will be longer than the rest. This is due to the `arp` activity described above. Now enter the `ping` command again, and you should see no extra time spent on the first `ping` display as opposed to all the others.

The `sysconfig` command is shown in more detail in later modules. Mention that it displays configured subsystems (network-related only). On a Tru64 UNIX system, the `sysconfig` command displays approximately 30 subsystems. The attributes shown on pages 2–19 and 2–20 are describing a characteristic of TCP/IP on your system. Many of these will be dynamically changeable. In this way, they resemble dynamic SYSGEN parameters.

Key Points

Be sure the students understand the following:

- The system supports both UCX style commands and UNIX style commands.
- The relationship of the commands introduced in this module and the communication protocol information presented in the last module.
- The trust that can exist between an OpenVMS system and a UNIX system.
- How to use FTP to transfer files.
- The `arp` command and the ARP cache.
- The `sysconfig` command as a UNIX style alternative to SYSGEN for dynamic attributes pertaining to the network.

Time Required to Teach

This module should take approximately 35 to 40 minutes to cover.

Installing and Configuring TCP/IP Services — Module 3

Instructor Notes

This module covers the installation and configuration of TCP/IP Services for OpenVMS. Most of the module shows screen captures of the TCPIP\$CONFIG utility.

Briefly mention the license PAK issue so that new customers are aware of the necessity to license this product. If you get any comments about the fact that most other TCP/IP implementations are free (which they are), mention that the license fee is a Compaq business decision and you can only discuss it peripherally.

The disk space numbers on page 3–2 are from an SPD. Direct any further inquiries about disk space to the most recent SPD.

The installation is shown on page 3–3. Make sure to note the suggestion that students **not** delete any files left over from a previous version of the product. During the configuration, they will be asked if they want to use existing files to generate the new databases. It could save a lot of time for some customers. Note the list on the bottom of page 3–5.

TCPIP\$CONFIG is a menu-driven utility, although the syntax to use it without the menus is shown in the middle of page 3–6.

The main menu breaks the configuration into: Core, Client, Server, and Optional components. The core would be configured first, followed by any or all of the other options. Make sure the students understand that they can use this utility to simply start or stop the services if desired.

Students are asked to configure their machines in lab. The machines should have little or no services on them at the beginning of the course. If this is not possible, have the students reconfigure their interfaces to use different (instructor-chosen) IP addresses and subnet masks, using the default mask for the first day's lab.

Most of these configuration samples are covered again when the actual module discussing the software is presented. You may want to give a brief (very brief) introduction to the software components as the configuration pages are visited.

Page Number	Software Component	Description	Used for:
3-8	Interfaces		Network interface configurations, IP address, subnet mask, and so forth.
3-9	FTP	File transfer protocol	File copies. (Note: the display for FTP on the writer's V7.2 system showed SERVER when it should have showed CLIENT.)
3-10	LPD	Line printer daemon	Remote printing.
3-12	NFS	Network File System	Transparent file system or directory sharing.
3-12	RSH	Remote shell	Executing commands on a remote system.
3-12	REXEC	Remote exec	Executing commands on a remote system.
3-13	RLOGIN	Remote login	Remote login capability.
3-13	SMTP	Simple mail transport protocol	Remote mail.
3-14	Telnet		Remote logins.
3-16	BIND	Berkeley Internet Name Domain	Node name to IP address translation (and the reverse).
3-16	BOOTP	Boot Protocol	Remote booting.
3-16	TFTP	Trivial File Transfer Protocol	Downloading of files.
3-20	PCNFS		NFS with personal computers.
3-20	PORTMAPPER		Mapping of remote procedure call (RPC) numbers to IP port numbers.
3-21	SNMP	Simple Network Management Protocol	Remote network management.
3-22	NTP	Network Time Protocol	A common view of time across the network.
3-22	METRIC SERVER		Dynamic TCP/IP cluster load balancing.
3-23	POP	Post Office Protocol	Mail delivery to laptops.
3-23	FINGER	Remote finger command	Getting information about users.
3-23	RMT		Remote magnetic tape access.
3-24	LBROKER		Dynamic TCP/IP cluster load balancing.

The students should understand that many of the changes they make through this utility do not take effect until the software is stopped and restarted.

The last page (page 3–29) shows how to get the services to start automatically at boot time. Make sure the students know that if they **do not** do this, the software will not automatically start after the system has booted. They can start it by using the TCPIP\$CONFIG menu or entering the @SYS\$STARTUP:TCPIP\$STARTUP command. They can also use @SYS\$STARTUP:TCPIP\$SHUTDOWN to shut down the services.



Note

The Configuration lab comes before the Commands lab to let the students actually use some of the commands.

Key Points

Be sure the students understand the following:

- A license is required for this product.
- The installation preserves files from a previous version.
- How to use the PRODUCT INSTALL command and the approximate amount of disk space necessary for the installation.
- How to use the TCPIP\$CONFIG utility to configure services, start the services, and stop the services.
- How to use the command procedures TCPIP\$STARTUP and TCPIP\$SHUTDOWN in SYS\$STARTUP to start and stop TCP/IP without going through the menus.

Time Required to Teach

This module should take approximately 35 to 40 minutes to cover.

