



Sun Educational Services

Solaris Operating Environment – TCP/IP Network Administration

SA-387





Copyright 2000 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303, U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun Logo Sun, Solaris, SunATM, Sun Quad FastEthernet, SunFastEthernet, SunFDDI, SunTRI, Solstice AdminSuite, SunNet Manager, OpenWindows, and SunSoft are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

The OPEN LOOK and Sun Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

U.S. Government approval required when exporting the product.

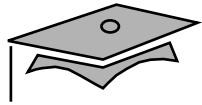
RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Govt is subject to restrictions of FAR 52.227-14(g) (2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015 (b)(6/95) and DFAR 227.7202-3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS, AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.



Sun Educational Services

About This Course



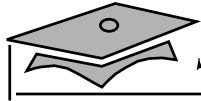
Course Goal

The *Solaris Operating Environment – TCP/IP Network Administration* course teaches you the advanced administration skills required to plan, create, administer, and troubleshoot a local area network (LAN).

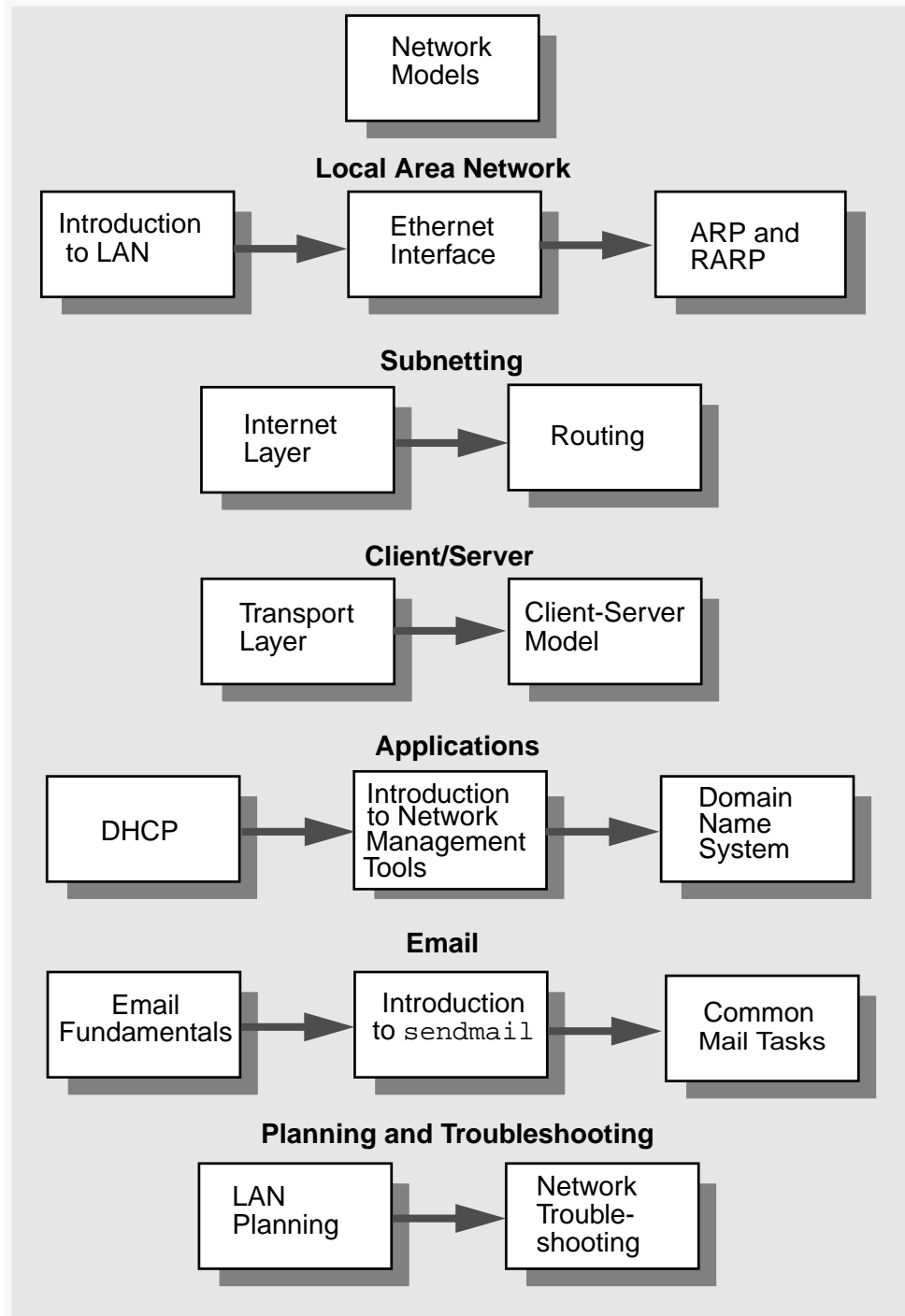


Course Overview

- Hands-on experience with:
 - Network configuration
 - Network planning
 - Network troubleshooting
- Topics include:
 - Domain Name Service (DNS)
 - Sendmail
 - DHCP



Course Map





Module Overview

- Module 1 – "Network Models"
- Module 2 – "Introduction to Local Area Networks"
- Module 3 – "Ethernet Interface"
- Module 4 – "ARP and RARP"
- Module 5 – "Internet Layer"
- Module 6 – "Routing"



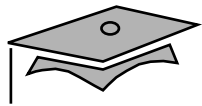
Module Overview

- Module 7 – "Transport Layer"
- Module 8 – "Client-Server Model"
- Module 9 – "DHCP"
- Module 10 – "Introduction to Network Management Tools"
- Module 11 – "Domain Name System"
- Module 12 – "Email Fundamentals"



Module Overview

- Module 13 – "Introduction to sendmail"
- Module 14 – "Common Mail Tasks"
- Module 15 – "LAN Planning"
- Module 16 – "Networking Troubleshooting"



Module Pacing

Module	Day 1	Day 2	Day 3	Day 4	Day 5
"Network Models"	A.M.				
"Introduction to Local Area Networks"	A.M.				
"Ethernet Interface"	P.M.				
"ARP and RARP"	P.M.				
"The Internet Layer"		A.M.			
"Routing"		P.M.			
"The Transport Layer"			A.M.		
"The Client-Server Model"			A.M.		
"DHCP"			P.M.		
"Introduction to Network Management Tools"			P.M.		
"Domain Name System"				A.M.	
"Email Fundamentals"				P.M.	
"Introduction to sendmail"				P.M.	
"Common Mail Tasks"				P.M.	
"LAN Planning"					A.M.
"Networking Troubleshooting"					P.M.



Topics Not Covered

- Solaris™ system administration
- Server storage administration
- NIS+
- Solaris OS tuning



How Prepared Are You?

- Perform basic host operations?
- Manipulate startup and shutdown scripts?
- Install and configure user accounts?
- Install system software packages?



Introductions

- Name
- Company affiliation
- Title, function, and job responsibility
- Networking experience
- Reasons for enrolling in this course
- Course expectations



How to Use Course Materials

- Course map
- Relevance
- Overhead image
- Lecture
- Exercise
- Check your progress
- Think beyond



Module 1

Network Models



Overview

- Objectives
- Relevance



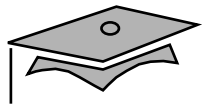
Network Models

- ISO/OSI reference model
- TCP/IP suite (TCP/IP model or TCP/IP)

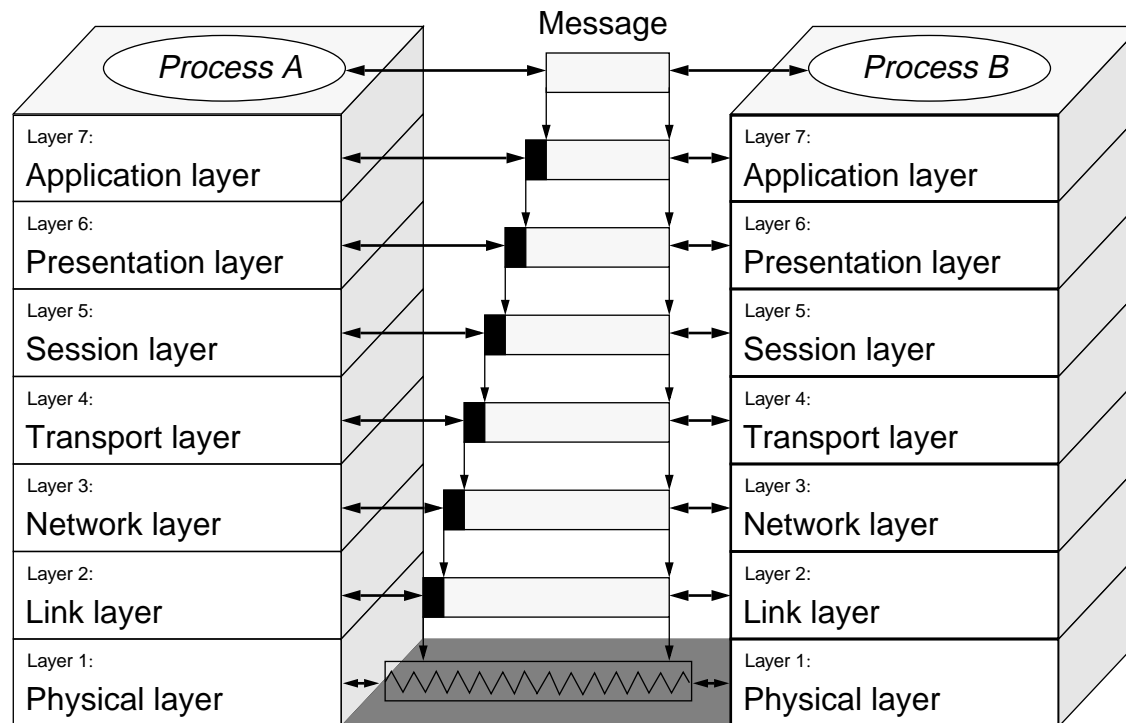


ISO/OSI 7 Layer Model

- Application layer
- Presentation layer
- Session layer
- Transport layer
- Network layer
- Data Link layer
- Physical layer



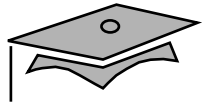
Data Exchange Between Application Processes





Physical Layer

- Regulates the transmission of data bits
- Is transmission medium dependent
- Uses Ethernet predominantly on UNIX[®] workstations



Data Link Layer

- Encapsulates user data into datagrams
- Supports error detection by using a checksum
- Supports following protocols:
 - Link Access Procedure (LAPB; X.25)
 - Ethernet V.2 and Ethernet IEEE 802.3
 - Token Bus IEEE 802.4 and Token Ring IEEE 802.5



Network Layer

- Performs routing
- Supports the following protocol:
 - CLNS/CONS (OSI)



Transport Layer

- Handles the transport of messages
- Supports following protocol:
 - TP-0 to TP-4 (OSI)



Session Layer

- Controls the exchange of messages
- Synchronizes packets
- Re-establishes interrupted connections



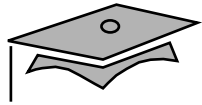
Presentation Layer

- Stipulates transfer syntax
- Represents data based on architecture
- Supports XDR



Application Layer

- Represents the application process
- Supports following common protocols:
 - SMTP (Simple Mail Transfer Protocol)
 - FTP (File Transfer Protocol)
 - TELNET (Remote Terminal Protocol)
 - NFSTM (Network File System)
 - SNMP (Simple Network Management Protocol)



TCP/IP

- Is a set of protocols
- Allows cooperating computers to share network resources
- Supports wide range of platforms and networks
- Provides important network services



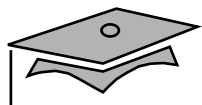
TCP/IP Network Model

- It is implemented as a layered protocol stack
- Each layer serves a specific purpose
- Each layer corresponds with equivalent layers on peer machines
- Each layer is independent of other layers

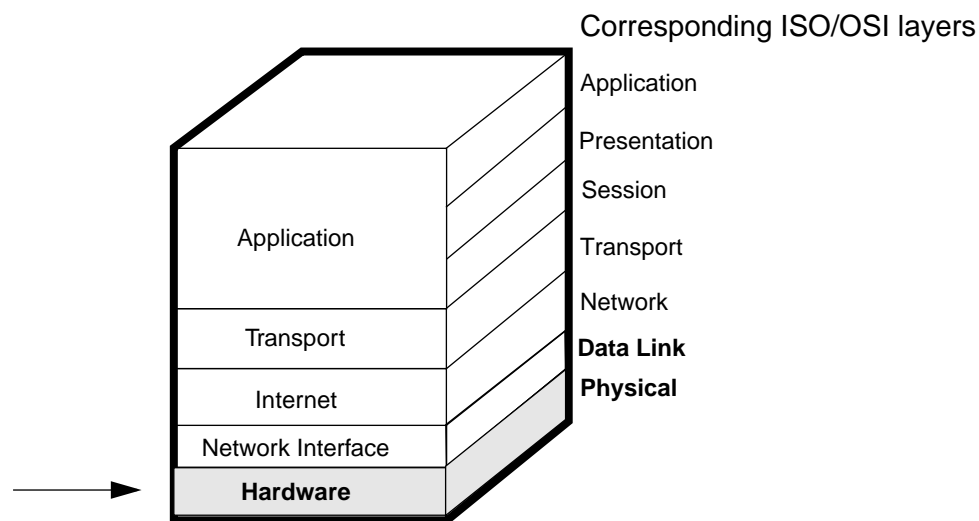


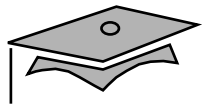
TCP/IP Layers

- Application layer
- Transport layer
- Internet layer
- Network Interface layer
- Hardware layer

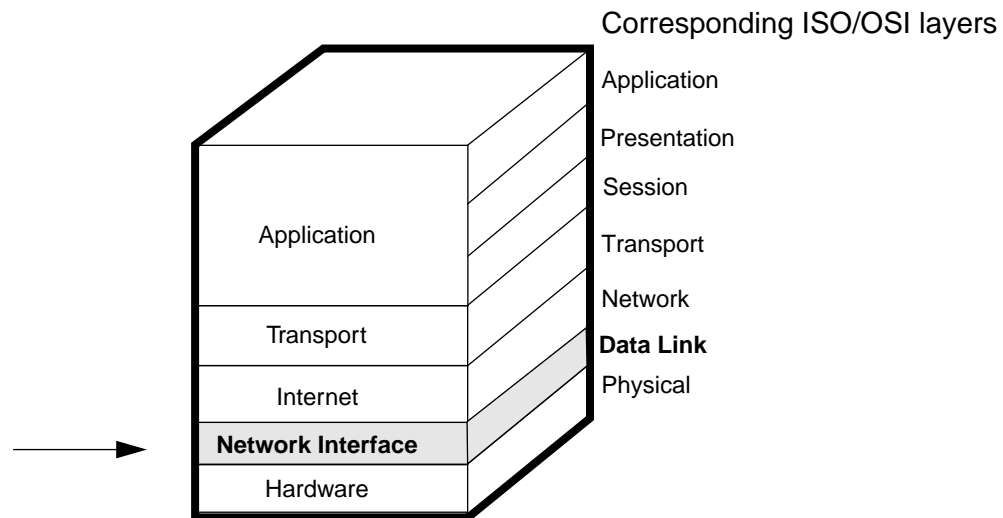


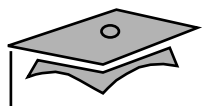
Hardware Layers



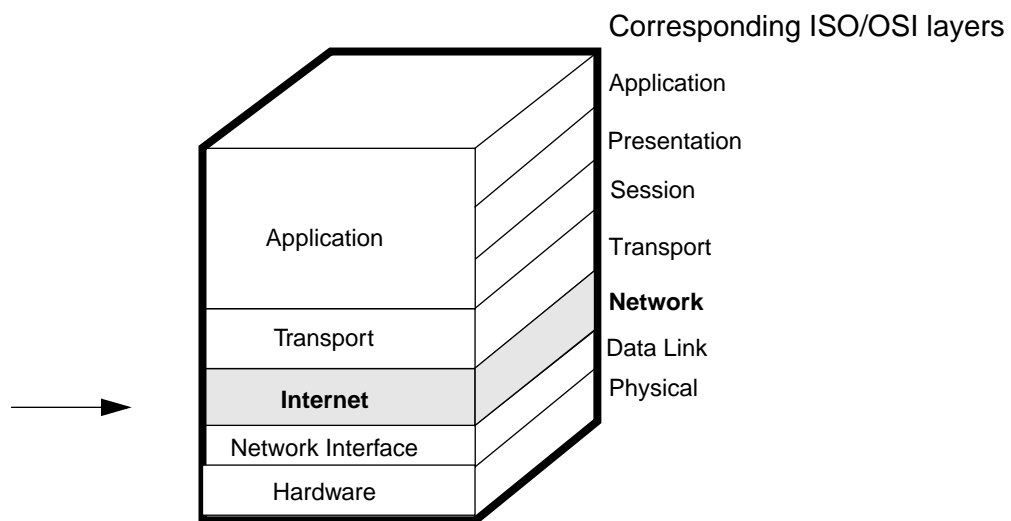


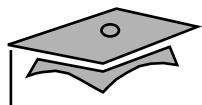
Network Interface Layer



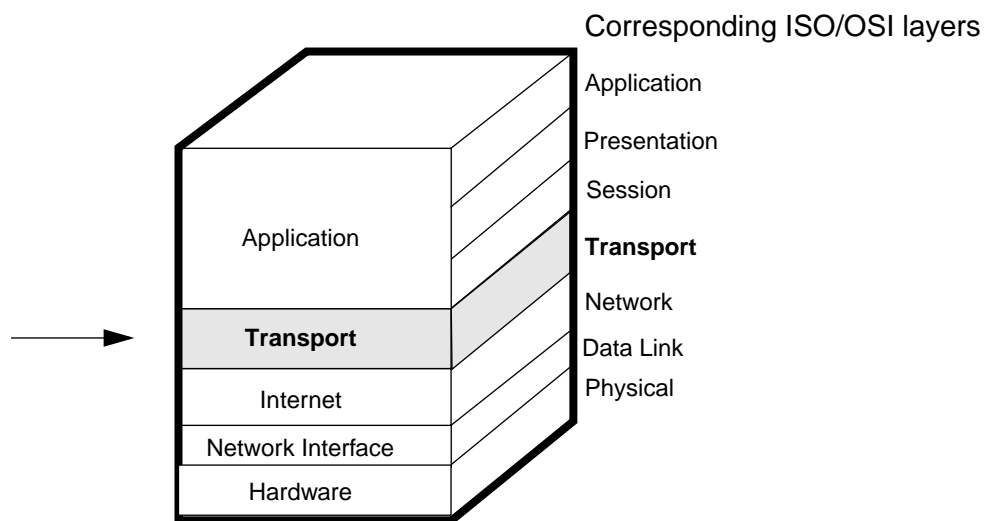


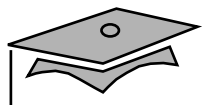
Internet Layer



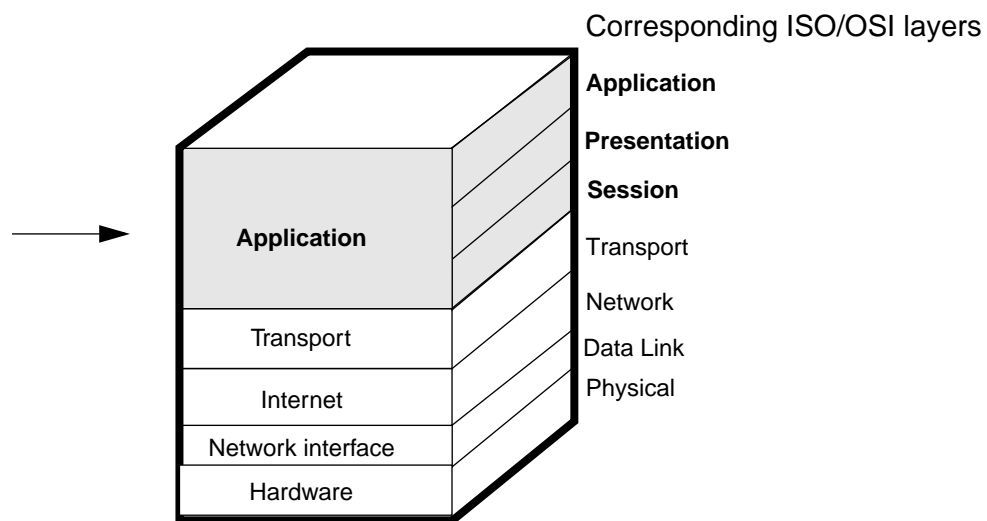


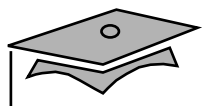
Transport Layer



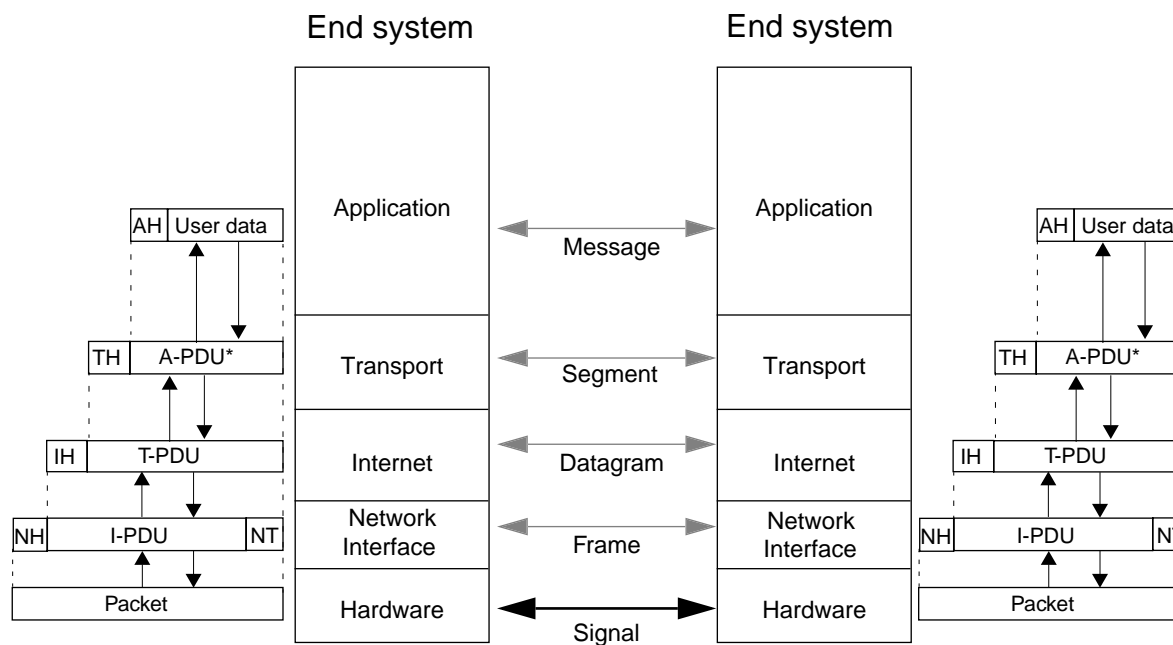


Application Layer





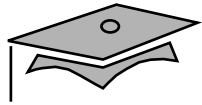
Peer-to-Peer Communication





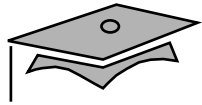
TCP/IP Protocol Stack

TCP/IP Protocol	TCP/IP Layer
NFS, NIS+, DNS, telnet, ftp, rlogin, SMTP, DHCP, SNMP, others	Application
TCP, UDP	Transport
IP, ARP, RARP, and ICMP	Internet
SLIP, PPP, IEEE 802.2	Network Interface
Ethernet (IEEE 802.3) Token Bus (IEEE 802.4), Token Rings (IEEE 802.5), RS-232, others	Hardware



Module 2

Introduction to Local Area Networks



Overview

- Objectives
- Relevance



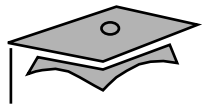
Introduction to Local Area Network

- Definition of local area network (LAN)
- Benefits of having a LAN
- LAN architecture
 - Hardware
 - Software

