Internetworking With TCP/IP

Douglas Comer

Computer Science Department
Purdue University
250 N. University Street
West Lafayette, IN 47907-2066

http://www.cs.purdue.edu/people/comer

PART I

COURSE OVERVIEW AND INTRODUCTION

Topic And Scope

Internetworking: an overview of concepts, terminology, and technology underlying the TCP/IP Internet protocol suite and the architecture of an internet.

You Will Learn

- Terminology (including acronyms)
- Concepts and principles
  - The underlying model
  - Encapsulation
  - End-to-end paradigm
- Naming and addressing
- Functions of protocols including ARP, IP, TCP, UDP, SMTP, FTP, DHCP, and more
- Layering model

You Will Learn
(continued)

- Internet architecture and routing
- Applications

What You Will NOT Learn

- A list of vendors, hardware products, software products, services, comparisons, or prices
- Alternative internetworking technologies (they have all disappeared!)

Schedule Of Topics

- Introduction
- Review of
  - Network hardware
  - Physical addressing
- Internet model and concept
- Internet (IP) addresses
- Higher-level protocols and the layering principle
- Examples of internet architecture

Schedule Of Topics
(continued)

- Routing update protocols
- Application-layer protocols

Why Study TCP/IP?

- The Internet is everywhere
- Most applications are distributed

Remainder Of This Section

- History of Internet protocols (TCP/IP)
- Organizations
- Documents

Vendor Independence

- Before TCP/IP and the Internet
  - Only two sources of network protocols
    * Specific vendors such as IBM or Digital Equipment
    * Standards bodies such as the ITU (formerly known as CCITT)
- TCP/IP
  - Vendor independent

Who Built TCP/IP?

- Internet Architecture Board (IAB)
- Originally known as *Internet Activities Board*
- Evolved from Internet Research Group
- Forum for exchange among researchers
- About a dozen members
- Reorganized in 1989 and 1993
- Merged into the Internet Society in 1992

Components Of The IAB Organization

- IAB (Internet Architecture Board)
  - Board that oversees and arbitrates
  - URL is

    http://www.iab.org/iab

- IRTF (Internet Research Task Force)
  - Coordinates research on TCP/IP and internetworking
  - Virtually defunct, but may re-emerge

Components Of The IAB Organization (continued)

- IETF (Internet Engineering Task Force)
  - Coordinates protocol and Internet engineering
  - Headed by Internet Engineering Steering Group (IESG)
  - Divided into *N* areas (*N* is 10 plus or minus a few)
  - Each area has a manager
  - Composed of working groups (volunteers)
  - URL is

http://www.ietf.org

ICANN

- Internet Corporation for Assigned Names and Numbers

  http://www.icann.org

- Formed in 1998 to subsume IANA contract
- Not-for-profit managed by international board
- Now sets policies for addresses and domain names
- Support organizations
  - Address allocation (ASO)
  - Domain Names (DNSO)
  - Protocol parameter assignments (PSO)

ICANN

- Internet Corporation for Assigned Names and Numbers

  http://www.icann.org

- Formed in 1998 to subsume IANA contract
- Not-for-profit managed by international board
- Now sets policies for addresses and domain names
- Support organizations
  - Address allocation (ASO)
  - Domain Names (DNSO)
  - Protocol parameter assignments (PSO)
- For fun see http://www.icannwatch.org

World Wide Web Consortium

- Organization to develop common protocols for World Wide Web
- Open membership
- Funded by commercial members
- URL is

http://w3c.org

Internet Society

- Organization that promotes the use of the Internet
- Formed in 1992
- Not-for-profit
- Governed by a board of trustees
- Members worldwide
- URL is

http://www.isoc.org

Protocol Specifications And Documents

- Protocols documented in series of reports
- Documents known as *Request For Comments* (*RFCs*)

RFCs

- Series of reports that include
  - TCP/IP protocols
  - The Internet
  - Related technologies
- Edited, but not peer-reviewed like scientific journals
- Contain:
  - Proposals
  - Surveys and measurements
  - Protocol standards

RFCs

- Series of reports that include
  - TCP/IP protocols
  - The Internet
  - Related technologies
- Checked and edited by IESG
- Contain:
  - Proposals
  - Surveys and measurements
  - Protocol Standards
  - Jokes!

RFCs
(continued)

- Numbered in chronological order
- Revised document reissued under new number
- Numbers ending in 99 reserved for summary of previous 100 RFCs
- Index and all RFCs available on-line

Requirements RFCs

- Host Requirements Documents
  - Major revision/clarification of most TCP/IP protocols
  - RFC 1122 (Communication Layers)
  - RFC 1123 (Application & Support)
  - RFC 1127 (Perspective on 1122-3)
- Router Requirements
  - Major specification of protocols used in IP gateways (routers)
  - RFC 1812 (updated by RFC 2644)

Special Subsets Of RFCs

- For Your Information (FYI)
  - Provide general information
  - Intended for beginners
- Best Current Practices (BCP)
  - Engineering hints
  - Reviewed and approved by IESG

A Note About RFCs

- RFCs span two extremes
  - Protocol standards
  - Jokes
- Question: how does one know which are standards?

TCP/IP Standards (STD)

- Set by vote of IETF
- Documented in subset of RFCs
- Found in *Internet Official Protocol Standards* RFC and on IETF web site
  - Issued periodically
  - Current version is RFC 3600

Internet Drafts

- Preliminary RFC documents
- Often used by IETF working groups
- Available on-line from several repositories
- Either become RFCs within six months or disappear

Obtaining RFCs And Internet Drafts

- Available via
  - Email
  - FTP
  - World Wide Web

    http://www.ietf.org/
- IETF report contains summary of weekly activity

  http://www.isoc.org/ietfreport/

Summary

- TCP/IP is vendor-independent
- Standards set by IETF
- Protocol standards found in document series known as *Request For Comments* (*RFCs*)
- Standards found in subset of RFCs labeled STD


Questions?

PART II

REVIEW OF NETWORK HARDWARE AND PHYSICAL ADDRESSING

The TCP/IP Concept

- Use existing network hardware
- Interconnect networks
- Add abstractions to hide heterogeneity

The Challenge

- Accommodate *all* possible network hardware
- Question: what kinds of hardware exist?

Network Hardware Review

- We will
  - Review basic network concepts
  - Examine example physical network technologies
  - Introduce physical (hardware) addressing

Two Basic Categories Of Network Hardware

- Connection oriented
- Connectionless

Connection Oriented (Circuit Switched Technology)

- Paradigm
  - Form a ‘‘connection’’ through the network
  - Send / receive data over the connection
  - Terminate the connection
- Can guarantee bandwidth
- Proponents argue that it works well with real-time applications
- Example: ATM network

Connectionless
(Packet Switched Technology)

- Paradigm
  - Form ‘‘packet’’ of data
  - Pass to network
- Each packet travels independently
- Packet includes identification of the destination
- Each packet can be a different size
- The maximum packet size is fixed (some technologies limit packet sizes to 1,500 octets or less)

Broad Characterizations Of Packet Switching Networks

- Local Area Network (LAN)
- Wide Area Network (WAN)
- Categories are informal and qualitative

Local Area Networks

- Engineered for
  - Low cost
  - High capacity
- Direct connection among computers
- Limited distance

Wide Area Networks
(Long Haul Networks)

- Engineered for
  - Long distances
  - Indirect interconnection via special-purpose hardware
- Higher cost
- Lower capacity (usually)

Examples Of Packet Switched Networks

- Wide Area Nets
  - ARPANET, NSFNET, ANSNET
  - Common carrier services
- Leased line services
  - Point-to-point connections
- Local Area Nets
  - Ethernet
  - Wi-Fi

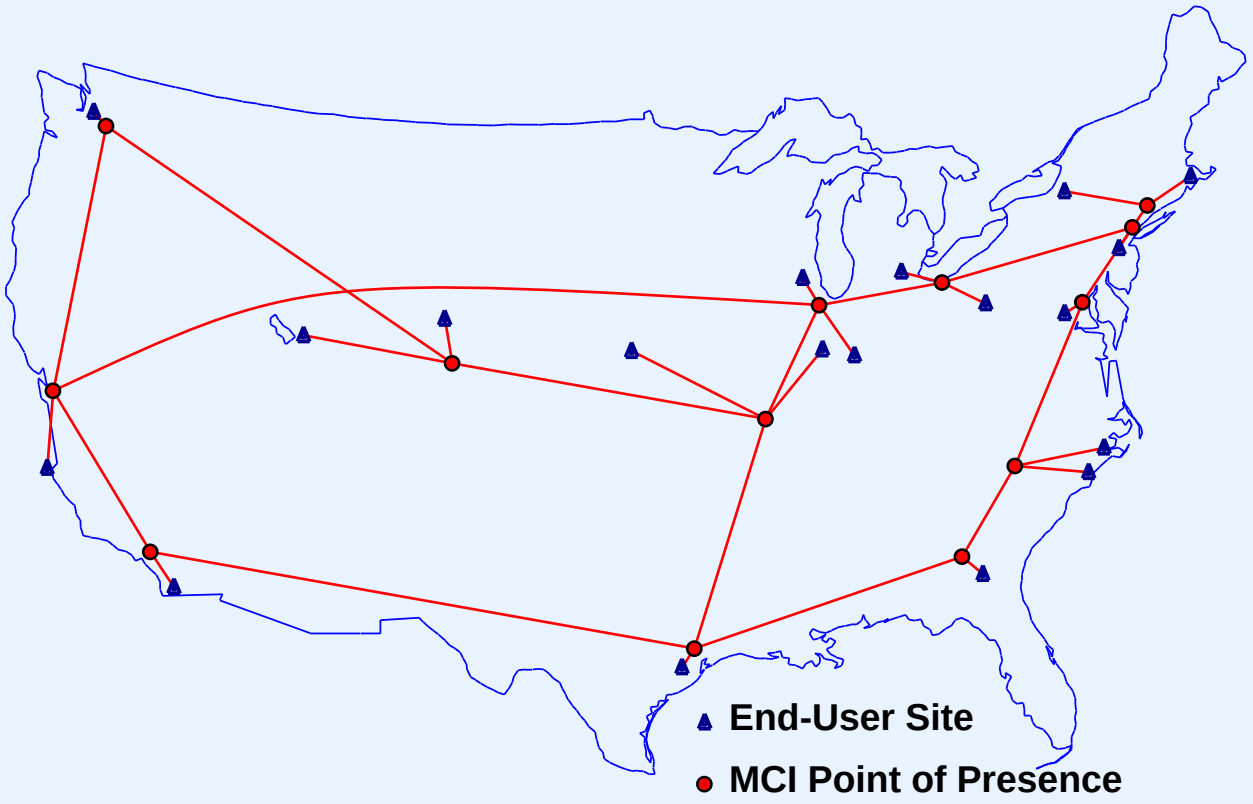
ARPANET (1969-1989)

- Original backbone of Internet
- Wide area network around which TCP/IP was developed
- Funding from Advanced Research Project Agency
- Initial speed 50 Kbps

NSFNET (1987-1992)

- Funded by National Science Foundation
- Motivation: Internet backbone to connect all scientists and engineers
- Introduced Internet hierarchy
  - Wide area backbone spanning geographic U.S.
  - Many mid-level (regional) networks that attach to backbone
  - Campus networks at lowest level
- Initial speed 1.544 Mbps

ANSNET (1992-1995)



- Backbone of Internet before commercial ISPs
- Typical topology

Wide Area Networks Available From Common Carriers

- Point-to-point digital circuits
  - T-series (e.g., T1 = 1.5 Mbps, T3 = 45 Mbps)
  - OC-series (e.g., OC-3 = 155 Mbps, OC-48 = 2.4 Gbps)
- Packet switching services also available
  - Examples: ISDN, SMDS, Frame Relay, ATM

Example Local Area Network: Ethernet

- Extremely popular
- Can run over
  - Copper (twisted pair)
  - Optical fiber
- Three generations
  - *10Base-T* operates at 10 Mbps
  - *100Base-T* (fast Ethernet) operates at 100 Mbps
  - *1000Base-T* (gigabit Ethernet) operates at 1 Gbps
- IEEE standard is 802.3

Ethernet Frame Format

- Header format fixed (Destination, Source, Type fields)
- Frame data size can vary from packet to packet
  - Maximum 1500 octets
  - Minimum 46 octets
- Preamble and CRC removed by framer hardware before frame stored in computer's memory

Example Ethernet Frame In Memory

02	07	01	00	27	ba	08	00	2b	0d	44	a7	08	00	45	00
00	54	82	68	00	00	ff	01	35	21	80	0a	02	03	80	0a
02	08	08	00	73	0b	d4	6d	00	00	04	3b	8c	28	28	20
0d	00	08	09	0a	0b	0c	0d	0e	0f	10	11	12	13	14	15
16	17	18	19	1a	1b	1c	1d	1e	1f	20	21	22	23	24	25
26	27	28	29	2a	2b	2c	2d	2e	2f	30	31	32	33	34	35
36	37														

- Octets shown in hexadecimal
- Destination is 02.07.01.00.27.ba
- Source is 08.00.2b.0d.44.a7
- Frame type is 08.00 (IP)

Point-to-Point Network

- Any direct connection between two computers
  - Leased line
  - Connection between two routers
  - Dialup connection
- Link-level protocol required for framing
- TCP/IP views as an independent network

Note: some pundits argue the terminology is incorrect because a connection limited to two endpoints is not technically a ‘‘network’’

Hardware Address

- Unique number assigned to each machine on a network
- Used to identify destination for a packet

Hardware Address Terminology

- Known as
  - MAC (Media Access Control) address
  - Physical address
  - Hardware unicast address
- Hardware engineers assign fine distinctions to the above terms
- We will treat all terms *equally*

Use Of Hardware Address

- Sender supplies
  - Destination's address
  - Source address (in most technologies)
- Network hardware
  - Uses destination address to forward packet
  - Delivers packet to proper machine.
- Important note: each technology defines its own addressing scheme

Three Types Of Hardware Addressing Schemes

- Static
  - Address assigned by hardware vendor
- Configurable
  - Address assigned by customer
- Dynamic
  - Address assigned by software at startup

Examples Of Hardware Address Types

- Configurable: proNET-10 (Proteon)
  - 8-bit address per interface card
  - All 1s address reserved for broadcast
  - Address assigned by customer when device installed
- Dynamic MAC addressing: LocalTalk (Apple)
  - Randomized bidding
  - Handled by protocols in software

Examples Of Hardware Address Types (continued)

- Static MAC addressing: Ethernet
  - 48-bit address
  - Unicast address assigned when device manufactured
  - All 1s address reserved for broadcast
  - One-half address space reserved for multicast (restricted form of broadcast)
- Ethernet's static addressing is now most common form

Bridge

- Hardware device that connects multiple LANs and makes them appear to be a single LAN
- Repeats all packets from one LAN to the other and vice versa
- Introduces delay of 1 packet-time
- Does not forward collisions or noise
- Called *Layer 2 Interconnect* or *Layer 2 forwarder*
- Makes multiple LANs appear to be a single, large LAN
- Often embedded in other equipment (e.g., DSL modem)

Bridge
(continued)

- Watches packets to learn which computers are on which side of the bridge
- Uses hardware addresses to filter

Layer 2 Switch

- Electronic device
- Computers connect directly
- Applies bridging algorithm
- Can separate computers onto virtual networks (*VLAN switch*)

Physical Networks As Viewed By TCP/IP

- TCP/IP protocols accommodate
  - Local Area Network
  - Wide Area Network
  - Point-to-point link
  - Set of bridged LANs

The Motivation For Heterogeneity

- Each network technology has advantages for some applications
- Consequence: an internet may contain combinations of technologies

Heterogeneity And Addressing

- Recall: each technology can define its own addressing scheme
- Heterogeneous networks imply potential for heterogeneous addressing
- Conclusion: cannot rely on hardware addressing

Summary

- TCP/IP is designed to use all types of networks
  - Connection-oriented
  - Connectionless
  - Local Area Network (LAN)
  - Wide Area Network (WAN)
  - Point-to-point link
  - Set of bridged networks

Summary
(continued)

- Each technology defines an addressing scheme
- TCP/IP must accommodate heterogeneous addressing schemes


Questions?

PART III

INTERNETWORKING CONCEPT AND ARCHITECTURAL MODEL

Accommodating Heterogeneity

- Approach 1
  - Application gateways
  - Gateway forwards data from one network to another
  - Example: file transfer gateway
- Approach 2
  - Network-level gateways
  - Gateway forwards individual packets
- Discussion question: which is better?

Desired Properties

- Universal service
- End-to-end connectivity
- Transparency

Agreement Needed To Achieve Desired Properties

- Data formats
- Procedures for exchanging information
- Identification
  - Services
  - Computers
  - Applications
- Broad concepts: naming and addressing

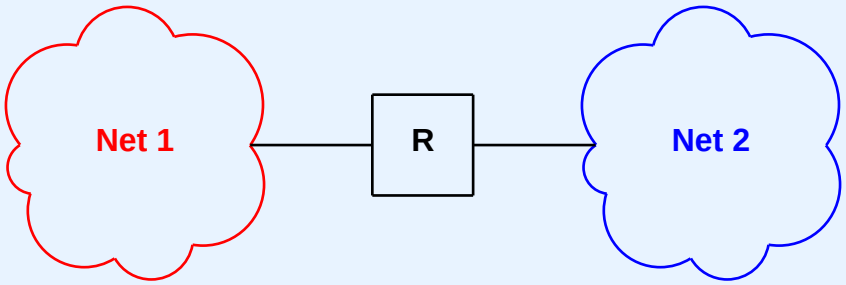
The TCP/IP Internet Concept

- Use available networks
- Interconnect physical networks
  - Network of networks
  - Revolutionary when proposed
- Devise abstractions that hide
  - Underlying architecture
  - Hardware addresses
  - Routes

Network Interconnection

- Uses active system
- Each network sees an additional computer attached
- Device is *IP router* (originally called *IP gateway*)

Illustration Of Network Interconnection

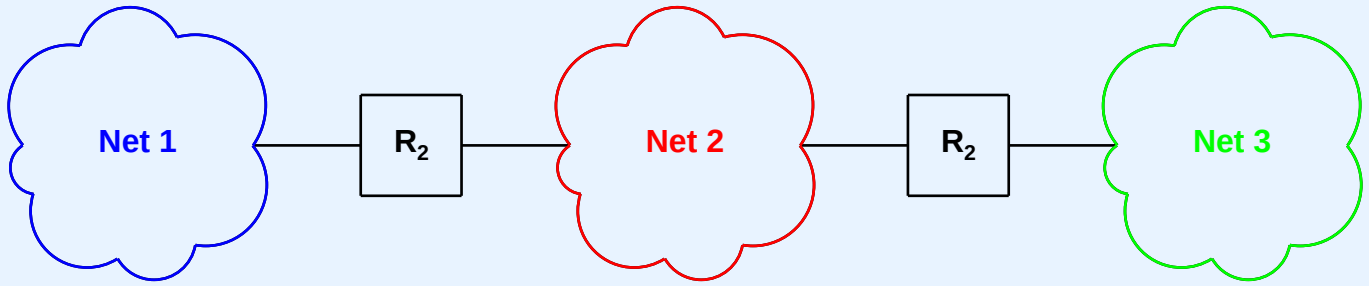


- Network technologies can differ
  - LAN and WAN
  - Connection-oriented and connectionless

Building An Internet

- Use multiple IP routers
- Ensure that each network is reachable
- Do not need router between each pair of networks

Example Of Multiple Networks



- Networks can be heterogeneous
- No direct connection from network 1 to network 3

Physical Connectivity

In a TCP/IP internet, special computers called IP routers *or* IP gateways *provide interconnections among physical networks.*

Packet Transmission Paradigm

- Source computer
  - Generates a packet
  - Sends across one network to a router
- Intermediate router
  - Forwards packet to ‘‘next’’ router
- Final router
  - Delivers packet to destination

An Important Point About Forwarding

Routers use the destination network, not the destination computer, when forwarding packets.

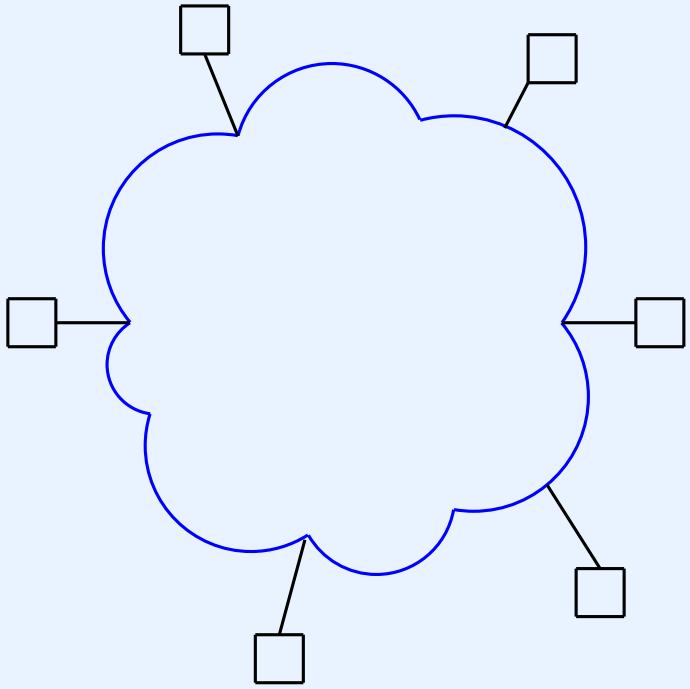
Equal Treatment

The TCP/IP internet protocols treat all networks equally. A Local Area Network such as an Ethernet, a Wide Area Network used as a backbone, or a point-to-point link between two computers each count as one network.

User's View Of Internet

- Single large (global) network
- User's computers all attach directly
- No other structure visible

Illustration Of User's View Of A TCP/IP Internet

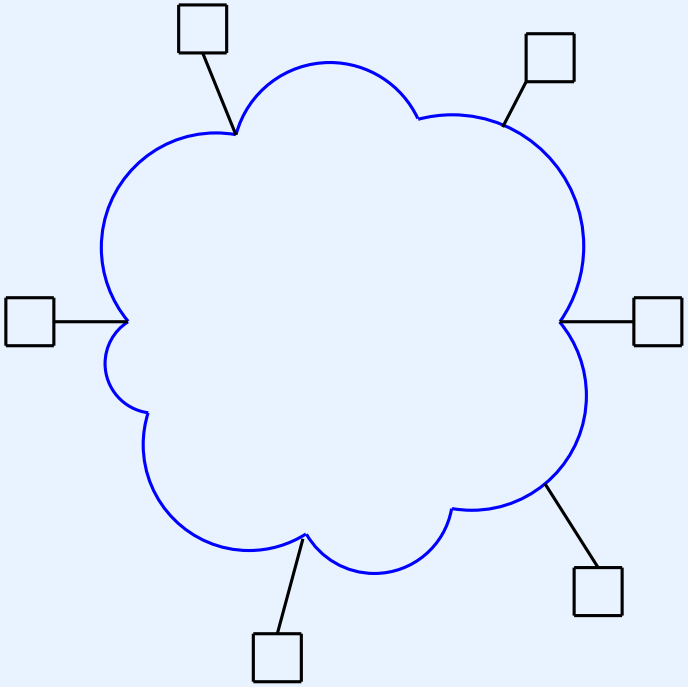


user's view

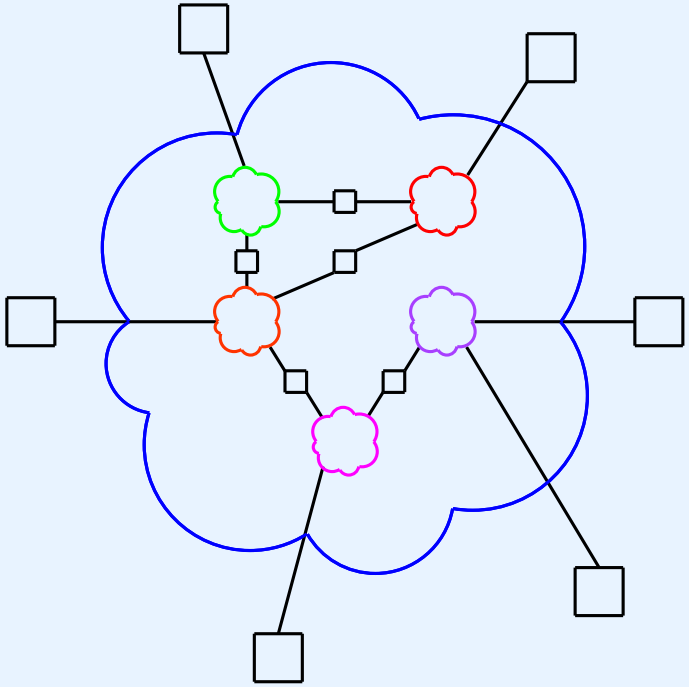
Actual Internet Architecture

- Multiple physical networks interconnected
- Each host attaches to one network
- Single *virtual* network achieved through software that implements abstractions

The Two Views Of A TCP/IP Internet



user's view



actual connections

Architectural Terminology

- End-user system is called *host* computer
  - Connects to physical network
  - Possibly many hosts per network
  - Possibly more than one network connection per host
- Dedicated systems called *IP gateways* or *IP routers* interconnect networks
  - Router connects two or more networks

Many Unanswered Questions

- Addressing model and relationship to hardware addresses
- Format of packet as it travels through Internet
- How a host handles concurrent communication with several other hosts

Summary

- Internet is set of interconnected (possibly heterogeneous) networks
- Routers provide interconnection
- End-user systems are called host computers
- Internetworking introduces abstractions that hide details of underlying networks


Questions?

PART IV

CLASSFUL INTERNET ADDRESSES

Definitions

- Name
  - Identifies *what* an entity is
  - Often textual (e.g., ASCII)
- Address
  - Identifies *where* an entity is located
  - Often binary and usually compact
  - Sometimes called locator
- Route
  - Identifies *how* to get to the object
  - May be distributed

Internet Protocol Address (IP Address)

- Analogous to hardware address
- Unique value assigned as unicast address to each host on Internet
- Used by Internet applications

IP Address Details

- 32-bit binary value
- Unique value assigned to each host in Internet
- Values chosen to make routing efficient

IP Address Division

- Address divided into two parts
  - Prefix (network ID) identifies network to which host attaches
  - Suffix (host ID) identifies host on that network

Classful Addressing

- Original IP scheme
- Explains many design decisions
- New schemes are backward compatible

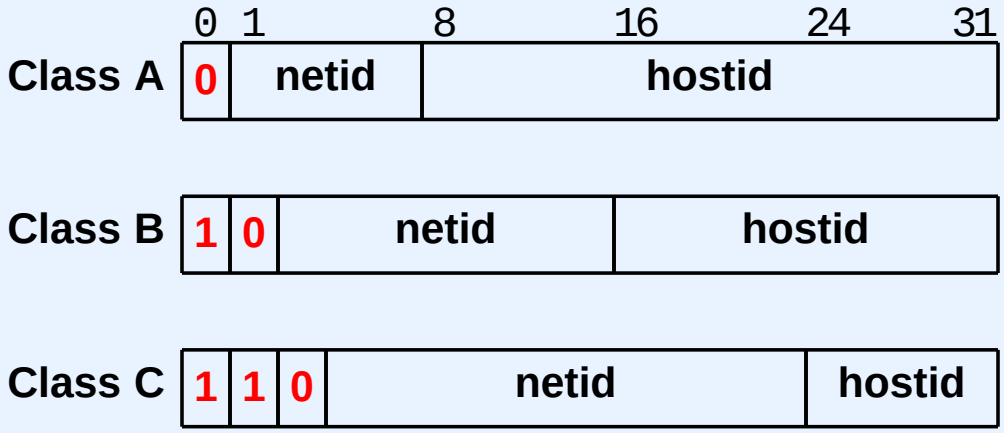
Desirable Properties Of An Internet Addressing Scheme

- Compact (as small as possible)
- Universal (big enough)
- Works with all network hardware
- Supports efficient decision making
  - Test whether a destination can be reached directly
  - Decide which router to use for indirect delivery
  - Choose next router along a path to the destination

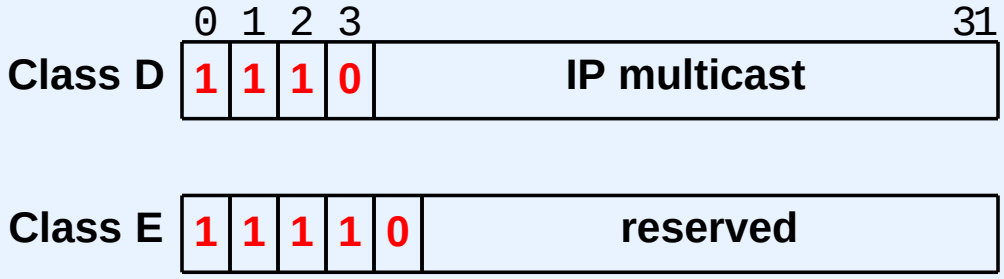
Division Of Internet Address Into Prefix And Suffix

- How should division be made?
  - Large prefix, small suffix means many possible networks, but each is limited in size
  - Large suffix, small prefix means each network can be large, but there can only be a few networks
- Original Internet address scheme designed to accommodate both possibilities
  - Known as *classful* addressing

Original IPv4 Address Classes



Three Principle Classes



Other (seldom used) Classes

Important Property

- Classful addresses are *self-identifying*
- Consequences
  - Can determine boundary between prefix and suffix from the address itself
  - No additional state needed to store boundary information
  - Both hosts and routers benefit

Endpoint Identification

Because IP addresses encode both a network and a host on that network, they do not specify an individual computer, but a connection to a network.

IP Address Conventions

- When used to refer to a network
  - Host field contains all *0* bits
- Broadcast on the local wire
  - Network and host fields both contain all *1* bits
- Directed broadcast: broadcast on specific (possibly remote) network
  - Host field contains all *1* bits
  - Nonstandard form: host field contains all *0* bits

Assignment Of IP Addresses

- All hosts on same network assigned same address prefix
  - Prefixes assigned by central authority
  - Obtained from ISP
- Each host on a network has a unique suffix
  - Assigned locally
  - Local administrator must ensure uniqueness

Advantages Of Classful Addressing

- Computationally efficient
  - First bits specify size of prefix / suffix
- Allows mixtures of large and small networks

Directed Broadcast

IP addresses can be used to specify a directed broadcast in which a packet is sent to all computers on a network; such addresses map to hardware broadcast, if available. By convention, a directed broadcast address has a valid netid and has a hostid with all bits set to 1.

Limited Broadcast

- All 1’s
- Broadcast limited to local network only (no forwarding)
- Useful for bootstrapping

All Zeros IP Address

- Can only appear as source address
- Used during bootstrap before computer knows its address
- Means ‘‘this’’ computer

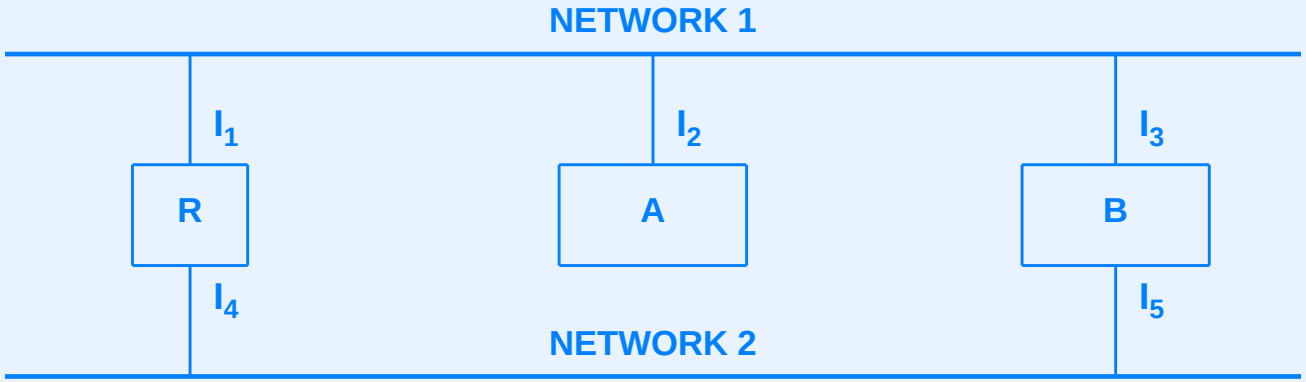
Internet Multicast

- IP allows Internet multicast, but no Internet-wide multicast delivery system currently in place
- Class D addresses reserved for multicast
- Each address corresponds to group of participating computers
- IP multicast uses hardware multicast when available
- More later in the course

Consequences Of IP Addressing

- If a host computer moves from one network to another, its IP address must change
- For a multi-homed host (with two or more addresses), the path taken by packets depends on the address used

Multi-Homed Hosts And Reliability



- Knowing that B is multi-homed increases reliability
- If interface I_3 is down, host A can send to the interface I_5

Dotted Decimal Notation

- Syntactic form for expressing 32-bit address
- Used throughout the Internet and associated literature
- Represents each octet in decimal separated by periods (dots)

Example Of Dotted Decimal Notation

- A 32-bit number in binary

  10000000 00001010 00000010 00000011

- The same 32-bit number expressed in dotted decimal notation

  128.10.2.3

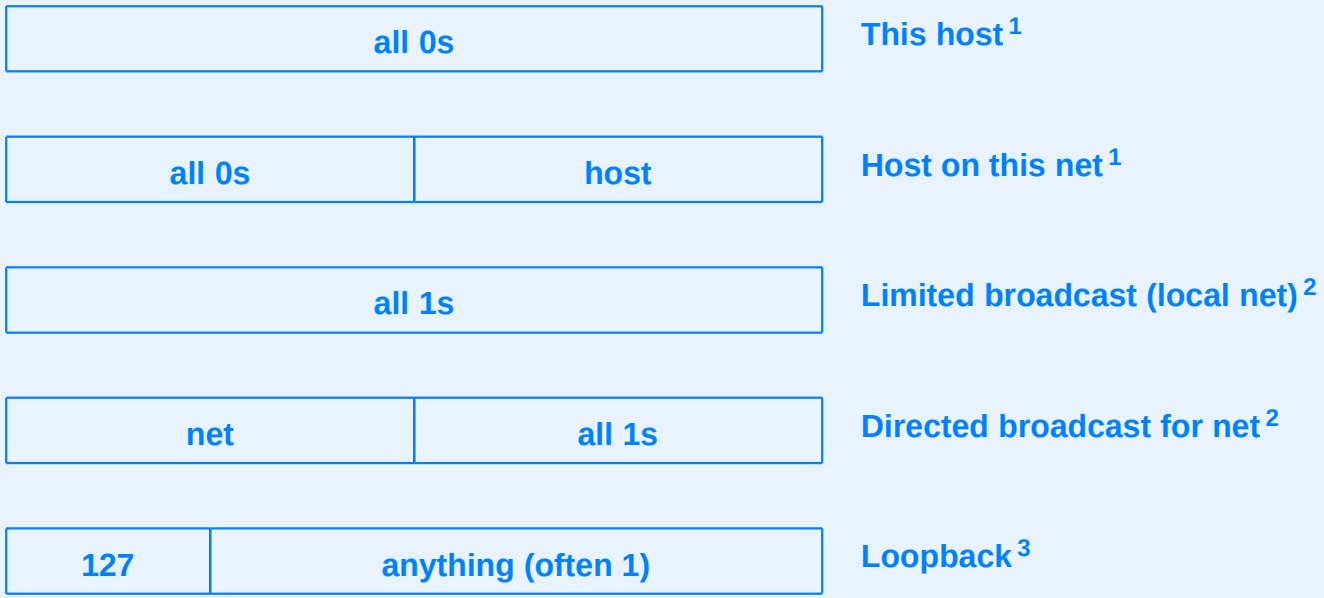
Loopback Address

- Used for testing
- Refers to local computer (never sent to Internet)
- Address is 127.0.0.1

Classful Address Ranges

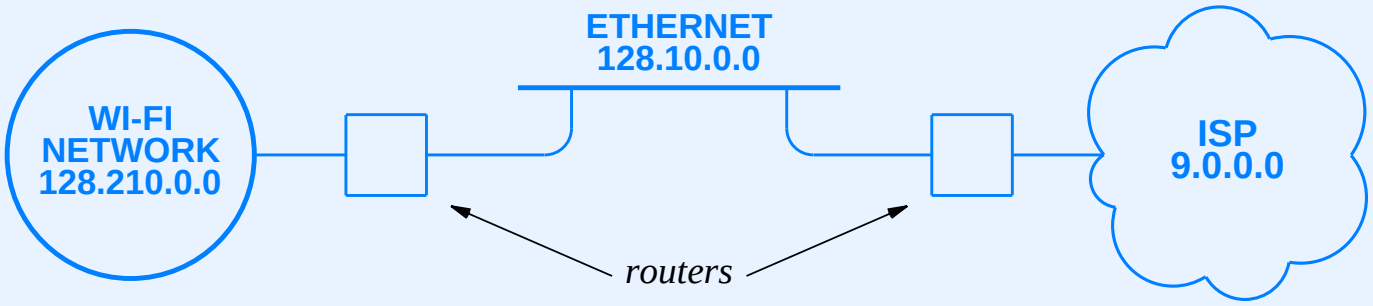
Class	Lowest Address	Highest Address
A	1.0.0.0	126.0.0.0
B	128.1.0.0	191.255.0.0
C	192.0.1.0	223.255.255.0
D	224.0.0.0	239.255.255.255
E	240.0.0.0	255.255.255.254

Summary Of Address Conventions

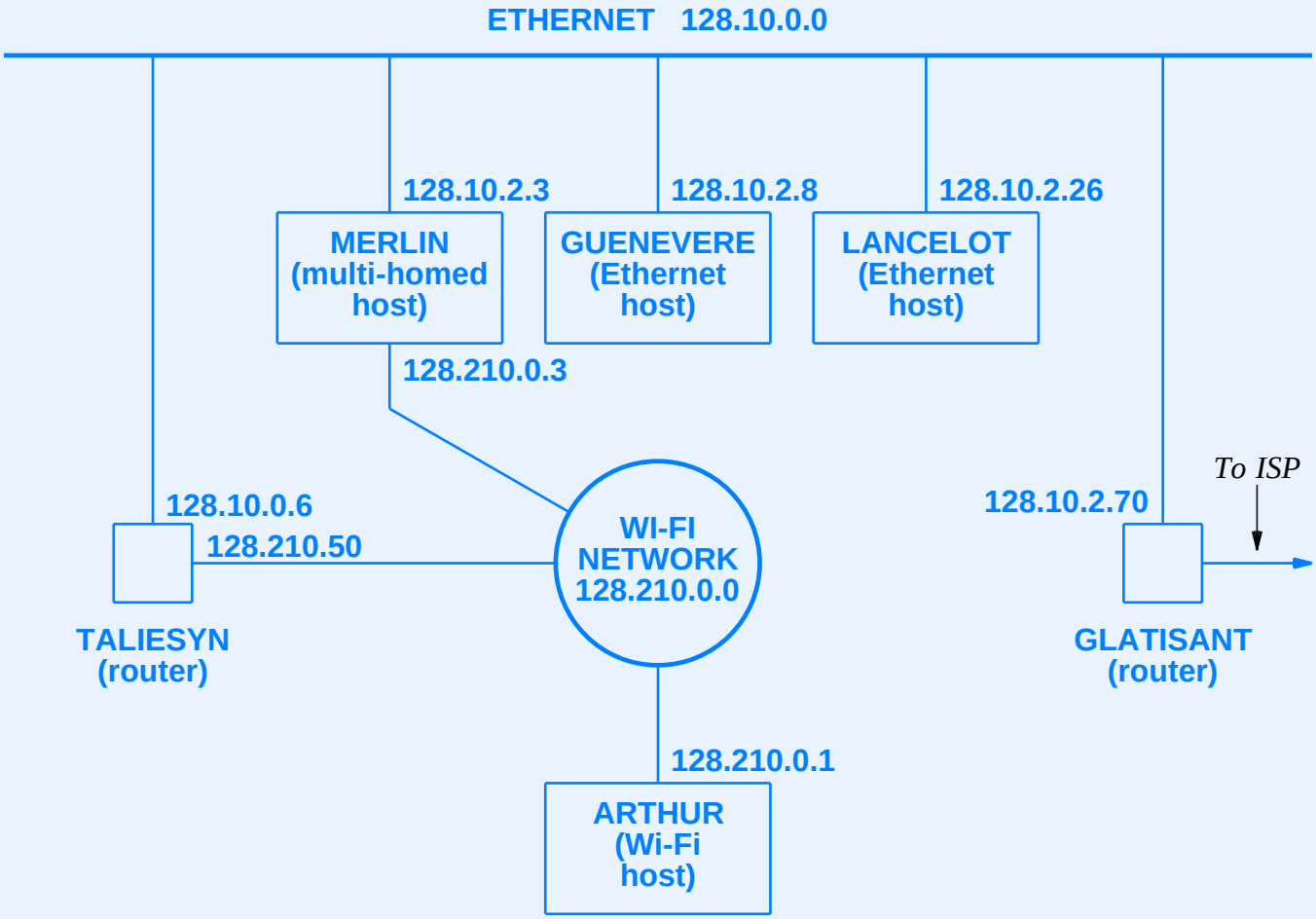


Notes: [1] Allowed only at system startup and is never a valid destination address.
[2] Never a valid source address.
[3] Should never appear on a network.

An Example Of IP Addresses



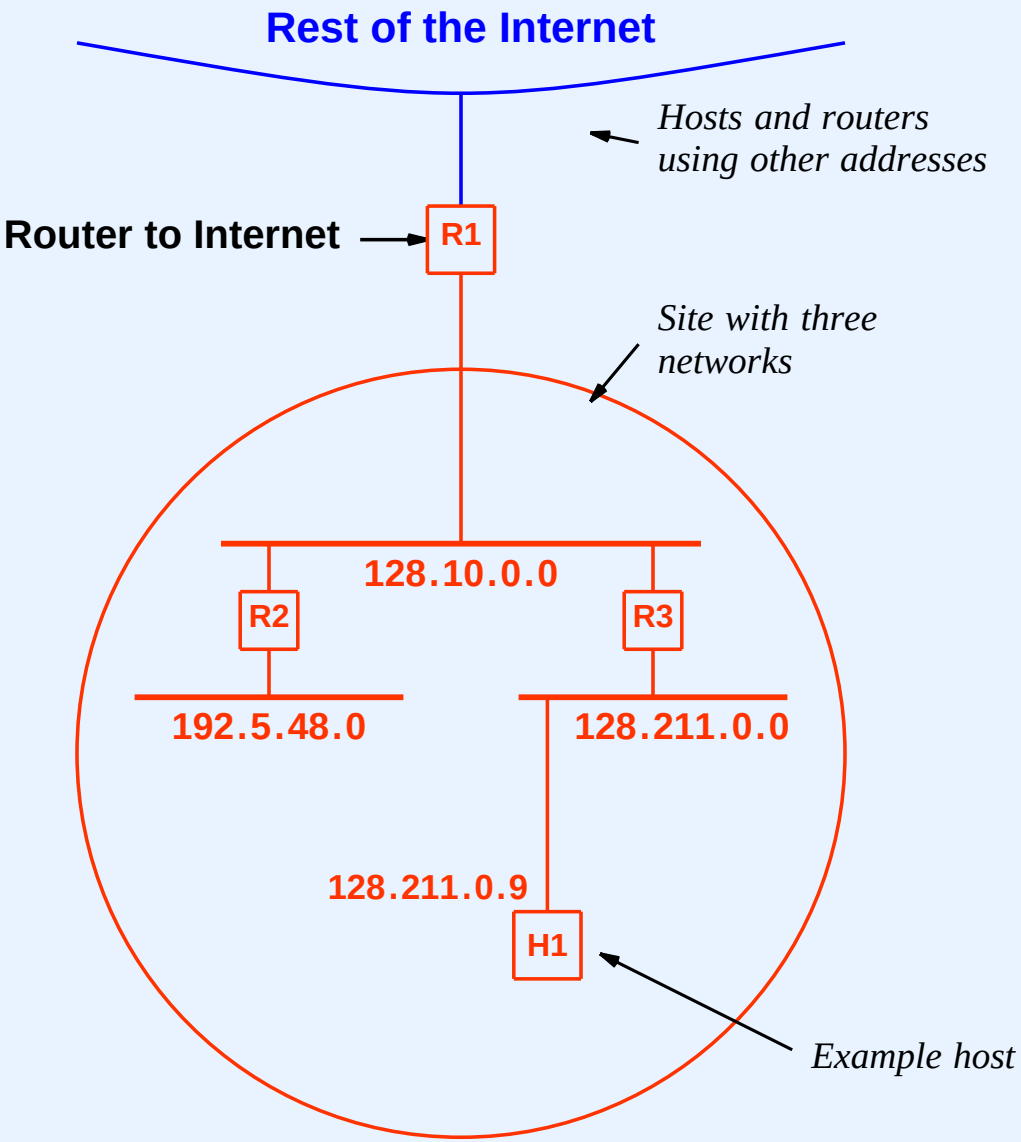
Example Host Addresses



Another Addressing Example

- Assume an organization has three networks
- Organization obtains three prefixes, one per network
- Host address must begin with network prefix

Illustration Of IP Addressing



Summary

- IP address
  - 32 bits long
  - Prefix identifies network
  - Suffix identifies host
- Classful addressing uses first few bits of address to determine boundary between prefix and suffix

Summary
(continued)

- Special forms of addresses handle
  - Limited broadcast
  - Directed broadcast
  - Network identification
  - This host
  - Loopback


Questions?

PART V

MAPPING INTERNET ADDRESSES TO PHYSICAL ADDRESSES (ARP)

Motivation

- Must use hardware (physical) addresses to communicate over network
- Applications only use Internet addresses

Example

- Computers A and B on same network
- Application on A generates packet for application on B
- Protocol software on A must use B's hardware address when sending a packet

Consequence

- Protocol software needs a mechanism that maps an IP address to equivalent hardware address
- Known as *address resolution* problem

Address Resolution

- Performed at each step along path through Internet
- Two basic algorithms
  - Direct mapping
  - Dynamic binding
- Choice depends on type of hardware

Direct Mapping

- Easy to understand
- Efficient
- Only works when hardware address is small
- Technique: assign computer an IP address that encodes the hardware address

Example Of Direct Mapping

- Hardware: proNet ring network
- Hardware address: 8 bits
- Assume IP address 192.5.48.0 (24-bit prefix)
- Assign computer with hardware address K an IP address 192.5.48.K
- Resolving an IP address means extracting the hardware address from low-order 8 bits

Dynamic Binding

- Needed when hardware addresses are large (e.g., Ethernet)
- Allows computer A to find computer B's hardware address
  - A starts with B's IP address
  - A knows B is on the local network
- Technique: broadcast query and obtain response
- Note: dynamic binding only used across one network at a time

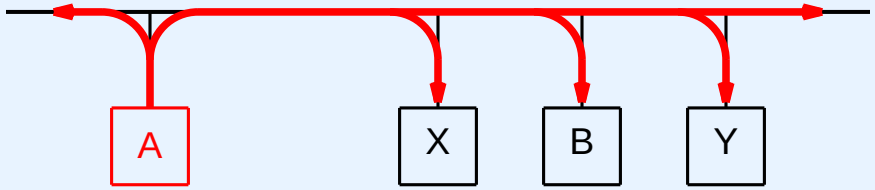
Internet Address Resolution Protocol (ARP)

- Standard for dynamic address resolution in the Internet
- Requires hardware broadcast
- Intended for LAN
- Important idea: ARP only used to map addresses within a single physical network, never across multiple networks

ARP

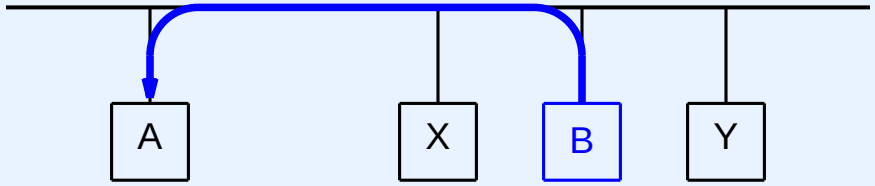
- Machine A broadcasts ARP request with B's IP address
- All machines on local net receive broadcast
- Machine B replies with its physical address
- Machine A adds B's address information to its table
- Machine A delivers packet directly to B

Illustration Of ARP Request And Reply Messages



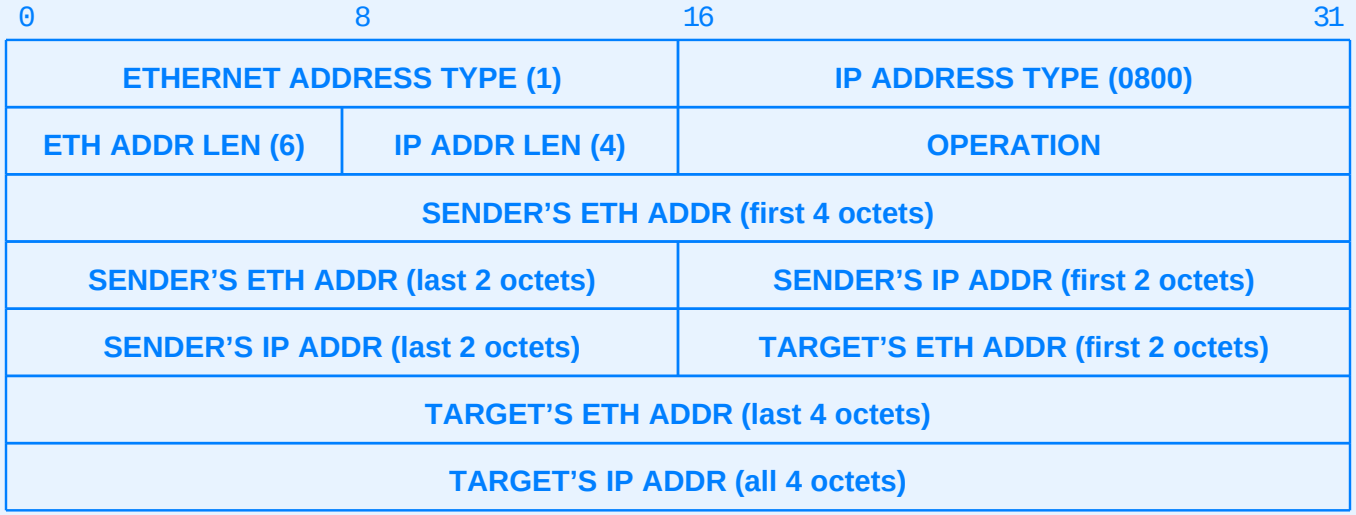
A broadcasts request for B

(across local net only)



B replies to request

ARP Packet Format When Used With Ethernet



Observations About Packet Format

- General: can be used with
  - Arbitrary hardware address
  - Arbitrary protocol address (not just IP)
- Variable length fields (depends on type of addresses)
- Length fields allow parsing of packet by computer that does not understand the two address types

Retention Of Bindings

- Cannot afford to send ARP request for each packet
- Solution
  - Maintain a table of bindings
- Effect
  - Use ARP one time, place results in table, and then send many packets

ARP Caching

- ARP table is a cache
- Entries time out and are removed
- Avoids stale bindings
- Typical timeout: 20 minutes

Algorithm For Processing ARP Requests

- Extract sender's pair, (I_A, E_A) and update local ARP table if it exists
- If this is a request and the target is ‘‘me’’
  - Add sender's pair to ARP table if not present
  - Fill in target hardware address
  - Exchange sender and target entries
  - Set operation to *reply*
  - Send reply back to requester

Algorithm Features

- If A ARPs B, B keeps A's information
  - B will probably send a packet to A soon
- If A ARPs B, other machines do not keep A's information
  - Avoids clogging ARP caches needlessly

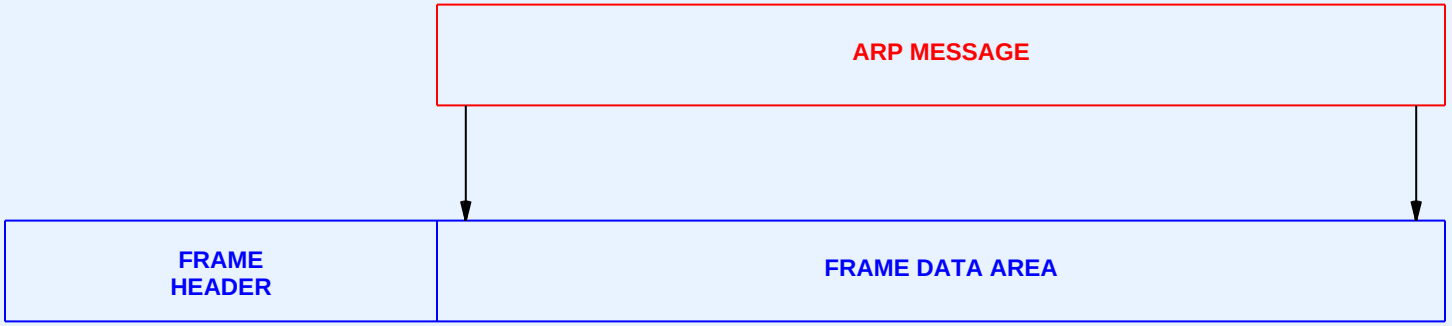
Conceptual Purpose Of ARP

- Isolates hardware address at low level
- Allows application programs to use IP addresses

ARP Encapsulation

- ARP message travels in data portion of network frame
- We say ARP message is *encapsulated*

Illustration Of ARP Encapsulation



Ethernet Encapsulation

- ARP message placed in frame data area
- Data area padded with zeroes if ARP message is shorter than minimum Ethernet frame
- Ethernet type 0x0806 used for ARP

Reverse Address Resolution Protocol

- Maps Ethernet address to IP address
- Same packet format as ARP
- Intended for bootstrap
  - Computer sends its Ethernet address
  - RARP server responds by sending computer's IP address
- Seldom used (replaced by DHCP)

Summary

- Computer's IP address independent of computer's hardware address
- Applications use IP addresses
- Hardware only understands hardware addresses
- Must map from IP address to hardware address for transmission
- Two types
  - Direct mapping
  - Dynamic mapping

Summary
(continued)

- Address Resolution Protocol (ARP) used for dynamic address mapping
- Important for Ethernet
- Sender broadcasts ARP request, and target sends ARP reply
- ARP bindings are cached
- Reverse ARP was originally used for bootstrap


Questions?

