Solaris™ 8 Operating Environment– TCP/IP Network Administration

SA-389

About This Course

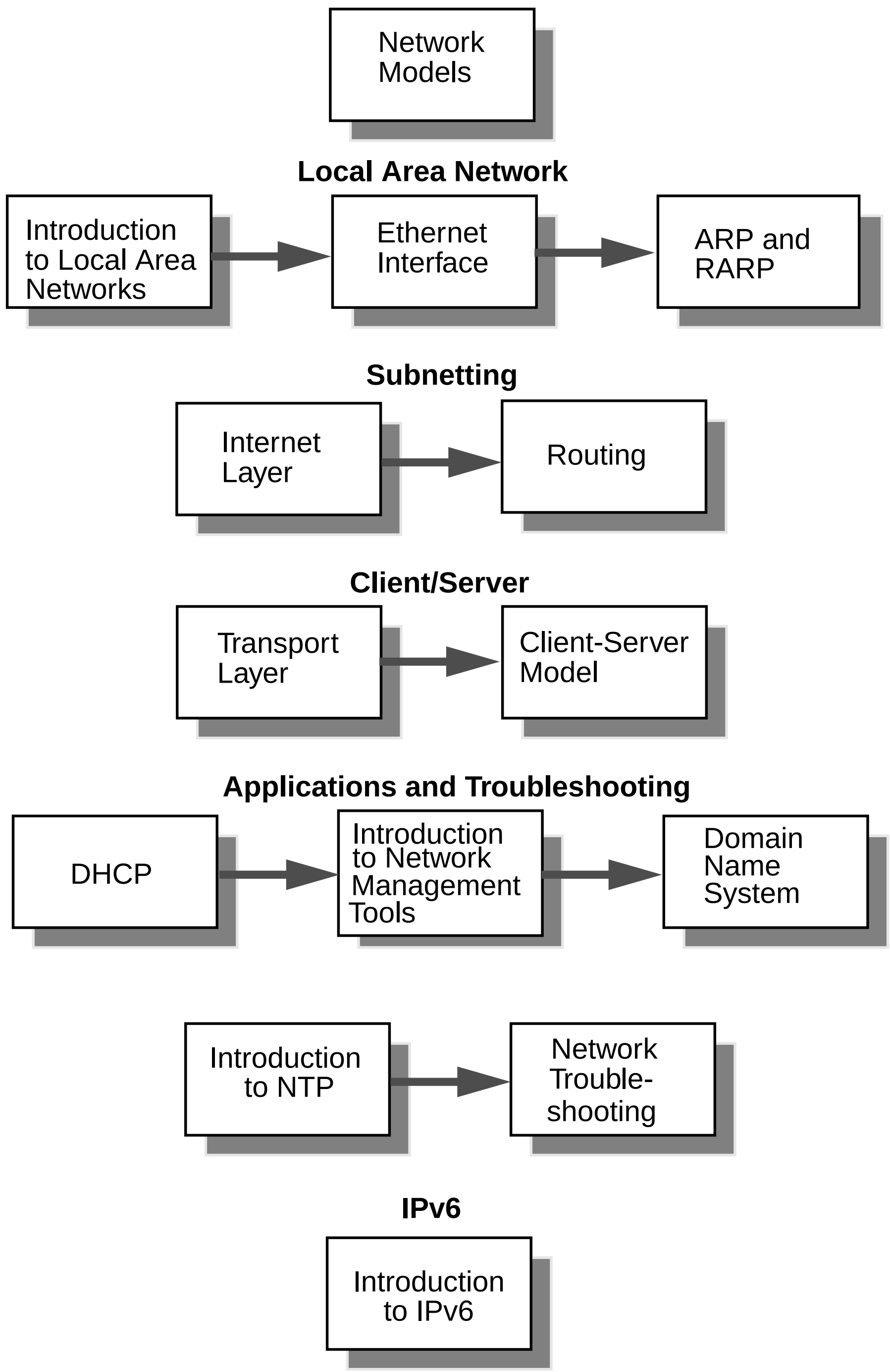
Course Goal

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 troubleshooting
- Topics include:
  - Dynamic Host Configuration Protocol (DHCP)
  - Domain Name Service (DNS)
  - Network Time Protocol (NTP)
  - IPv6

Course Map


Network Models
Local Area Network
Introduction to Local Area Networks
Ethernet Interface
ARP and RARP
Subnetting
Internet Layer
Routing
Client/Server
Transport Layer
Client-Server Model
Applications and Troubleshooting
DHCP
Introduction to Network Management Tools
Domain Name System
Introduction to NTP
Network Trouble-shooting
IPv6
Introduction to IPv6

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 – “Introduction to NTP”

Module Overview

- Module 13 – “Network Troubleshooting”
- Module 14 – “Introduction to IPv6”

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.	
"Introduction to NTP"				P.M.	
"Network Troubleshooting"					A.M.
"Introduction to IPv6"					P.M.

Topics Not Covered

- Solaris™ Operating Environment system administration
- Server storage administration
- NIS+
- Solaris Operating Environment 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

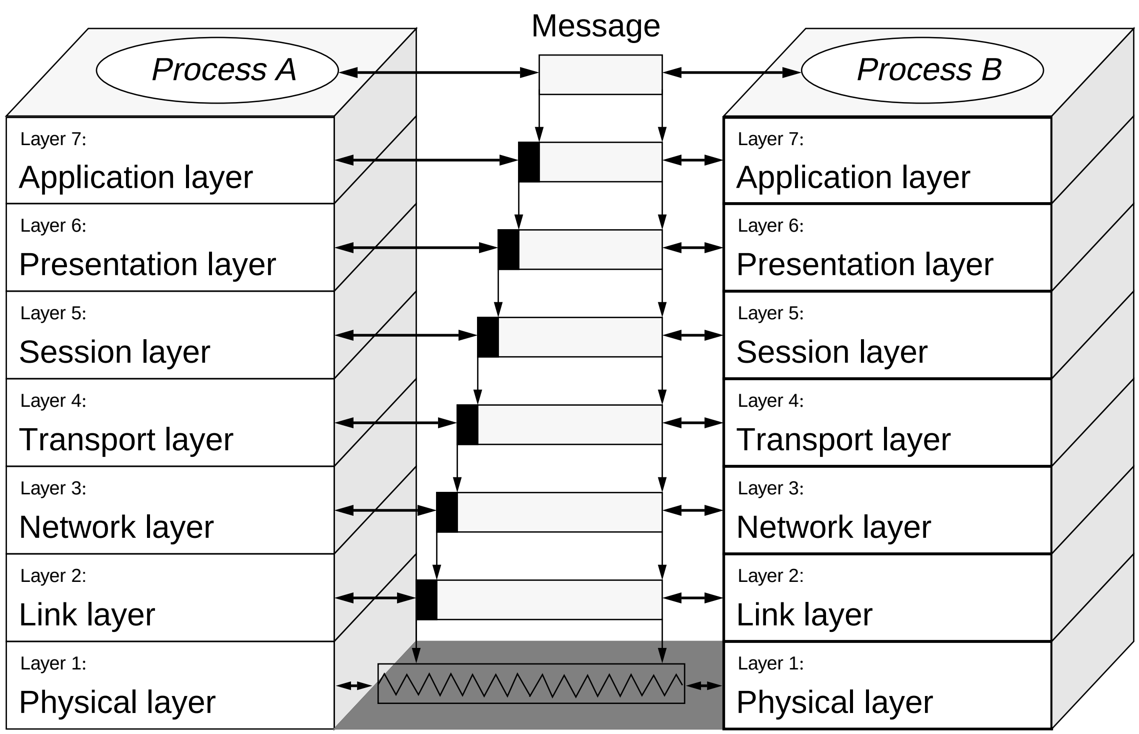
Network Models

- International Organization for Standardization/Open Systems Interconnection (ISO/OSI) reference model
- Transmission Control Protocol/Internet Protocol (TCP/IP) suite (TCP/IP model or TCP/IP)

ISO/OSI Seven-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:
  - Connectionless-mode/connection-mode (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 External Data Representation (XDR)

Application Layer

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

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

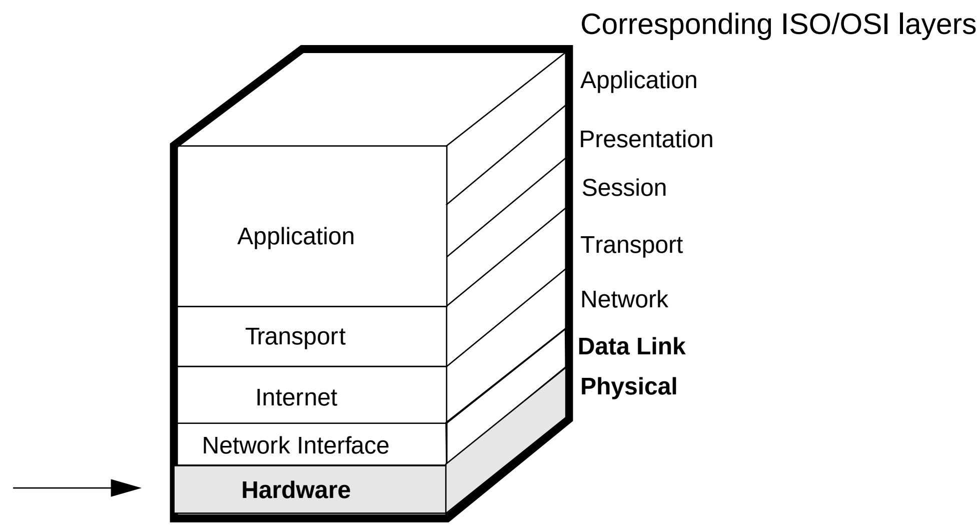
TCP/IP Network Model

- 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.