Protocols — Module 4

Instructor Notes

This module delves deeper into the world of the protocol stack. The main point to make is that there is no magic. Everything is orderly and takes place because of established number schemes and common software.

Make sure the students realize that when their machine is attached to a broadcast medium (like Ethernet), they will not be taking a CPU interrupt for each datagram that flows down the wire. Some decisions are made by the hardware itself (page 4–2). Only when the hardware sees a destination MAC address that matches the MAC address on the interface card (or is a broadcast MAC address) is an interrupt caused.

That interrupt causes the communication driver's interrupt service routine to execute. The driver examines the datagram and deciphers the header to figure out which upper level network protocol gets this data (DECnet, IP, LAT, or others). All of the network-level protocols are identified by numbers held in the type field of the datagram (page 4–3).

Once the network layer software is presented with the data, it searches its header for a transport protocol number (TCP or UDP). IP passes the data (minus its header) up to TCP, which checks for a port number indicating the application that gets this data. Make sure students can follow that sequence.

The Ethernet picture on page 4–3 is actually showing the RFC 894 Ethernet version. It contains the type number at bytes 12 and 13. The 802.2 and 802.3 (RFC 1042) version of this picture would show the type number at bytes 20 and 21. Do not make a big deal out of this. The point is that the driver can figure out what network-level software to deliver the data to by looking at type information held in the header fields of the datagram.

The IP header is shown with the protocol field bolded on page 4–5. This is to emphasize that this is all tied together rationally. Feel free to present a brief introduction to TCPTRACE at this time. Get a Telnet session (uses TCP and IP) going and look at a few of the packets.

The TCP header is shown on page 4–6 with the **Port number** field bolded. Point out several of the well-known ports listed under the first bullet. Check for port 23 on your TCPTRACE command output. If the group would benefit, describe the TCP header fields.

Header Field	Description
Source Port	Identifies the sending application.
Destination Port	Identifies the receiving application.
Sequence Number	Identifies this packet.
Acknowledgement	Identifies the last packet we received.
Header Length	When does the data start?
URG	Packet is urgent.
ACK	Packet contains an acknowledgement number.
PSH	Push this packet to the front of the queue.
RST	Reset the connection.
SYN	Synchronize so we can talk.
FIN	Finish this connection.
Window Size	How much unacknowledged data can we accept?
Checksum	Cyclic redundancy check for data integrity.
Urgent Pointer	Where in the data is the urgent stuff?
Options	Used if there are any additional header fields.
Data	Data

The IP protocol header fields are discussed on page 4–13. Take students through it field by field. Make sure they are aware that the mysterious TOS field (pages 4–14 and 4–15) is not used in many implementations. Students can find additional reading on these fields in the Stephens and Comer books.

Key Points

Be sure the students understand the following:

- The details of the data travelling up or down the protocol stack.
- No magic! Device drivers, kernel software, and applications do all the work.
- Interrupts are filtered by the hardware interface.
- IP header fields and the TCP header fields.
- The difference between TCP (connection oriented, data acknowledgement, check-summing for data integrity) and UDP (datagram based, no checksum, less overhead, no acknowledgements).
- How to use TCPTRACE output.

Time Required to Teach

This module should take approximately 55 to 65 minutes to cover.

NFS — Module 5

Instructor Notes

Present NFS to the students as a heavily used file system (or directory) sharing mechanism. Develop the simple diagrams on the first few pages to make sure that the students understand what NFS accomplishes for the customer. NFS is one of the most important network services.

Make sure the students realize that this implementation uses NFS V2. Tru64 UNIX uses NFS V3 (or V2 if necessary). The essence of NFS is the same no matter what the version.

The difficulty with NFS V3 in the OpenVMS system may be with file locking. NFS implements locking through daemon processes. OpenVMS uses the kernel-based distributed lock manager.

Be sure to point out the disk savings shown by contrasting pages 5–2 and 5–3.

The students should understand that NFS was written by Sun Microsystems, and that their software culture developed around using remote procedure calls (RPCs) and not sockets (page 5–5). Most other network software is built around sockets and ports. To use the Sun based NFS software, a daemon process was put in place to translate (map) the RPC numbers to port numbers (thus the name PORTMAP).

Be careful to point out the identification issues between UNIX and OpenVMS (page 5–6). Proxy accounts are used to handle the identification issues.

OpenVMS has a complex file system supporting many different types of files (sequential, relative, indexed-sequential) while UNIX supports sequential only. To handle these and other differences, the OpenVMS NFS server can serve out traditional OpenVMS file systems or UNIX style file systems called container file systems (CFS) because they contain the UNIX semantics (page 5–7).

Be sure to distinguish the two types of proxy accounts (page 5–10). Proxy means “the authority or power to act for another” (*Merriam-Webster Concise Dictionary*).

The proxy types (page 5–10) are not intuitive, so be sure to point out the list at the bottom of the page. They are shown as part of the output from `SHOW PROXY`.

The client configuration example starting on page 5–11 uses a UNIX system for the server. Note the quotes on the `mount` command (page 5–14). They are important because the request is being sent to a UNIX system, which is case sensitive.

Be sure to point out that the successful `mount` command on page 5–17 uses the `/uid` and `/gid` qualifiers to identify the issuer. This is redone on page 5–22 using proxies to eliminate the qualifiers.