PART VI

INTERNET PROTOCOL: CONNECTIONLESS DATAGRAM DELIVERY

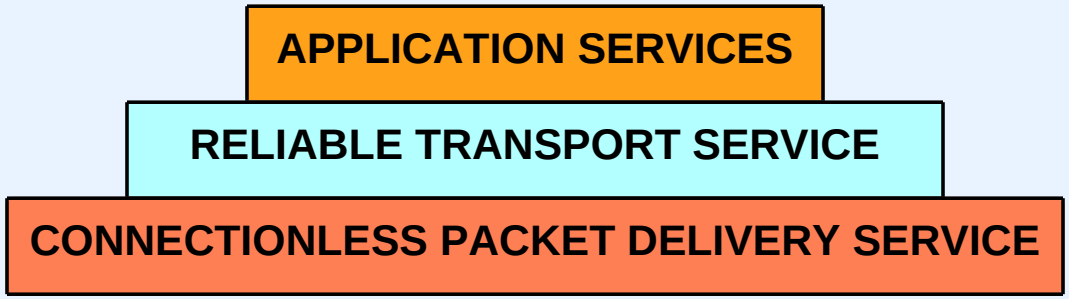
Internet Protocol

- One of two major protocols in TCP/IP suite
- Major goals
  - Hide heterogeneity
  - Provide the illusion of a single large network
  - Virtualize access

The Concept

IP allows a user to think of an internet as a single virtual network that interconnects all hosts, and through which communication is possible; its underlying architecture is both hidden and irrelevant.

Internet Services And Architecture Of Protocol Software



- Design has proved especially robust

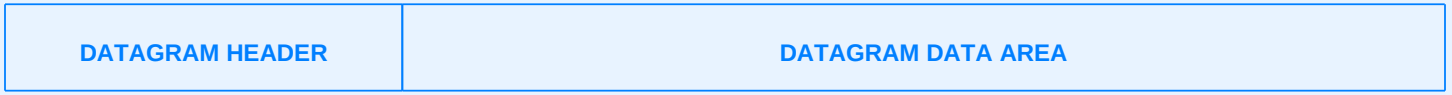
IP Characteristics

- Provides connectionless packet delivery service
- Defines three important items
  - Internet addressing scheme
  - Format of packets for the (virtual) Internet
  - Packet forwarding

Internet Packet

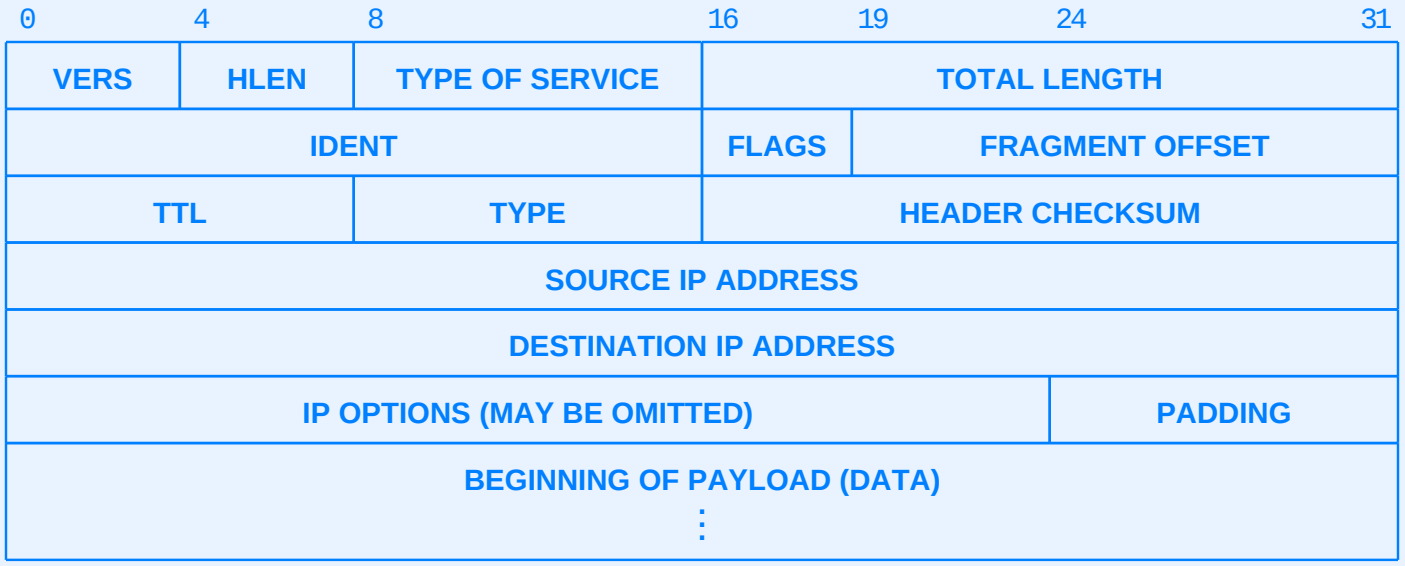
- Analogous to physical network packet
- Known as *IP datagram*

IP Datagram Layout



- Header contains
  - Source Internet address
  - Destination Internet address
  - Datagram type field
- Payload contains data being carried

Datagram Header Format



Addresses In The Header

- SOURCE is the address of original source
- DESTINATION is the address of ultimate destination

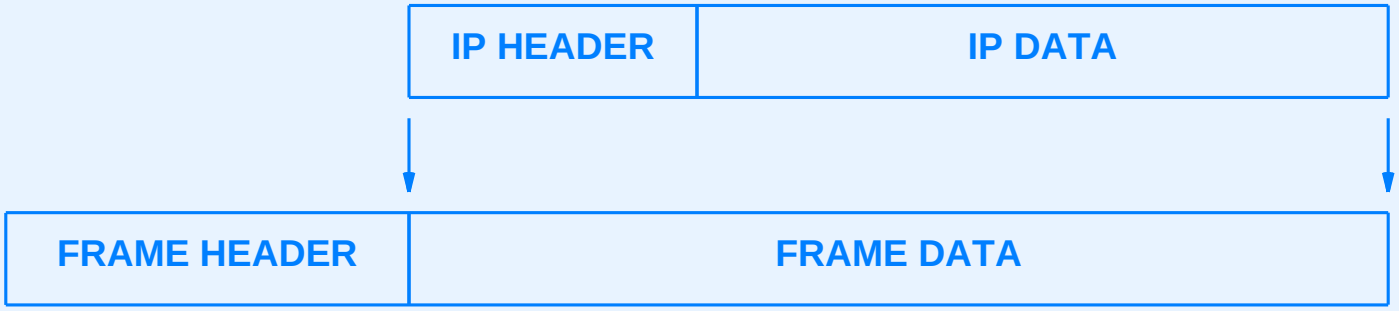
IP Versions

- Version field in header defines version of datagram
- Internet currently uses version 4 of IP, IPv4
- Preceding figure is the IPv4 datagram format
- IPv6 discussed later in the course

Datagram Encapsulation

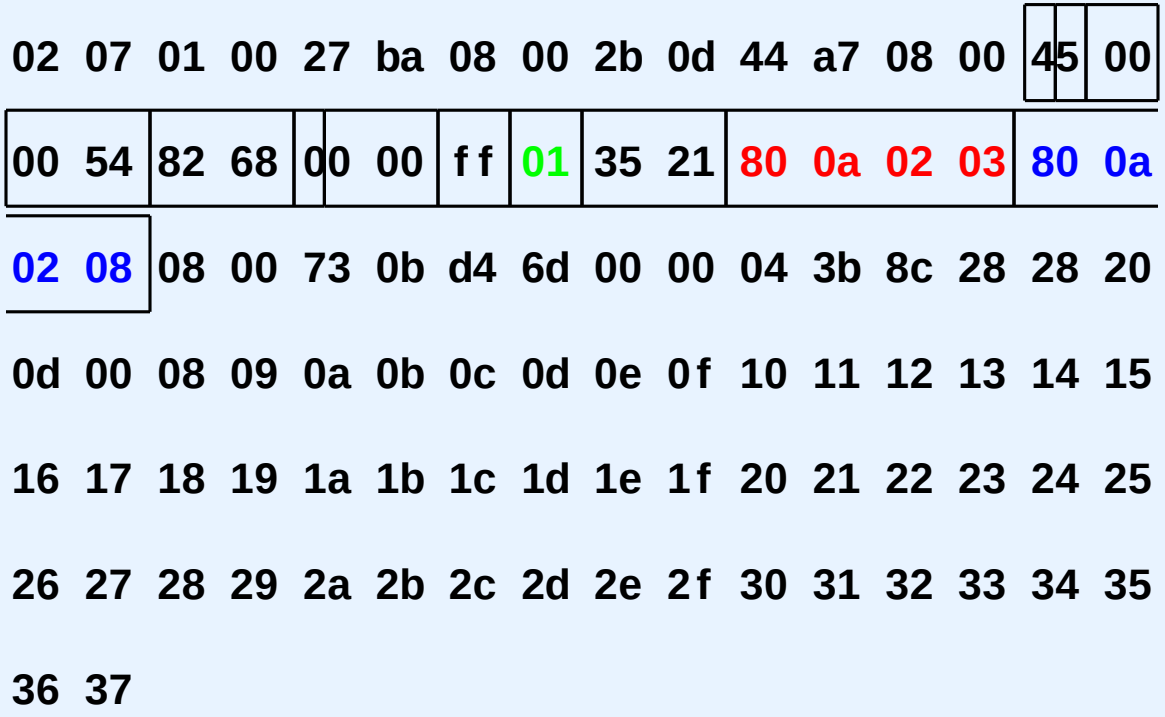
- Datagram *encapsulated* in network frame
- Network hardware treats datagram as data
- Frame type field identifies contents as datagram
  - Set by sending computer
  - Tested by receiving computer

Datagram Encapsulation For Ethernet



- Ethernet header contains Ethernet hardware addresses
- Ethernet type field set to 0x0800

Datagram Encapsulated In Ethernet Frame



- 20-octet IP header follows Ethernet header
- IP source: 128.10.2.3 (800a0203)
- IP destination: 128.10.2.8 (800a0208)
- IP type: 01 (ICMP)

Standards For Encapsulation

- TCP/IP protocols define encapsulation for each possible type of network hardware
  - Ethernet
  - Frame Relay
  - Others

Encapsulation Over Serial Networks

- Serial hardware transfers stream of octets
  - Leased serial data line
  - Dialup telephone connection
- Encapsulation of IP on serial network
  - Implemented by software
  - Both ends must agree
- Most common standards: Point to Point Protocol (PPP)

Encapsulation For Avian Carriers (RFC 1149)

- Characteristics of avian carrier
  - Low throughput
  - High delay
  - Low altitude
  - Point-to-point communication
  - Intrinsic collision avoidance
- Encapsulation
  - Write in hexadecimal on scroll of paper
  - Attach to bird’s leg with duct tape
- For an implementation see

http://www.blug.linux.no/rfc1149

A Potential Problem

- A datagram can contain up to 65535 total octets (including header)
- Network hardware limits maximum size of frame (e.g., Ethernet limited to 1500 octets)
  - Known as the network *Maximum Transmission Unit* (*MTU*)
- Question: how is encapsulation handled if datagram exceeds network MTU?

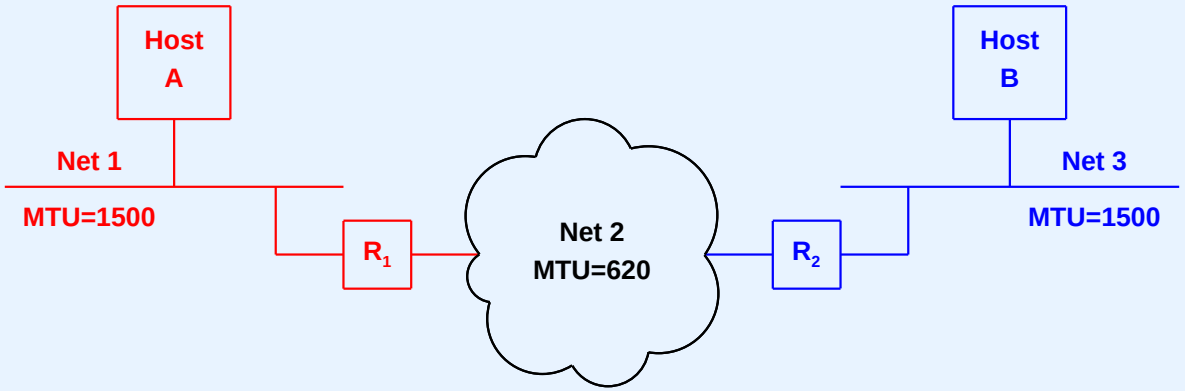
Possible Ways To Accommodate Networks With Differing MTUs

- Force datagram to be less than smallest possible MTU
  - Inefficient
  - Cannot know minimum MTU
- Hide the network MTU and accommodate arbitrary datagram size

Accommodating Large Datagrams

- Cannot send large datagram in single frame
- Solution
  - Divide datagram into pieces
  - Send each piece in a frame
  - Called *datagram fragmentation*

Illustration Of When Fragmentation Needed

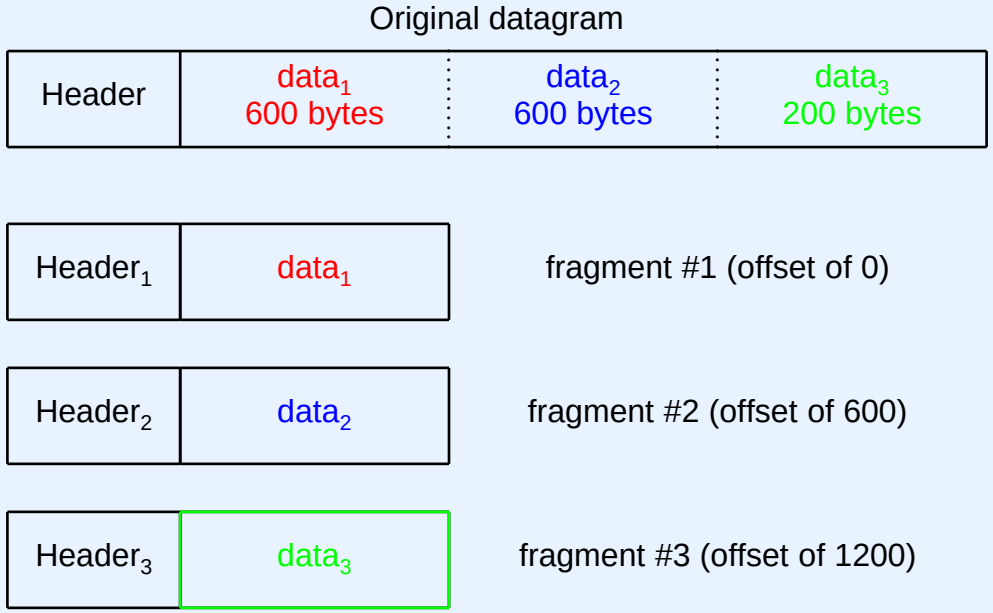


- Hosts A and B send datagrams of up to 1500 octets
- Router R_1 fragments large datagrams from Host A before sending over Net 2
- Router R_2 fragments large datagrams from Host B before sending over Net 2

Datagram Fragmentation

- Performed by routers
- Divides datagram into several, smaller datagrams called fragments
- Fragment uses same header format as datagram
- Each fragment forwarded independently

Illustration Of Fragmentation



- Offset specifies where data belongs in original datagram
- Offset actually stored as multiples of 8 octets
- MORE FRAGMENTS bit turned off in header of fragment #3

Fragmenting A Fragment

- Fragment can be further fragmented
- Occurs when fragment reaches an even-smaller MTU
- Discussion: which fields of the datagram header are used, and what is the algorithm?

Reassembly

- Ultimate destination puts fragments back together
  - Key concept!
  - Needed in a connectionless Internet
- Known as *reassembly*
- No need to reassemble subfragments first
- Timer used to ensure all fragments arrive
  - Timer started when first fragment arrives
  - If timer expires, entire datagram discarded

Time To Live

- TTL field of datagram header decremented at each hop (i.e., each router)
- If TTL reaches zero, datagram discarded
- Prevents datagrams from looping indefinitely (in case forwarding error introduces loop)
- IETF recommends initial value of 255 (max)

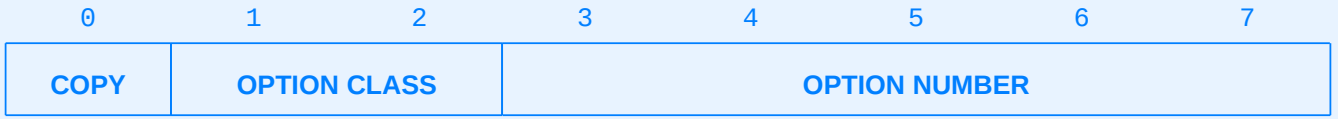
Checksum Field In Datagram Header

- 16-bit 1’s complement checksum
- Over IP header only!
- Recomputed at each hop

IP Options

- Seldom used
- Primarily for debugging
- Only *some* options copied into fragments
- Are variable length
- Note: padding needed because header length measured in 32-bit multiples
- Option starts with option code octet

Option Code Octet



Option Class	Meaning
0	Datagram or network control
1	Reserved for future use
2	Debugging and measurement
3	Reserved for future use

IP Semantics

- IP uses best-effort delivery
  - Makes an attempt to deliver
  - Does not guarantee delivery
- In the Internet, routers become overrun or change routes, meaning that:
  - Datagrams can be lost
  - Datagrams can be duplicated
  - Datagrams can arrive out of order or scrambled
- Motivation: allow IP to operate over the widest possible variety of physical networks

Output From PING Program

```
PING venera.isi.edu (128.9.0.32): 64 data bytes
      at 1.0000 second intervals

72 bytes from 128.9.0.32: icmp_seq=0. time=170. ms
72 bytes from 128.9.0.32: icmp_seq=1. time=150. ms
72 bytes from 128.9.0.32: icmp_seq=1. time=160. ms
72 bytes from 128.9.0.32: icmp_seq=2. time=160. ms
72 bytes from 128.9.0.32: icmp_seq=3. time=160. ms

----venera.isi.edu PING Statistics----
4 packets transmitted, 5 packets received,
             -25% packet loss
round-trip (ms)  min/avg/max = 150/160/170
```

- Shows actual case of duplication

Summary

- Internet Protocol provides basic connectionless delivery service for the Internet
- IP defines *IP datagram* to be the format of packets on the Internet
- Datagram header
  - Has fixed fields
  - Specifies source, destination, and type
  - Allows options
- Datagram encapsulated in network frame for transmission

Summary
(continued)

- Fragmentation
  - Needed when datagram larger than MTU
  - Usually performed by routers
  - Divides datagram into fragments
- Reassembly
  - Performed by ultimate destination
  - If some fragment(s) do not arrive, datagram discarded
- To accommodate all possible network hardware, IP does not require reliability (best-effort semantics)


Questions?

PART VII

INTERNET PROTOCOL: FORWARDING IP DATAGRAMS

Datagram Transmission

- Host delivers datagrams to directly connected machines
- Host sends datagrams that cannot be delivered directly to router
- Routers forward datagrams to other routers
- Final router delivers datagram directly

Question

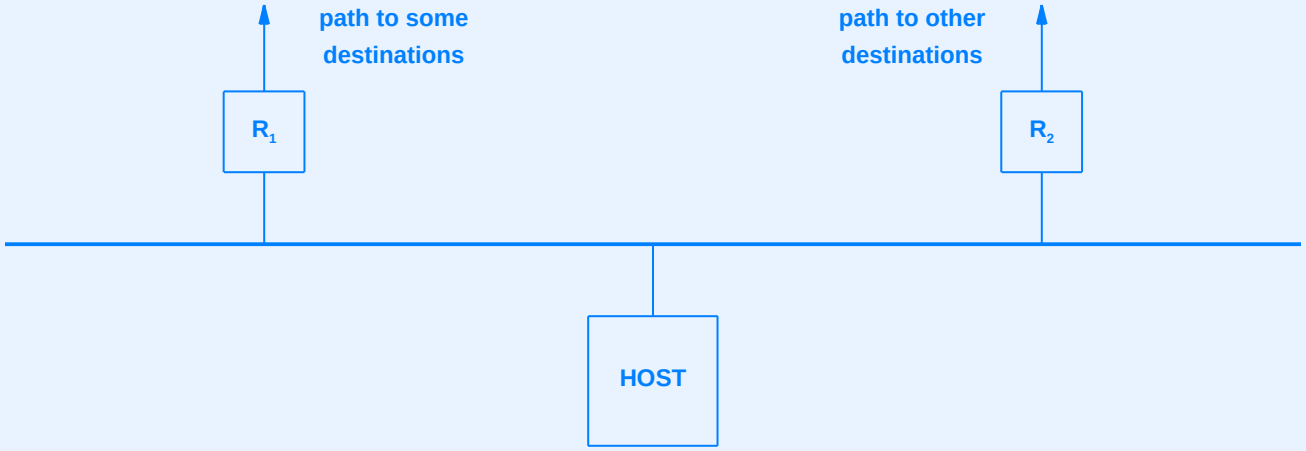
Does a host need to make forwarding choices?

Question

Does a host need to make forwarding choices?

Answer: YES!

Example Host That Must Choose How To Forward Datagrams



- Note: host is singly homed!

Two Broad Cases

- Direct delivery
  - Ultimate destination can be reached over one network
  - The ‘‘last hop’’ along a path
  - Also occurs when two communicating hosts both attach to the same physical network
- Indirect delivery
  - Requires intermediary (router)

Important Design Decision

Transmission of an IP datagram between two machines on a single physical network does not involve routers. The sender encapsulates the datagram in a physical frame, binds the destination IP address to a physical hardware address, and sends the resulting frame directly to the destination.

Testing Whether A Destination Lies On The Same Physical Network As The Sender

Because the Internet addresses of all machines on a single network include a common network prefix and extracting that prefix requires only a few machine instructions, testing whether a machine can be reached directly is extremely efficient.

Datagram Forwarding

- General paradigm
  - Source host sends to first router
  - Each router passes datagram to next router
  - Last router along path delivers datagram to destination host
- Only works if routers cooperate

General Concept

Routers in a TCP/IP Internet form a cooperative, interconnected structure. Datagrams pass from router to router until they reach a router that can deliver the datagram directly.

Efficient Forwarding

- Decisions based on table lookup
- Routing tables keep only network portion of addresses (size proportional to number of networks, not number of hosts)
- Extremely efficient
  - Lookup
  - Route update

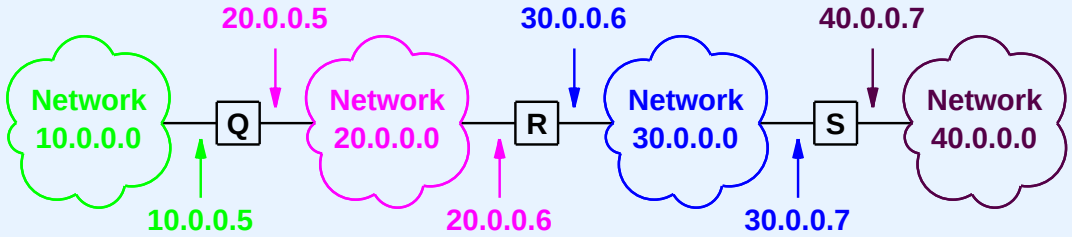
Important Idea

- Table used to decide how to send datagram known as *routing table* (also called a *forwarding table*)
- Routing table only stores address of next router along the path
- Scheme is known as *next-hop forwarding* or *next-hop routing*

Terminology

- Originally
  - *Routing* used to refer to passing datagram from router to router
- More recently
  - Purists decided to use *forwarding* to refer to the process of looking up a route and sending a datagram
- But...
  - Table is usually called a *routing table*

Conceptual Contents Of Routing Table Found In An IP Router



An example Internet with IP addresses

TO REACH NETWORK	ROUTE TO THIS ADDRESS
20.0.0.0 / 8	DELIVER DIRECT
30.0.0.0 / 8	DELIVER DIRECT
10.0.0.0 / 8	20.0.0.5
40.0.0.0 / 8	30.0.0.7

The routing table for router R

Special Cases

- Default route
- Host-specific route

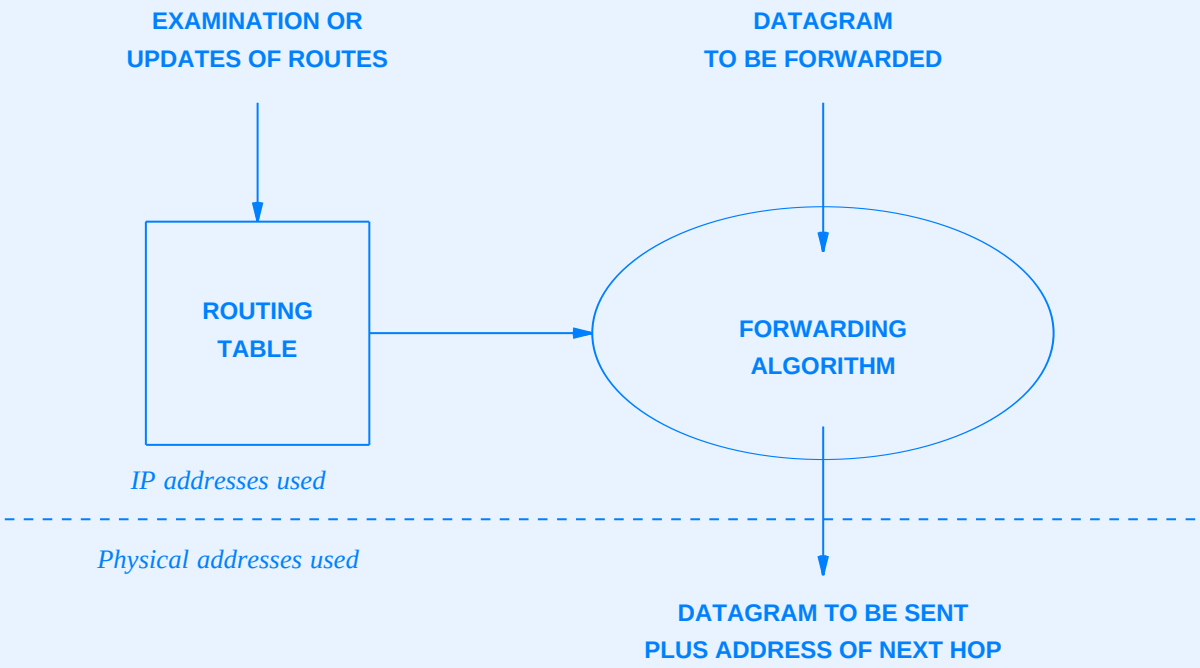
Default Route

- Special entry in IP routing table
- Matches ‘‘any’’ destination address
- Only one default permitted
- Only selected if no other match in table

Host-Specific Route

- Entry in routing table
- Matches entire 32-bit value
- Can be used to send traffic for a specific host along a specific path (i.e., can differ from the network route)
- More later in the course

Level Of Forwarding Algorithm



- Routing table uses IP addresses, not physical addresses

Summary

- IP uses routing table to forward datagrams
- Routing table
  - Stores pairs of network prefix and next hop
  - Can contain host-specific routes and a default route


Questions?

PART VIII

ERROR AND CONTROL MESSAGES (ICMP)

Errors In Packet Switching Networks

- Causes include
  - Temporary or permanent disconnection
  - Hardware failures
  - Router overrun
  - Routing loops
- Need mechanisms to detect and correct

Error Detection And Reporting Mechanisms

- IP header checksum to detect transmission errors
- Error reporting mechanism to distinguish between events such as lost datagrams and incorrect addresses
- Higher level protocols (i.e., TCP) must handle all other problems

Error Reporting Mechanism

- Named *Internet Control Message Protocol* (*ICMP*)
- Required and integral part of IP
- Used primarily by routers to report delivery or routing problems to original source
- Also includes informational (nonerror) functionality
- Uses IP to carry control messages
- No error messages sent about error messages

ICMP Purpose

The Internet Control Message Protocol allows a router to send error or control messages to the source of a datagram, typically a host. ICMP provides communication between the Internet Protocol software on one machine and the Internet Protocol software on another.

Error Reporting Vs. Error Correction

- ICMP does not
  - Provide interaction between a router and the source of trouble
  - Maintain state information (each packet is handled independently)
- Consequence

When a datagram causes an error, ICMP can only report the error condition back to the original source of the datagram; the source must relate the error to an individual application program or take other action to correct the problem.

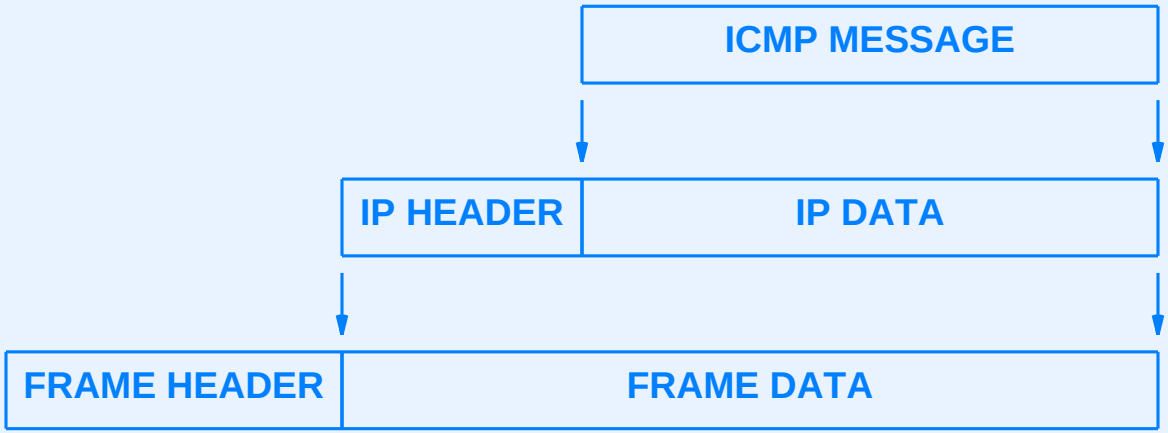
Important Restriction

- ICMP only reports problems to original source
- Discussion question: what major problem in the Internet cannot be handled with ICMP?

ICMP Encapsulation

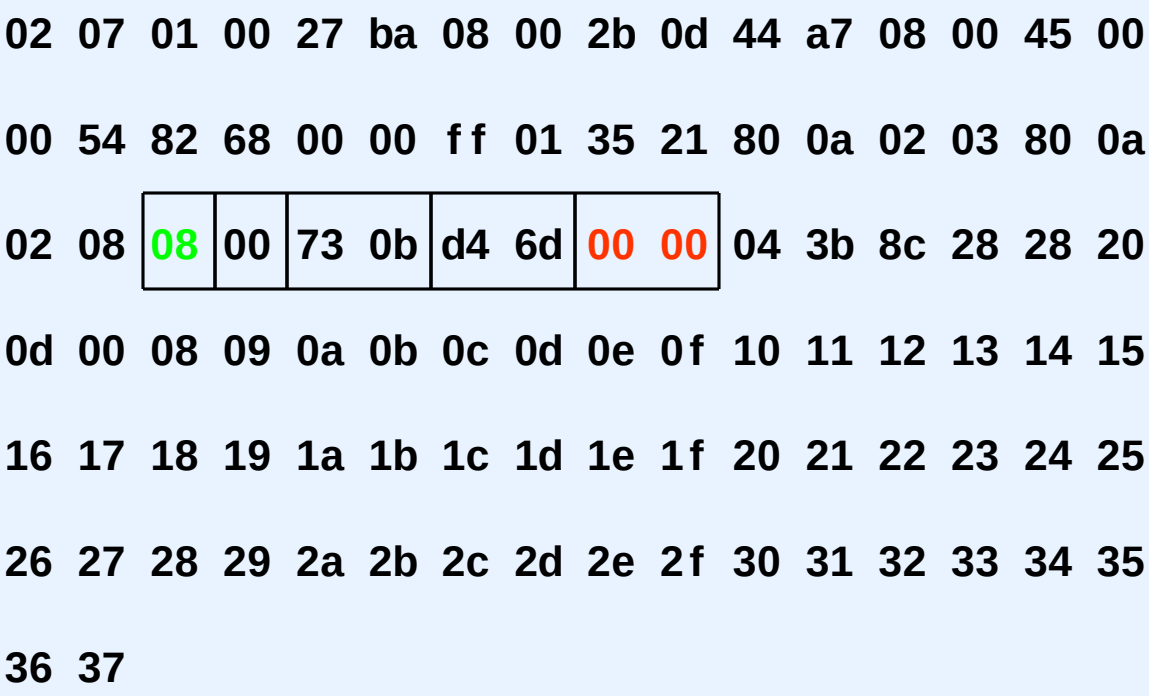
- ICMP message travels in IP datagram
- Entire ICMP message treated as data in the datagram
- Two levels of encapsulation result

ICMP Message Encapsulation



- ICMP message has header and data area
- Complete ICMP message is treated as data in IP datagram
- Complete IP datagram is treated as data in physical network frame

Example Encapsulation In Ethernet



- ICMP header follows IP header, and contains eight bytes
- ICMP type field specifies echo request message (08)
- ICMP sequence number is zero

ICMP Message Format

- Multiple message types
- Each message has its own format
- Messages
  - Begin with 1-octet TYPE field that identifies which of the basic ICMP message types follows
  - Some messages have a 1-octet CODE field that further classifies the message
- Example
  - TYPE specifies destination unreachable
  - CODE specifies whether host or network was unreachable

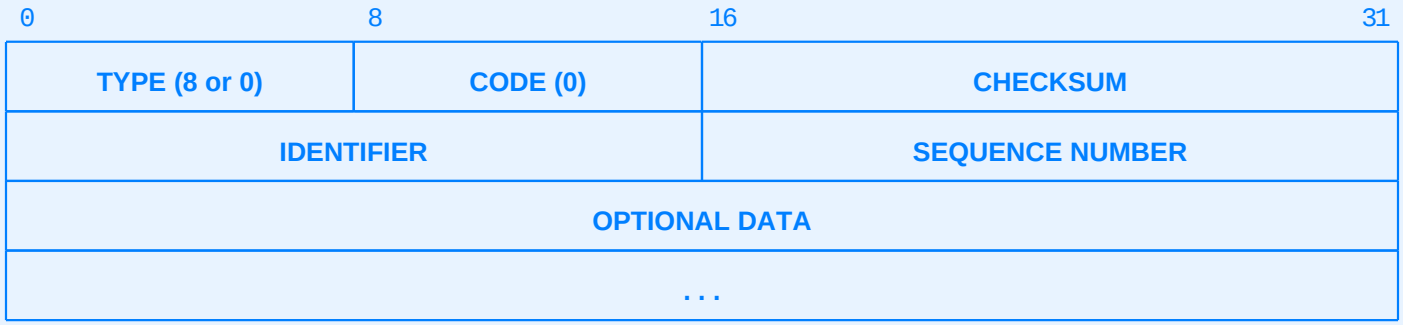
ICMP Message Types

Type Field	ICMP Message Type
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect (change a route)
6	Alternate Host Address
8	Echo Request
9	Router Advertisement
10	Router Solicitation
11	Time Exceeded for a Datagram
12	Parameter Problem on a Datagram
13	Timestamp Request
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply

ICMP Message Types (continued)

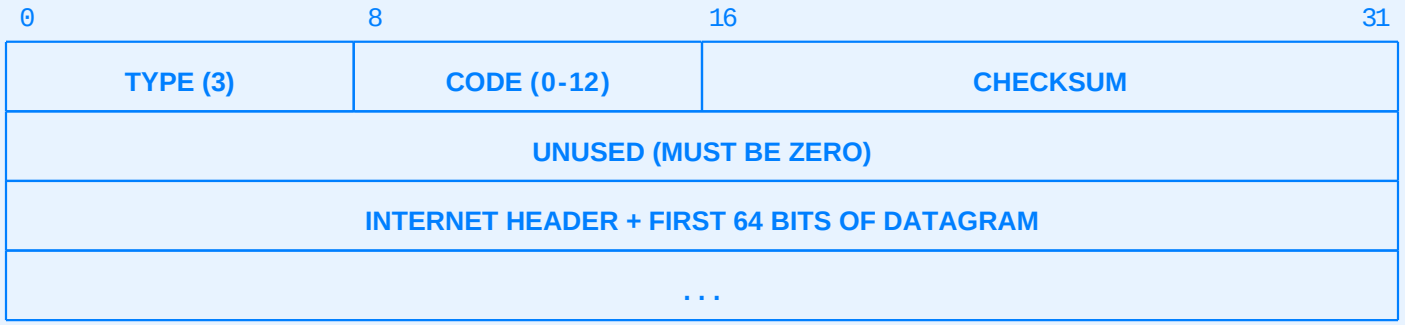
Type Field	ICMP Message Type
30	Traceroute
31	Datagram Conversion Error
32	Mobile Host Redirect
33	IPv6 Where-Are-You
34	IPv6 I-Am-Here
35	Mobile Registration Request
36	Mobile Registration Reply
37	Domain Name Request
38	Domain Name Reply
39	SKIP
40	Photuris

Example ICMP Message (ICMP Echo Request)



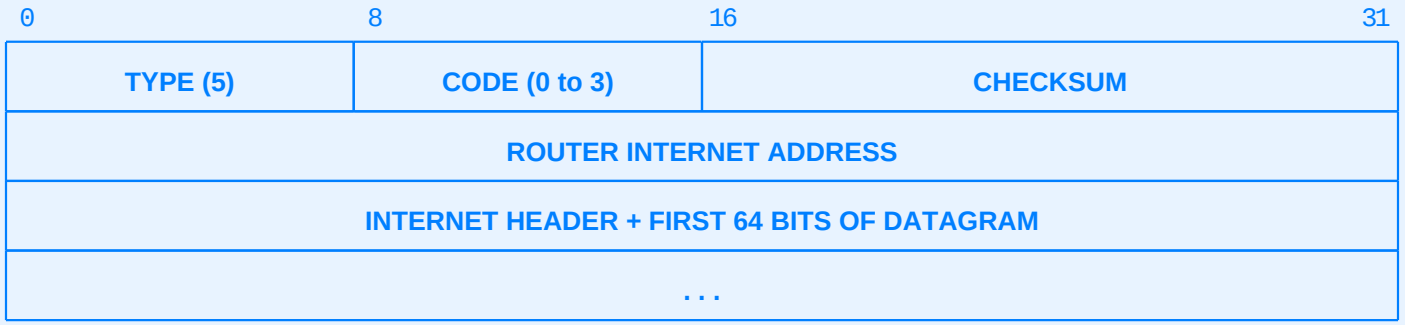
- Sent by *ping* program
- Used to test reachability

Example ICMP Message (Destination Unreachable)



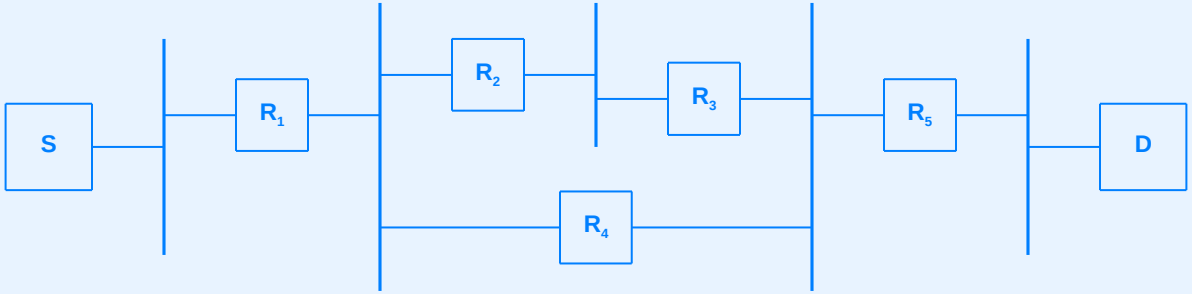
- Used to report that datagram could not be delivered
- Code specifies details

Example ICMP Message (Redirect)



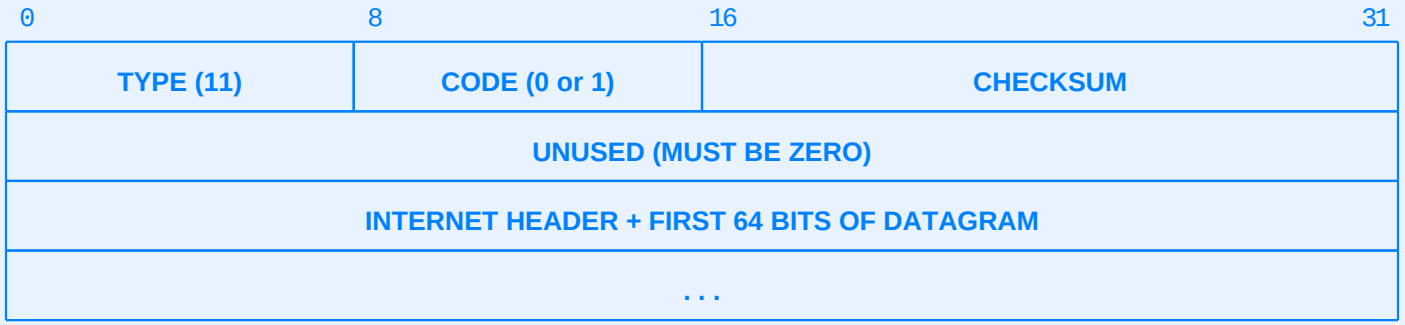
- Used to report incorrect route

Situation Where An ICMP Redirect Cannot Be Used



- R_5 cannot redirect R_1 to use shorter path

Example ICMP Message (Time Exceeded)



- At least one fragment failed to arrive, or
- TTL field in IP header reached zero

ICMP Trick

- Include datagram that caused problem in the error message
  - Efficient (sender must determine how to correct problem)
  - Eliminates need to construct detailed message
- Problem: entire datagram may be too large
- Solution: send IP header plus 64 bits of data area (sufficient in most cases)

Summary

- ICMP
  - Required part of IP
  - Used to report errors to original source
  - Reporting only: no interaction or error correction
- Several ICMP message types, each with its own format
- ICMP message begins with 1-octet TYPE field
- ICMP encapsulated in IP for delivery


Questions?

PART IX

INTERNET PROTOCOL: CLASSLESS AND SUBNET ADDRESS EXTENSIONS (CIDR)

Recall

In the original IP addressing scheme, each physical network is assigned a unique network address; each host on a network has the network address as a prefix of the host's individual address.

- Routers only examine prefix (small routing tables)

An Observation

- Division into prefix and suffix means: site can assign and use IP addresses in unusual ways provided
  - All hosts and routers at the site honor the site's scheme
  - Other sites on the Internet can treat addresses as a network prefix and a host suffix

Classful Addressing

- Three possible classes for networks
- Class C network limited to 254 hosts (cannot use all-1s or all-0s)
- Personal computers result in networks with many hosts
- Class B network allows many hosts, but insufficient class B prefixes

Question

- How can we minimize the number of assigned network prefixes (especially class B) without abandoning the 32-bit addressing scheme?