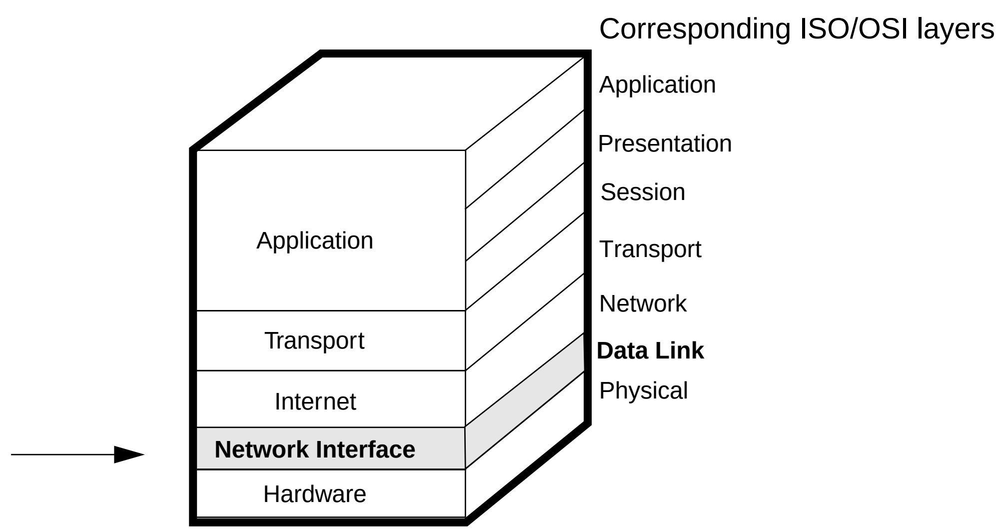
TCP/IP Layers

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

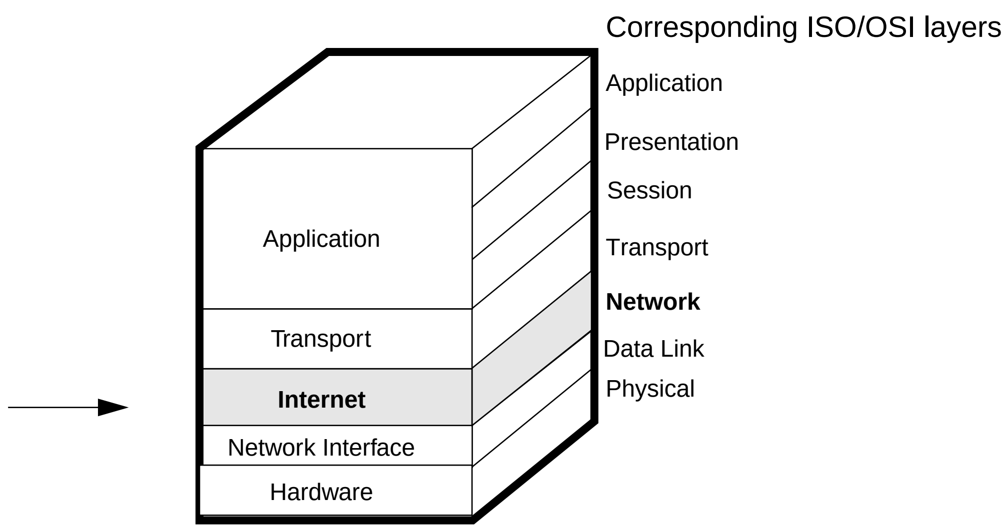
Hardware Layers



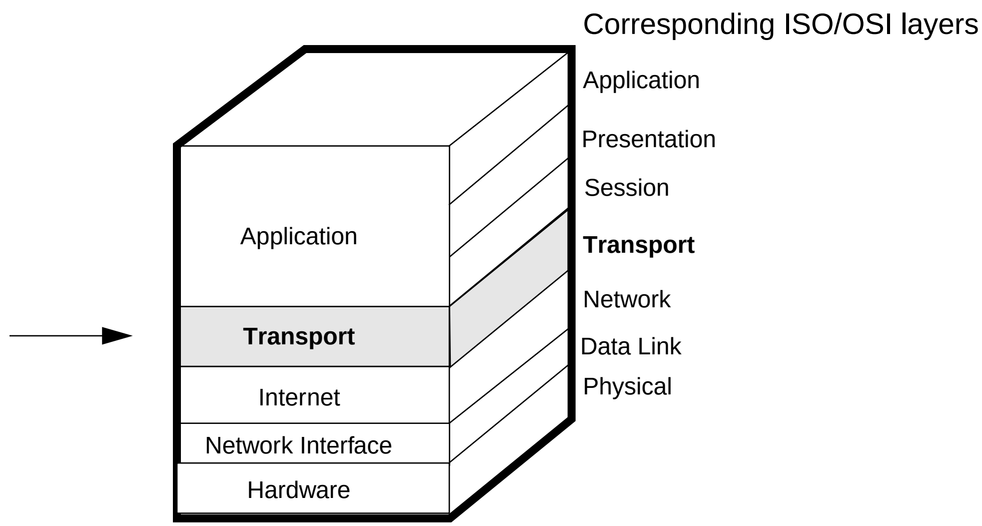
Network Interface Layer



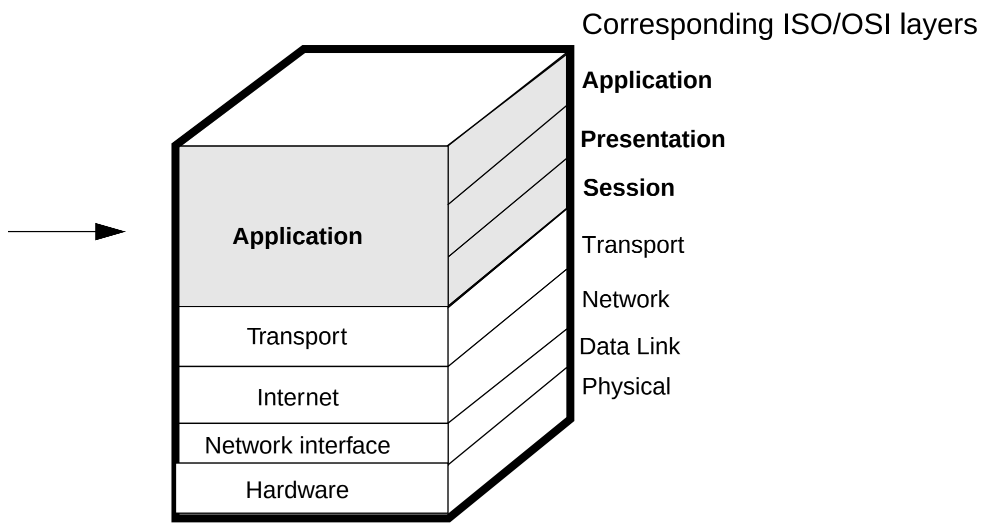
Internet Layer



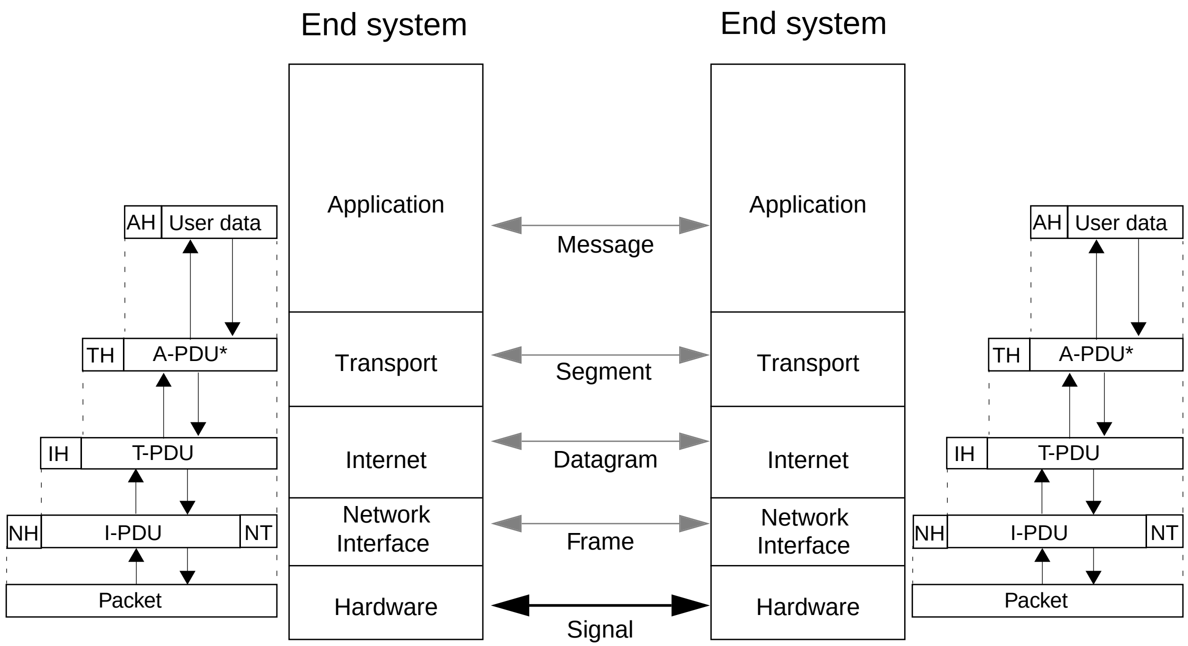
Transport Layer



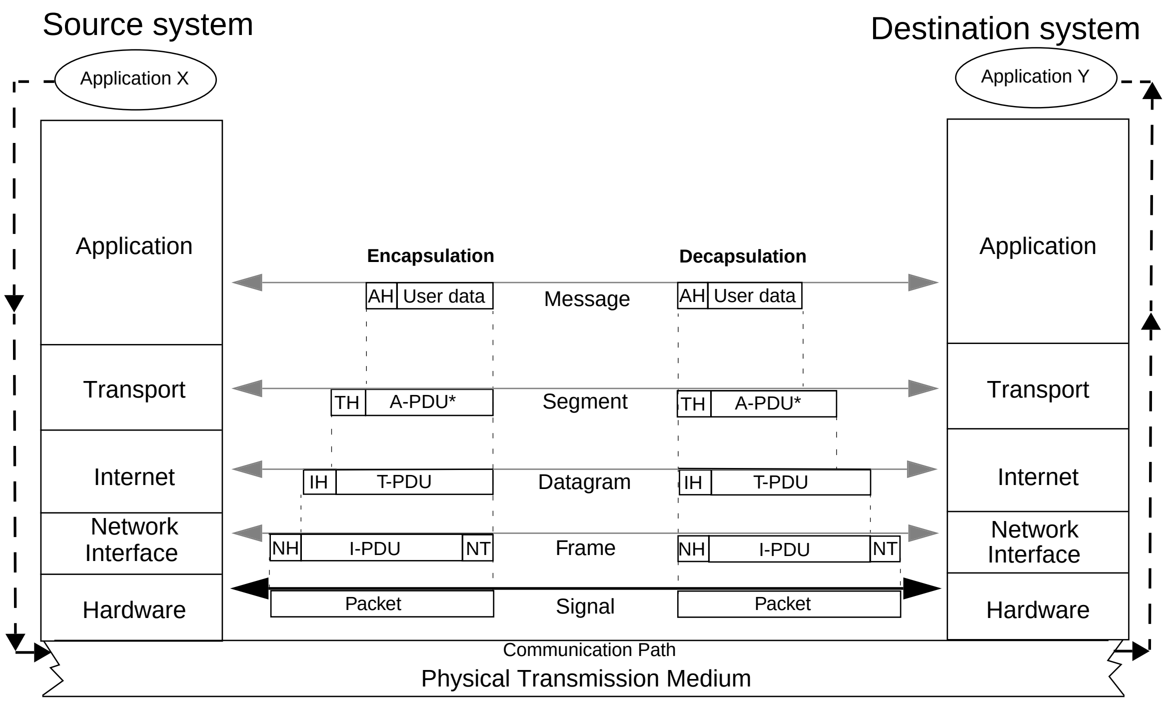
Application Layer



Peer-to-Peer Communication



Peer-to-Peer Communication



AH = Application header
TH = Transport header
IH = Internet header
NH = Network interface header
NT = Network interface trailer
PDU = Packet data unit

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

Network Media

- 10BASE-5
- 10BASE-2
- 10BASE-T
- 10BASE-F
  - 10BASE-FL
  - 10BASE-FB
  - 10BASE-FP
- 100BASE-TX

Network Media

- 100BASE-T4
- 100BASE-FX
- 1000BASE-X
  - 1000BASE-SX
  - 1000BASE-LX
  - 1000BASE-CX
- 1000BASE-T

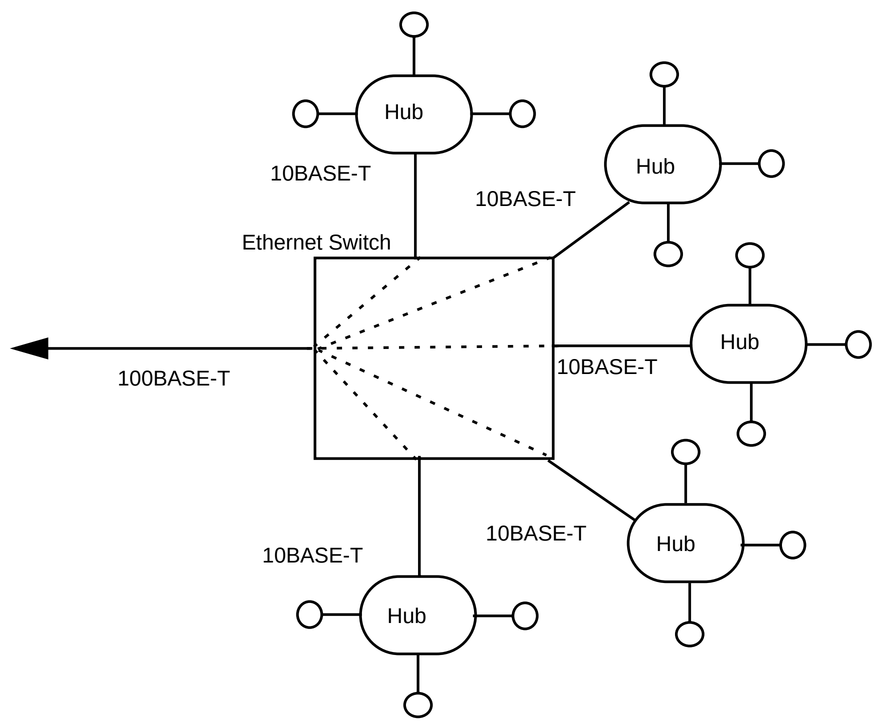
LAN Components

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

Switches

- 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.

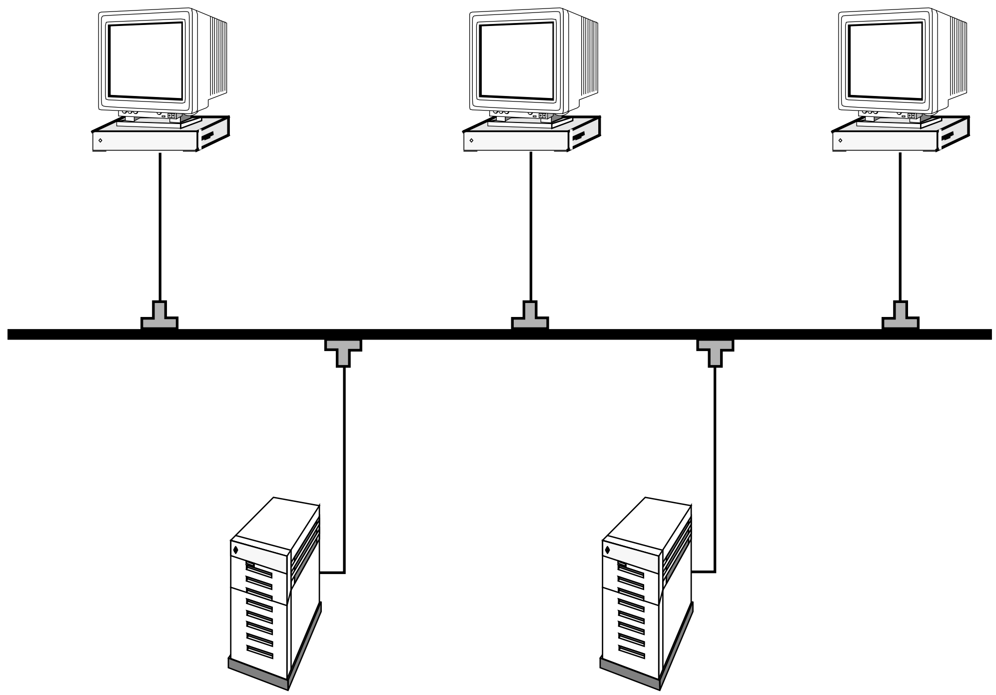
Switched Ethernet Diagram



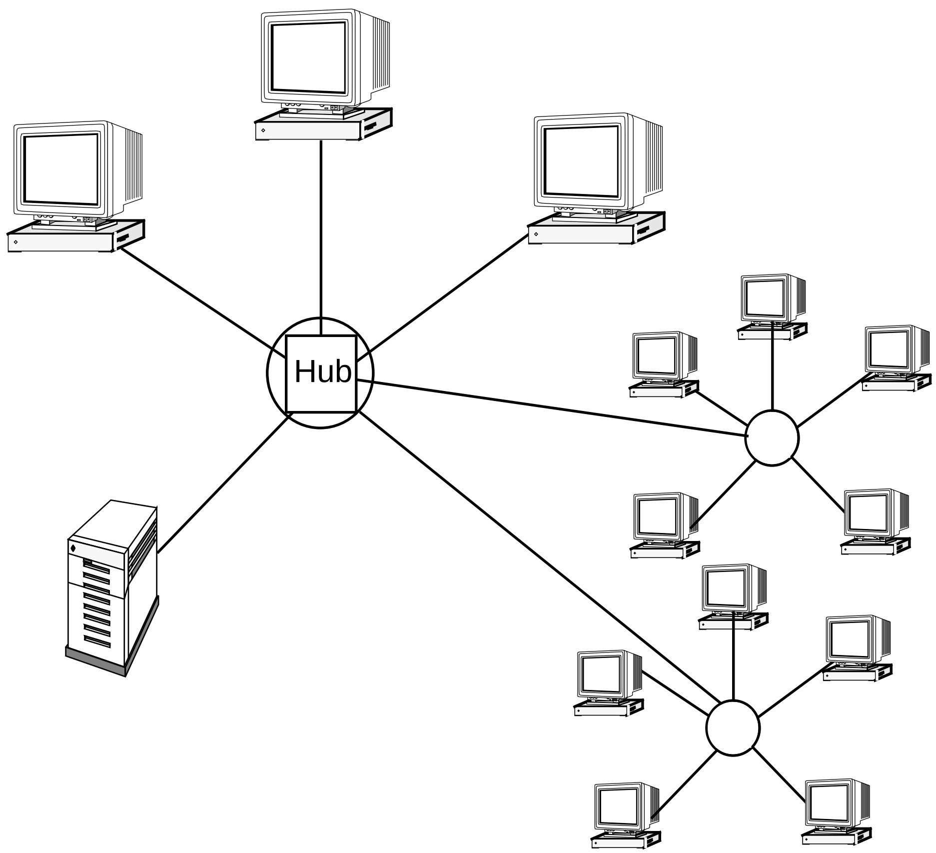
LAN Topology

- Bus
- Star
- Ring

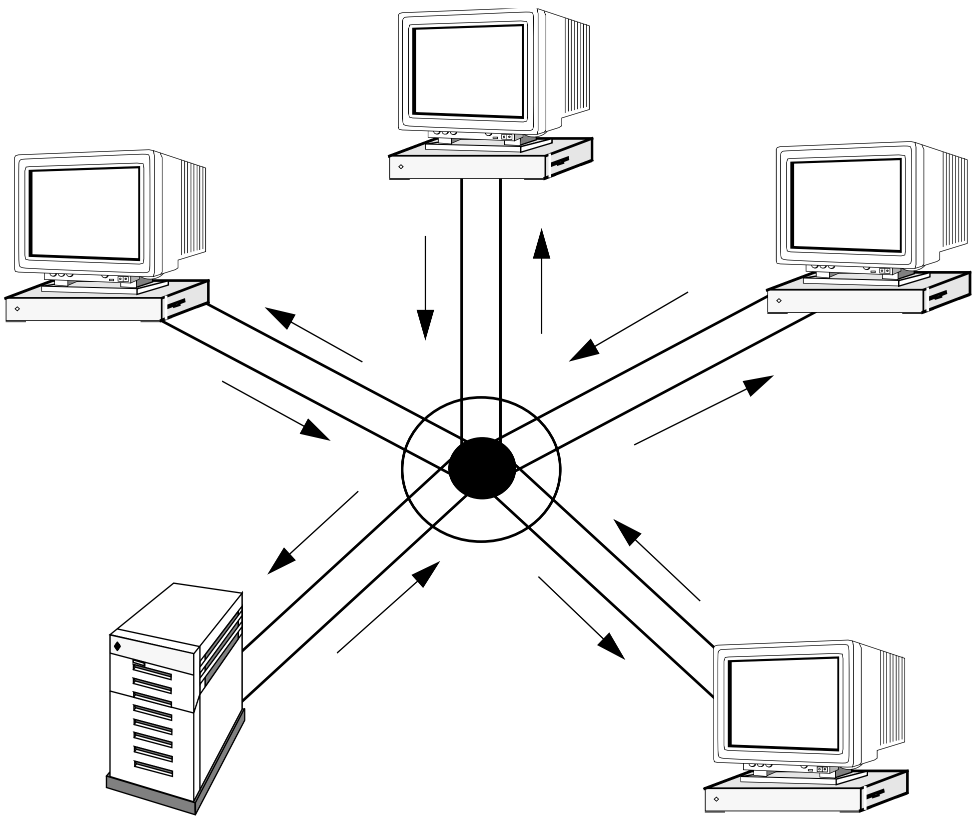
Bus Configuration



Star Configuration



Ring Configuration



LAN Methodologies

- Ethernet – IEEE 802.3
- Asynchronous Transfer Mode (ATM)
- Token Ring – IEEE 802.5
- Fiber Distributed Data Interface (FDDI)