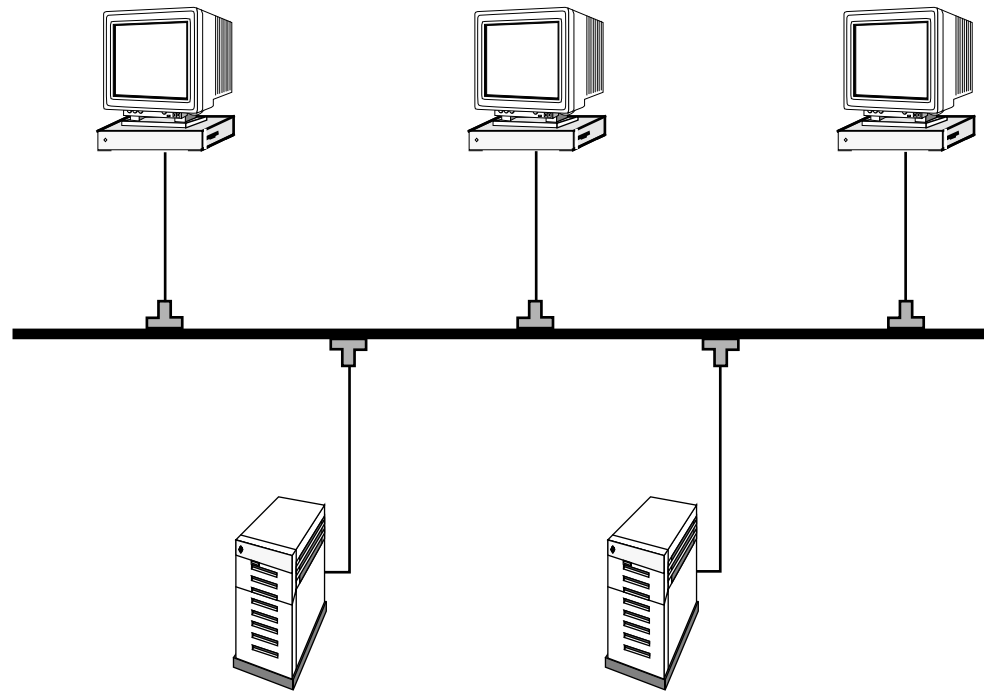


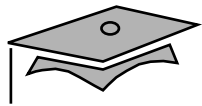
LAN Topology

- Bus
- Star
- Ring

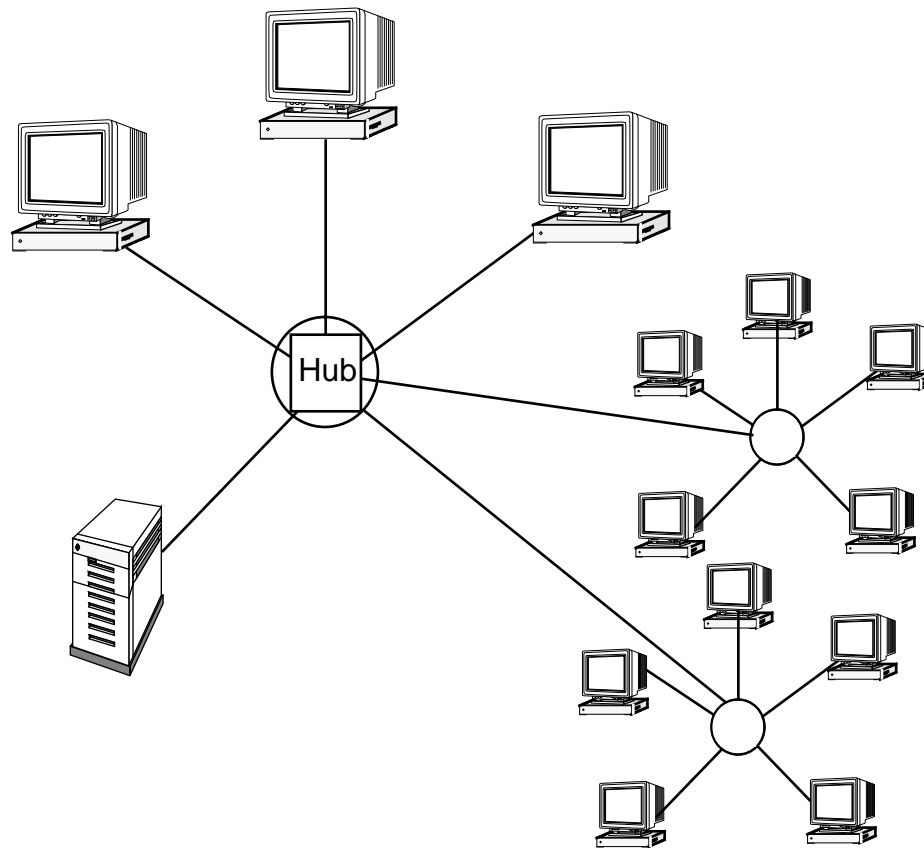


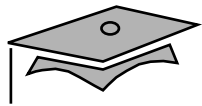
Bus Configuration



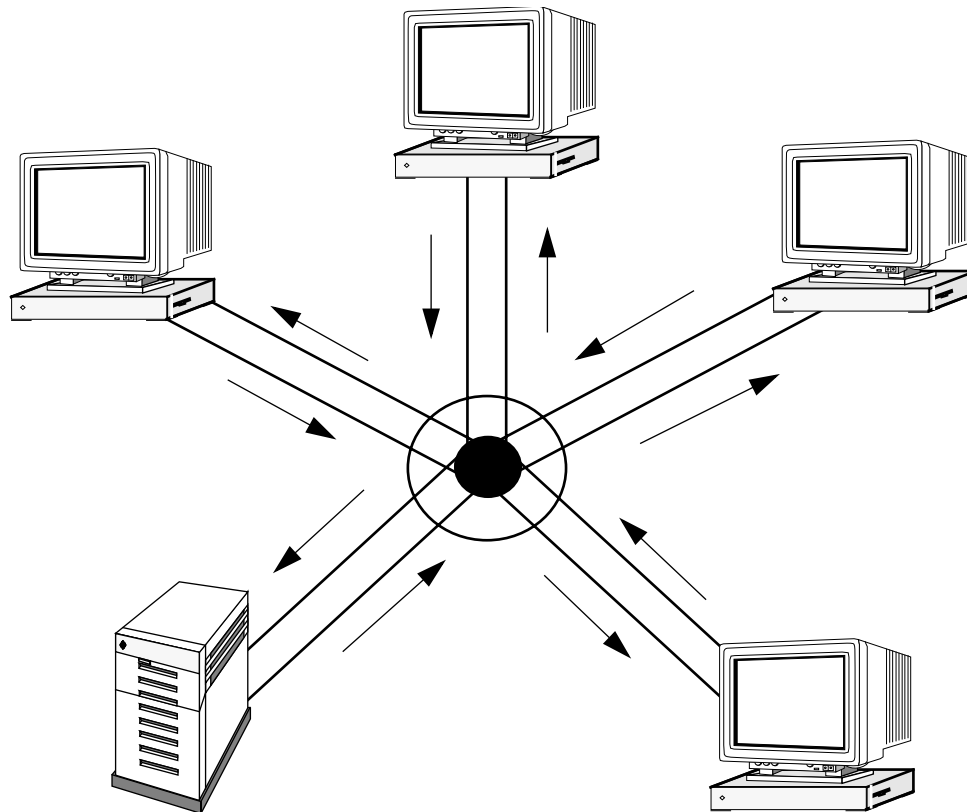


Star Configuration





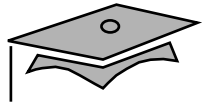
Ring Configuration





LAN Components

- Backbone
- Segment
- Repeater
- Hub
- Bridge
- Switch
- Router
- Gateway
- Concentrator



Ethernet Components

- Ethernet controller
- Transceiver
- Transceiver cable
- Thick Ethernet RG-8 coaxial cable, 50 Ohm
- Thin Ethernet RG-58 coaxial cable, 50 Ohm
- Terminator resisters, 50 Ohm
- Twisted pair cable (shielded and unshielded)



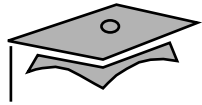
Sun Communications Controller

- ATM
- Ethernet
- Fast Ethernet
- FDDI
- Token Ring
- Gigabit Ethernet



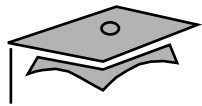
LAN Methodologies

- Ethernet – IEEE 802.3
- Asynchronous Transfer Mode
- Token Ring – IEEE 802.5
- Fiber Distributed Data Interface

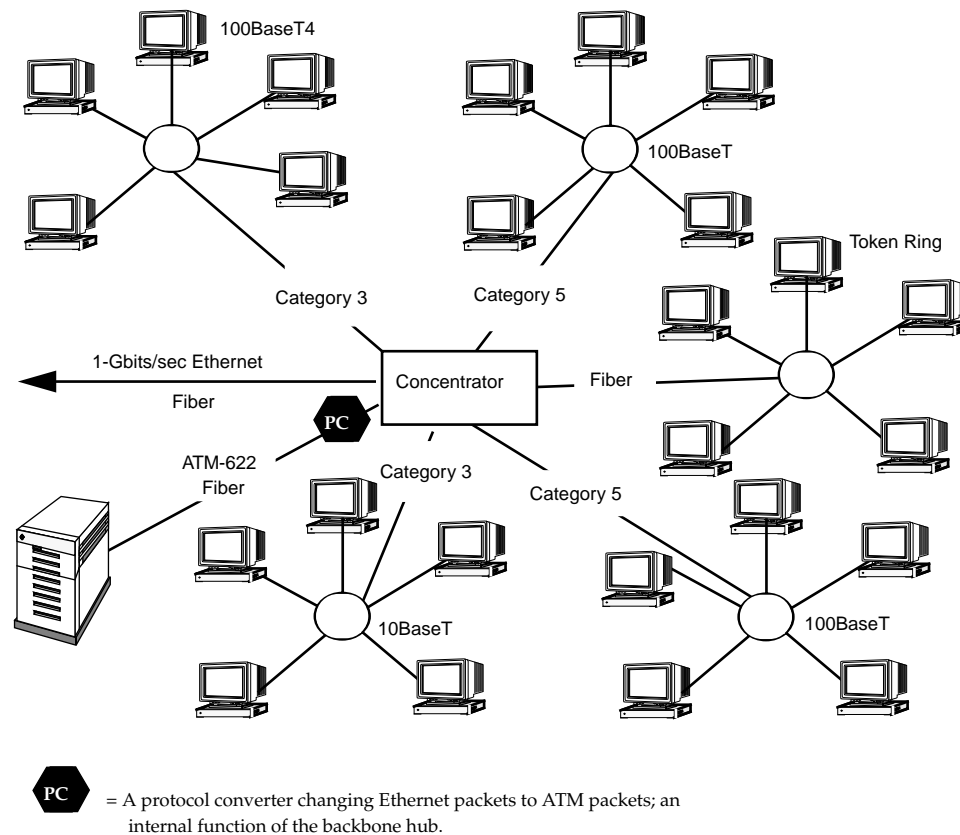


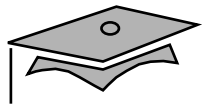
Network Media

- 10BASE-5
- 10BASE-2
- 10BASE-T
- 100BASE-TX
- 100BASE-T4
- 100BASE-FX

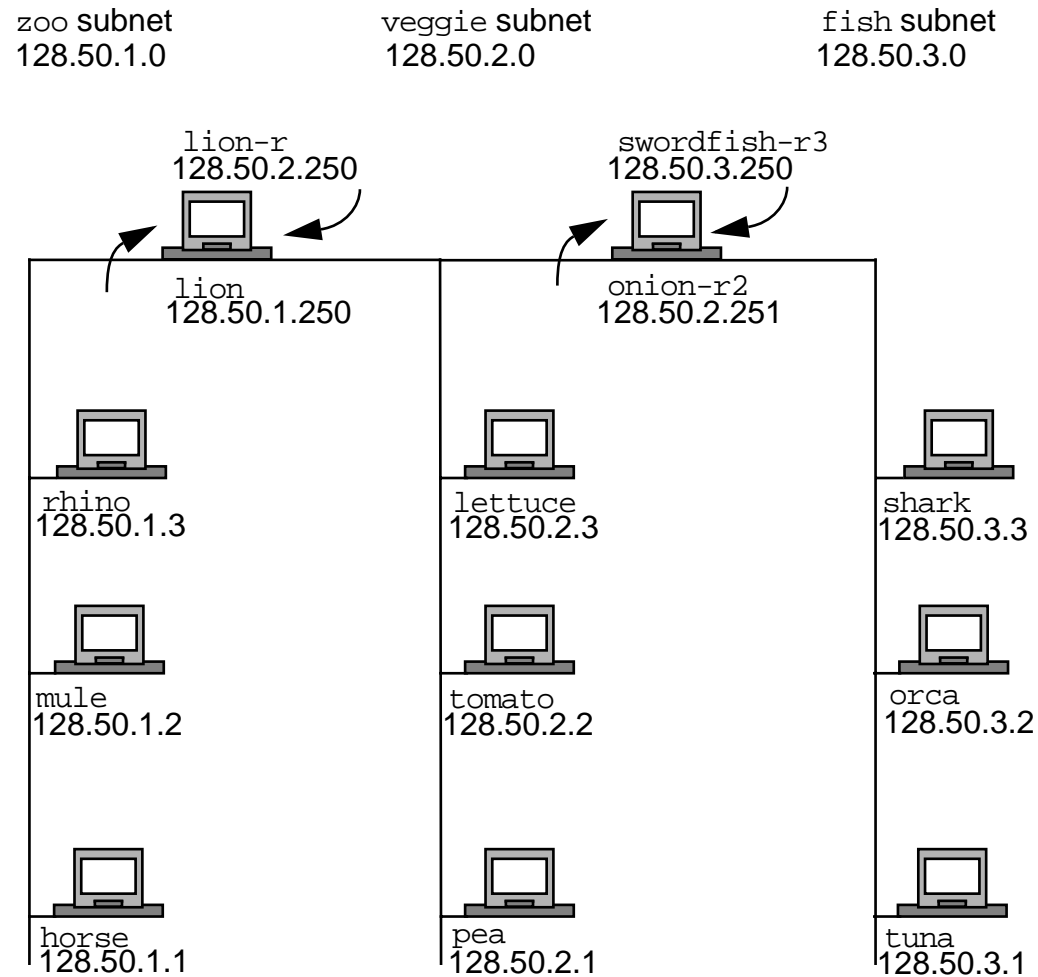


Multimode Ethernet





Lab Configuration





Module 3

Ethernet Interface



Overview

- Objectives
- Relevance

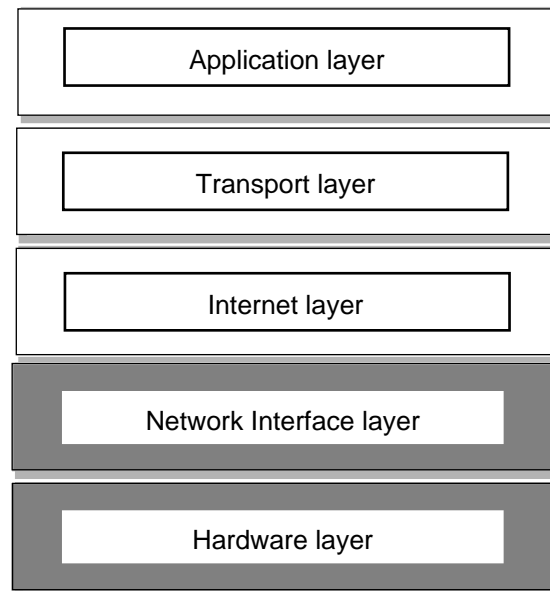


Introduction to Ethernet

- Is the most widely installed local area network technology
- Was developed by DEC, Intel, and Xerox
- Is specified in the IEEE 802.3 standard



Ethernet TCP/IP Layers





Ethernet Major Elements

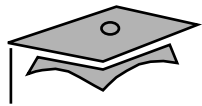
- Hardware network interface
- Network access method
 - Carrier Sense Multiple Access with Collision Detection (CSMA/CD)
 - Switched Ethernet
- Ethernet packet



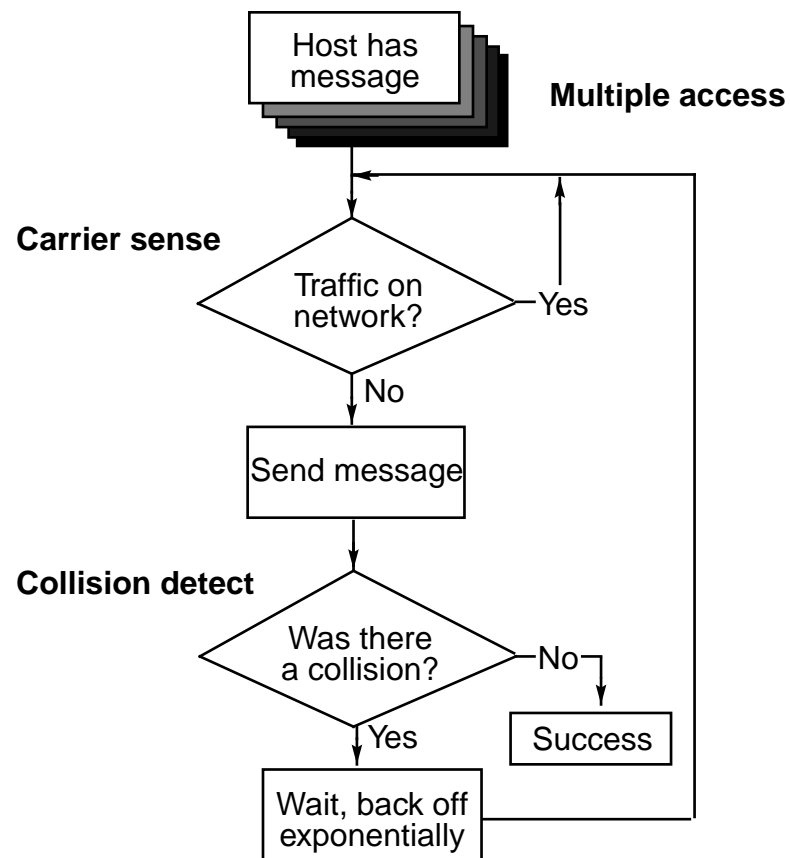
Access Method

CSMA/CD:

- Resolves conflicts due to multiple machines simultaneously accessing common medium
 - Listens for systems currently accessing medium
 - Waits for available medium
 - Senses collisions
 - Backs off and retries



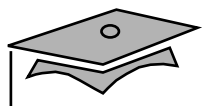
CSMA/CD Flowchart



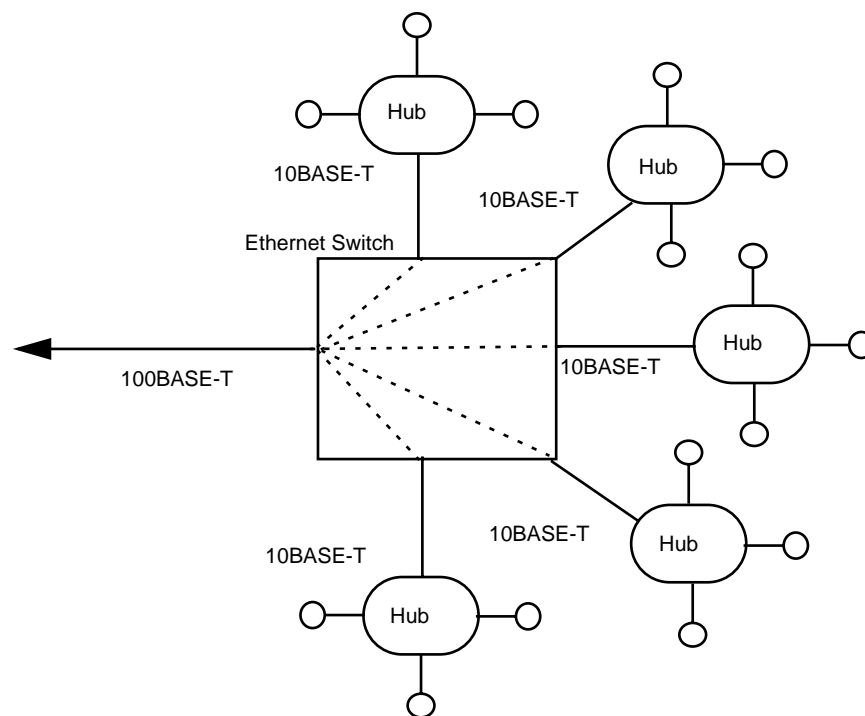


Switched Ethernet

- Reduces the number of collisions on a network
- Has central hub replace backbone medium
 - The hub consists of multiple ports.
 - There is one node (or hub) per port.
 - The hub switches between ports (nodes) as needed.
 - Common medium arbitration is eliminated.
 - Packet buffering and retransmission are supported.



Switched Ethernet Diagram



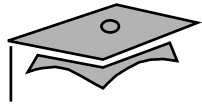


Ethernet Address

- Is host's unique network interface address
- Is administered by IEEE and assigned in manufacturing
- Is 48 bits long
- Displays as 12 hexadecimal digits using colon notation
- Has first three octets as vendor-specific identifier
- Has last three octets as network interface-specific identifier

Example:

08:00:20:1e:56:7d



Sending Messages

- Three types of Ethernet addresses
 - Unicast address
 - Broadcast address
 - Multicast address

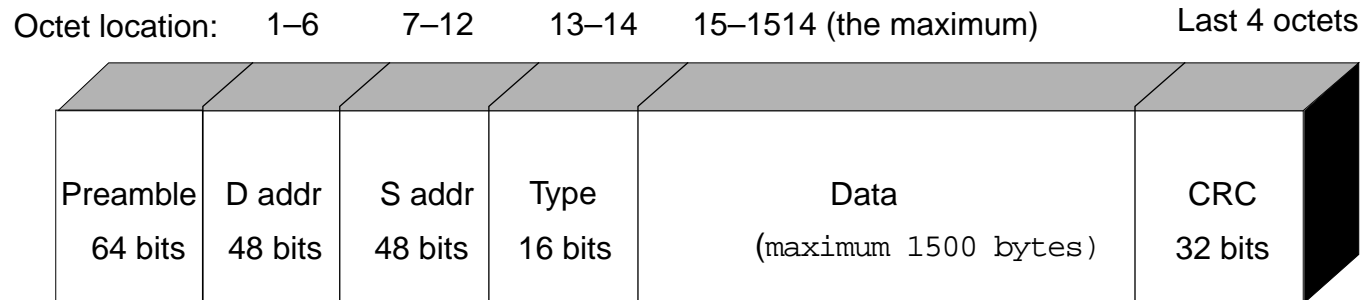


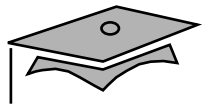
Ethernet Frame

- Preamble
- Destination address
- Source address
- Type
- Data
- Cyclical redundancy check (CRC)

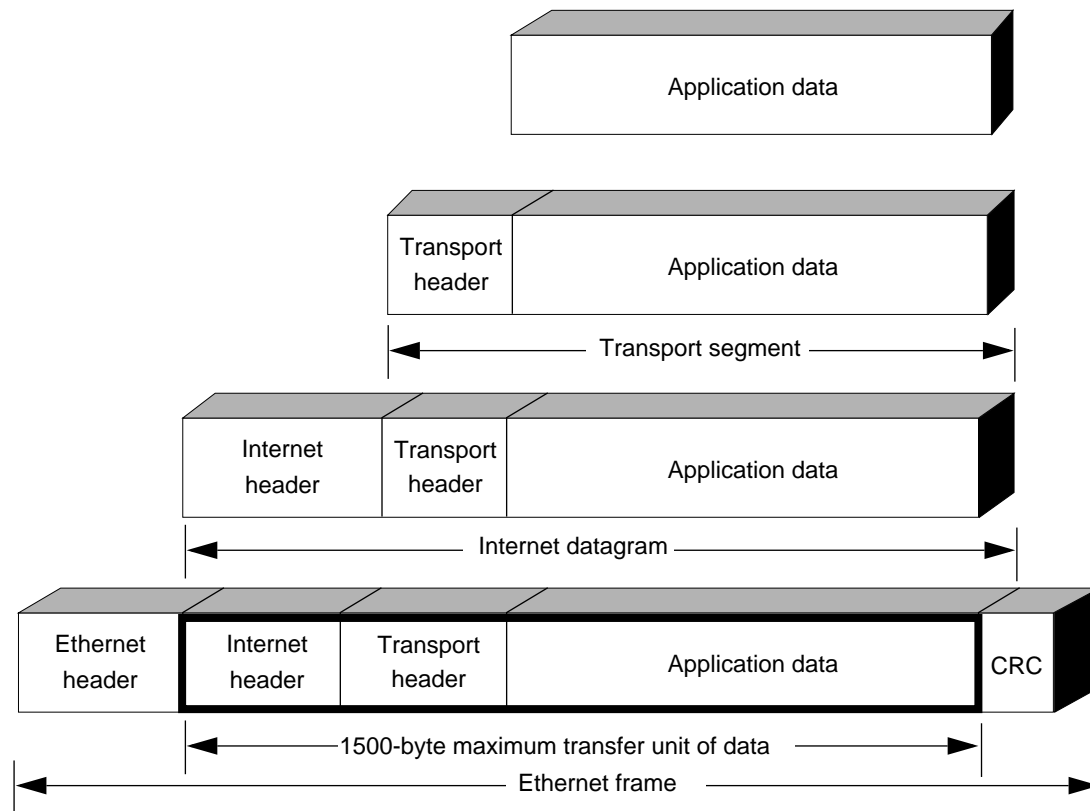


Ethernet Version 2 Frame Fields



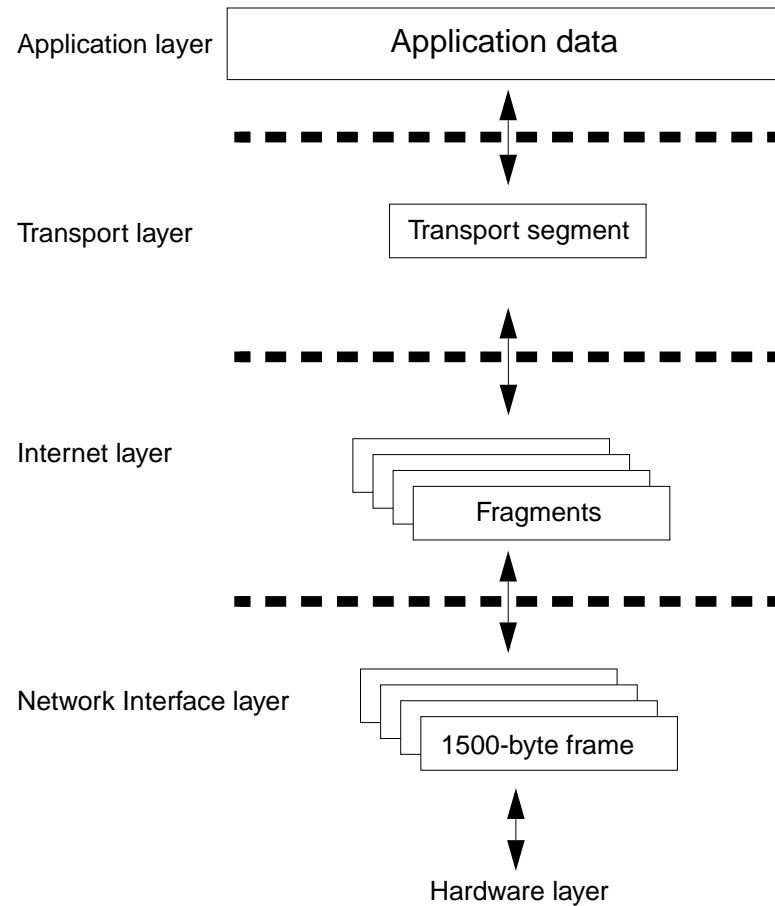


TCP/IP Layer Encapsulation





Ethernet Maximum Transfer Unit





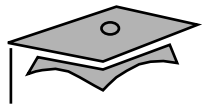
Ethernet Error Checking

- Runts
- Jabbers
- Bad CRC



Useful Troubleshooting Commands

- `snoop`
- `netstat`
- `ifconfig`



snoop

```
# snoop broadcast
```

```
Using device /dev/le (promiscuous mode)
skunk --> 128.50.255.255 RUSERS C
zebra --> 128.50.255.255 RUSERS C
mil02lab -> (broadcast) RIP R (25 destinations)
mil02lab -> (broadcast) RIP R (25 destinations)
mil02lab -> (broadcast) RIP R (25 destinations)
```



snoop -v

```
# snoop -v broadcast
```

```
Using device /dev/hme (promiscuous mode)
```

```
ETHER: ----- Ether Header -----
```

```
ETHER:
```

```
ETHER: Packet 1 arrived at 15:28:16.62
```

```
ETHER: Packet size = 60 bytes
```

```
ETHER: Destination = ff:ff:ff:ff:ff:ff, (broadcast)
```

```
ETHER: Source      = 8:0:20:e:d:56, Sun
```

```
ETHER: Ethertype = 0806 (ARP)
```

```
ETHER:
```

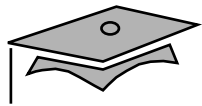
```
ARP: ----- ARP/RARP Frame -----
```

```
ARP:
```

```
ARP: Hardware type = 1
```

```
ARP: Protocol type = 0800 (IP)
```

```
.
```



snoop -V

```
# snoop -V cherries
```

```
Using device /dev/hme (promiscuous mode)
```

```
wrapper -> cherries ETHER Type=0800 (IP), size = 98 bytes
```

```
wrapper -> cherries IP D=129.150.165.123 S=129.150.165.114 LEN=84, ID=7780
```

```
wrapper -> cherries ICMP Echo request
```

```
cherries -> wrapper ETHER Type=0800 (IP), size = 98 bytes
```

```
cherries -> wrapper IP D=129.150.165.114 S=129.150.165.123 LEN=84, ID=5905
```

```
cherries -> wrapper ICMP Echo reply
```



netstat -i

```
# netstat -i
```

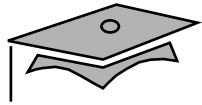
Name	Mtu	Net/Dest	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll	Queue
lo0	8232	loopback	localhost	5248	0	5248	0	0	0
hme0	1500	128.50.0.0	mule	77553	4	39221	2	2103	0



ifconfig

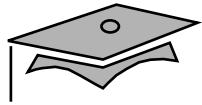
```
# ifconfig hme
```

```
hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST>  
mtu 1500 inet 129.150.65.124 netmask ffffffff  
broadcast 129.150.65.255 ether 8:0:20:80:d0:a7
```



Module 4

ARP and RARP



Overview

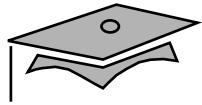
- Objectives
- Relevance



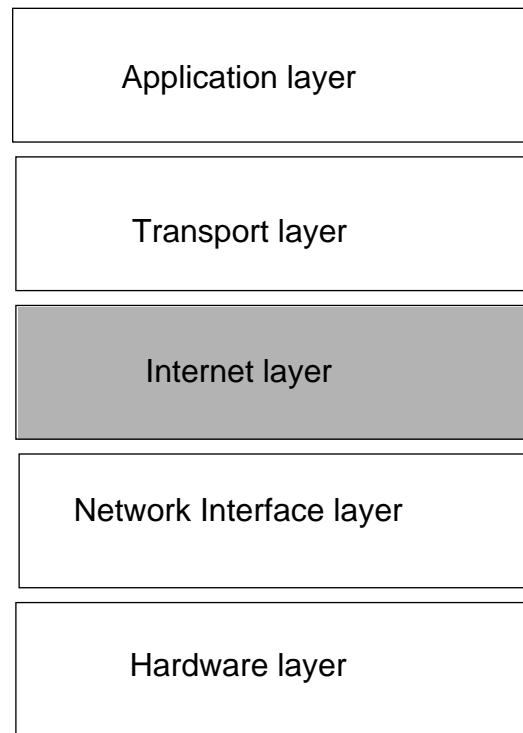
Introduction to Address Resolution

The two resolutions performed by the ARP and RARP protocols are:

- Address resolution – Process of mapping a 32-bit IP address to a 48-bit Ethernet address
- Reverse address resolution – Process of mapping a 48-bit Ethernet address to a 32-bit IP address



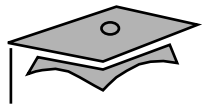
Address Resolution TCP/IP Layers



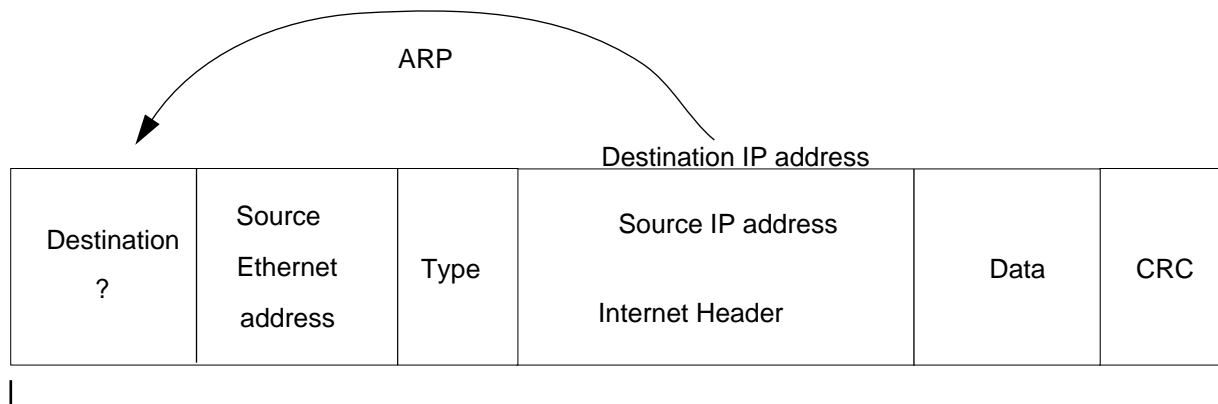


Why ARP Is Required

- Data is encapsulated into an Ethernet frame which contains all the necessary information except for the destination Ethernet address
- Destination Ethernet address is obtained using the ARP protocol



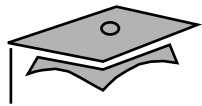
Ethernet Frame Address Resolution





Address Resolution Protocol

- ARP is the process that builds an address link between the Internet layer and Network Interface layer.
- Key ARP elements are:
 - ARP table
 - ARP request
 - ARP reply
 - ARP reply caching



ARP Request

```
# snoop -v arp
```

```
Using device /dev/le (promiscuous mode)
```

```
ETHER: ----- Ether Header -----
```

```
ETHER:
```

```
ETHER: Packet 1 arrived at 16:15:29.64
```

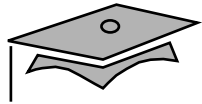
```
ETHER: Packet size = 42 bytes
```

```
ETHER: Destination = ff:ff:ff:ff:ff:ff, (broadcast)
```

```
ETHER: Source = 8:0:20:75:6e:6f, Sun
```

```
ETHER: Ethertype = 0806 (ARP)
```

```
ETHER:
```



ARP Request

ARP: ----- ARP/RARP Frame -----

ARP: Hardware type = 1

ARP: Protocol type = 0800 (IP)

ARP: Length of hardware address = 6 bytes

ARP: Length of protocol address = 4 bytes

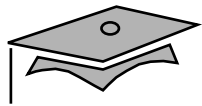
ARP: Opcode 1 (ARP Request)

ARP: Sender's hardware address = 8:0:20:75:6e:6f

ARP: Sender's protocol address = 128.50.1.2, mule

ARP: Target hardware address = ?

ARP: Target protocol address = 128.50.1.3, rhino



ARP Reply

```
# snoop -v arp
```

```
ETHER: ----- Ether Header -----
```

```
ETHER:
```

```
ETHER: Packet 2 arrived at 16:15:29.64
```

```
ETHER: Packet size = 60 bytes
```

```
ETHER: Destination = 8:0:20:75:6e:6f, Sun
```

```
ETHER: Source = 8:0:20:75:8b:59, Sun
```

```
ETHER: Ethertype = 0806 (ARP)
```

```
ETHER:
```



ARP Reply

ARP: Hardware type = 1

ARP: Protocol type = 0800 (IP)

ARP: Length of hardware address = 6 bytes

ARP: Length of protocol address = 4 bytes

ARP: Opcode 2 (ARP Reply)

ARP: Sender's hardware address = 8:0:20:75:8b:59

ARP: Sender's protocol address = 128.50.1.3, rhino

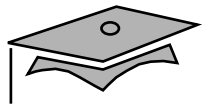
ARP: Target hardware address = 8:0:20:75:6e:6f

ARP: Target protocol address = 128.50.1.2, mule



ARP Table Management

- `arp -a`
- `arp -s hostname ethernet_address`
- `arp -d hostname`
- `arp -f filename`



ARP Command Examples

```
# arp -a
```

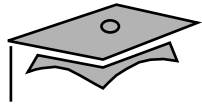
```
Net to Media Table
```

Device	IP Address	Mask	Flags	Phys Addr
hme0	rhino	255.255.255.255		08:00:20:75:8b:59
hme0	mule	255.255.255.255	SP	08:00:20:75:6e:6f
hme0	horse	255.255.255.255	U	
hme0	224.0.0.0	240.0.0.0	SM	01:00:5e:00:00:00



Reverse Address Resolution

- Process that builds an address link between the Network Interface layer and Internet layer
- RARP protocol begins with a known Ethernet address to obtain an unknown IP address
- Common uses include:
 - Diskless systems
 - JumpStart™ systems



RARP Request

```
# snoop -v rarp
```

```
Using device /dev/le (promiscuous mode)
```

```
ETHER: ----- Ether Header -----
```

```
ETHER:
```

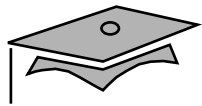
```
ETHER: Packet 1 arrived at 16:29:55.70
```

```
ETHER: Packet size = 64 bytes
```

```
ETHER: Destination = ff:ff:ff:ff:ff:ff, (broadcast)
```

```
ETHER: Source = 8:0:20:75:8b:59, Sun
```

```
ETHER: Ethertype = 8035 (RARP)
```



RARP Request

ARP: ----- ARP/RARP Frame -----

ARP: Hardware type = 1

ARP: Protocol type = 0800 (IP)

ARP: Length of hardware address = 6 bytes

ARP: Length of protocol address = 4 bytes

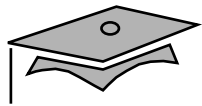
ARP: Opcode 3 (REVARP Request)

ARP: Sender's hardware address = 8:0:20:75:8b:59

ARP: Sender's protocol address = 255.255.255.255, BROADCAST

ARP: Target hardware address = 8:0:20:75:8b:59

ARP: Target protocol address = ?



RARP Reply

```
# snoop -v rarp
```

```
ETHER: ----- Ether Header -----
```

```
ETHER:
```

```
ETHER: Packet 2 arrived at 16:29:58.78
```

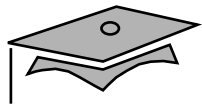
```
ETHER: Packet size = 42 bytes
```

```
ETHER: Destination = 8:0:20:75:8b:59, Sun
```

```
ETHER: Source = 8:0:20:75:6e:6f, Sun
```

```
ETHER: Ethertype = 8035 (RARP)
```

```
ETHER:
```



RARP Reply

ARP: ----- ARP/RARP Frame -----

ARP: Hardware type = 1

ARP: Protocol type = 0800 (IP)

ARP: Length of hardware address = 6 bytes

ARP: Length of protocol address = 4 bytes

ARP: Opcode 4 (REVARP Reply)

ARP: Sender's hardware address = 8:0:20:75:6e:6f

ARP: Sender's protocol address = 128.50.1.2, mule

ARP: Target hardware address = 8:0:20:75:8b:59

ARP: Target protocol address = 128.50.1.3, rhino



Troubleshooting the `in.rarpd` Server

- Run the `snoop -v rarp` command on a third disinterested diskless client
 - No diskless client RARP request – network hardware problem
- If server fails to reply to RARP request, check:
 - `/etc/inet/hosts` file
 - `/etc/ethers` file
 - `in.rarpd` process is running



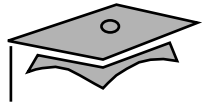
Module 5

Internet Layer



Overview

- Objectives
- Relevance

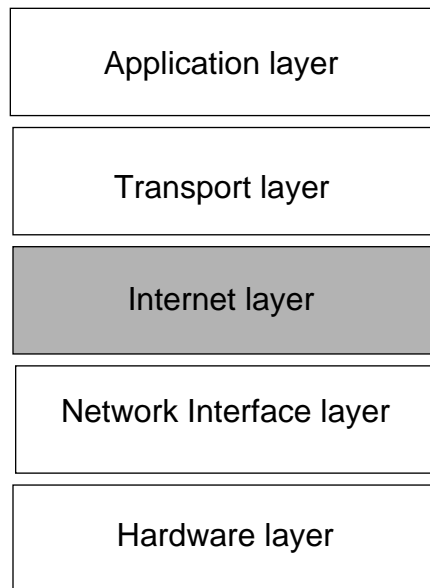


Introduction to Internet

- The early days
- Berkeley Software Distribution
- Rapid growth
- The future



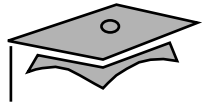
TCP/IP Layered Model





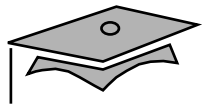
Internet Layer

- Internet Protocol
- Datagrams
- Internet Control Message Protocol
- Fragmentation

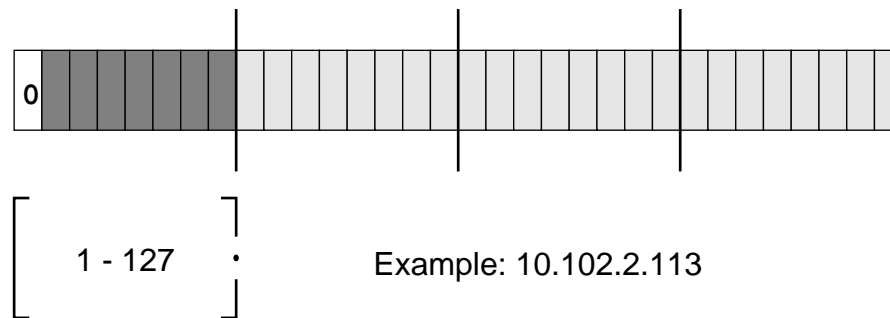


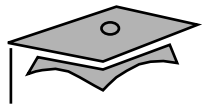
Classful IPv4 Addressing

- Class A – Very large networks (up to 16 million hosts)
- Class B – Large networks (up to 65,000 hosts)
- Class C – Small and mid-sized networks (up to 254 hosts)
- Class D – Multicast address

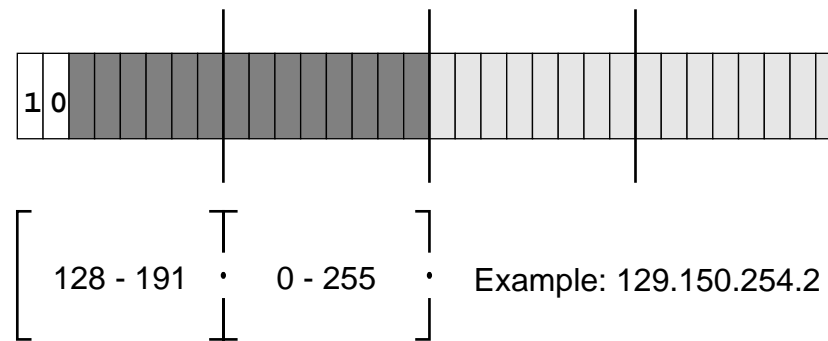


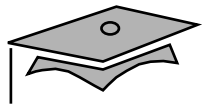
Class A Address Format



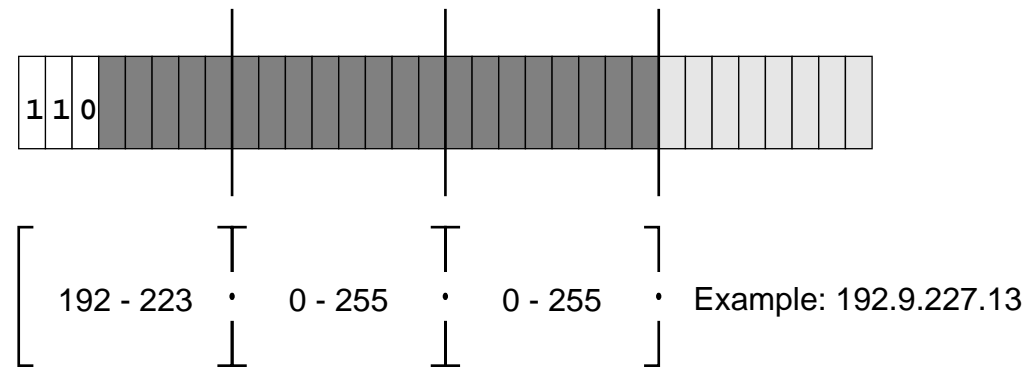


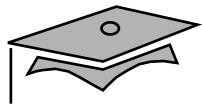
Class B Address Format



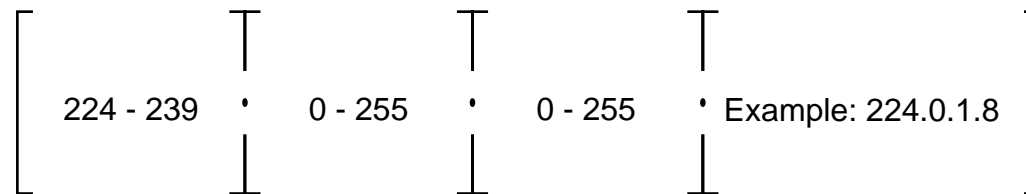
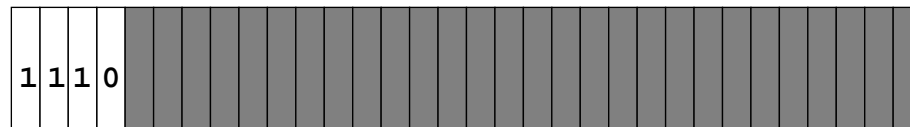


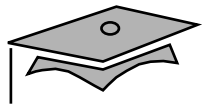
Class C Address Format





Class D Address Format





Special IPv4 Addresses

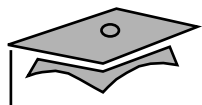
- IPv4 broadcast addresses
- Reserved network and host IPv4 values

IPv4 Address	Description
127.x.x.x	Reserved for loopback.
Network number followed by all bits set to 0	Network address, such as 128.50.0.0.
Network number followed by all bits set to 1	Broadcast address, such as 128.50.255.255.
0.0.0.0	Special address used by systems that do not yet know its own IP address. Protocols such as RARP and BOOTP use this address when attempting to communicate with a server.
255.255.255.255	Generic broadcast.



IPv4 Netmasks

- Explicitly identifies network number
- Supports IPv4 default netmasks
 - Class A – 255.0.0.0
 - Class B – 255.255.0.0
 - Class C – 255.255.255.0



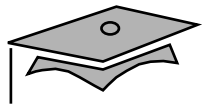
Computing Network Number

IPv4 address in decimal:	171.63.14.3
IPv4 address in binary:	10101011 00111111 00001110 00000011
Class B netmask in decimal:	255.255.0.0
Class B netmask in binary:	11111111 11111111 00000000 00000000
Apply the logical AND operator:	
IPv4 address (decimal):	171 63 14 3
IPv4 address (binary):	10101011 00111111 00001110 00000011
AND netmask:	11111111 11111111 00000000 00000000
Network # (binary):	<u>10101011 00111111 00000000 00000000</u>
Network # (decimal):	171 63 0 0



Reasons to Subnet

- Isolation of traffic
- Security
- Localization of protocols
- Geographical or departmental association
- Administration



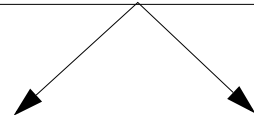
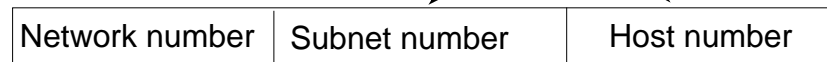
Defining Subnets

- Address hierarchy

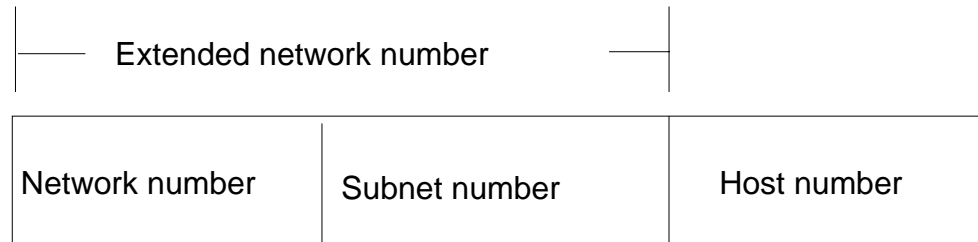
Two-level hierarchy



Three-level hierarchy



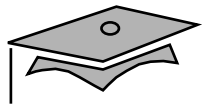
- Extended network number



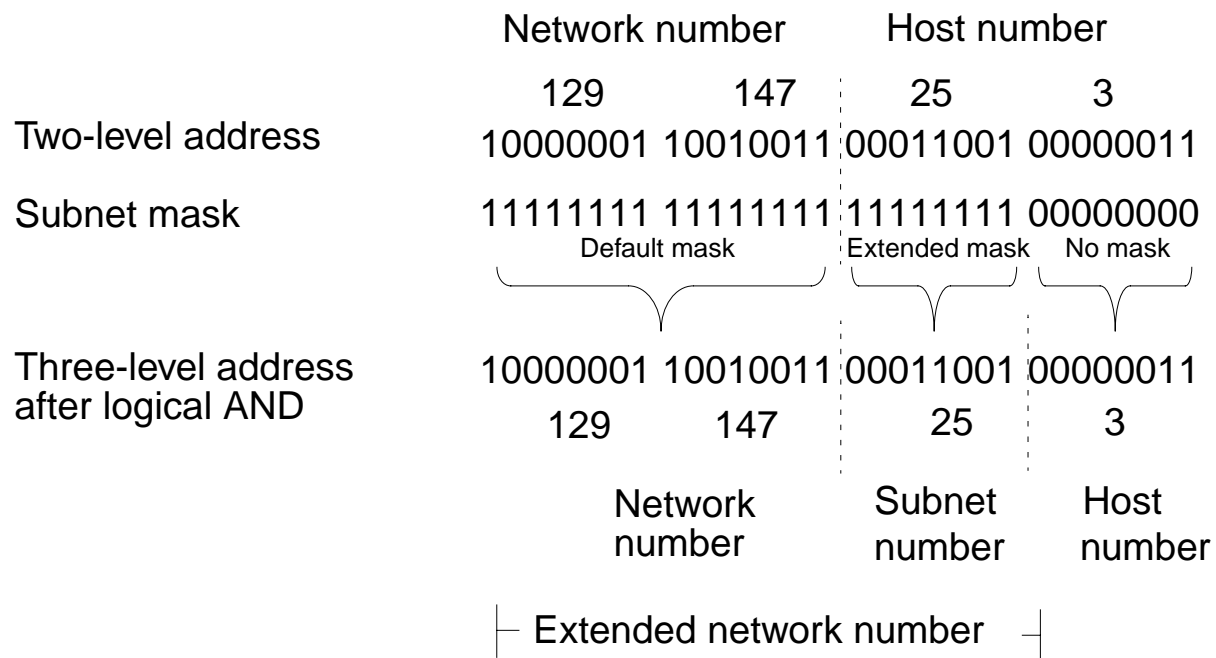


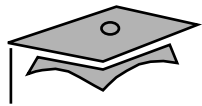
Subnet Mask

- Defines the extended-network-number
- Extends default netmask into the host-number field
- Supports logical AND operations



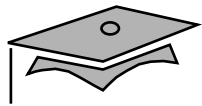
Computation of Extended Network Number



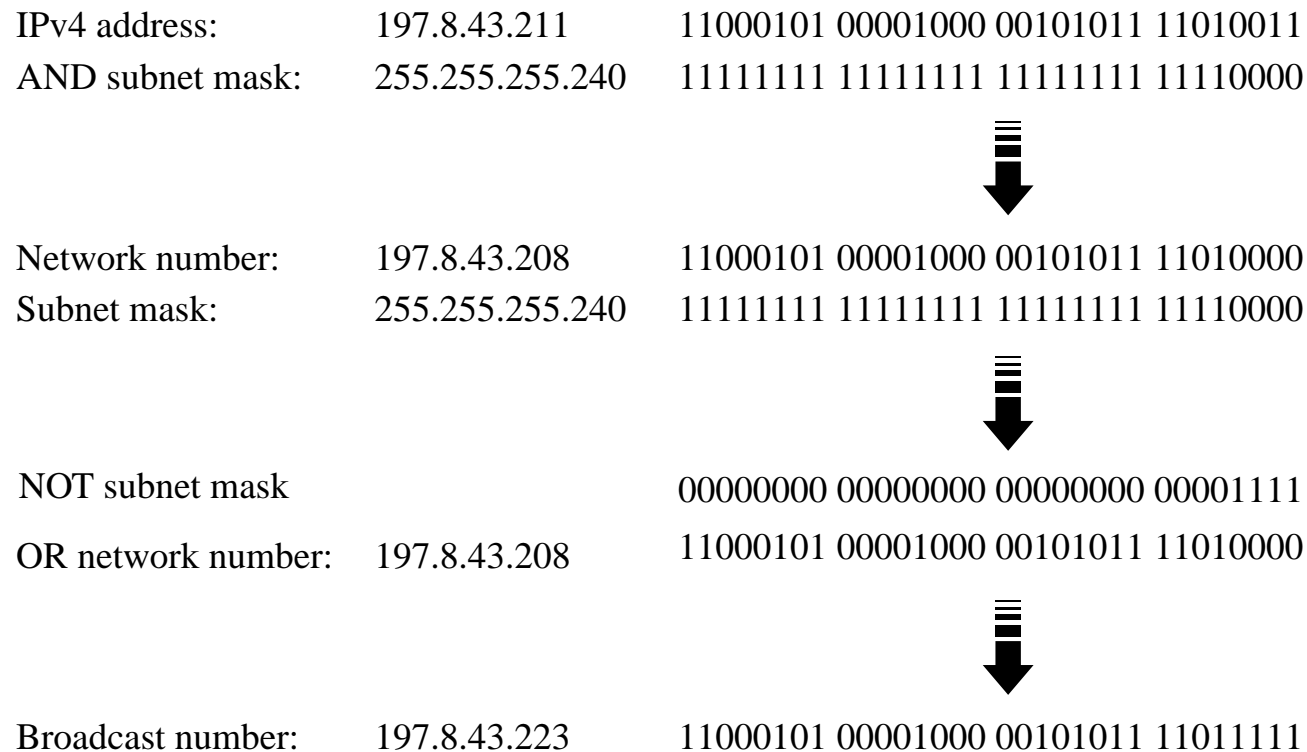


Non-Byte Bounded Subnet Masks

		Network number			Host number		
	197	8	43	211			
IPv4 class C address	11000101	00001000	00101011	11010011			
Subnet mask	11111111	11111111	11111111	11110000			
		}			}		
Extended-network-number	11000101	00001000	00101011	11010000			
		Network number		Subnet number			
Dot-notation (decimal)	197	.	8	.	43	.	208



Computing the Broadcast Address





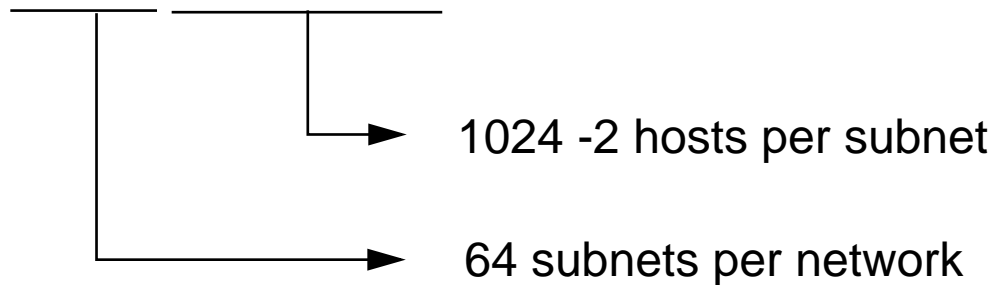
Variable Length Subnet Masks (VLSM)

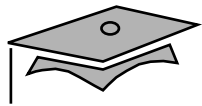
- Advantages
- Efficient use of IP address space
- Route aggregation
- Associated protocols



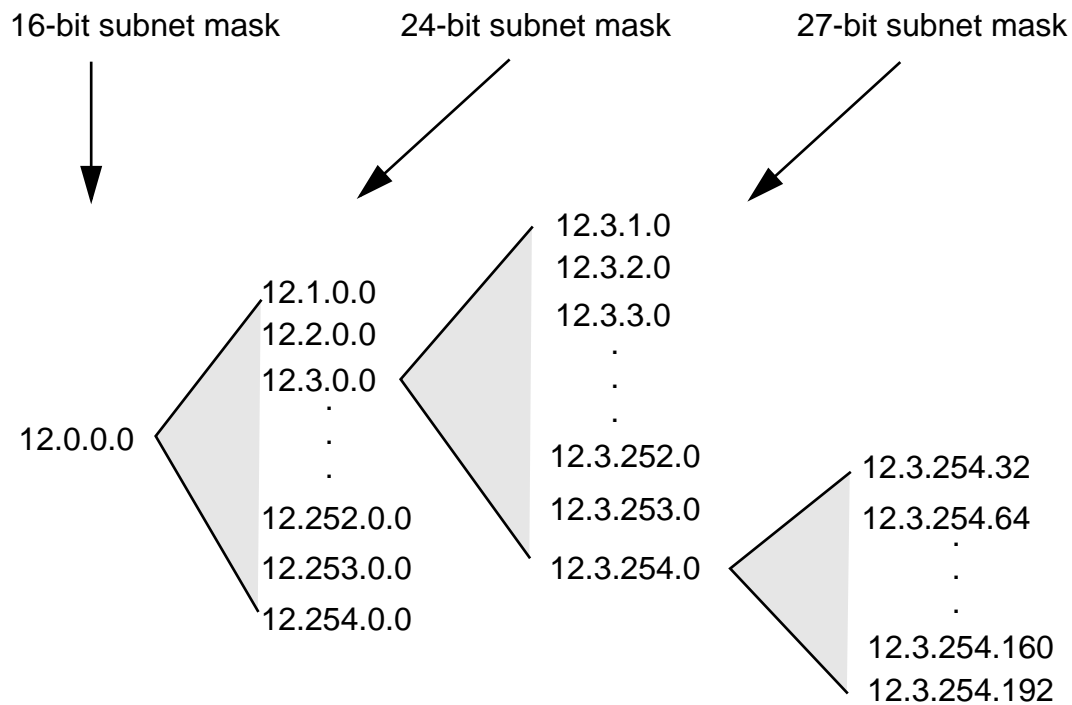
Class B Subnet Mask Yield

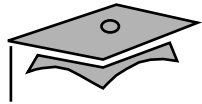
11111111 11111111 11111100 00000000





Class A Network Using VLSM





Permanent Subnet Masks

- `/etc/inet/netmasks` file
- Example of a class B network:

`128.50.0.0 255.255.255.0`

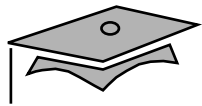
- Example of a class C network:

`197.8.43.0 255.255.255.240`



Recommended Subnet Masks

- Contiguous – Recommended
- Non-contiguous – Not recommended



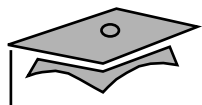
Class B Subnet Masks

Mask in Decimal	Mask in Binary	Number of Subnets	Number of Hosts per Subnets
255.255.0.0	11111111 11111111 00000000 00000000	1	65534
255.255	11111111 11111111 10000000 00000000	2	32766
255.255	11111111 11111111 11000000 00000000	4	16382
255.255	11111111 11111111 11100000 00000000	8	8190
255.255	11111111 11111111 11110000 00000000	16	4094
255.255	11111111 11111111 11111000 00000000	32	2046
255.255	11111111 11111111 11111100 00000000	64	1022
255.255	11111111 11111111 11111110 00000000	128	510
255.255.255.0	11111111 11111111 11111111 00000000	256	254
255.255.255.128	11111111 11111111 11111111 10000000	512	126
255.255.255.192	11111111 11111111 11111111 11000000	1024	62
255.255.255.224	11111111 11111111 11111111 11100000	2048	30
255.255.255.240	11111111 11111111 11111111 11110000	4096	14
255.255.255.248	11111111 11111111 11111111 11111000	8192	6
255.255.255.252	11111111 11111111 11111111 11111100	16384	2



Class C Subnet Masks

Mask in Decimal	Mask in Binary	Number of Subnets	Number of Hosts per Subnets
255.255.255.0	11111111 11111111 11111111 00000000	1	254
255.255.255.128	11111111 11111111 11111111 10000000	2	126
255.255.255.192	11111111 11111111 11111111 11000000	4	62
255.255.255.224	11111111 11111111 11111111 11100000	8	30
255.255.255.240	11111111 11111111 11111111 11110000	16	14
255.255.255.248	11111111 11111111 11111111 11111000	32	6
255.255.255.252	11111111 11111111 11111111 11111100	64	2

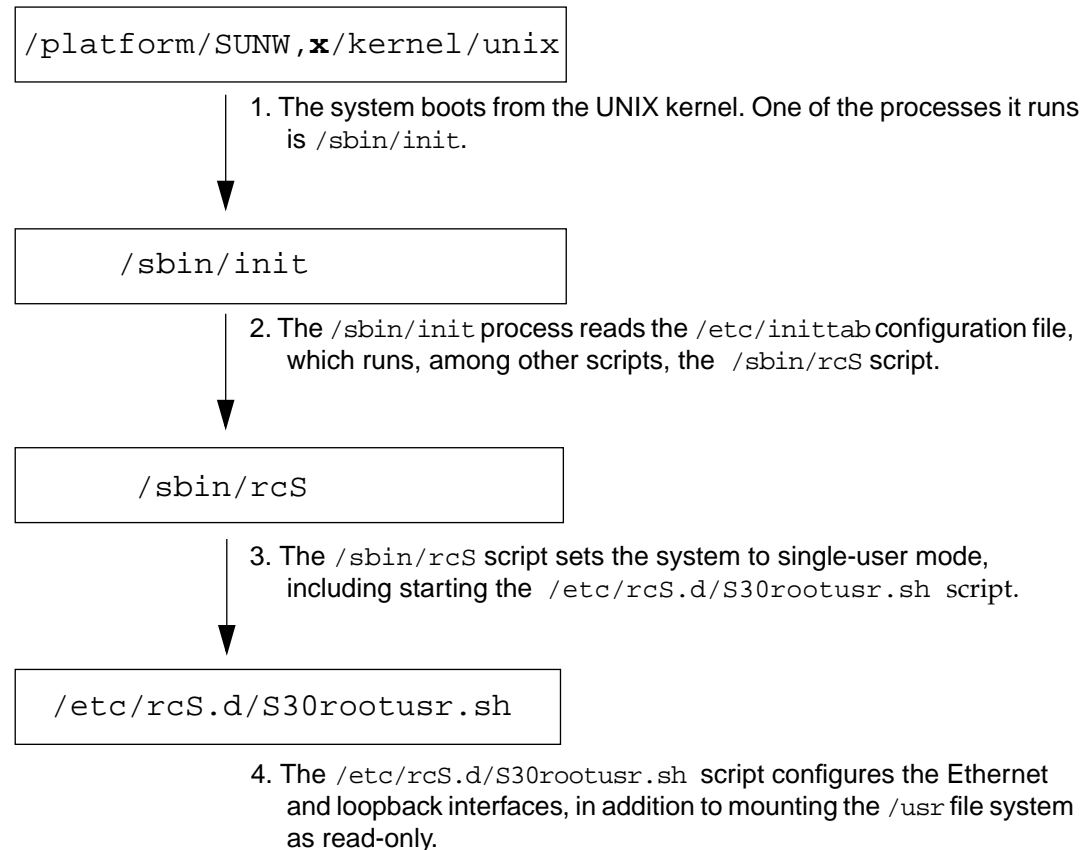


Configuring a Subnet

- Router setup
- Host setup
 - Subnet setup using NIS
 - Subnet setup using NIS+
 - Subnet setup without a name service



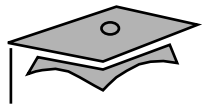
Network Interface Configuration





`/sbin/ifconfig` Command

- Configures network interfaces
- Is invoked by `/etc/rcS.d/S30rootusr` at startup



Examining Network Interfaces

```
# ifconfig -a
```

```
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
```

```
inet 127.0.0.1 netmask ff000000
```

```
hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
```

```
inet 128.50.1.2 netmask ffff0000 broadcast 128.50.255.255
```

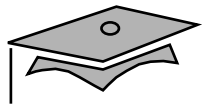
```
ether 8:0:20:75:6e:6f
```

```
# ifconfig hme0
```

```
hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
```

```
inet 128.50.1.2 netmask ffff0000 broadcast 128.50.255.255
```

```
ether 8:0:20:75:6e:6f
```



Enable and Disable Interface Examples

```
# ifconfig hme0 up
# ifconfig hme0
hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 128.50.1.2 netmask ffff0000 broadcast 128.50.255.255
    ether 8:0:20:75:6e:6f
```

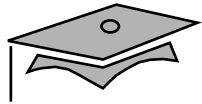
```
# ifconfig hme0 down
# ifconfig hme0
hme0: flags=862<BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 128.50.1.2 netmask ffff0000 broadcast 128.50.255.255
    ether 8:0:20:75:6e:6f
```



Close and Open Interface Examples

```
# ifconfig hme0 unplumb
# ifconfig hme0
ifconfig: SIOCGIFFLAGS: hme0: no such interface

# ifconfig hme0 plumb
# ifconfig hme0
hme0: flags=842<BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 0.0.0.0 netmask 0 ether 8:0:20:75:6e:6f
```



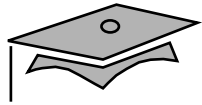
Set IP Address, Enable Interface, and Disable Trailers

```
# ifconfig hme0 inet 128.50.1.2 -trailers up
# ifconfig hme0
hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 128.50.1.2 netmask ffff0000 broadcast 128.50.255.255
    ether 8:0:20:75:6e:6f
```



Change Netmask and Broadcast Value

```
# ifconfig hme0 down
# ifconfig hme0 netmask 255.255.255.0 broadcast + up
# ifconfig hme0
hme0:flags=843<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 128.50.1.2 netmask ffffffff broadcast 128.50.1.255
  ether 8:0:20:75:6e:6f
```



Troubleshooting the Network Interface

- All interfaces are up.
- The IP address is correct.
- The netmask is correct.
- The broadcast address is correct.



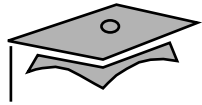
Module 6

Routing



Overview

- Objectives
- Relevance

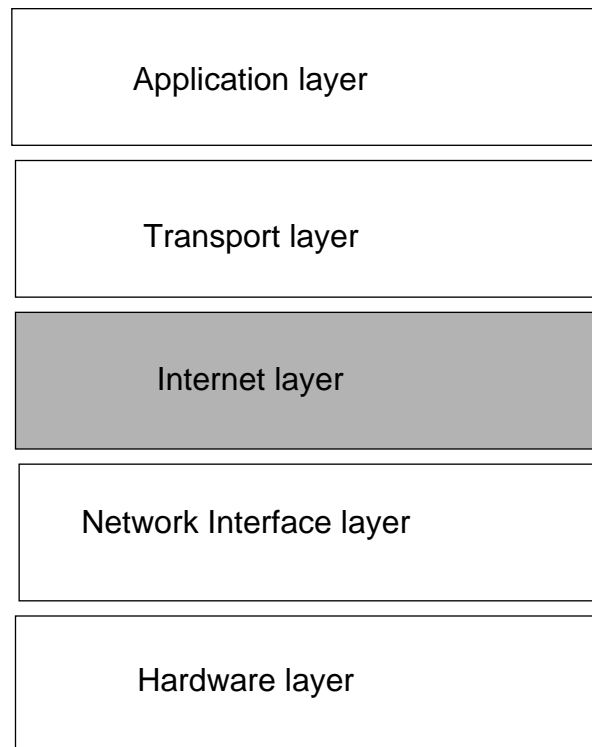


Introduction to Routing

- Mechanism used to forward packets from one network to another
- Critical to LAN communication
- Associated with the Internet layer



Internet TCP/IP Layer





Routing Schemes

- Table-driven routing
- Static routing
- Dynamic routing
- Internet Control Messaging Protocol redirects
- Default routing

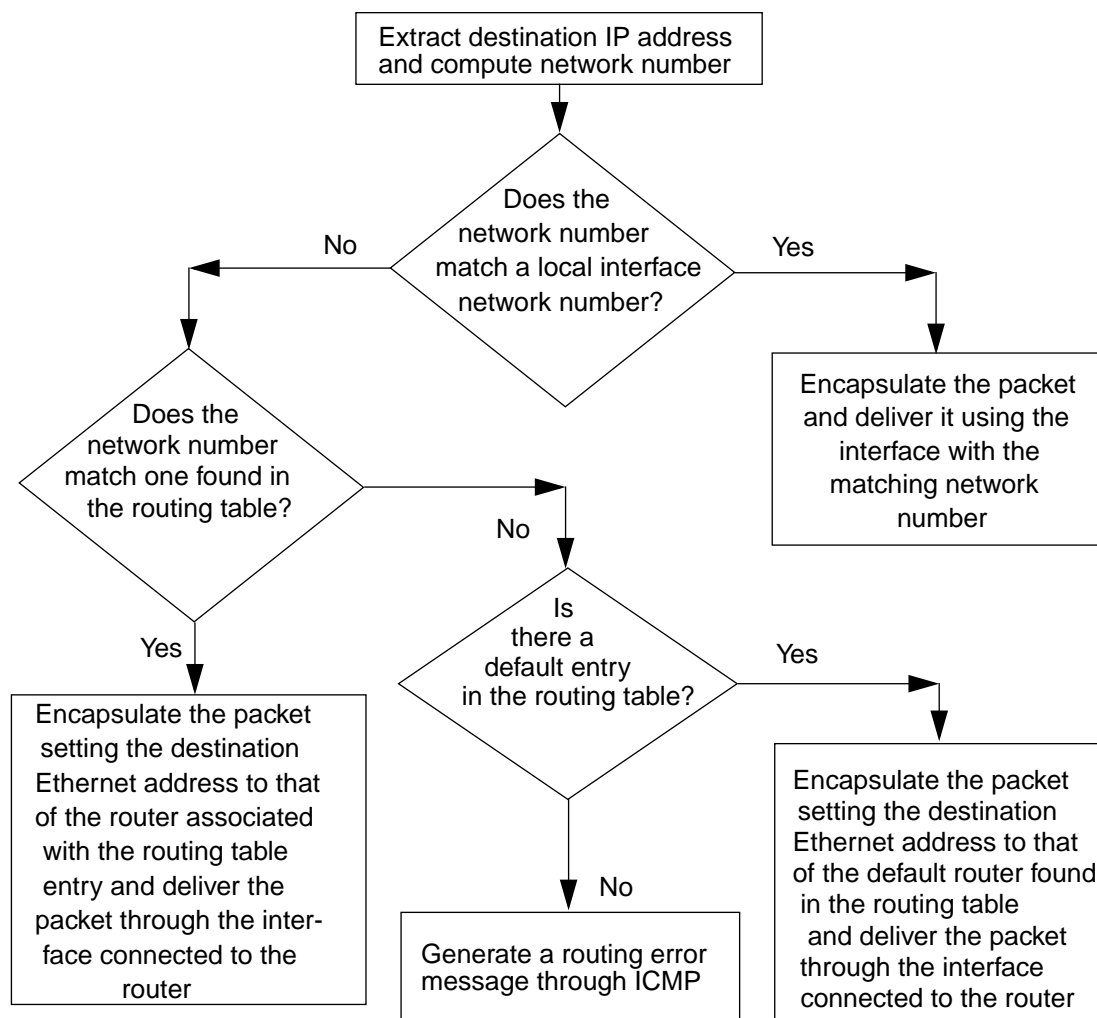


Routing Algorithm

- Check LAN for destination hosts
- Check routing table for matching IP host address
- Check routing table for matching network number
- Check for a *default* entry in the routing table
- If no route to host, generate ICMP error message



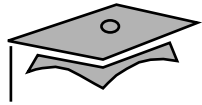
Kernel Routing Process





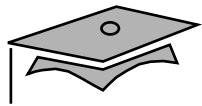
Autonomous System (AS)

- Collection of networks and routers under a single administrative control
- Associated routing table protocols
 - Exterior Gateway Protocol
 - Interior Gateway Protocol

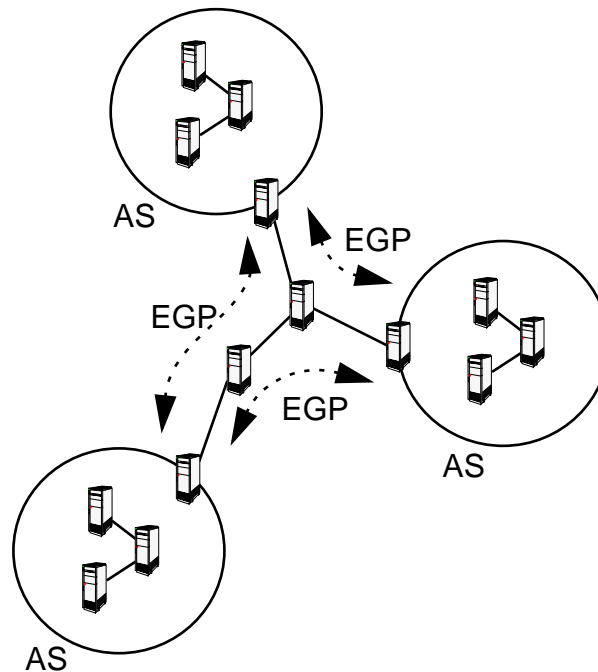


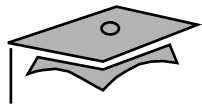
Gateway Protocols

- Exterior Gateway Protocol
 - Exterior Gateway Protocol
 - Border Gateway Protocol
- Interior Gateways Protocol
 - Open Shortest Path First
 - Routing Information Protocol

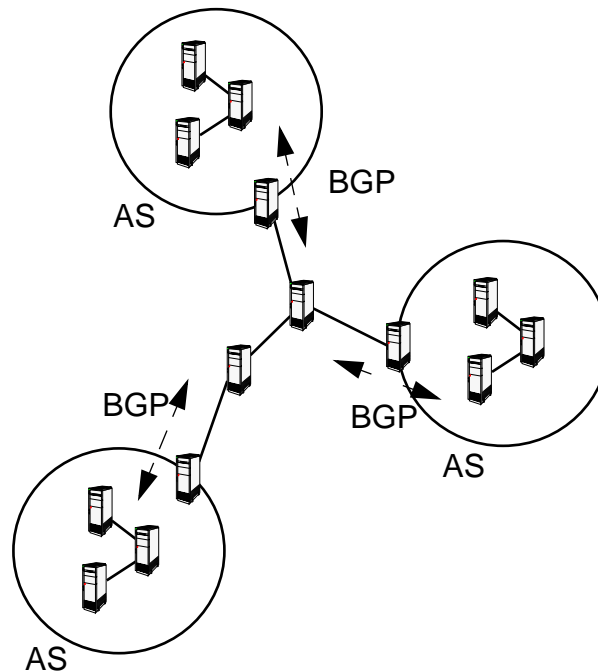


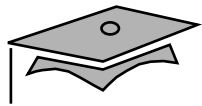
Exterior Gateway Protocol



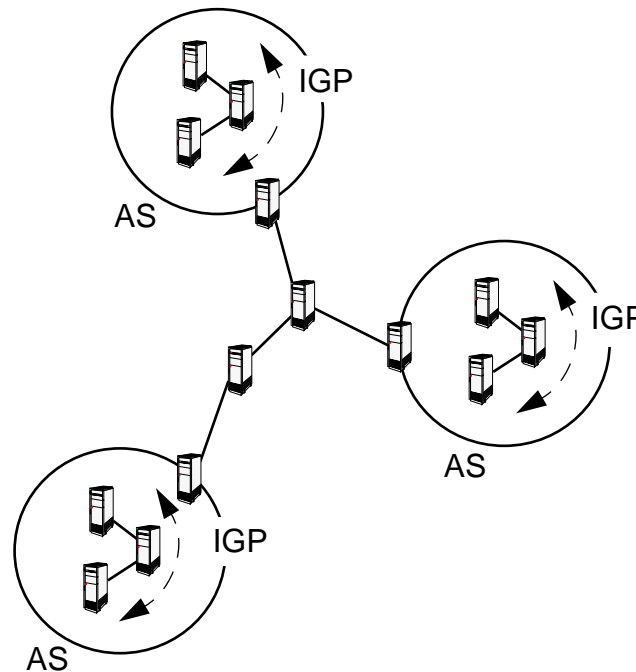


Border Gateway Protocol





Interior Gateway Protocol





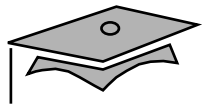
Open Shortest Path First (OSPF)

- Link-state protocol
- Fast, loopless convergency
- Support of multiple metrics
- Multiple paths

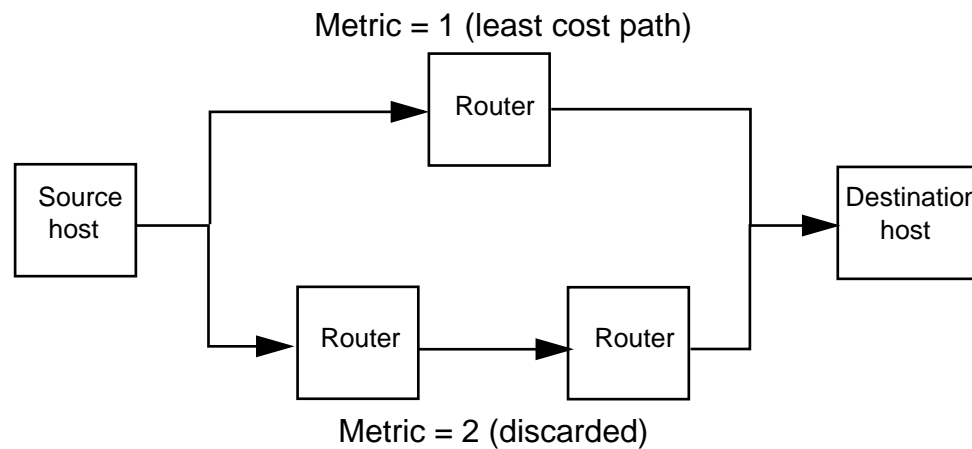


Routing Information Protocol

- Is a distance-vector protocol
- Is a common, easily implemented, and stable protocol
- Updates routing table every 30 seconds
- Updates routing table dynamically



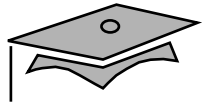
Least Cost Path





Stability Features

- Hop-count limit
- Hold-down state
- Split horizons
- Triggered updates with route poisoning



`/usr/sbin/in.routed`

- Start `in.routed` process in quiet mode

```
# /usr/sbin/in.routed -q
```

- Advertise multi-homed system route

```
# /usr/sbin/in.routed -s
```

- Log `in.routed` process actions

```
# /usr/sbin/in.routed -v /var/adm/routelog
```



Network Router Discovery (RDISC)

- Sends and receives router advertisement messages
- Is implemented through the `in.rdisc` process
- Is routing protocol independent
- Uses multicast address
- Results in smaller routing table
- Uses multiple default route entries to provide redundancy



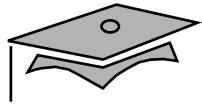
`/usr/sbin/in.rdisc`

- Non-router host

```
# /usr/sbin/in.rdisc -s
```

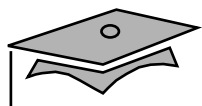
- Router host

```
# /usr/sbin/in.rdisc -r
```

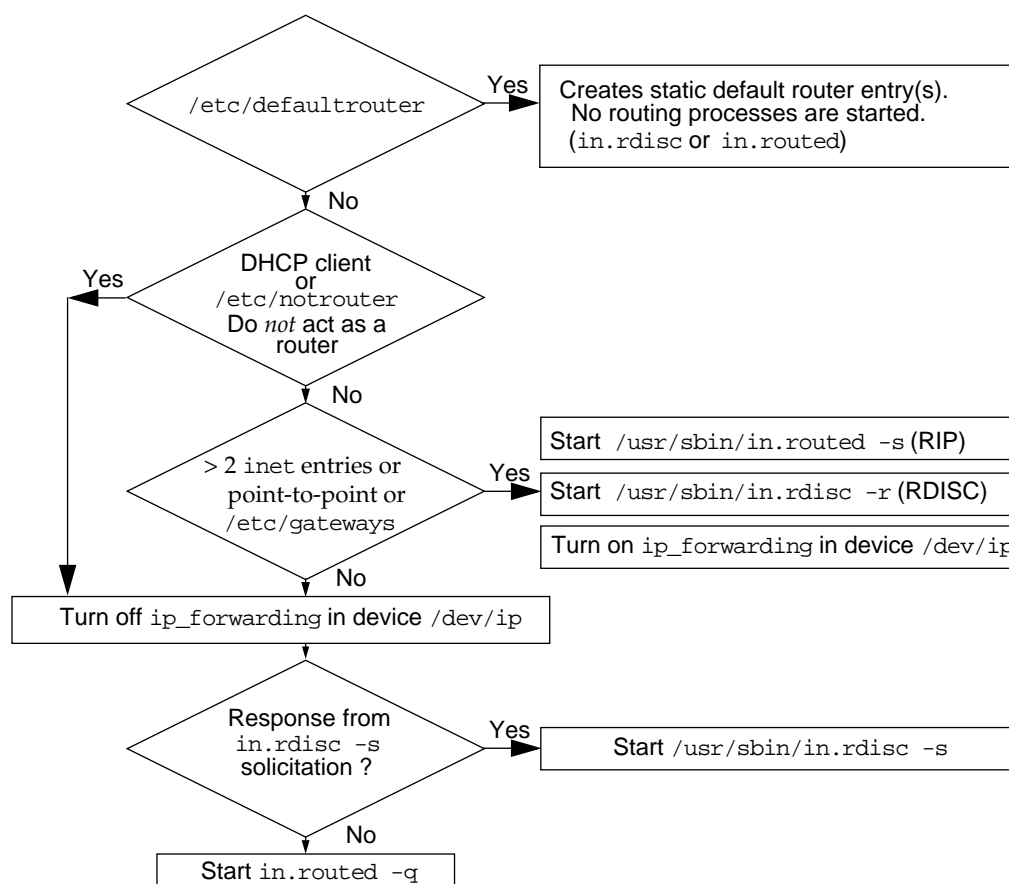


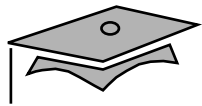
Multihomed Host

- A host with more than two network interfaces that does not run routing protocols or forward IP packets
 - NFS servers
 - Database servers
 - Firewall gateways



Routing Initialization



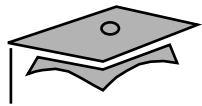


Displaying the Routing Table

```
# netstat -r
```

```
Routing Table:
```

Destination	Gateway	Flags	Ref	Use	Interface
-----	-----	-----	-----	-----	-----
localhost	localhost	UH	0	2272	lo0
128.50.1.0	bear	U	3	562	hme0
128.50.2.0	potato-r	UG	10	1562	
128.50.3.0	skunk	UG	3	562	
default	128.50.1.250	UG	0	1613	
224.0.0.0	bear	U	3	0	hme0



/etc/inet/networks File

A Sample File

```
fish      128.50.3.0      The_School      fish-net
veggie    128.50.2.0      The_Vegetables  veggie-net
zoo       128.50.1.0      The_Animals     zoo-net
```

netstat -r

Routing Table:

Destination	Gateway	Flags	Ref	Use	Interface
-----	-----	-----	-----	-----	-----
localhost	localhost	UH	0	2272	lo0
zoo	bear	U	3	562	hme0
veggie	potato-r	UG	10	1562	
fish	skunk	UG	3	562	hme0
224.0.0.0	bear	U	3	0	hme0



Manually Manipulating Routing Table

- Add a route

```
# route add net 128.50.3.0 skunk 1
```

- Add a route using a network name

```
# route add net Animal -net potato-r 1
```

- Delete a route

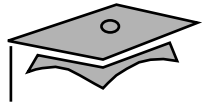
```
# route delete net 128.50.2.0 sword-r
```

- Flush routing table

```
# route flush
```

- Add multicast path for 224.0.0.0

```
# route add 224.0.0.0 `uname -n` 0
```



/etc/gateways File

- Is used by `in.routed` to build its routing table
- Has the syntax:

```
net dest.net gateway router metric cnt [passive][active]
```



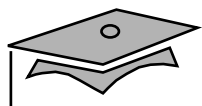
Router Configuration

- Create a `/etc/hostname.interface` file
- Edit the file `/etc/inet/hosts`
- Perform a reconfigure boot
- Verify the new interface parameters