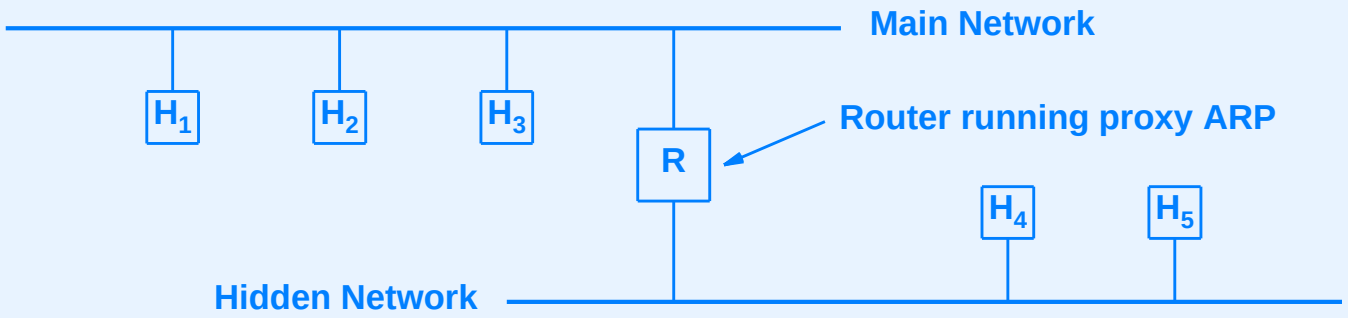
Two Answers To The Minimization Question

- Proxy ARP
- Subnet addressing

Proxy ARP

- Layer 2 solution
- Allow two physical networks to share a single IP prefix
- Arrange special system to answer ARP requests and forward datagrams between networks

Illustration Of Proxy ARP



- Hosts think they are on same network
- Known informally as *the ARP hack*

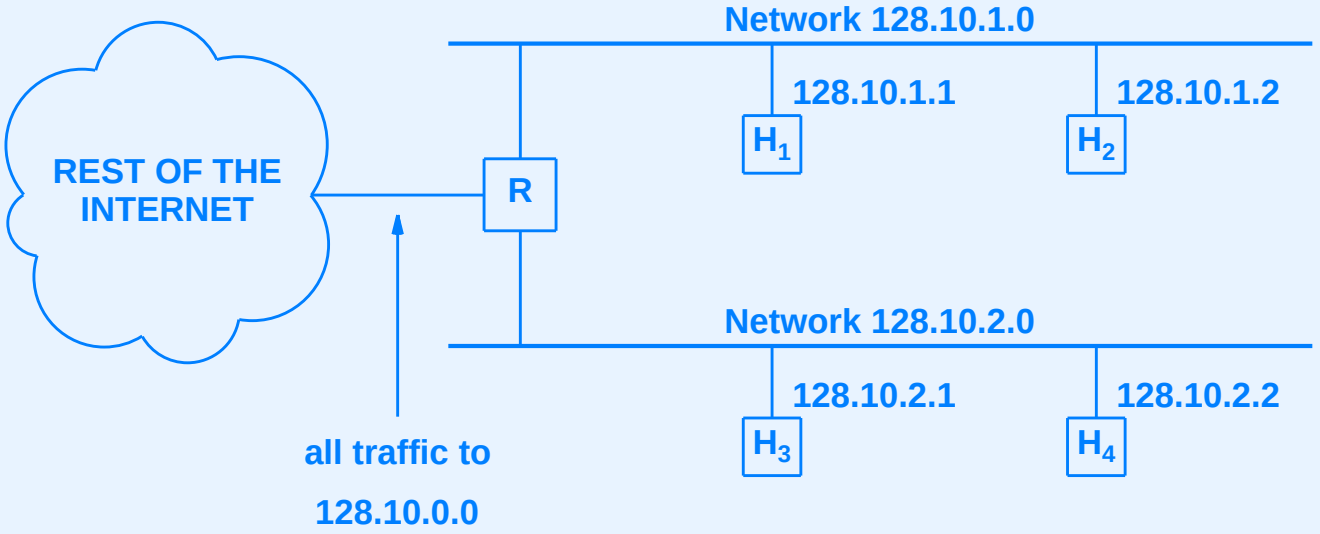
Assessment Of Proxy ARP

- Chief advantages
  - Transparent to hosts
  - No change in IP routing tables
- Chief disadvantages
  - Does not generalize to complex topology
  - Only works on networks that use ARP
  - Most proxy ARP systems require manual configuration

Subnet Addressing

- Not part of original TCP/IP address scheme
- Allows an organization to use a single network prefix for multiple physical networks
- Subdivides the host suffix into a pair of fields for physical network and host
- Interpreted only by routers and hosts at the site; treated like normal address elsewhere

Example Of Subnet Addressing

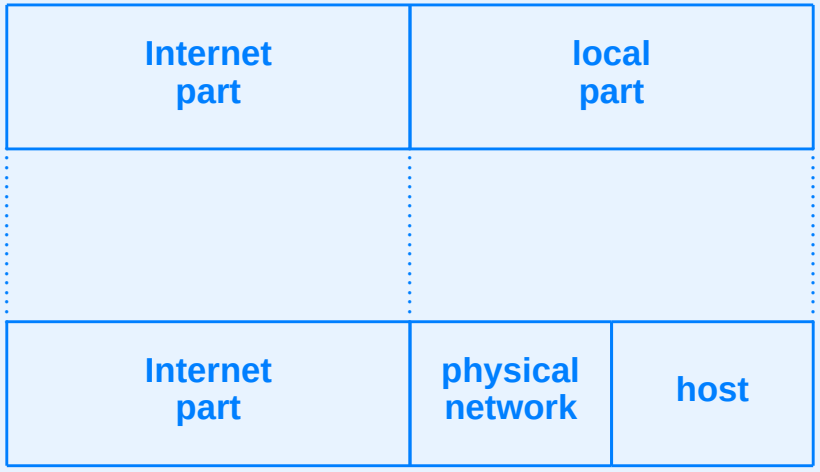


- Both physical networks share prefix 128.10
- Router R uses third octet of address to choose physical net

Interpretation Of Addresses

- Classful interpretation is two-level hierarchy
  - Physical network identified by prefix
  - Host on the net identified by suffix
- Subnetted interpretation is three-level hierarchy
  - Site identified by network prefix
  - Physical net at site identified by part of suffix
  - Host on the net identified by remainder of suffix

Example Of Address Interpretation (Subnetted Class B Address)



Note: in this case, 16-bit host portion is divided into two 8-bit fields

Choice Of Subnet Size

- How should host portion of address be divided?
- Answer depends on topology at site and number of hosts per network

Example Of Site With Hierarchical Topology

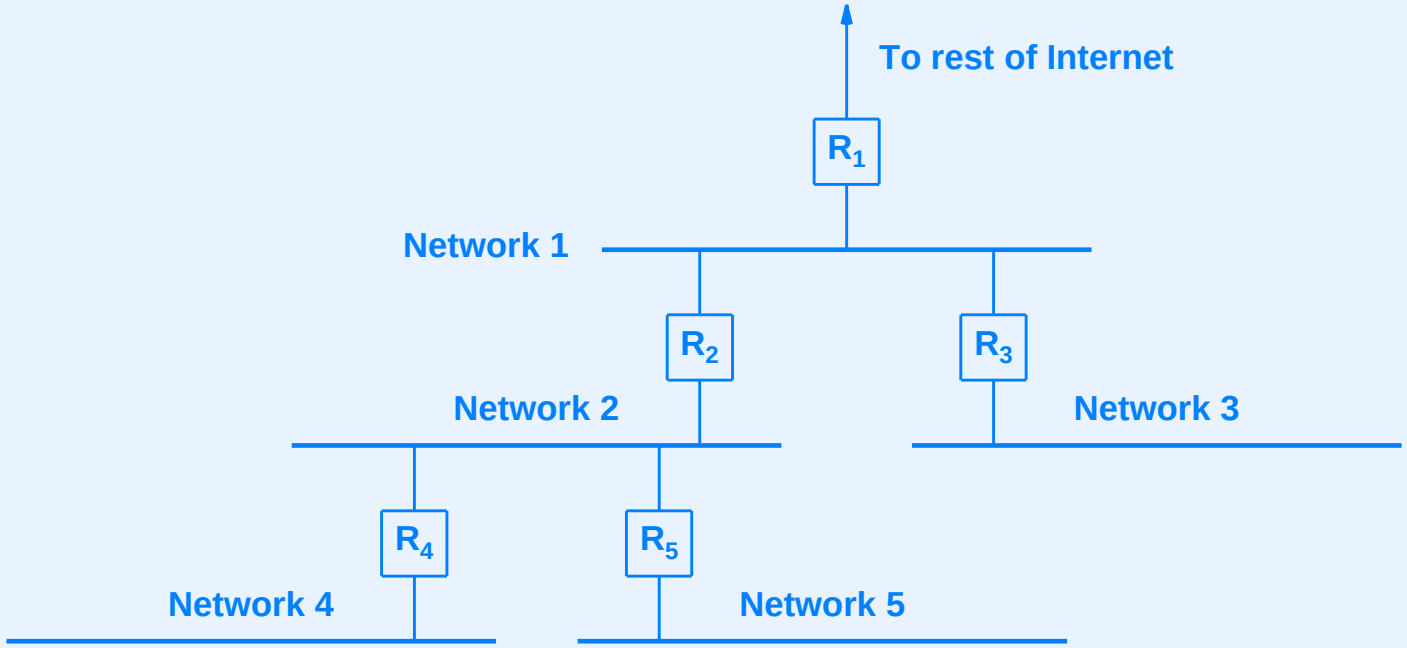
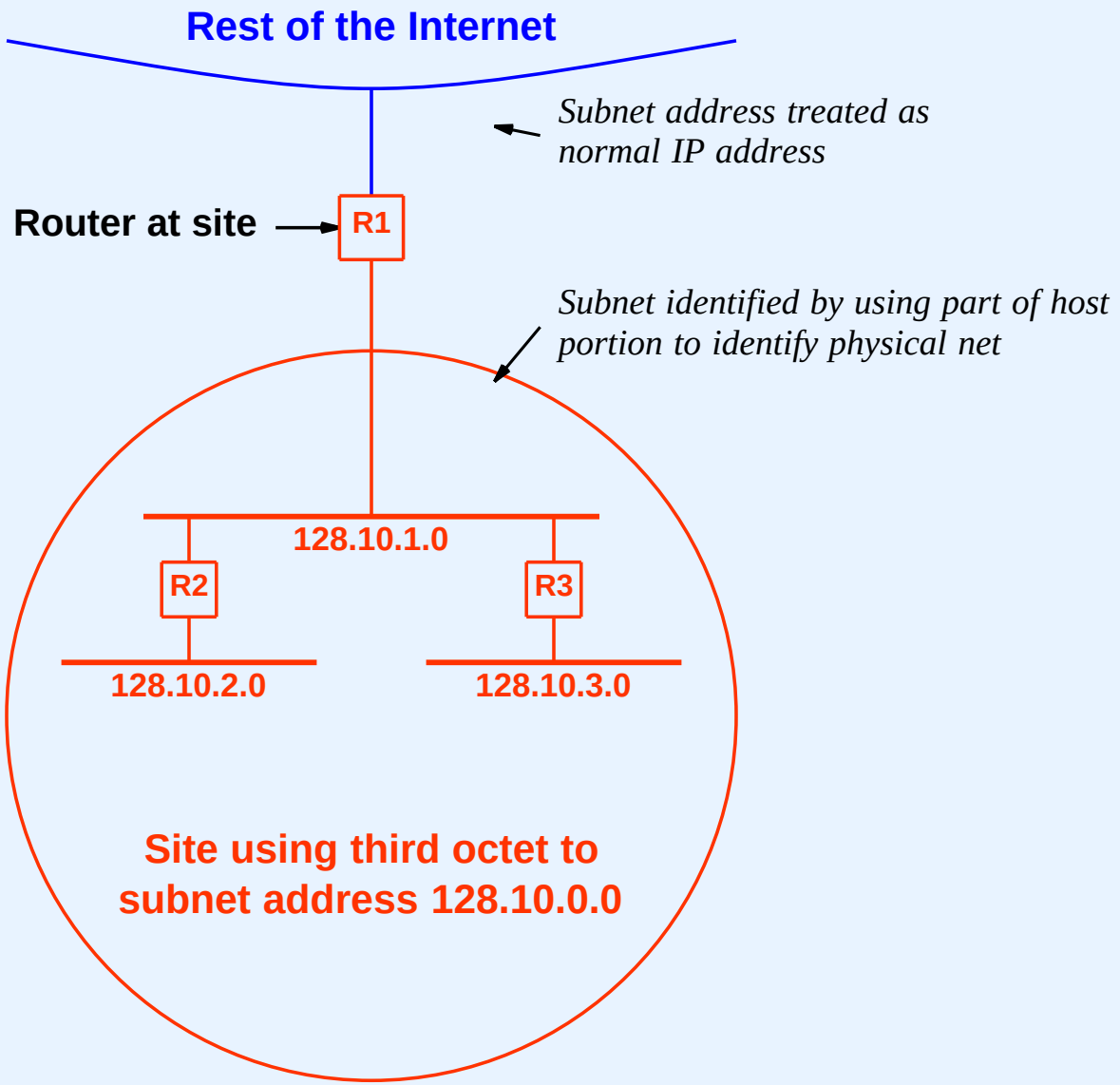


Illustration Of Subnet Addressing



Address Mask

- Each physical network is assigned 32-bit *address mask* (also called *subnet mask*)
- One bits in mask cover network prefix plus zero or more bits of suffix portion
- Logical *and* between mask and destination IP address extracts the prefix and subnet portions

Two Possible Mask Assignments

- Fixed-length subnet masks
- Variable-length subnet masks

Fixed-length Subnet Masks

- Organization uses same mask on all networks
- Advantages
  - Uniformity
  - Ease of debugging / maintenance
- Disadvantages
  - Number of nets fixed for entire organization
  - Size of physical nets fixed for entire organization

Possible Fixed-Length Subnets For Sixteen Bit Host Address

Bits in mask	# subnets	# hosts/subnet
16	1	65534
18	2	16382
19	6	8190
20	14	4094
21	30	2046
22	62	1022
23	126	510
24	254	254
25	510	126
26	1022	62
27	2046	30
28	4094	14
29	8190	6
30	16382	2

- All-*0*s and all-*1*s values must be omitted
- Organization chooses one line in table

Variable-Length Subnet Masks (VLSM)

- Administrator chooses size for each physical network
- Mask assigned on per-network basis
- Advantages
  - Flexibility to mix large and small nets
  - More complete use of address space
- Disadvantages
  - Difficult to assign / administer
  - Potential address ambiguity
  - More routes

Use Of Address Space (Start With 16 Bits Of Host Suffix)

- One possible VLSM assignment (92.9% of addresses used)
  - 11 networks of 2046 hosts each
  - 24 networks of 254 hosts each
  - 256 networks of 126 hosts each
- Another possible VLSM assignment (93.1% of addresses used)
  - 9 networks of 2046 hosts each
  - 2 networks of 1022 hosts each
  - 40 networks of 510 hosts each
  - 160 networks of 126 hosts each

Subnet Details

- Two interesting facts
  - *Can* assign all-0's or all-1's subnet
  - *Can* assign noncontiguous subnet mask bits
- In practice
  - Should avoid both
- Discussion question: why does the subnet standard allow the all-1's and all-0's subnet numbers?

VLSM Example

- Use low-order sixteen bits of 128.10.0.0
- Create seven subnets
- Subnet 1
  - Up to 254 hosts
  - Subnet mask is 24 bits
- Subnets 2 through 7
  - Up to 62 hosts each
  - Subnet mask is 26 bits

Example VLSM Prefixes

- Subnet 1 (up to 254 hosts)

```
mask:   11111111 11111111 11111111 00000000
prefix: 10000000 00001010 00000001 00000000
```

- Subnet 2 (up to 62 hosts)

```
mask:   11111111 11111111 11111111 11000000
prefix: 10000000 00001010 00000000 10000000
```

- Subnet 3 (up to 62 hosts)

```
mask:   11111111 11111111 11111111 11000000
prefix: 10000000 00001010 00000000 11000000
```

Example VLSM Prefixes
(continued)

- Subnet 4 (up to 62 hosts)

```
mask:   11111111 11111111 11111111 11000000
prefix: 10000000 00001010 00000001 00000000
```

- Subnet 5 (up to 62 hosts)

```
mask:   11111111 11111111 11111111 11000000
prefix: 10000000 00001010 00000001 01000000
```

- Subnet 6 (up to 62 hosts)

```
mask:   11111111 11111111 11111111 11000000
prefix: 10000000 00001010 00000001 10000000
```

Example VLSM Prefixes
(continued)

- Subnet 7 (up to 62 hosts)

```
mask:   11111111 11111111 11111111 11000000
prefix: 10000000 00001010 00000001 11000000
```

Address Ambiguity

- Address of host 63 on subnet 1 is

```
mask:   11111111 11111111 11111111 00000000
prefix: 10000000 00001010 00000001 00000000
host:   10000000 00001010 00000001 00111111
```

- Directed broadcast address on subnet 4 is

```
mask:   11111111 11111111 11111111 11000000
prefix: 10000000 00001010 00000001 00000000
bcast:  10000000 00001010 00000001 00111111
```

Address Ambiguity

- Address of host 63 on subnet 1 is

```
mask:   11111111 11111111 11111111 00000000
prefix: 10000000 00001010 00000001 00000000
host:   10000000 00001010 00000001 00111111
```

- Directed broadcast address on subnet 4 is

```
mask:   11111111 11111111 11111111 11000000
prefix: 10000000 00001010 00000001 00000000
bcast:  10000000 00001010 00000001 00111111
```

- Same value!

More Address Ambiguity

- Directed broadcast address on subnet 1 is

```
mask:      11111111 11111111 11111111 00000000
prefix:    10000000 00001010 00000001 00000000
broadcast: 10000000 00001010 00000001 11111111
```

- Directed broadcast address on subnet 7 is

```
mask:      11111111 11111111 11111111 11000000
prefix:    10000000 00001010 00000001 11000000
broadcast:10000000 00001010 00000001 11111111
```

More Address Ambiguity

- Directed broadcast address on subnet 1 is

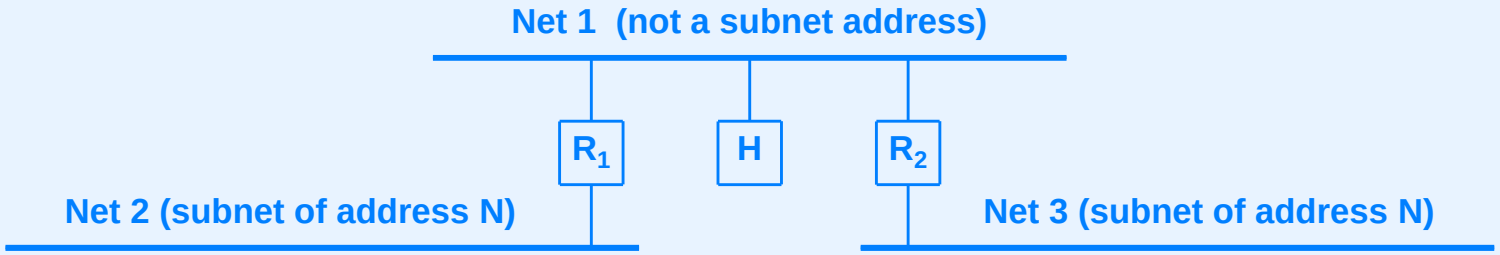
```
mask:      11111111 11111111 11111111 00000000
prefix:    10000000 00001010 00000001 00000000
broadcast: 10000000 00001010 00000001 11111111
```

- Directed broadcast address on subnet 7 is

```
mask:      11111111 11111111 11111111 11000000
prefix:    10000000 00001010 00000001 11000000
broadcast:10000000 00001010 00000001 11111111
```

- Same value!

Example Of Illegal Subnet Assignment



- Host cannot route among subnets
- Rule: subnets *must* be contiguous!

Variety Of Routes

- Forwarding must accommodate
  - Network-specific routes
  - Subnet-specific routes
  - Host-specific routes
  - Default route
  - Limited broadcast
  - Directed broadcast to network
  - Directed broadcast to specific subnet
- Single algorithm with address masks can accommodate all the above

Use Of Address Masks

- Each entry in routing table also has address mask
- All-1s mask used for host-specific routes
- Network mask used for network-specific routes
- Subnet mask used for subnet-specific routes
- All-0s mask used for default route

Unified Forwarding Algorithm

Algorithm:

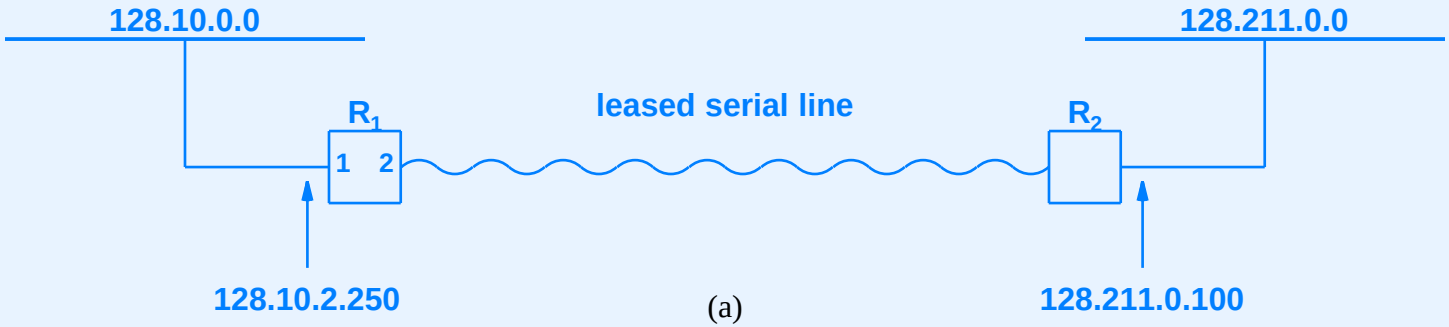
Forward_IP_Datagram (datagram, routing_table)

```
Extract destination IP address, ID, from datagram;
If prefix of ID matches address of any directly connected
   network send datagram to destination over that network
   (This involves resolving ID to a physical address,
   encapsulating the datagram, and sending the frame.)
else
   for each entry in routing table do
      Let N be the bitwise-and of ID and the subnet mask
      If N equals the network address field of the entry then
        forward the datagram to the specified next hop address
   endforloop
If no matches were found, declare a forwarding error;
```

Special Case: Unnumbered Serial Network

- Only two endpoints
- Not necessary to assign (waste) network prefix
- Trick: use remote IP address as next hop

Example Unnumbered Serial Network



(a)

TO REACH HOSTS ON NETWORK	ROUTE TO THIS ADDRESS	USING THIS INTERFACE
128.10.0.0	DELIVER DIRECT	1
default	128.211.0.100	2

(b)

Classless Inter-Domain Routing (CIDR)

- Problem
  - Continued exponential Internet growth
  - Subnetting insufficient
  - Limited IP addresses (esp. Class B)
- Dire prediction made in 1993:

  We will exhaust the address space ‘‘in a few years’’.

  Note: address space is *not* near exhaustion

CIDR Addressing

- Solution to problem
  - Temporary fix until next generation of IP
  - Backward compatible with classful addressing
  - Extend variable-length subnet technology to prefixes
- CIDR was predicted to work ‘‘for a few years’’
  - Extremely successful!
  - Will work for at least 25 years!

One Motivation For CIDR: Class C

- Fewer than seventeen thousand Class B numbers (total)
- More than two million Class C network numbers
- No one wants Class C (too small)
- CIDR allows
  - Merging 256 Class C numbers into a single prefix that is equivalent to Class B
  - Splitting a Class B along power of two boundaries

CIDR Notation

- Addresses written *NUMBER*/*m*
  - *NUMBER* is IP prefix
  - *m* is ‘‘address mask’’ length
- Example

  214.5.48.0/20

- Prefix occupies 20 bits
- Suffix occupies 12 bits
- Mask values must be converted to dotted decimal when configuring a router (and binary internally)

Route Proliferation

- If classful forwarding used, CIDR addresses result in more routes
- Example:
  - Single CIDR prefix spans 256 Class C network numbers (*supernetting*)
  - Classful routing table requires 256 separate entries

Route Condensation

- Solution: change forwarding as well as addressing
- Store address mask with each route
- Send pair of (address, mask) whenever exchanging routing information
- Known as a *CIDR block*

Example Of A CIDR Block

	Dotted Decimal	32-bit Binary Equivalent
lowest	128.211.168.0	10000000 11010011 10101000 00000000
highest	128.211.175.255	10000000 11010011 10101111 11111111

Dotted Decimal Equivalents

CIDR Notation	Dotted Decimal	CIDR Notation	Dotted Decimal
/1	128.0.0.0	/17	255.255.128.0
/2	192.0.0.0	/18	255.255.192.0
/3	224.0.0.0	/19	255.255.224.0
/4	240.0.0.0	/20	255.255.240.0
/5	248.0.0.0	/21	255.255.248.0
/6	252.0.0.0	/22	255.255.252.0
/7	254.0.0.0	/23	255.255.254.0
/8	255.0.0.0	/24	255.255.255.0
/9	255.128.0.0	/25	255.255.255.128
/10	255.192.0.0	/26	255.255.255.192
/11	255.224.0.0	/27	255.255.255.224
/12	255.240.0.0	/28	255.255.255.240
/13	255.248.0.0	/29	255.255.255.248
/14	255.252.0.0	/30	255.255.255.252
/15	255.254.0.0	/31	255.255.255.254
/16	255.255.0.0	/32	255.255.255.255

Example Of /30 CIDR Block

	Dotted Decimal	32-bit Binary Equivalent
lowest	128.211.176.212	10000000 11010011 10110000 11010100
highest	128.211.176.215	10000000 11010011 10110000 11010111

- Useful when customer of ISP has very small network

Implementation Of CIDR Route Lookup

- Each entry in routing table has address plus mask
- Search is organized from most-specific to least-specific (i.e., entry with longest mask is tested first)
- Known as *longest-prefix lookup* or *longest-prefix search*

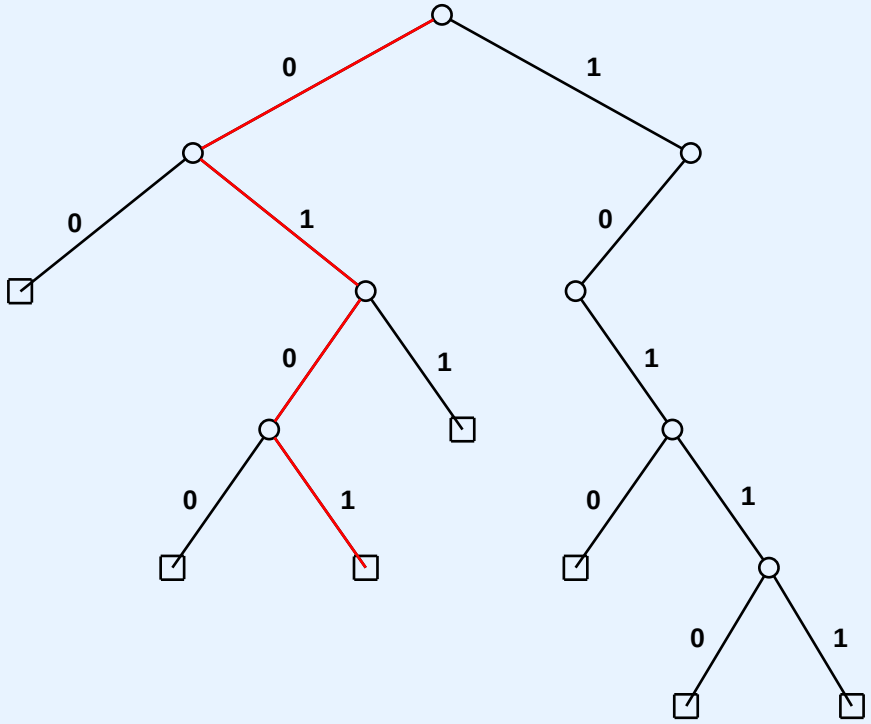
Implementing Longest-Prefix Matching

- Cannot easily use hashing
- Data structure of choice is *binary trie*
- Identifies unique prefix needed to match route

Example Of Unique Prefixes

32-Bit Address	Unique Prefix
00110101 00000000 00000000 00000000	00
01000110 00000000 00000000 00000000	0100
01010110 00000000 00000000 00000000	0101
01100001 00000000 00000000 00000000	011
10101010 11110000 00000000 00000000	1010
10110000 00000010 00000000 00000000	10110
10111011 00001010 00000000 00000000	10111

Example Binary Trie For The Seven Prefixes



- Path for 0101 is shown in red

Modifications And Extensions

- Several variations of trie data structures exist
  - PATRICIA trees
  - Level-Compressed tries (LC-tries)
- Motivation
  - Handle longest-prefix match
  - Skip levels that do not distinguish among routes

Nonroutable Addresses

- CIDR blocks reserved for use within a site
- Must never appear on the Internet
- ISPs do not maintain routes
- Also called *private addresses*

Prefix	Lowest Address	Highest Address
10 / 8	10.0.0.0	10.255.255.255
172.16 / 12	172.16.0.0	172.31.255.255
192.168 / 16	192.168.0.0	192.168.255.255
169.254 / 16	169.254.0.0	169.254.255.255

Summary

- Original IP addressing scheme was classful
- Two extensions added
  - Subnet addressing
  - CIDR addressing
- Subnetting used only within a site
- CIDR used throughout the Internet
- Both use 32-bit address mask
  - CIDR mask identifies division between network prefix and host suffix
  - Subnet mask identifies boundary between subnet and individual host

Summary
(continued)

- Single unified forwarding algorithm handles routes that are
  - Network-specific
  - Subnet-specific
  - Host-specific
  - Limited broadcast
  - Directed broadcast to network
  - Directed broadcast to subnet
  - Default
- Longest-prefix match required
  - Typical implementation: binary trie


Questions?

PART X

PROTOCOL LAYERING

Motivation For Layering

- Communication is difficult to understand
- Many subproblems
  - Hardware failure
  - Network congestion
  - Packet delay or loss
  - Data corruption
  - Data duplication or inverted arrivals

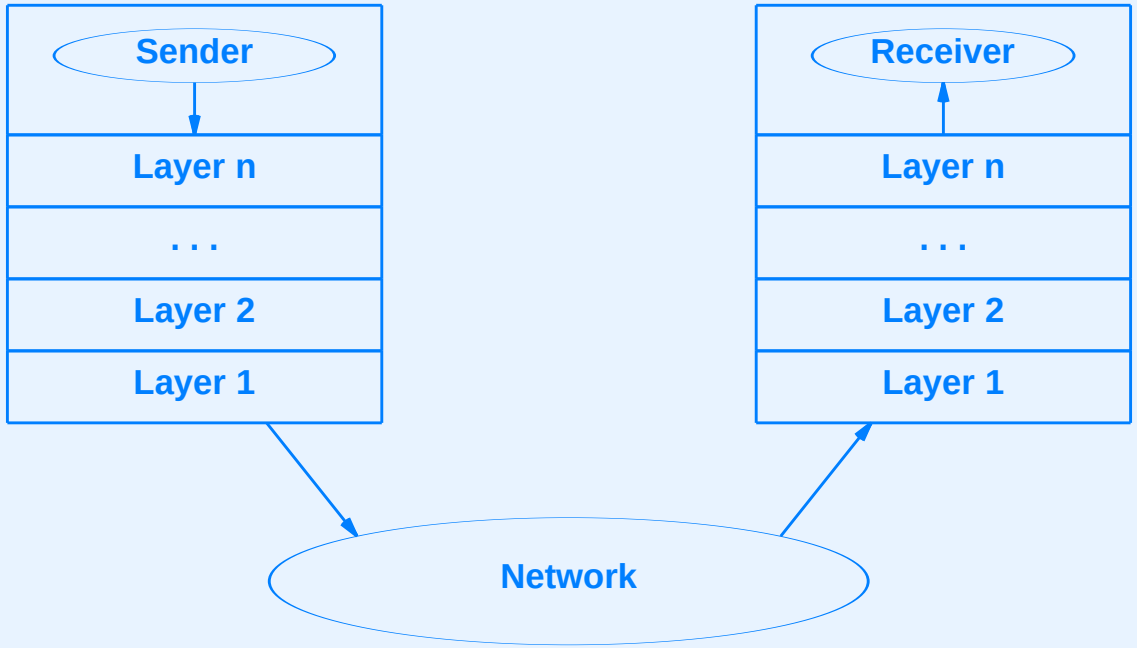
Solving The Problem

- Divide the problem into pieces
- Solve subproblems separately
- Combine into integrated whole
- Result is *layered protocols*

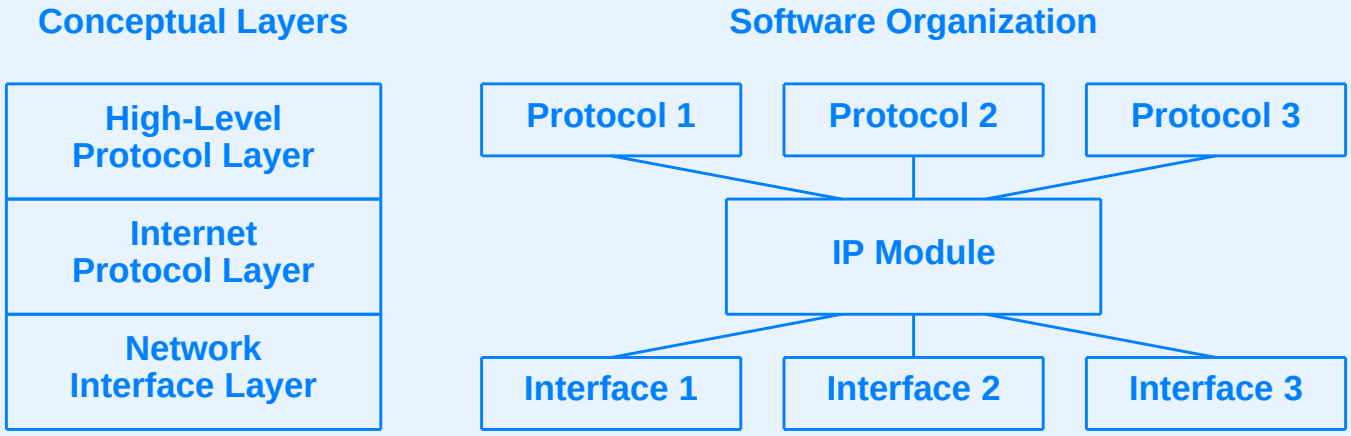
Protocol Layering

- Separates protocol functionality
- Each layer solves one part of the communication problem
- Intended primarily for protocol designers
- Set of layers is called a *protocol stack*

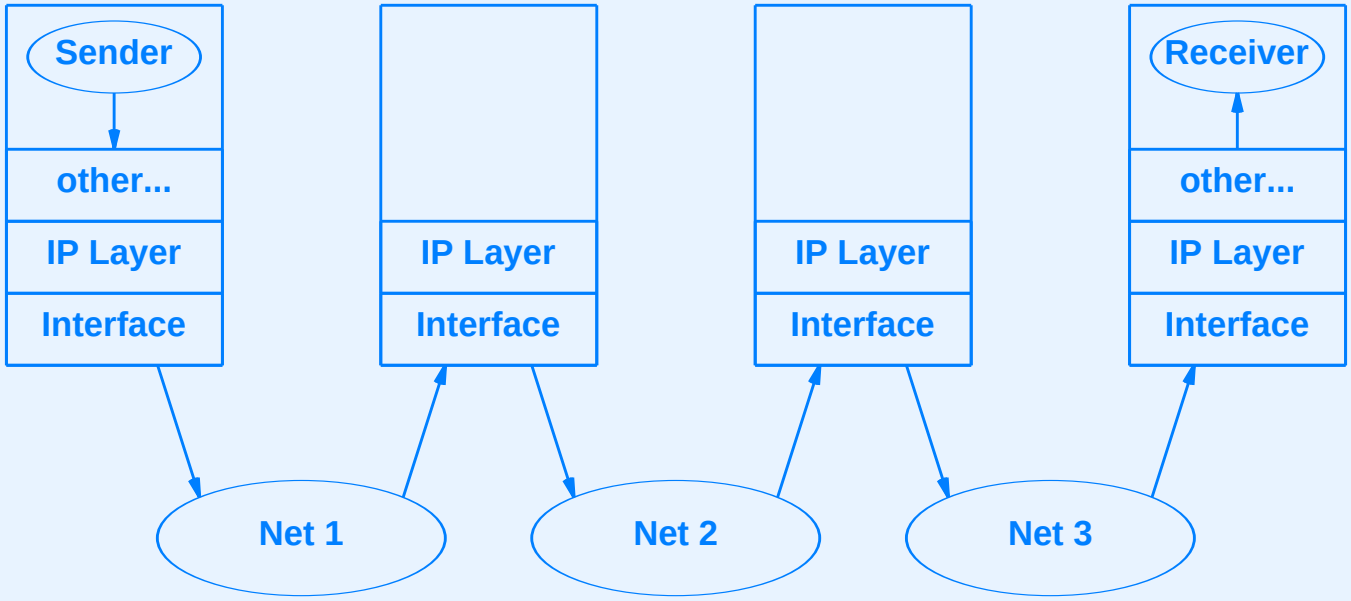
Concept Of Layering



More Realistic Layering



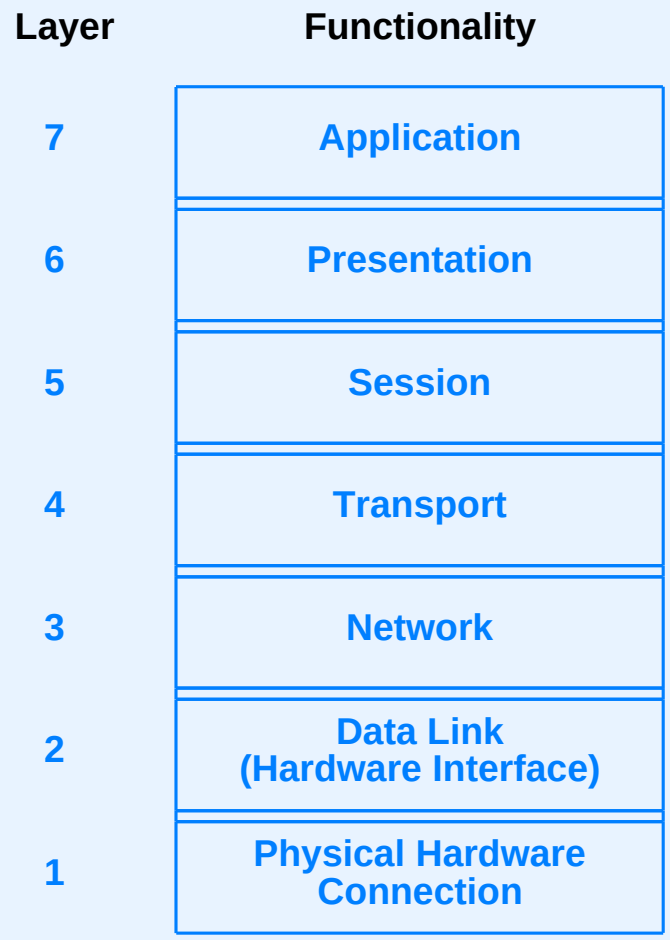
Layering In An Internet



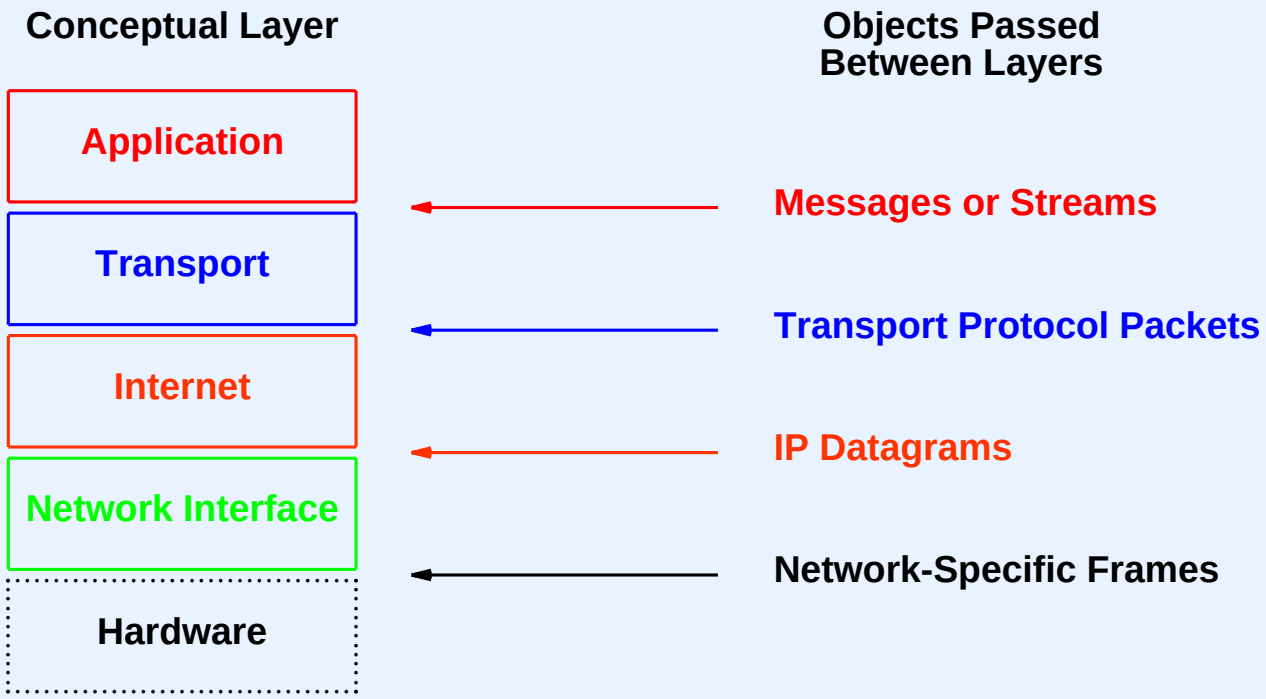
Examples Of Layering

- Two models exist
- ISO 7-layer reference model for *Open System Interconnection* (*OSI*)
  - Predates TCP/IP
  - Does not include an Internet layer
  - Prescriptive (designed before protocols)
- Internet 5-layer reference model
  - Designed for TCP/IP
  - Descriptive (designed along with actual protocols)

ISO 7-Layer Reference Model



TCP/IP 5-Layer Reference Model



- Only four layers above hardware

TCP/IP Layer 1: Physical Hardware

- Defines electrical signals used in communication (e.g., voltages on wires between two computers)
- Uninteresting except to electrical engineers

TCP/IP Layer 2: Network Interface

- Defines communication between computer and network hardware
- Isolates details of hardware (MAC) addressing
- Example protocol: ARP
- Code is usually in the operating system

TCP/IP Layer 3: Internet

- Protocol is IP
- Provides machine to machine communication
- Defines best-effort, connectionless datagram delivery service for the Internet
- Code is usually in the operating system

TCP/IP Layer 4: Transport

- Provides end-to-end connection from application program to application program
- Often handles reliability, flow control
- Protocols are TCP and UDP
- Code is usually in the operating system

TCP/IP Layer 5: Application

- Implemented by application programs
- Many application-specific protocols in the Internet
- Built on top of transport layer

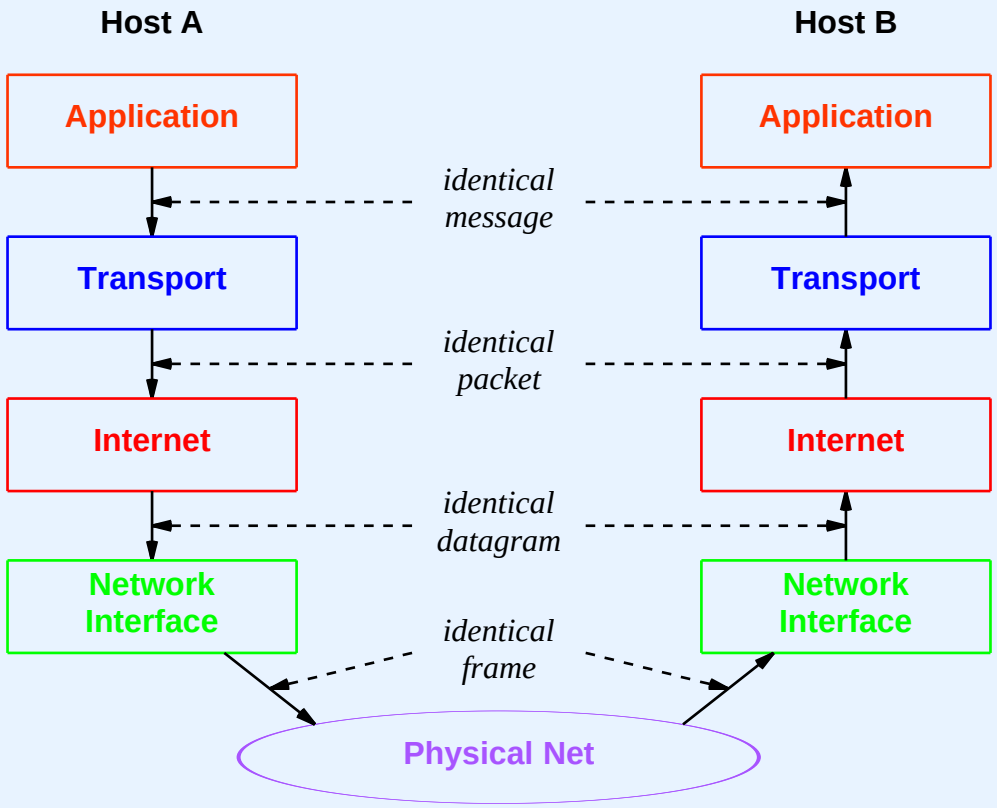
Two Differences Between TCP/IP And Other Layered Protocols

- TCP/IP uses end-to-end reliability instead of link-level reliability
- TCP/IP places the locus of intelligence and decision making at the edge of the network instead of the core

The Layering Principle

Software implementing layer n at the destination receives exactly the message sent by software implementing layer n at the source.

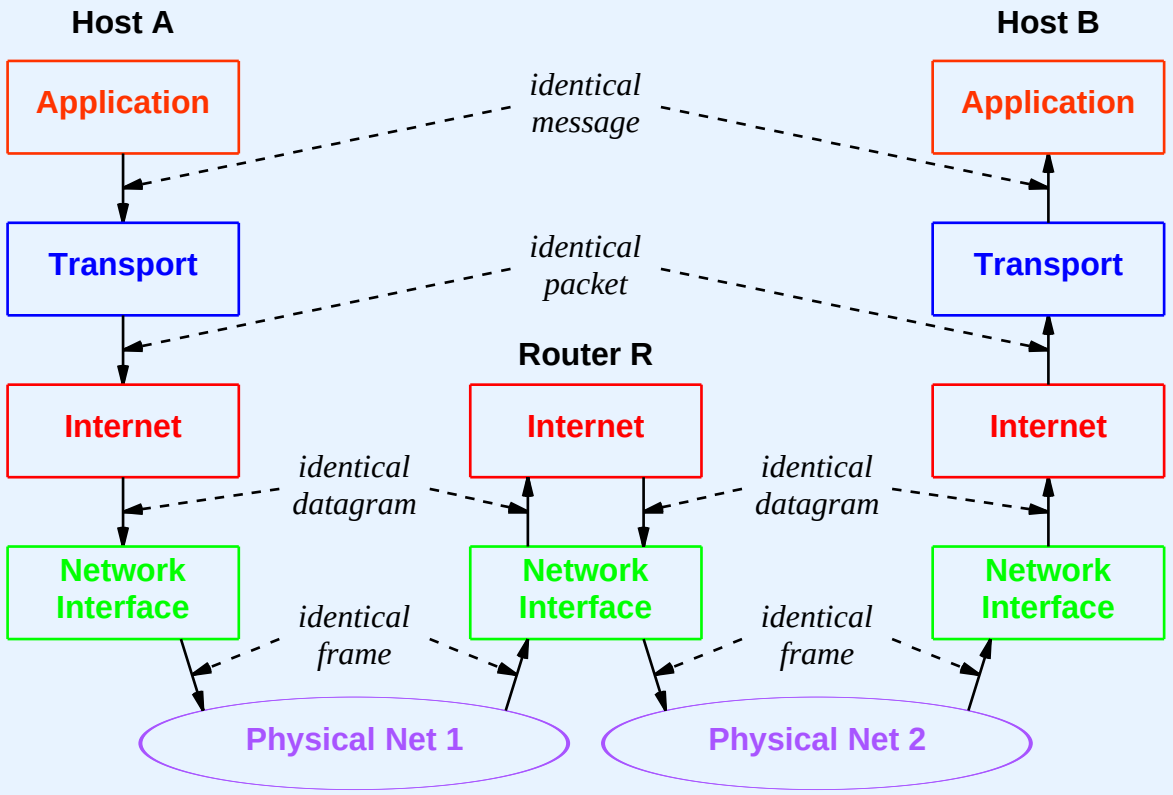
Illustration Of Layering Principle



When A Datagram Traverses The Internet

- All layers involved at
  - Original source
  - Ultimate destination
- Only up through IP layer involved at
  - Intermediate routers

Illustration Of Layering In An Internet



A Key Definition

- A protocol is classified as *end-to-end* if the layering principle applies from one end of the Internet to the other
- Examples
  - IP is *machine-to-machine* because layering principle only applies across one hop
  - TCP is *end-to-end* because layering principle from original source to ultimate destination

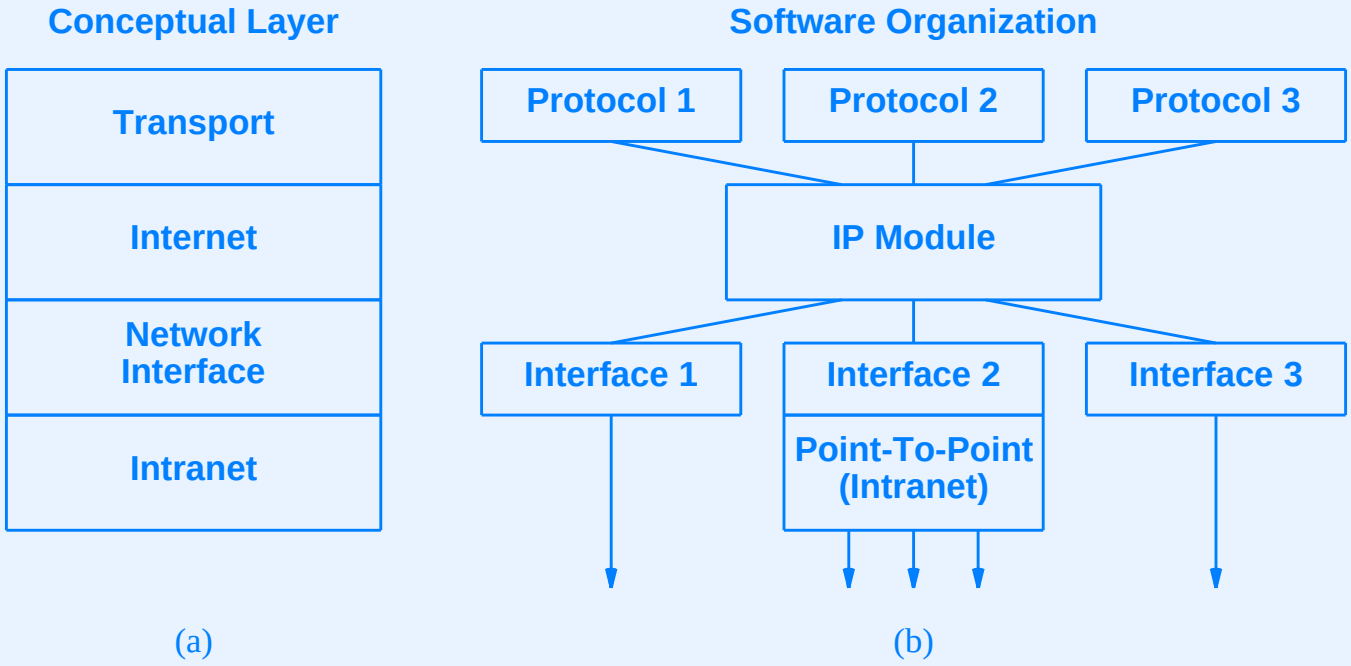
Practical Aspect Of Layering

- Multiple protocols at each layer
- One protocol used at each layer for given datagram

Example Of Two Protocols At Network Interface Layer: SLIP And PPP

- Both used to send IP across
  - Serial data circuit
  - Dialup connection
- Each defines standards for
  - Framing (encapsulation)
  - Addressing
- Incompatible

Notion Of Multiple Interfaces And Layering



(a) (b)

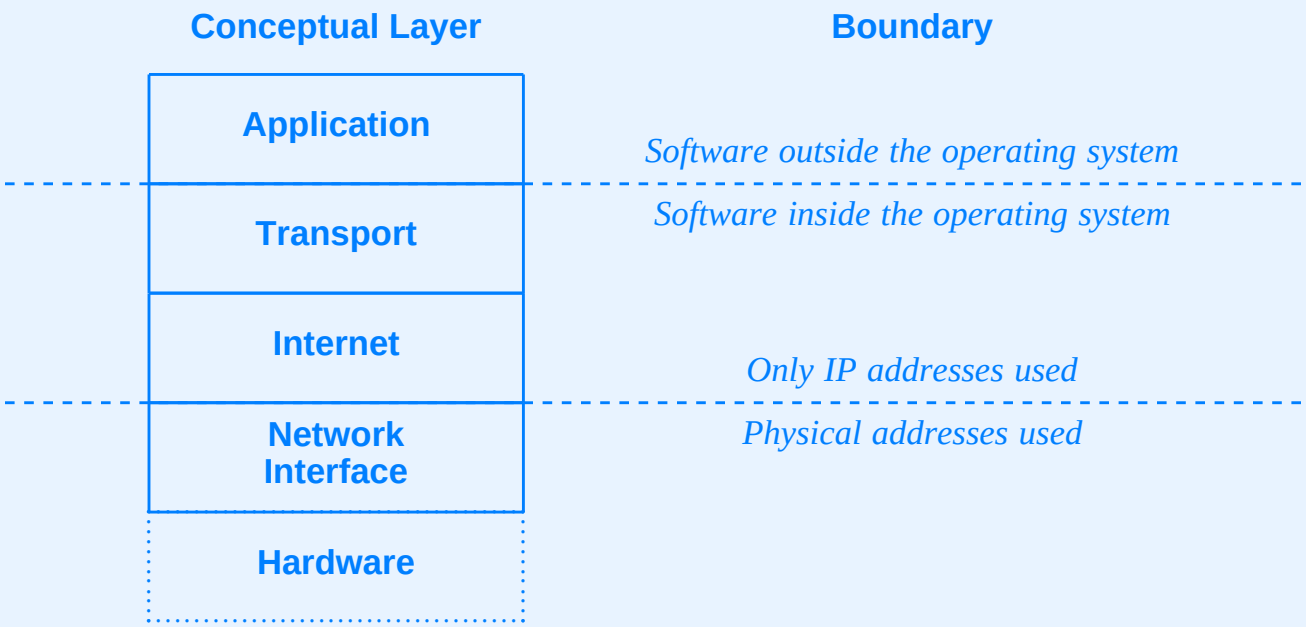
Boundaries In The TCP/IP Layering Model

- High-level protocol address boundary
  - Division between software that uses hardware addresses and software that uses IP addresses
- Operating system boundary
  - Division between application program running outside the operating system and protocol software running inside the operating system

The Consequence Of An Address Boundary

Application programs as well as all protocol software from the Internet layer upward use only IP addresses; the network interface layer handles physical addresses.