Sun Communications Controller

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

Module 3

Ethernet Interface

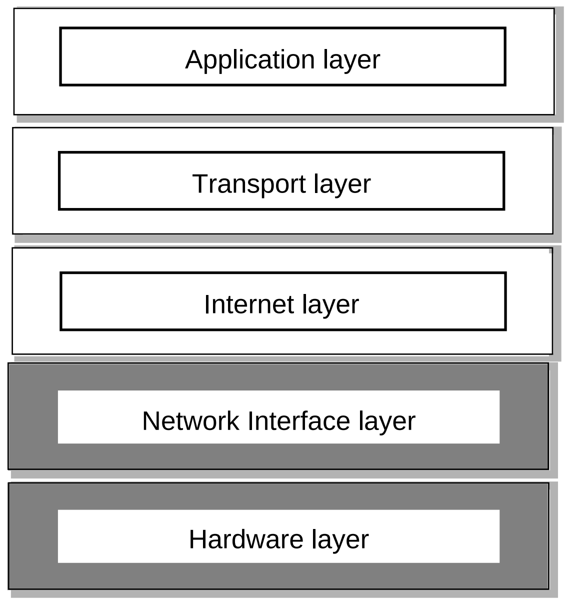
Overview

- 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



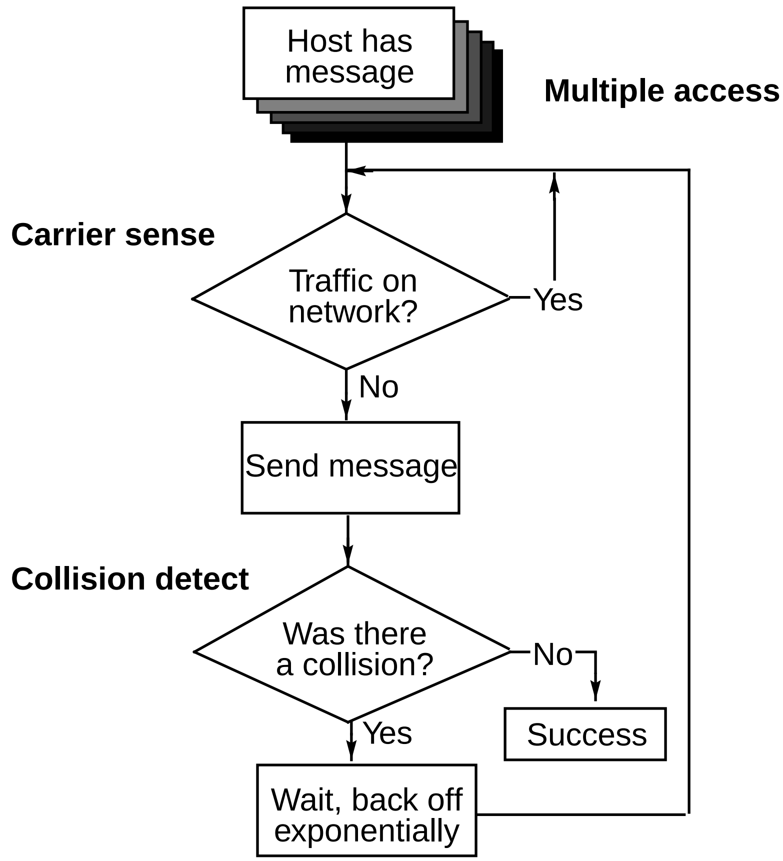
Ethernet Major Elements

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

The CSMA/CD Access Method

- 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



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

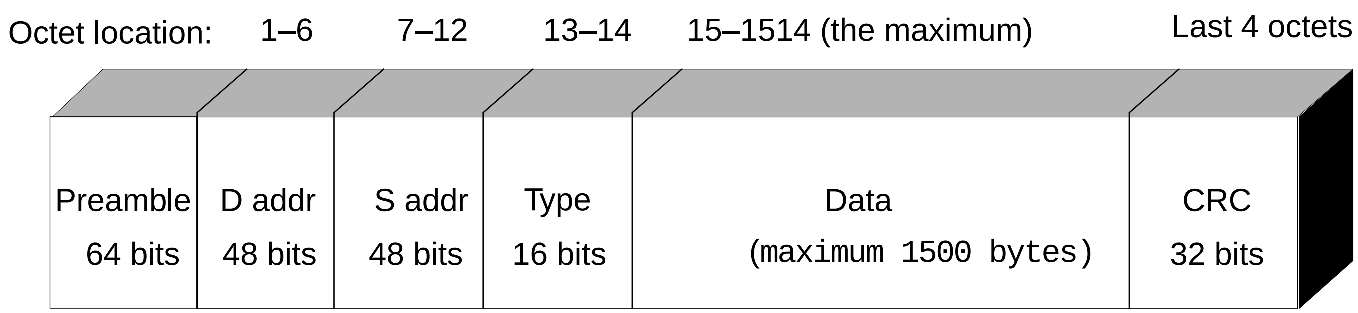
Sending Messages

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

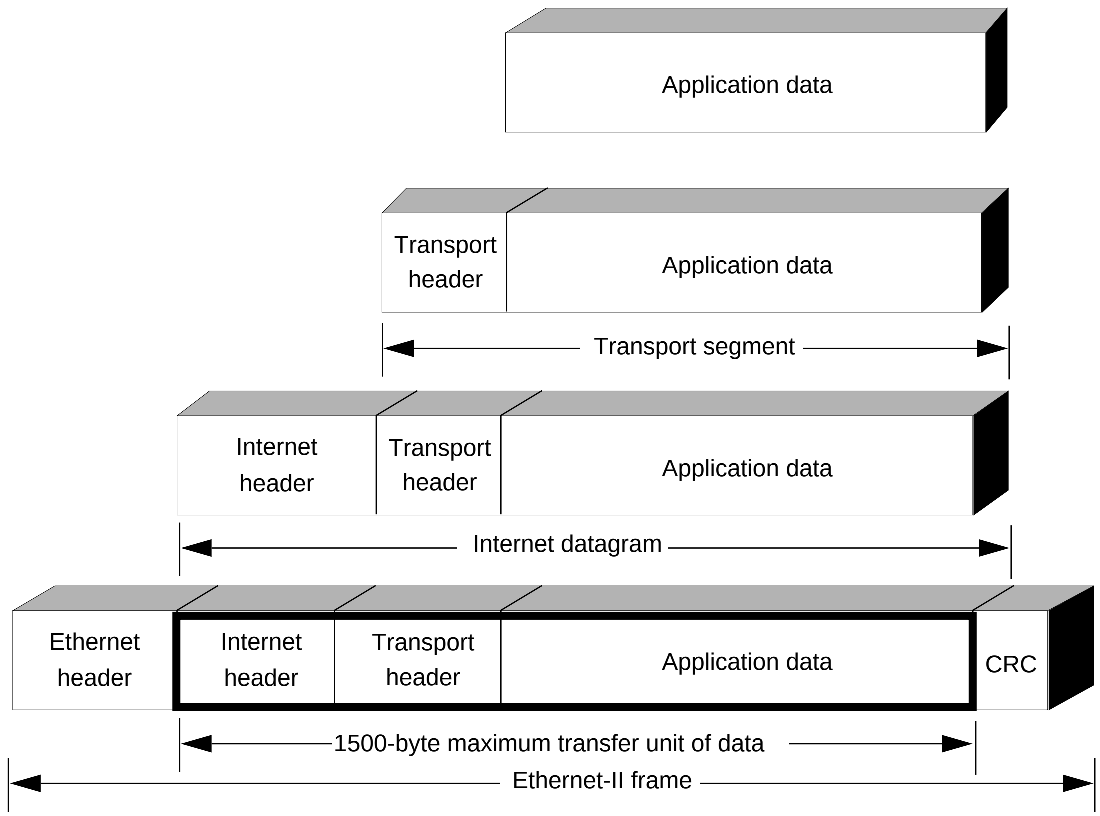
Ethernet-II Frame

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

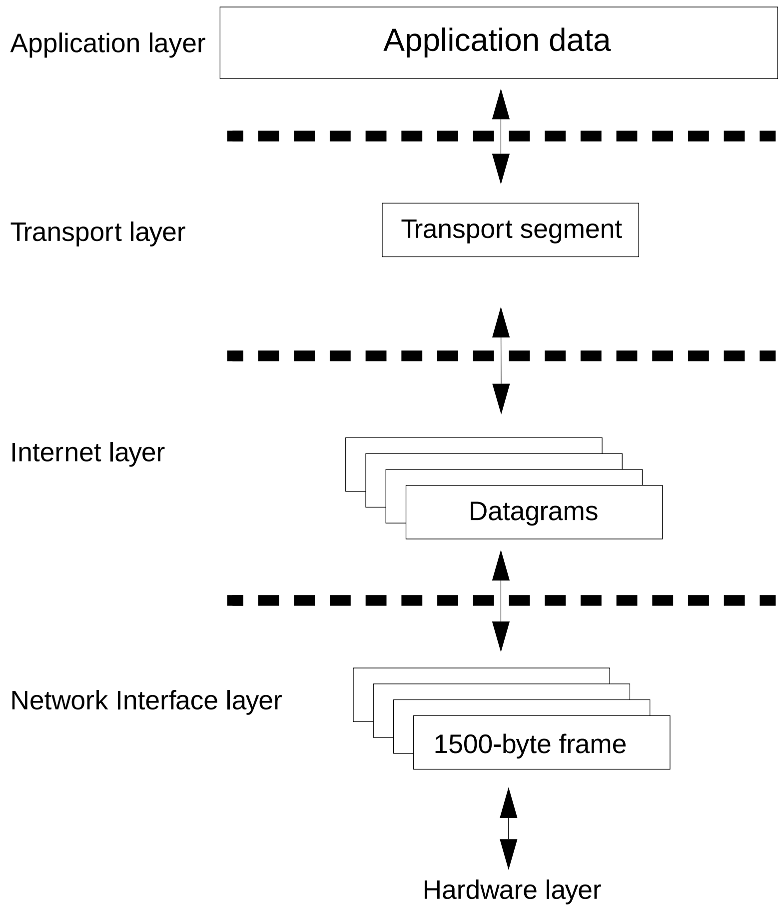
Ethernet-II Frame Fields



TCP/IP Layer Encapsulation



Ethernet Maximum Transfer Unit



Ethernet Error Checking

- Runts
- Jabbers
- Bad CRC
- Giants
- Long
- Frame Check Sequence (FCS) Error

Network Utilities

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

snoop

```
# snoop broadcast

Using device /dev/hme (promiscuous mode)
Using device /dev/hme (promiscuous mode)
lion --> 128.50.255.255 RUSERS C
bear --> 128.50.255.255 RUSERS C
lion -> (broadcast) RIP R (25 destinations)
lion -> (broadcast) RIP R (25 destinations)
lion -> (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 128.50.1.250
Using device /dev/hme (promiscuous mode)
________________________________
bear -> 128.50.1.250 ETHER Type=0800 (IP), size = 98 bytes
bear -> 128.50.1.250 IP  D=128.50.1.250 S=128.50.1.1 LEN=84, ID=7780
bear -> 128.50.1.250 ICMP Echo request
________________________________
128.50.1.250 -> bear ETHER Type=0800 (IP), size = 98 bytes
128.50.1.250 -> bear IP  D=128.50.1.1 S=128.50.1.250 LEN=84, ID=5905
128.50.1.250 -> bear 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  bear      77553   4    39221  2      2103   0
```

Module 4

ARP and RARP

Overview

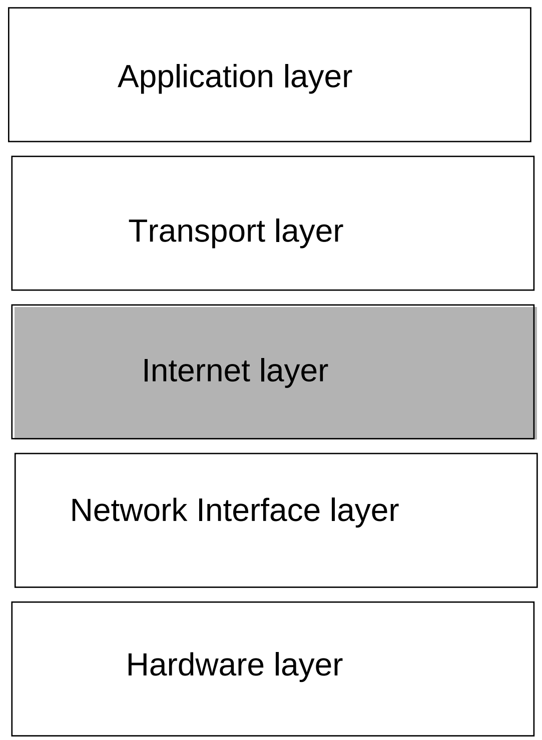
- Objectives
- Relevance

Introduction to Address Resolution

The two resolutions performed by the Address Resolution Protocol (ARP) and Reverse Address Resolution Protocol (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

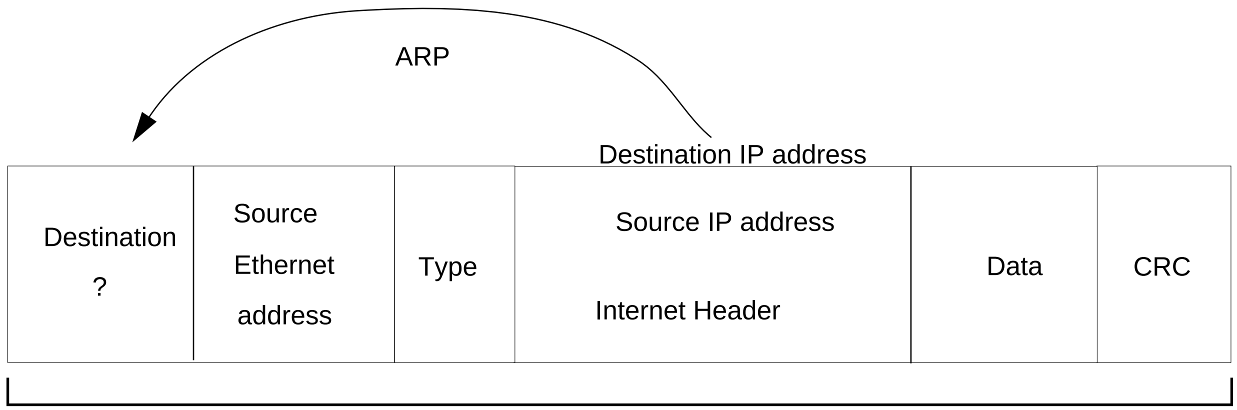
Address Resolution TCP/IP Layers



Why ARP Is Required

- Data is encapsulated into an Ethernet frame that 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

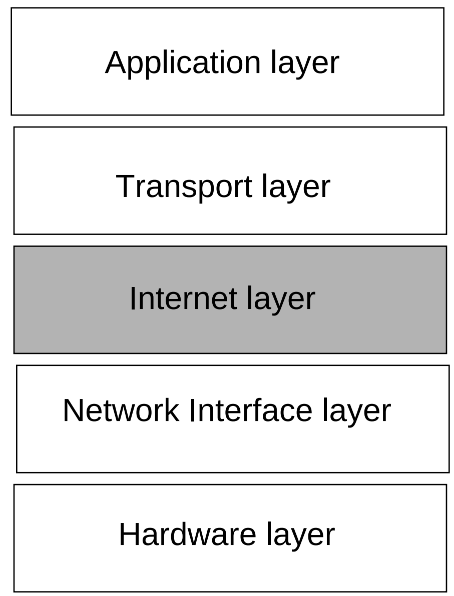
Overview

- Objectives
- Relevance

Introduction to Internet

- Berkeley Software Distribution
- Rapid growth
- The future

TCP/IP Layered Model



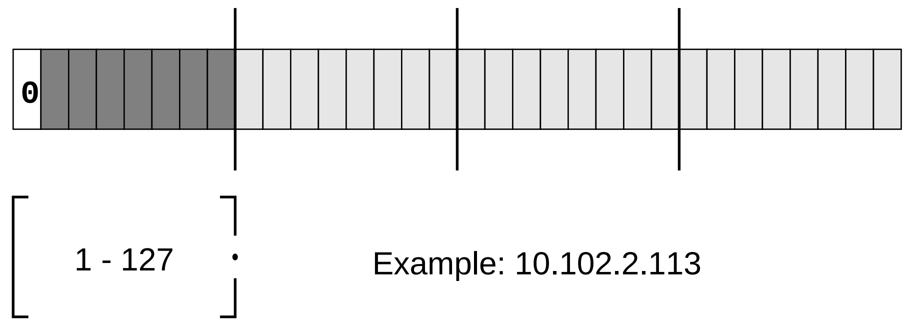
Internet Layer

- Internet Protocol
- Datagrams
- Internet Control Message Protocol (ICMP)
- Fragmentation