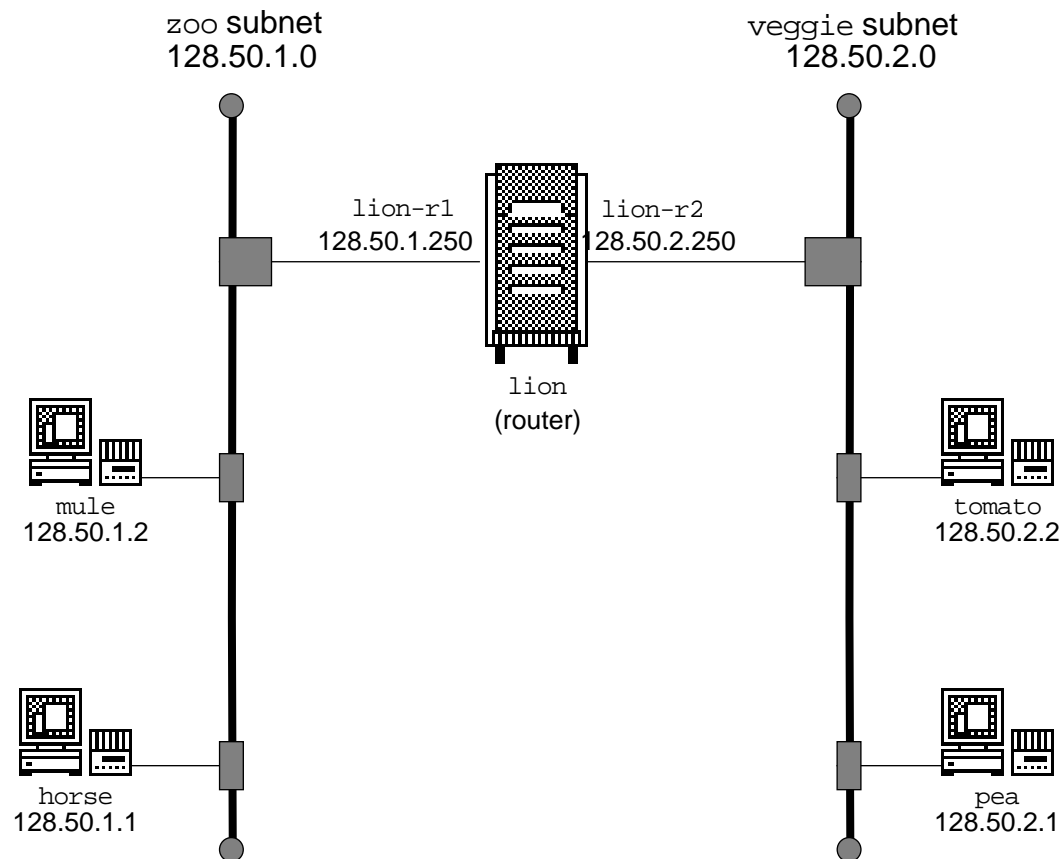


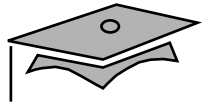
Troubleshooting Router Configuration

- Check device information
- Check `ifconfig` information
- Verify correct device and file name
- Verify correct IP address



Lab Network Configuration





Module 7

Transport Layer



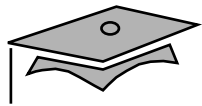
Overview

- Objectives
- Relevance

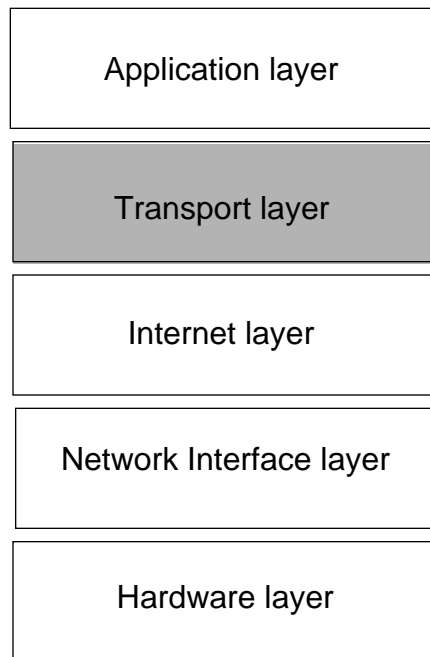


Introduction to the Transport Layer

- End-to-end communication
- Destination port number
- Data segmenting



TCP/IP Layered Model





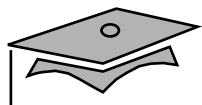
Types of Protocols

- Connectionless
 - Has virtually no reliability features
 - Requires that transmission quality be augmented
 - Is very fast
- Connection oriented
 - Is highly reliable
 - Requires more computational processing



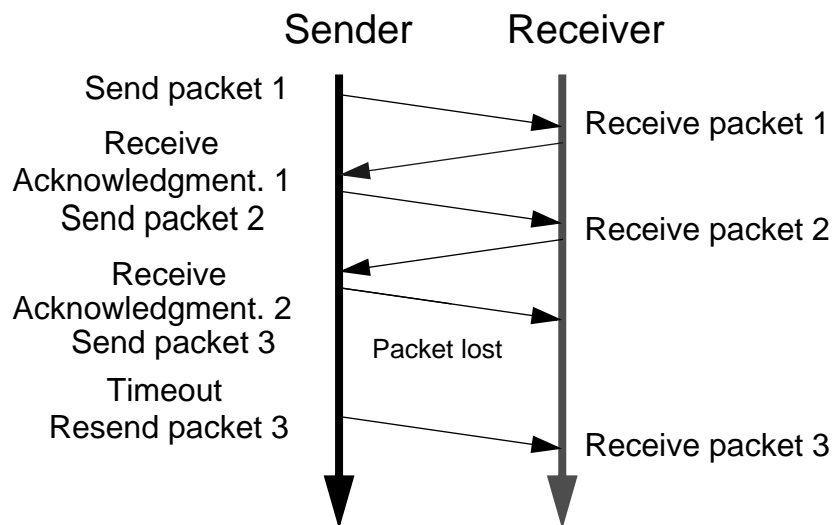
Stateful Versus Stateless

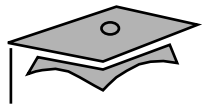
- Stateful – Data includes the state of the client
- Stateless – Data does not include the state of the client



Reliable Protocols

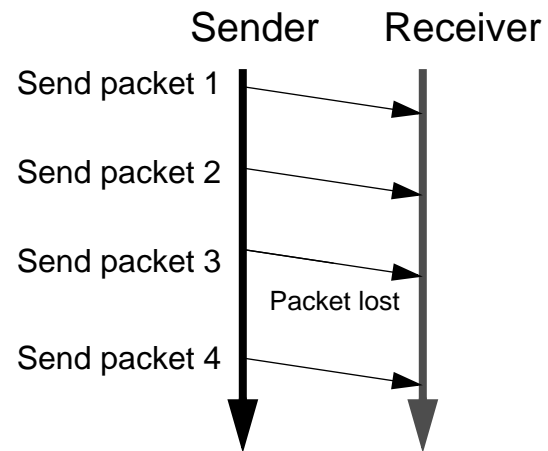
- Requires transmission acknowledgment

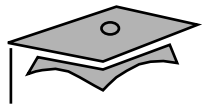




Unreliable Protocols

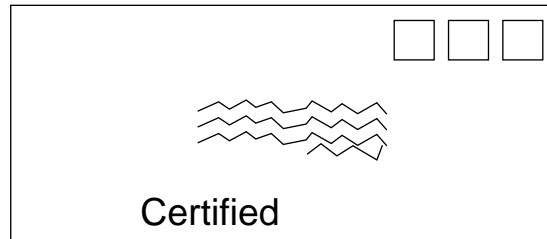
- No transmission acknowledgment



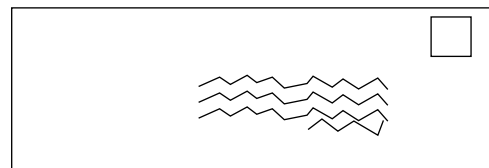


Transport Protocols

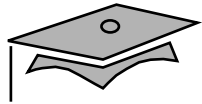
- Transport Control Protocol (TCP)
- User Datagram Protocol (UDP)



TCP



UDP



Transport Layer Protocol Features

Features	UDP	TCP
Connection oriented	No	Yes
Message boundaries	Yes	No
Data checksum	Optional	Yes
Positive acknowledgment	No	Yes
Timeout and retransmit	No	Yes
Duplicate detection	No	Yes
Sequencing	No	Yes
Flow control	No	Yes



User Datagram Protocol

- Unreliable and connectionless
- Non-acknowledged
- Datagrams



Transmission Control Protocol

- Unstructured stream orientation
- Virtual circuit connection
- Buffered transfer
- Full duplex connection



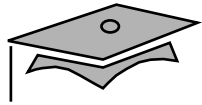
TCP Flow Control

- Sliding window principle
- Congestion window



Module 8

Client-Server Model



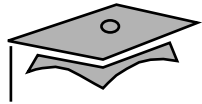
Overview

- Objectives
- Relevance

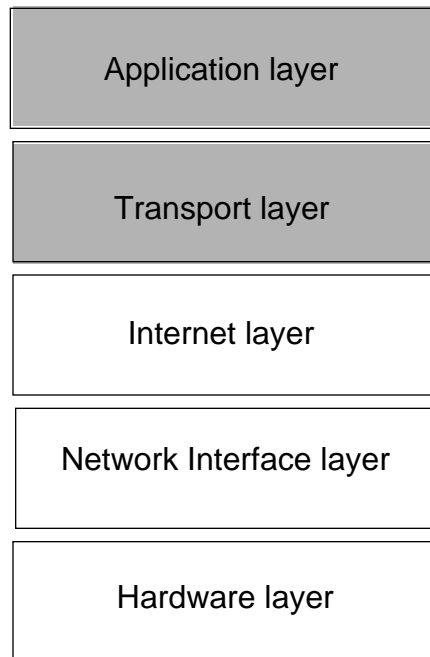


The Client-Server Model

- Service
- Client
- Server
- TCP/IP model



Application Layer





ONC+ Technologies

- Is Sun™'s open systems distributed computing environment
- Provides core services to developers
- Includes tools to administer client/server networks



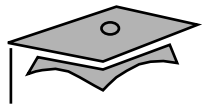
ONC+ Distributed Computing Platform

RPC application programs	
TI-RPC	XDR
TLI	Sockets
TCP or UDP port numbers	



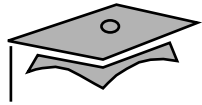
ONC+ Technologies

- XDR
- TLI
- Sockets
- XDR
- NFSTM
- NIS+



Port Numbers

- Address space
- Arbitrary port
- Well-known port
- Unique port number
- `/etc/inet/services`
- Reserved ports



/etc/inet/services Extract

ftp-data	20/tcp	
ftp	21/tcp	
telnet	23/tcp	
smtp	25/tcp	mail
sunrpc	111/udp	rpcbind
sunrpc	111/tcp	rpcbind



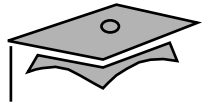
How a Server Process Is Started

- Server process responds to a client request
- Process starts at run level 2 and additional services at level 3
- Some services start by demand
- The `inetd` process is started
- The `/etc/inet/inetd.conf` file is read



Remote Procedure Call

- Many unique port numbers are required
- `rpcbind` is used
- The `/etc/inet/inetd.conf` file is used



Status Commands

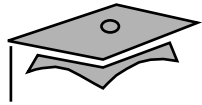
- `/usr/bin/rpcinfo`
- `/usr/bin/netstat -a`



```
/usr/bin/rpcinfo -p
```

```
# rpcinfo -p [hostname]
```

program	ver	proto	port	service
100000	4	tcp	111	portmapper
100007	1	udp	32771	ypbind
100008	1	udp	32803	walld
100012	1	udp	32805	sprayd



```
/usr/bin/rpcinfo -b
```

```
# rpcinfo -b mountd 1
```

```
192.9.200.10.199    servera  
192.9.200.13.187    serverb
```

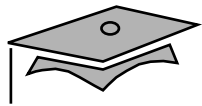


```
/usr/bin/rpcinfo -u
```

```
# rpcinfo -u servera mountd
```

```
program 100005 version 1 ready and waiting
```

```
program 100005 version 2 ready and waiting
```



/usr/bin/netstat -a

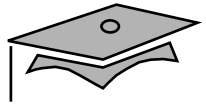
```
# /usr/bin/netstat -a
```

```
UDP
```

Local Address	State
-----	-----
*.route	Idle
.	Unbound
*.sunrpc	Idle
*.nfsd	Idle

```
TCP
```

Local Address	Address	Swind	Send-Q	Rwind	Recv-Q	State
-----	-----	-----	-----	-----	-----	-----
.	*.*	0	0	8576	0	Idle
*.ftp	*.*	0	0	8576	0	LISTEN
*.telnet	*.*	0	0	8576	0	LISTEN
*.login	*.*	0	0	8576	0	LISTEN
*.sunrpc	*.*	0	0	8576	0	LISTEN
chesapeake.login	yogi.1023	16384	0	16384	0	ESTABLISHED



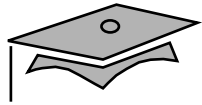
Module 9

DHCP



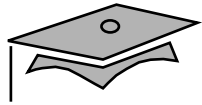
Overview

- Objectives
- Relevance



Dynamic Host Configuration Protocol

- Supports centrally located network administration
- Automates assignment of Internet Protocol (IP) addresses
- Reduces cost of managing networks
- Provides a solution for the rapid depletion of IP addresses



How DHCP Uses BOOTP

- Offers re-usable IP addresses
- Eliminates the need to set up a BOOTP table
- Permits the allocation of an IP address based on
 - Physical connection to a particular subnet
 - A client identification string designated by the network manager
 - A hardware address of the Ethernet card



DHCP Features

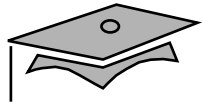
- Automatic management of IP addresses
- Support for BOOTP clients
- Programmable lease times
- Dynamic IP addresses assigned to selected Ethernet hardware addresses
- Dynamically allocated pool or pools of IP addresses on the same network
- Two or more dynamic IP address pools on separate IP networks (or subnets)



DHCP Client/Server

The DHCP protocol has two functions with regard to the client:

- Establish an endpoint for network communications
- Provide system- and application-level software parameters



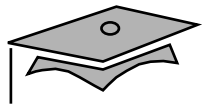
Server Side

- DHCP server manages the IP address space of networks directly connected to that server.
- BOOTP relay agents allow forwarding of DHCP or BOOTP requests to server on other networks.
- Servers are configured as primary and/or secondary.
 - Primary server passes IP addresses to the client.
 - Secondary server confirms existing configurations.



Server Databases

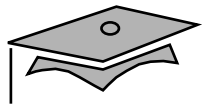
- *dhcp_network* – Client identifier to an IP address and the associated configuration parameters of that address
- *dhcptab* – Information related to client configuration



dhcp_network Entry Format

`Client_ID|Flags|Client_IP|Server_IP|Lease|Macro`

- `Client_ID` – Unique identifier of DHCP client
- `Flags` – The dispensation of the IP address
- `Client_IP` – IP address to be assigned
- `Server_IP` – Primary server of the IP address
- `Lease` – Absolute lease expiration time
- `Macro` – Macro to be passed as defined in `dhcptab`



dhcp_network Examples

Client_ID	Flags	Client_IP	Server_IP	Lease	Macro
00	00	129.146.86.205	129.146.86.181	0	inet01
010800209b0d45	03	129.146.86.205	129.146.86.181	-1	inet07
010800209b0d45	00	129.146.86.205	129.146.86.181	905704239	inet01
00	04	129.146.86.205	129.146.86.181	0	inet01



dhcptab Entry Format

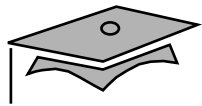
Name | Type | Value

- Name – Identifies the record and is used as the search key to the dhcptab table
- Type – Specifies the type of record; symbol or macro
- Value – Contains the value for the specified record type



Symbols and Macros

- `Symbol` – Defines vendor- and site-specific options
- `Macro` – Contains information which determines how client machines access a network



Symbol Characteristics

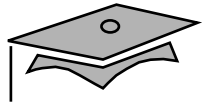
Context | Code | Type | Granularity | Maximum

- Context – Context in which the Symbol definition is to be used; Extend, Site, or Vendor=Client Class
- Code – Option code number assigned to Symbol
- Type – Type of data expected as a value for this Symbol
- Granularity – How many objects of Type define a single instance of the Symbol value
- Maximum – Granularity in a definition using Symbol



Macro Definitions

- Client Class
- Network
- IP Address
- Client Identifier



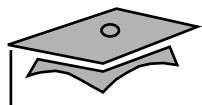
Lease Time Policy

- Can be set to permanent or temporary
- Is defined in the `dhcptab` file
 - `LeaseTim`
 - `LeaseNeg`



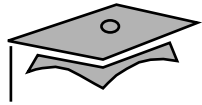
Lease Flags (*dhcp_network*)

- Indicates the conditions under which the IP address can be assigned
- Can be set to a combination of the following:
 - 0 (Dynamic)
 - 1 (Permanent)
 - 2 (Manual)
 - 4 (Unusable)



dhcptab Examples

Name	Type	Value
SN_TZ	Symbol	Vendor=SUNW,13,ASCII,1,0
SUNW	Macro	:UTCoffst=25200:SN_TZ="PST8PDT":
inet01	Macro	:Include=SUNW:Timeserv=129.146.86.181:\ :LeaseTim=72:DNSdmain=Eng.Sun.COM: \ :DNSserv=129.146.1.151 129.146.1.152 \ 129.144.1.57 129.144.134.19:LeaseNeg:



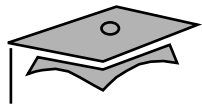
DHCP Administration Commands

- `pntadm` – Manages *dhcp_network*
- `dhtadm` – Manages *dhcptab*



DHCP Server Configuration

- Collect information about network.
- Decide whether to store data in NIS+ or in local files.
- Run the `dhcpcfg` utility to install DHCP on server



Configuring DHCP on the Server

*** DHCP Configuration ***

Would you like to:

- 1) Configure DHCP Service
- 2) Configure BOOTP Relay Agent
- 3) Unconfigure DHCP or Relay Service
- 4) Exit

Choice:



Configuring DHCP on the Client

- By default, the Solaris DHCP client is disabled.
- To enable it, create a `/etc/dhcp.interface_name` for each network interface you want to configure with DHCP.

Example for interface `hme1`:

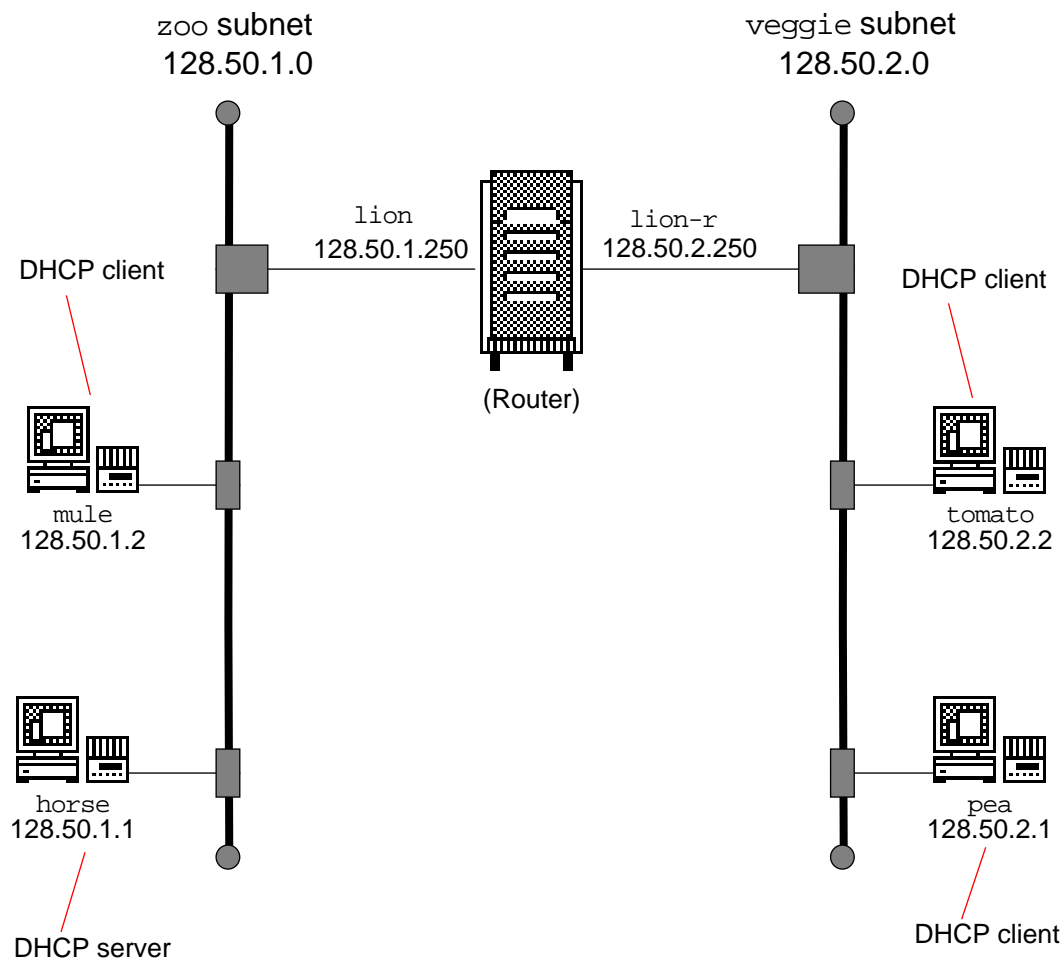
```
# touch /etc/dhcp.hme1
```

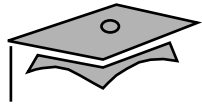


Troubleshooting DHCP

- snoop command
- DHCP client debug mode
- DHCP server debug mode
- Reboot
- DHCP server daemon

DHCP Lab Network Configuration





Module 10

Introduction to Network Management Tools



Overview

- Objectives
- Relevance



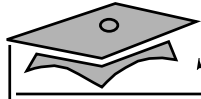
Network Management

- ISO defined
 - Configuration management
 - Fault management
 - Performance management
 - Accounting management
 - Security management
- Management system, network management application, and device to manage

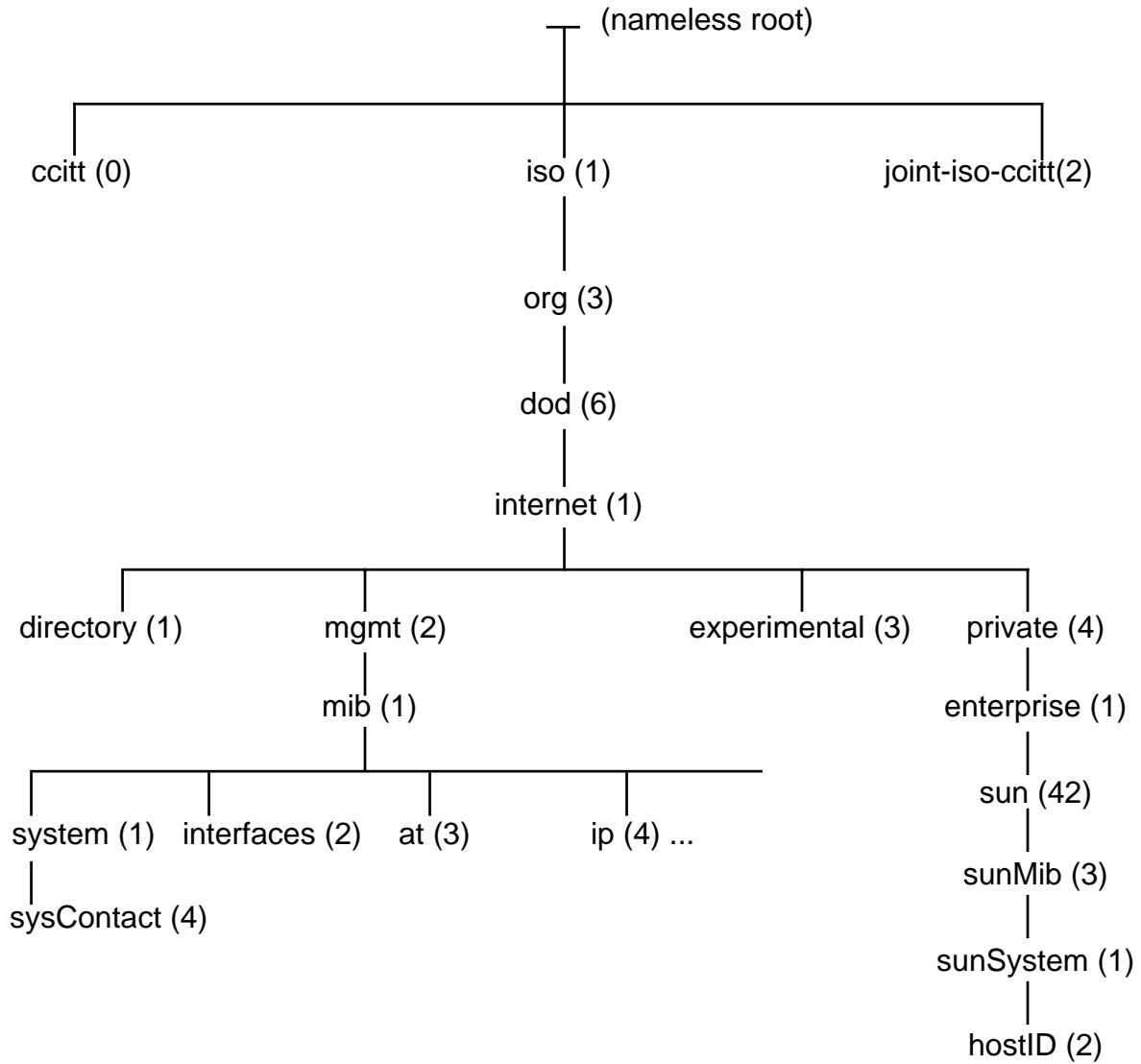


Simple Network Management Protocol

- IP based, uses UDP
- SNMP functions
 - Get
 - Set
 - Trap
- SNMP structure
 - Structure of management information (SMI)
 - Object identifier (OID)



OID Global Tree





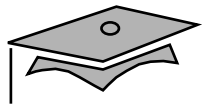
Simple Network Management Protocol

- Management Information Base (MIB)
- ASN.1

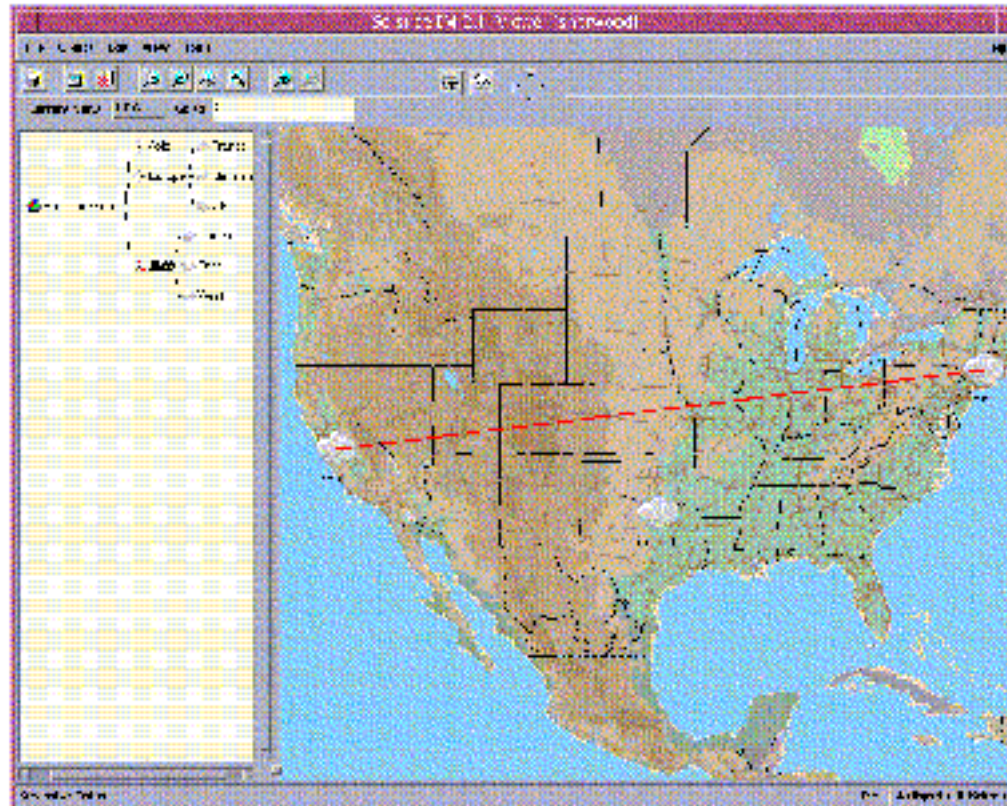


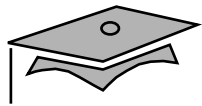
SNMP-based Management Applications

- Solstice Site Manager™
- Solstice Domain Manager™
- Solstice Enterprise Manager™
- Solstice Enterprise Agents™