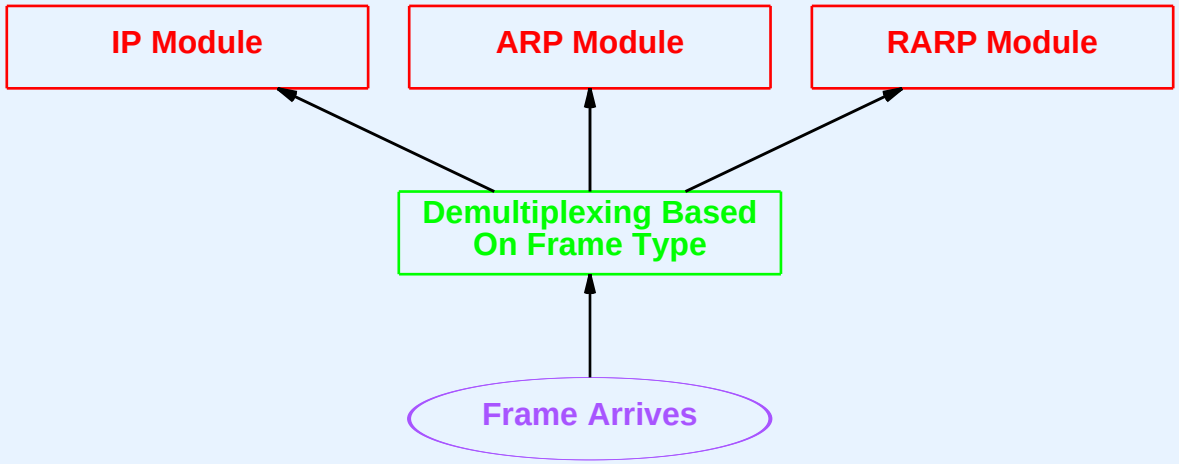
Illustration Of The Two Boundaries



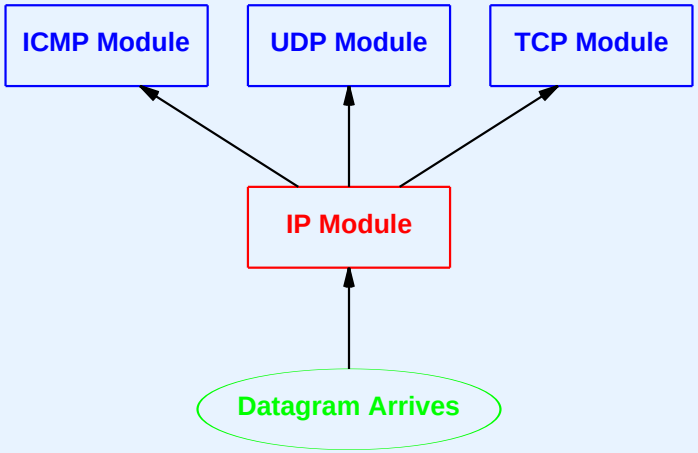
Handling Multiple Protocols Per Layer

- Sender places field in header to say which protocol used at each layer
- Receiver uses field to determine which protocol at next layer receives the packet
- Known as *multiplexing* and *demultiplexing*

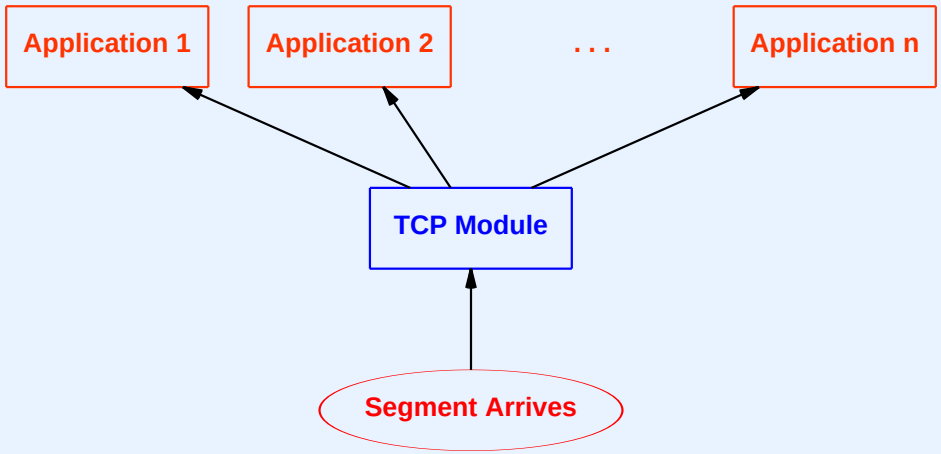
Example Of Demultiplexing An Incoming Frame



Example Of Demultiplexing Performed By IP



Example Of Demultiplexing Performed By TCP



- TCP is part of operating system
- Transfer to application program must cross operating system boundary

Discussion

- What are the key advantages and disadvantages of multiplexing / demultiplexing?
- Can you think of an alternative?

Summary

- Layering
  - Intended for designers
  - Helps control complexity in protocol design
- TCP/IP uses 5-layer reference model
- Conceptually, a router only needs layers 2 and 3, and a host needs all layers
- IP is machine-to-machine protocol
- TCP is end-to-end protocol
- Demultiplexing used to handle multiple protocols at each layer


Questions?

PART XI

USER DATAGRAM PROTOCOL (UDP)

Identifying The Ultimate Destination

- IP address only specifies a computer
- Need a way to specify an application program (process) on a computer
- Unfortunately
  - Application programs can be created and destroyed rapidly
  - Each operating system uses its own identification

Specifying An Application Program

- TCP/IP introduces its own specification
- Abstract destination point known as *protocol port number* (positive integer)
- Each OS determines how to bind protocol port number to specific application program

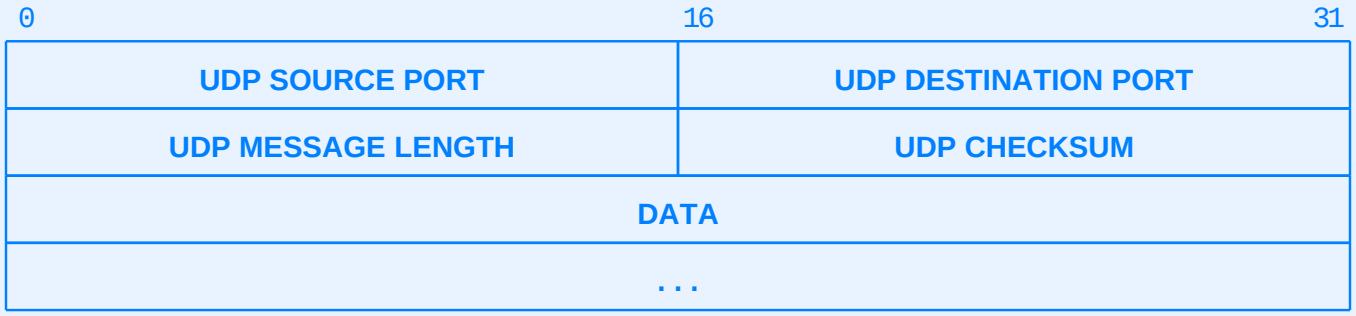
User Datagram Protocol

- Transport-layer protocol (Layer 4)
- Connectionless service: provides application programs with ability to send and receive messages
- Allows multiple, application programs on a single machine to communicate concurrently
- Same best-effort semantics as IP
  - Message can be delayed, lost, or duplicated
  - Messages can arrive out of order
- Application accepts full responsibility for errors

The Added Benefit Of UDP

The User Datagram Protocol (UDP) provides an unreliable connectionless delivery service using IP to transport messages between machines. It uses IP to carry messages, but adds the ability to distinguish among multiple destinations within a given host computer.

UDP Message Format



- If *UDP CHECKSUM* field contains zeroes, receiver does not verify the checksum

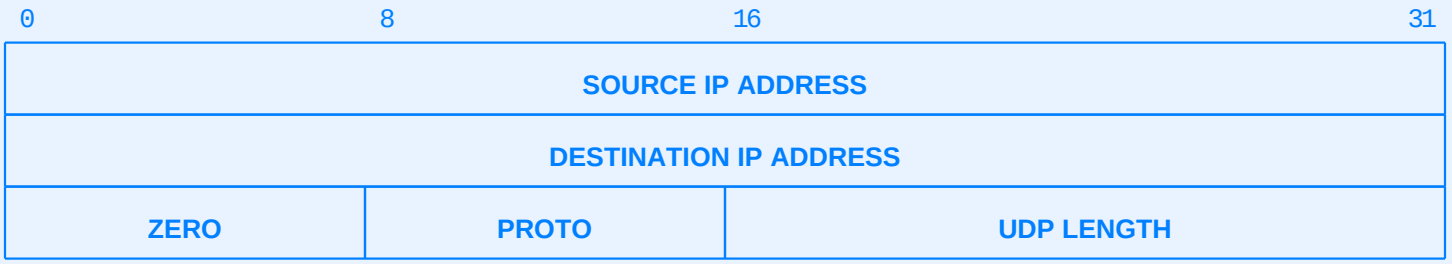
Port Numbers In A UDP Message

- SOURCE PORT identifies application on original source computer
- DESTINATION PORT identifies application on ultimate destination computer
- Note: IP addresses of source and destination do not appear explicitly in header

UDP Pseudo-Header

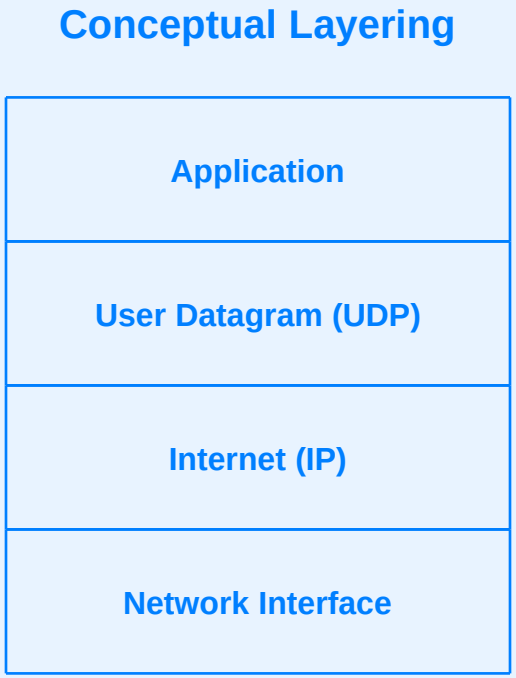
- Used when computing or verifying a checksum
- Temporarily prepended to UDP message
- Contains items from IP header
- Guarantees that message arrived at correct destination
- Note: pseudo header is *not* sent across Internet

Contents Of UDP Pseudo-Header



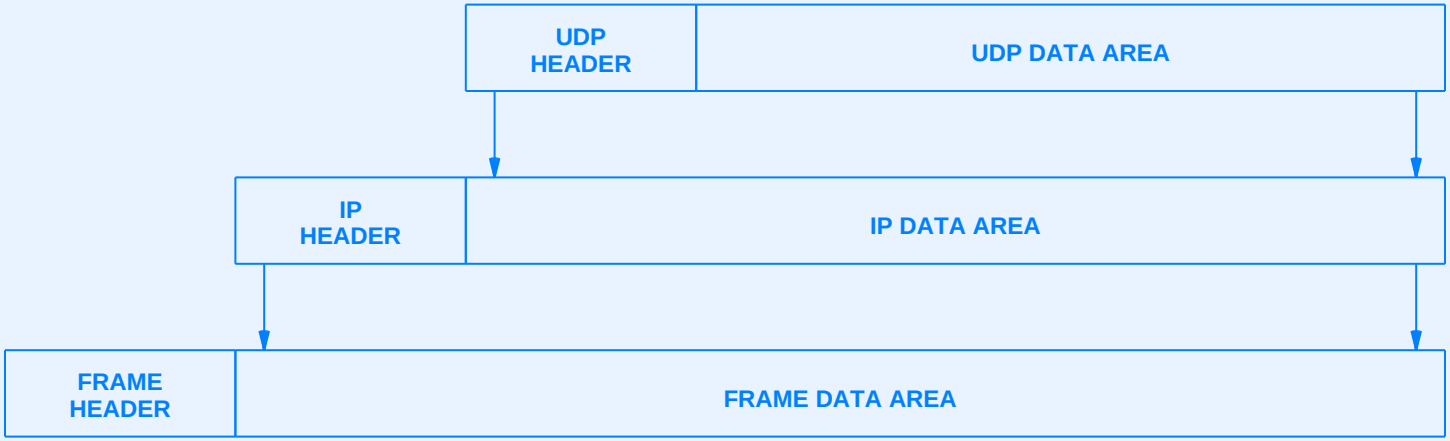
- SOURCE ADDRESS and DESTINATION ADDRESS specify IP address of sending and receiving computers
- PROTO contains the Type from the IP datagram header

Position Of UDP In Protocol Stack



- UDP lies between applications and IP

Encapsulation

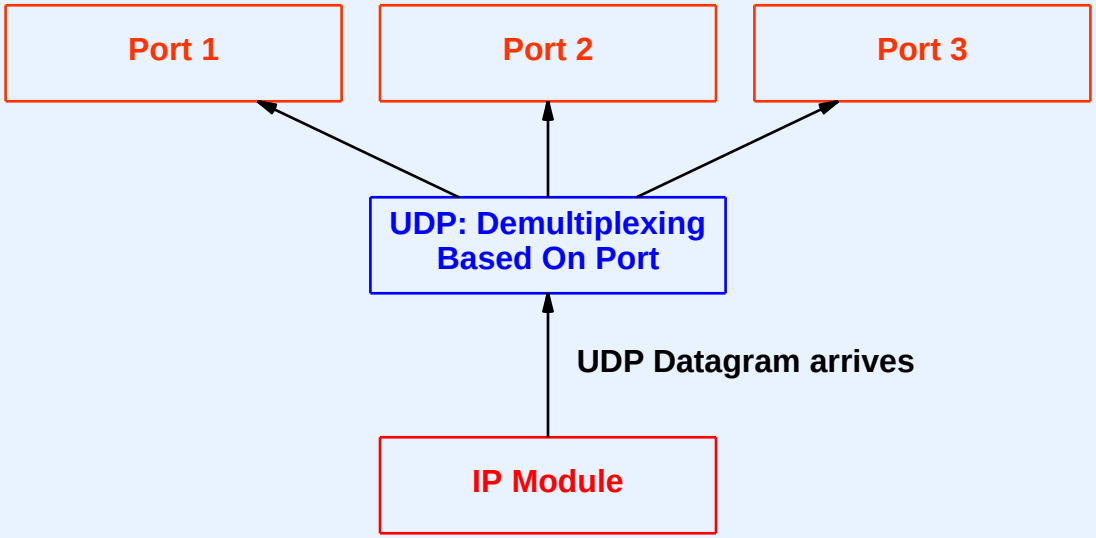


Division Of Duties Between IP and UDP

The IP layer is responsible for transferring data between a pair of hosts on an internet, while the UDP layer is responsible for differentiating among multiple sources or destinations within one host.

- IP header only identifies computer
- UDP header only identifies application programs

Demultiplexing Based On UDP Protocol Port Number



Assignment Of UDP Port Numbers

- Small numbers reserved for specific services
  - Called *well-known ports*
  - Same interpretation throughout the Internet
  - Used by server software
- Large numbers not reserved
  - Available to arbitrary application program
  - Used by client software
- More later in the course

Examples Of Assigned UDP Port Numbers

Decimal	Keyword	UNIX Keyword	Description
0	-	-	Reserved
7	ECHO	echo	Echo
9	DISCARD	discard	Discard
11	USERS	systat	Active Users
13	DAYTIME	daytime	Daytime
15	-	netstat	Network Status Program
17	QUOTE	qotd	Quote of the Day
19	CHARGEN	chargen	Character Generator
37	TIME	time	Time
42	NAMESERVER	name	Host Name Server
43	NICNAME	whois	Who Is
53	DOMAIN	nameserver	Domain Name Server
67	BOOTPS	bootps	BOOTP or DHCP Server
68	BOOTPC	bootpc	BOOTP or DHCP Client
69	TFTP	tftp	Trivial File Transfer
88	KERBEROS	kerberos	Kerberos Security Service
111	SUNRPC	sunrpc	Sun Remote Procedure Call
123	NTP	ntp	Network Time Protocol
161	-	snmp	Simple Network Management Protocol
162	-	snmp-trap	SNMP traps
512	-	biff	UNIX comsat
513	-	who	UNIX rwho Daemon
514	-	syslog	System Log
525	-	timed	Time Daemon

Summary

- User Datagram Protocol (UDP) provides connectionless, best-effort message service
- UDP message encapsulated in IP datagram for delivery
- IP identifies destination computer; UDP identifies application on the destination computer
- UDP uses abstraction known as *protocol port numbers*


Questions?

PART XII

RELIABLE STREAM TRANSPORT SERVICE (TCP)

Transmission Control Protocol (TCP)

- Major transport service in the TCP/IP suite
- Used for most Internet applications (esp. World Wide Web)

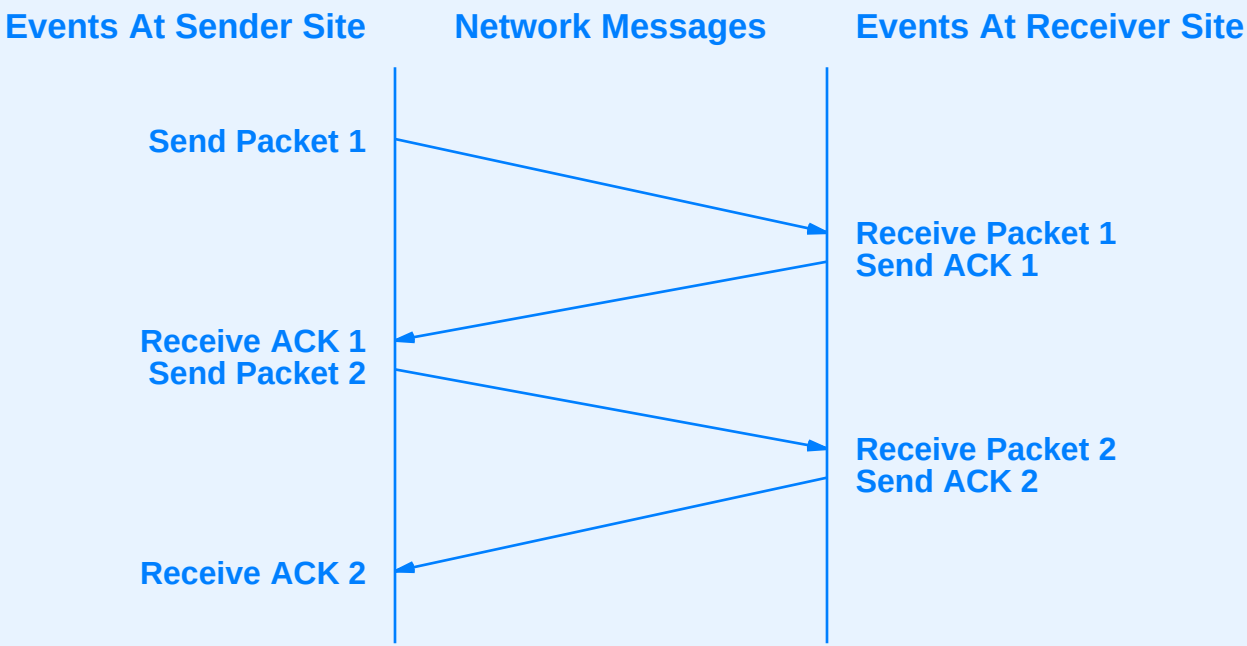
TCP Characteristics

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

Providing Reliability

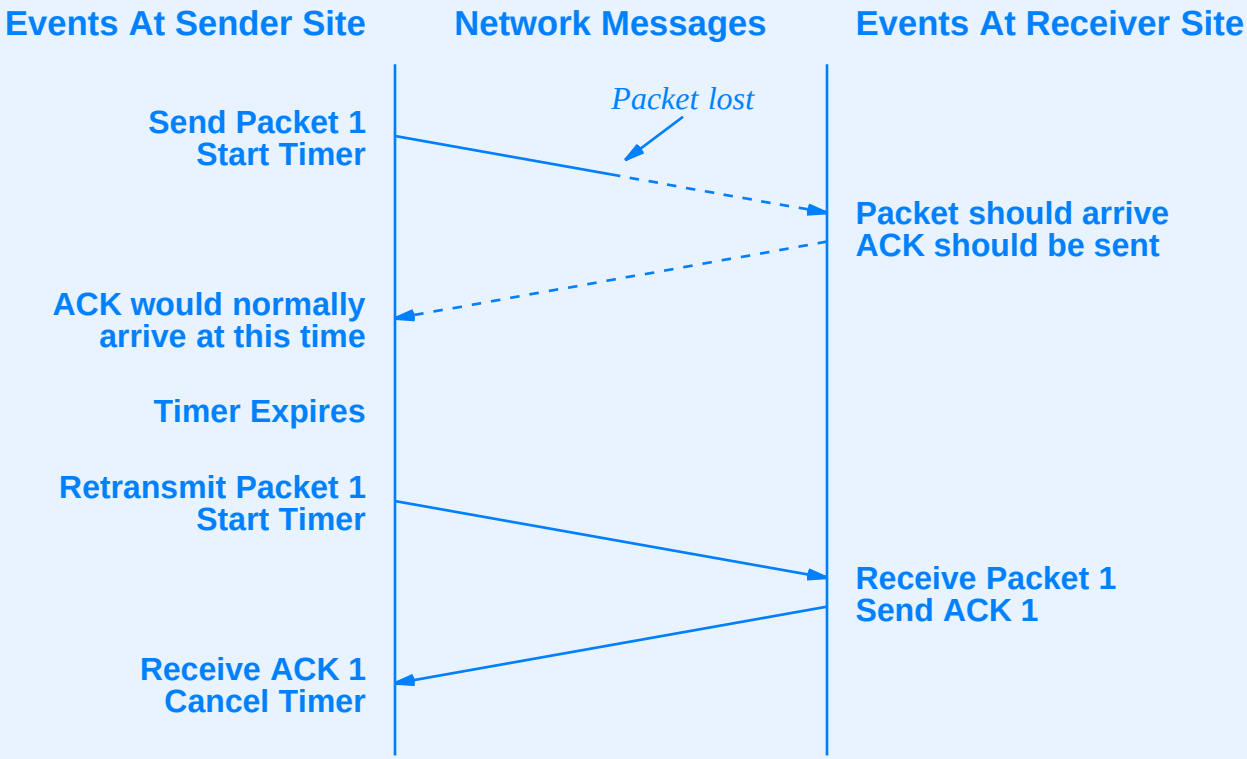
- Traditional technique: Positive Acknowledgement with Retransmission (PAR)
  - Receiver sends *acknowledgement* when data arrives
  - Sender starts timer whenever transmitting
  - Sender retransmits if timer expires before acknowledgement arrives

Illustration Of Acknowledgements



- Time moves from top to bottom in the diagram

Illustration Of Recovery After Packet Loss



The Problem With Simplistic PAR

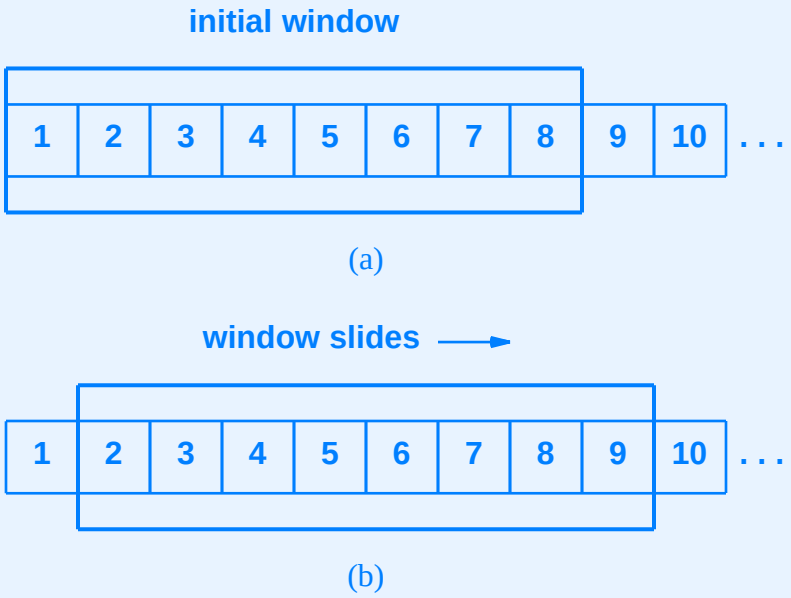
A simple positive acknowledgement protocol wastes a substantial amount of network bandwidth because it must delay sending a new packet until it receives an acknowledgement for the previous packet.

- Problem is especially severe if network has long latency

Solving The Problem

- Allow multiple packets to be outstanding at any time
- Still require acknowledgements and retransmission
- Known as *sliding window*

Illustration Of Sliding Window

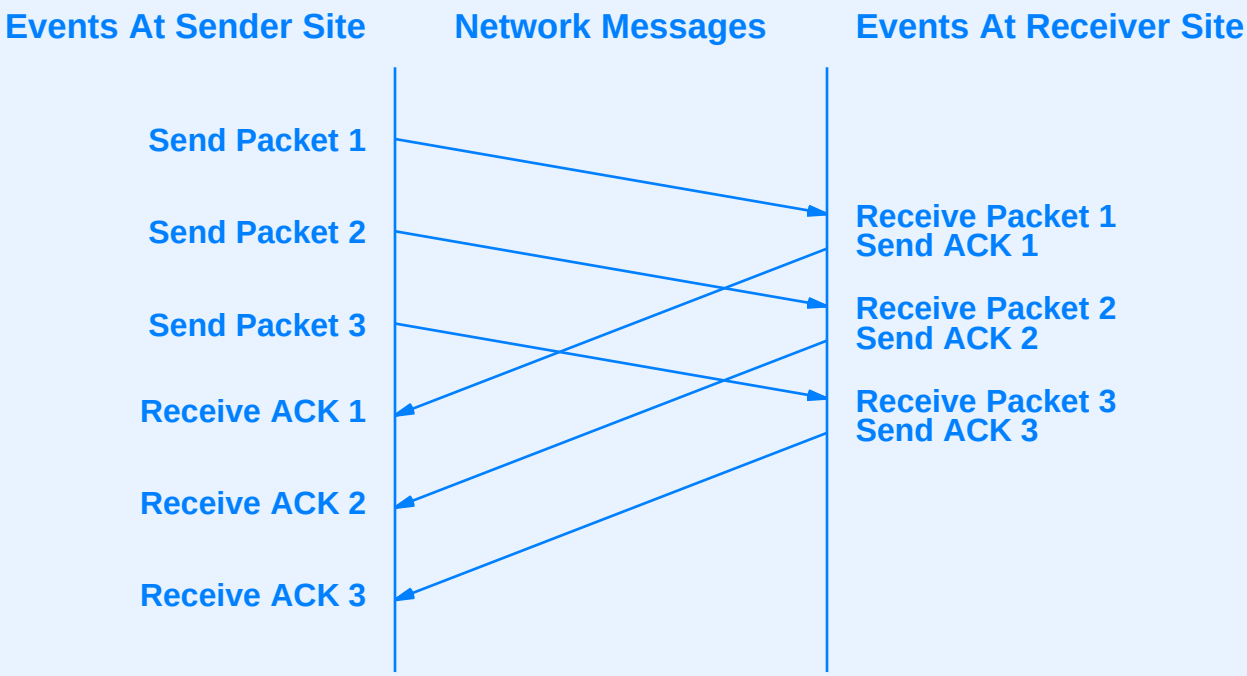


- Window size is fixed
- As acknowledgement arrives, window moves forward

Why Sliding Window Works

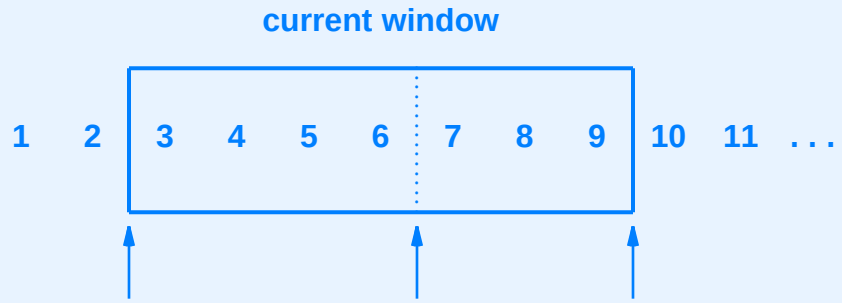
Because a well-tuned sliding window protocol keeps the network completely saturated with packets, it obtains substantially higher throughput than a simple positive acknowledgement protocol.

Illustration Of Sliding Window



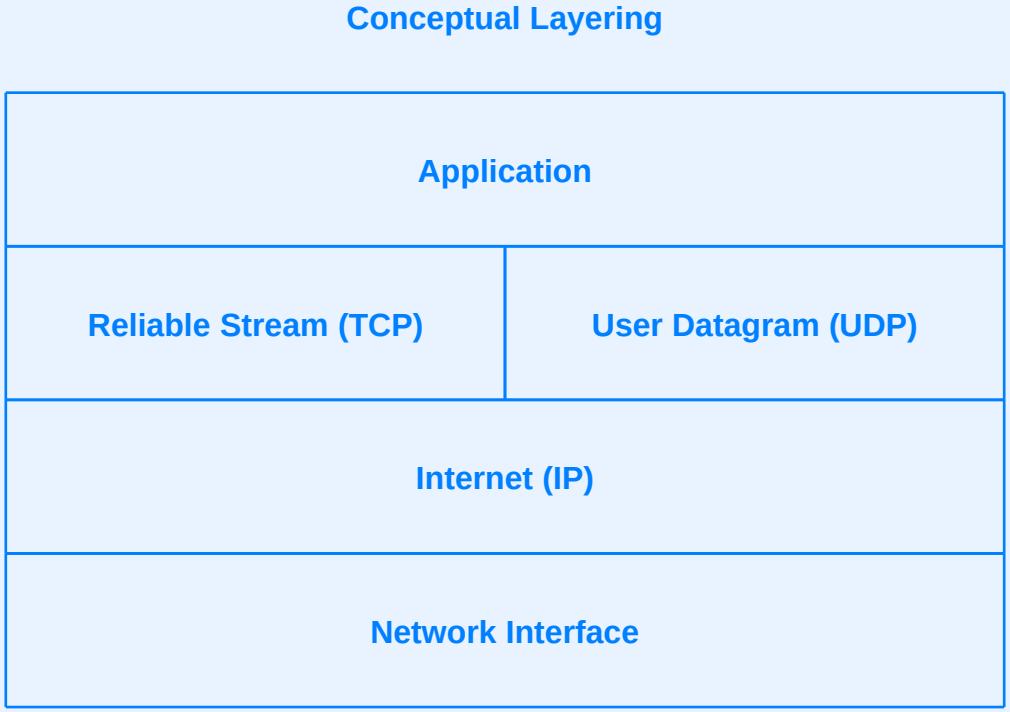
Sliding Window Used By TCP

- Measured in byte positions
- Illustration



- Bytes through 2 are acknowledged
- Bytes 3 through 6 not yet acknowledged
- Bytes 7 though 9 waiting to be sent
- Bytes above 9 lie outside the window and cannot be sent

Layering Of The Three Major Protocols



TCP Ports, Connections, And Endpoints

- Endpoint of communication is application program
- TCP uses protocol port number to identify application
- TCP connection between two endpoints identified by four items
  - Sender's IP address
  - Sender's protocol port number
  - Receiver's IP address
  - Receiver's protocol port number

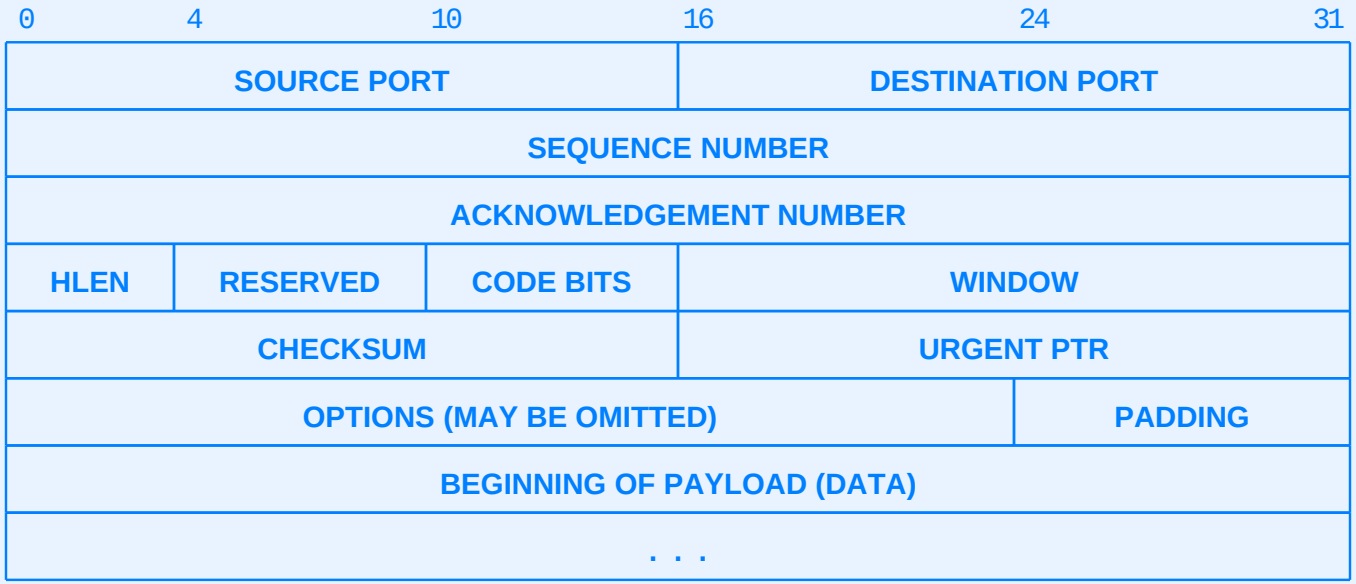
An Important Idea About Port Numbers

Because TCP identifies a connection by a pair of endpoints, a given TCP port number can be shared by multiple connections on the same machine.

Passive And Active Opens

- Two sides of a connection
- One side waits for contact
  - A server program
  - Uses TCP's *passive open*
- One side initiates contact
  - A client program
  - Uses TCP's *active open*

TCP Segment Format



- Offset specifies header size (offset of data) in 32-bit words

Code Bits In The TCP Segment Header

Bit (left to right)	Meaning if bit set to 1
URG	Urgent pointer field is valid
ACK	Acknowledgement field is valid
PSH	This segment requests a push
RST	Reset the connection
SYN	Synchronize sequence numbers
FIN	Sender has reached end of its byte stream

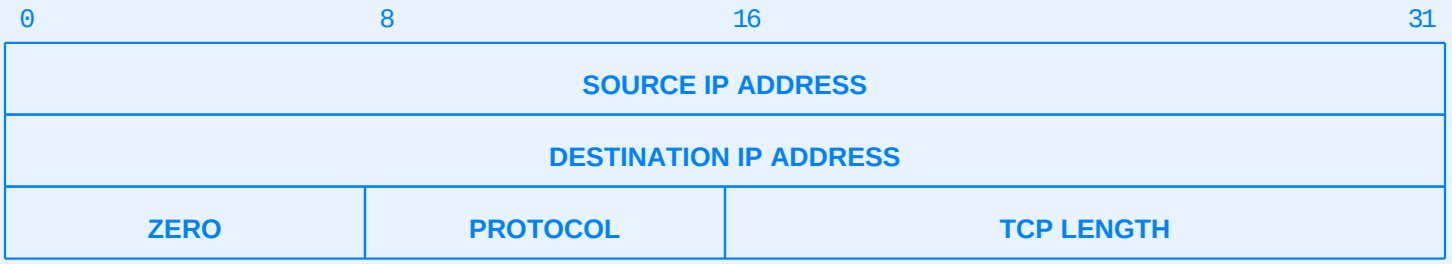
Flow Control And TCP Window

- Receiver controls flow by telling sender size of currently available buffer measured in bytes
- Called *window advertisement*
- Each segment, including data segments, specifies size of window *beyond acknowledged byte*
- Window size may be zero (receiver cannot accept additional data at present)
- Receiver can send additional acknowledgement later when buffer space becomes available

TCP Checksum Computation

- Covers entire segment (header plus data)
- Required (unlike UDP)
- Pseudo header included in computation as with UDP

TCP Pseudo Header



TCP Retransmission

- Designed for Internet environment
  - Delays on one connection vary over time
  - Delays vary widely between connections
- Fixed value for timeout will fail
  - Waiting too long introduces unnecessary delay
  - Not waiting long enough wastes network bandwidth with unnecessary retransmission
- Retransmission strategy must be adaptive

Adaptive Retransmission

- TCP keeps estimate of round-trip time (RTT) on each connection
- Round-trip estimate derived from observed delay between sending segment and receiving acknowledgement
- Timeout for retransmission based on current round-trip estimate

Difficulties With Adaptive Retransmission

- The problem is knowing when to retransmit
- Segments or ACKs can be lost or delayed, making round-trip estimation difficult or inaccurate
- Round-trip times vary over several orders of magnitude between different connections
- Traffic is bursty, so round-trip times fluctuate wildly on a single connection

Difficulties With Adaptive Retransmission (continued)

- Load imposed by a single connection can congest routers or networks
- Retransmission can *cause* congestion
- Because an internet contains diverse network hardware technologies, there may be little or no control for intra-network congestion

Solution: Smoothing

- Adaptive retransmission schemes keep a statistically smoothed round-trip estimate
- Smoothing keeps running average from fluctuating wildly, and keeps TCP from overreacting to change
- Difficulty: choice of smoothing scheme

Original Smoothing Scheme

- Let RTT be current (old) average round-trip time
- Let NRT be a new sample
- Compute

$$RTT = \alpha * RTT + \beta * NRT$$

  where

$$\alpha + \beta = 1$$

- Example: $\alpha = .8$, $\beta = .2$
- Large α makes estimate less susceptible to a single long delay (more stable)
- Large β makes estimate track changes in round-trip time quickly

Problems With Original Scheme

- Associating ACKs with transmissions
  - TCP acknowledges receipt of data, not receipt of transmission
  - Assuming ACK corresponds to most recent transmission can cause instability in round-trip estimate (Cypress syndrome)
  - Assuming ACK corresponds to first transmission can cause unnecessarily long timeout
  - Both assumptions lead to lower throughput

Partridge / Karn Scheme†

- Solves the problem of associating ACKs with correct transmission
- Specifies ignoring round-trip time samples that correspond to retransmissions
- Separates timeout from round-trip estimate for retransmitted packets

†Also called *Karn's Algorithm*

Partridge / Karn Scheme (continued)

- Starts (as usual) with retransmission timer as a function of round-trip estimate
- Doubles retransmission timer value for each retransmission without changing round-trip estimate
- Resets retransmission timer to be function of round-trip estimate when ACK arrives for nonretransmitted segment

Flow Control And Congestion

- Receiver advertises window that specifies how many additional bytes it can accept
- Window size of zero means sender must not send normal data (ACKs and urgent data allowed)
- Receiver can never decrease window beyond previously advertised point in sequence space
- Sender chooses effective window smaller than receiver's advertised window if congestion detected

Jacobson / Karels Congestion Control

- Assumes long delays (packet loss) due to congestion
- Uses successive retransmissions as measure of congestion
- Reduces effective window as retransmissions increase
- Effective window is minimum of receiver's advertisement and computed quantity known as the *congestion window*

Multiplicative Decrease

- In steady state (no congestion), the congestion window is equal to the receiver's window
- When segment lost (retransmission timer expires), reduce congestion window by half
- Never reduce congestion window to less than one maximum sized segment

Jacobson / Karels Slow Start

- Used when starting traffic or when recovering from congestion
- Self-clocking startup to increase transmission rate rapidly as long as no packets are lost
- When starting traffic, initialize the congestion window to the size of a single maximum sized segment
- Increase congestion window by size of one segment each time an ACK arrives without retransmission

Jacobson / Karels Congestion Avoidance

- When congestion first occurs, record one-half of last successful congestion window (flightsize) in a *threshold* variable
- During recovery, use slow start until congestion window reaches threshold
- Above threshold, slow down and increase congestion window by one segment per window (even if more than one segment was successfully transmitted in that interval)

Jacobson / Karels Congestion Avoidance (continued)

- Increment window size on each ACK instead of waiting for complete window

  increase = segment / window

  Let *N* be segments per window, or

  N = congestion_window / max segment size

  so

  increase = segment / N

  = (MSS bytes / N)

  = MSS / (congestion_window/MSS)

  or

  increase = (MSS*MSS)/congestion_window

Changes In Delay

- Original smoothing scheme tracks the mean but not changes
- To track changes, compute

$$DIFF = SAMPLE - RTT$$
$$RTT = RTT + \delta * DIFF$$
$$DEV = DEV + \delta (|DIFF| - DEV)$$

- DEV estimates mean deviation
- δ is fraction between 0 and 1 that weights new sample
- Retransmission timer is weighted average of RTT and DEV:

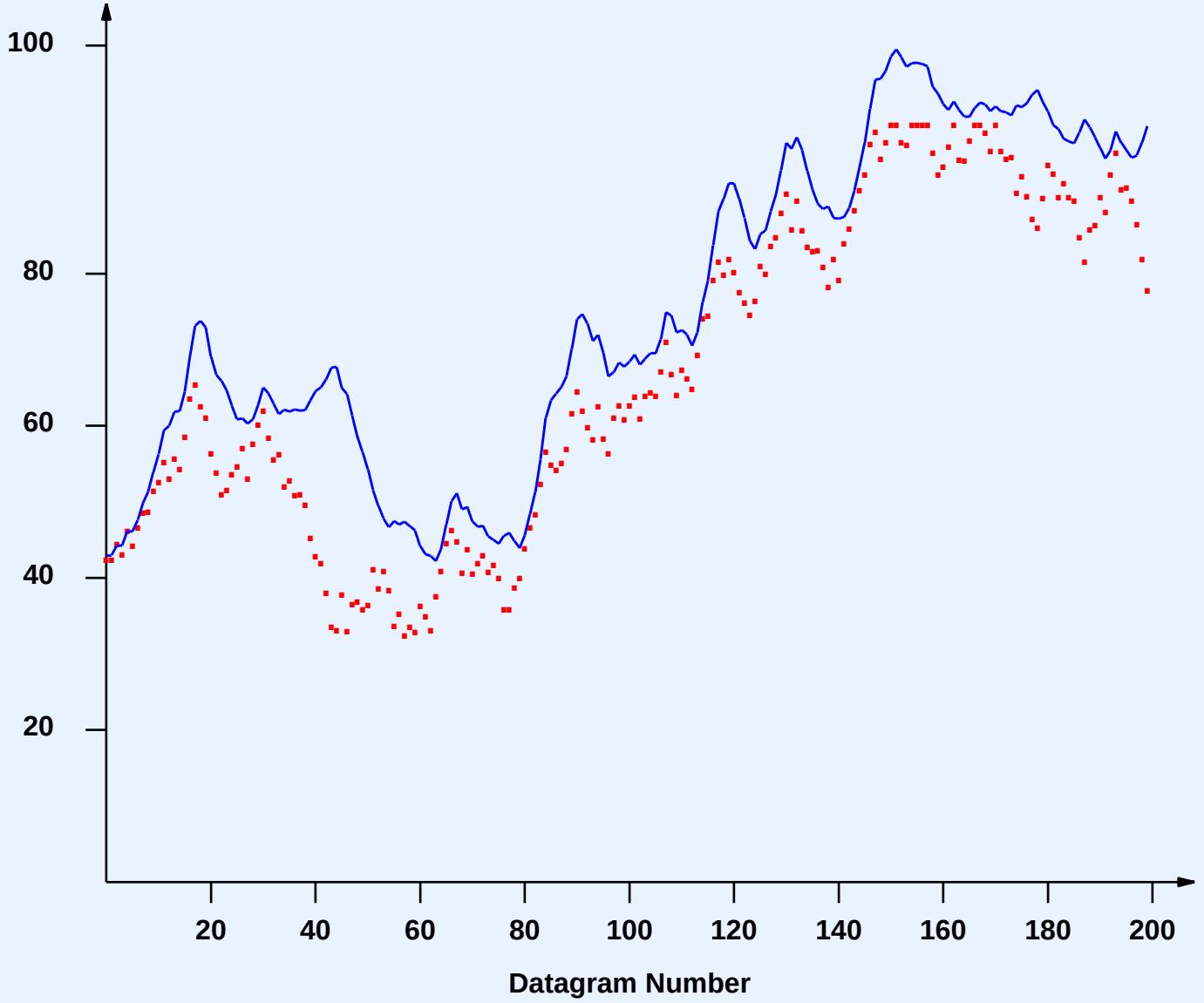
$$RTO = \mu * RTT + \phi * DEV$$

- Typically, $\mu = 1$ and $\phi = 4$

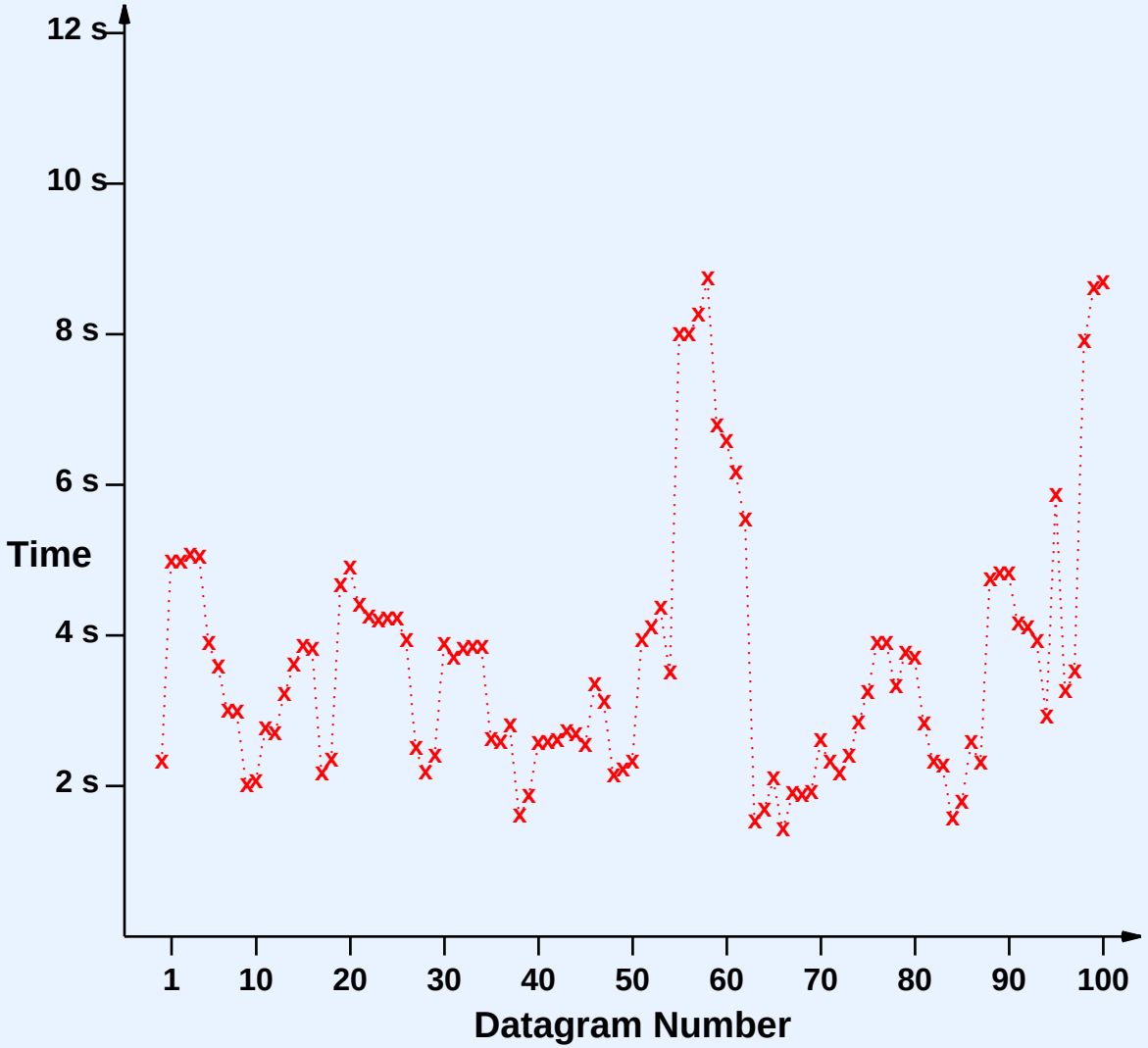
Computing Estimated Deviation

- Extremely efficient (optimized) implementation possible
  - Scale computation by 2^n
  - Use integer arithmetic
  - Choose δ to be $1/2^n$
  - Implement multiplication or division by powers of 2 with shifts
  - Research shows $n = 3$ works well

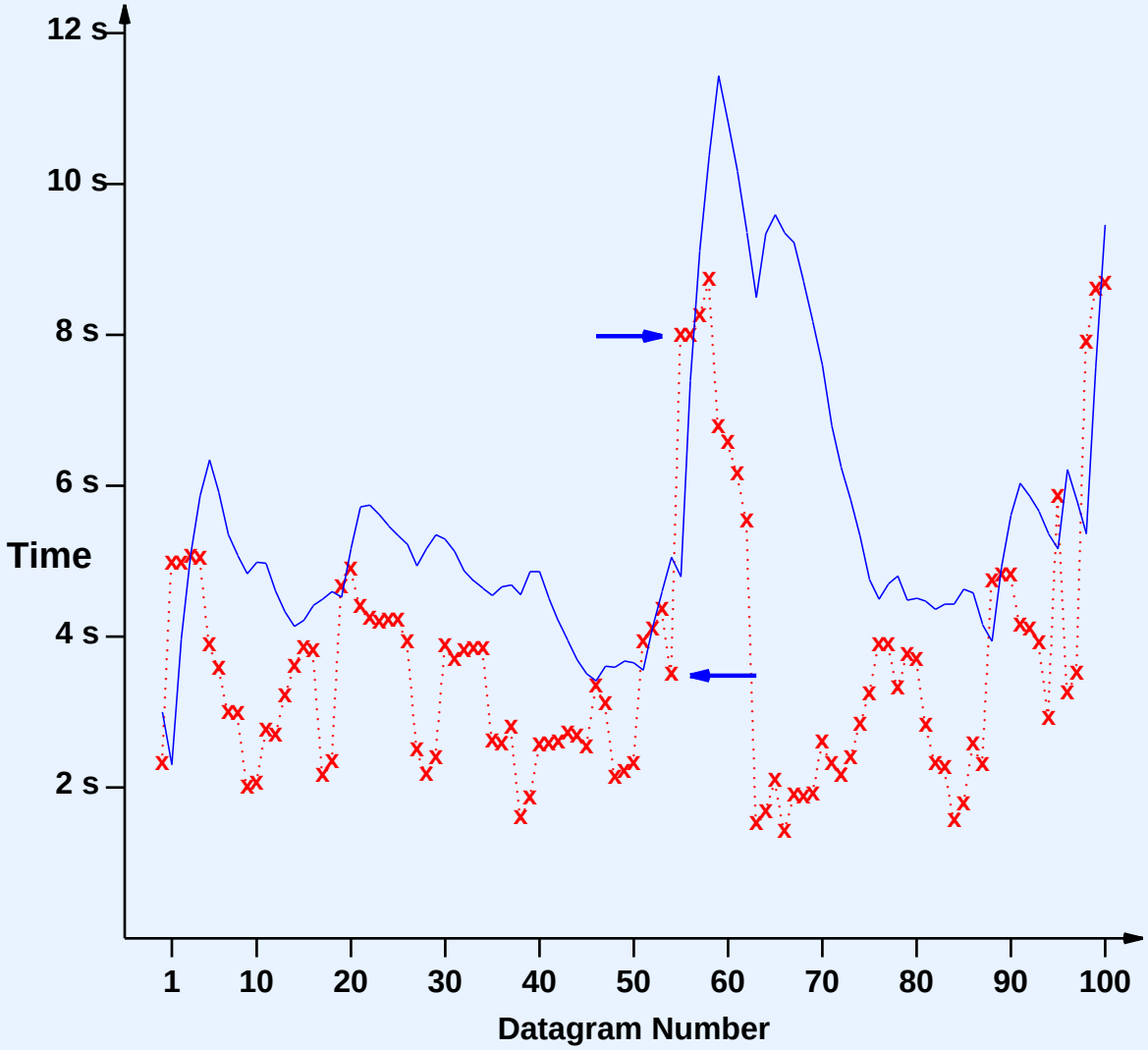
TCP Round-Trip Estimation



Measurement Of Internet Delays For 100 Successive Packets At 1 Second Intervals



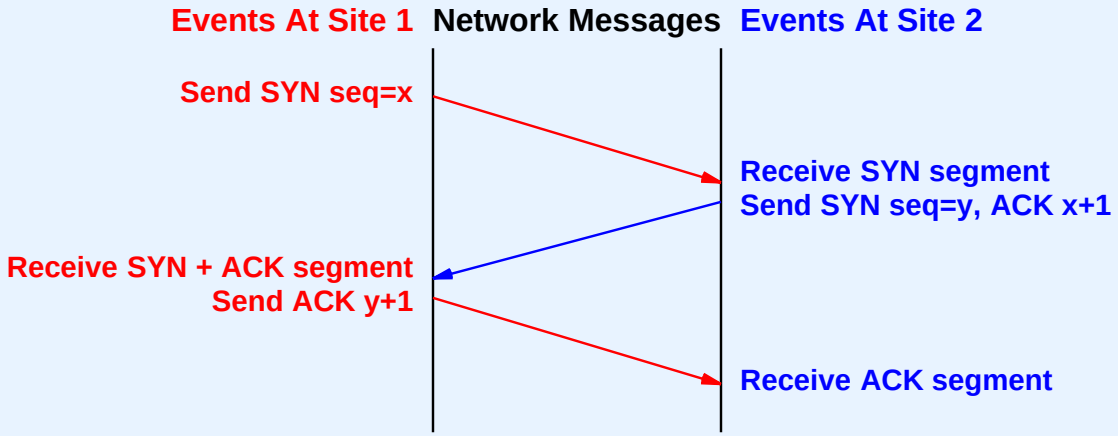
TCP Round-Trip Estimation For Sampled Internet Delays