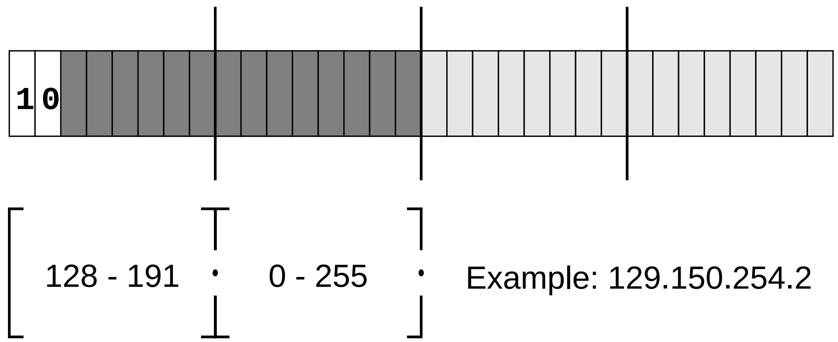
Classful IPv4 Addressing

- 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

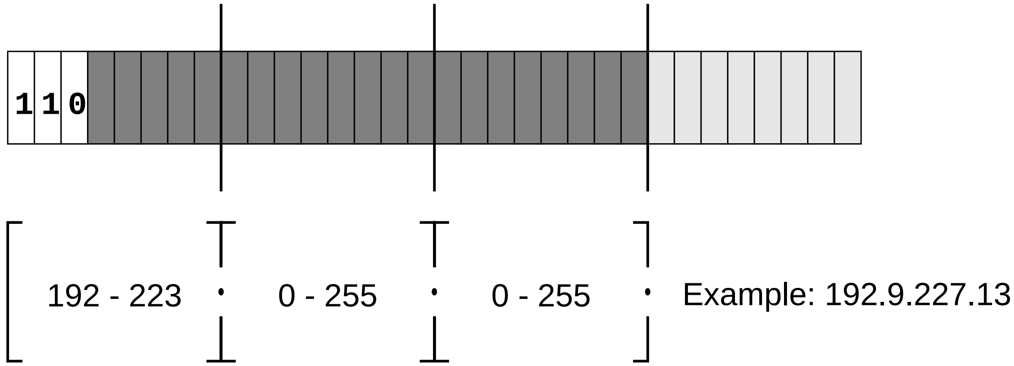
Class A Address Format



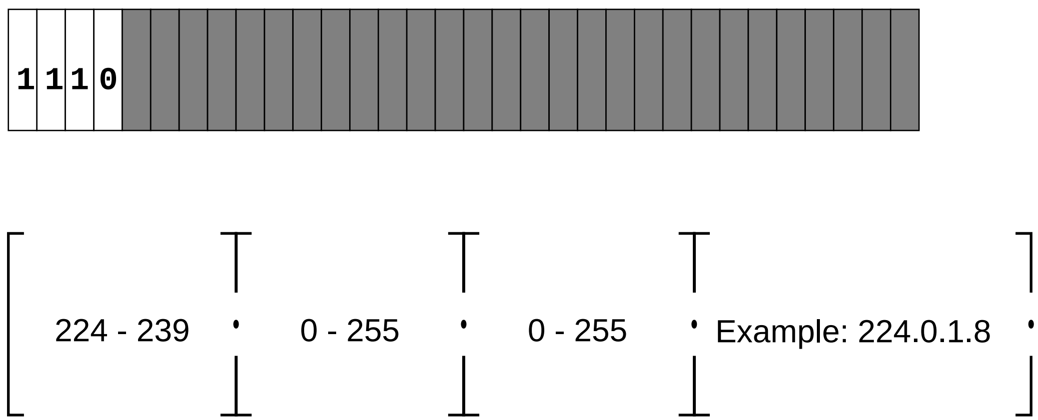
Class B Address Format



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.

`10.0.0.0 - 10.255.255.255`
`172.16.0.0 - 172.31.255.255`
`192.168.0.0 - 192.168.255.255`
	INTERNIC Pre-Reserved Private Network - see RFC 1918 for network numbers that are reserved for private use (networks not on the internet or behind a firewall using NAT).

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:

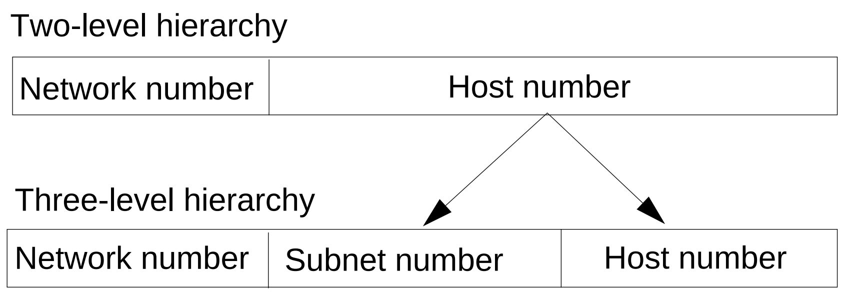
IPv4 address (decimal):	171	63	14	3
IPv4 address (binary):	10101011	00111111	00001110	00000011
AND netmask:	11111111	11111111	00000000	00000000
Network # (binary):	10101011	00111111	00000000	00000000
Network # (decimal):	171	63	0	0

Reasons to Subnetwork

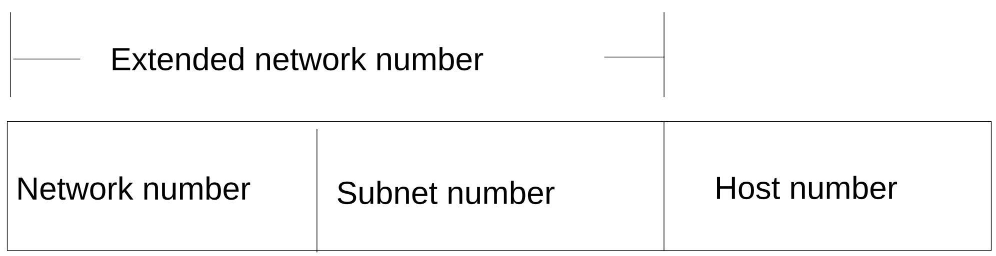
- Isolation of traffic
- Security
- Localization of protocols
- Geographical or departmental association
- Administration

Defining Subnets

- Address 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

	Network number		Host number	
	129	147	25	3
Two-level address	10000001	10010011	00011001	00000011
Subnet mask	11111111	11111111	11111111	00000000
	Default mask		Extended mask	No mask
Three-level address after logical AND	10000001	10010011	00011001	00000011
	129	147	25	3
	Network number		Subnet number	Host number

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

IPv4 address:	197.8.43.211	11000101 00001000 00101011 11010011
AND subnet mask:	255.255.255.240	11111111 11111111 11111111 11110000
Network number:	197.8.43.208	11000101 00001000 00101011 11010000
Subnet mask:	255.255.255.240	11111111 11111111 11111111 11110000
NOT subnet mask		00000000 00000000 00000000 00001111
OR network number:	197.8.43.208	11000101 00001000 00101011 11010000
Broadcast number:	197.8.43.223	11000101 00001000 00101011 11011111

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 Recommended

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

Subnet Masks

- Contiguous – Recommended
- Non-contiguous – Not recommended

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
```

Variable Length Subnet Masks

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

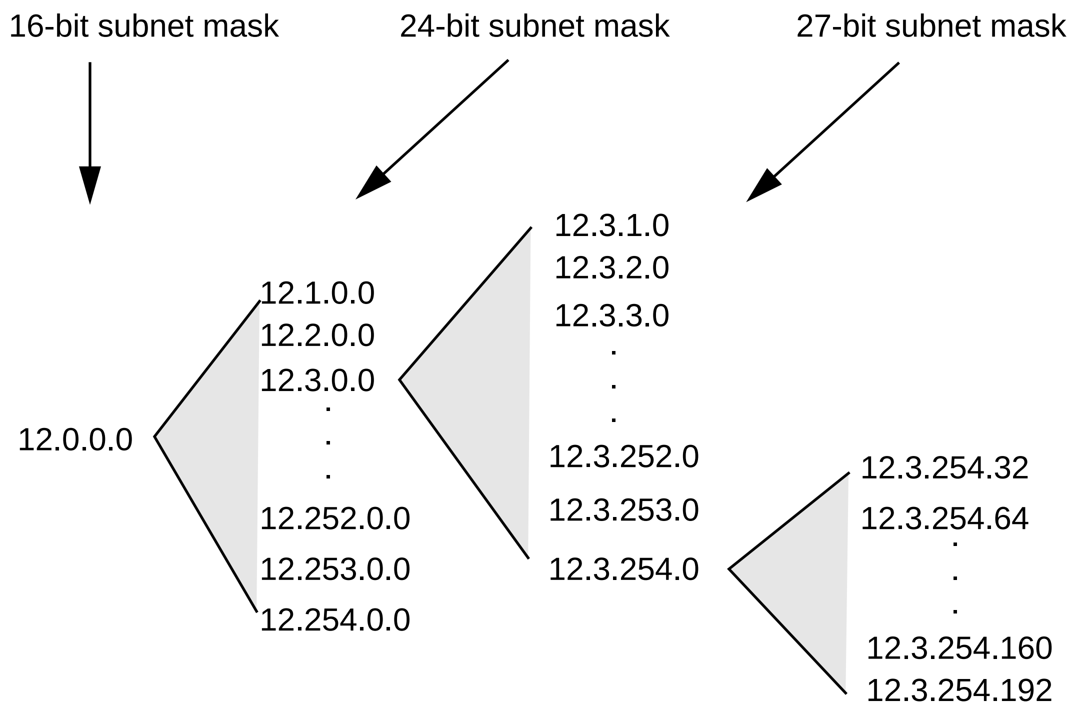
Class B Subnet Mask Yield

11111111 11111111 11111100 00000000

1024 -2 hosts per subnet

64 subnets per network

Class A Network Using VLSM



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

Network Interface Configuration

`/platform/SUNW,x/kernel/unix`

1. The system boots from the UNIX kernel. One of the processes it runs is `/sbin/init`.

`/sbin/init`

2. The `/sbin/init` process reads the `/etc/inittab` configuration file, which runs, among other scripts, the `/sbin/rcS` script.

`/sbin/rcS`

3. The `/sbin/rcS` script sets the system to single-user mode, including starting the `/etc/rcS.d/S30network.sh` script.

`/etc/rcS.d/S30network.sh`

4. The `/etc/rcS.d/S30network.sh` script configures the Ethernet and loopback interfaces, in addition to mounting the `/usr` file system as read-only.

`/sbin/ifconfig` Command

- Configures network interfaces
- Is invoked by `/etc/rcS.d/S30network` at startup

Examining Network Interfaces

```
# ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232 index 1
    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
index 3
    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 down
# ifconfig hme0
hme0: flags=862<BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
index2
    inet 128.50.1.2 netmask ffff0000 broadcast 128.50.255.255
    ether 8:0:20:75:6e:6f

# ifconfig hme0 up
# ifconfig hme0
hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
index 2
    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 index 3
    inet 0.0.0.0 netmask 0 ether 8:0:20:75:6e:6f
```

Set IP Address, Enable Interface, and Disable Trailers

```
# ifconfig hme0 128.50.1.2 -trailers up
# ifconfig hme0
hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
index 3
    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 index 3
    inet 128.50.1.2 netmask ffffff00 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