Focus the students on the file ownership issues shown on page 5–19. A file created by the system account on the OpenVMS client is owned by user “nobody” from the UNIX server perspective. However, a file created by a standard user (obrien) reflects ownership as expected. NFS assumes that you do not want to allow root access to the remote root user (for security reasons). It therefore maps the root account to the user account “nobody.” The root user on OpenVMS is *system*, so a file created by system shows as owned by *nobody*. Be aware that nobody is really somebody. There actually is a *nobody* account on the UNIX side.

Focus the students on the last line of page 5–19. The class will make a change to that line to eliminate the nobody ownership issues stated in the previous paragraph. Contrast with the last line on page 5–23.

To handle the extensive OpenVMS file characteristics not found within UNIX, the NFS client software creates a parallel file to hold the extra file information (page 5–23).

Make sure the students know how to configure an OpenVMS NFS server (pages 5–26 through 5–28). To support the UNIX style syntax in a client mount request (page 5–29), tell the students about the MAP command. It maps devices and directories to UNIX style directory specifications (fake mount points). Be sure the students can map and export a file system or directory (page 5–30).

Make note of the extra steps necessary to export a container file system (page 5–35). Proceed slowly through these examples no matter what time of day it is. The concept of a container is counter intuitive, but explain that it is a solution for UNIX clients wanting to use an OpenVMS server and retain the simplicity of the UNIX files.

Key Points

Be sure the students understand the following:

- NFS is a client/server oriented solution for distributing disk-based data over a network.
- This is transparent to the users and applications (but not to the administrators).
- The function of the PORTMAPPER (maps application port number to RPC numbers).
- How to manipulate proxy accounts, which are used to identify NFS users.
- CFS allows serving an OpenVMS file system entity that conforms to UNIX file system rules.
- Use TCPIP\$CONFIG to set up the OpenVMS system as a client or as a server (or both).
- File ownership and access permissions (protections) can be counter intuitive in the case of root (system) access. By default, access by the *system* user is interpreted as access by the *nobody* account.
- Use the MAP command to associate UNIX semantics to an OpenVMS device or directory.
- Container files require double mapping: one for the disk and one for the directory.

Time Required to Teach

This module should take approximately 65 to 80 minutes to cover.

Serial Lines — Module 6

Instructor Notes

Students should be able to configure a serial line using SLIP to complete the second part of the second day's labs. This short module provides details for setting up the SLIP connection. It also provides PPP concepts and syntax, and contrasts PPP with SLIP.

Make sure the students are aware of the point-to-point nature of serial line connections. Until now, most of the discussions have assumed an Ethernet or some other broadcast style of media. Point out that this module involves two computers communicating through their serial ports—no extra hardware (unless modems are involved).

You should provide several serial cables (like DECconnect Office Cable E116273), and you will probably need an adapter (H8571-J) to connect the cable to a nine-pin serial port. This varies from system to system, but basically you wire a serial port (typically used for a terminal connection or a printer connection) directly to a serial port on another machine. This eliminates modems from the lab necessities.

Be sure to mention that for a long distance connection, the configuration would be the same except that the serial line would connect to a modem. The modem would connect to the phone line, and then to the remote modem, which would connect to the serial line on the remote system. If available, and if the students' interest and your expertise merit, add the modems in the lab. See Section 3.3.2 of the *TCP/IP Services for OpenVMS Management Guide* for additional information.

Make sure the students understand the simplistic nature of the SLIP encapsulation (page 6–3), and can contrast it with the PPP encapsulation (page 6–4).

Students should carefully examine the SLIP configuration example on pages 6–6 through 6–12. The first major point is on the bottom of page 6–6 where the SET INTERFACE command is shown. That is how the serial line (TTA0:) is associated with IP. Make sure students understand that before this point, the terminal device driver had control of TTA0:. Now the network software is working in conjunction with the terminal device driver to send the appropriate data over the wire.

Ask the students about MAC addresses and ARP. Are they pertinent here? There is no need for MAC addresses because it is point-to-point. The data goes out the wire to whatever is at the other end of the wire. Because it is not broadcast oriented, and there are no MAC addresses, there is no need for ARP to hunt for low-level identification.

Make sure the students know that serial lines are typically slower than Ethernet (see the ping round-trip times on page 6–9).

Routing is covered in the next module, but you can introduce it by explaining why the `ping` command on the top of page 6–7 failed. There has to be some information in the routing table to describe how to handle data destined for the remote system. The SLIP lines were configured with a different network address (see page 6–6, note that the network portion of the IP address is 192.206.127—not the usual 192.206.126 that has been used so far). This forces the IP software to search the routing table to figure out which interface to use.

Both nodes are enabled to function as routers on page 6–10. Make sure students realize that this is not necessary if all you have are the two systems at either end of the wire. However, if the serial line connects two LANs (as it does in the lab), the nodes at the ends of the serial line must forward packets to their ultimate destination. Make the IP software aware that it is responsible for forwarding packets with the `SET PROTOCOL IP/FORWARD` command (shown on page 6–10).