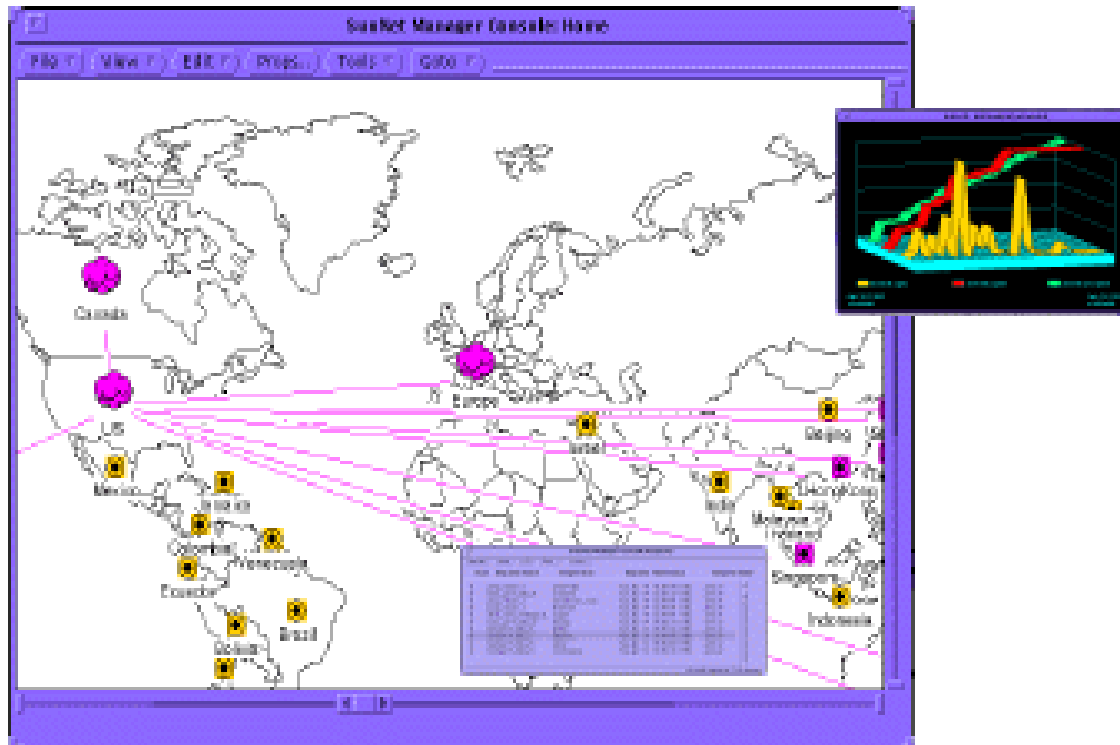


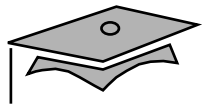
Solstice Site Manager



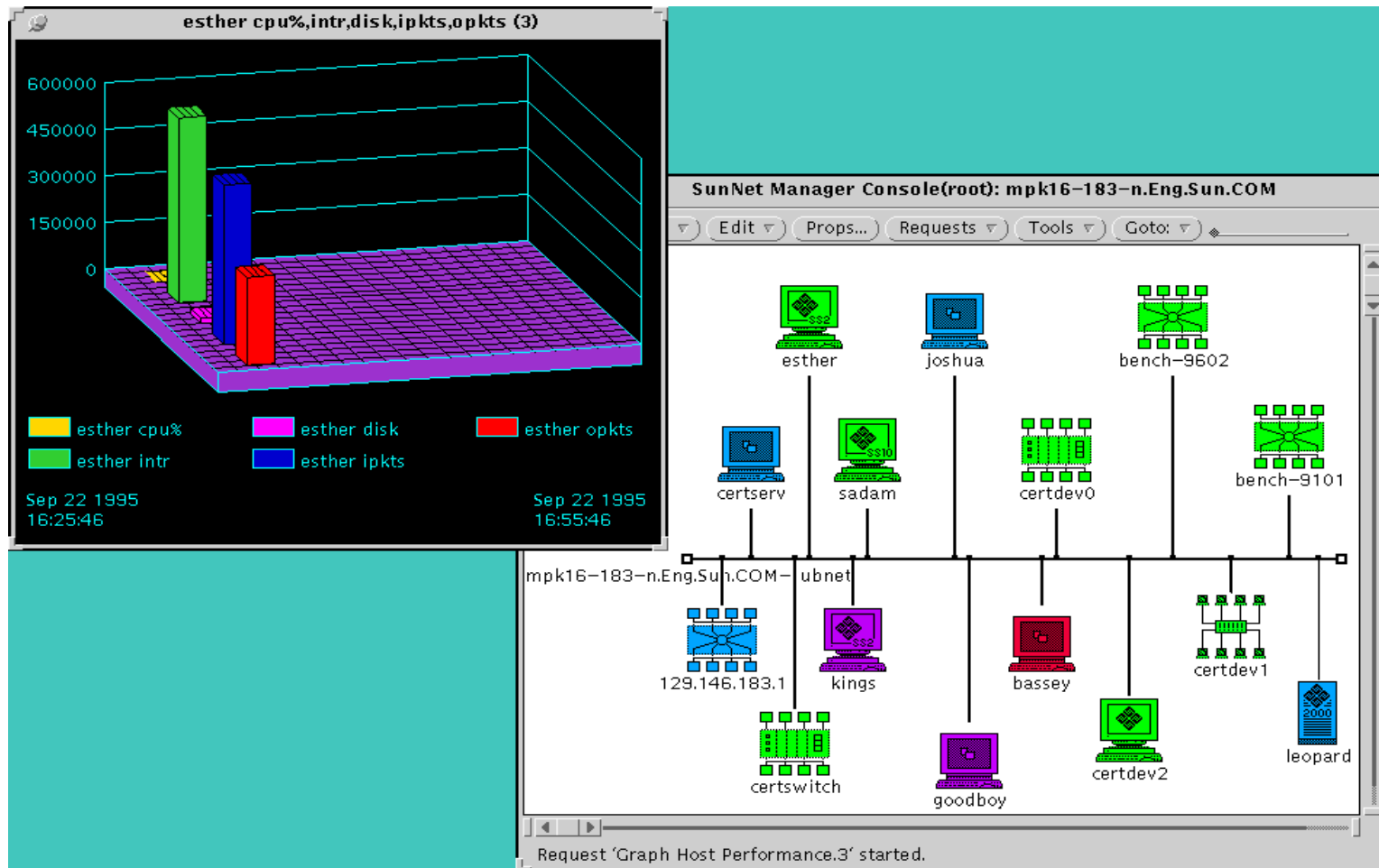


Solstice Domain Manager





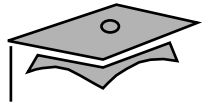
Solstice Enterprise Manager





Module 11

Domain Name System



Overview

- Objectives
- Relevance



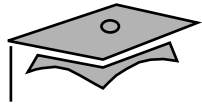
Why DNS – A Brief History

- Early Internet naming problems
 - Name uniqueness
 - HOSTS .TXT file maintenance
 - Server/network load
- The solution
 - Name uniqueness
 - HOSTS .TXT file maintenance
 - Server/network load



DNS Namespace – Domains

- Is a collection of names
- Specifies keys for DNS look up
- Is an inverted tree structure
- Is capable of spanning a large physical area
- Can be broken into subdomains
- Supports parent/child domain relationships



DNS Namespace – Structure

- Nameless root domain
- Top-level domains

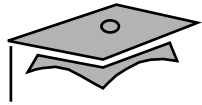
Domain	Description
com	Commercial organizations
edu	Educational organizations
gov	Governmental (U.S.) organizations
mil	Military (U.S) organizations
net	Networking organizations and ISPs
org	Non-profit and other organizations
arpa	Used mainly for inverse address lookups
ca	Country based domains, Canada in this example

- Second-level domains
- Lower-level domains



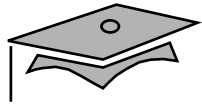
DNS Namespace – Domain Naming

- Fully qualified name of a domain (FQDN)
- Relative domain name (RDN)
- Domain naming rules
 - A 255 character limit per FQDN
 - A 63 character limit per domain
 - Only alphas, numerics, and the dash are permitted
 - Naming conventions decided by domain administrator
- `in-addr.arpa.domain`



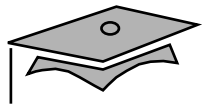
Zones of Authority

- Is the portion of the name space for which a server is authoritative
- Consists of domains and all associated data
- Can be one or more domains



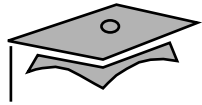
DNS Servers

- Root servers
- Primary (master) servers
- Secondary (slave) servers
- Caching-only servers
- Forwarding servers



DNS Answers

- Authoritative
 - Are from primary or secondary authoritative servers
 - May not be correct
 - Are “as good as it gets”
 - Are typically correct
- Non-authoritative
 - Are from cache of non-authoritative server
 - Are typically correct
 - May be incorrect

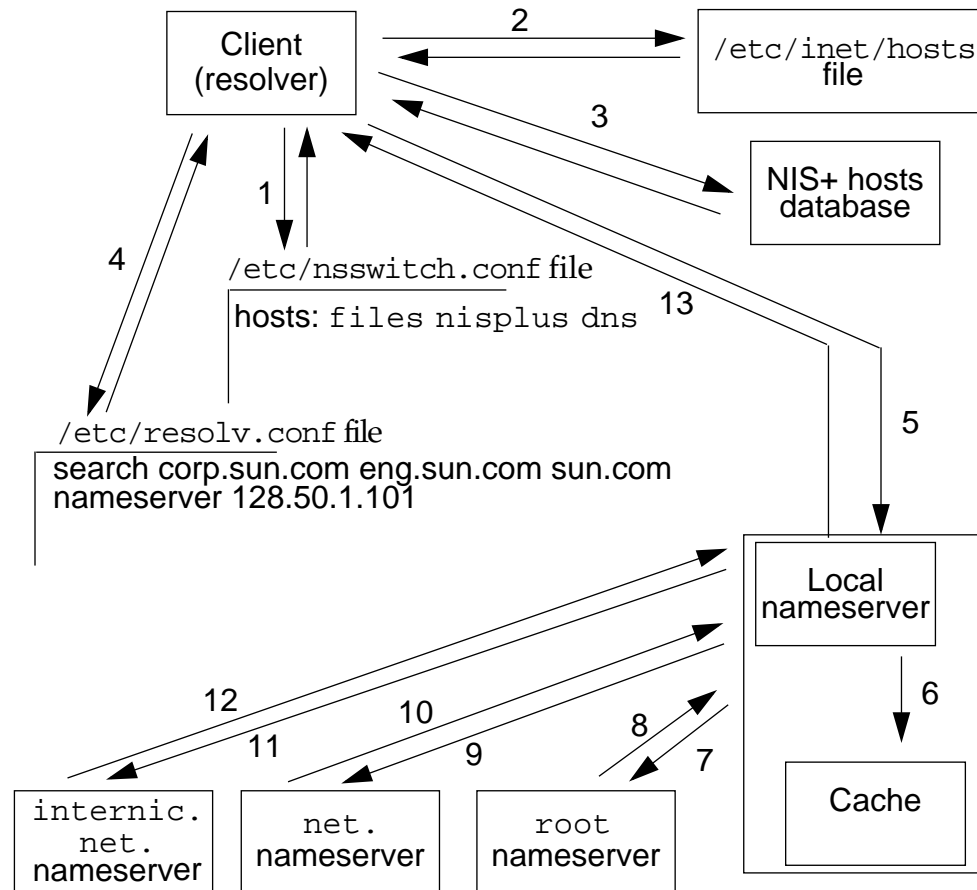


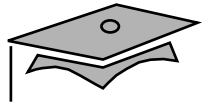
Client Resolver

- Simplified interfaces to the local DNS server
- Queries to local DNS server
 - `/etc/resolv.conf`
- Local DNS server replies
 - From cache or remote server



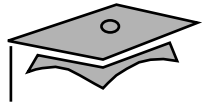
Resolution Process





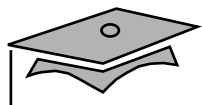
BIND

- Most frequently used DNS implementation
- Available at <http://www.isc.org/bind.html>
- Solaris 7 implements BIND Version 8.1.2
- Latest BIND version may not be supported



DNS Server Configuration

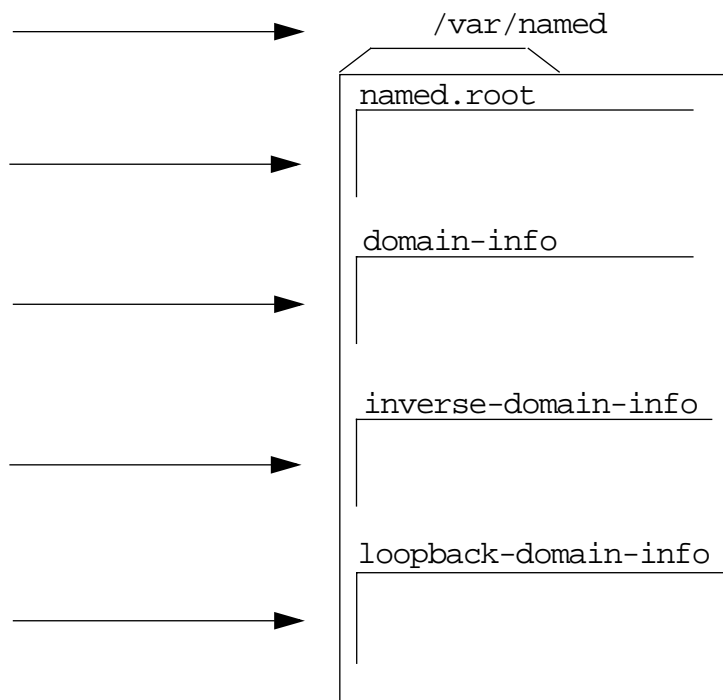
- Location of names and addresses of root servers
- Information to resolve all domains for which the server is authoritative
- Information to resolve all inverse domains for which the server is authoritative
- Location of servers one level below the domain being served

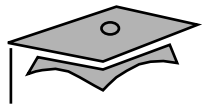


named.conf - BIND Configuration File

/etc/named.conf

```
options {  
    DIRECTORY "/var/named";  
};  
zone "." in {  
    type hint;  
    file "named.root";  
};  
zone "zoo.edu" in {  
    type master;  
    file "domain-info";  
};  
zone "1.50.128.in-addr.arpa" in {  
    type master;  
    file "inverse-domain-info";  
};  
zone "127.in-addr.arpa" in {  
    type master;  
    file "loopback-domain-info";  
};
```





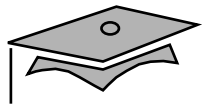
/etc/named.conf Statement Definitions

Statement	Definition
acl	Defines a named IP address match list used for access control. The address match list designates one or more IP addresses or IP prefixes. The named IP address match list must be defined by an acl statement before it can be used elsewhere; no forward references are allowed.
include	Inserts an include file at the point where the include statement is encountered. Use include to break up the configuration into more easily managed chunks.
key	Specifies a key ID used for authentication and authorization on a particular name server. See the server statement.
logging	Specifies the information the server logs and the destination of log messages.
options	Controls global server configuration options and sets default values for other statements.
server	Sets designated configuration options associated with a remote name server. Selectively applies options on a per-server basis, rather than to all servers
zone	Defines a zone. Selectively applies options on a per-zone basis, rather than to all zones.



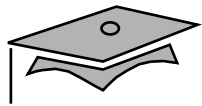
DNS Resource Records

- Contains records in the name server database file
- Contains information pertaining to a particular machine
- Uses format which includes:
 - Domain name
 - Time to live
 - Class
 - Record type
 - Record data



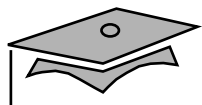
Resource Record Types

Record Type	Purpose
A	The A record (address record) yields an IP address that corresponds to a host name. There can be multiple IP addresses corresponding to a single host name; there can also be multiple host names, each of which maps to the same IP address.
CNAME	The CNAME (Canonical Name) record is used to define an alias host name.
MX	MX records specify a list of hosts that are configured to receive mail sent to this domain name.
NS	Each subdomain that is a separate nameserver must have at least one corresponding name service (NS) record. Name servers use NS records to find each other.
PTR	PTR allows special names to point to some other location in the domain. PTR records are used only in reverse (IN-ADDR.ARPA) domains. There must be exactly one PTR record for each Internet address.
SOA	Start of Authority (SOA) record identifies who has authoritative responsibility for this domain.
TXT	The TXT (text) record allows you to associate any arbitrary text with a host name.



`/var/named/named.root` File

- Specifies name-to-address mappings root servers
- Provides “hints” as to the identity of root servers
- Uses hints to determine actual root servers
- Reuses hints when cache information times out
- Is available at
`ftp://ftp.rs.internic.net/domain/named.root`



named.root File Excerpt

```
; formerly NS.INTERNIC.NET
```

```
.                IN NS      A.ROOT-SERVERS.NET.
```

```
A.ROOT-SERVERS.NET.  IN A      198.41.0.4
```

```
; formerly NS1.ISI.EDU
```

```
.                IN NS      B.ROOT-SERVERS.NET.
```

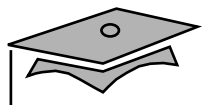
```
B.ROOT-SERVERS.NET.  IN A      128.9.0.107
```

```
.
```

```
.
```

```
.
```

```
; End of File
```



domain-info File

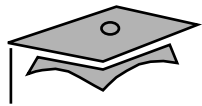
```
; Information for the "forward" domain zoo.edu.

; The SOA record must be present and must be first.

@                IN SOA horse.zoo.edu.
hostmaster.zoo.edu. (
                    1          ; Serial number
                    43200     ; Refresh timer - 12 hours
                    3600     ; Retry timer - 1 hour
                    604800   ; Expire timer - 1 week
                    86400    ; Minimum timer - 1 day
                    )

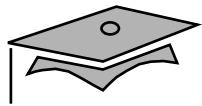
; Define name servers for this domain.

                    IN NS  horse.zoo.edu   ; primary
                    IN NS  pea.veggie.edu  ; secondary
                    IN NS  tuna.fish.edu   ; secondary
```



domain-info File

```
pea.veggie.edu.          IN A    128.50.2.1
tuna.fish.edu.          IN A    128.50.3.1
; Define name to address mappings for this domain.
lion                    IN A    128.50.1.250
|                       IN A    128.50.2.250
lion-r2                 IN A    128.50.2.250
rino                   IN A    128.50.1.3
mule                   IN A    128.50.1.2
horse                  IN A    128.50.1.1
; CNAME aliases.
www                    IN CNAME two
; Loopback domain definition (required).
localhost              IN A    127.0.0.1
```



inverse-domain-info File

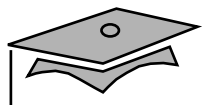
```
; Information for the "inverse" domain 1.50.128.in-addr.arpa.
@
hostmaster.zoo.edu. (
                        1          ; Serial number
                        43200     ; Refresh timer - 12 hours
                        3600      ; Retry timer - 1 hour
                        604800    ; Expire timer - 1 week
                        86400     ; Minimum timer - 1 day
                        )
; Define name servers for this domain.
                        IN NS  horse.zoo.edu. ; primary
                        IN NS  pea.veggie.edu. ; secondary
                        IN NS  tuna.fish.edu. ; secondary
```



inverse-domain-info File

`; Define address to name mappings for this domain.`

```
250          IN PTR lion.zoo.edu.  
3           IN PTR rino.zoo.edu.  
2           IN PTR mule.zoo.edu.  
1           IN PTR horse.zoo.edu.
```



loopback-domain-info File

```
; Information for the loopback domain 127.in-addr.arpa.

@                IN SOA horse.zoo.edu.
hostmaster.zoo.edu. (

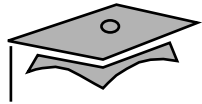
                    1          ; Serial number
                    43200     ; Refresh timer - 12 hours
                    3600      ; Retry timer - 1 hour
                    604800    ; Expire timer - 1 week
                    86400     ; Minimum timer - 1 day
                    )

; Define name servers for this domain.

                    IN NS  horse.zoo.edu.

; Define appropriate mappings for this domain.

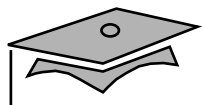
1.0.0             IN PTR localhost.zoo.edu.
```



/etc/nsswitch.conf

- Name resolution method and ordering
- Example

```
hosts: files nisplus dns
```



/etc/resolv.conf

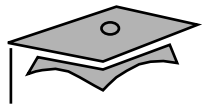
- Search list when names are not FQDN
- Example

```
; resolv.conf file for DNS clients of the zoo.edu.domain.  
search zoo.edu edu  
nameserver 128.50.1.1      ; Primary Master Server for zoo  
nameserver 128.50.1.250 ; Root server (not usually a good idea!)
```



nslookup

- Send queries to and display replies from any resource record types
- Query the DNS server of choice
- Debug domain that is not protected by a firewall



nslookup Examples

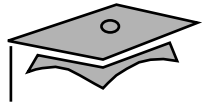
```
horse# nslookup
Default Server:  horse.zoo.edu
Address:  128.50.1.1

> lion.zoo.edu.
Server:  horse.zoo.edu
Address:  128.50.1.1

Name:  lion.zoo.edu
Address:  128.50.1.250

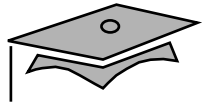
> set type=ns
> zoo.edu.
...
zoo.edu. nameserver = horse.zoo.edu
horse.zoo.edu  internet address = 128.50.1.1

> set type=ptr
> 128.50.1.1
...
1.1.50.128.in-addr.arpa name = horse.zoo.edu
```



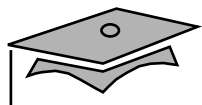
BIND Debugging Tools

- `pskill -INT in.named`
- `pskill -USR1 in.named`
- `pskill -USR2 in.named`
- `pskill -HUP in.named`



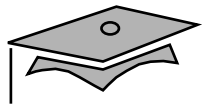
Secondary DNS Server Setup

- `/etc/named.conf` file on the secondary server
- `/var/named/domain-info` file on primary server
- Testing and debugging



named.conf File – Secondary Server

```
options {
    DIRECTORY "/var/named";
};
zone "." in {
    type hint;
    file "named.root";
};
| zone "127.in-addr.arpa" in {
    type master;
    file "loopback-domain-info";
};
zone "zoo.edu" in {
    type slave;
    file "zoo-backup";
    masters {
        128.50.1.1;
    };
};
```



DNS Security

- Using BIND configuration file
- Restricting queries
 - Restricting all queries
 - Restricting queries in a particular zone
- Preventing unauthorized zone transfers
 - Authorizing zone transfer
 - Authorizing global zone transfer



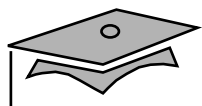
Miscellaneous DNS Topics

- DNS configuration file \$ directives
 - \$ORIGIN domain.name.
- h2n
- DIG



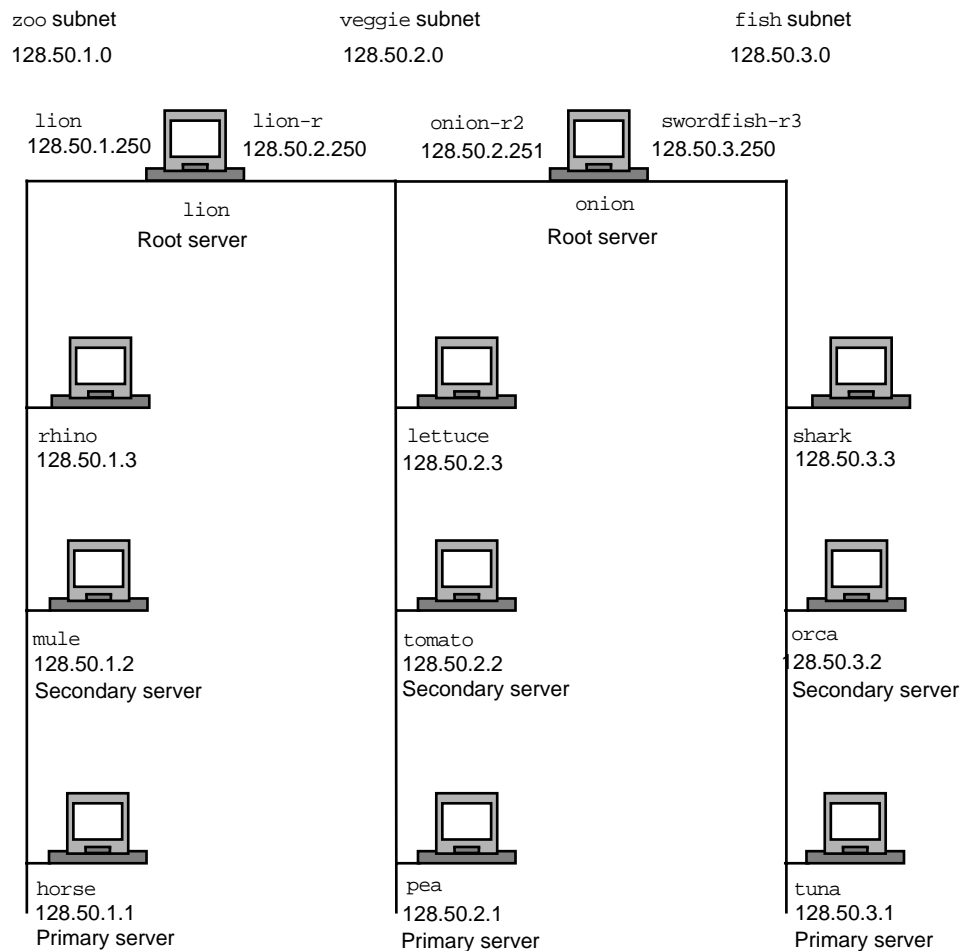
DNS Resources

- `info.bind` newsgroup
- `www.internic.net`.
- RFCs



DNS Lab Layout

Domain: edu.