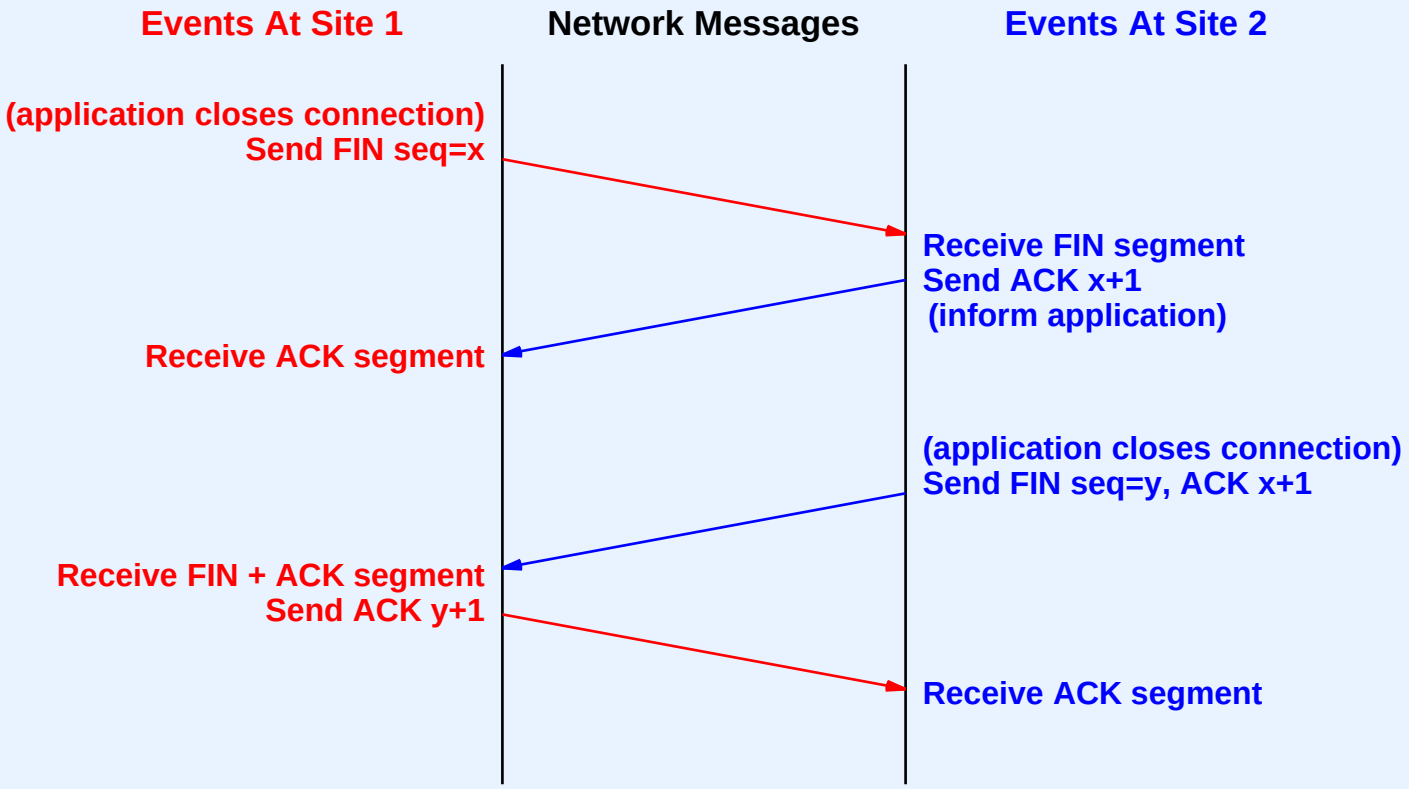
TCP Details

- Data flow may be shut down in one direction
- Connections started reliably, and terminated gracefully
- Connection established (and terminated) with a 3-way handshake

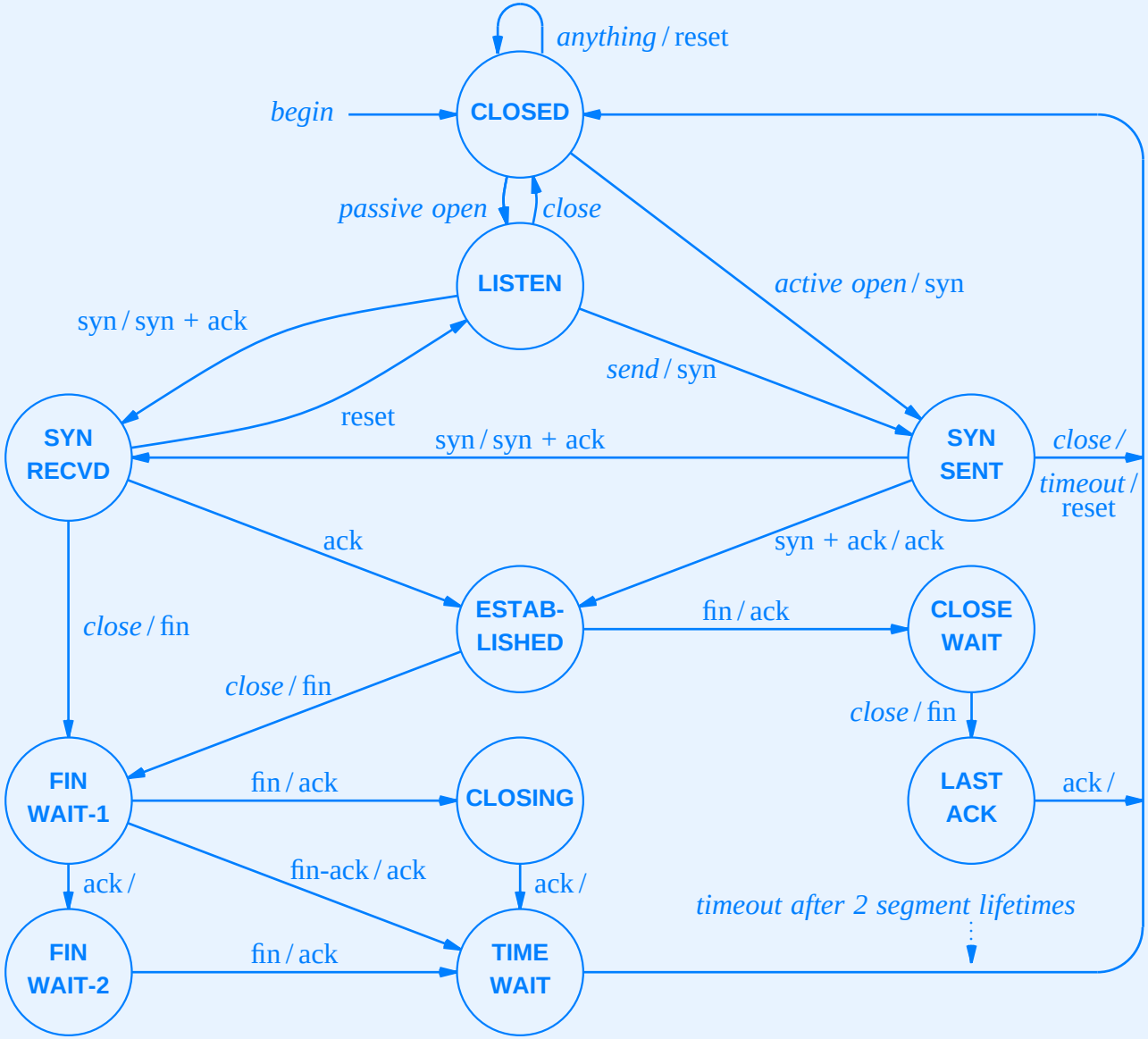
3-Way Handshake
For Connection Startup



3-Way Handshake
For Connection Shutdown



TCP Finite State Machine



TCP Urgent Data

- Segment with urgent bit set contains pointer to last octet of urgent data
- Urgent data occupies part of normal sequence space
- Urgent data can be retransmitted
- Receiving TCP should deliver urgent data to application ‘‘immediately’’ upon receipt

TCP Urgent Data
(continued)

- Two interpretations of standard
  - Out-of-band data interpretation
  - Data mark interpretation

Data-Mark Interpretation Of Urgent Data

- Has become widely accepted
- Single data stream
- Urgent pointer marks end of urgent data
- TCP informs application that urgent data arrived
- Application receives all data in sequence
- TCP informs application when end of urgent data reached

Data-Mark Interpretation Of Urgent Data (continued)

- Application
  - Reads all data from one stream
  - Must recognize start of urgent data
  - Must buffer normal data if needed later
- Urgent data marks *read* boundary

Urgent Data Delivery

- Receiving application placed in *urgent mode*
- Receiving application leaves urgent mode after reading urgent data
- Receiving application acquires *all* available urgent data when in urgent mode

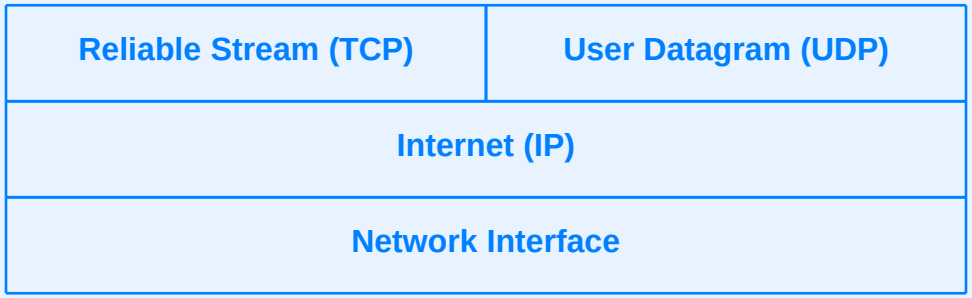
Fast Retransmit

- Coarse-grained clock used to implement RTO
  - Typically 300 to 500ms per tick
- Timer expires up to 1s after segment dropped
- Fast retransmission
  - Sender uses three duplicate ACKs as trigger
  - Sender retransmits ‘‘early’’
  - Sender reduces congestion window to half

Other TCP Details

- Silly Window Syndrome (SWS) avoidance
- Nagle algorithm
- Delayed ACKs
- For details, read the text

Comparison Of UDP And TCP



- TCP and UDP lie between applications and IP
- Otherwise, completely different

Comparison Of UDP and TCP

UDP	TCP
between apps. and IP	between apps. and IP
packets called datagrams	packets called segments
unreliable	reliable
checksum optional	checksum required
connectionless	connection-oriented
record boundaries	stream interface
intended for LAN	useful over WAN or LAN
no flow control	flow control
1-to-1, 1-many, many-1	1-to-1
allows unicast, multicast or broadcast	unicast only

TCP Vs. UDP Traffic

Around 95% of all bytes and around 85-95% of all packets on the Internet are transmitted using TCP.

– Eggert, et. al. CCR

Summary Of TCP

- Major transport service in the Internet
- Connection oriented
- Provides end-to-end reliability
- Uses adaptive retransmission
- Includes facilities for flow control and congestion avoidance
- Uses 3-way handshake for connection startup and shutdown


Questions?

PART XIII

ROUTING: CORES, PEERS, AND ALGORITHMS

Internet Routing
(review)

- IP implements datagram forwarding
- Both hosts and routers
  - Have an IP module
  - Forward datagrams
- IP forwarding is table-driven
- Table known as *routing table*

How / When Are IP Routing Tables Built?

- Depends on size / complexity of internet
- Static routing
  - Fixes routes at boot time
  - Useful only for simplest cases
- Dynamic routing
  - Table initialized at boot time
  - Values inserted / updated by protocols that propagate route information
  - Necessary in large internets

Routing Tables

- Two sources of information
  - Initialization (e.g., from disk)
  - Update (e.g., from protocols)
- Hosts tend to freeze the routing table after initialization
- Routers use protocols to learn new information and update their routing table dynamically

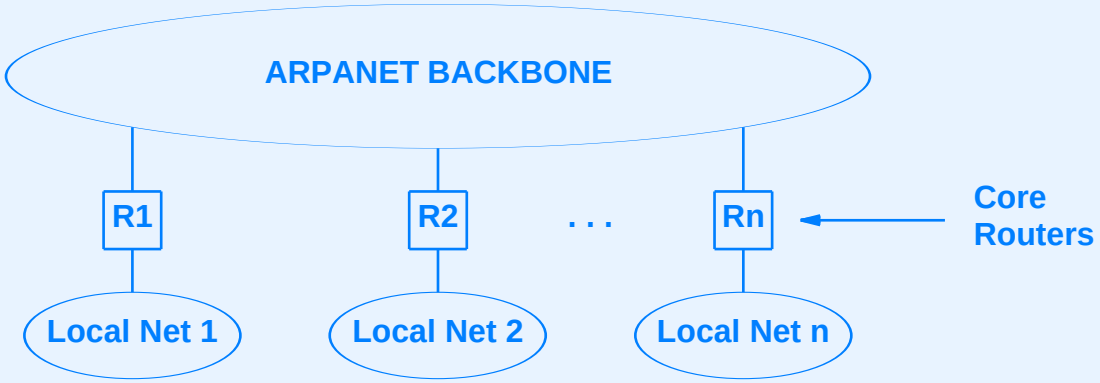
Routing With Partial Information

A host can forward datagrams successfully even if it only has partial routing information because it can rely on a router.

Routing With Partial Information
(continued)

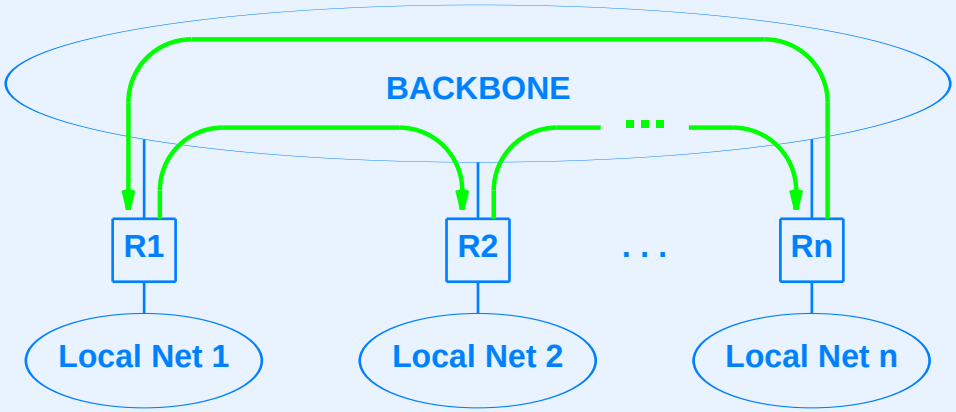
The routing table in a given router contains partial information about possible destinations. Routing that uses partial information allows sites autonomy in making local routing changes, but introduces the possibility of inconsistencies that may make some destinations unreachable from some sources.

Original Internet



- Backbone network plus routers each connecting a local network

Worst Case If All Routers Contain A Default Route

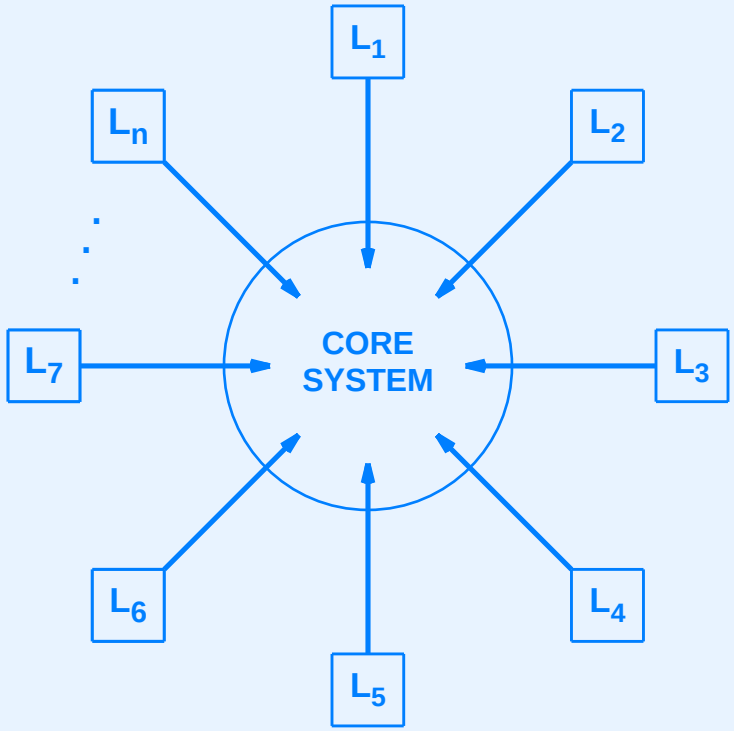


- Datagram sent to nonexistent destination loops until TTL expires

Original Routing Architecture

- Small set of ‘‘core’’ routers with complete information about all destinations
- Other routers know local destinations and use the core as central router

Illustration Of Default Routes In The Original Internet Core



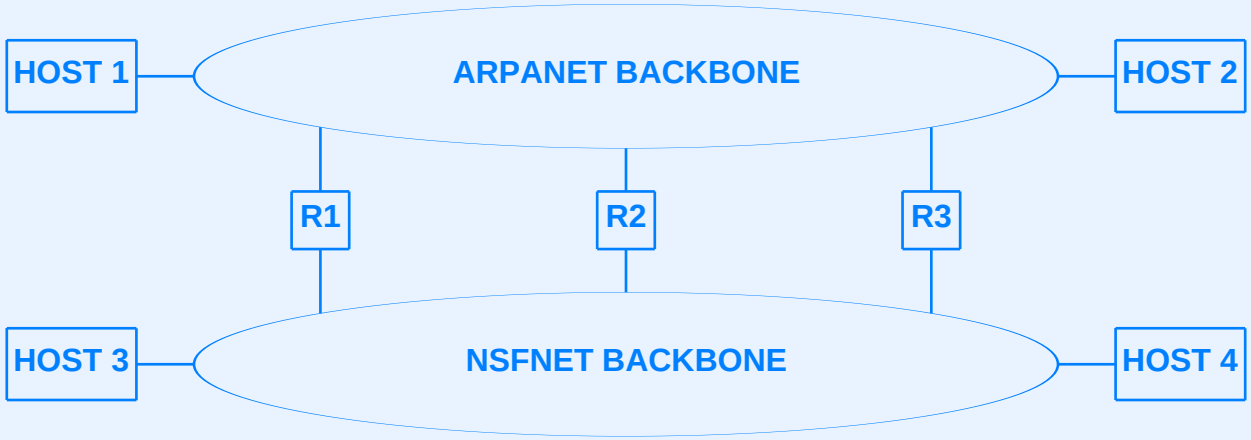
Disadvantage Of Original Core

- Central bottleneck for all traffic
- No shortcut routes possible
- Does not scale

Beyond A Core Architecture

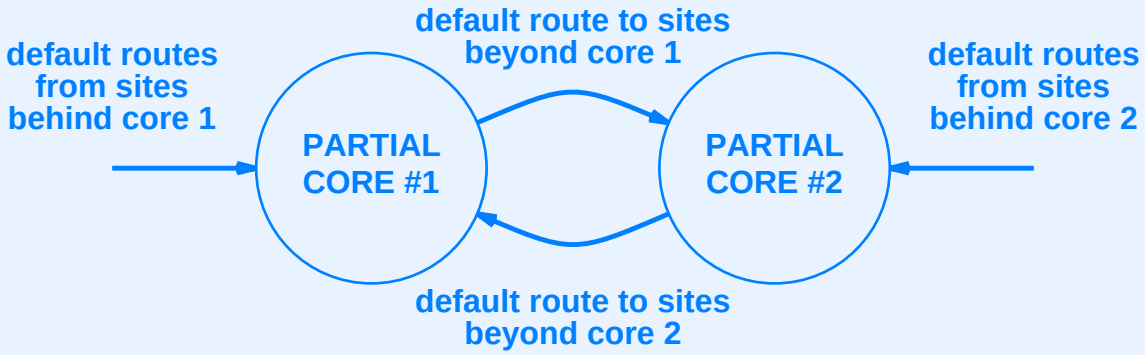
- Single core insufficient in world where multiple ISPs each have a wide-area backbone
- Two backbones first appeared when NSF and ARPA funded separate backbone networks
- Known as *peer backbones*

Illustration Of Peer Backbones



Partial Core

- Cannot have ‘‘partial core’’ scheme
- Proof:



- Datagram destined for nonexistent destination loops until TTL expires

When A Core Routing Architecture Works

A core routing architecture assumes a centralized set of routers serves as the repository of information about all possible destinations in an internet. Core systems work best for internets that have a single, centrally managed backbone. Expanding the topology to multiple backbones makes routing complex; attempting to partition the core architecture so that all routers use default routes introduces potential routing loops.

General Idea

- Have a set of core routers know routes to all locations
- Devise a mechanism that allows other routers to contact the core to learn routes (spread necessary routing information automatically)
- Continually update routing information

Automatic Route Propagation

- Two basic algorithms used by routing update protocols
  - Distance-vector
  - Link-state
- Many variations in implementation details

Distance-Vector Algorithm

- Initialize routing table with one entry for each directly-connected network
- Periodically run a distance-vector update to exchange information with routers that are reachable over directly-connected networks

Dynamic Update With Distance-Vector

- One router sends list of its routes to another
- List contains pairs of destination network and distance
- Receiver replaces entries in its table by routes to the sender if routing through the sender is less expensive than the current route
- Receiver propagates new routes next time it sends out an update
- Algorithm has well-known shortcomings (we will see an example later)

Example Of Distance-Vector Update

Destination	Distance	Route
Net 1	0	direct
Net 2	0	direct
Net 4	8	Router L
Net 17	5	Router M
Net 24	6	Router J
Net 30	2	Router Q
Net 42	2	Router J

(a)

Destination	Distance
Net 1	2
→ Net 4	3
Net 17	6
→ Net 21	4
Net 24	5
Net 30	10
→ Net 42	3

(b)

- (a) is existing routing table
- (b) incoming update (marked items cause change)

Link-State Algorithm

- Alternative to distance-vector
- Distributed computation
  - Broadcast information
  - Allow each router to compute shortest paths
- Avoids problem where one router can damage the entire internet by passing incorrect information
- Also called *Shortest Path First* (SPF)

Link-State Update

- Participating routers learn internet topology
- Think of routers as nodes in a graph, and networks connecting them as edges or links
- Pairs of directly-connected routers periodically
  - Test link between them
  - Propagate (broadcast) status of link
- All routers
  - Receive link status messages
  - Recompute routes from their local copy of information

Summary

- Routing tables can be
  - Initialized at startup (host or router)
  - Updated dynamically (router)
- Original Internet used core routing architecture
- Current Internet accommodates peer backbones
- Two important routing algorithms
  - Distance-vector
  - Link state


Questions?

PART XIV

ROUTING: EXTERIOR GATEWAY PROTOCOLS AND AUTONOMOUS SYSTEMS (BGP)

General Principle

Although it is desirable for routers to exchange routing information, it is impractical for all routers in an arbitrarily large internet to participate in a single routing update protocol.

- Consequence: routers must be divided into groups

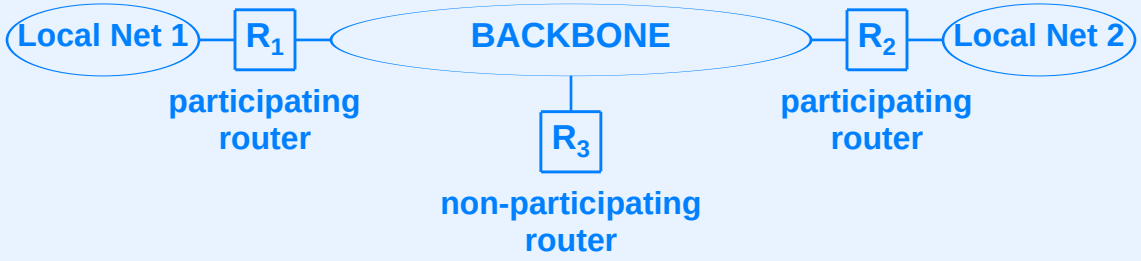
A Practical Limit On Group Size

It is safe to allow up to a dozen routers to participate in a single routing information protocol across a wide area network; approximately five times as many can safely participate across a set of local area networks.

Router Outside A Group

- Does not participate directly in group's routing information propagation algorithm
- Will not choose optimal routes if it uses a member of the group for general delivery

The Extra Hop Problem



- Non-participating router picks one participating router to use (e.g., R_2)
- Non-participating router routes all packets to R_2 across backbone
- Router R_2 routes some packets back across backbone to R_1

Statement Of The Problem

Treating a group of routers that participate in a routing update protocol as a default delivery system can introduce an extra hop for datagram traffic; a mechanism is needed that allows nonparticipating routers to learn routes from participating routers so they can choose optimal routes.

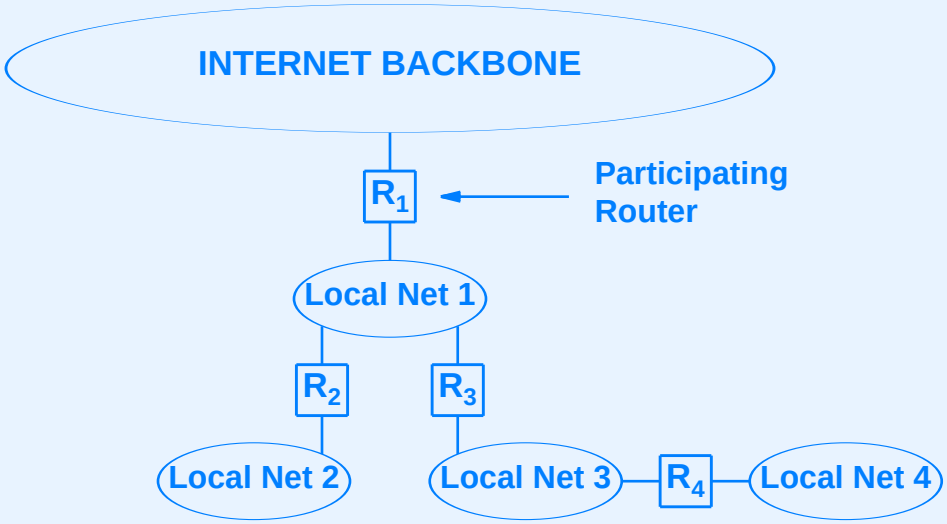
Solving The Extra Hop Problem

- Not all routers can participate in a single routing exchange protocol (does not scale)
- Even nonparticipating routers should make routing decisions
- Need mechanism that allows nonparticipating routers to obtain correct routing information automatically (without the overhead of participating fully in a routing exchange protocol)

Hidden Networks

- Each site has complex topology
- Nonparticipating router (from another site) cannot attach to all networks

Illustration Of Hidden Networks



- Propagation of route information is independent of datagram routing
- Group must learn routes from nonparticipating routers
- Example: owner of networks 1 and 3 must tell group that there is a route to network 4

A Requirement For Reverse Information Flow

Because an individual organization can have an arbitrarily complex set of networks interconnected by routers, no router from another organization can attach directly to all networks. A mechanism is needed that allows nonparticipating routers to inform the other group about hidden networks.

Autonomous System Concept (AS)

- Group of networks under one administrative authority
- Free to choose internal routing update mechanism
- Connects to one or more other autonomous systems

Modern Internet Architecture

A large TCP/IP internet has additional structure to accommodate administrative boundaries: each collection of networks and routers managed by one administrative authority is considered to be a single autonomous system that is free to choose an internal routing architecture and protocols.

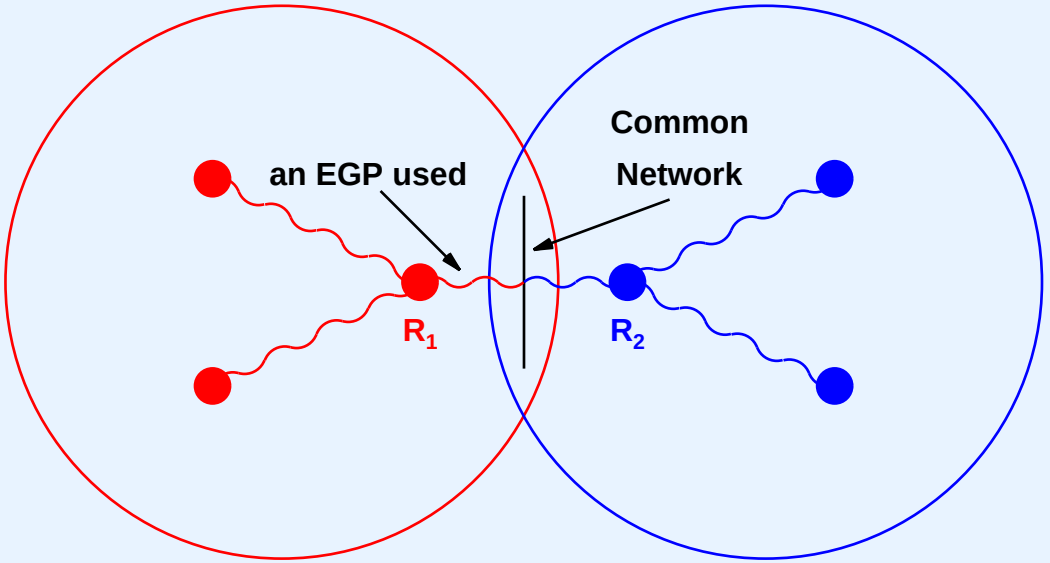
EGPs: Exterior Gateway Protocols

- Originally a single protocol for communicating routes between two autonomous systems
- Now refers to any exterior routing protocol
- Solves two problems
  - Allows router outside a group to advertise networks hidden in another autonomous system
  - Allows router outside a group to learn destinations in the group

Border Gateway Protocol

- The most popular (virtually the only) EGP in use in the Internet
- Current version is BGP-4
- Allows two autonomous systems to communicate routing information
- Supports CIDR (mask accompanies each route)
- Each AS designates a *border router* to speak on its behalf
- Two border routers become *BGP peers*

Illustration Of An EGP (Typically BGP)



Key Characteristics Of BGP

- Provides inter-autonomous system communication
- Propagates reachability information
- Follows next-hop paradigm
- Provides support for policies
- Sends path information
- Permits incremental updates
- Allows route aggregation
- Allows authentication

Additional BGP Facts

- Uses reliable transport (i.e., TCP)
  - Unusual: most routing update protocols use connectionless transport (e.g., UDP)
- Sends *keepalive* messages so other end knows connection is valid (even if no new routing information is needed)

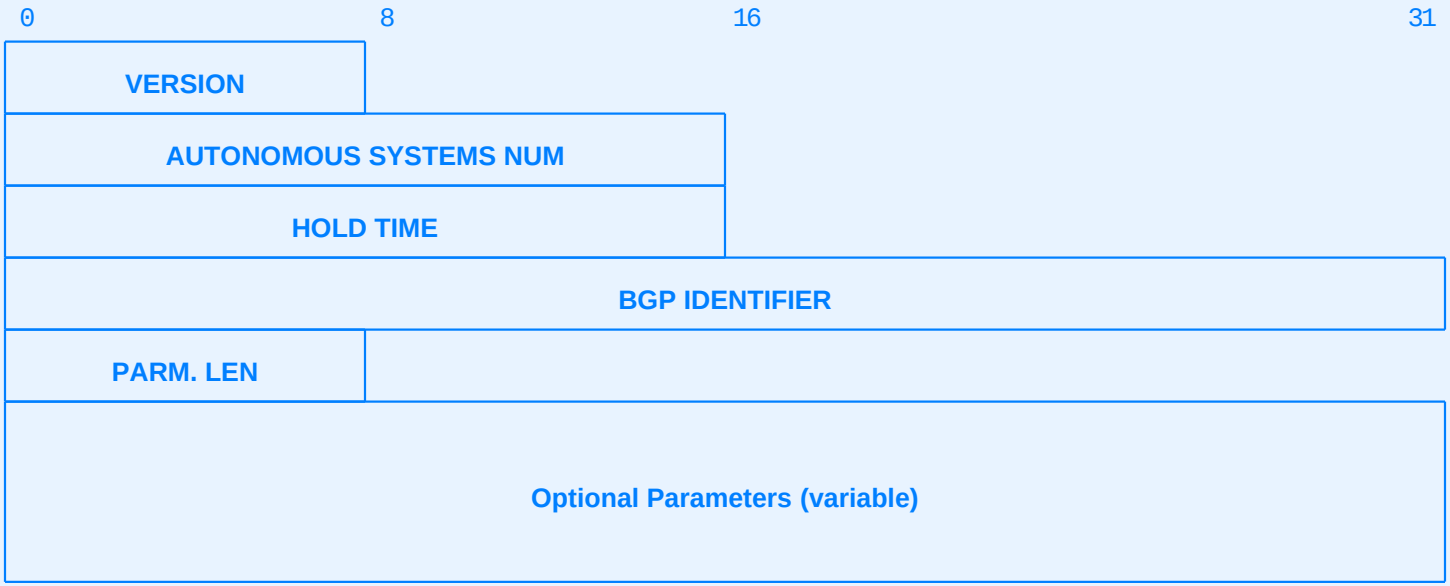
Four BGP Message Types

Type Code	Message Type	Description
1	OPEN	Initialize communication
2	UPDATE	Advertise or withdraw routes
3	NOTIFICATION	Response to an incorrect message
4	KEEPALIVE	Actively test peer connectivity

BGP Message Header

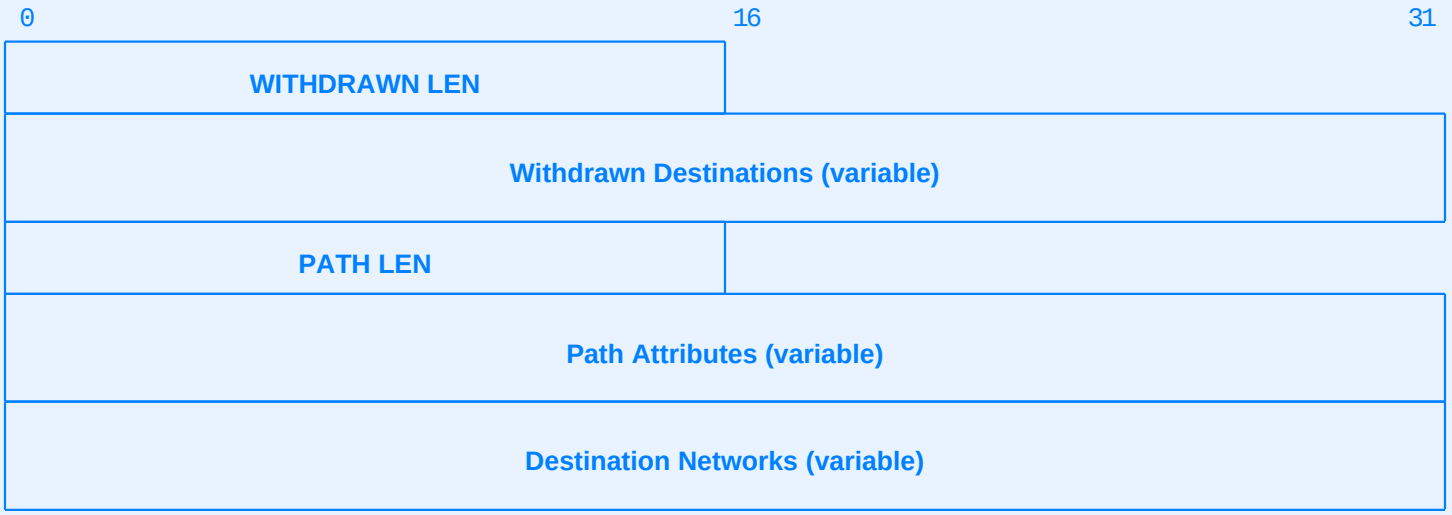
- Each BGP message starts with this header

BGP Open Message



- Used to start a connection
- HOLD TIME specifies max time that can elapse between BGP messages

BGP Update Message

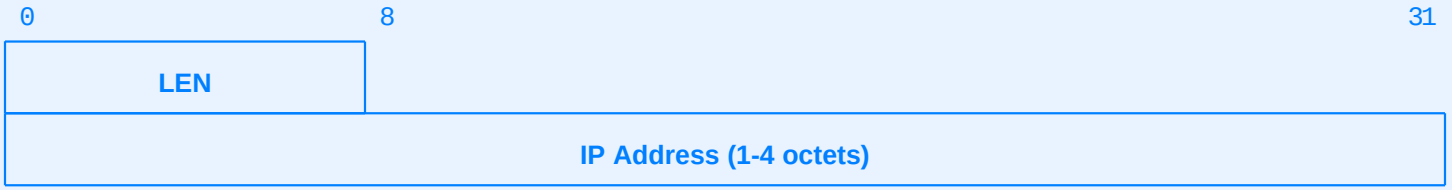


- Sender can advertise new routes or withdraw old routes

Compressed Address Entries

- Each route entry consists of address and mask
- Entry can be compressed to eliminate zero bytes

Format Of BGP Address Entry That Permits Compression

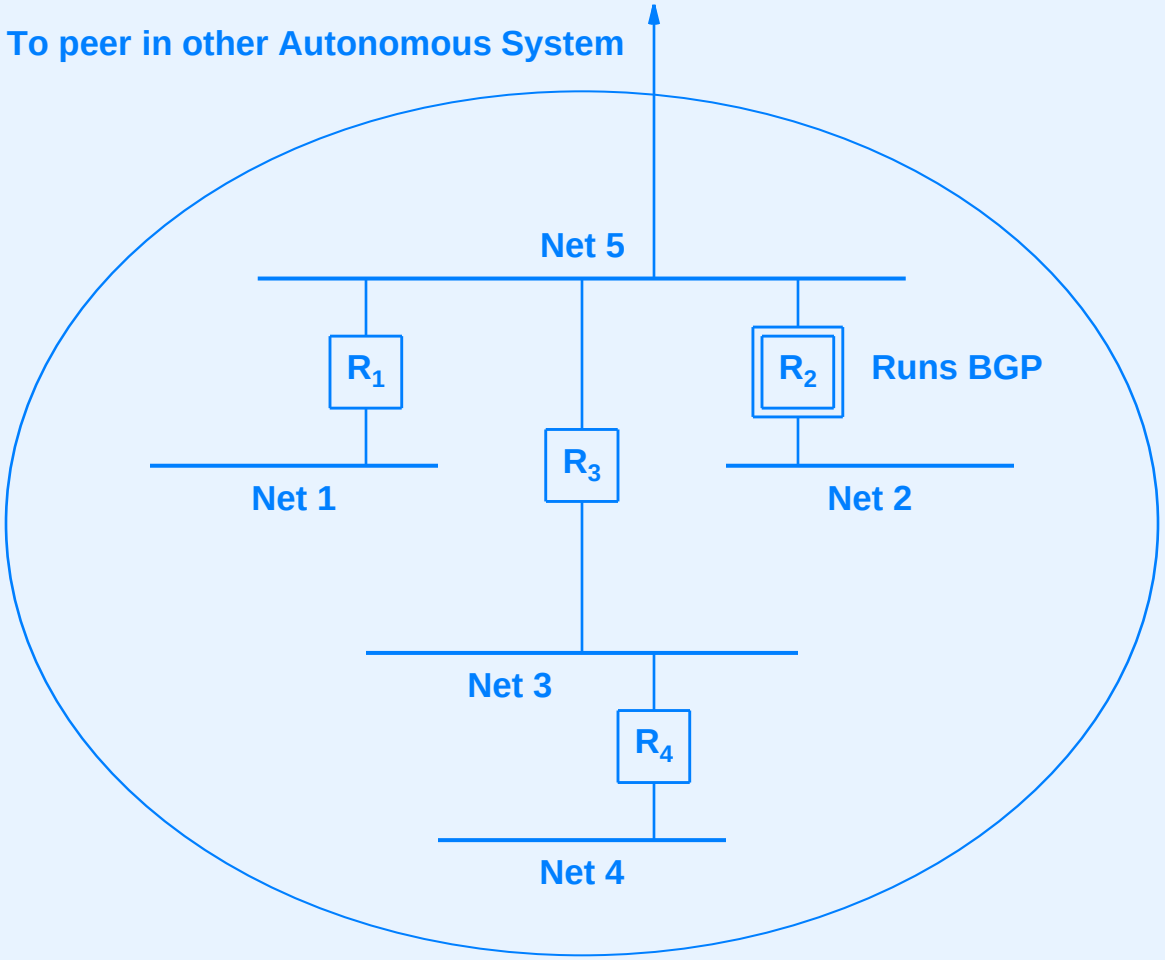


- LEN field specifies size of address that follows

Third-Party Routing Information

- Many routing protocols extract information from the local routing table
- BGP must send information ‘‘from the receiver’s perspective’’

Example Of Architecture In Which BGP Must Consider Receiver's Perspective



Metric Interpretation

- Each AS can use its own routing protocol
- Metrics differ
  - Hop count
  - Delay
  - Policy-based values
- EGP communicates between two separate autonomous systems

Key Restriction On An EGP

An exterior gateway protocol does not communicate or interpret distance metrics, even if metrics are available.

- Interpretation: ‘‘my autonomous system provides a path to this network’’

The Point About EGPs

Because an Exterior Gateway Protocol like BGP only propagates reachability information, a receiver can implement policy constraints, but cannot choose a least cost route. A sender must only advertise paths that traffic should follow.

Summary

- Internet is too large for all routers to participate in one routing update protocol
- Group of networks and routers under one administrative authority is called *Autonomous System* (*AS*)
- Each AS chooses its own interior routing update protocol
- Exterior Gateway Protocol (EGP) is used to communicate routing information between two autonomous systems
- Current exterior protocol is Border Gateway Protocol version 4, BGP-4
- An EGP provides reachability information, but does not associate metrics with each route


Questions?

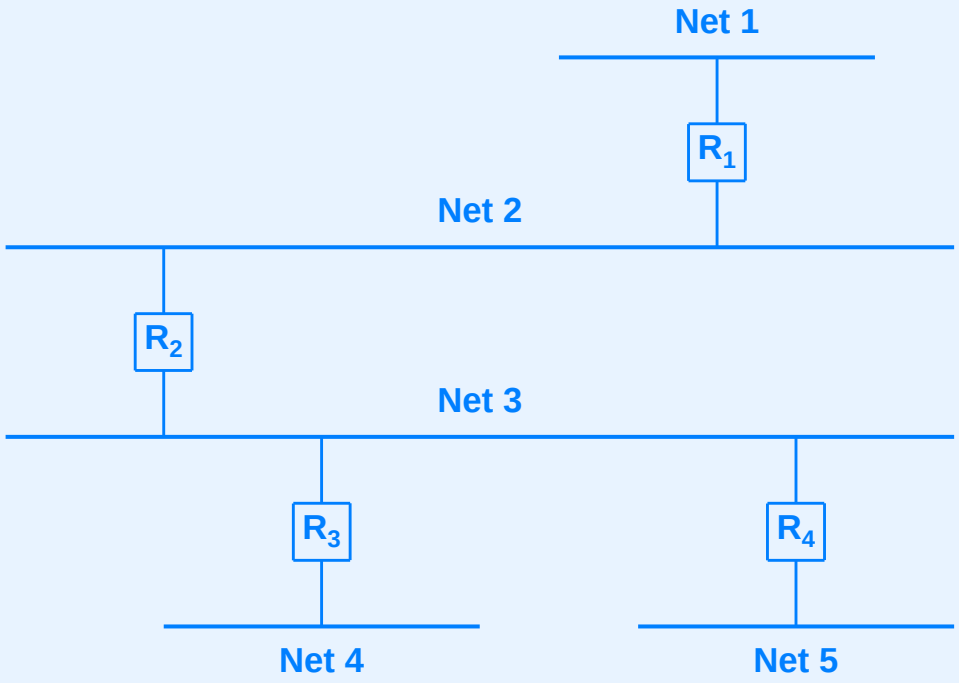
PART XV

ROUTING: INSIDE AN AUTONOMOUS SYSTEM (RIP, OSPF, HELLO)

Static Vs. Dynamic Interior Routes

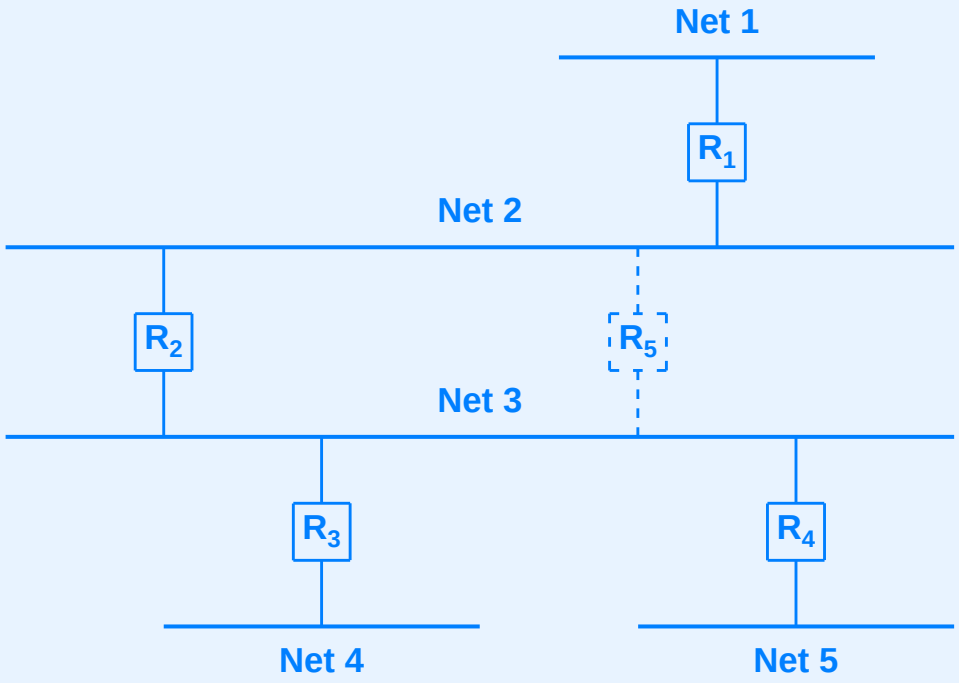
- Static routes
  - Initialized at startup
  - Never change
  - Typical for host
  - Sometimes used for router
- Dynamic router
  - Initialized at startup
  - Updated by route propagation protocols
  - Typical for router
  - Sometimes used in host

Illustration Of Topology In Which Static Routing Is Optimal



- Only one route exists for each destination

Illustration Of Topology In Which Dynamic Routing Is Needed

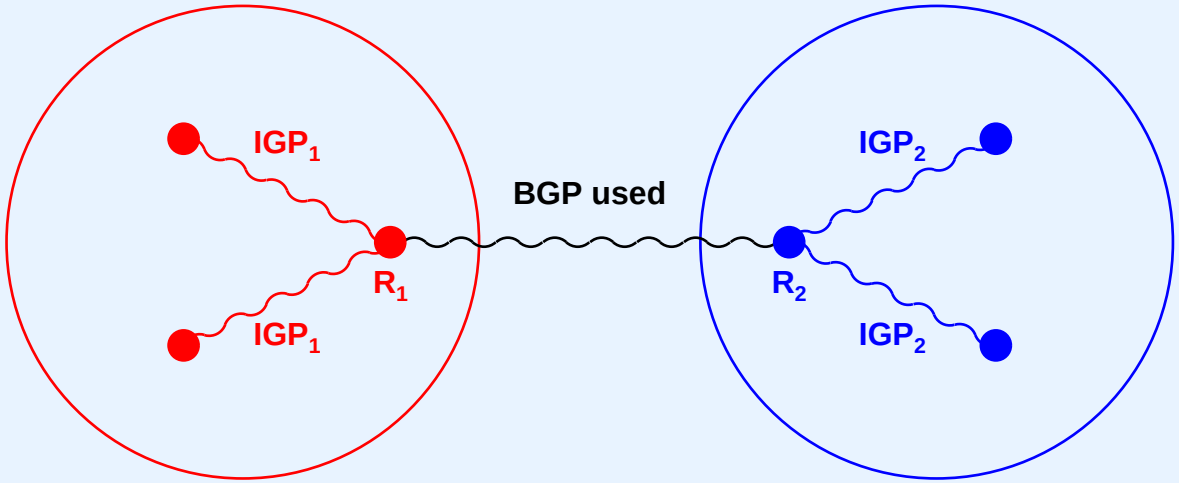


- Additional router introduces multiple paths

Exchanging Routing Information Within An Autonomous System

- Mechanisms called interior gateway protocols, IGPs
- Choice of IGP is made by autonomous system
- Note: if AS connects to rest of the world, a router in the AS must use an EGP to advertise network reachability to other autonomous systems.