PPP is presented as an option but is not shown configured. See Section 3.2.1 of the *TCP/IP Services for OpenVMS Management Guide* for additional information and examples of configuring PPP. Make sure the students are aware that PPP can handle more than just the IP protocol, while SLIP is strictly an IP option.

If you need to set up the serial line on a Tru64 UNIX node, use `netconfig` or `netsetup` as follows:

```
# netsetup
```

```
**** MAIN MENU ****
```

```
1 Configure Network Interfaces
```

```
2 Enable/Disable Network Daemons and Add Static Routes
```

```
3 Add/Delete Host Information
```

```
4 Display Network Configuration
```

```
5 Exit
```

Enter the number for your choice: **1**

```
***** CONFIGURE/DELETE NETWORK INTERFACES *****
```

You can configure or delete network interfaces. Configuration information is updated in `/etc/rc.config` and `/etc/hosts`. Choose "configure" or "delete" at the prompt.

Enter whether you want to "**(c)onfigure**" or "(d)elele" network interfaces.

If you are finished, press the RETURN key: **c**

You want to "configure" interfaces. Is this correct [yes]?

You will now be asked a series of questions about the system.

Default answers are shown in square brackets ([]). To use a default answer, press the RETURN key.

This machine contains the following network interfaces:

```
ln0
```

```
sl0
```

```
sl1
```

Which interface do you want to configure [ln0]: **sl0**
You want to configure "sl0". Is this correct [yes]?
Interface "sl0" is a **Serial Line Internet Protocol (SLIP) interface**.
Enter the hostname for SLIP interface "sl0" []: **alfman_sl**
The hostname for SLIP interface "sl0" is "**alfman_sl.vastek.com**".
Is this correct [yes]?
Enter the Internet Protocol (IP) address for SLIP interface "sl0"
in dot notation []: **192.206.127.12**
The IP address for SLIP interface "sl0" is "**192.206.127.12**".
Is this correct [yes]?
Enter the IP address of the **remote SLIP interface**
in dot notation []: **192.206.127.13**
The IP address for the remote SLIP interface is "**192.206.127.13**".
Is this correct [yes]?
Subnetworks allow the systems on a local area network to be on different
physical networks. For the following question, use the default answer
unless the existing local area network is using subnet routing.
If the local area network is using subnet routing, you need to know
the subnet mask.

Enter the subnet mask in dot notation [255.255.255.0]:
The subnet mask for "sl0" is "255.255.255.0".
Is this correct [yes]?
For the following question USE THE DEFAULT ANSWER unless you would like
to add additional flags (found in the ifconfig reference page) to the
ifconfig command. Normally, you will USE THE DEFAULT ANSWER.
Do you want to use additional ifconfig flags for this interface [no]?
The configuration looks like:
 interface "sl0" hostname: "alfman_sl.vastek.com"
 ifconfig sl0 192.206.127.12 192.206.127.13 netmask 255.255.255.0
Is this correct [yes]?

The slattach command attaches a tty line to a Serial Line Internet Protocol (SLIP) interface (see the slattach reference page).

Enter the slattach flags for "sl0" separated by a space.

If you do not want any flags, press the RETURN key:

You do not want any flags. Is this correct [yes]?

Enter the tty line to attach to "sl0" []: **tty00**

The tty line to attach to "sl0" is "tty00". Is this correct [yes]?

The baud rate for tty line "tty00" is "9600". Is this correct [yes]?

The slattach command looks like:

```
slattach tty00 9600
```

Is this correct [yes]?

```
***** UPDATING /etc/rc.config *****
```

```
"sl0" is configured in /etc/rc.config
```

Do you want to configure another network interface [yes]? no

Enter whether you want to "(c)onfigure" or "(d)elele" network interfaces.

If you are finished, press the RETURN key:

```
**** MAIN MENU ****
```

- 1 Configure Network Interfaces
- 2 Enable/Disable Network Daemons and Add Static Routes
- 3 Add/Delete Host Information
- 4 Display Network Configuration

5 Exit

Enter the number for your choice: **5**

To perform additional tasks in setting up the network, see the Network Configuration reference. For the netsetup modifications to take effect, either restart the network services on this system with the following command:

```
/usr/sbin/rcinet restart
```

or reboot this system with the following command:

```
/usr/sbin/shutdown -r now
```

NOTE: If you are going to use the '/usr/sbin/rcinet restart' command, warn the users that the network services on this system are going to be restarted. Also, any NFS filesystems not mounted via fstab or automount will not be remounted.

```

Do you want netsetup to automatically restart the network
services on this system [no]? yes
Stopping Internet services on this system. Please wait...
Unmounting NFS filesystems
Internet name service stopped
Internet services on this system are stopped.
Starting Internet services on this system. Please wait...
Configuring network
hostname: alfman.vastek.com
rwho daemon started
writing to routing socket: File exists
add net default: gateway 192.206.126.1: ioctl returns 17
File exists
Setting kernel timezone variable
Internet name service started
NIS domain name set to puppy_room
ypserv daemon started
ypbind daemon started
NIS passwd daemon started
NIS high-speed transfer daemon started
Mounting NFS filesystems
Can't get net id for host
NFS mount daemon started
NFS export service started
NFS IO service started
NFS Locking: rpc.statd and rpc.lockd started
3 aliases, longest (mailer-daemon) 19 bytes, 53 bytes total
SMTP Mail Service started
Internet management service started
Internet services provided.
Internet services on this system are started.
***** NETWORK SETUP COMPLETE *****

```

Key Points

Be sure the students understand the following:

- Serial lines are not broadcast oriented.
- SLIP handles IP only. PPP can handle more than IP.
- The `SET INTERFACE SL0` command sets up SLIP.
- Serial interfaces need network addresses different from the Ethernet interface network addresses.
- You may need to set up IP forwarding (routing) on the SLIP nodes.