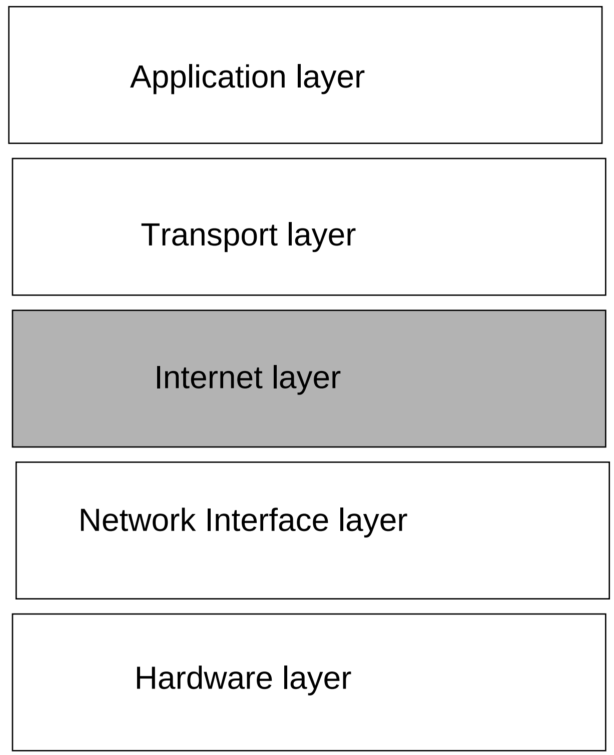
Overview

- Objectives
- Relevance

Introduction to Routing

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

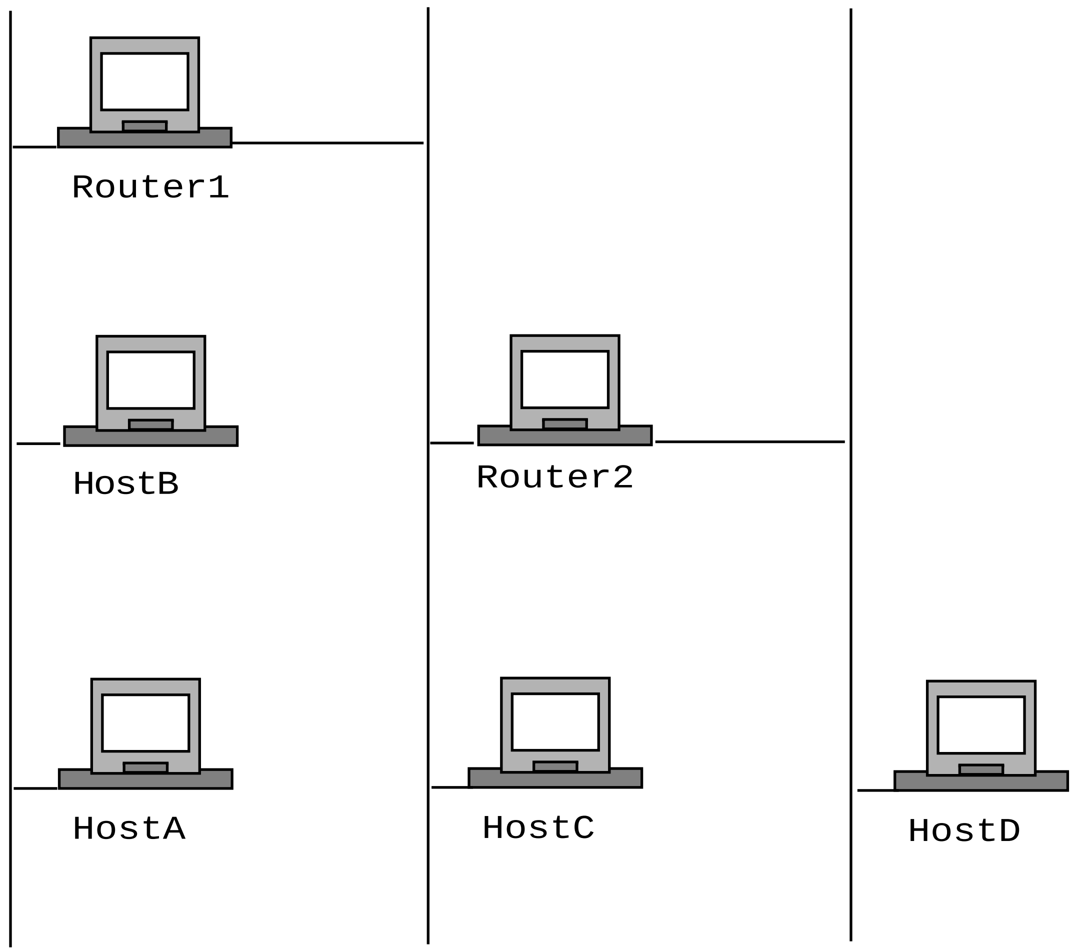
Internet TCP/IP Layer



Routing Schemes

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

Routing Schemes



Manually Manipulating Routing Table

- Add a route

```
# route add net 128.50.3.0 tuna1
```

- Add a route using a network name

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

- Delete a route

```
# route delete net 128.50.3.0 sword-r
```

- Flush routing table

```
# route flush
```

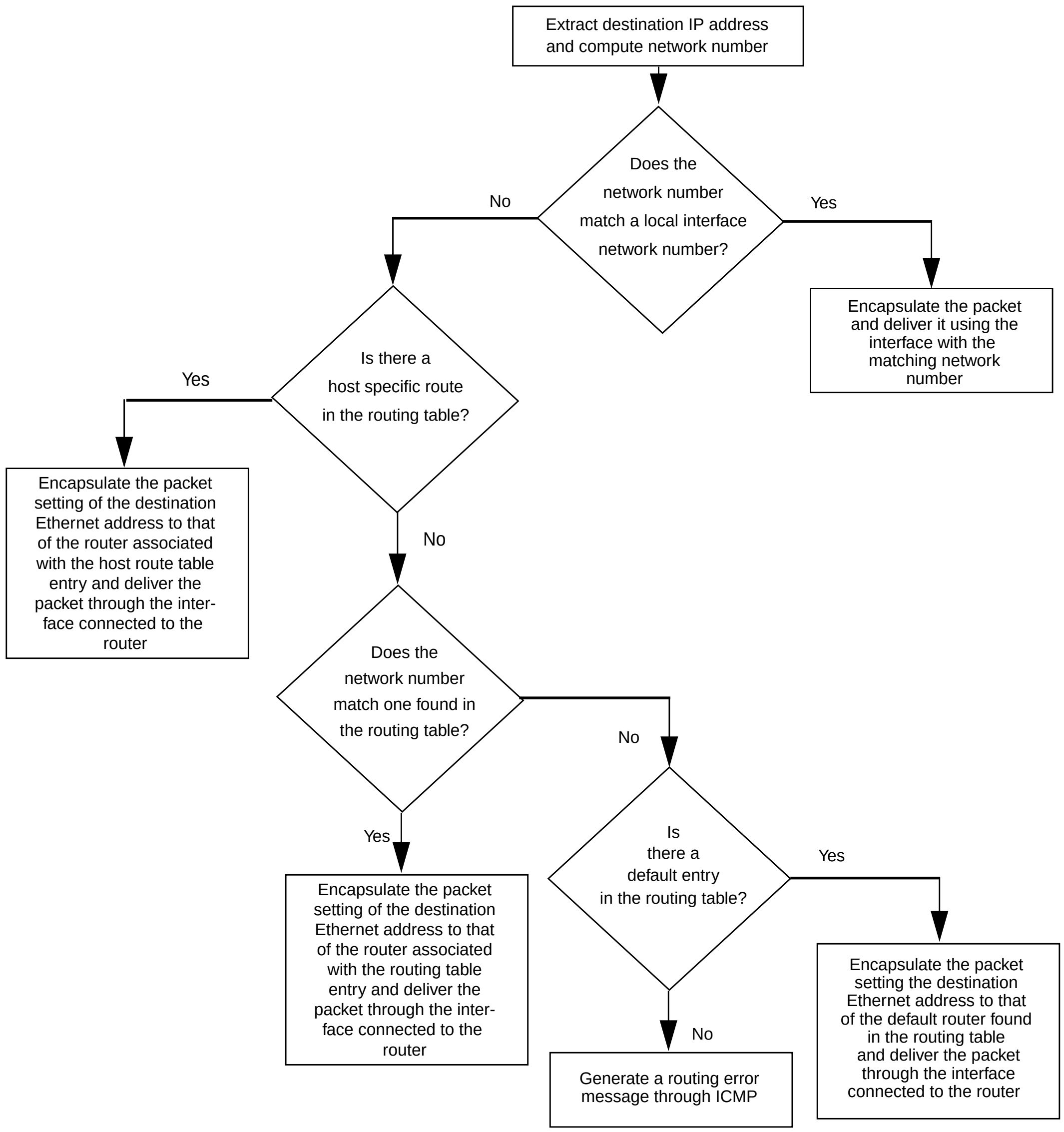
- Add multicast path for 224.0.0.0

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

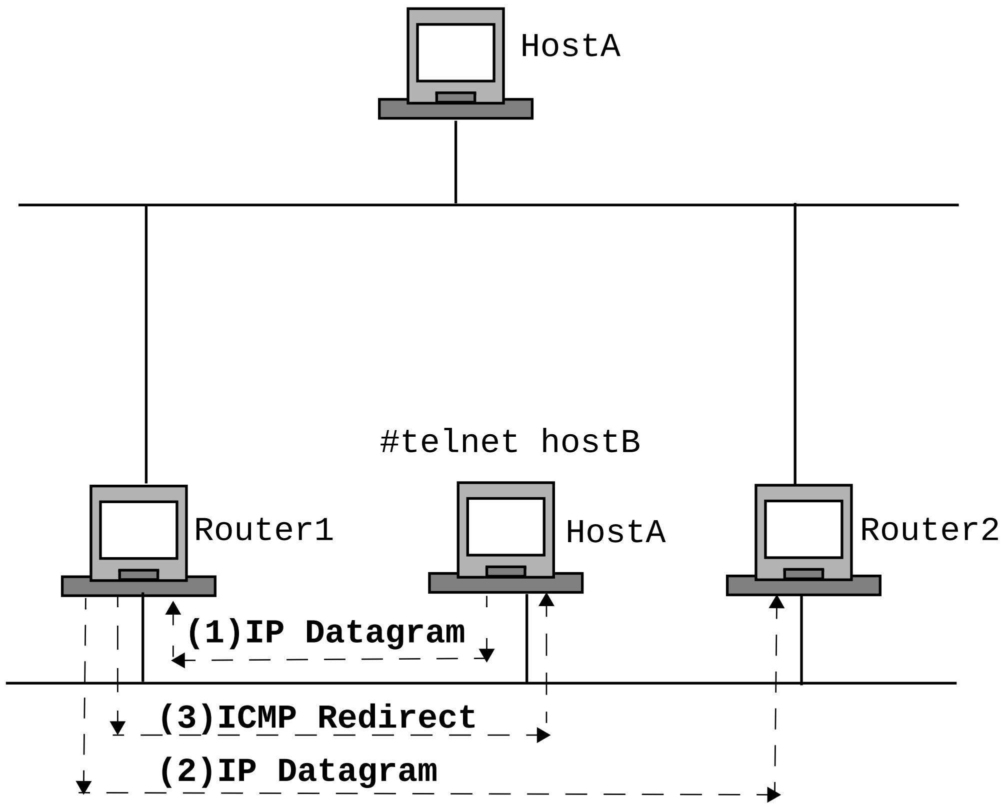
Routing Algorithm

- 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



ICMP Redirect



Router Configuration

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

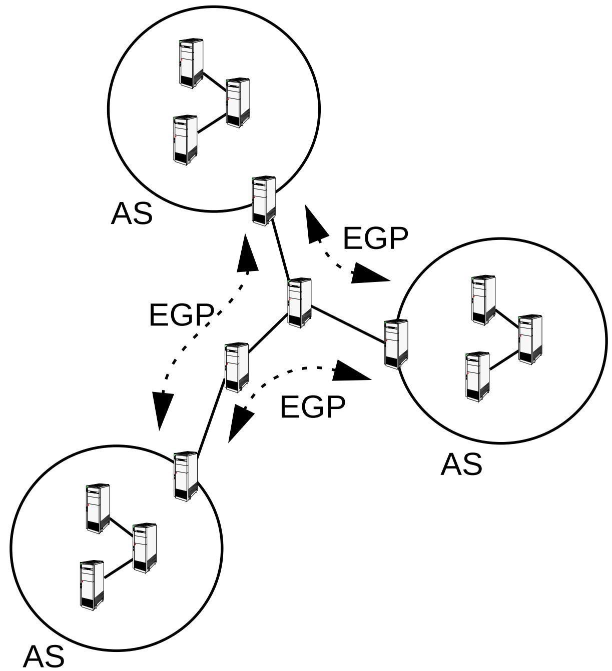
Autonomous System

- Collection of networks and routers under a single administrative control
- Associated routing table protocols
  - Exterior Gateway Protocol
  - Interior Gateway Protocol
  - Allows use of Classless Interdomain Routing (CIDR)

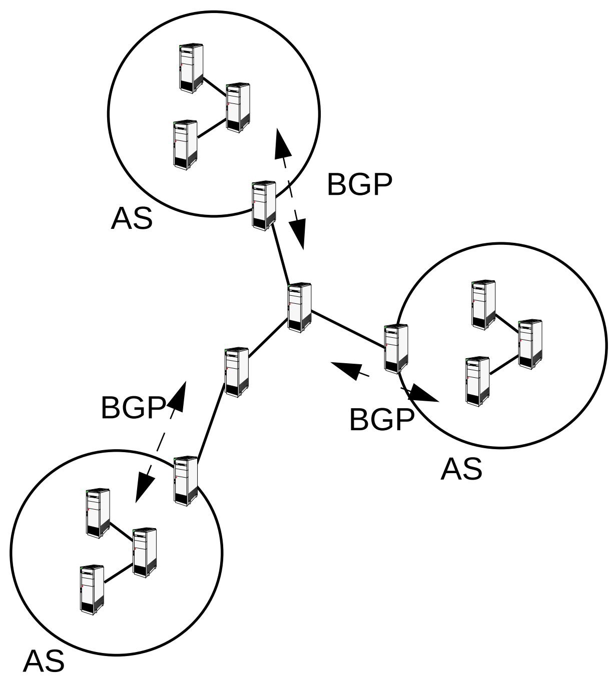
Gateway Protocols

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

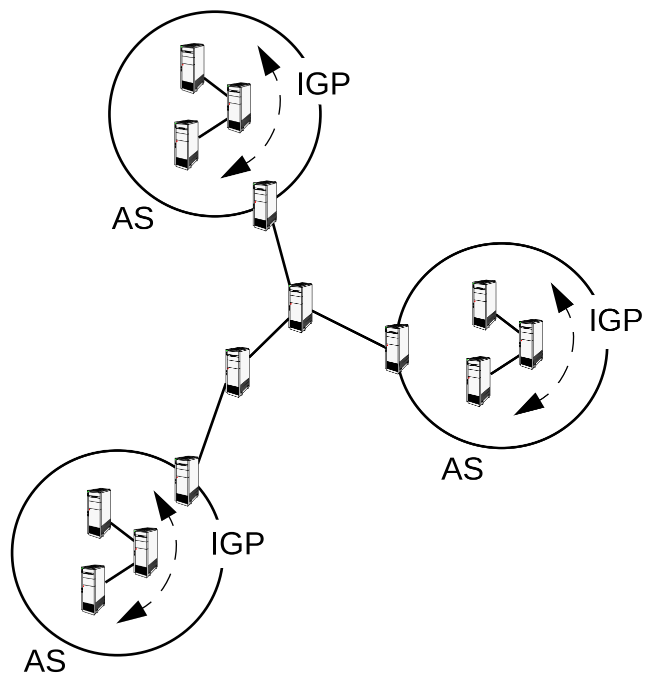
Exterior Gateway Protocol



Border Gateway Protocol



Interior Gateway Protocol



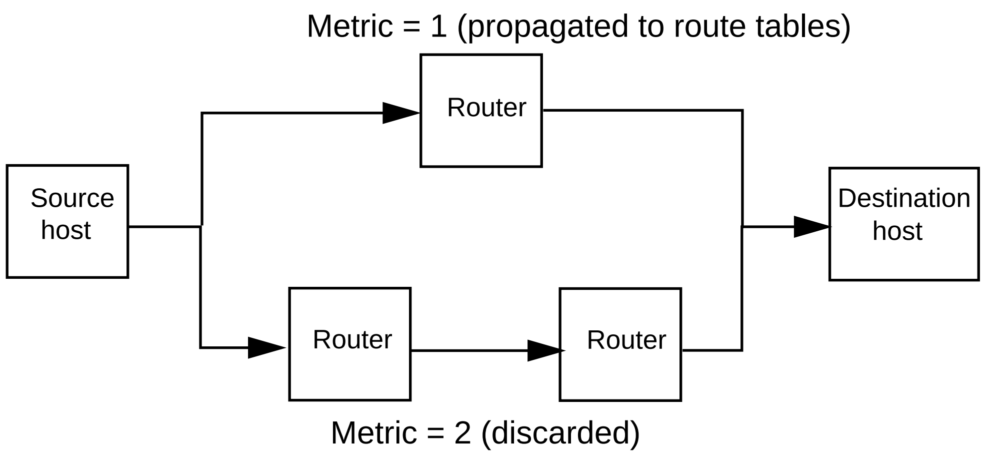
Open Shortest Path First

- 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
```

`/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]`

- The following directives may also be put in a `/etc/gateways` file:
  - `norip <interface>`
  - `noripin <interface>`
  - `noripout <interface>`

Network Router Discovery

- 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

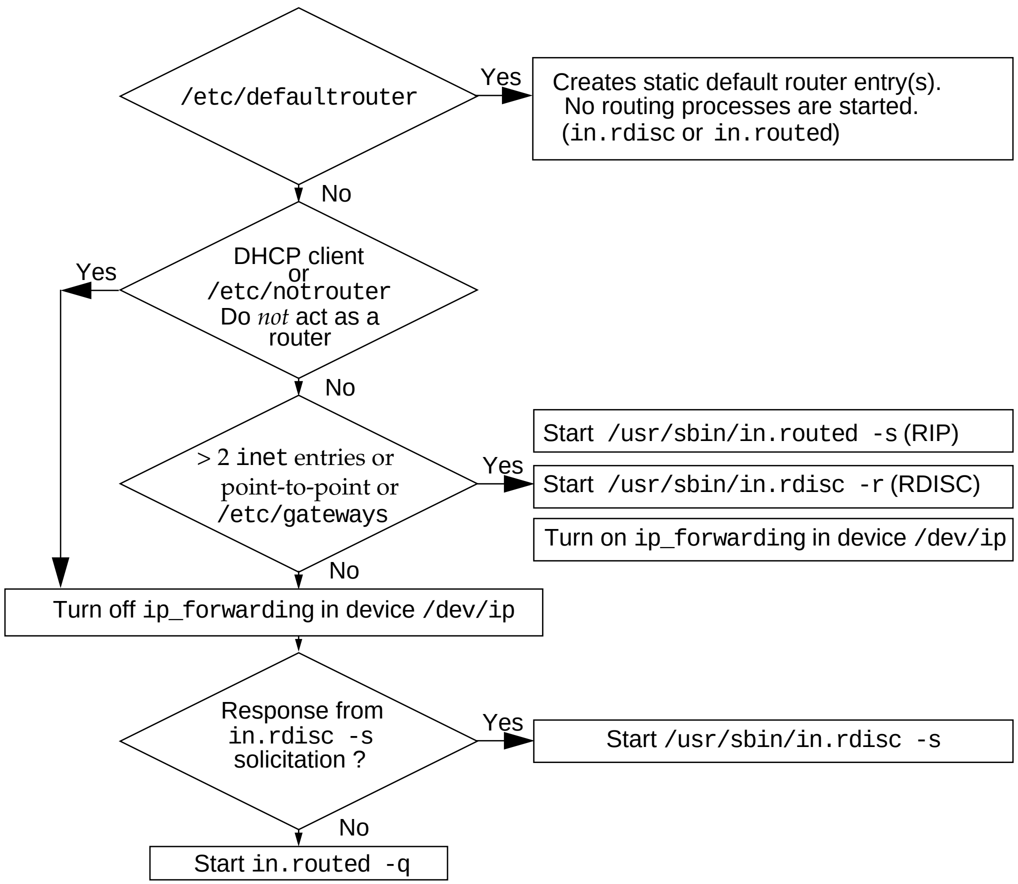
- Non-router host

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

- Router host

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

Routing Initialization



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

/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             lion-r  UG     3562    hme0
veggie          potato-r    UG      10   1562
224.0.0.0       bear        U       3    0      hme0
```