Module 12

Email Fundamentals



Overview

- Objectives
- Relevance



Introduction to Electronic Mail

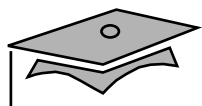
Electronic mail (email) is the exchange of computer-stored messages by telecommunication. It:

- Supports communication within the local domain and among other, external domains
- Has a history
 - Standardized by Internet Engineering Task Force (IETF)

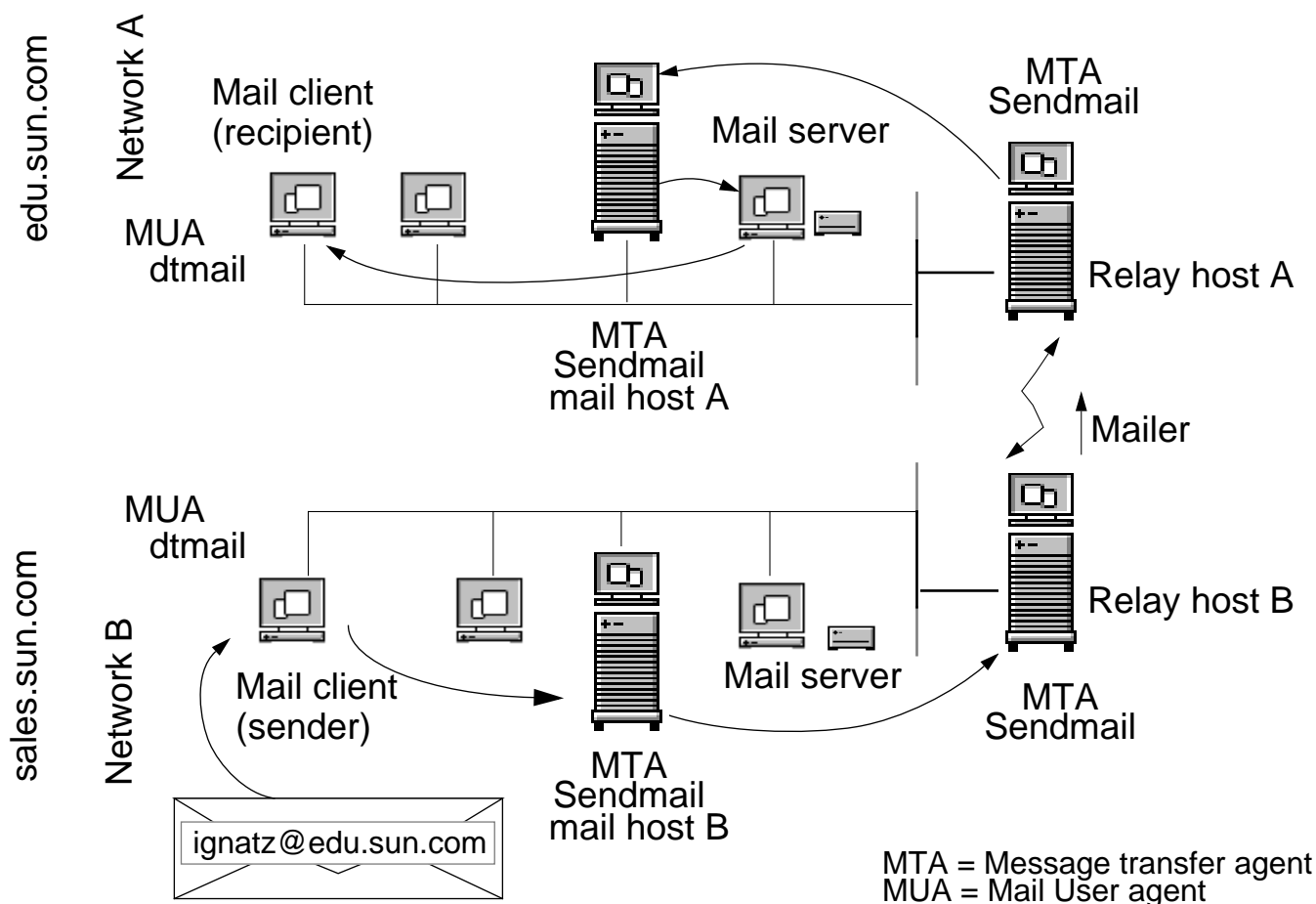


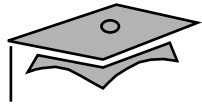
Concept of Mail Routing

- Sender/recipient
- Routing
 - Mail host
 - Relay host
 - Gateway
 - Mail server
 - Mail client



Electronic Mail Routing Diagram





Types of Mail Addresses

- Unqualified address

username

- Qualified address

username@machine

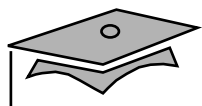
- Fully qualified address

username@subdomain2.subdomain1.top-level-domain

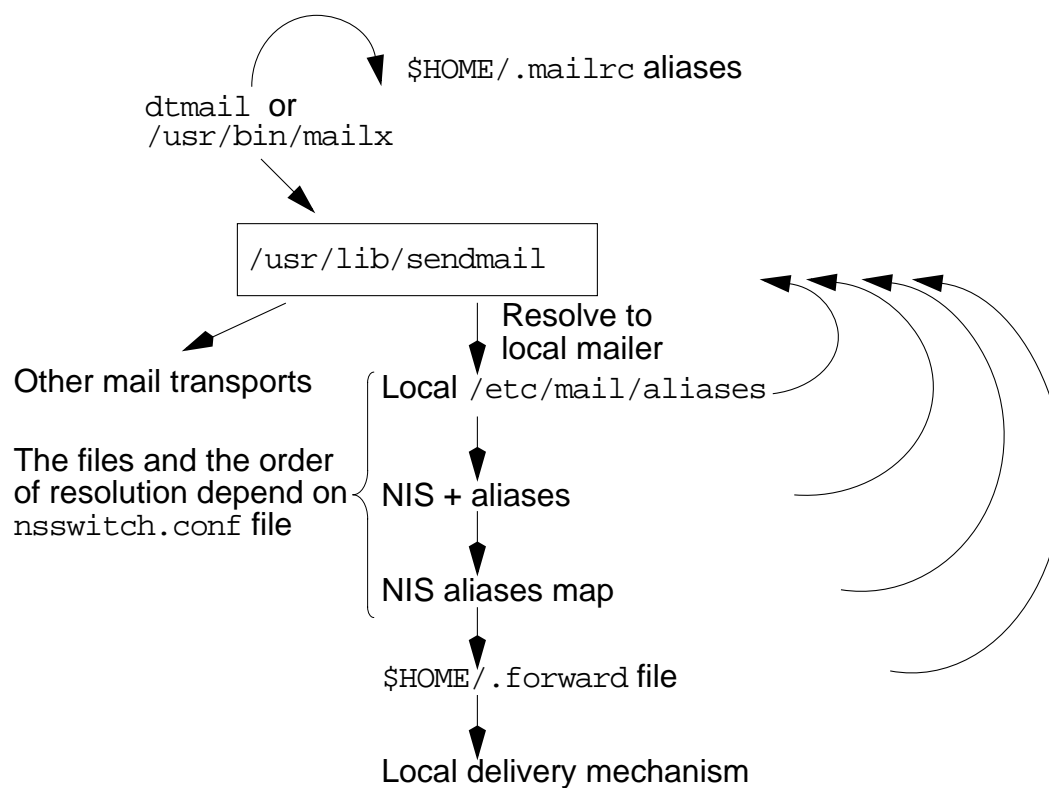


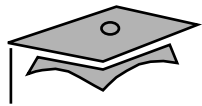
Elements of an Address

- Recipient
- Delimiter
- Destination address



Alias Resolution





Using Mail Aliases

- Alias resolution
 - `/etc/nsswitch.conf`
- Files
 - `$HOME/.mailrc` file
 - `/etc/mail/aliases` file
 - Network Information Services Plus (NIS+) aliases
 - Network Information Services (NIS) aliases map
 - `.forward` file



Using Mail Aliases – `.mailrc`

- Is used to customize a user's mail user agent
- Can contain local aliases
- Is located in the home directory of the user
- Uses `$HOME/.mailrc` syntax

```
alias managers hank@pyramid mary@egypt frank@mexico
```

```
alias group jane@cirrus bill@cs.berkeley.edu sue@lonestar
```

```
alias all managers group
```



Using Mail Aliases – aliases

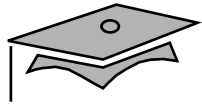
- System-wide alias file
- Available to all users on the system
- `/etc/mail/aliases` syntax

```
alias_name: user
```

```
alias_name: /file
```

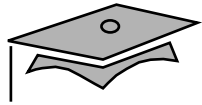
```
alias_name: |program
```

```
alias_name: :include: list
```



Sample /etc/mail/aliases File

```
# Following alias is required by the mail protocol, RFC 822
# Set to address of a HUMAN who deals with this system's mail problems.
Postmaster: dave
# Alias for mailer daemon; returned messages from our MAILER-DAEMON
# should be routed to our local Postmaster.
MAILER-DAEMON: postmaster
# Aliases to handle mail to programs or files, eg news or vacation
# decode: "|/usr/bin/uudecode"
nobody: /dev/null
# Alias for distribution list, members specified here:
staff:wnj,mosher,sam,ecc,mckusick,sklower,olson,rwh@ernie
# Alias for distribution list, members specified elsewhere:
keyboards: :include:/usr/jfarrell/keyboards.list
#####
# Local aliases below #
#####
sandy: sjp
fredphone: 6038523341@mobile.att.net
ann: ann@worldnet.att.net
```



Using Mail Aliases – `.forward`

- Same purpose as `/etc/mail/aliases`
- Located in user's home directory
- `$HOME/.forward` syntax

`user`

`/file`

`|program`

`\user, " |program"`



.forward Examples

```
\bob
```

```
/export/home/bob/mail.backup
```

```
\bob, " | /usr/bin/vacation bob || exit 75"
```

```
/export/home/bob/mail.backup
```



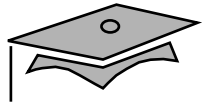
Setting Up the Postmaster

- `/etc/mail/aliases`

```
# Following alias is required by the mail protocol,  
# RFC 822
```

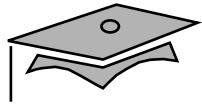
```
# Set to address of a HUMAN who deals with this  
# system's mail problems.
```

```
Postmaster: root
```



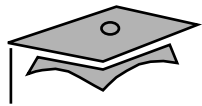
Determining Required Mail System Elements

- `sendmail.cf` configuration file
- Alias files
- Mailbox
- Postmaster alias

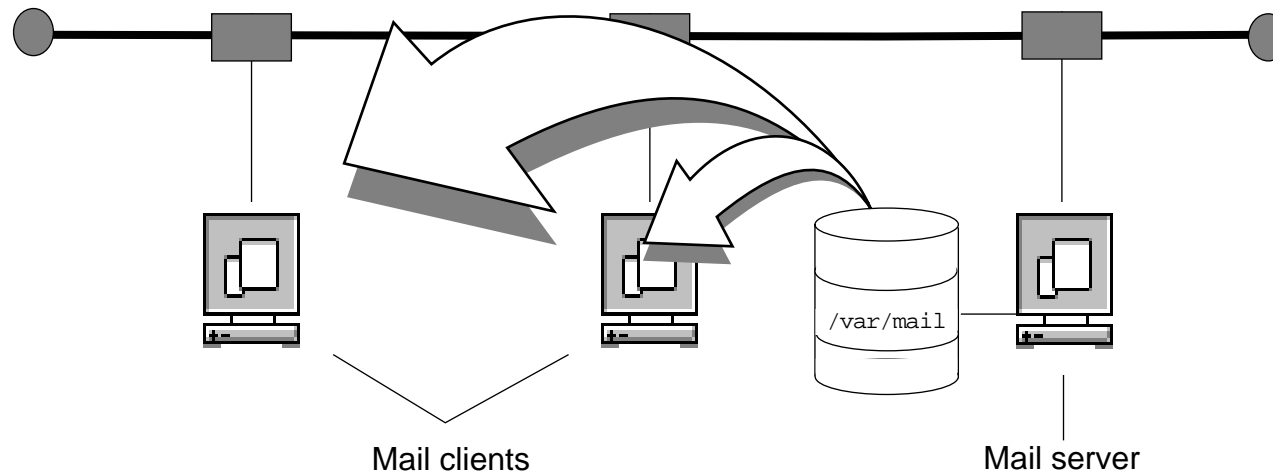


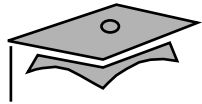
Configuring Local Mail Only

- `sendmail.cf` file on each mail client
- Designated server as the MTA
- `mailhost.domainname` in `hosts` file on mail host
- Local MTA IP address in `hosts` file of all mail clients
- Matching `aliases` files
- Storage space in `/var/mail` for each host



Configuring Local Mail in Remote Mode





Configuring Local Mail in Remote Mode

- `sendmail.cf` file on each mail client
- Designated server as the local MTA
- `mailhost.domainname` in `hosts` file on mail host
- Local MTA IP address in `hosts` file of all mail clients
- Matching aliases files
- Mail client's `/etc/vfstab` or `/etc/auto_direct` file
- Storage space in `/var/mail`



Internet Message Access Protocol

- Off-line access
- On-line access
- Disconnected access



Module 13

Introduction to sendmail



Overview

- Objectives
- Relevance



sendmail Overview

- History of sendmail
- Definition and features of sendmail
- Security issues with sendmail
- Functions of sendmail processing
- Changes to sendmail under the Solaris 7 OS
- Directory structure for sendmail
- Configuration files for sendmail
- The stopping and starting of sendmail



History of sendmail

- Originally written by Eric Allman at University of California, Berkley
- V8.7 and later written in conjunction with Internet Engineering Task Force (IETF)
- sendmail version 8.9.1 is distributed with Solaris 7



Features of sendmail

- It supports UNIX System V mail, UNIX Version 7 mail, and Internet mail.
- It uses existing software for delivery whenever possible.
- It can be configured to handle complex environments using configuration files.
- Groups can maintain their own mailing lists.
- Individual forwarding can be specified without modifying the domain-wide alias file.
- Each user can specify a custom mailer to process incoming mail.



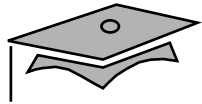
Security Issues With `sendmail`

- `sendmail` Version 8.9.1 is more secure than earlier versions.
- Due to its open exchange of arbitrary data, `sendmail` still has some security drawbacks.
- For information about `sendmail` security and other `sendmail` topics, refer to:
 - Costales, Brian. 1997. *sendmail*, Second Edition, O'Reilly.
 - Sun Microsystems web site: <http://www.sun.com>
 - The `sendmail` web site: <http://www.sendmail.org>



sendmail Processing

- Argument processing and address parsing
 - Scanning of the arguments
 - Processing of the option specifications
- Message collection
 - *Envelope, message header, and message body*
- Message delivery
- Queue for retransmission
- Return to sender



Changes to sendmail Under the Solaris 7 OS

- The sendmail program now has its own packages.
 - SUNWsndmr installs the configuration files.
 - SUNWsndmu installs the binaries.
- Configuration files are built using the m4 language.
- Strong anti-spam support is available.
- Permissions and the ownership have been changed to increase security.



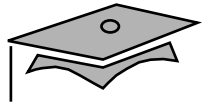
Changes to Sendmail Under Solaris 7

- There is increased security on `.forward` files.
- The `.forward` and `:include:` files cannot be group or world writable.
- `sendmail` reacts differently when an existing owner alias changes.
- The `sendmail` program requires a fully qualified host name when starting.
- Additional detailed information on the Solaris version of `sendmail` can be found at:
`http://www.sendmail.org/sun-specific/migration+sun.html`



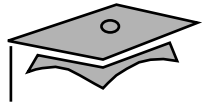
Directory Structure for `sendmail`

- The files and locations of the `sendmail` hierarchy include the:
 - Hierarchy of `/usr/lib/mail` directory
 - Featured files in `/usr/lib/mail`
 - Contents of the `/etc/mail` directory



Configuration Files for `sendmail`

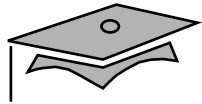
- `sendmail.cf` contains most of the `sendmail` configuration and provides the following functionality:
 - It defines the `sendmail` environment using symbols, classes, options, and parameters
 - It specifies how `sendmail` will rewrite addresses
 - It determines how addresses are to be interpreted
 - It determines how mail will be routed



Contents of the `sendmail.cf` File

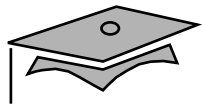
The `sendmail.cf` file consists of:

- Mail delivery agents – The programs used to deliver mail
- Macros – Built-in or user defined variables
- Options – Definitions of `sendmail` behavior
- Rule sets – A subroutine of rewrite rules
- Rewrite rules – Rules governing the transformation of addresses



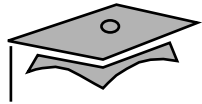
m4 Preprocessor

- Translates the simplified `sendmail` configuration language to the terse native configuration used in the file `sendmail.cf`
- Directories under `/usr/lib/mail` contain m4 macros, such as:
 - `VERSIONID(mail-v7sun.mc)`
 - `OSTYPE(solaris2.ml)`
 - `DOMAIN(solaris-generic)`
 - `FEATURE (relay_entire_domain)`
 - `MAILER([local,smtp])`



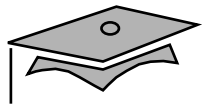
sendmail Configuration Files

- Three important files that are found in the `/etc` directory are needed to set up your email system:
 - `sendmail.cf` is the file used by `sendmail` for the configuration parameters.
 - `main.cf` is a template file used by the mail host, relay host, and gateway for mail addressing.
 - `subsidiary.cf` is a template file used on a machine that is not a mail host, a relay host, or a gateway that contains the default configuration.



Mail Exchange (MX) Records

- DNS supplies routing information for email through the use of a mail exchange (MX) record which is used:
 - When a site has a central mailhub
 - To ensure that mail addressed to a remote site can be relayed through the appropriate gateway
 - To provide a “backup” host to deliver mail to if the destination host is down
 - To specify hosts which will accept mail for a specific domain object
 - To ensure the lowest preference value is tried first and equally weighted entries are tried randomly



The Stopping and Starting of sendmail

- Automatic startup is performed using the `/etc/init.d/sendmail` script
- `sendmail` can be stopped and restarted using
`/etc/init.d/sendmail stop`
`/etc/init.d/sendmail start`
- Manual startup can be performed using the `sendmail` command lines located within the startup script to maintain proper startup values.



Miscellaneous sendmail Support

- Configuration file version values
- Additional mail service programs and files, such as:
 - `/usr/bin` directory used for mail services
 - `/usr/lib` mail files
 - Additional files used for mail services



Module 14

Common Mail Tasks



Overview

- Objectives
- Relevance



Overview of Mail

- Debugging `sendmail` with `mconnect`
- Using Simple Mail Transfer Protocol (SMTP)
- Testing the mail configuration
- Verifying a user
- Expanding an alias or list
- Setting up the postmaster alias
- Administering the mail configuration
- Defining SPAM
- Detecting masquerading hosts



Debugging sendmail With mconnect

There is more than one way to debug sendmail problems.

- Use `mailx -v`
- Issue SMTP commands
 - `mconnect`
 - `telnet`



Simple Mail Transfer Protocol

- Is used to send/receive a message
- Is a simple command set
- Follows a basic handshaking process



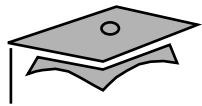
Testing the Mail Configuration

- Run tests after changing the configuration files
- Record common problems
- Gather additional diagnostic information



Verifying a User

- The VRFY command can be used to verify that the server knows about the user.
- It is used by network managers instead of SMTP clients.
- The VRFY command can be considered a security risk by some organizations.



Expanding an Alias or List

- EXPN command enables you to expand a user mail path.
- EXPN command enables you to expand a mailing list or an alias.
- EXPN command can be considered a security risk by some organizations.



Setting Up the Postmaster Alias

- Create one in each local `/etc/mail/aliases` file.
- Change `root` to the mail address of the person who will act as postmaster.
- Create a separate mailbox for the postmaster to keep postmaster mail separate from personal mail.



Administering the Mail Configuration

- Duties of the postmaster
 - Check the mail queues to be sure mail is flowing in and out.
 - Check any downed systems where mail is backing up. If the system is not needed, delete it from the mail service, or bring the system up to keep mail moving.
 - Fix personal aliases, as required.
 - Administer alias databases as people move in and out of the domain.
 - Set up temporary forwarding files.
 - Contact owners of mailing lists and help them fix mailing list problems.
 - Go through postmaster mail daily and look for problems, like broken `.forward` files and mail alias loops.
 - Answer questions outside the company.
 - Truncate log files periodically.



Administering the Mail Configuration

- Contact owners of mailing lists and help them fix mailing list problems.
- Go through postmaster mail daily looking for problems.
- Answer questions outside the company.
- Truncate log files periodically.



SPAM

- Defining
- Forging a mail header
- Deciphering email headers
- Using trace tools
- Disabling SPAM
- Stopping SPAM with `sendmail`



Host Masquerading

- Hide hostnames
 - `MASQUERADE_AS(host.domain)`
 - `MASQUERADE_DOMAIN(otherhost.domain)`
 - `FEATURE(masquerade_envelope)`
 - `EXPOSED_USER(usernames)`
- Enable host masquerading



Module 15

LAN Planning



Overview

- Objectives
- Relevance



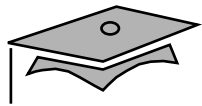
Planning Considerations

- Relationship of the LAN to the organization's goals
- Generic function
- Industry standards
- Design specifications
- Analysis of data gathered
- Mission-critical requirements

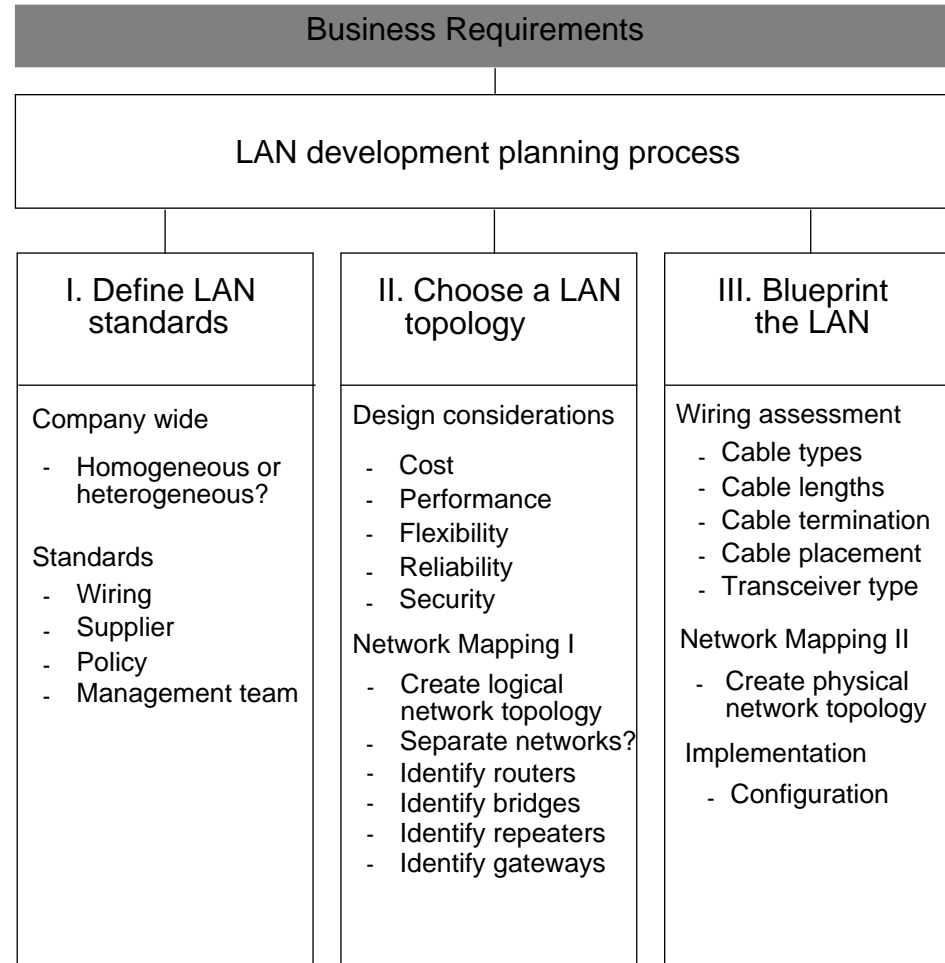


Determining Business Requirements

- Types of applications
- Network loads
- Shared resources
- Future growth
- Required network hardware



LAN Development Planning Process





Defining LAN Standards

- Homogeneous or heterogeneous
- Network media
- Suppliers
- Policy
- Management team



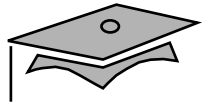
Choosing a LAN Topology

- Cost
- Performance
- Flexibility
- Reliability
- Security



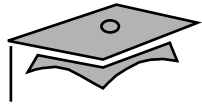
Network Media Costs

Media Type	Relative Cost
10BASE5	Thick Ethernet is expensive, though it may be cheaper to install in existing structures than 10BASET.
10BASE2	Thin Ethernet is less expensive than 10BASE5, though it is expensive to place in existing walled structures. It should be used for open laboratory environments. It is also useful to tie into existing network interfaces that require 10BASE2, such as PC-based systems.
10BASET	Twisted-pair may already be present in an office building and may be the least expensive way to provide a 10BASET network. Data-grade twisted-pair should be used for all new structures.
10BASEF	Fiber-optic Ethernet is very expensive to install and maintain.
Category 5 (100BASET)	Upgrade to 10BASET network. Data-grade twisted-pair should be used for all new structures. Average cost is 30–40 percent higher than 10BASET mainly due to rigorous testing during installation.



Network Media Flexibility

Media Type	Relative Flexibility
10BASE5	Thick Ethernet is difficult to install, though it may be appropriate for existing structures.
10BASE2	Thin Ethernet is easier to handle than 10BASE2. It has less distance capabilities than 10BASE2 and 10BASET and may not be appropriate for LANs spanning many rooms.
10BASET	Twisted-pair is easy to install and allows for flexibility in network design since it is much easier to route around ceilings and into offices than 10BASE5 or 10BASE2.
10BASEF	Fiber-optic Ethernet is difficult to install and maintain.
Category 5 (100BASET)	Like 10BASET, Category 5 twisted-pair is easy to install and allows for flexibility in network design since it is much easier to route around ceilings and into offices than 10BASE5 or 10BASE2.



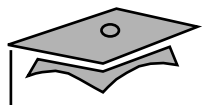
Reliability

Cable Type	Ease of Troubleshooting
10BASE5	Thick Ethernet is difficult to troubleshoot.
10BASE2	Thin Ethernet wiring can be accidentally disconnected and damaged quite easily. Like 10BASE5, it is fairly difficult to troubleshoot.
10BASET	Some manufacturers offer intelligent 10BASET hubs that support the Simple Network Management Protocol (SNMP). These devices offer powerful network management capabilities, such as the ability to remotely shut down a port of an offending host.
10BASEF	Once installed, fiber-optic Ethernet is fairly easy to troubleshoot.
Category 5 (100BASET)	Category 5 initially requires much more testing during the installation phase. Once the media has passed testing, it is considered very reliable. Intelligent Category 5 hubs that support the Simple Network Management Protocol (SNMP) are standard.