Time Required to Teach

This module should take approximately 30 to 40 minutes to cover.

Routing — Module 7

Instructor Notes

Make sure the students realize that ROUTED and GATED do not do the routing. Their job is to keep the routing tables full of useful information. The act of routing a packet (forwarding it toward its destination) is done by the IP software (page 7–2).

Be prepared for students thinking that a router is a piece of hardware. They will be thinking of Cisco routers and such. Describe those routers as computers with a narrow focus—they handle routing. Students will configure an OpenVMS system to handle routing along with all of its other responsibilities.

A typical router has more than one network interface (page 7–3). Make sure the students are comfortable with the decision-making process described on pages 7–4 and 7–5. Make sure the students can interpret the output of `SHOW ROUTE` and `netstat -rn` (page 7–7). The `netstat -rn USE` column is particularly enlightening for troubleshooting. It counts the number of packets going out of (or into) that interface.

Distinguish static routing from dynamic routing. The first two routing labs are static oriented (page 7–8)—no ROUTED or GATED. Force the students to deal with the routing table by hand. Then they will appreciate what ROUTED and GATED do for them. The students should **not** place entries in the TCPIP\$ROUTE database (page 7–9) until they have suffered through having to recreate the routing table several times after a restart.

The dynamic routing daemon processes (ROUTED and GATED) are presented on pages 7–11 through 7–18. Make sure the students realize that ROUTED is older and simpler than GATED, handles RIP only, and is the most common dynamic routing daemon. GATED is more complex, handles more protocols, and is not used as often as ROUTED (probably due to administrator inertia).

The routing labs involve three different approaches:

- Tuesday's lab involves getting static routing going between different networks (assigned by the instructor) using serial lines to connect the tables in the room.
- Wednesday's routing lab involves static routing also, but uses subnets. The whole room will be in one network but will be subnetted—thus requiring subnet routing.
- Thursday's routing lab uses the previous configuration and includes dynamic routing (using ROUTED). This lab can be altered to using GATED, but you should start with ROUTED and progress to GATED if time and interest allows.

The configuration example (starting on page 7–20) shows static routing being enabled. Make sure that the students realize that if they are using a nondefault subnet mask, they **must** specify the mask in the command to add a route (page 7–8). The only time this is not true is when adding a default route. When troubleshooting during the labs, especially Wednesday's lab, check the USE count field of a configured route. If it shows 0, or no increment when you think it should be incremented, delete the route and add it again specifying the subnet mask on the command line (`route add -net ... netmask 255.255.255.224`).

Emphasize the power and complexity of GATED by visiting the GATED configuration file on page 7–17. Several of the protocols mentioned (OSPF and RIP) are discussed in Appendix C of the Student Guide. Depending on the interests and abilities of the class, you may decide to cover Appendix C on Thursday. You should cover the material in this module at least one day before you cover Appendix C.

Point out that the default route appears in some displays as 0.0.0.0 (see page 7–29). TCPTRACE is used at the end of the module to watch the dynamic routing daemons communicate.

Key Points

Be sure the students understand the following:

- Routing is not done by ROUTED or GATED. They dynamically update the routing tables.
- All nodes have routing tables. Routers typically have more information in their routing tables.
- You can configure UNIX and OpenVMS nodes to route packets along with their other responsibilities.
- IP software actually does the routing.
- Dynamic routing provides for changes to the routing table automatically. Static routing is done by hand.
- Static routes can be placed in a file, TCPIP\$ROUTES.
- GATED supports multiple protocols; ROUTED supports RIP only.
- Nondefault subnet masks require special attention when adding static routes.
- TCPTRACE can monitor communications between dynamic routing daemons on port 520.

Time Required to Teach

This module should take approximately 60 to 70 minutes to cover.

DNS and BIND — Module 8

Instructor Notes

Emphasize the fact that this implementation uses BIND v8.1.2. If the students have some experience with BIND, it will probably be with v4.n. The nomenclature and files have changed. Be sure they are aware that they may hear *primary master* and *secondary master* during BIND conversations. Those are V4 terms for *master* and *slave*, which are the V5 terms.

Emphasize the big picture. What is this software designed to do? How does it help in the grand scheme? Make sure they are aware of the history of BIND (pages 8–3 through 8–5) because the history explains the point of the software—to provide a scalable node-name-to-address (and reverse) translation scheme.

Be sure students know that the clients do not run a separate daemon process; their applications run software within libraries (called resolver routines).

The concept of a zone as opposed to a subdomain can get confusing (page 8–6). Explain that subdomains break up the name hierarchy, while zones break up the administrative hierarchy.

Make sure students know that the master server has the source files. Slave servers can have files also, but they are loaded from the master and are used if the master is not available.