Troubleshooting Router Configuration

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

Module 7

Transport Layer

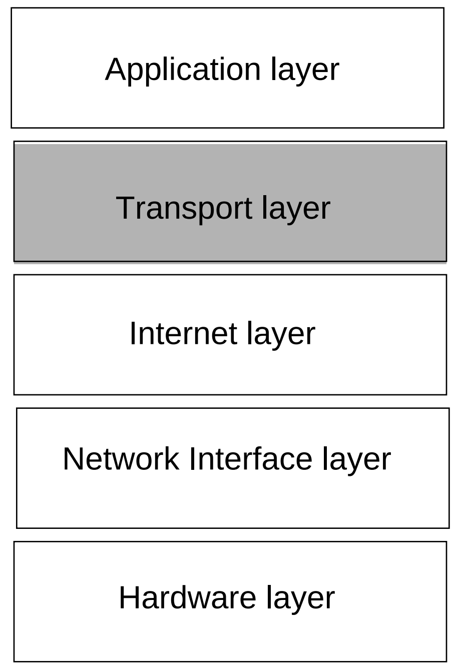
Overview

- Objectives
- Relevance

Introduction to the Transport Layer

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

TCP/IP Layered Model



Types of Protocols

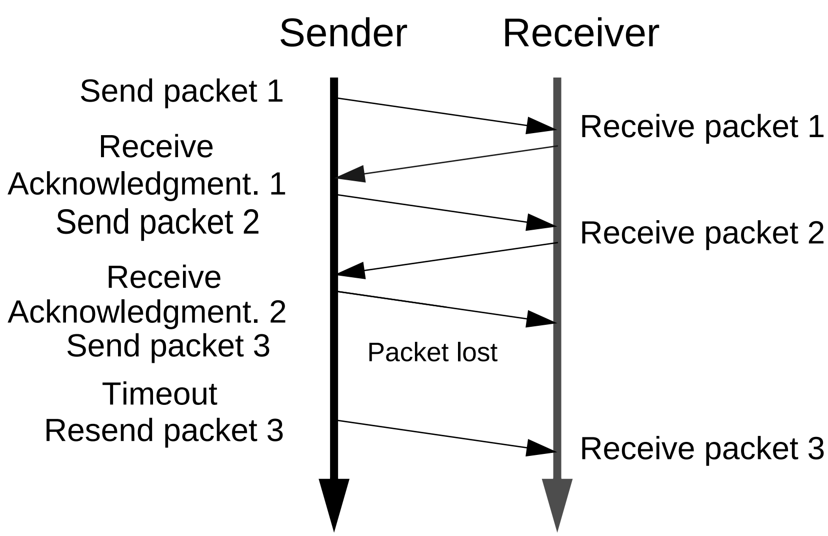
- Connection oriented
  - Is highly reliable
  - Requires more computational processing
- Connectionless
  - Has virtually no reliability features
  - Requires that transmission quality be augmented
  - Is very fast

Stateful Compared to Stateless Protocols

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

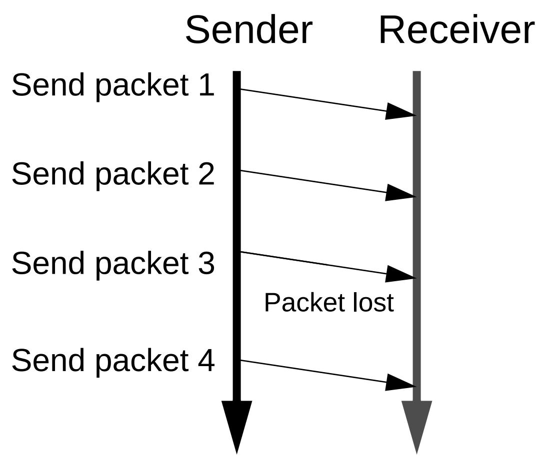
Reliable Protocols

- Requires transmission acknowledgment



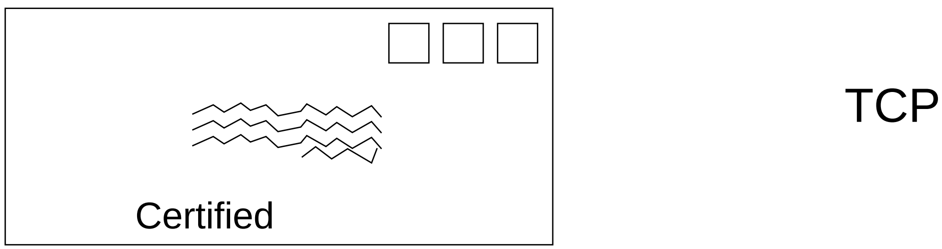
Unreliable Protocols

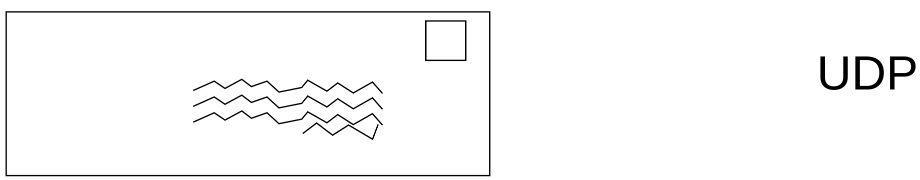
- No transmission acknowledgment



Transport Protocols

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





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

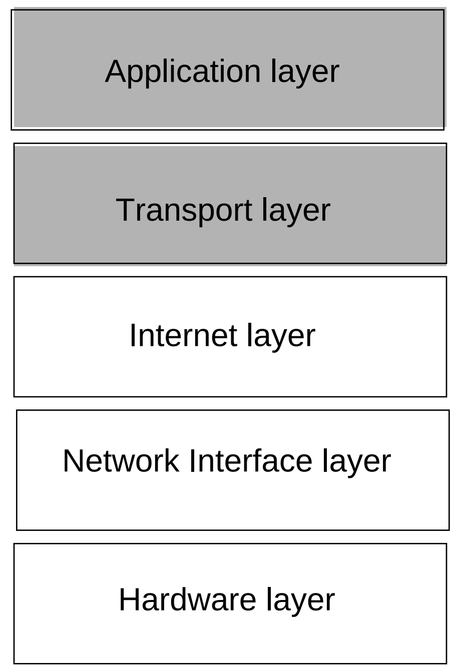
Overview

- Objectives
- Relevance

The Client-Server Model

- Service
- Client
- Server
- TCP/IP model

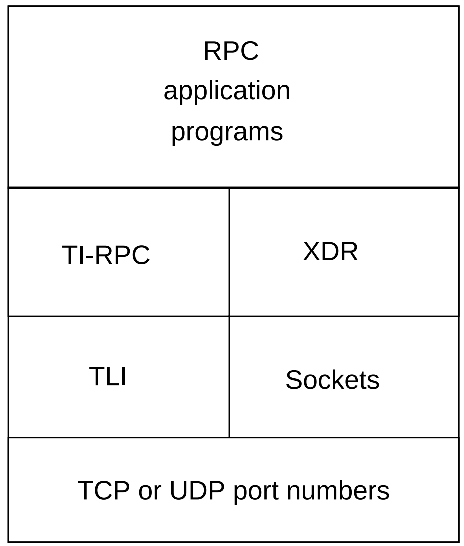
Application Layer



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



ONC+ Technologies

- XDR
- TLI
- Sockets
- NFS
- 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                   Remote
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 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

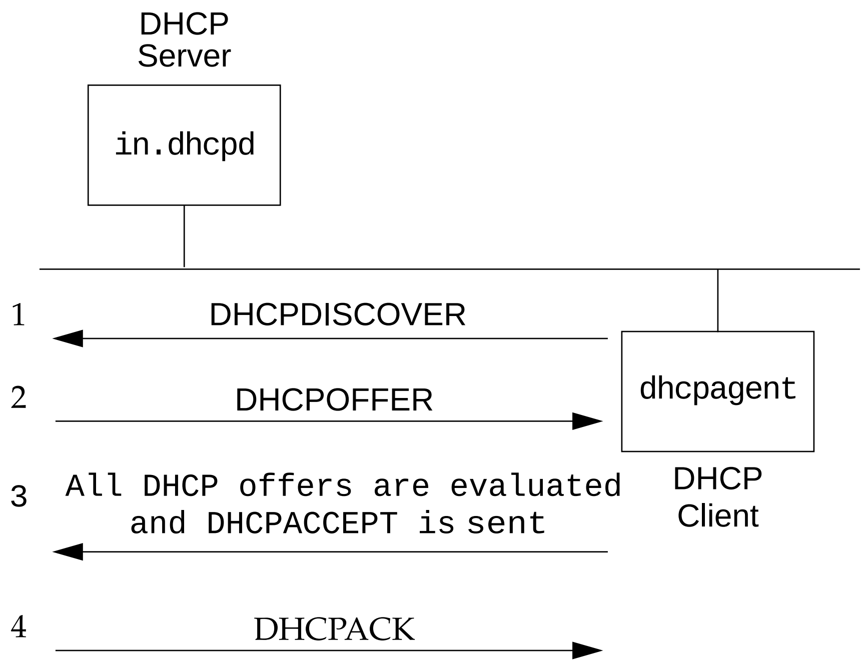
- Automatic management of IP addresses
- Support for BOOTP clients
- Programmable lease times
- Dynamic IP addresses assigned to selected Ethernet hardware addresses
- Dynamically allocated pool or pools of IP addresses on the same network
- Two or more dynamic IP address pools on separate IP networks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

DHCP Client-Server

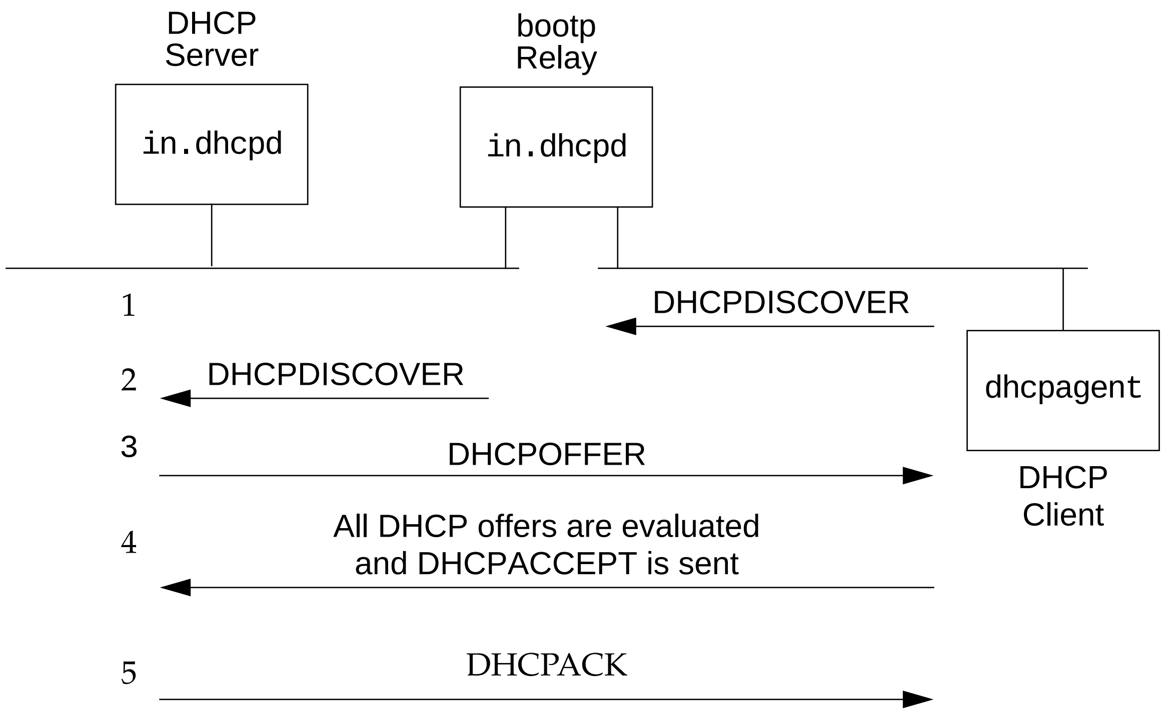
The DHCP protocol has two functions with regard to the client:

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

DHCP Client and Server Interaction



DHCP Client and Server Interaction Across a bootp Relay



Server Side

- DHCP server manages the IP address space of networks directly connected to that server.
- BOOTP relay agents allow fowarding of DHCP or BOOTP requests to server on other networks.
- Servers are configured as primary and/or secondary.
  - The primary server passes IP addresses to the client.
  - The 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 that determines how client machines access a network

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 `dhcpconfig` 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.