Example Of Two Autonomous Systems And the Routing Protocols Used



Example IGPs

- RIP
- HELLO
- OSPF

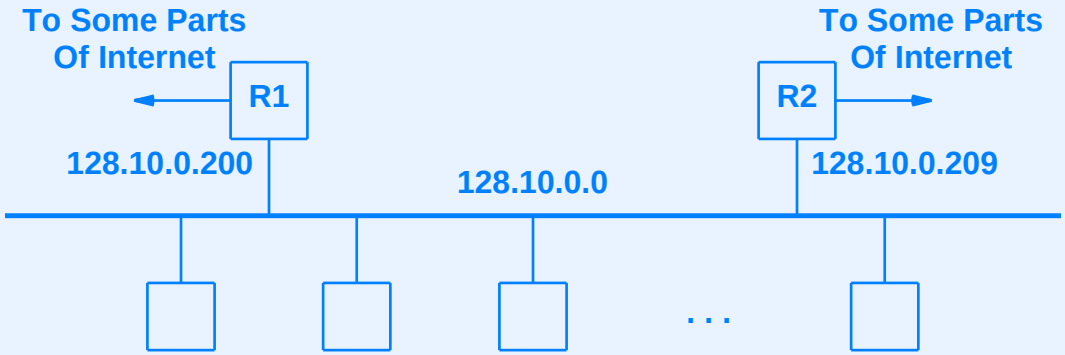
Routing Information Protocol (RIP)

- Implemented by UNIX program *routed*
- Uses hop count metric
- Distance-vector protocol
- Relies on broadcast
- Assumes low-delay local area network
- Uses split horizon and poison reverse techniques to solve inconsistencies
- Current standard is RIP2

Two Forms Of RIP

- Active
  - Form used by routers
  - Broadcasts routing updates periodically
  - Uses incoming messages to update routes
- Passive
  - Form used by hosts
  - Uses incoming messages to update routes
  - Does not send updates

Illustration Of Hosts Using Passive RIP



- Host routing table initialized to:

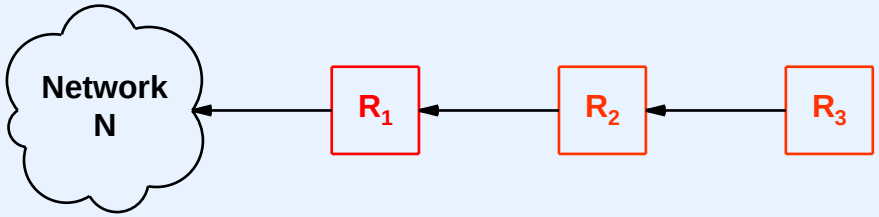
Destination	Route
128.10.0.0	direct
default	128.10.0.200

- Host listens for RIP broadcast and uses data to update table
- Eliminates ICMP redirects

RIP Operation

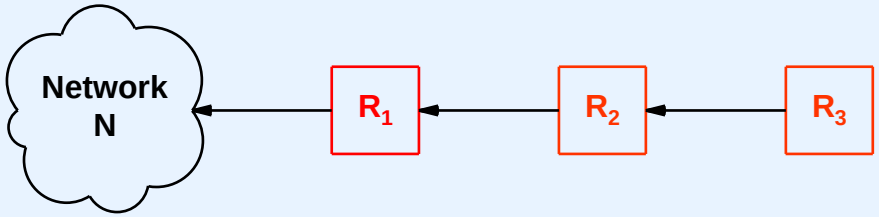
- Each router sends update every 30 seconds
- Update contains pairs of
  (destination address, distance)
- Distance of 16 is *infinity* (i.e., no route)

Slow Convergence Problem (Count To Infinity)

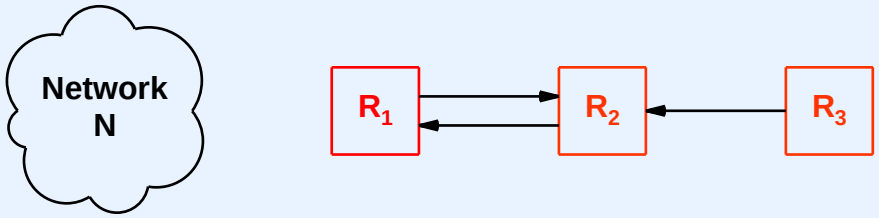


Routers with routes to network N

Slow Convergence Problem (Count To Infinity)

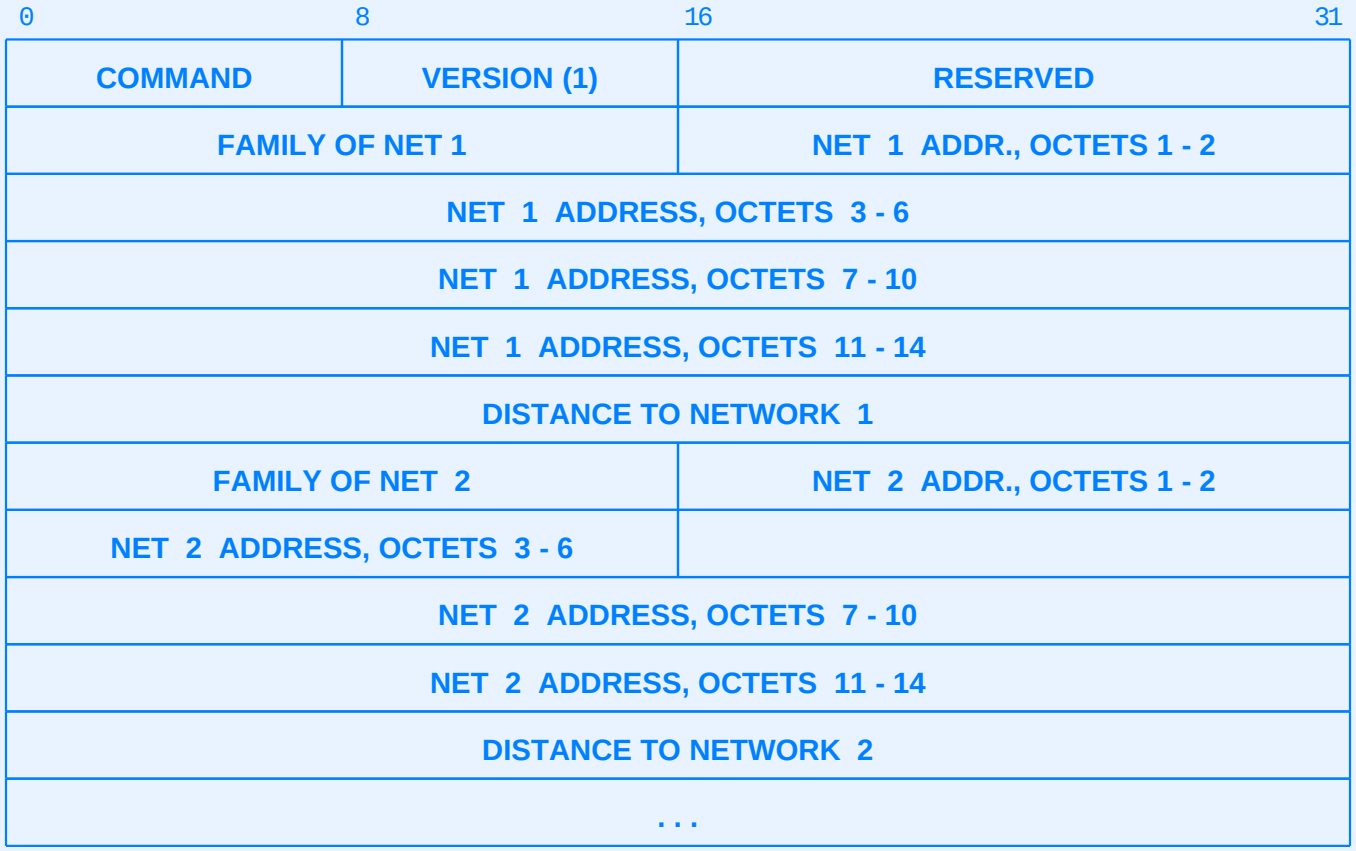


Routers with routes to network N



R_1 erroneously routes to R_2 after failure

RIP1 Update Format

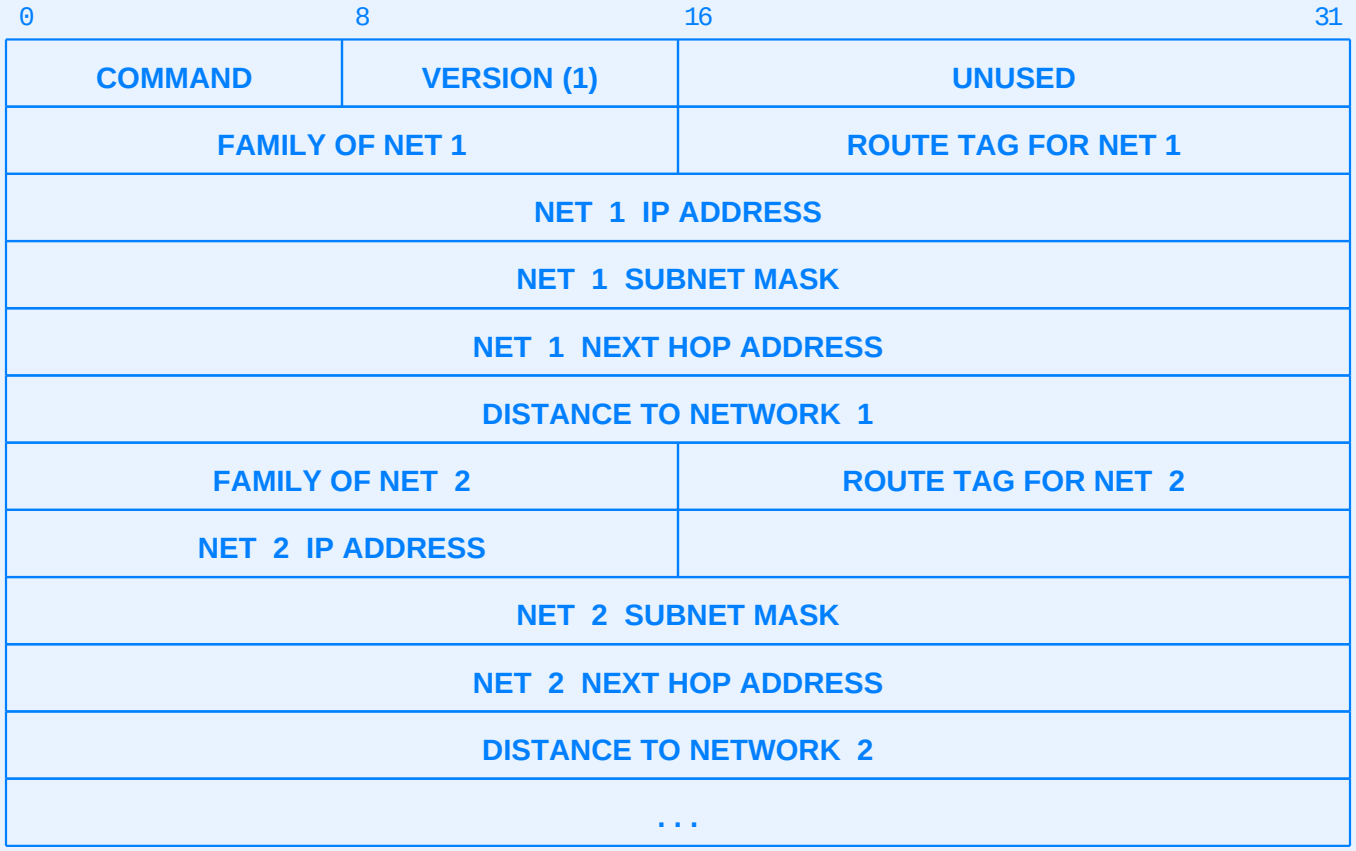


- Uses *FAMILY* field to support multiple protocols
- IP address sent in octets *3* - *6* of address field
- Message travels in UDP datagram

Changes To RIP
In Version 2

- Update includes subnet mask
- Authentication supported
- Explicit next-hop information
- Messages can be multicast (optional)
  - IP multicast address is 224.0.0.9

RIP2 Update Format



- Packet format is backward compatible
- Infinity still limited to *16*
- RIP2 *can* be broadcast

Measures Of Distance That Have Been Used

- Hops
  - Zero-origin
  - One-origin (e.g., RIP)
- Delay
- Throughput
- Jitter

HELLO: A Protocol That Used Delay

- Developed by Dave Mills
- Measured delay in milliseconds
- Used by NSFNET fuzzballs
- Now historic

How HELLO Worked

- Participants kept track of delay between pairs of routers
- HELLO propagated delay information across net
- Route chosen to minimize total delay

Route Oscillation

- Effective delay depends on traffic (delay increases as traffic increases)
- Using delay as metric means routing traffic where delay is low
- Increased traffic raises delay, which means route changes
- Routes tend to oscillate

Why HELLO Worked

- HELLO used only on NSFNET backbone
- All paths had equal throughput
- Route changes damped to avoid oscillation

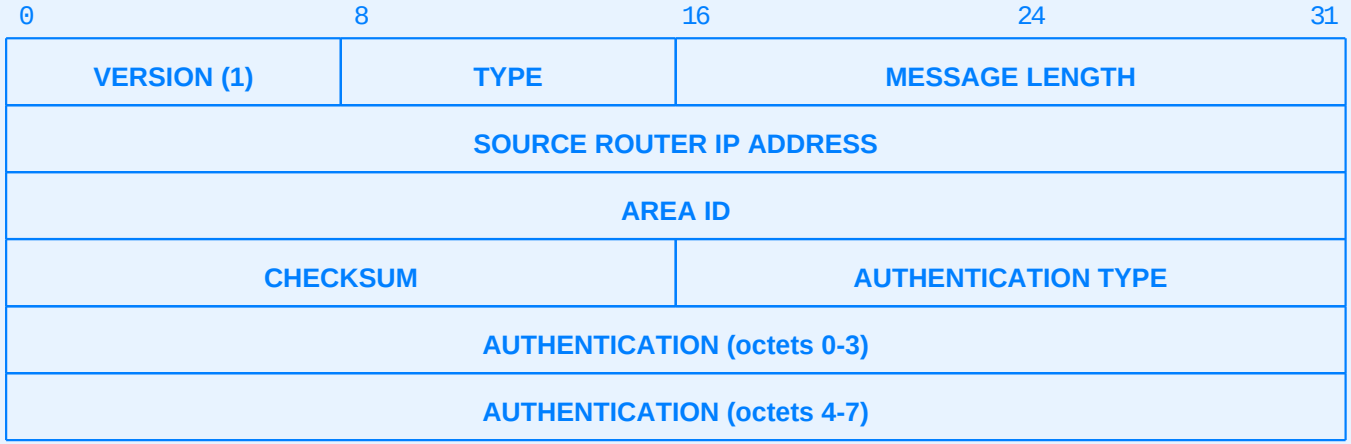
Open Shortest Path First (OSPF)

- Developed by IETF in response to vendors' proprietary protocols
- Uses SPF (link-state) algorithm
- More powerful than most predecessors
- Permits hierarchical topology
- More complex to install and manage

OSPF Features

- Type of service routing
- Load balancing across multiple paths
- Networks partitioned into subsets called *areas*
- Message authentication
- Network-specific, subnet-specific, host-specific, and CIDR routes
- Designated router optimization for shared networks
- Virtual network topology abstracts away details
- Can import external routing information

OSPF Message Header

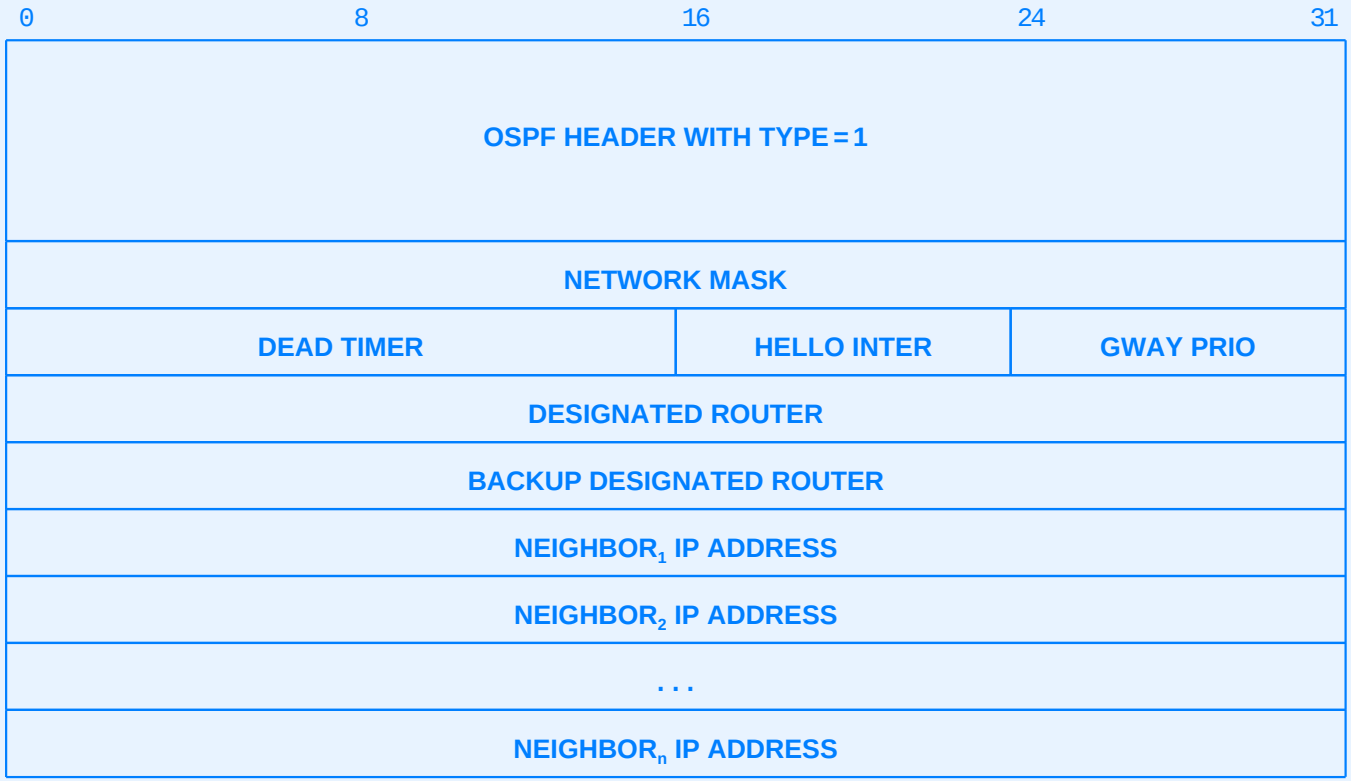


- Each message starts with same header

OSPF Message Types

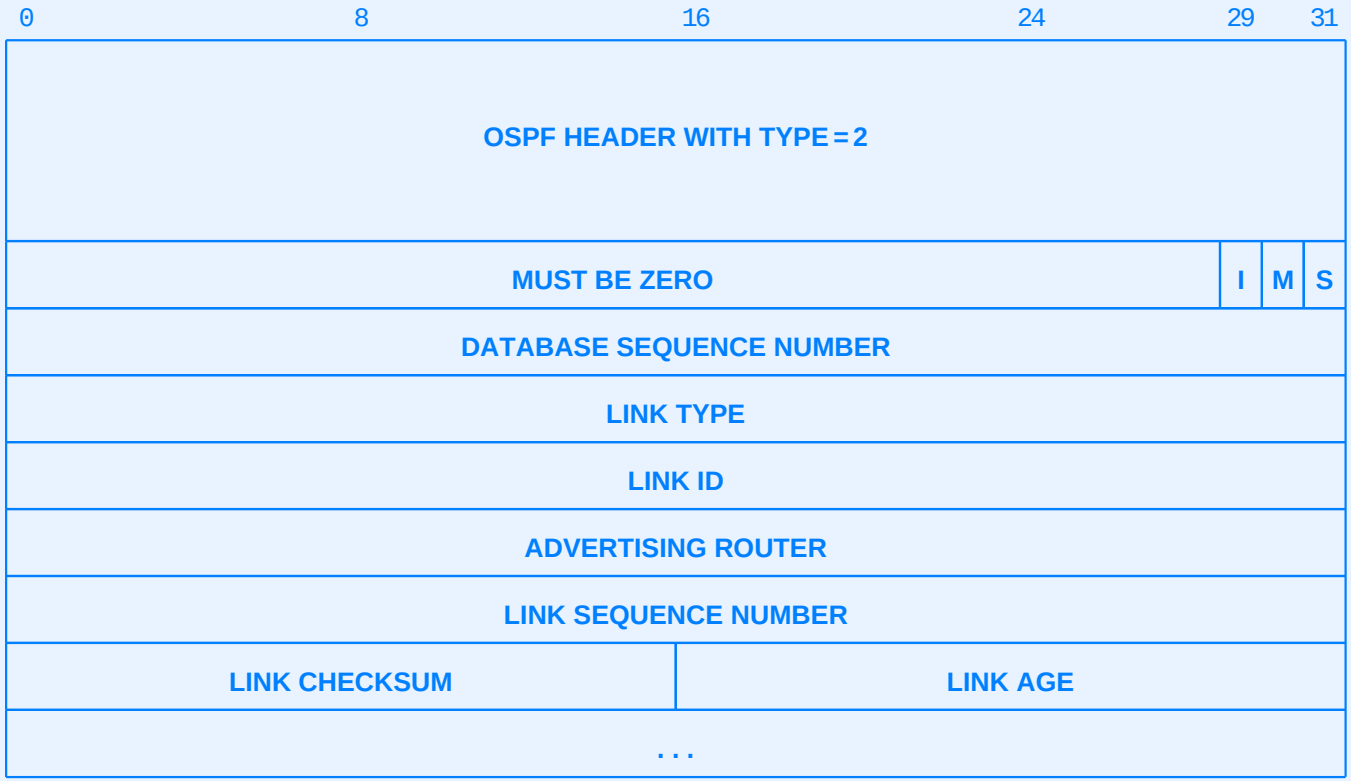
Type	Meaning
1	Hello (used to test reachability)
2	Database description (topology)
3	Link status request
4	Link status update
5	Link status acknowledgement

OSPF HELLO Message Format



- Used to test reachability

OSPF Database Description Message Format

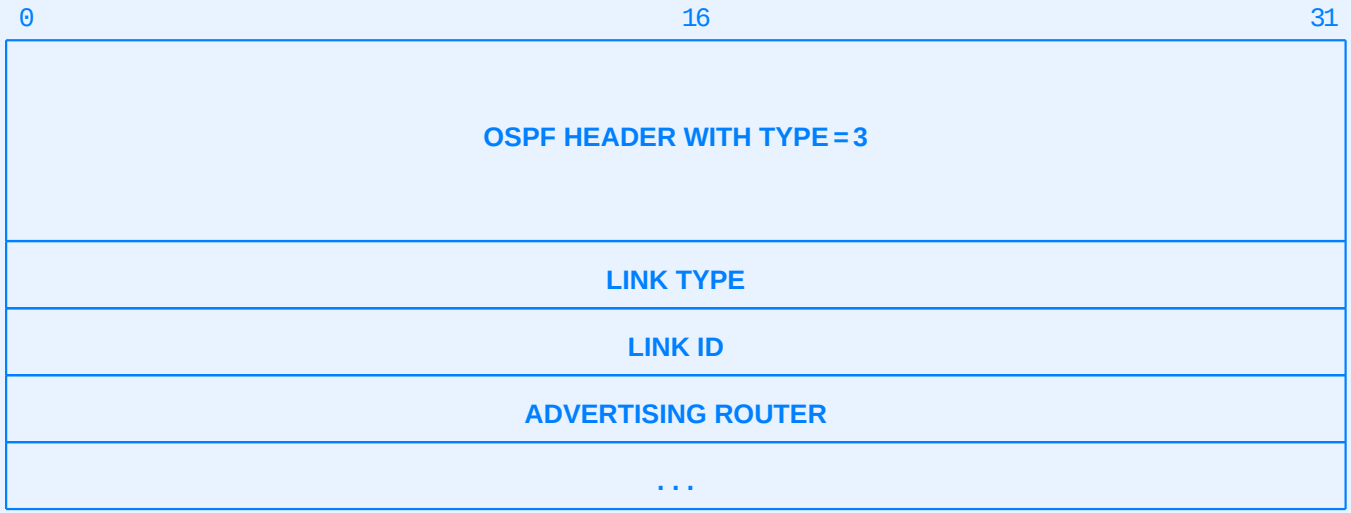


- Fields starting at *LINK TYPE* are repeated

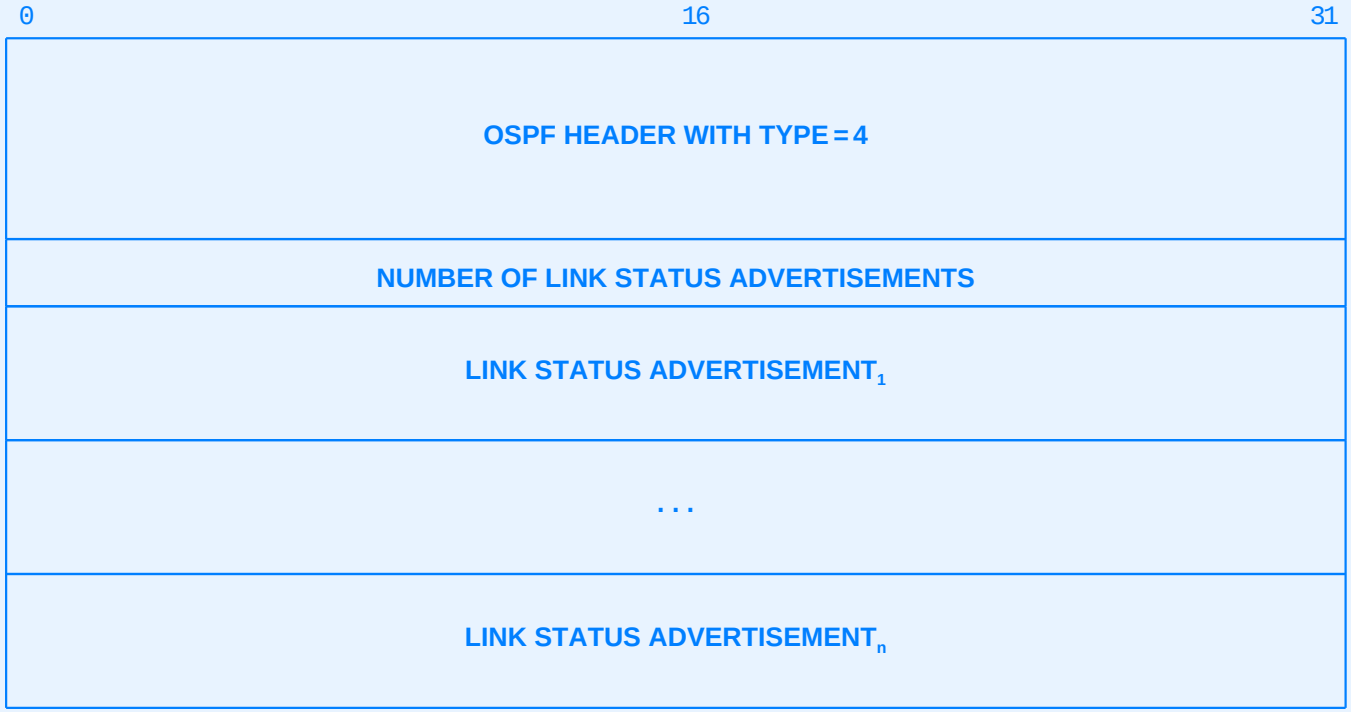
Values In The LINK Field

Link Type	Meaning
1	Router link
2	Network link
3	Summary link (IP network)
4	Summary link (link to border router)
5	External link (link to another site)

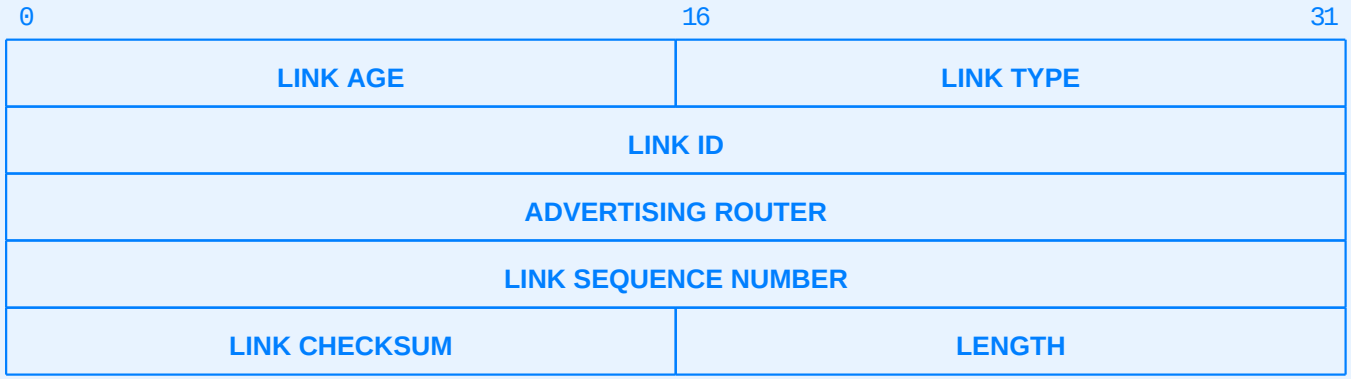
OSPF Link Status Request Message Format



OSPF Link Status Update Message Format



Header Used In OSPF Link Status Advertisements



- Four possible formats follow
  - Links from a router to given area
  - Links from a router to physical net
  - Links from a router to physical nets of a subnetted IP network
  - Links from a router to nets at other sites

Discussion Question

- What are the tradeoffs connected with the issue of routing in the presence of partial information?

Summary

- Interior Gateway Protocols (IGPs) used within an AS
- Popular IGPs include
  - RIP (distance vector algorithm)
  - OSPF (link-state algorithm)


Questions?

PART XVI

INTERNET MULTICASTING

Hardware Multicast

- Form of broadcast
- Only one copy of a packet traverses the net
- NIC initially configured to accept packets destined to
  - Computer's unicast address
  - Hardware broadcast address
- User can dynamically add (and later remove)
  - One or more multicast addresses

A Note About Hardware Multicast

Although it may help to think of multicast addressing as a generalization that subsumes unicast and broadcast addresses, the underlying forwarding and delivery mechanisms can make multicast less efficient.

Ethernet Multicast

- Determined by low-order bit of high-order byte
- Example in dotted decimal:

$$01.00.00.00.00.00_{16}$$

- Remaining bits specify a *multicast group*

IP Multicast

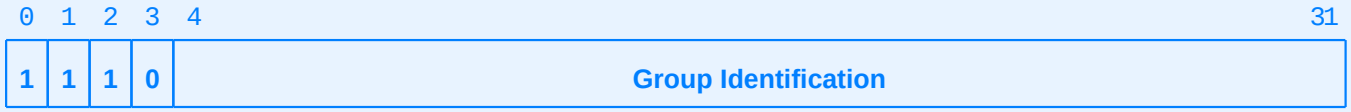
- Group address: each multicast group assigned a unique class D address
- Up to 2^{28} simultaneous multicast groups
- Dynamic group membership: host can join or leave at any time
- Uses hardware multicast where available
- Best-effort delivery semantics (same as IP)
- Arbitrary sender (does not need to be a group member)

Facilities Needed For Internet Multicast

- Multicast addressing scheme
- Effective notification and delivery mechanism
- Efficient Internet forwarding facility

IP Multicast Addressing

- Class D addresses reserved for multicast
- General form:



- Two types
  - Well-known (address reserved for specific protocol)
  - Transient (allocated as needed)

Multicast Addresses

- Address range

  224.0.0.0 through 239.255.255.255

- Notes
  - 224.0.0.0 is reserved (never used)
  - 224.0.0.1 is ‘‘all systems’’
  - 224.0.0.3 is ‘‘all routers’’
  - Address up through 224.0.0.255 used for multicast routing protocols

Example Multicast Address Assignments

Address	Meaning
224.0.0.0	Base Address (Reserved)
224.0.0.1	All Systems on this Subnet
224.0.0.2	All Routers on this Subnet
224.0.0.3	Unassigned
224.0.0.4	DVMRP Routers
224.0.0.5	OSPFIGP All Routers
224.0.0.6	OSPFIGP Designated Routers
224.0.0.7	ST Routers
224.0.0.8	ST Hosts
224.0.0.9	RIP2 Routers
224.0.0.10	IGRP Routers
224.0.0.11	Mobile-Agents
224.0.0.12	DHCP Server / Relay Agent
224.0.0.13	All PIM Routers
224.0.0.14	RSVP-Encapsulation
224.0.0.15	All-CBT-Routers
224.0.0.16	Designated-Sbm
224.0.0.17	All-Sbms
224.0.0.18	VRRP

Example Multicast Address Assignments (continued)

Address	Meaning
224.0.0.19 through 224.0.0.255	Other Link Local Addresses
224.0.1.0 through 238.255.255.255	Globally Scoped Addresses
239.0.0.0 through 239.255.255.255	Scope restricted to one organization

Mapping An IP Multicast Address To An Ethernet Multicast Address

- Place low-order 23 bits of IP multicast address in low-order 23 bits of the special Ethernet address:

$$01.00.5E.00.00.00_{16}$$

- Example IP multicast address 224.0.0.2 becomes Ethernet multicast address

$$01.00.5E.00.00.02_{16}$$

Transmission Of Multicast Datagrams

- Host does *not* install route to multicast router
- Host uses hardware multicast to transmit multicast datagrams
- If multicast router is present on net
  - Multicast router receives datagram
  - Multicast router uses destination address to determine routing

Multicast Scope

- Refers to range of members in a group
- Defined by set of networks over which multicast datagrams travel to reach group
- Two techniques control scope
  - IP's TTL field (TTL of 1 means local net only)
  - Administrative scoping

Host Participation In IP Multicast

- Host can participate in one of three ways:

Level	Meaning
0	Host can neither send nor receive IP multicast
1	Host can send but not receive IP multicast
2	Host can both send and receive IP multicast

- Note: even level 2 requires additions to host software

Host Details For Level 2 Participation

- Host uses *Internet Group Management Protocol* (*IGMP*) to announce participation in multicast
- If multiple applications on a host join the same multicast group, each receives a copy of messages sent to the group
- Group membership is associated with a specific network:

A host joins a specific IP multicast group on a specific network.

IGMP

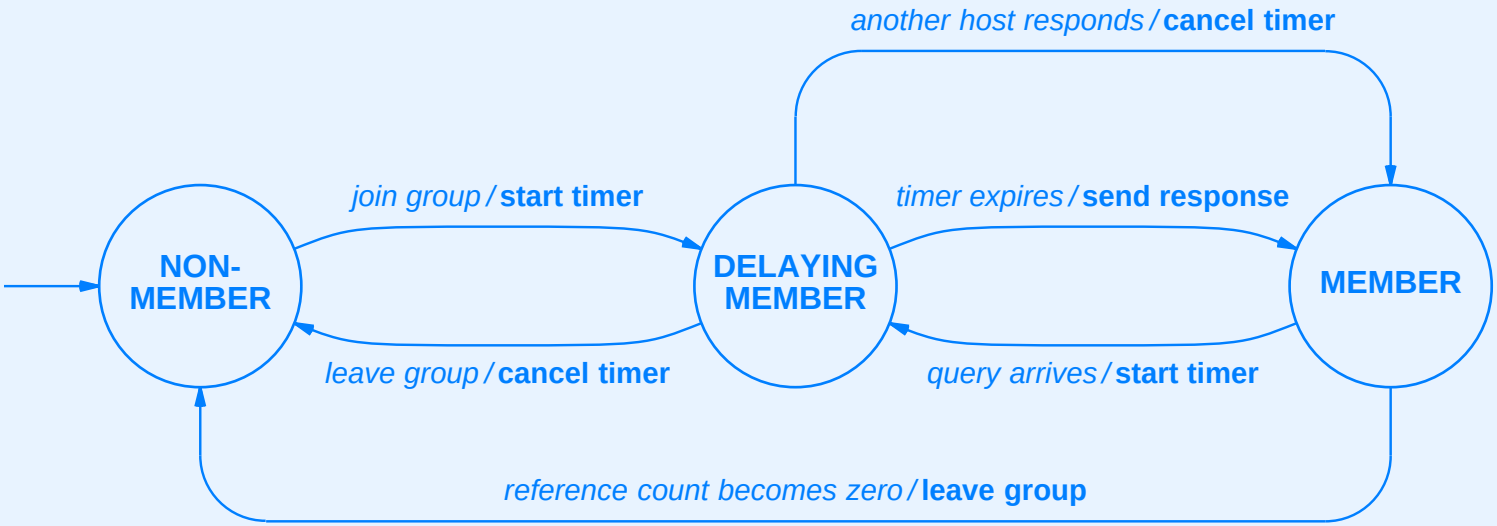
- Allows host to register participation in a group
- Two conceptual phases
  - When it joins a group, host sends message declaring membership
  - Multicast router periodically polls a host to determine if any host on the network is still a member of a group

IGMP Implementation

- All communication between host and multicast router uses hardware multicast
- Single query message probes for membership in all active groups
- Default polling rate is every 125 seconds
- If multiple multicast routers attach to a shared network, one is elected to poll
- Host waits random time before responding to poll (to avoid simultaneous responses)
- Host listens to other responses, and suppresses unnecessary duplicate responses

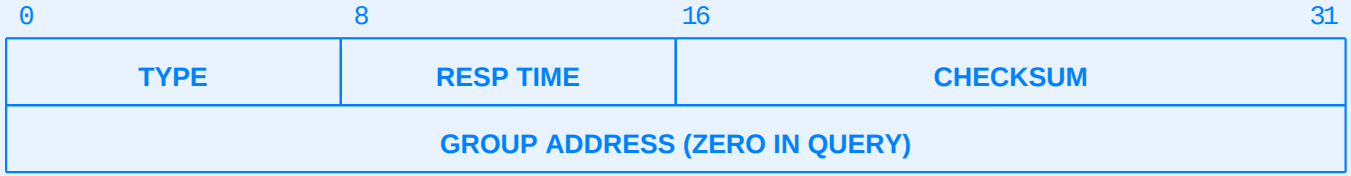
IGMP State Transitions

- Host uses FSM to determine actions:



- Separate state kept for each multicast group

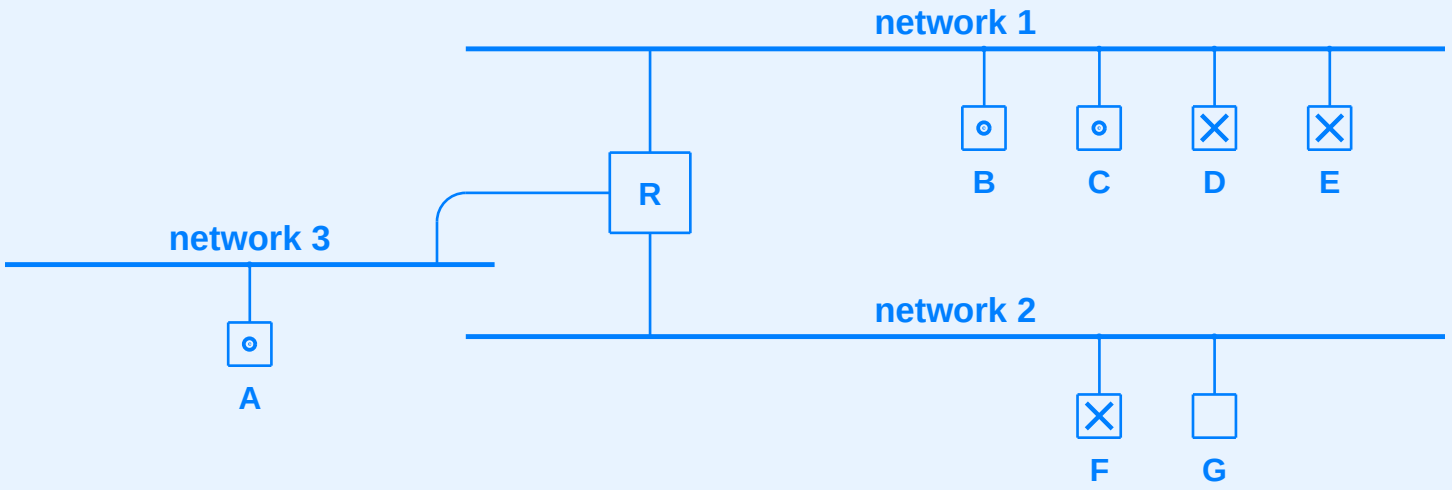
IGMP Message Format



- Message TYPE field is one of:

Type	Group Address	Meaning
0x11	unused (zero)	General membership query
0x11	used	Specific group membership query
0x16	used	Membership report
0x17	used	Leave group
0x12	used	Membership report (version 1)

Multicast Forwarding Example



- Hosts marked with dot participate in one group
- Hosts marked with X participate in another group
- Forwarding depends on group membership

The Complexity Of Multicast Routing

Unlike unicast routing in which routes change only when the topology changes or equipment fails, multicast routes can change simply because an application program joins or leaves a multicast group.

Multicast Forwarding Complication

Multicast forwarding requires a router to examine more than the destination address.

- In most cases, forwarding depends on the source address as well as the destination address

Final Item That Complicates IP Multicast

A multicast datagram may originate on a computer that is not part of the multicast group, and may be forwarded across networks that do not have any group members attached.

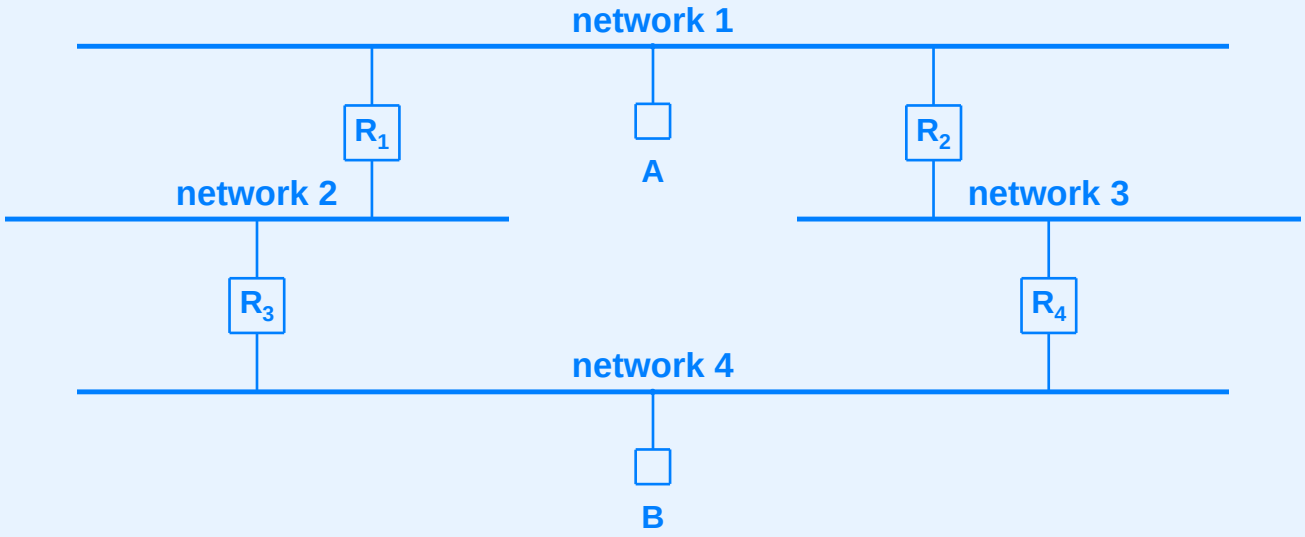
Multicast Routing Paradigms

- Two basic approaches
- Flood-and-prune
  - Send a copy to all networks
  - Only stop forwarding when it is known that no participant lies beyond a given point
- Multicast trees
  - Routers interact to form a ‘‘tree’’ that reaches all networks of a given group
  - Copy traverses branches of the tree

Reverse Path Forwarding

- Early flood-and-prune approach
- Actual algorithm is *Truncated Reverse Path Forwarding* (*TRPF*)

Example Topology In Which TRPF Delivers Multiple Copies



Multicast Trees

A multicast forwarding tree is defined as a set of paths through multicast routers from a source to all members of a multicast group. For a given multicast group, each possible source of datagrams can determine a different forwarding tree.

Examples Of Multicast Routing Protocols

- Reverse Path Multicasting (RPM)
- Distance-Vector Multicast Routing Protocol (DVMRP)
- Core-Based Trees (CBT)
- Protocol Independent Multicast - Dense Mode (PIM-DM)
- Protocol Independent Multicast - Sparse Mode (PIM-SM)

Reverse Path Multicasting (RPM)

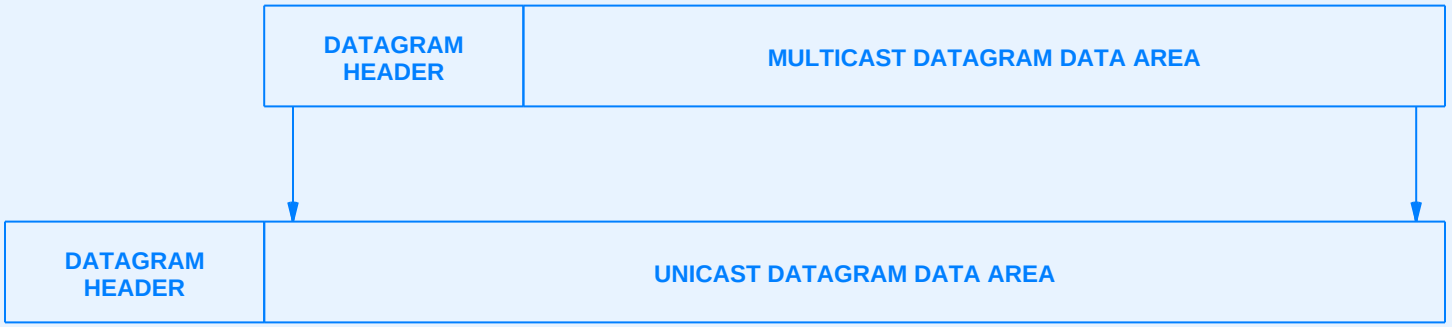
- Early form
- Routers flood datagrams initially
- Flooding pruned as group membership information learned

Distance-Vector Multicast Routing Protocol (DVMRP)

- Early protocol
- Defines extension of IGMP that routers use to exchange multicast routing information
- Implemented by Unix *mrouted* program
  - Configures tables in kernel
  - Supports tunneling
  - Used in Internet's *Multicast backBONE* (*MBONE*)

Topology In Which Tunneling Needed

Encapsulation Used With Tunneling



- IP travels in IP

Core-Based Trees (CBT)

- Proposed protocol
- Better for sparse network
- Does not forward to a net until host on the net joins a group
- Request to join a group sent to ‘‘core’’ of network
- Multiple cores used for large Internet

Division Of Internet

Because CBT uses a demand-driven paradigm, it divides the internet into regions and designates a core router *for each region; other routers in the region dynamically build a forwarding tree by sending join requests to the core.*

Protocol Independent Multicast - Dense Mode (PIM-DM)

- Allows router to build multicast forwarding table from information in conventional routing table
- Term ‘‘dense’’ refers to density of group members
- Best for high density areas
- Uses flood-and-prune approach

Protocol Independent Multicast - Sparse Mode (PIM-SM)

- Allows router to build multicast forwarding table from information in conventional routing table
- Term ‘‘sparse’’ refers to relative density of group members
- Best for situations with ‘‘islands’’ of participating hosts separated by networks with no participants
- Uses tree-based approach

Question For Discussion

- How can we provide reliable multicast?

Summary

- IP multicasting uses hardware multicast for delivery
- Host uses Internet Group Management Protocol (IGMP) to communicate group membership to local multicast router
- Two forms of multicast routing used
  - Flood-and-prune
  - Tree-based

Summary
(continued)

- Many multicast routing protocols have been proposed
  - TRPF
  - DVMRP
  - CBT
  - PIM-DM
  - PIM-SM


Questions?

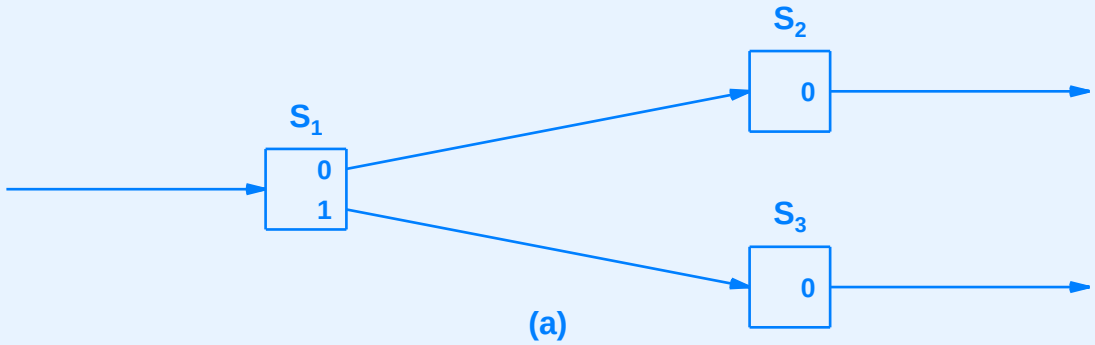
PART XVII

IP Switching And MPLS

Switching Technology

- Designed as a higher-speed alternative to packet forwarding
- Uses array lookup instead of destination address lookup
- Often associated with *Asynchronous Transfer Mode (ATM)*

Switching Concept

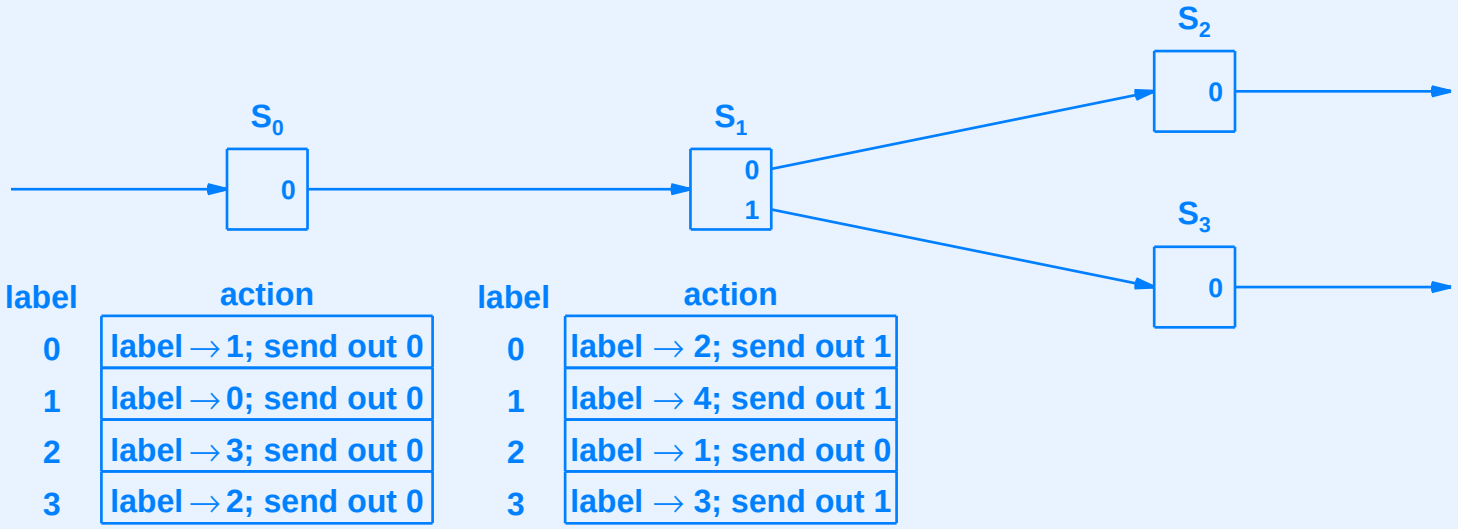


label	action
0	send out interface 1
1	send out interface 1
2	send out interface 0
3	send out interface 1
	⋮

(b)

- Part (b) shows table for switch S_1
- Identifier in packet known as *label*
- All labels except 2 go out interface 1

Extending Switching To A Large Network



- Label replacement known as *label swapping*
- A path through the network corresponds to a sequence of labels

An Important Note

Switching uses a connection-oriented approach. To avoid the need for global agreement on the use of labels, the technology allows a manager to define a path of switches without requiring that the same label be used across the entire path.