Make sure the students know the format of the resource records that make up the contents of the files in the BIND database (pages 8–9 and 8–10).

Discuss the types of servers (pages 8–12 and 8–13). Master and slave concepts are the most important, but make sure they understand caching-only and forwarding servers as well.

The first example has the OpenVMS node functioning as a resolver (client). Note that the resolver (node den255) can ping node COMPCOM despite the fact that there is no information about COMPCOM in the local database (page 8–24).

Introduce NSLOOKUP as a troubleshooting tool (pages 8–26 through 8–30). Appendix B contains a list of NSLOOKUP commands.

The materials include a sample master server configuration on an OpenVMS node starting on page 8–31. Direct students' attention to the files named at the bottom of page 8–33. These files will contain the resource records. Mention that if a customer has preexisting configuration files (from UCX), they are converted as you configure the server. The conversion omits the NS records from the new files, so be aware that you must add those later.

Point out that the TCPIP\$BIND_RUN.LOG file is useful if you are having configuration problems (page 8–35). Point out that students must create their own BIND configuration file from the supplied template (page 8–36). Spend some time looking at the resource records in the various files (pages 8–37 through 8–53). The supplied configuration example shows the administrator correcting the various databases (mainly by adding the missing NS resource records). NSLOOKUP is used also to look at the resource records.

Make sure the students are aware of the meaning of a TCP/IP *cluster*. They will naturally think of an OpenVMS Cluster. Remind them that a TCP/IP cluster is a group of TCP/IP nodes all being referenced by the same name so that the processing load can be distributed by the name server. The server takes the next address in the list and presents it to the resolver (client) to use for this service request. The load broker and metric server (page 8–56) enable more sophisticated load distribution based on the actual load on a series of nodes (as opposed to a simple round-robin approach).

Key Points

Be sure the students understand the following:

- This is a new version of BIND (v8.1.2).
- BIND provides node-name-to-address translations (and reverse translations).
- The client runs resolver software from library routines (not a daemon process).
- The dot (.) is the top of the name hierarchy.
- Subdomains break up the name space. Zones break up the administrative space.
- The master server sends files to the slave through a zone transfer.
- The slave asks for a zone transfer when the serial number field in the SOA resource record is lower than the serial number reported by the master server. The master server should update the serial number each time a change is made to the master file.
- Files consist of resource records.
- You configure the resolver and server with TCPIP\$CONFIG.
- NSLOOKUP is the main troubleshooting tool.
- A TCP/IP cluster is not an OpenVMS Cluster. You can use a TCP/IP cluster for load balancing. Metric server and load broker provide more sophisticated load balancing (available to V8 nodes only).

Time Required to Teach

This module should take approximately 60 to 75 minutes to cover.

BOOTP — Module 9

Instructor Notes

Focus the students on the fact that a machine with no operating system can download an operating system using this protocol. Make sure they appreciate the enormity of the task.

Emphasize the role of the server. It must be aware of the possibility of a download request. It must also be aware of which machines it has the responsibility to download, and which software goes to which system.

Review the communication on page 9–3. Note that TFTP is the protocol that actually sends the boot file. Note that the client must have an interface (Ethernet) that understands BOOTP, TFTP, and some IP as well. Be aware that not all Ethernet interfaces support this activity. An inexpensive card probably will not have this support.

Examine the configuration example based on the needs and interests of the students. Make sure they realize the importance of the BOOTP configuration file, the MAC address used as client ID, and the ports used during the communication.



Note

This service is undergoing some changes and will probably have a different look in a future release.

Key Points

Be sure the students understand the following:

- The client requests a download from a console command (`>>>b -p bootp ewa0`).
- The server responds with a download of the operating system for the client.
- The client is identified by its MAC address.
- The server records MAC address, boot file name, and location in the server database (TCPIP\$BOOTP).
- Use TFTP to download the actual boot file (not BOOTP).

Time Required to Teach

This module should take approximately 30 to 35 minutes to cover.

DHCP — Module 10

Instructor Notes

Make sure the students understand the point of the protocol—serving out IP addresses dynamically to various machines for a finite amount of time (effectively leasing the IP addresses).

Point out the advantages of using DHCP (page 10–4). Point out the plethora of configuration files (page 10–6). The key file is the DHCPCAP file (page 10–8). This file is incredibly cryptic (thanks to our UNIX colleagues). The key to understanding the DHCPCAP file is in the DHCPTAGS file (page 10–9).

The bottom of page 10–12 shows the one-line NETS file. It contains the range of addresses to be distributed.

The commands set up on pages 10–14 through 10–16 allow communication with the DHCP daemon and allow convenient starting of the DHCP GUI (shown on pages 10–20 through 10–24). Think of the signals mentioned on page 10–15 as the (rough) UNIX equivalent of an AST.

Do not overlook the support commands on page 10–16.

Point out that DHCP can be set up in an OpenVMS Cluster such that a DHCP server process can be available when the currently active DHCP server node goes down (pages 10–25 and 10–26).