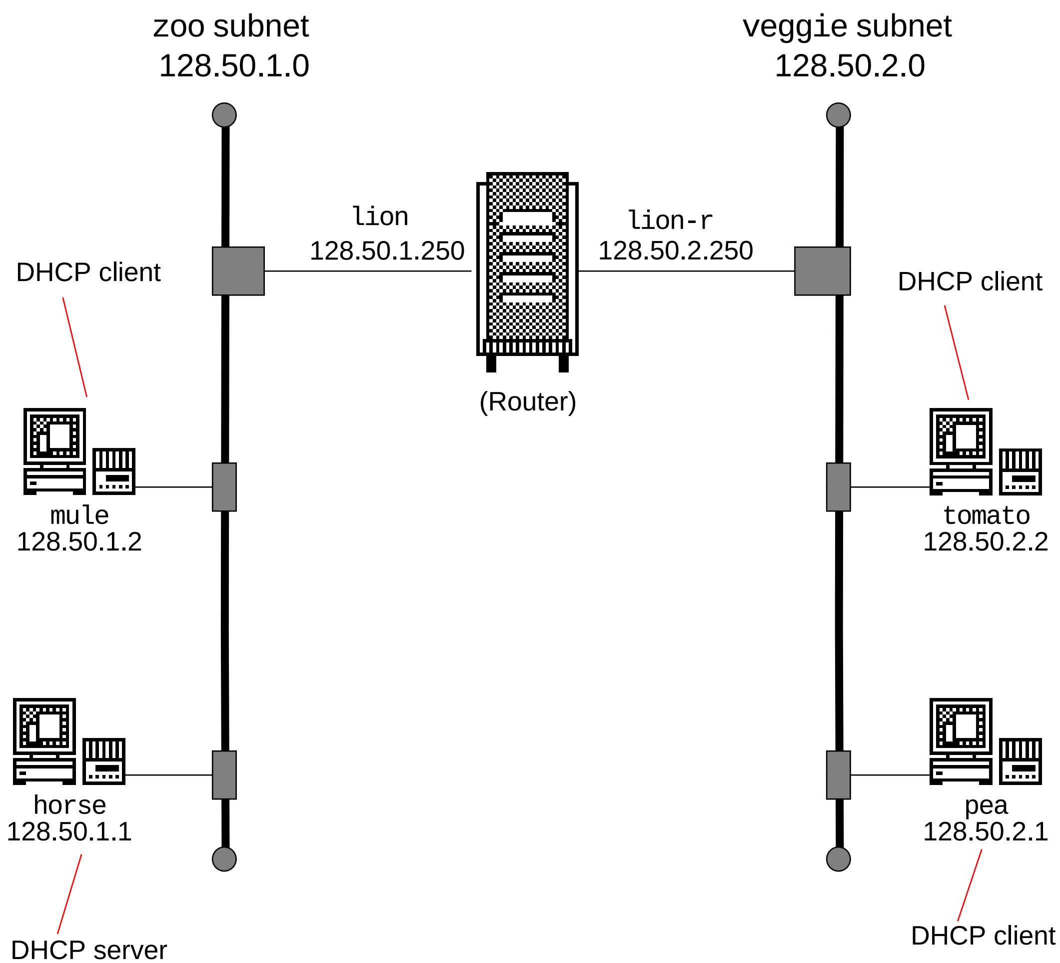
  Example for interface `hme1`:

  ```
  # 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

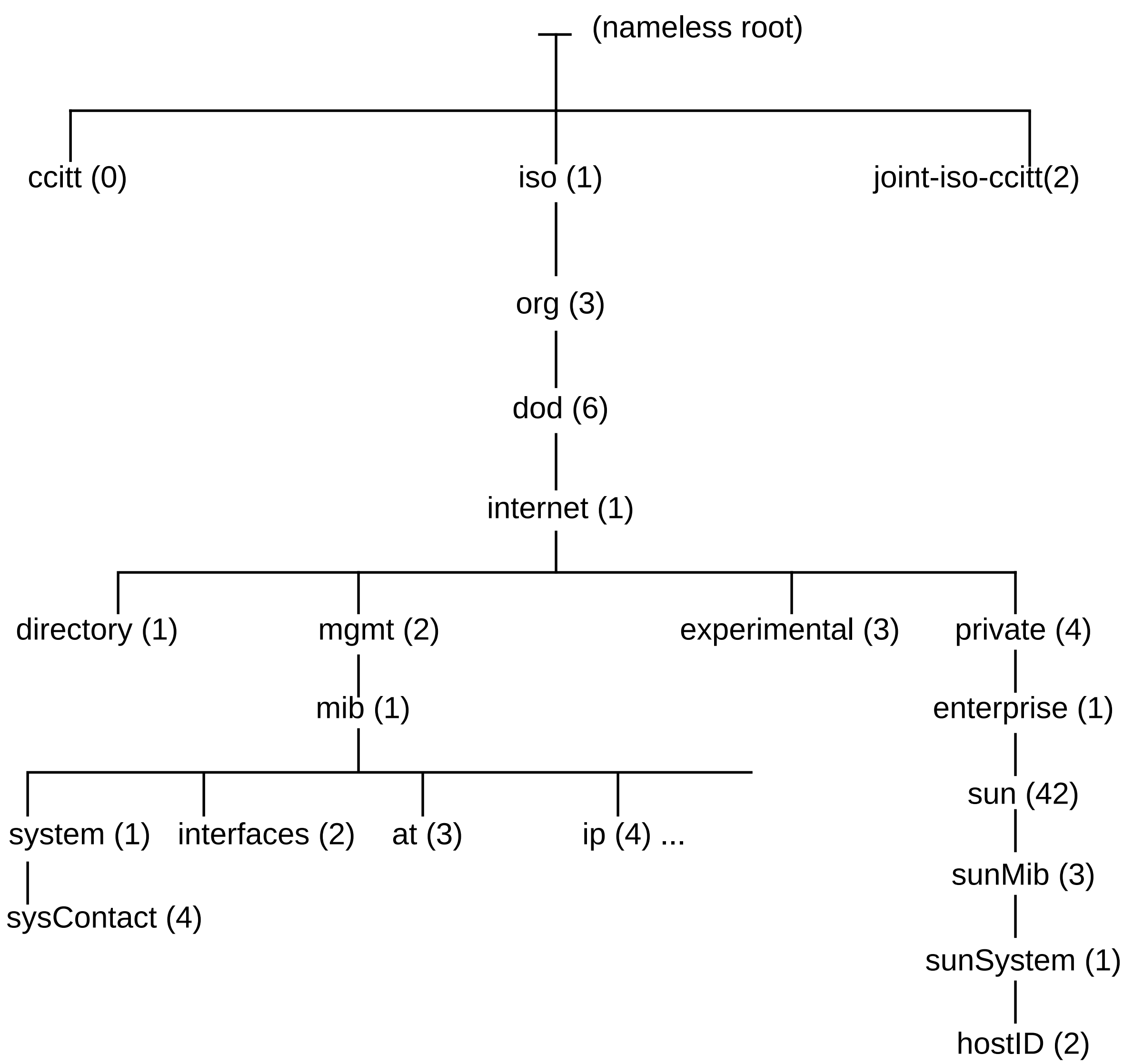
Network Management

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

Introduction to SNMP

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

OID Global Tree



Introduction to SNMP

- Management Information Base (MIB)
- ASN.1

SNMP-based Management Applications

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

Module 11

Domain Name System

Overview

- Objectives
- Relevance

A Brief History of DNS

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

BIND

- Most frequently used DNS implementation
- Available at `http://www.isc.org/bind.html`
- Solaris version 8 implements BIND version 8.1.2
- Latest BIND version may not be supported

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

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

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

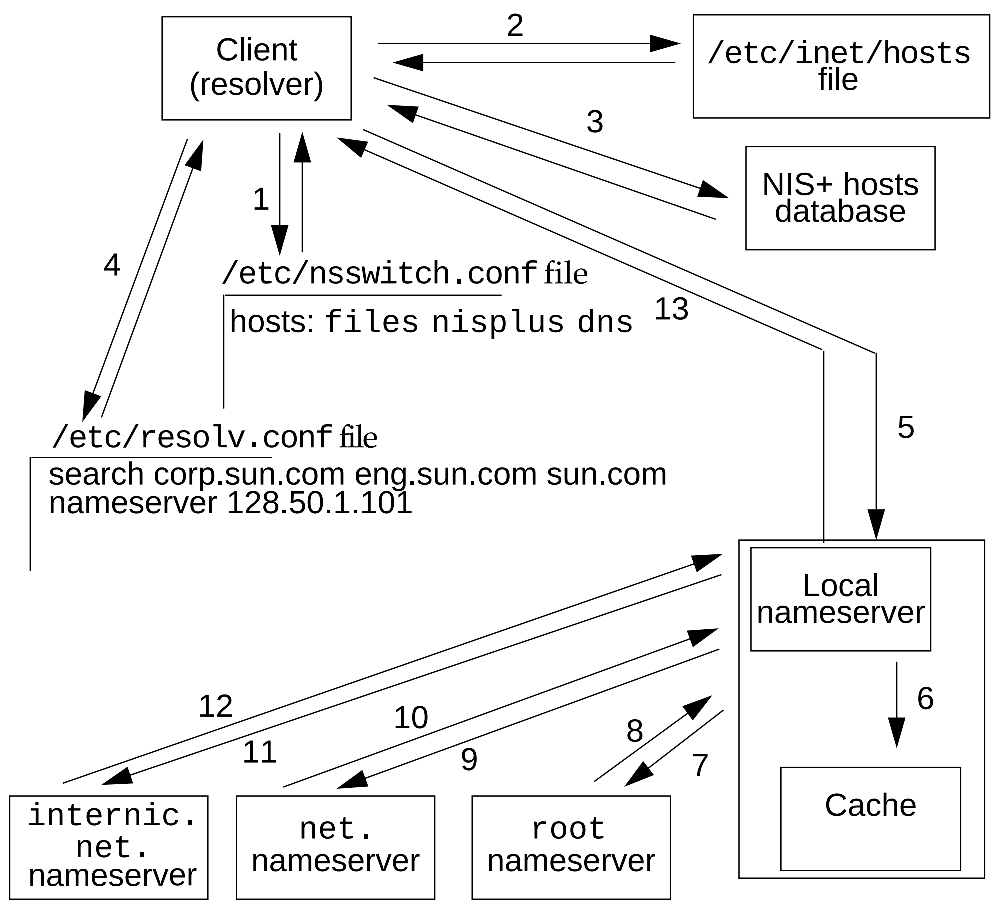
DNS Answers

- 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



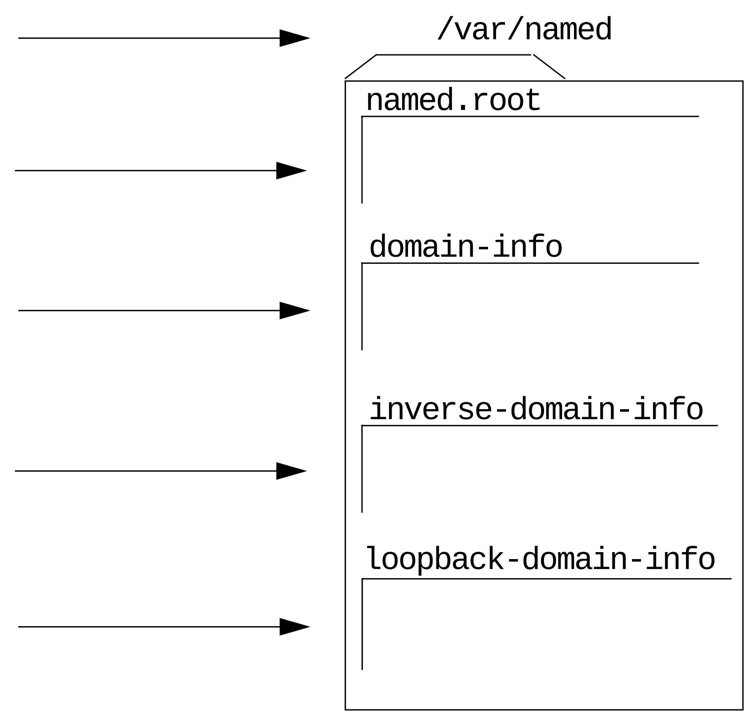
DNS Server Configuration

- 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 “zoo.edu.zone”;
};
zone “1.50.128.in-addr.arpa” in {
        type master;
        file “zoo.edu.rzone”;
};
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 that 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.

`/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.

@                       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
```

`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

- `pkill -INT in.named`
- `pkill -USR1 in.named`
- `pkill -USR2 in.named`
- `pkill -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;
         };

};zone "150.128.in-addr.arpa" in {

type slave;

file "zoo-rbackup";

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

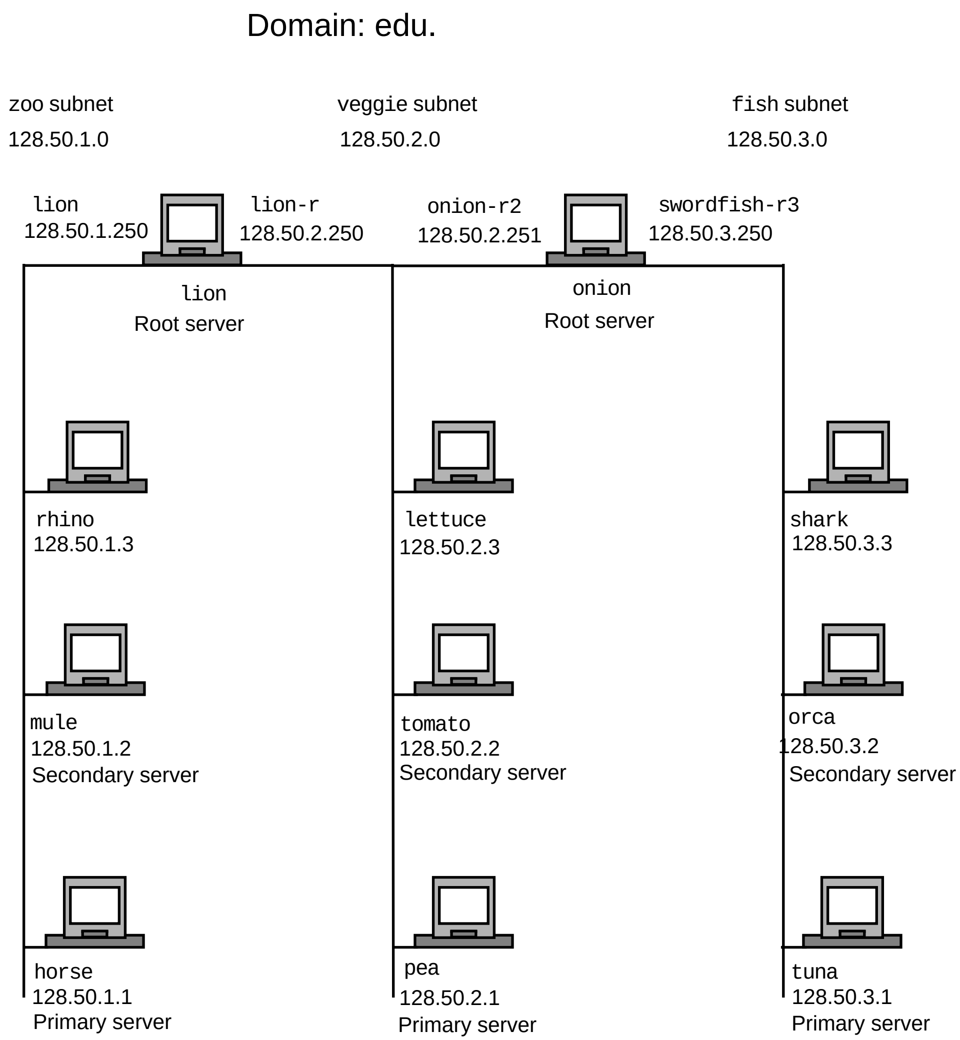
Miscellaneous DNS Topics

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

DNS Resources

- `info.bind` newsgroup
- `http://www.internic.net.`
- RFCs

DNS Lab Layout



Module 12

Introduction to NTP

Overview

- Objectives
- Relevance

What is Network Time Protocol?

- Synchronize time between many computers
  - Multicast
  - Broadcast
- What is UTC?
  - Combination of time estimates
- NTP Applications

What is Network Time Protocol?

- NTP terms
  - Stratum-1 Server
  - Drift file
  - `xntpd`
  - `ntp.conf`

What is Network Time Protocol?

- Defining an NTP environment
- How does NTP work?
  - `ntp.conf` file
  - Both server and client broadcast or multicast
  - Local time is included with broadcast / multicast
  - Takes up to five minutes to update time

What is Network Time Protocol?

- NTP daemon
- Configuring NTP
  - Configuring an NTP Server
  - Configuring an NTP Client

Logging and Daemon Control

- Viewing NTP `syslog` logs
  - `/var/adm/messages`
- Starting and Stopping the NTP daemon
  - `/etc/init.d/xntpd stop`
  - `/etc/init.d/xntpd start`

Monitoring Systems Running the `xntpd` Daemon

- `xntpdc` utility
  - `?`
  - `timerstats`
  - `host`
- `ntpq` utility
  - `?`
  - `peers`