Potential Advantages Of Switching For IP Forwarding

- Faster forwarding
- Aggregated route information
- Ability to manage aggregate flows

IP Switching

- Pioneered by Ipsilon Corporation
- Originally used ATM hardware
- Variants by others known as
  - Layer 3 switching
  - Tag switching
  - Label switching
- Ideas eventually consolidated into *Multi-Protocol Label Switching (MPLS)*

MPLS Operation

- Internet divided into
  - Standard routers
  - MPLS core
- Datagram encapsulated when entering the MPLS core and de-encapsulated when leaving
- Within the core, MPLS labels are used to forward packets

Processing An Incoming Datagram

- Datagram *classified*
  - Multiple headers examined
  - Example: classification can depend on TCP port numbers as well as IP addresses
- Classification used to assign a label
- Note: each label corresponds to ‘‘flow’’ that may include may TCP sessions

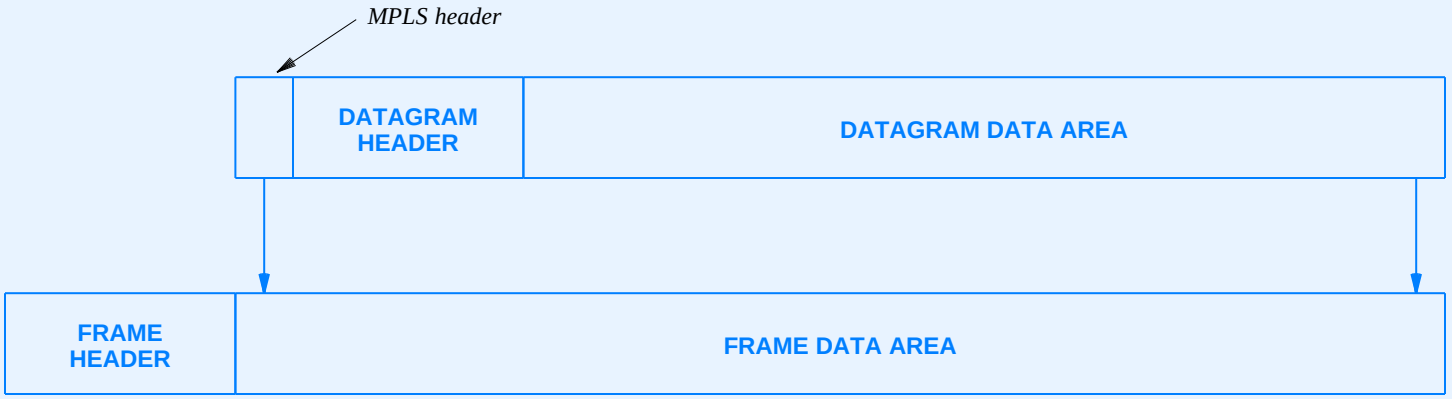
Hierarchical MPLS

- Multi-level hierarchy is possible
- Example: corporation with three campuses and multiple buildings on each campus
  - Conventional forwarding within a building
  - One level of MPLS for buildings within a campus
  - Additional level of MPLS between campuses
- To accommodate hierarchy, MPLS uses *stack* of labels

MPLS Label Processing

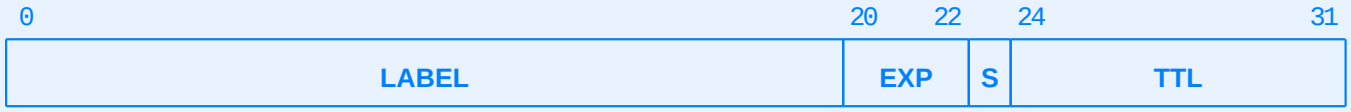
- Only top label is used to forward
- When entering new level of hierarchy, push addtional label on stack
- When leaving a level of the hierarchy, pop the top label from the stack

MPLS Encapsulation



- MPLS can run over conventional networks
- Shim header contains labels

Fields In An MPLS Shim Header



- Shim header
  - Prepended to IP datagram
  - Only used while datagram in MPLS core
- MPLS switches use LABEL in shim when forwarding packet

Label Switching Router (LSR)

- Device that connects between conventional Internet and MPLS core
- Handles classification
- Uses data structure known as *Next Hop Label Forwarding Table (NHLFT)* to choose an action

Next Hop Label Forwarding Entry

- Found in NHLFT
- Specifies
  - Next hop information (e.g., the outgoing interface)
  - Operation to be performed
  - Encapsulation to use (optional)
  - How to encode the label (optional)
  - Other information needed to handle the packet (optional)

Possible Operations

- Replace label at top of stack
- Pop label at top of stack
- Replace label at top of stack, and then push one or more new labels onto stack

Control Processing And Label Distribution

- Needed to establish Label Switched Path (LSP)
  - Coordinate labels along the path
  - Configure next-hop forwarding in switches
- Performed by *Label Distribution mechanism*
- Series of labels selected automatically

Protocols For MPLS Control

- Two primary protocols proposed
  - Label Distribution Protocol (MPLS-LDP)
  - Constraint-Based Routing LDP (CR-LDP)
- Other proposals to extend routing protocols
  - OSPF
  - BGP

Notes About Fragmentation

- Outgoing
  - MPLS prepends shim header to each datagram
  - If datagram fills network MTU, fragmentation will be required
- Incoming
  - Classification requires knowledge of headers (e.g., TCP port numbers)
  - Only first fragment contains needed information
  - LSR must collect fragments and reassemble before classification

Mesh Topology

- Used in many MPLS cores
- LSP established between each pair of LSRs
- Parallel LSPs can be used for levels of service
- Example
  - One LSP reserved for VOIP traffic
  - Another LSP used for all other traffic

Service Differentiation

Because MPLS classification can use arbitrary fields in a datagram, including the IP source address, the service a datagram receives can depend on the customer sending the datagram as well as the type of data being carried.


Questions?

PART XVIII

MOBILE IP

Mobility And IP Addressing

- Recall: prefix of IP address identifies network to which host is attached
- Consequence: when moving to a new network either
  - Host must change its IP address
  - All routers install host-specific routes

Mobile IP

- Technology to support mobility
  - Allows host to retain original IP address
  - Does not require routers to install host-specific routes

Characteristics Of Mobile IP

- Transparent to applications and transport protocols
- Interoperates with standard IPv4
- Scales to large Internet
- Secure
- Macro mobility (intended for working away from home rather than moving at high speed)

General Approach

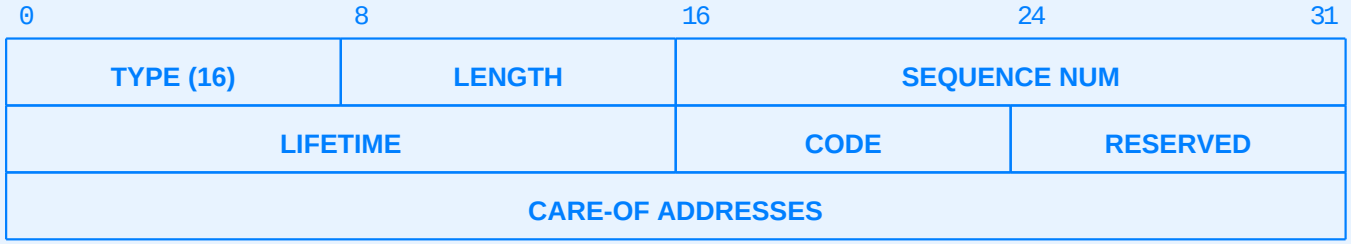
- Host visiting a *foreign* network obtains second IP address that is local to the site
- Host informs router on *home* network
- Router at home uses second address to forward datagrams for the host to the foreign network
  - Datagrams sent in a tunnel
  - Uses IP-in-IP encapsulation

Two Broad Approaches

- Foreign network runs system known as *foreign agent*
  - Visiting host registers with foreign agent
  - Foreign agent assigns host a temporary address
  - Foreign agent registers host with *home agent*
- Foreign network does not run a *foreign agent*
  - Host uses DHCP to obtain temporary address
  - Host registers directly with home agent

Foreign Agent Advertisement Extension

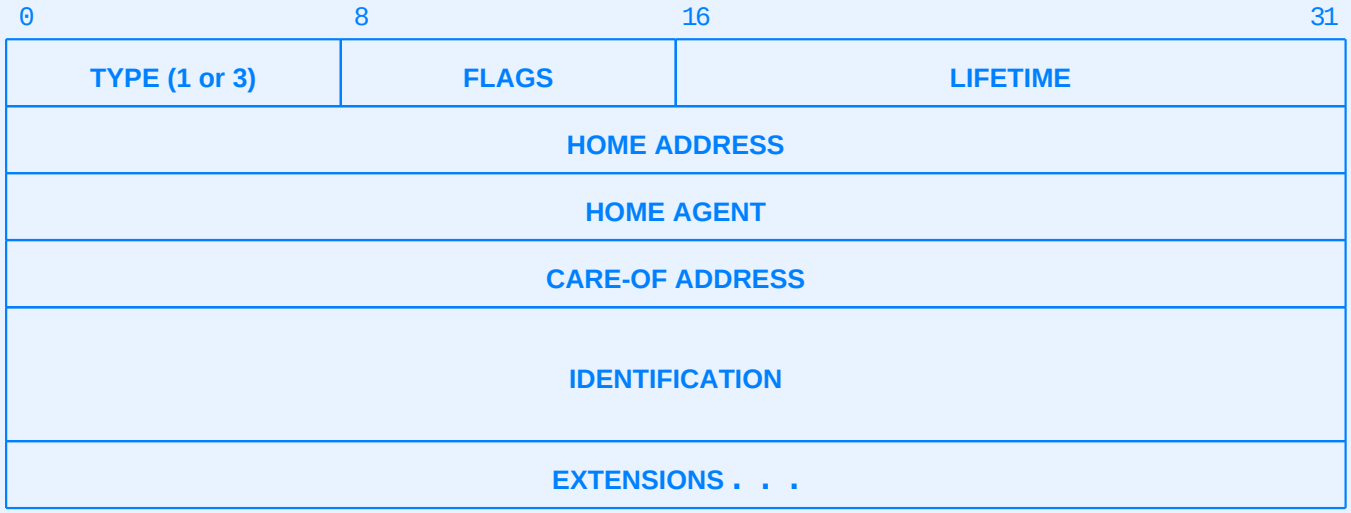
- Sent by router that runs foreign agent
- Added to ICMP router advertisement
- Format:



CODE Field In Advertisement Message

Bit	Meaning
0	Registration with an agent is required; co-located care-of addressing is not permitted
1	The agent is busy and is not accepting registrations
2	Agent functions as a home agent
3	Agent functions as a foreign agent
4	Agent uses minimal encapsulation
5	Agent uses GRE-style encapsulation
6	Agent supports header compression when communicating with mobile
7	Unused (must be zero)

Host Registration Request



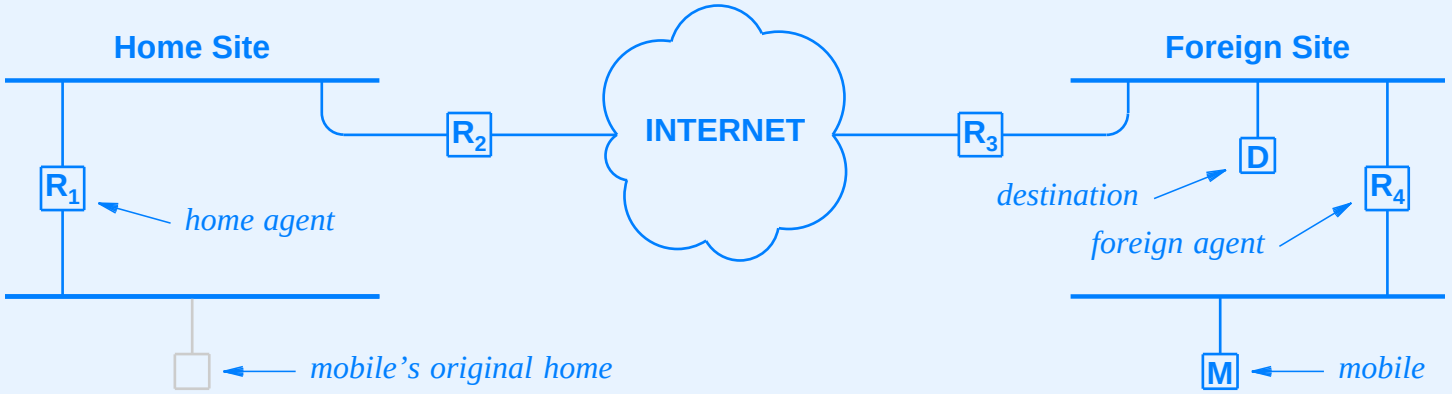
FLAGS Field In Host Registration Request

Bit	Meaning
0	This is a simultaneous (additional) address rather than a replacement.
1	Mobile requests home agent to tunnel a copy of each broadcast datagram
2	Mobile is using a co-located care-of address and will decapsulate datagrams itself
3	Mobile requests agent to use minimal encapsulation
4	Mobile requests agent to use GRE encapsulation
5	Mobile requests header compression
6-7	Reserved (must be zero)

Consequence Of Mobile IP

Because a mobile uses its home address as a source address when communicating with an arbitrary destination, each reply is forwarded to the mobile's home network, where an agent intercepts the datagram, encapsulates it in another datagram, and forwards it either directly to the mobile or to the foreign agent the mobile is using.

Illustration Of The Two-Crossing Problem



A Severe Problem

Mobile IP introduces a routing inefficiency known as the two-crossing problem that occurs when a mobile visits a foreign network far from its home and then communicates with a computer near the foreign site. Each datagram sent to the mobile travels across the Internet to the mobile's home agent which then forwards the datagram back to the foreign site. Eliminating the problem requires propagating host-specific routes; the problem remains for any destination that does not receive the host-specific route.

Summary

- Mobile IP allows a host to visit a foreign site without changing its IP address
- A visiting host obtains a second, temporary address which is used for communication while at the site
- The chief advantage of mobile IP arises from transparency to applications
- The chief disadvantage of mobile IP arises from inefficient routing known as a two-crossing problem


Questions?

PART XIX

PRIVATE NETWORK INTERCONNECTION (NAT AND VPN)

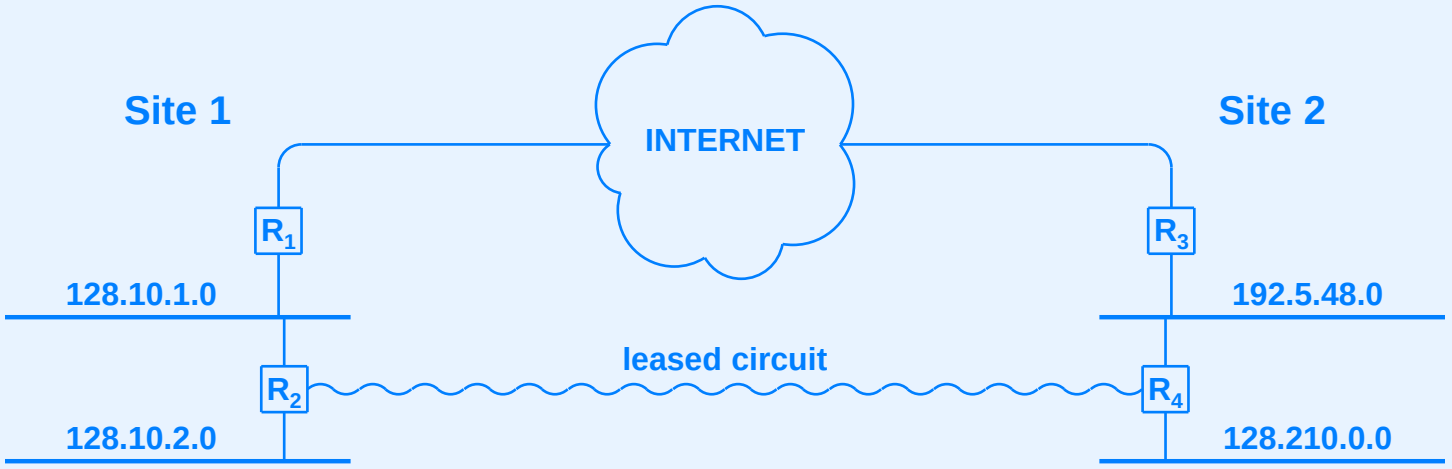
Definitions

- An internet is *private* to one group (sometimes called *isolated*) if none of the facilities or traffic is accessible to other groups
  - Typical implementation involves using leased lines to interconnect routers at various sites of the group
- The global Internet is *public* because facilities are shared among all subscribers

Hybrid Architecture

- Permits some traffic to go over private connections
- Allows contact with global Internet

Example Of Hybrid Architecture



The Cost Of Private And Public Networks

- Private network extremely expensive
- Public Internet access inexpensive
- Goal: combine safety of private network with low cost of global Internet

Question

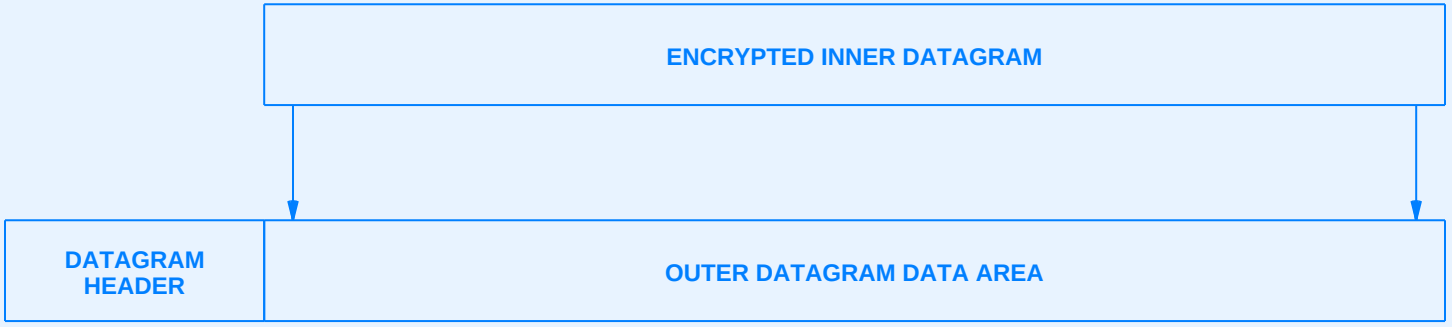
How can an organization that uses the global Internet to connect its sites keep its data private?

- Answer: *Virtual Private Network* (*VPN*)

Virtual Private Network

- Connect all sites to global Internet
- Protect data as it passes from one site to another
  - Encryption
  - IP-in-IP tunneling

Illustration Of Encapsulation Used With VPN



The Point

A Virtual Private Network sends data across the Internet, but encrypts intersite transmissions to guarantee privacy.

Example Of VPN Addressing And Routing

destination	next hop
128.10.1.0	direct
128.10.2.0	R_2
192.5.48.0	tunnel to R_3
128.210.0.0	tunnel to R_3
default	ISP's router

Routing table in R_1

Example VPN With Private Addresses

- Advantage: only one globally valid IP address needed per site

General Access With Private Addresses

- Question: how can a site provide multiple computers at the site access to Internet services without assigning each computer a globally-valid IP address?
- Two answers
  - Application gateway (one needed for each service)
  - *Network Address Translation* (*NAT*)

Network Address Translation (NAT)

- Extension to IP addressing
- IP-level access to the Internet through a single IP address
- Transparent to both ends
- Implementation
  - Typically software
  - Usually installed in IP router
  - Special-purpose hardware for highest speed

Network Address Translation (NAT) (continued)

- Pioneered in Unix program *slirp*
- Also known as
  - *Masquerade* (Linux)
  - *Internet Connection Sharing* (Microsoft)
- Inexpensive implementations available for home use

NAT Details

- Organization
  - Obtains one globally valid address per Internet connection
  - Assigns nonroutable addresses internally (net 10)
  - Runs NAT software in router connecting to Internet
- NAT
  - Replaces source address in outgoing datagram
  - Replaces destination address in incoming datagram
  - Also handles higher layer protocols (e.g., pseudo header for TCP or UDP)

NAT Translation Table

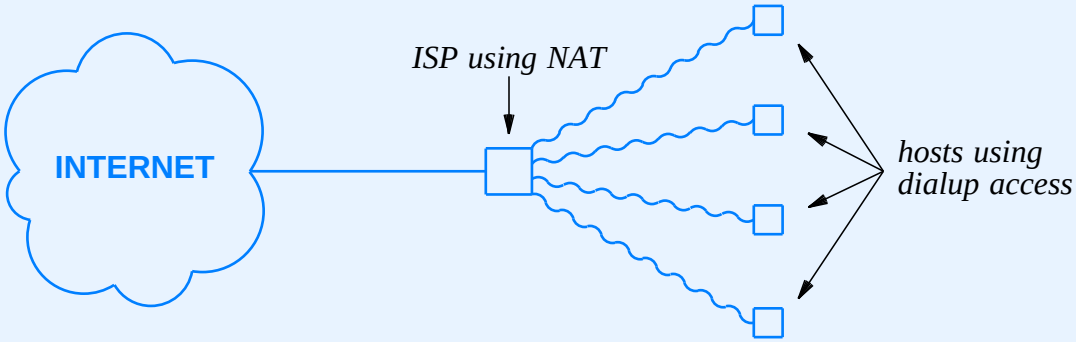
- NAT uses translation table
- Entry in table specifies local (private) endpoint and global destination.
- Typical paradigm
  - Entry in table created as side-effect of datagram leaving site
  - Entry in table used to reverse address mapping for incoming datagram

Example NAT Translation Table

Private Address	Private Port	External Address	External Port	NAT Port	Protocol Used
10.0.0.5	21023	128.10.19.20	80	14003	tcp
10.0.0.1	386	128.10.19.20	80	14010	tcp
10.0.2.6	26600	207.200.75.200	21	14012	tcp
10.0.0.3	1274	128.210.1.5	80	14007	tcp

- Variant of NAT that uses protocol port numbers is known as *Network Address and Port Translation* (*NAPT*)

Use Of NAT By An ISP



Higher Layer Protocols And NAT

- NAT must
  - Change IP headers
  - Possibly change TCP or UDP source ports
  - Recompute TCP or UDP checksums
  - Translate ICMP messages
  - Translate port numbers in an FTP session

Applications And NAT

NAT affects ICMP, TCP, UDP, and other higher-layer protocols; except for a few standard applications like FTP, an application protocol that passes IP addresses or protocol port numbers as data will not operate correctly across NAT.

Summary

- Virtual Private Networks (VPNs) combine the advantages of low cost Internet connections with the safety of private networks
- VPNs use encryption and tunneling
- Network Address Translation allows a site to multiplex communication with multiple computers through a single, globally valid IP address.
- NAT uses a table to translate addresses in outgoing and incoming datagrams


Questions?

PART XX

CLIENT-SERVER MODEL OF INTERACTION

Client-Server Paradigm

- Conceptual basis for virtually all distributed applications
- One program initiates interaction to which another program responds
- Note: ‘‘peer-to-peer’’ applications use client-server paradigm internally

Definitions

- Client
  - Any application program
  - Contacts a server
  - Forms and sends a request
  - Awaits a response
- Server
  - Usually a specialized program that offers a service
  - Awaits a request
  - Computes an answer
  - Issues a response

Server Persistence

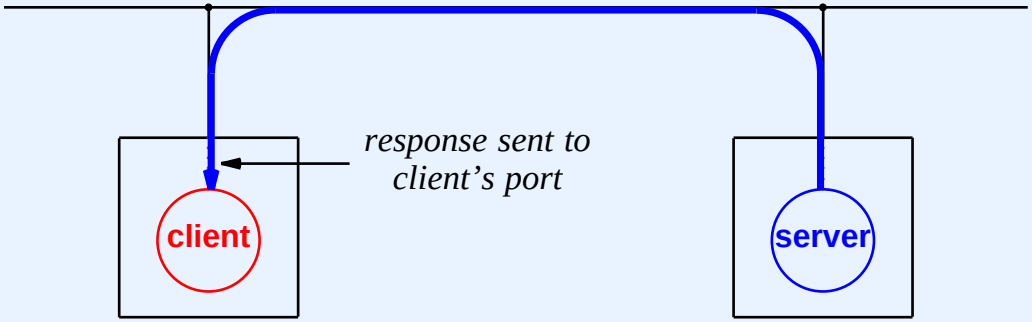
A server starts execution before interaction begins and (usually) continues to accept requests and send responses without ever terminating. A client is any program that makes a request and awaits a response; it (usually) terminates after using a server a finite number of times.

Illustration Of The Client-Server Paradigm

Client sends request

Illustration Of The Client-Server Paradigm

Client sends request



Server sends response

Use Of Protocol Ports

A server waits for requests at a well-known port that has been reserved for the service it offers. A client allocates an arbitrary, unused, nonreserved port for its communication.

Client Side

- Any application program can become a client
- Must know how to reach the server
  - Server's Internet address
  - Server's protocol port number
- Usually easy to build

Server Side

- Finds client's location from incoming request
- Can be implemented with application program or in operating system
- Starts execution before requests arrive
- Must ensure client is authorized
- Must uphold protection rules
- Must handle multiple, concurrent requests
- Usually complex to design and build

Concurrent Server Algorithm

- Open well-known port
- Wait for next client request
- Create a new socket for the client
- Create thread / process to handle request
- Continue with *wait* step

Complexity Of Servers

Servers are usually more difficult to build than clients because, although they can be implemented with application programs, servers must enforce all the access and protection policies of the computer system on which they run and must protect themselves against all possible errors.

Summary

- Client-server model is basis for distributed applications
- Server is specialized, complex program (process) that offers a service
- Arbitrary application can become a client by contacting a server and sending a request
- Most servers are concurrent


Questions?

PART XXI

THE SOCKET INTERFACE

Using Protocols

- Protocol software usually embedded in OS
- Applications run outside OS
- Need an *Application Program Interface* (*API*) to allow application to access protocols

API

- TCP/IP standards
  - Describe general functionality needed
  - Do not give details such as function names and arguments
- Each OS free to define its own API
- In practice: *socket interface* has become de facto standard API

Socket API

- Defined by U.C. Berkeley as part of BSD Unix
- Adopted (with minor changes) by Microsoft as *Windows Sockets*

Characteristics Of Socket API

- Follows Unix's open-read-write-close paradigm
- Uses Unix's *descriptor* abstraction
  - First, create a socket and receive an integer descriptor
  - Second, call a set of functions that specify all the details for the socket (descriptor is argument to each function)
- Once socket has been established, use *read* and *write* or equivalent functions to transfer data
- When finished, close the socket

Creating A Socket

result = socket(pf, type, protocol)

- Argument specifies protocol family as TCP/IP

Terminating A Socket

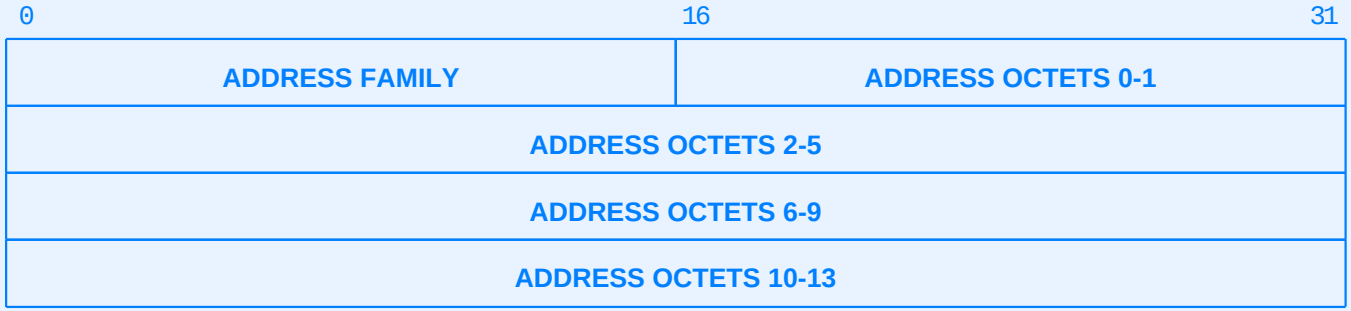
close(socket)

- Closing a socket permanently terminates the interaction

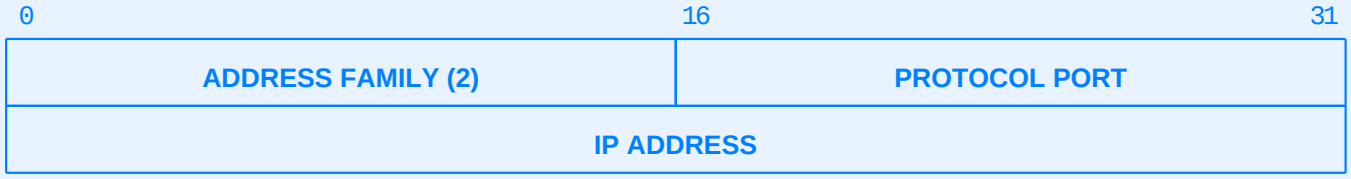
Specifying A Local Address For The Socket

bind(socket, localaddr, addrlen)

Format Of A Sockaddr Structure (Generic)



Format Of A Sockaddr Structure When Used With TCP/IP



Connecting A Socket To A Destination Address

connect(socket, destaddr, addrlen)

- Can be used with UDP socket to specify remote endpoint address

Sending Data Through A Socket

send(socket, message, length, flags)

- Note
  - Function *write* can also be used
  - Alternatives exist for connectionless transport (UDP)

Receiving Data Through A Socket

recv(socket, buffer, length, flags)

- Note
  - Function *read* can also be used
  - Alternatives exist for connectionless transport (UDP)

Obtaining Remote And Local Socket Addresses

getpeername(socket, destaddr, addrlen)

and

getsockname(socket, localaddr, addrlen)

Set Maximum Queue Length (Server)

listen(socket, qlength)

- Maximum queue length can be quite small

Accepting New Connections (Server)

newsock = accept(socket, addr, addrlen)

- Note:
  - Original socket remains available for accepting connections
  - New socket corresponds to one connection
  - Permits server to handle requests concurrently

Handling Multiple Services With One Server

- Server
  - Creates socket for each service
  - Calls *select* function to wait for any request
  - Select specifies which service was contacted
- Form of select

  nready = select(ndesc, indesc, outdesc, excdesc, timeout)

Socket Functions Used For DNS

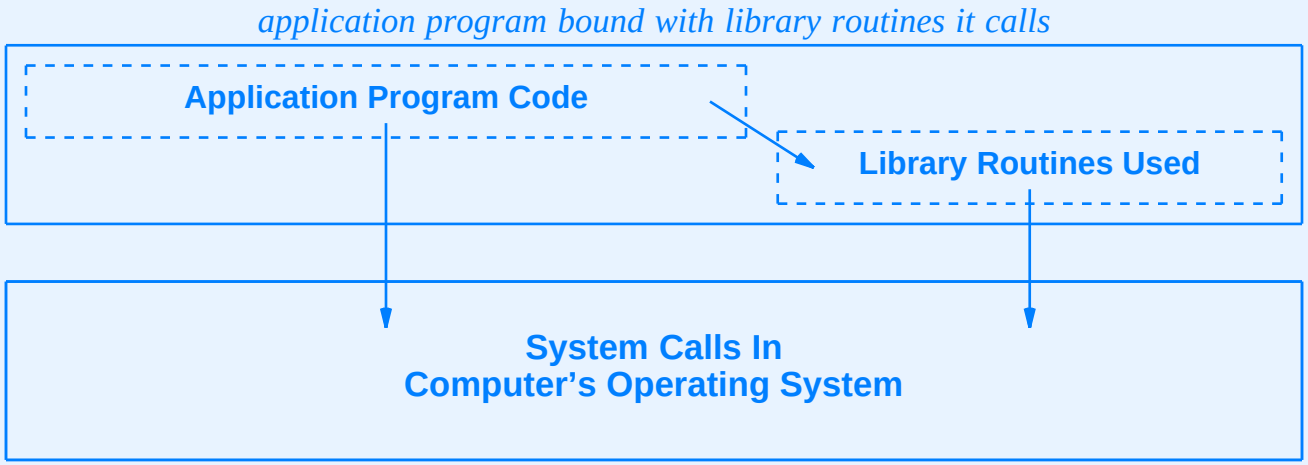
- Mapping a host name to an IP address

  gethostname(name, length)

- Obtaining the local domain

  getdomainname(name, length)

Illustration Of A Socket Library



Byte Order Conversion Routines

- Convert between network byte order and local host byte order
- If local host uses big-endian, routines have no effect

```
localshort = ntohs(netshort)
locallong = ntohl(netlong)
netshort = htons(localshort)
netlong = htonl(locallong)
```

IP Address Manipulation Routines

- Convert from dotted decimal (ASCII string) to 32-bit binary value
- Example:

address = inet_addr(string)

Other Socket Routines

- Many other functions exist
- Examples: obtain information about
  - Protocols
  - Hosts
  - Domain name

Example Client Program

```
/* whoisclient.c - main */

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

/*------------------------------------------------------------------------
 * Program:     whoisclient
 *
 * Purpose:     UNIX application program that becomes a client for the
 *              Internet "whois" service.
 *
 * Use:         whois hostname username
 *
 * Author:      Barry Shein, Boston University
 *
 * Date:        Long ago in a universe far, far away
 *
 *------------------------------------------------------------------------
 */
```

Example Client Program (Part 2)

```
main(argc, argv)
int argc;                       /* standard UNIX argument declarations  */
char *argv[];
{
  int s;                        /* socket descriptor                    */
  int len;                      /* length of received data              */
  struct sockaddr_in sa;        /* Internet socket addr. structure      */
  struct hostent *hp;           /* result of host name lookup           */
  struct servent *sp;           /* result of service lookup             */
  char buf[BUFSIZ+1];           /* buffer to read whois information     */
  char *myname;                 /* pointer to name of this program      */
  char *host;                   /* pointer to remote host name          */
  char *user;                   /* pointer to remote user name          */

  myname = argv[0];
```

Example Client (Part 3)

```
/*
 * Check that there are two command line arguments
 */
if(argc != 3) {
      fprintf(stderr, "Usage: %s host username\n", myname);
      exit(1);
}
host = argv[1];
user = argv[2];
/*
 * Look up the specified hostname
 */
if((hp = gethostbyname(host)) == NULL) {
      fprintf(stderr,"%s: %s: no such host?\n", myname, host);
      exit(1);
}
/*
 * Put host's address and address type into socket structure
 */
bcopy((char *)hp->h_addr, (char *)&sa.sin_addr, hp->h_length);
sa.sin_family = hp->h_addrtype;
```

Example Client (Part 4)

```
/*
 * Look up the socket number for the WHOIS service
 */
if((sp = getservbyname("whois","tcp")) == NULL) {
      fprintf(stderr,"%s: No whois service on this host\n", myname);
      exit(1);
}
/*
 * Put the whois socket number into the socket structure.
 */
sa.sin_port = sp->s_port;
/*
 * Allocate an open socket
 */
if((s = socket(hp->h_addrtype, SOCK_STREAM, 0)) < 0) {
      perror("socket");
      exit(1);
}
```

Example Client (Part 5)

```
    /*
     * Connect to the remote server
     */
    if(connect(s, &sa, sizeof sa) < 0) {
            perror("connect");
            exit(1);
    }
    /*
     * Send the request
     */
    if(write(s, user, strlen(user)) != strlen(user)) {
            fprintf(stderr, "%s: write error\n", myname);
            exit(1);
    }
    /*
     * Read the reply and put to user's output
     */
    while( (len = read(s, buf, BUFSIZ)) > 0)
            write(1, buf, len);
    close(s);
    exit(0);
}
```

Example Server Program

```
/* whoisserver.c - main */

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <pwd.h>

/*------------------------------------------------------------------------
 * Program:     whoisserver
 *
 * Purpose:     UNIX application program that acts as a server for
 *              the "whois" service on the local machine.  It listens
 *              on well-known WHOIS port (43) and answers queries from
 *              clients.  This program requires super-user privilege to
 *              run.
 *
 * Use:         whois hostname username
 *
```

Example Server (Part 2)

```
 * Author:      Barry Shein, Boston University
 *
 * Date:        Long ago in a universe far, far away
 *
 *-----------------------------------------------------------------------
 */

#define BACKLOG         5       /* # of requests we're willing to queue */
#define MAXHOSTNAME     32      /* maximum host name length we tolerate */

main(argc, argv)
int argc;                       /* standard UNIX argument declarations  */
char *argv[];
{
  int s, t;                     /* socket descriptors                   */
  int i;                        /* general purpose integer              */
  struct sockaddr_in sa, isa;   /* Internet socket address structure    */
  struct hostent *hp;           /* result of host name lookup           */
  char *myname;                 /* pointer to name of this program      */
  struct servent *sp;           /* result of service lookup             */
  char localhost[MAXHOSTNAME+1];/* local host name as character string  */
```

Example Server (Part 3)

```
myname = argv[0];
/*
 * Look up the WHOIS service entry
 */
if((sp = getservbyname("whois","tcp")) == NULL) {
      fprintf(stderr, "%s: No whois service on this host\n", myname);
      exit(1);
}
/*
 * Get our own host information
 */
gethostname(localhost, MAXHOSTNAME);
if((hp = gethostbyname(localhost)) == NULL) {
      fprintf(stderr, "%s: cannot get local host info?\n", myname);
      exit(1);
}
```

Example Server (Part 4)

```
/*
 * Put the WHOIS socket number and our address info
 * into the socket structure
 */
sa.sin_port = sp->s_port;
bcopy((char *)hp->h_addr, (char *)&sa.sin_addr, hp->h_length);
sa.sin_family = hp->h_addrtype;
/*
 * Allocate an open socket for incoming connections
 */
if((s = socket(hp->h_addrtype, SOCK_STREAM, 0)) < 0) {
      perror("socket");
      exit(1);
}
/*
 * Bind the socket to the service port
 * so we hear incoming connections
 */
if(bind(s, &sa, sizeof sa) < 0) {
      perror("bind");
      exit(1);
}
```

Example Server (Part 5)

```
  /*
   * Set maximum connections we will fall behind
   */
  listen(s, BACKLOG);
  /*
   * Go into an infinite loop waiting for new connections
   */
  while(1) {
        i = sizeof isa;
        /*
         * We hang in accept() while waiting for new customers
         */
        if((t = accept(s, &isa, &i)) < 0) {
          perror("accept");
          exit(1);
        }
        whois(t);           /* perform the actual WHOIS service */
        close(t);
  }
}
```

Example Server (Part 6)

```
/*
 * Get the WHOIS request from remote host and format a reply.
 */
whois(sock)
int sock;
{
  struct passwd *p;
  char buf[BUFSIZ+1];
  int i;

  /*
   * Get one line request
   */
  if( (i = read(sock, buf, BUFSIZ)) <= 0)
        return;
  buf[i] = '\0';        /* Null terminate */
```

Example Server (Part 7)

```
  /*
   * Look up the requested user and format reply
   */
  if((p = getpwnam(buf)) == NULL)
        strcpy(buf,"User not found\n");
  else
        sprintf(buf, "%s: %s\n", p->pw_name, p->pw_gecos);
  /*
   * Return reply
   */
  write(sock, buf, strlen(buf));
  return;
}
```

Summary

- Socket API
  - Invented for BSD Unix
  - Not official part of TCP/IP
  - De facto standard in the industry
  - Used with TCP or UDP
  - Large set of functions
- General paradigm: create socket and then use a set of functions to specify details


Questions?

PART XXII

BOOTSTRAP AND AUTOCONFIGURATION (DHCP)

System Startup

- To keep protocol software general
  - IP stack designed with many parameters
  - Values filled in when system starts
- Two possible sources of information
  - Local storage device (e.g., disk)
  - Server on the network

Bootstrapping

- BOOTstrap Protocol (BOOTP)
  - Early alternative to RARP
  - Provided more than just an IP address
  - Obtained configuration parameters from a server
  - Used UDP
- Dynamic Host Configuration Protocol (DHCP)
  - Replaces and extends BOOTP
  - Provides dynamic address assignment

Apparent Contradiction

- DHCP used to obtain parameters for an IP stack
- DHCP uses IP and UDP to obtain the parameters
- Stack must be initialized before being initialized

Solving The Apparent Contradiction

- DHCP runs as application
- Only needs basic facilities
- In particular:

An application program can use the limited broadcast IP address to force IP to broadcast a datagram on the local network before IP has discovered the IP address of the local network or the machine's IP address.

- Note: server cannot use ARP when replying to client because client does not know its own IP address

DHCP Retransmission

- Client handles retransmission
- Initial timeout selected at random
- Timeout for successive retransmissions doubled

Two-Step Bootstrap

- DHCP provides information, not data
- Client receives
  - Name of file that contains boot image
  - Address of server
- Client must use another means to obtain the image to run (typically TFTP)

Dynamic Address Assignment

- Needed by ISPs
  - Client obtains an IP address and uses temporarily
  - When client finishes, address is available for another client
- Also used on many corporate networks

DHCP Address Assignment

- Backward compatible with BOOTP
- Can assign addresses in three ways
  - Manual (manager specifies binding as in BOOTP)
  - Automatic (address assigned by server, and machine retains same address)
  - Dynamic (address assigned by server, but machine may obtain new address for successive request)
- Manager chooses type of assignment for each address

DHCP Support For Autoconfiguration

Because it allows a host to obtain all the parameters needed for communication without manual intervention, DHCP permits autoconfiguration. Autoconfiguration is, of course, subject to administrative constraints.

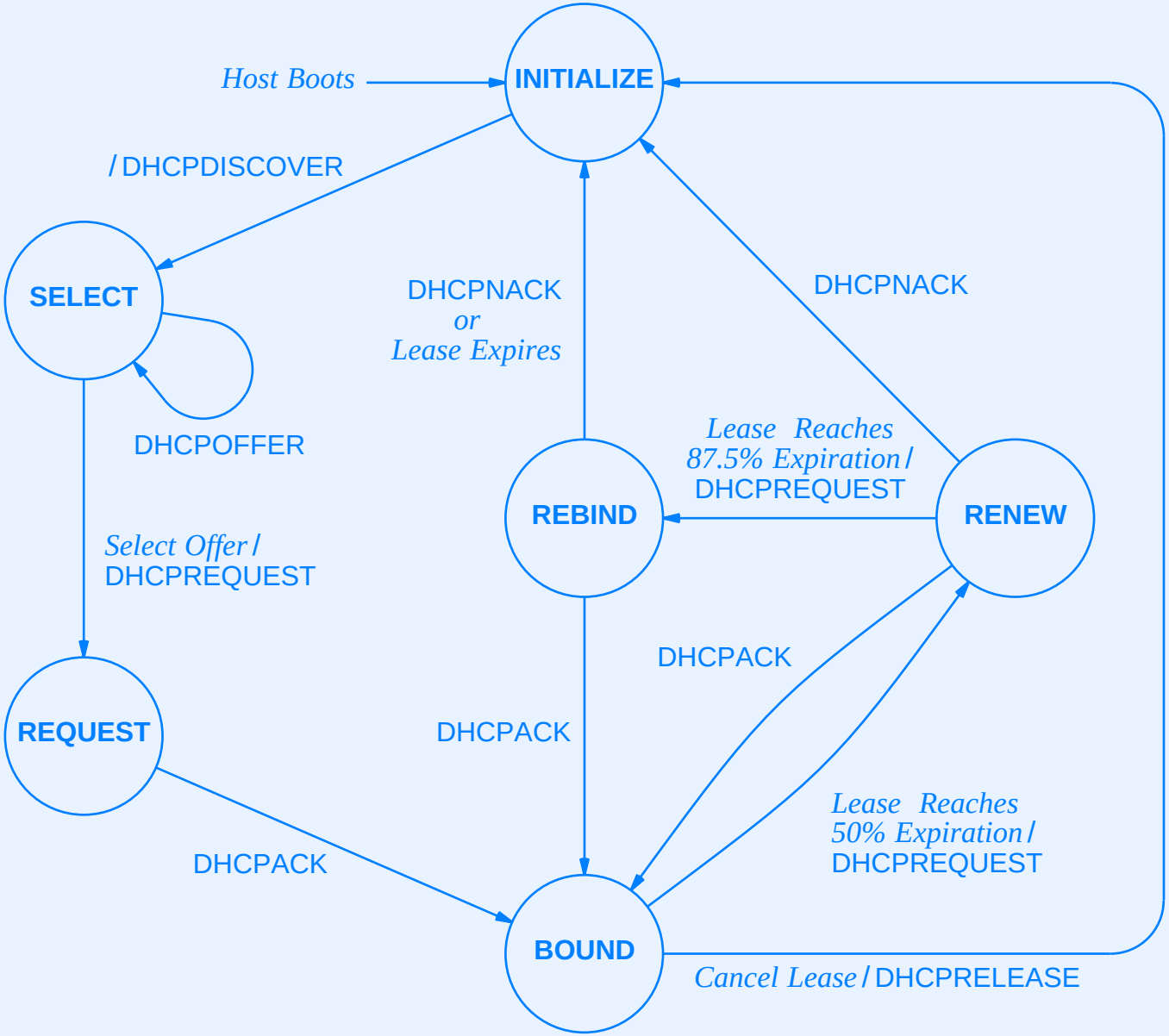
Dynamic Address Assignment

- Client is granted a *lease* on an address
- Server specifies length of lease
- At end of lease, client must renew lease or stop using address
- Actions controlled by finite state machine

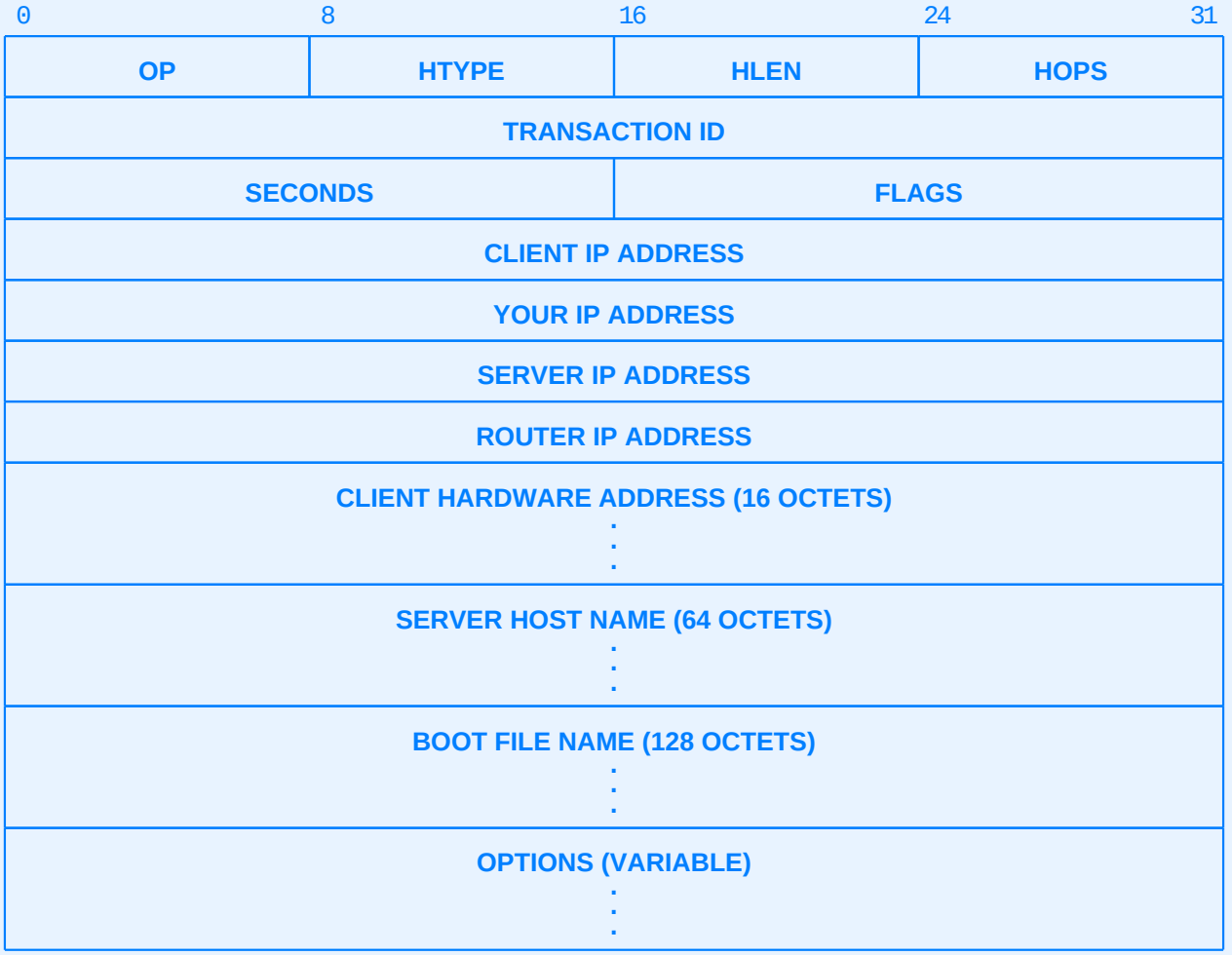
Server Contact

To use DHCP, a host becomes a client by broadcasting a message to all servers on the local network. The host then collects offers from servers, selects one of the offers, and verifies acceptance with the server.

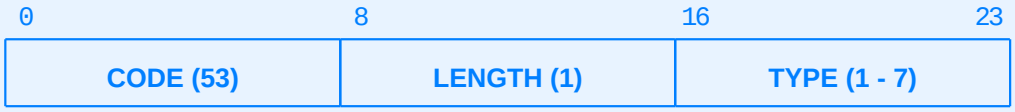
DHCP Finite State Machine



DHCP Message Format



Message Type Field



TYPE FIELD	Corresponding DHCP Message Type
1	DHCPDISCOVER
2	DHCPOFFER
3	DHCPREQUEST
4	DHCPDECLINE
5	DHCPACK
6	DHCPNACK
7	DHCPRELEASE
8	DHCPINFORM

Questions For Discussion

- Explain the relationship between DHCP and DNS
- What basic facility is needed? Why?

Summary

- Two protocols available for bootstrapping
  - BOOTP (static binding of IP address to computer)
  - DHCP (extension of BOOTP that adds dynamic binding of IP addresses)
- DHCP
  - Server grants lease for an address
  - Lease specifies length of time
  - Host must renew lease or stop using address when lease expires
  - Actions controlled by finite state machine


Questions?