Key Points

Be sure the students understand the following:

- DHCP leases IP addresses to clients and centralizes IP address administration.
- DHCP identifies clients by MAC address (similar to BOOTP).
- DHCP is supported by six configuration files on the server (DHCPCAP, DHCPTAGS, NAMEPOOL, NETMASKS, NETS, and SERVER.PCY).
- Create DHCP specific foreign commands by running `SYSS$MANAGER:DHCP_SETUPCOMMANDS.COM`.
- The GUI is accessible through the `dhcpageui` or `xjoin` commands.
- DHCP supports OpenVMS Cluster failover.

Time Required to Teach

This module should take approximately 40 to 45 minutes to cover.

NTP — Module 11

Instructor Notes

Again, it is important to focus the students on the big picture. Multiple nodes in a network must have a coordinated and synchronized view of time. The solution is to settle on a single definition of the current time and serve that out to the clients.

The job becomes complicated when issues such as system time drift, network speed differences, and software vagaries are factored in. NTP provides an Internet-based solution for this problem.

Be sure that the students understand the concept of *stratum*, indicating the level of accuracy of the server's view of time.

Point out the World Wide Websites willing to serve out time to clients (page 11–5). Unless the lab environment has access to the Internet, you will appoint one system as the local reference clock and use it as the time server.

Emphasize that a system can function as both a client and a server (page 11–6). Make sure the students are aware of the template file (TCPIP\$NTP.TEMPLATE) and that they will create a configuration file from it (page 11–7).

Point out the troubleshooting commands NTPTRACE, NTPDC, NTPQ, and, of course, TCPTRACE.

Key Points

Be sure the students understand the following:

- How a system's time is updated (clock interrupts 1024/sec on most Alpha systems).
- Stratum number indicates the accuracy of the system's time. The lower the number, the more accurate the time
- Many systems on the Internet are willing to serve out highly accurate time.
- You can configure NTP in three modes: client/server (peer keyword in configuration file), client (server keyword in configuration file), and broadcast (broadcast keyword in configuration file).
- The template file (TCPIP\$NTP.TEMPLATE) must be customized into your own configuration file (TCPIP\$NTP.CONF).
- NTP is supported by a drift file (TCPIP\$NTP.DRIFT) to handle time errors in the frequency of the CPU hardware clock.
- NTP is also supported by a log file (TCPIP\$NTP.LOG), which tracks time adjustments.
- NTPTRACE and other commands (TCPTRACE, NTPQ, and NTPDC) support NTP troubleshooting.

Time Required to Teach

This module should take approximately 30 to 35 minutes to cover.

SNMP — Module 12

Instructor Notes

Make sure the students realize that the agents (of which there are many) are the servers. They serve out their information to the client (of which there may be just one). Therefore, the normal client/server terminology applies, but the numbers are the reverse of what you usually expect (many clients and a few servers). The SNMP environment usually has many servers (agents) and few clients (NMS—Network Management System).

Point out that the normal communication between the agents and the NMS involves *get* requests (to get some information from an agent), or *set* requests (to set some value in the agent). Both of these are issued from the NMS; however, an agent can initiate a *trap*, which sends an asynchronous message to the NMS (see pages 12–3 and 12–4).

Emphasize that the SNMP is accompanied by several other protocols describing the data to be transferred (pages 12–5 through 12–8). Take your time through these pages. The students will be mystified as all the protocols are introduced (SNMP, MIB, ASN.1). Explain that it was necessary to define a common way of representing the data being requested or set. The actual locations in the kernel for the various data items would be different on each platform. The MIB and ASN.1 protocols allow a standard, human readable way of representing data, and also provides a convenient numeric representation.

Examine the .MY files in [SYSHLP.EXAMPLES.TCPIP.SNMP] to see how the values are set up.

Note that the OpenVMS implementation of the SNMP agent includes master agent software and subagents. The subagents are responsible for a portion of the MIB (page 12–9). The master agent listens on port 161 (page 12–11) and distributes the incoming requests to the appropriate subagent.

For troubleshooting, direct the students to the log files (page 12–12). TCPIP\$SNMP_RUN.LOG is shown on pages 12–14 through 12–18. It shows the steps involved in starting SNMP.

Point out the configuration file (TCPIP\$SNMP_CONF.DAT) and how you can alter it using SET CONFIG SNMP commands (page 12–19). Mention that the community name is functioning as a vague password. You have to know the community name to get information from an agent.

If a full NMS is not available, use the SNMP_REQUEST command to mimic what a NMS might send to the agent systems (pages 12–22 and 12–23). Watch the action using the TCPTRACE command (page 12–25).

A portion of a .MY file is included at the end of the module (pages 12–29 through 12–33) for easy reference. Note how the MIB numbers come to life as you see them being defined (page 12–29). A few individual objects on the following pages show how they are defined. There are many more objects defined in the actual files. Have the students use the TYPE/PAGE command on a few of them.

Key Points

Be sure the students understand the following:

- The agents are the servers and the NMS is the client.
- A trap is an example of communication started by the agent. All other communication is started by the NMS.
- MIB and ASN.1 are protocols related to SNMP. SNMP defines the transfer encapsulation. ASN.1 (Abstract Syntax Notation) provides (allegedly) a human readable version of the MIB numbered data.
- Variables are defined in .MY files, examples of which can be found in subdirectories under SYS\$EXAMPLES.
- The master agent dispatches incoming requests to the appropriate subagents.
- Log files in SYS\$SYSDEVICE:[TCPIP\$SNMP] are useful for troubleshooting. Start with TCPIP\$SNMP_RUN.LOG.
- The Community name serves as a password to the agent.
- The SNMP_REQUEST command allows “poor man” NMS experiments.
- SNMP uses port 161.