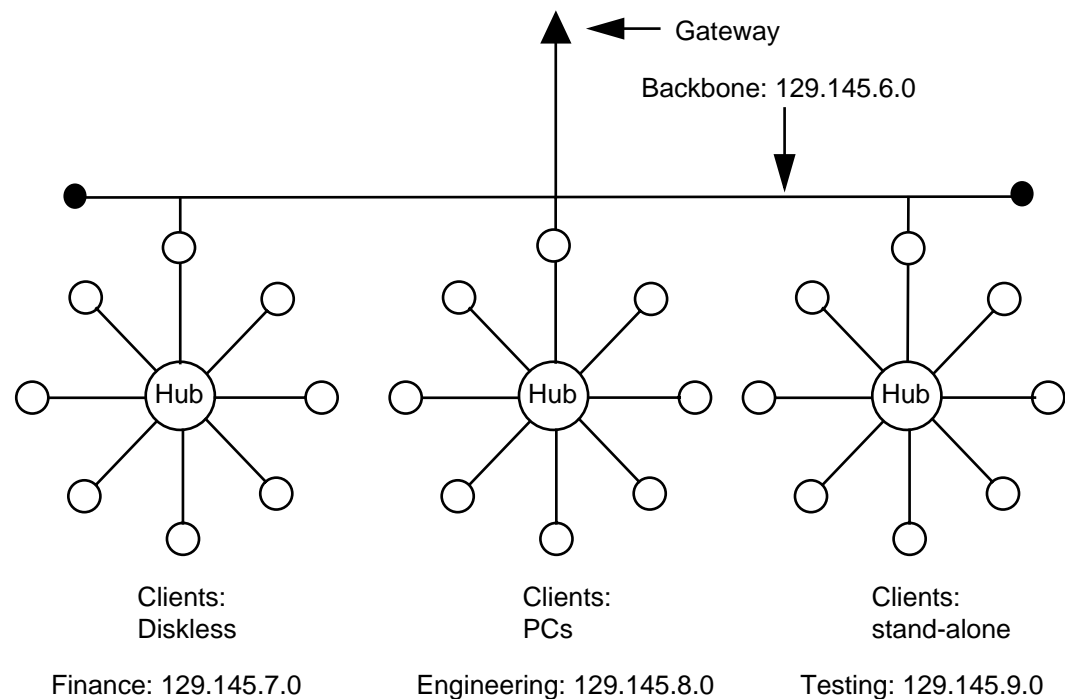


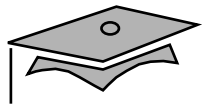
LAN Topology

- Network mapping
- Hierarchy
- Network segmentation
 - Performance
 - Security
 - Management
 - Flexibility

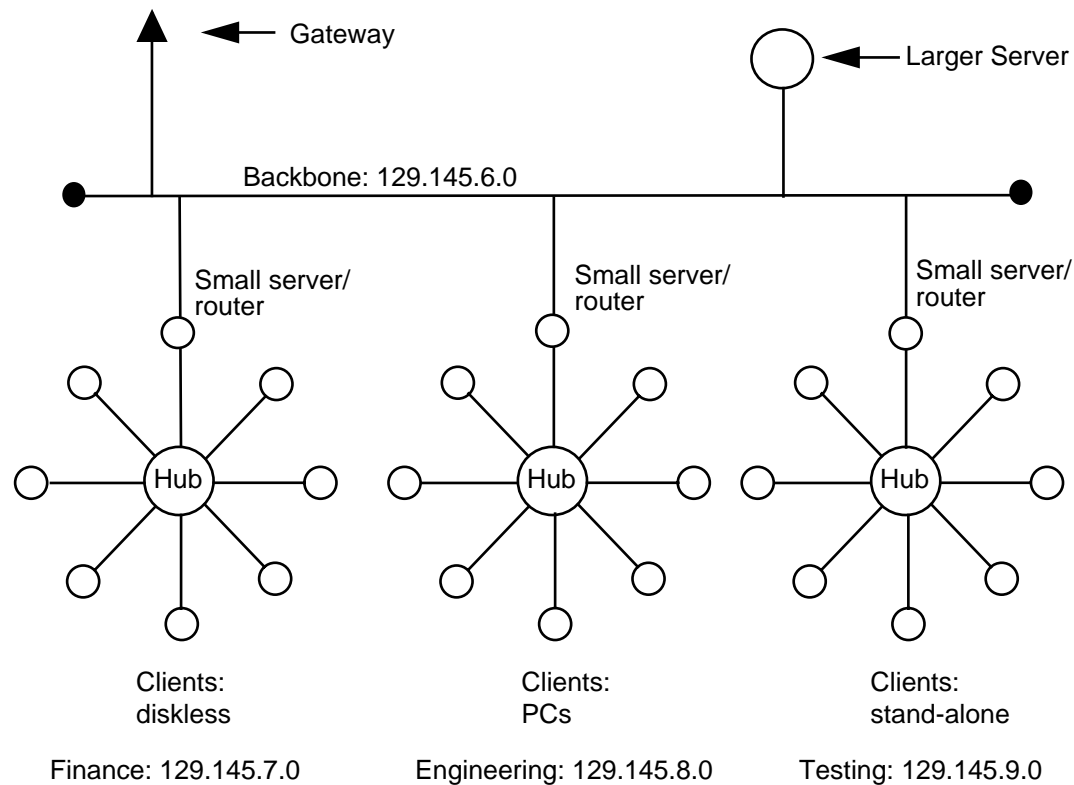


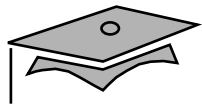
Network Backbone Layout



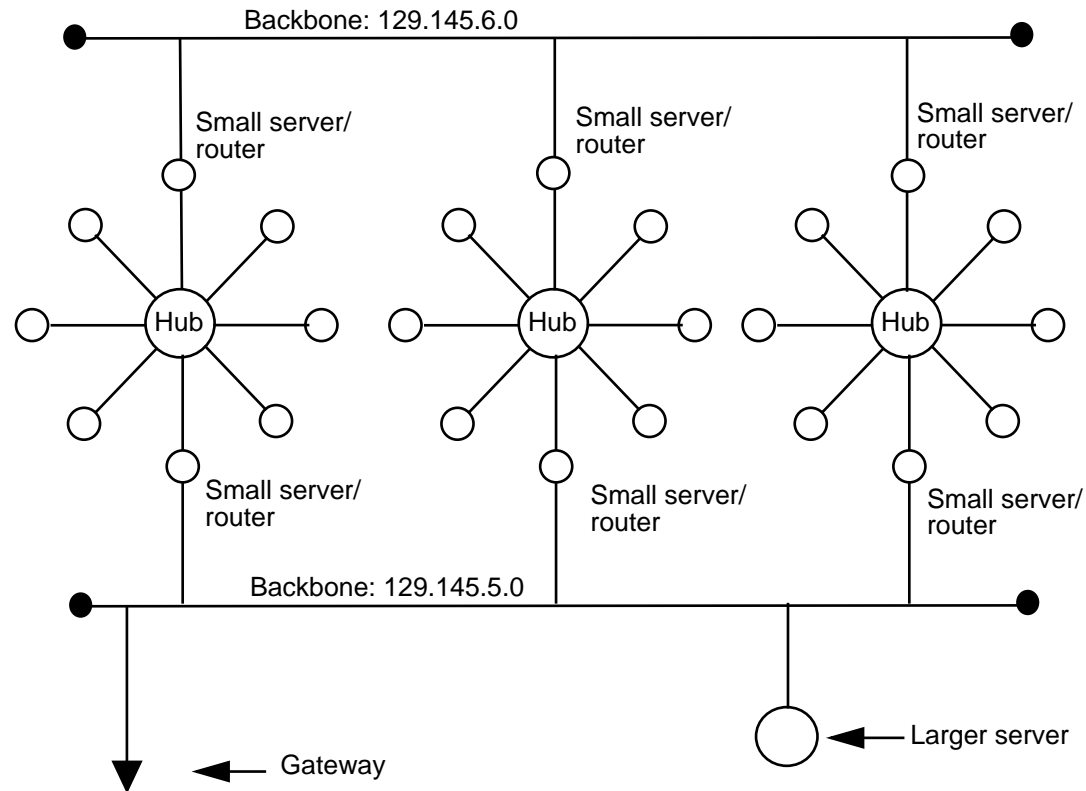


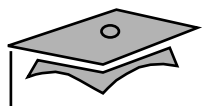
Two-Tiered Network Topology



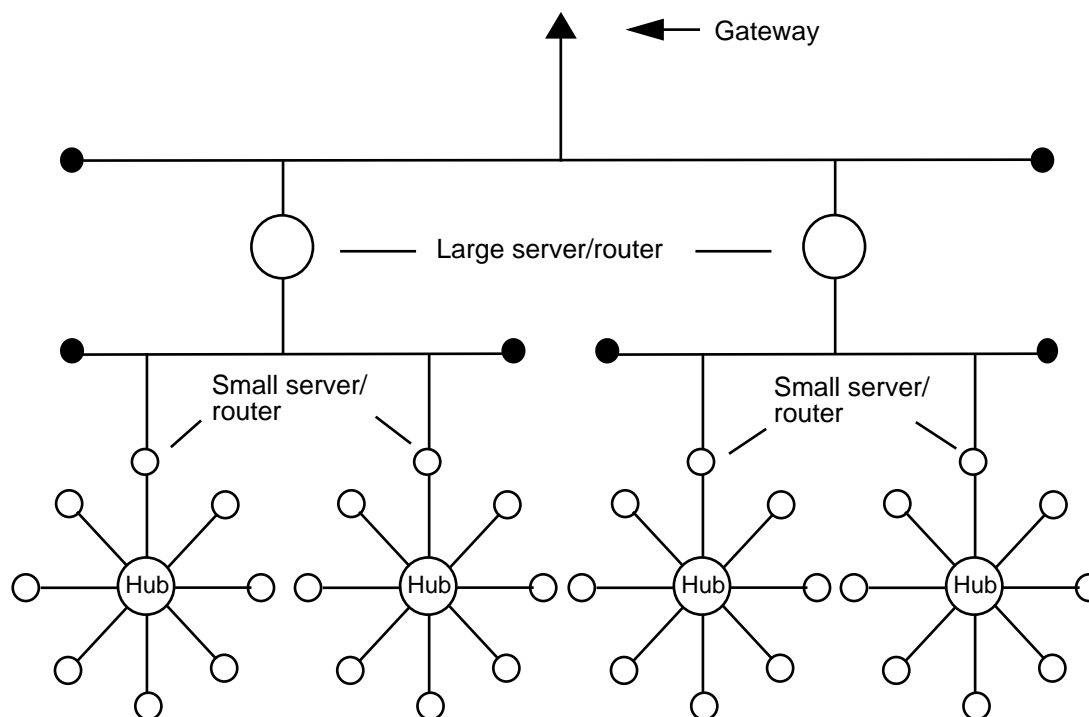


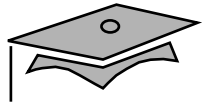
Dual-Backbone Network Topology





Two-Tiered Network Flexibility





Blueprinting a LAN

- Is similar to LAN topology but it is more detailed
- Use planning tool to assess placement of LAN cable/ components



Module 16

Network Troubleshooting



Overview

- Objectives
- Relevance



Troubleshooting

- General troubleshooting guidelines
 - Define problem in your own words
 - Locate lowest level of failure
 - Take nothing for granted
 - Back up, document, and test
 - Make permanent changes



Troubleshooting Tools

- ping
 - Use ICMP echo
 - Use `ping -s`
 - Broadcast ping (255)



Troubleshooting Tools

- `ifconfig`
 - Display status of interface
 - Use two versions
 - Use `plumb`



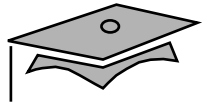
Troubleshooting Tools

- arp
 - Trace duplicate IP addresses
 - Determine manufacturer of Ethernet card
 - Check arp table



Troubleshooting Tools

- snoop
 - Use for remote troubleshooting
 - Write to file
 - Use three modes
 - View specific packets



Troubleshooting Tools

- ndd
 - Be very careful
 - Perform routing/IP forwarding
 - Check interface speed
 - Check interface mode



Troubleshooting Tools

- netstat
 - View routing tables (-r)
 - Display IP addresses instead of host names (-n)
 - Use verbose mode (-v)



Troubleshooting Tools

- traceroute
 - Route network traffic
 - Acquire benchmark
 - Use ttl and ICMP
 - Display IP addresses (-n)



Common Network Problems

- Cabling
- mdi
- Encryption
- Security, blocked ports
- Routing
- Interfaces not plumbed
- Bad name service data



Connectivity Problems

- Logical line of questioning
- Global or isolated problem
- Changes
- What connectivity, if any, exists
- snoop uses



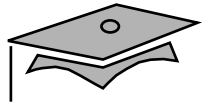
Troubleshooting Techniques

- Work up or down through the TCP/IP model layers
 - Application layer
 - Transport layer and Internet layer
 - Network Interface layer
 - Physical layer



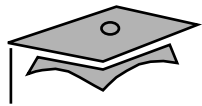
Troubleshooting Scenarios

- Use multi-homed system which acts as a core router
- Use traceroute
- Create `/etc/notrouter`

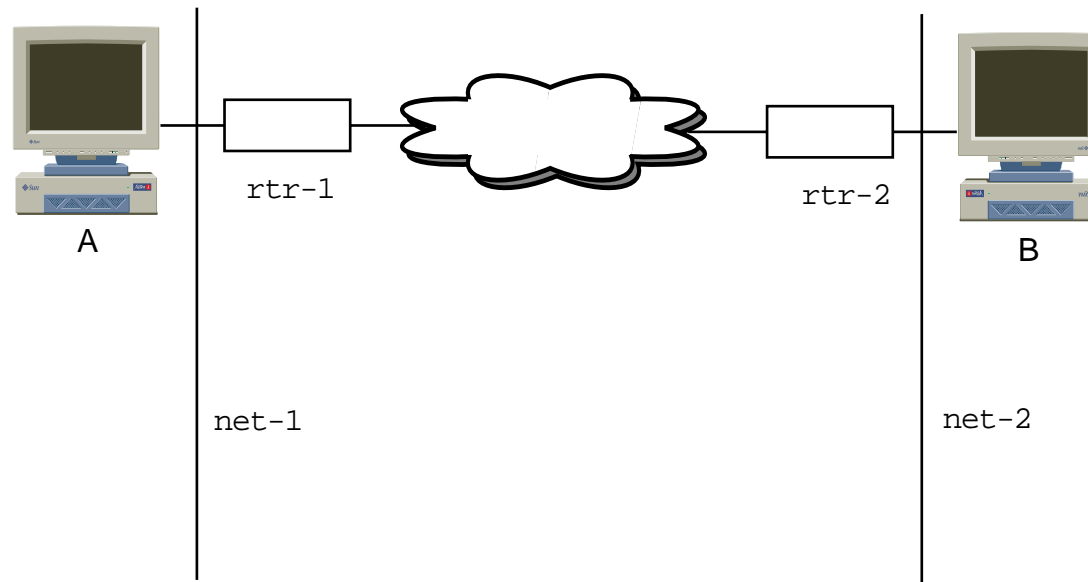


Troubleshooting Scenarios

- Faulty cable
- Router log files
- Replace cable



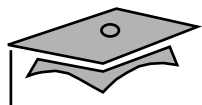
Faulty Cable Diagram



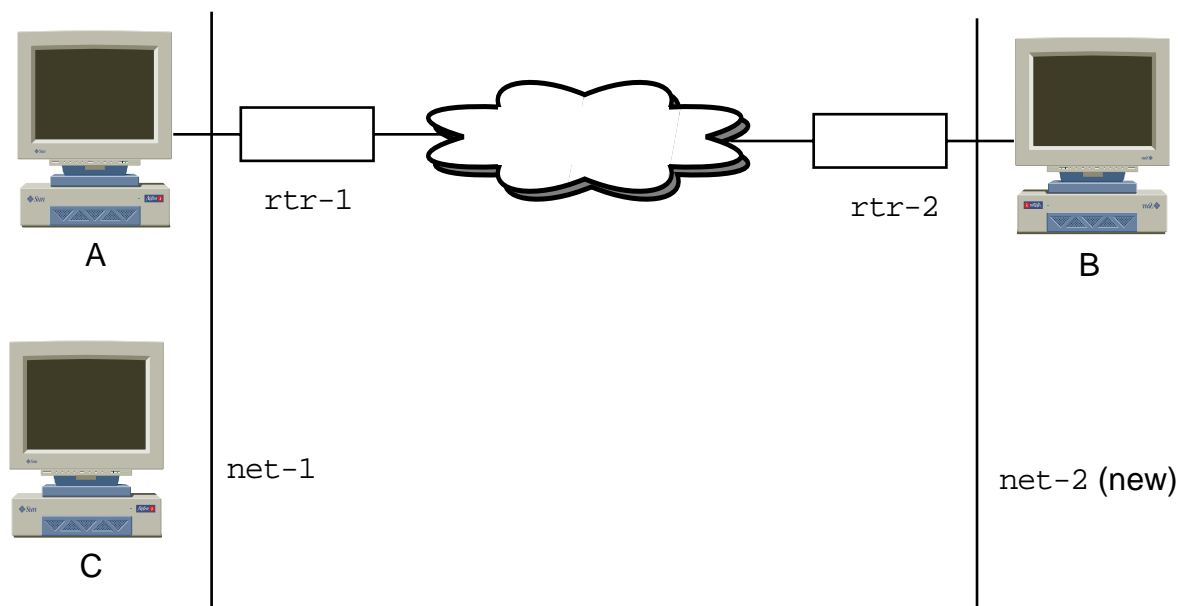


Troubleshooting Scenarios

- Duplicate IP address
- ping failed
- traceroute failed
- arp cache incomplete
- Reconfigured IP address

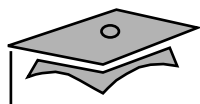


Duplicate IP Address

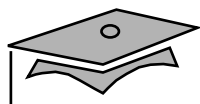


Course Contents

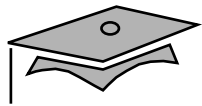
About This Course	About This Courses-1
Course Goal	About This Courses-2
Course Overview	About This Courses-3
Course Map	About This Courses-4
Module Overview	About This Courses-5
Module Pacing	About This Courses-8
Topics Not Covered	About This Courses-9
How Prepared Are You?	About This Courses-10
Introductions	About This Courses-11
How to Use Course Materials	About This Courses-12
Network Models	1-1
Overview	1-2
Network Models	1-3
ISO/OSI 7 Layer Model	1-4
Data Exchange Between Application Processes	1-5
Physical Layer	1-6
Data Link Layer	1-7
Network Layer	1-8
Transport Layer	1-9
Session Layer	1-10
Presentation Layer	1-11
Application Layer	1-12
TCP/IP	1-13
TCP/IP Network Model	1-14
TCP/IP Layers	1-15



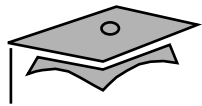
Hardware Layers	1-16
Network Interface Layer	1-17
Internet Layer	1-18
Transport Layer	1-19
Application Layer	1-20
Peer-to-Peer Communication	1-21
TCP/IP Protocol Stack	1-22
Introduction to Local Area Networks	2-1
Overview	2-2
Introduction to Local Area Network	2-3
LAN Topology	2-4
Bus Configuration	2-5
Star Configuration	2-6
Ring Configuration	2-7
LAN Components	2-8
Ethernet Components	2-9
Sun Communications Controller	2-10
LAN Methodologies	2-11
Network Media	2-12
Multimode Ethernet	2-13
Lab Configuration	2-14
Ethernet Interface	3-1
Overview	3-2
Introduction to Ethernet	3-3
Ethernet TCP/IP Layers	3-4
Ethernet Major Elements	3-5
Access Method	3-6
CSMA/CD Flowchart	3-7



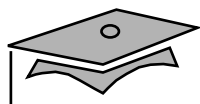
Switched Ethernet	3-8
Switched Ethernet Diagram	3-9
Ethernet Address	3-10
Sending Messages	3-11
Ethernet Frame	3-12
Ethernet Version 2 Frame Fields	3-13
TCP/IP Layer Encapsulation	3-14
Ethernet Maximum Transfer Unit	3-15
Ethernet Error Checking	3-16
Useful Troubleshooting Commands	3-17
snoop	3-18
snoop -v	3-19
snoop -V	3-20
netstat -i	3-21
ifconfig	3-22
ARP and RARP	4-1
Overview	4-2
Introduction to Address Resolution	4-3
Address Resolution TCP/IP Layers	4-4
Why ARP Is Required	4-5
Address Resolution Protocol	4-7
ARP Request	4-8
ARP Reply	4-10
ARP Table Management	4-12
ARP Command Examples	4-13
Reverse Address Resolution	4-14
RARP Request	4-15
RARP Reply	4-17
Troubleshooting the in.rarpd Server	4-19



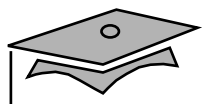
<i>Internet Layer</i>	5-1
Overview	5-2
Introduction to Internet	5-3
TCP/IP Layered Model	5-4
Internet Layer	5-5
Classful IPv4 Addressing	5-6
Class A Address Format	5-7
Class B Address Format	5-8
Class C Address Format	5-9
Class D Address Format	5-10
Special IPv4 Addresses	5-11
IPv4 Netmasks	5-12
Computing Network Number	5-13
Reasons to Subnet	5-14
Defining Subnets	5-15
Subnet Mask	5-16
Computation of Extended Network Number	5-17
Non-Byte Bounded Subnet Masks	5-18
Computing the Broadcast Address	5-19
Variable Length Subnet Masks (VLSM)	5-20
Class B Subnet Mask Yield	5-21
Class A Network Using VLSM	5-22
Permanent Subnet Masks	5-23
Recommended Subnet Masks	5-24
Class B Subnet Masks	5-25
Class C Subnet Masks	5-26
Configuring a Subnet	5-27
Network Interface Configuration	5-28
/sbin/ifconfig Command	5-29
Examining Network Interfaces	5-30



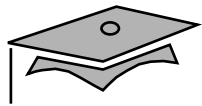
Enable and Disable Interface Examples	5-31
Close and Open Interface Examples	5-32
Set IP Address, Enable Interface, and Disable Trailers	5-33
Change Netmask and Broadcast Value	5-34
Troubleshooting the Network Interface	5-35
Routing	6-1
Overview	6-2
Introduction to Routing	6-3
Internet TCP/IP Layer	6-4
Routing Schemes	6-5
Routing Algorithm	6-6
Kernel Routing Process	6-7
Autonomous System (AS)	6-8
Gateway Protocols	6-9
Exterior Gateway Protocol	6-10
Border Gateway Protocol	6-11
Interior Gateway Protocol	6-12
Open Shortest Path First (OSPF)	6-13
Routing Information Protocol	6-14
Least Cost Path	6-15
Stability Features	6-16
/usr/sbin/in.routed	6-17
Network Router Discovery (RDISC)	6-18
/usr/sbin/in.rdisc	6-19
Multihomed Host	6-20
Routing Initialization	6-21
Displaying the Routing Table	6-22
/etc/inet/networks File	6-23
Manually Manipulating Routing Table	6-24



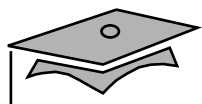
/etc/gateways File	6-25
Router Configuration	6-26
Troubleshooting Router Configuration	6-27
Lab Network Configuration	6-28
Transport Layer	7-1
Overview	7-2
Introduction to the Transport Layer	7-3
TCP/IP Layered Model	7-4
Types of Protocols	7-5
Stateful Versus Stateless	7-6
Reliable Protocols	7-7
Unreliable Protocols	7-8
Transport Protocols	7-9
Transport Layer Protocol Features	7-10
User Datagram Protocol	7-11
Transmission Control Protocol	7-12
TCP Flow Control	7-13
Client-Server Model	8-1
Overview	8-2
The Client-Server Model	8-3
ONC+ Technologies	8-5
Port Numbers	8-8
/etc/inet/services Extract	8-9
How a Server Process Is Started	8-10
Remote Procedure Call	8-11
Status Commands	8-12
/usr/bin/rpcinfo -p	8-13
/usr/bin/rpcinfo -b	8-14



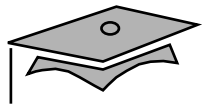
/usr/bin/rpcinfo -u	8-15
/usr/bin/netstat -a	8-16
DHCP	9-1
Overview	9-2
Dynamic Host Configuration Protocol	9-3
How DHCP Uses BOOTP	9-4
DHCP FEATURES	9-5
DHCP Client/Server	9-6
Server Side	9-7
SERVER DATABASES	9-8
<i>dhcp_network</i> ENTRY FORMAT	9-9
<i>dhcp_network</i> Examples	9-10
dhcptab Entry Format	9-11
Symbols and Macros	9-12
Symbol Characteristics	9-13
Macro Definitions	9-14
Lease Time Policy	9-15
Lease Flags (<i>dhcp_network</i>)	9-16
dhcptab Examples	9-17
DHCP ADMINISTRATION COMMANDS	9-18
DHCP SERVER CONFIGURATION	9-19
CONFIGURING DHCP ON THE SERVER	9-20
CONFIGURING DHCP ON THE CLIENT	9-21
Troubleshooting DHCP	9-22
DHCP Lab Network Configuration	9-23
<i>Introduction to Network</i>	
<i>Management Tools</i>	10-1
Overview	10-2



Network Management	10-3
Simple Network	
Management Protocol	10-4
SNMP-based Management Applications	10-7
Solstice Site Manager	10-8
Solstice Domain Manager	10-9
Solstice Enterprise Manager	10-10
Domain Name System	11-1
Overview	11-2
Why DNS – A Brief History	11-3
DNS Namespace – Domains	11-4
DNS Namespace – Structure	11-5
DNS Namespace – Domain Naming	11-6
Zones of Authority	11-7
DNS Servers	11-8
DNS Answers	11-9
Client Resolver	11-10
Resolution Process	11-11
BIND	11-12
DNS Server Configuration	11-13
named.conf - BIND Configuration File	11-14
/etc/named.conf Statement Definitions	11-15
DNS Resource Records	11-16
Resource Record Types	11-17
/var/named/named.root File	11-18
named.root File Excerpt	11-19
domain-info File	11-20
inverse-domain-info File	11-22
loopback-domain-info File	11-24



/etc/nsswitch.conf	11-25
/etc/resolv.conf	11-26
nslookup	11-27
nslookup Examples	11-28
BIND Debugging Tools	11-29
Secondary DNS Server Setup	11-30
named.conf File – Secondary Server	11-31
DNS Security	11-32
Miscellaneous DNS Topics	11-33
DNS Resources	11-34
DNS Lab Layout	11-35
Email Fundamentals	12-1
Overview	12-2
Introduction to Electronic Mail	12-3
Concept of Mail Routing	12-4
Electronic Mail Routing Diagram	12-5
Types of Mail Addresses	12-6
Elements of an Address	12-7
Alias Resolution	12-8
Using Mail Aliases	12-9
Using Mail Aliases – .mailrc	12-10
Using Mail Aliases – aliases	12-11
Sample /etc/mail/aliases File	12-12
Using Mail Aliases – .forward	12-13
.forward Examples	12-14
Setting Up the Postmaster	12-15
Determining Required Mail System Elements	12-16
Configuring Local Mail Only	12-17
Configuring Local Mail in Remote Mode	12-18



Internet Message Access Protocol	12-20
Introduction to sendmail	13-1
Overview	13-2
sendmail Overview	13-3
History of sendmail	13-4
Features of sendmail	13-5
Security Issues With sendmail	13-6
sendmail Processing	13-7
Changes to sendmail Under the Solaris 7 OS	13-8
Changes to Sendmail Under Solaris 7	13-9
Directory Structure for sendmail	13-10
Configuration Files for sendmail	13-11
Contents of the sendmail.cf File	13-12
m4 Preprocessor	13-13
sendmail Configuration Files	13-14
Mail Exchange (MX) Records	13-15
The Stopping and Starting of sendmail	13-16
Miscellaneous sendmail Support	13-17
Common Mail Tasks	14-1
Overview	14-2
Overview of Mail	14-3
Debugging sendmail With mconnect	14-4
Simple Mail Transfer Protocol	14-5
Testing the Mail Configuration	14-6
Verifying a User	14-7
Expanding an Alias or List	14-8
Setting Up the Postmaster Alias	14-9
Administering the Mail Configuration	14-10



SPAM	14-12
Host Masquerading	14-13
LAN Planning	15-1
Overview	15-2
Planning Considerations	15-3
Determining Business Requirements	15-4
LAN Development Planning Process	15-5
Defining LAN Standards	15-6
Choosing a LAN Topology	15-7
Network Media Costs	15-8
Network Media Flexibility	15-9
Reliability	15-10
LAN Topology	15-11
Network Backbone Layout	15-12
Two-Tiered Network Topology	15-13
Dual-Backbone Network Topology	15-14
Two-Tiered Network Flexibility	15-15
Blueprinting a LAN	15-16
Network Troubleshooting	16-1
Overview	16-2
Troubleshooting	16-3
Troubleshooting Tools	16-4
Common Network Problems	16-11
Connectivity Problems	16-12
Troubleshooting Techniques	16-13
Troubleshooting Scenarios	16-14
Duplicate IP Address	16-18

Copyright 2000 Sun Microsystems Inc., 901 San Antonio Road, Palo Alto, California 94303, Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley 4.3 BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company Ltd.

Sun, Sun Microsystems, le logo Sun, Solaris, SunATM, Sun Quad FastEthernet, SunFastEthernet, SunFDDI, SunTRI, Solstice AdminSuite, SunNet Manager, OpenWindows, et SunSoft. Solstice Enterprise Agents, NFS, JumpStart, Sun OS, ONC+, Solstice Site Manager, Solstice Domain Manager, Solstice Enterprise Manager, Solstice Enterprise Agents, et Solstice Internet Mail Server. sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays.

Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. IBM[®], DECnet[®], AppleTalk[®], et Novell[®].

UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

L'interface d'utilisation graphique OPEN LOOK et Sun[™] a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

L'accord du gouvernement américain est requis avant l'exportation du produit.

Le système X Window est un produit de X Consortium, Inc.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.