Module 13

Network Troubleshooting

Overview

- Objectives
- Relevance

Troubleshooting

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

Using `ping` as a Troubleshooting Tool

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

Using `ifconfig` as a Troubleshooting Tool

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

Using `arp` as a Troubleshooting Tool

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

Using snoop as a Troubleshooting Tool

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

Using ndd as a Troubleshooting Tool

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

Using `netstat` as a Troubleshooting Tool

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

Using `traceroute` as a Troubleshooting Tool

- 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

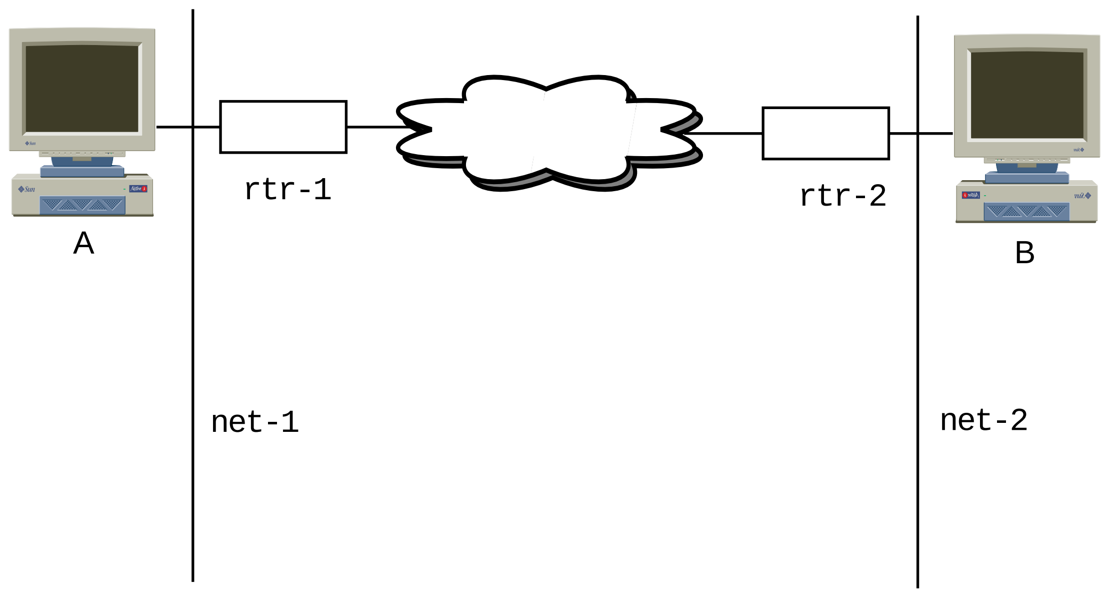
Troubleshooting Scenarios

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

Troubleshooting Scenarios

- Faulty cable
- Router log files
- Replace cable

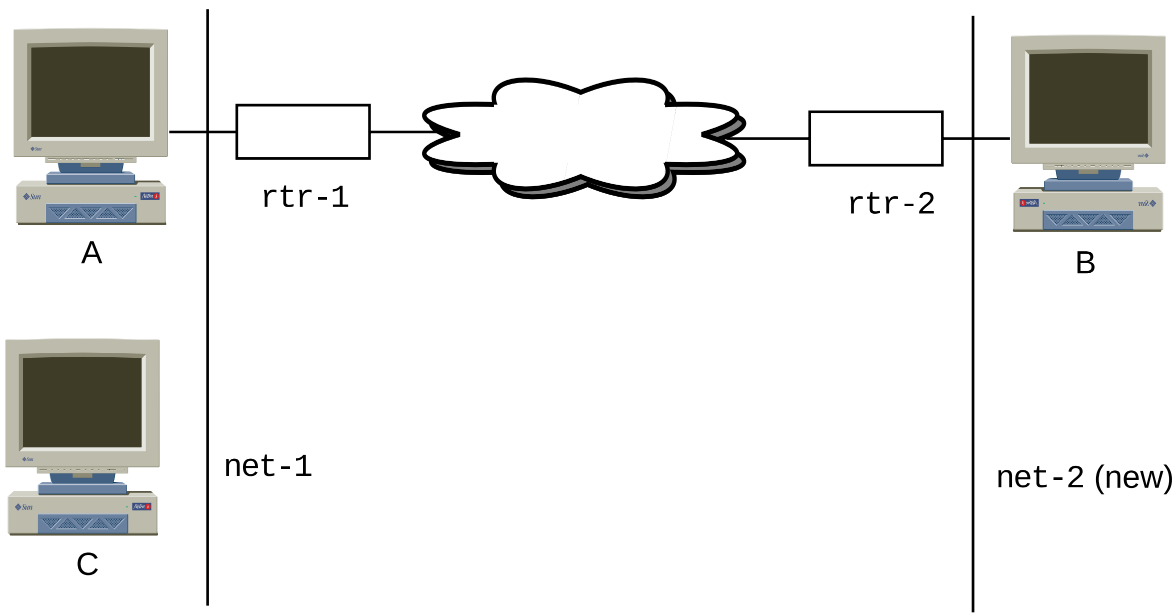
Faulty Cable Diagram



Troubleshooting Scenarios

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

Duplicate IP Address



Module 14

Introduction to IPv6

Overview

- Objectives
- Relevance

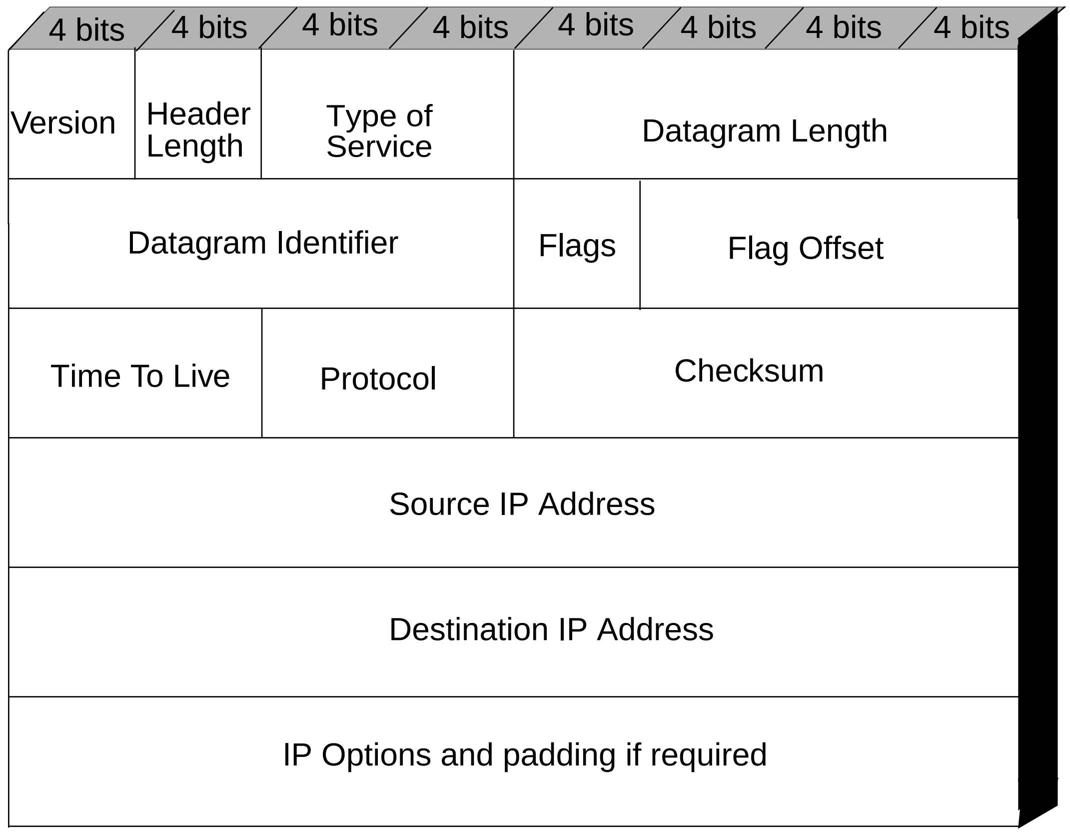
IPv6 History

- IPv6 history
- Why use IPv6?
  - Autoconfiguration
  - 128 bit address supports 340,282,366,920,938,463,463,347,607,431,768,211,456 nodes
  - Simplified headers
  - No router fragmentation

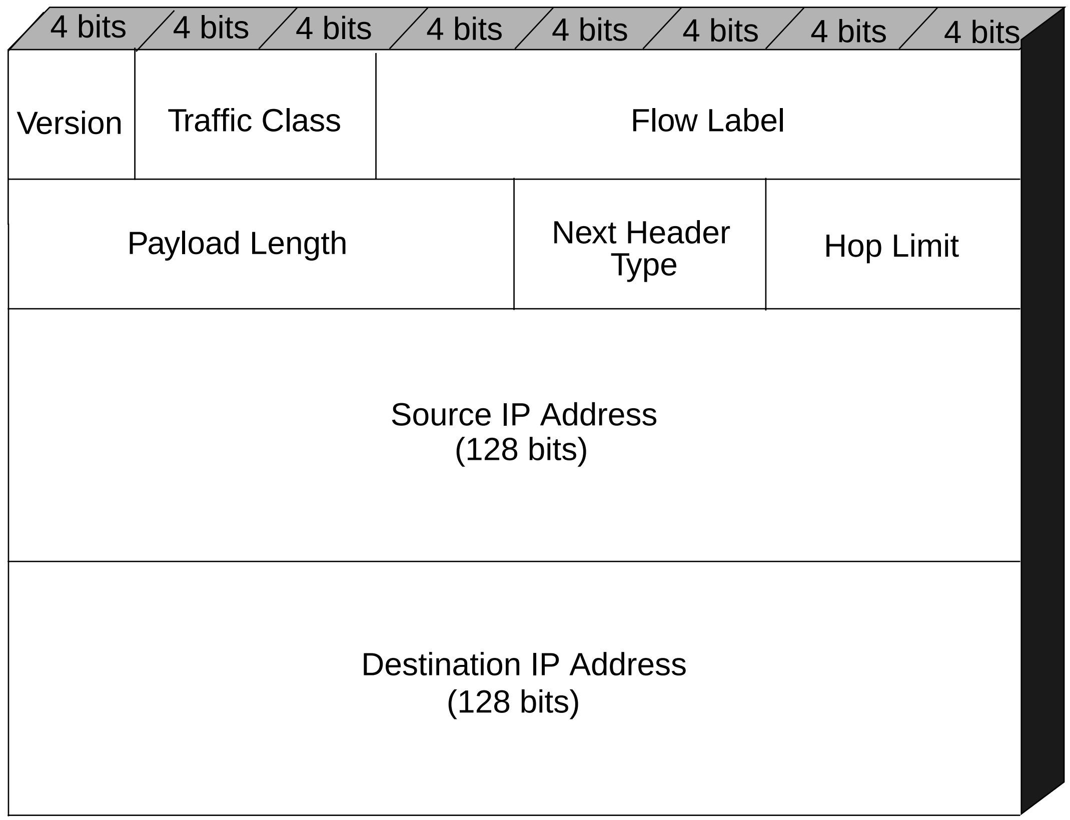
Features of IPv6

- More available addresses
- Simpler headers
  - Less load on routers
- Quality of service
- Compare an IPv4 header with an IPv6 header

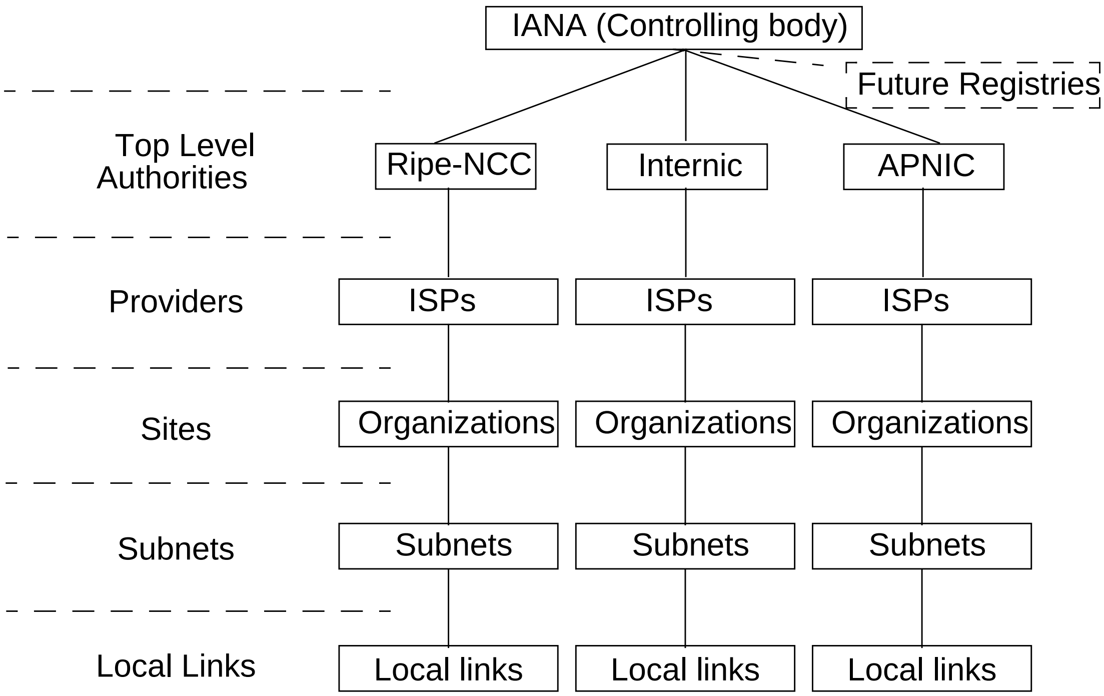
IPv4 Header


4 bits
4 bits
4 bits
4 bits
4 bits
4 bits
4 bits
4 bits
Version
Header Length
Type of Service
Datagram Length
Datagram Identifier
Flags
Flag Offset
Time To Live
Protocol
Checksum
Source IP Address
Destination IP Address
IP Options and padding if required

IPv6 Header


4 bits
4 bits
4 bits
4 bits
4 bits
4 bits
4 bits
4 bits
Version
Traffic Class
Flow Label
Payload Length
Next Header Type
Hop Limit
Source IP Address (128 bits)
Destination IP Address (128 bits)

IPv6 Addressing Hierarchy



IANA – Internet assigned numbers authority

Ripe-NCC – Réseaux IP Européens Network Coordination Centre

Internic – Internet Network Information Center

APNIC – Asian Pacific Network Information Center

ISPs – Internet Service Providers

IPv6 Autoconfiguration

- Stateful autoconfiguration
  - Requires configuration server such as DHCP
- Stateless autoconfiguration
  - No DHCP required
  - Only for link-local addresses

IPv6 Autoconfiguration

- Duplicate address detection
- Router detection

Autoconfiguration Address Calculation Example

48-bit MAC address – 08 : 00 : 20 : B5 : 41 : 37

0000 1000 0000 0000 0010 0000 1011 0101 0100 0001 0011 0111

- Toggle bit seven

0000 10**1**0 0000 0000 0010 0000 1011 0101 0100 0001 0011 0111

- Add two octets 0xFF and 0xFE

0000 1010 0000 0000 0010 0000 **1111 1111 1111 1110** 1011 0101 0100 0001 0011 0111

- Convert to hexadecimal and add colons

0A00 : 20FF : FEB5 : 4137

IPv6 Addressing

- Representing address types
  - Link-local – FE8
  - Site-local – FEC
  - Multicast – FF

Initial Allocation of FPs from RFC 2373

Allocation	FP (binary)	FP (hexadecimal)	Fraction of Address Space
Reserved	0000 0000	00	1/256
Aggregatable Global Unicast Addresses	001	2	1/8
Link-Local Unicast Addresses	1111 1110 10	FE8	1/1024
Site-Local Unicast Addresses	1111 1110 11	FEC	1/1024
Multicast Addresses	1111 1111	FF	1/256

IPv6 Addressing

- Mixing IPv4 and IPv6 Addresses
  - ::192.168.20.135
- Prefixing addresses and IPv6 subnetting
  - 12AB:0000:0000:CD30:1234:ABCD:56AE:1234/60

IPv6 Addressing

- Three address types
  - Unicast
  - Anycast
  - Multicast
- Compressing addresses
  - FF01::101

IPv6 Addressing

- Multicast address types
  - Multicast flags
  - Multicast scope
  - Multicast addresses

Multicast Addresses

- FF01:0:0:0:0:0:0:1 – Node-local nodes
- FF02:0:0:0:0:0:0:1 – Link-local nodes
- FF01:0:0:0:0:0:0:2 – Node-local routers
- FF02:0:0:0:0:0:0:2 – Link-local routers
- FF05:0:0:0:0:0:0:2 – Site-local routers
- FF02:0:0:0:0:0:0:9 – RIP Routers

Internet Layer

- Affected IPv4 Internet protocols
- ICMPv6
  - 14 new messages

IPv6 ICMP Types

Type	Meaning
1	Destination Unreachable
2	Packet Too Big
3	Time Exceeded
4	Parameter Problem
128	Echo Request
129	Echo Reply
130	Group Membership Query
131	Group Membership Report
132	Group Membership Reduction
133	Router Solicitation
134	Router Advertisement
135	Neighbor Solicitation
136	Neighbor Advertisement
137	Redirect

Internet Layer

- Internet Group Management Protocol (IGMP)
- ARP and RARP
- Neighbor Discovery Protocol

Neighbor Discovery and ICMP

- Router solicitation
- Router advertisement
- Neighbor solicitation
- Neighbor advertisement
- Redirect

Unicast Address Allocation Scheme

- Unspecified addresses – ::
- Loopback addresses – ::1
- Embedded IPv4 addresses
  - IPv4-compatible IPv6 addresses – ::192.168.20.135
  - IPv4-mapped IPv6 addresses – ::FFFF:192.168.20.135

Using the Dual-stack Approach in IPv6

- Enabling IPv6
- IPv6 files
- Configuring IPv6
  - NIS
  - NIS+
  - DNS
  - The `nsswitch.conf` file

Using the `netstat` Utility

- `netstat -f inet6`
- `netstat -ia`
- `netstat -rn -f inet6`

Using the `ifconfig` Utility

- `ifconfig hme0 inet6`
- Configuring Logical IPv6 Interfaces
  - `ifconfig hme0:1 inet6 plumb up`
  - `ifconfig hme0:1 inet6 down unplumb`

Routing IPv6

- Similar to routing IPv4 CIDR
- Routing daemons
  - `in.ripngd`
  - `in.ndpd`
- `/etc/inet/ndpd.conf`

Course Contents