Time Required to Teach

This module should take approximately 30 to 35 minutes to cover.

Remote Printing — Module 13

Instructor Notes

Point out that there are three print options, although the module primarily focuses on LPR/LPD. This option sets up printer characteristics in `SYSS$SPECIFIC:[TCPIP$LPD]TCPIP$PRINTCAP.DAT`. This file is similar to the UNIX `/etc/printcap` file. There is a utility similar to the UNIX `lprsetup` as well (`SYSS$SYSTEM:LPRSETUP.EXE`).

Show the students the local printer configuration (pages 13–4 through 13–9), which introduces the commands and files associated with LPR/LPD printing. Make sure you point out that all the configuration activities do nothing for you until you associate the queue with a device (page 13–10) using the DCL command `INIT/QUEUE/START/ON=TTA0: ...`

Focus briefly on the simple troubleshooting technique shown on page 13–12 (`TYPE /OUT=tta0:`). If that works, your problem is at the queue level. If it does not work, there are connection problems or hardware problems, so do not bother fussing with the queue yet.

Point out that remote printing requires the remote system name and the remote queue name (page 13–14). This will be the basis of student access to the printer service on the remote system.

Focus on the `PRINTCAP.DAT` file contents shown on page 13–17. It clearly shows the remote print queue name as *alfprint* and the remote node name as *alfman* (that is, `rm=alfman;`, `rp=alfprint:`).

The `TCPTRACE` output (page 13–20) shows the remote print request using port 515. Use the trace to solidify the students' understanding of remote printing. Be sure to point out that printer access can be controlled using the commands shown on page 13–23.

Briefly introduce `TELNETSYM` and PC-NFS printing. Point students to Modules 19 and 20 in the *TCP/IP Services for OpenVMS Management* manual for more information.

Key Points

Be sure the students understand the following:

- There are three printing options: LPR/LPD, TELNETSYM, and PC-NFS.
- LPRSETUP provides an interface to the PRINTCAP file, which holds printer information.
- Use LPRSETUP to set up local or remote print queues.
- Remote printer configuration requires knowledge of the remote queue name and the remote node name.
- Printer access control is available through the ADD PROXY command and the SET SERVICE command.
- TELNETSYM and PC-NFS are other printing options.

Time Required to Teach

This module should take approximately 30 to 35 minutes to cover.

Troubleshooting — Module 14

Instructor Notes

Point out the list of necessities for configuration troubleshooting: command file, port number, process name, and account information. Go through the list of configuration options on pages 14–4 through 14–5. Focus on the FTP logical names at the bottom of page 14–4 because students will use them in lab.

The suggestions on pages 14–6 through 14–7 are worth reviewing. In particular, focus on the last comment on page 14–7. Simple and effective tools are available.

Make sure you focus students on the routing problems (page 14–10). Most problems eventually involve routing, so the more comfortable students are with routing concepts, the better equipped they are to handle common network problems.

Performance tuning is always of great interest to students. Fortunately, the `sysconfig -r` command provides an easy way to alter network attributes. Do not spend too much time on this. Focus students on the suggestions at the beginning of the module—have them change one attribute at a time and retest the application.

Key Points

Be sure the students understand the following:

- There are two levels of troubleshooting: configuration level and network level.
- Examine the service for configuration-level trouble (`SHOW SERVICE/FULL ...`).
- Use logical names to alter FTP behavior.
- Examine log files for NFS and almost all services.
- Many tools are easily available for network-level troubleshooting: `ping`, `netstat`, `tcptrace`, `ripquery`, `arp`, `nslookup`, `ifconfig`, and `REPLY/ENABLE=NETWORK` are all excellent.
- You can change performance attributes using the `sysconfig -r` command.

Time Required to Teach

This module should take approximately 30 to 35 minutes to cover.

SMTP — Module 15

Instructor Notes

Point out the relationship between the mail protocols shown on page 15–2. Most students will have heard of these protocols, but will not have considered the relationships.

SMTP consists of a few simple commands delivered to an SMTP application. The commands (page 15–2) are interpreted and a response is sent back. On page 15–3, the example mimics a mail sender by using Telnet to communicate to the SMTP receiver. Note the simple communication that ensues.

Be sure to point out the special OpenVMS mail syntax needed to use SMTP (page 15–5).

Point out the involvement of BIND in SMTP. The BIND servers contain MX records, which designate mail exchange nodes to be used if the destination address is not currently accessible (page 15–8).

Key Points

Be sure the students understand the following:

- SMTP is the backbone of most email systems.
- You can use Telnet to mimic SMTP communications.
- OpenVMS Mail requires special syntax for SMTP (SMTP%obrien@alfman).
- Incoming mail is stored in SYSS\$SPECIFIC:[TCPIP\$SMTP].
- POP retrieves mail from the server.
- Create MX records with the SET MX_RECORDS command.
- MIME allows users to send more sophisticated data in their mail messages.

Time Required to Teach

This module should take approximately 30 to 35 minutes to cover.