PART XXIII

DOMAIN NAME SYSTEM (DNS)

Names For Computers

- Humans prefer pronounceable names rather than numeric addresses
- Two possibilities
  - Flat namespace
  - Hierarchical namespace

Naming Hierarchy

- Two possibilities
  - According to network topology
  - By organizational structure (independent of physical networks)
- Internet uses the latter

Internet Hierarchy

In a TCP/IP internet, hierarchical machine names are assigned according to the structure of organizations that obtain authority for parts of the namespace, not necessarily according to the structure of the physical network interconnections.

Internet Domain Names

- Flexible hierarchy
  - Universal naming scheme (same everywhere)
  - Each organization determines internal naming structure
- Mechanism known as *Domain Name System* (*DNS*)
- Name assigned to a computer known as *domain name*

Domain Name Syntax

- Set of *labels* separated by delimiter character (period)
- Example

cs.*purdue*.*edu*

- Three labels: *cs*, *purdue*, and *edu*
- String *purdue*.*edu* is also a domain
- Top-level domain is *edu*

Original Top-Level Domains

Domain Name	Assigned To
com	Commercial organizations
edu	Educational institutions (4-year)
gov	Government institutions
mil	Military groups
net	Major network support centers
org	Organizations other than those above
arpa	Temporary ARPANET domain (obsolete)
int	International organizations
country code	Each country (geographic scheme)

- Meaning assigned to each
- Three domains considered generic

  .com

  .net

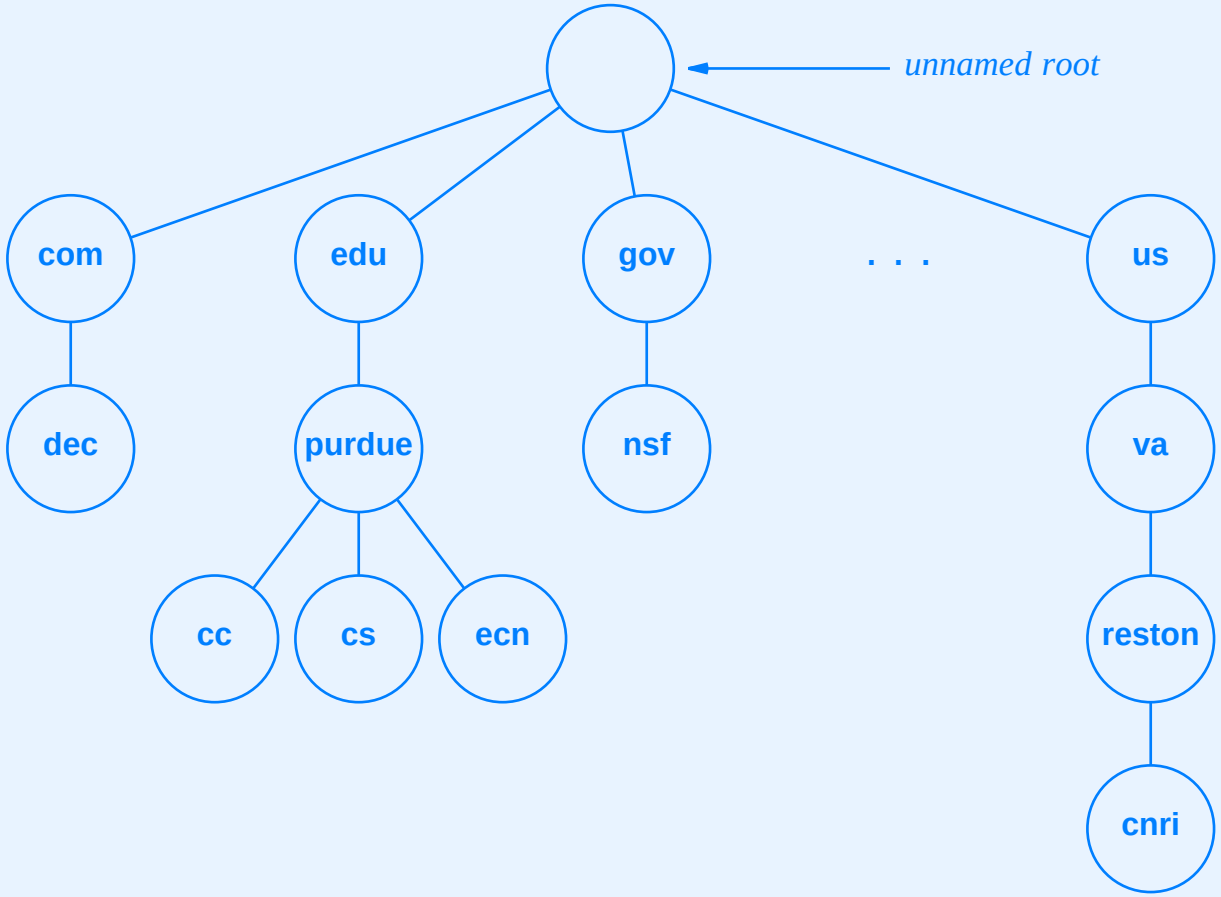
  .org

New Top-Level Domains

Domain Name	Assigned To
aero	Air-Transport Industry
biz	Businesses
coop	Non-Profit Cooperatives
info	Unrestricted
museum	Museums
name	Individuals
pro	Professionals (accountants, lawyers, physicians)

- Proponents argued (incorrectly) that DNS would collapse without additional TLDs
- New TLDs created legal nightmare

Illustration Of Part Of The DNS Tree



Authority For Names

- Authority delegated down the tree
- Example
  - Purdue University registers under top level domain *.edu* and receives authority for domain *purdue . edu*
  - Computer Science Department at Purdue registers with the Purdue authority, and becomes the authority for *cs . purdue . edu*
  - Owner of a lab in the CS Department registers with the departmental authority, and becomes the authority for *xinu . cs . purdue . edu*

DNS Database

- Record has (name, class)
- Class specifies type of object (e.g., computer, email exchanger)
- Consequence:

A given name may map to more than one item in the domain system. The client specifies the type of object desired when resolving a name, and the server returns objects of that type.

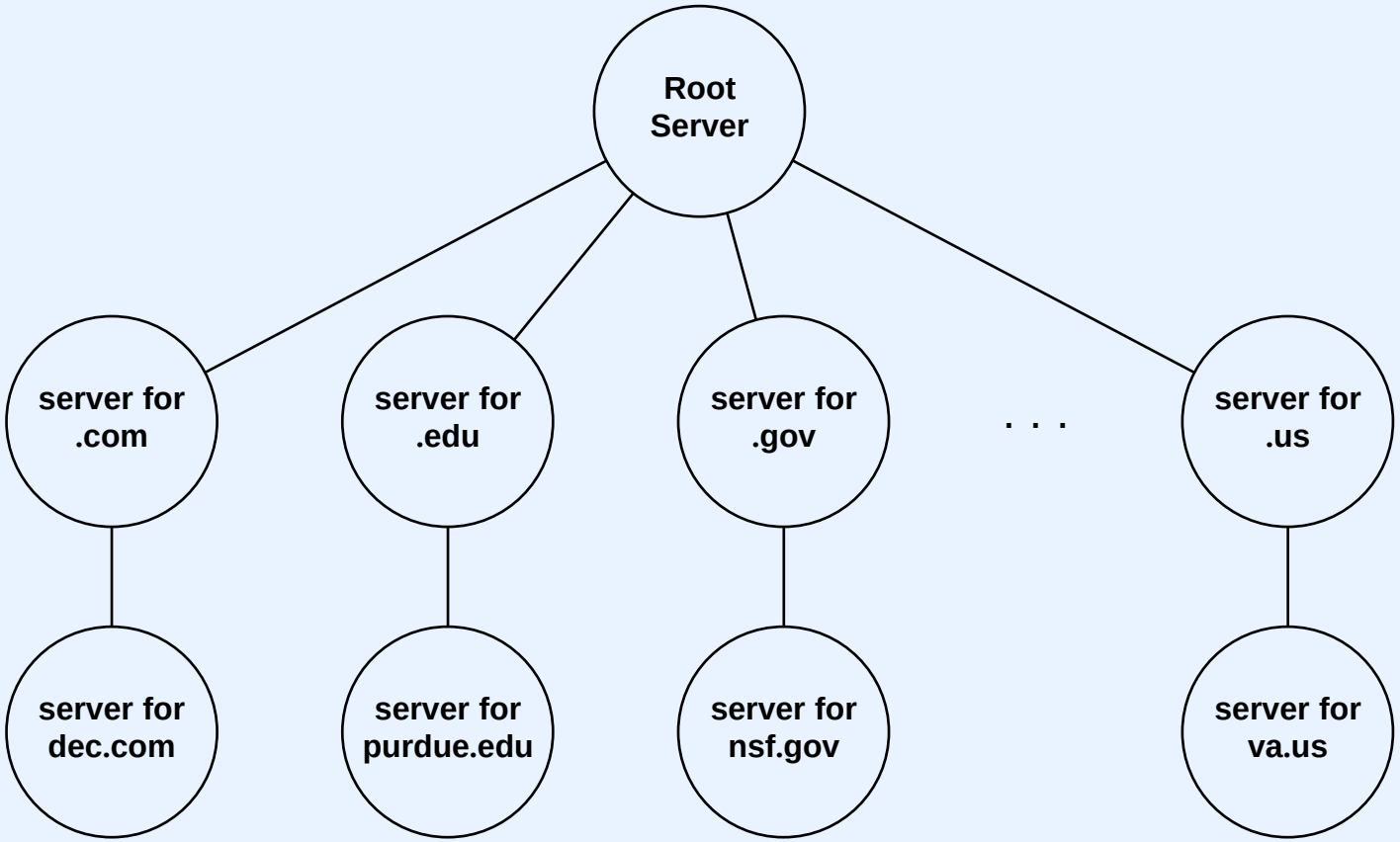
Mapping Domain Names To Addresses

- DNS uses a set of on-line servers
- Servers arranged in tree
- Given server can handle entire subtree
  - Example: ISP manages domain names for its clients (including corporations)

Terminology

- DNS server known as *name server*
- DNS client software known as *resolver*

Illustration Of Topology Among DNS Servers



In Practice

- Single server can handle multiple levels of the naming tree
- Example: root server handles all top-level domains

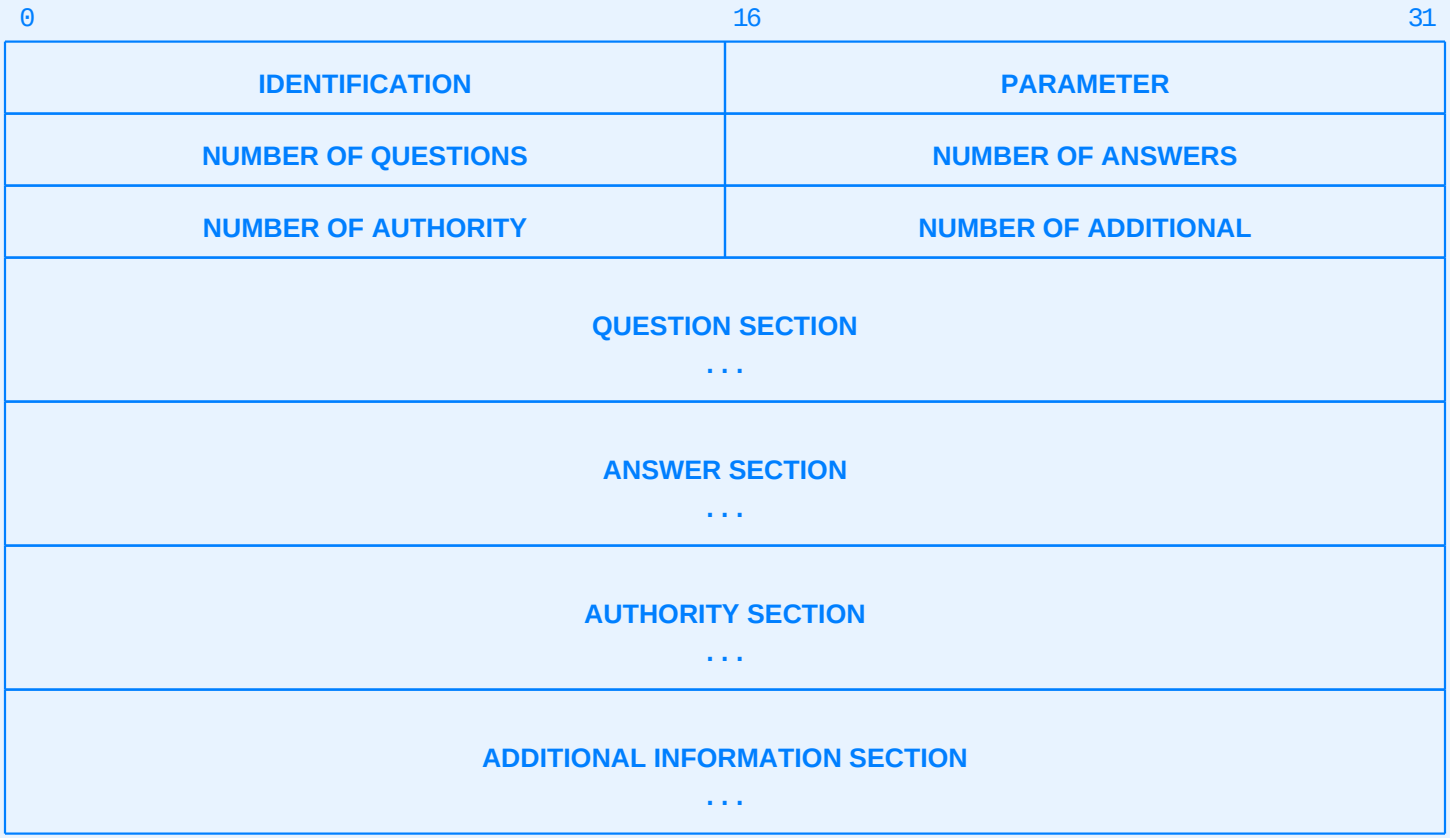
Domain Name Resolution

- Conceptually, must search from root of tree downward
- In practice
  - Every name server knows location of a root server
  - Only contacts root if no subdomain known
  - Lookup always starts with local server first (host can learn address of DNS server from DHCP)

Efficient Translation

- Facts
  - Most lookups refer to local names
  - Name-to-address bindings change infrequently
  - User is likely to repeat same lookup
- To increase efficiency
  - Initial contact begins with local name server
  - Every server caches answers (owner specifies cache timeout)

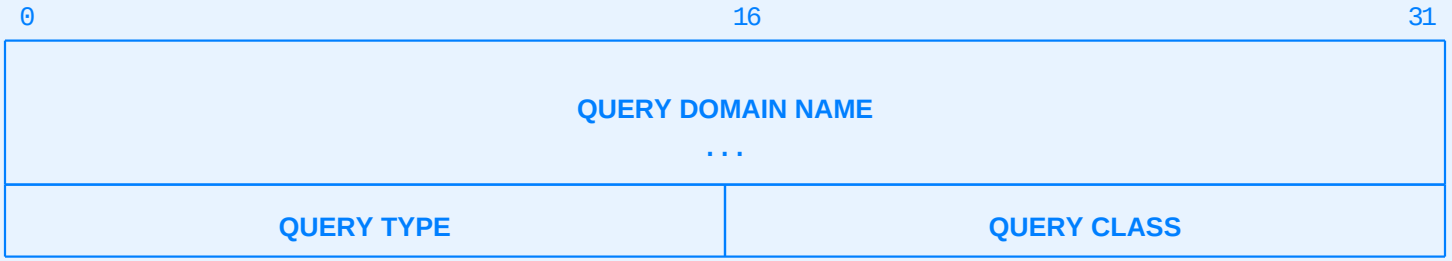
Domain Server Message Format



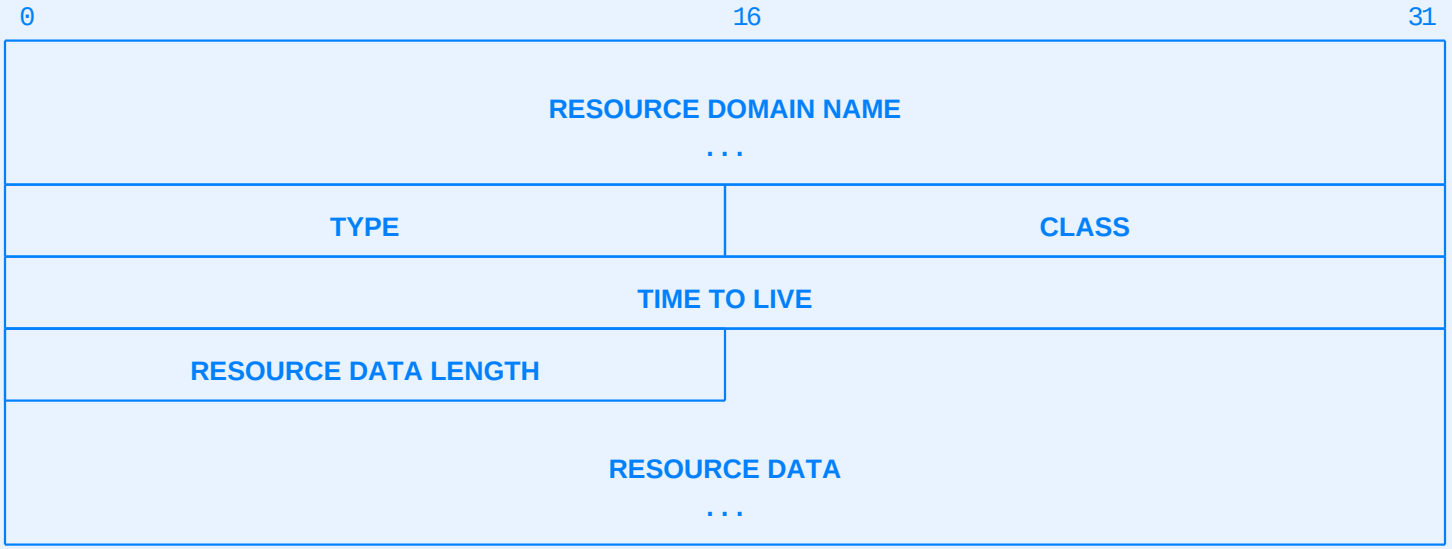
Parameter Bits

Bit of PARAMETER field	Meaning
0	Operation: 0 Query 1 Response
1-4	Query Type: 0 Standard 1 Inverse 2 Server status request 3 Completion (now obsolete) 4 Notify 5 Update
5	Set if answer authoritative
6	Set if message truncated
7	Set if recursion desired
8	Set if recursion available
9	Set if data is authenticated
10	Set if checking is disabled
11	Reserved
12-15	Response Type: 0 No error 1 Format error in query 2 Server failure 3 Name does not exist 5 Refused 6 Name exists when it should not 7 RR set exists 8 RR set that should exist does not 9 Server not authoritative for the zone 10 Name not contained in zone

Format Of Question Section



Format Of Resource Records



Abbreviation Of Domain Names

- DNS only recognizes full domain names
- Client software allows abbreviation

Example Of Domain Name Abbreviation

- Client configured with suffix list
  - .cs.purdue.edu
  - .cc.purdue.edu
  - .purdue.edu
  - *null*
- User enters abbreviation *xinu*
- Client tries the following in order
  - xinu.cs.purdue.edu
  - xinu.cc.purdue.edu
  - xinu.purdue.edu
  - xinu

The Point About Abbreviation

The Domain Name System only maps full domain names into addresses; abbreviations are not part of the Domain Name System itself, but are introduced by client software to make local names convenient for users.

Inverse Query

- Map in reverse direction
- Excessive overhead
- May not have unique answer
- Not used in practice

Pointer Query

- Special case of inverse mapping
- Convert IP address to domain name
- Trick: write IP address as a string and look up as a name

Example Of Pointer Query

- Start with dotted decimal address such as

  aaa.bbb.ccc.ddd

- Rearrange dotted decimal representation as a string:

  ddd.ccc.bbb.aaa.in-addr.arpa

- Look up using a *pointer query* type

Object Types That DNS Supports

Type	Meaning	Contents
A	Host Address	32-bit IP address
CNAME	Canonical Name	Canonical domain name for an alias
HINFO	CPU & OS	Name of CPU and operating system
MINFO	Mailbox info	Information about a mailbox or mail list
MX	Mail Exchanger	16-bit preference and name of host that acts as mail exchanger for the domain
NS	Name Server	Name of authoritative server for domain
PTR	Pointer	Domain name (like a symbolic link)
SOA	Start of Authority	Multiple fields that specify which parts of the naming hierarchy a server implements
TXT	Arbitrary text	Uninterpreted string of ASCII text
AAAA	Host Address	128-bit IPv6 address

Summary

- Domain Name System provides mapping from pronounceable names to IP addresses
- Domain names are hierarchical; top-level domains are dictated by a central authority
- Organizations can choose how to structure their domain names
- DNS uses on-line servers to answer queries
- Lookup begins with local server, which caches entries


Questions?

PART XXIV

APPLICATIONS: REMOTE LOGIN (TELNET AND RLOGIN)

Remote Interaction

- Devised when computers used (ASCII) terminals
- Terminal abstraction extended to remote access over a network

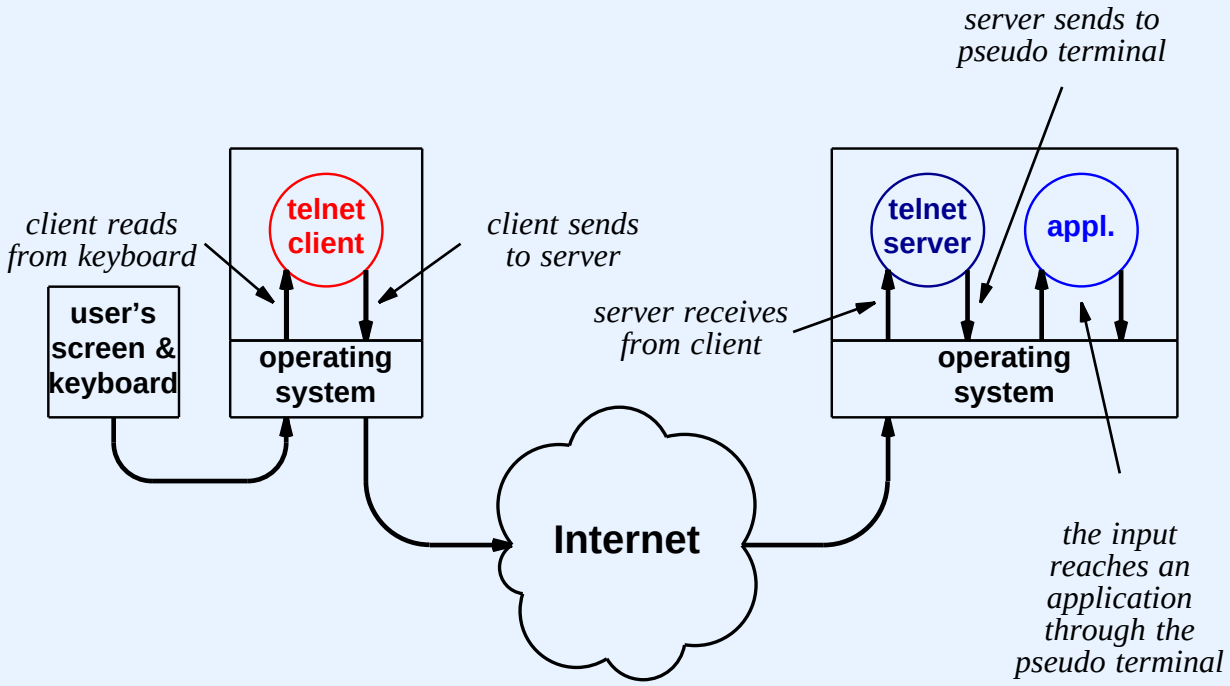
Client-Server Interaction

- Client
  - Invoked by user
  - Forms connection to remote server
  - Passes keystrokes from user's keyboard to server and displays output from server on user's screen
- Server
  - Accepts connection over the network
  - Passes incoming characters to OS as if they were typed on a local keyboard
  - Sends output over connection to client

TELNET

- Standard protocol for remote terminal access
- Three basic services
  - Defines *network virtual terminal* that provides standard interface
  - Mechanism that allows client and server to negotiate options (e.g., character set)
  - Symmetric treatment that allows either end of the connection to be a program instead of a physical keyboard and display

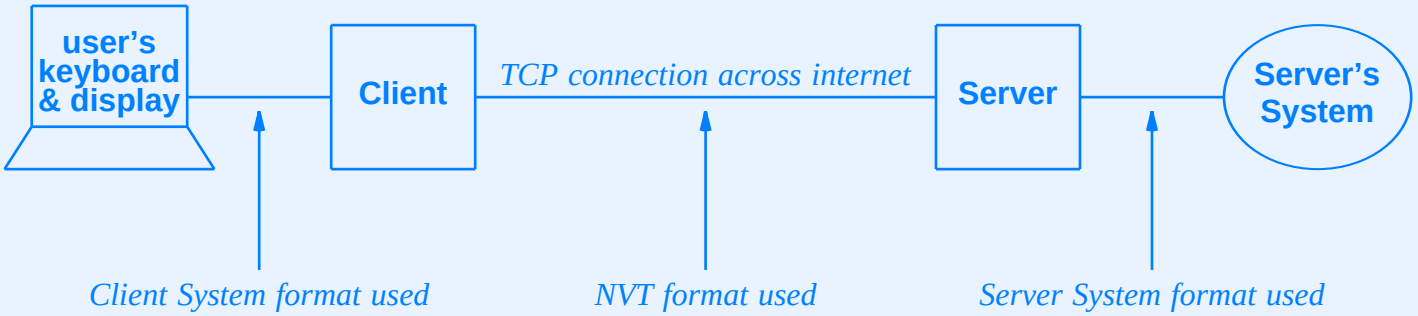
Illustration Of TELNET



Accommodating Heterogeneity

- *Network Virtual Terminal* (*NVT*) describes system-independent encoding
- TELNET client and server map NVT into local computer's representation

Illustration Of How NVT Accommodates Heterogeneity



Definition Of TELNET NVT

ASCII Control Code	Decimal Value	Assigned Meaning
NUL	0	No operation (has no effect on output)
BEL	7	Sound audible/visible signal (no motion)
BS	8	Move left one character position
HT	9	Move right to the next horizontal tab stop
LF	10	Move down (vertically) to the next line
VT	11	Move down to the next vertical tab stop
FF	12	Move to the top of the next page
CR	13	Move to the left margin on the current line
other control	–	No operation (has no effect on output)

TELNET NVT Control Functions

Signal	Meaning
IP	Interrupt Process (terminate running program)
AO	Abort Output (discard any buffered output)
AYT	Are You There (test if server is responding)
EC	Erase Character (delete the previous character)
EL	Erase Line (delete the entire current line)
SYNCH	Synchronize (clear data path until TCP urgent data point, but do interpret commands)
BRK	Break (break key or attention signal)

TELNET Commands

Command	Decimal Encoding	Meaning
IAC	255	Interpret next octet as command (when the IAC octet appears as data, the sender doubles it and sends the 2-octet sequence IAC-IAC)
DON'T	254	Denial of request to perform specified option
DO	253	Approval to allow specified option
WON'T	252	Refusal to perform specified option
WILL	251	Agreement to perform specified option
SB	250	Start of option subnegotiation
GA	249	The ''go ahead'' signal
EL	248	The ''erase line'' signal
EC	247	The ''erase character'' signal
AYT	246	The ''are you there'' signal
AO	245	The ''abort output'' signal
IP	244	The ''interrupt process'' signal
BRK	243	The ''break'' signal
DMARK	242	The data stream portion of a SYNCH (always accompanied by TCP Urgent notification)
NOP	241	No operation
SE	240	End of option subnegotiation
EOR	239	End of record

TELNET Control Sequences And TCP

TELNET cannot rely on the conventional data stream alone to carry control sequences between client and server because a misbehaving application that needs to be controlled might inadvertently block the data stream.

- Solution: use TCP's *urgent data* to send control sequences

TELNET Option Negotiation

TELNET uses a symmetric option negotiation mechanism to allow clients and servers to reconfigure the parameters controlling their interaction. Because all TELNET software understands a basic NVT protocol, clients and servers can interoperate even if one understands options another does not.

Remote Login (rlogin)

- Invented for BSD Unix
- Includes facilities specifically for Unix
- Allows manager to configure a set of computers so that if two or more computers have same login id, X, the logins are owned by the same individual
- Permits other forms of authentication

Remote Shell (rsh)

- Similar to rlogin
- Also part of BSD Unix
- Allows remote execution of a single command

Secure Remote Login (ssh)

- Alternative to TELNET/rlogin
- Transport layer protocol with service authentication
- User authentication protocol
- Connection protocol
  - Multiplexes multiple transfers
  - Uses encryption for privacy

Port Forwarding

- Novel aspect of ssh
- Similar to NAT
- Permits incoming TCP connection to be forwarded across secure tunnel

Remote Desktop

- Intended for systems that have a GUI interface
- Allows a remote user to see screen of remote system and use mouse as well as keyboard
- Examples include
  - Virtual Network Computing (VNC)
  - Remote Desktop Protocol (RDP)

Summary

- Remote interaction allows client software to connect local keyboard and screen to remote system
- Standard protocol is TELNET
- Alternatives include *rlogin*, *rsh*, and *ssh*
- Remote desktop extends remote access to handle GUI inteface


Questions?

PART XXV

APPLICATIONS:
FILE TRANSFER AND ACCESS
(FTP, TFTP, NFS)

On-Line File Sharing

- Always a popular application
- Two basic paradigms
  - Whole-file copying
  - Piecewise file access
- Piecewise access mechanism
  - Opaque: application uses special facilities to access remote file
  - Transparent: application uses same facilities to access local and remote files

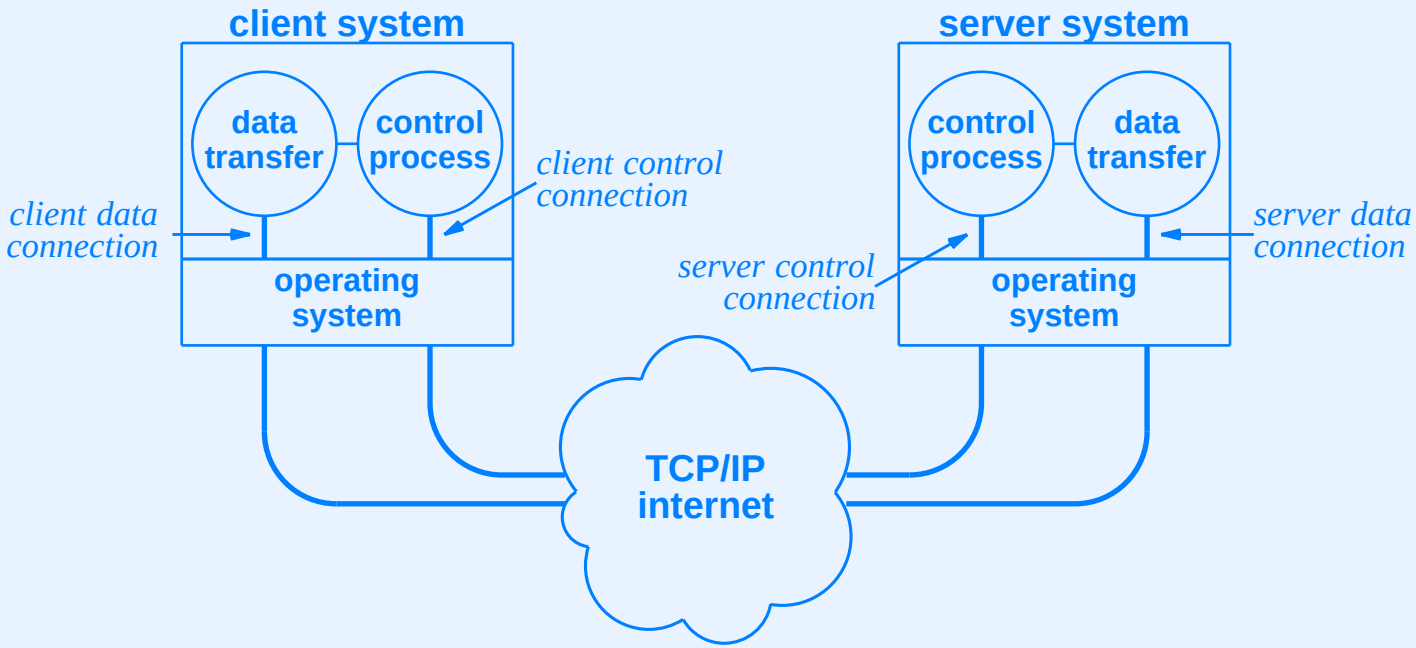
File Transfer

- Whole file copying
- Client
  - Contacts server
  - Specifies file
  - Specifies transfer direction
- Server
  - Maintains set of files on local disk
  - Waits for contact
  - Honors request from client

File Transfer Protocol (FTP)

- Major TCP/IP protocol for whole-file copying
- Uses TCP for transport
- Features
  - Interactive access
  - Format specification (ASCII or EBCDIC)
  - Authentication control (login and password)

FTP Process Model



- Separate processes handle
  - Interaction with user
  - Individual transfer requests

FTP's Use of TCP Connections

Data transfer connections and the data transfer processes that use them can be created dynamically when needed, but the control connection persists throughout a session. Once the control connection disappears, the session is terminated and the software at both ends terminates all data transfer processes.

Control Connection Vs. Data Connection

- For data transfer, client side becomes server and server side becomes client
- Client
  - Creates process to handle data transfer
  - Allocates port and sends number to server over control connection
  - Process waits for contact
- Server
  - Receives request
  - Creates process to handle data transfer
  - Process contacts client-side

Question For Discussion

- What special relationship is required between FTP and NAT?

Interactive Use Of FTP

- Initially a command-line interface
  - User invokes client and specifies remote server
  - User logs in and enters password
  - User issues series of requests
  - User closes connection
- Currently
  - Most FTP initiated through browser
  - User enters URL or clicks on link
  - Browser uses FTP to contact remote server and obtain list of files
  - User selects file for download

Anonymous FTP

- Login *anonymous*
- Password *guest*
- Used for ‘‘open’’ FTP site (where all files are publicly available
- Typically used by browsers

Secure File Transfer Protocols

- Secure Sockets Layer FTP (SSL-FTP)
  - Uses secure sockets layer technology
  - All transfers are confidential
- Secure File Transfer Program (sftp)
  - Almost nothing in common with FTP
  - Uses ssh tunnel
- Secure Copy (scp)
  - Derivative of Unix *remote copy (rcp)*
  - Uses ssh tunnel

Trivial File Transfer Protocol (TFTP)

- Alternative to FTP
- Whole-file copying
- Not as much functionality as FTP
- Code is much smaller
- Intended for use on Local Area Network
- Runs over UDP
- Diskless machine can use to obtain image at bootstrap

TFTP Packet Types

2-octet opcode	n octets	1 octet	n octets	1 octet
READ REQ. (1)	FILENAME	0	MODE	0

2-octet opcode	n octets	1 octet	n octets	1 octet
WRITE REQ. (2)	FILENAME	0	MODE	0

2-octet opcode	2 octets	up to 512 octets
DATA (3)	BLOCK #	DATA OCTETS...

2-octet opcode	2 octets
ACK (4)	BLOCK #

2-octet opcode	2 octets	n octets	1 octet
ERROR (5)	ERROR CODE	ERROR MESSAGE	0

TFTP Retransmission

- Symmetric (both sides implement timeout and retransmission)
- Data block is request for ACK
- ACK is request for next data block

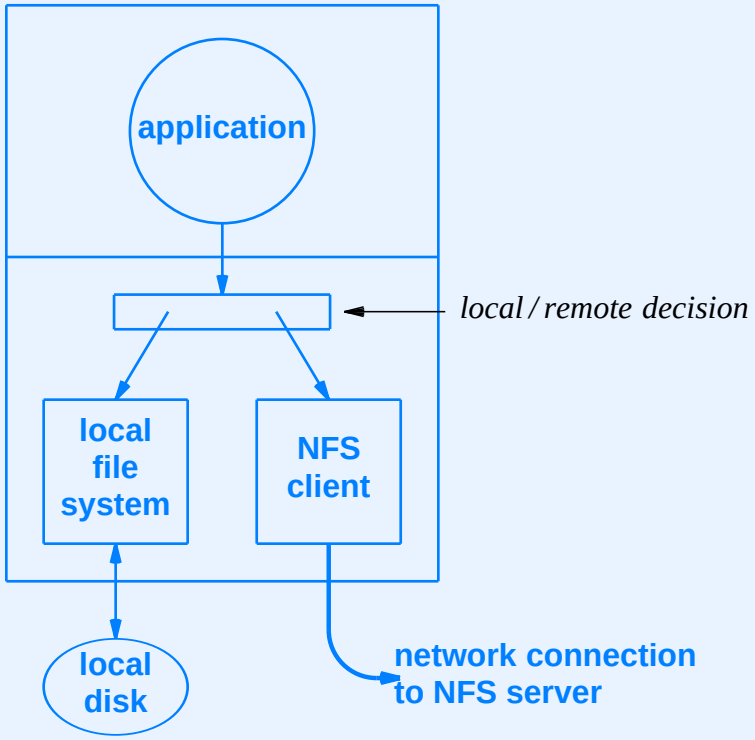
Sorcerer's Apprentice Bug

- Consequence of symmetric retransmission
- Duplicate packet is perceived as second request, which generates another transmission
- Duplicate response triggers duplicate packets from the other end
- Cycle continues

Network File System (NFS)

- Protocol for file access, not copying
- Developed by Sun Microsystems, now part of TCP/IP standards
- Transparent (application cannot tell that file is remote)

NFS Implementation



Remote Procedure Call (RPC)

- Also developed by Sun Microsystems, now part of TCP/IP standards
- Used in implementation of NFS
- Relies on *eXternal Data Representation* (*XDR*) standard for conversion of data items between heterogeneous computers

Summary

- Two paradigms for remote file sharing
  - Whole file copying
  - Piecewise file access
- File Transfer Protocol (FTP)
  - Standard protocol for file copying
  - Separate TCP connection for each data transfer
  - Client and server roles reversed for data connection
- Examples of secure alternatives to FTP
  - SSL-FTP, sftp, and scp

Summary
(continued)

- Trivial File Transfer Protocol (TFTP)
  - Alternative to FTP that uses UDP
  - Symmetric retransmission scheme
  - Packet duplication can result in Sorcerer's Apprentice problem
- Network File System (NFS)
  - Standard protocol for piecewise file access
  - Uses RPC and XDR


Questions?

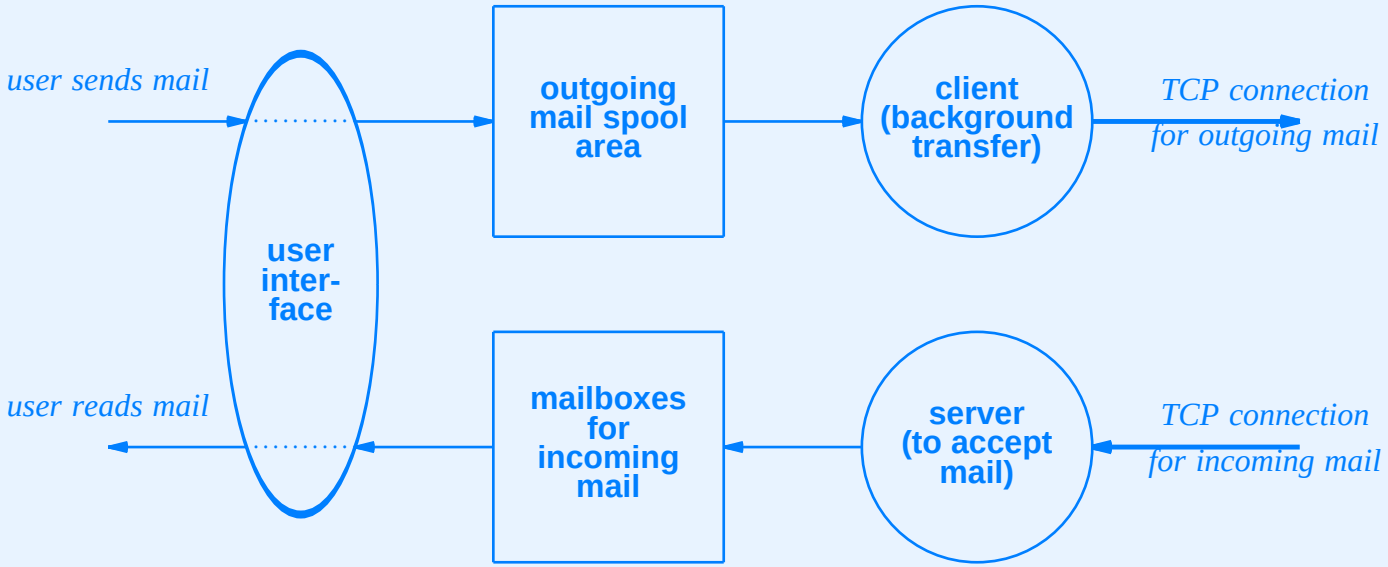
PART XXVI

APPLICATIONS: ELECTRONIC MAIL (SMTP, POP, IMAP, MIME)

Electronic Mail

- Among most widely used Internet services
- Two major components
  - User interface
  - Mail transfer software
- Paradigm: transfer is separate background activity

Illustration Of Email System Components



Mailbox Names And Aliases

- Email destination identified by pair

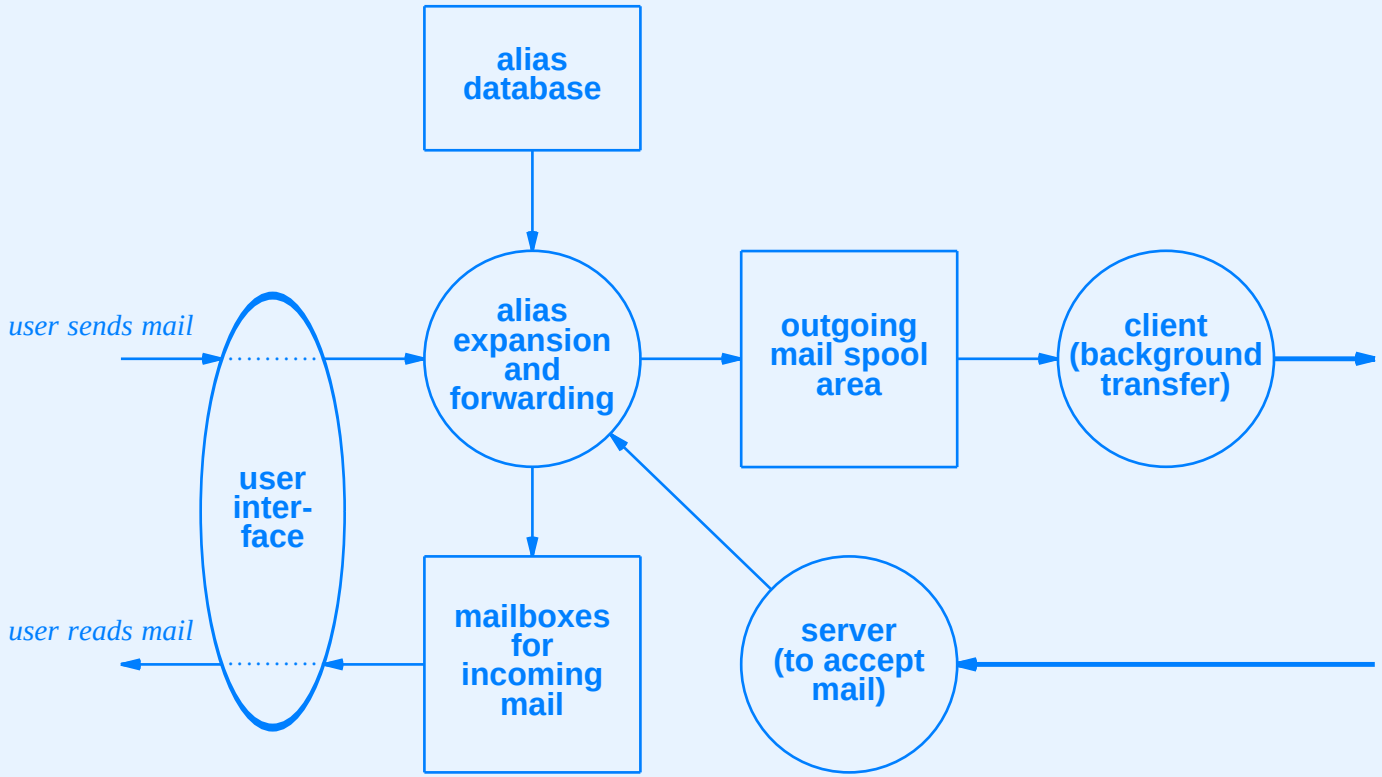
  (mailbox, computer)

- Aliases permitted (user enters alias that is expanded)

Forwarding

- Powerful idea
- Email arriving on a computer can be forwarded to an ultimate destination

Illustration Of Aliases And Forwarding



TCP/IP Standards For Email

- Syntax for email addresses
- Format of email message
- Protocols for email transfer and mailbox access

Email Address Syntax

- Mailbox identified by string

mailbox@computer

- String *computer* is domain name of computer on which a mailbox resides
- String *mailbox* is unique mailbox name on the destination computer

Format Of Email Message

- Message consists of
  - Header
  - Blank line
  - Body of message
- Headers have form

keyword : information

- Standard given in RFC 2822

Protocol For Email Transfer

- Specifies interaction between transfer components
  - Transfer client
  - Transfer server
- Standard protocol is *Simple Mail Transfer Protocol* (*SMTP*)

SMTP

- Application-level protocol
- Uses TCP
- Commands and responses encoded in ASCII

Example Of SMTP

```
S: 220 Beta.GOV Simple Mail Transfer Service Ready
C: HELO Alpha.EDU
S: 250 Beta.GOV

C: MAIL FROM:<Smith@Alpha.EDU>
S: 250 OK

C: RCPT TO:<Jones@Beta.GOV>
S: 250 OK

C: RCPT TO:<Green@Beta.GOV>
S: 550 No such user here

C: RCPT TO:<Brown@Beta.GOV>
S: 250 OK

C: DATA
S: 354 Start mail input; end with <CR><LF>.<CR><LF>
C: ...sends body of mail message...
C: ...continues for as many lines as message contains
C: <CR><LF>.<CR><LF>
S: 250 OK

C: QUIT
S: 221 Beta.GOV Service closing transmission channel
```

Protocol For Mailbox Access

- Used when user's mailbox resides on remote computer
- Especially helpful when user's local computer is not always on-line
- Two protocols exist
  - *Post Office Protocol* version 3 (*POP3*)
  - *Internet Message Access Protocol* (*IMAP*)
- Each provides same basic functionality
  - User authentication
  - Mailbox access commands

Multipurpose Internet Mail Extensions (MIME)

- Permits nontextual data to be sent in email
  - Graphics image
  - Voice or video clip
- Sender
  - Encodes binary item into printable characters
  - Places in email message for transfer
- Receiver
  - Receives email message containing encoded item
  - Decodes message to extract original binary value

MIME Header

- Header in email message describes encoding used
- Example

```
From: bill@acollege.edu
To: john@example.com
MIME-Version: 1.0
Content-Type: image/jpeg
Content-Transfer-Encoding: base64

...data for the image...
```

Seven Basic MIME Types

Content Type	Used When Data In the Message Is
text	Textual (e.g. a document).
image	A still photograph or computer-generated image
audio	A sound recording
video	A video recording that includes motion
application	Raw data for a program
multipart	Multiple messages that each have a separate content type and encoding
message	An entire e-mail message (e.g., a memo that has been forwarded) or an external reference to a message (e.g., an FTP server and file name)

Example Of Mixed / Multipart Message

```
From: bill@acollege.edu
To: john@example.com
MIME-Version: 1.0
Content-Type: Multipart/Mixed; Boundary=StartOfNextPart

--StartOfNextPart
Content-Type: text/plain
Content-Transfer-Encoding: 7bit
John,
   Here is the photo of our research lab I promised
to send you.  You can see the equipment you donated.

Thanks again, Bill

--StartOfNextPart
Content-Type: image/jpeg
Content-Transfer-Encoding: base64
     ...data for the image...
```

Summary

- Email operates at application layer
- Conceptual separation between
  - User interface
  - Mail transfer components
- Simple Mail Transfer Protocol (SMTP)
  - Standard for transfer
  - Uses ASCII encoding
- Post Office Protocol (POP) And Internet Mail Access Protocol (IMAP) allow access of remote mailbox.
- Multipurpose Internet Mail Extensions (MIME) permits transfer of nontextual information (e.g., images)


Questions?

PART XXVII

APPLICATIONS: WORLD WIDE WEB (HTTP)

World Wide Web

- Distributed hypermedia paradigm
- Major service on the Internet
- Use surpassed file transfer in 1995

Web Page Identifier

- Known as *Uniform Resource Locator* (*URL*)
- Encodes
  - Access protocol to use
  - Domain name of server
  - Protocol port number (optional)
  - Path through server's file system (optional)
  - Parameters (optional)
  - Query (optional)
- Format

http: // *hostname [: port] / path [; parameters] [? query]*

Web Standards

- Separate standards for
  - Representation
  - Transfer

Representation

- *HyperText Markup Language* (*HTML)*
- Document contains text plus embedded links
- HTML gives guidelines for display, not details
- Consequence: two browsers may choose to display same document differently

Transfer

- Used between browser and web server
- Protocol is *HyperText Transfer Protocol* (*HTTP*)
- Runs over TCP

HTTP Characteristics

- Application level
- Request / response paradigm
- Stateless
- Permits bi-directional transfer
- Offers capability negotiation
- Support for caching
- Support for intermediaries

HTTP Operation

- Browser sends requests to which server replies
- Typical request: *GET* used to fetch document
- Example

GET http://www.cs.purdue.edu/people/comer/ HTTP/1.1

- Relative URL also permitted

GET /people/comer/ HTTP/1.1

Error Messages

- HTTP includes set of error responses
- Server can format error as HTML message for user or use internal form and allow browser to format message

Persistent Connections

- HTTP version 1.0 uses one TCP connection per transfer
  - Browser forms TCP connection to server
  - Browser sends GET request
  - Server returns header describing item
  - Server returns item
  - Server closes connection
- HTTP version 1.1 permits connection to persist across multiple requests

HTTP Headers

HTTP uses MIME-like headers to carry meta information. Both browsers and servers send headers that allow them to negotiate agreement on the document representation and encoding to be used.

Handing Persistence

To allow a TCP connection to persist through multiple requests and responses, HTTP sends a length before each response. If it does not know the length, a server informs the client, sends the response, and then closes the connection.

Headers And Length Encoding

- HTTP headers use same syntax as email headers
  - Lines of text followed by blank line
  - Lines of text have form *keyword:information*
- For persistent connection header specifies length (in octets) of data item that follows

Items That Can Appear In An HTTP Header

Header	Meaning
Content-Length	Size of item in octets
Content-Type	Type of item
Content-Encoding	Encoding used for item
Content-Language	Language(s) used in item

Example Of Header

```
Content-Length:   34
Content-Language: english
Content-Encoding: ascii

<HTML> A trivial example. </HTML>
```

- Note: if length is not known in advance, server can inform browser that connection will close following transfer

Connection: close

Negotiation

- Either server or browser can initiate
- Items sent in headers
- Can specify representations that are acceptable with preference value assigned to each
- Example

Accept: text/html, text/plain; q=0.5, text/x-dvi; q=0.8

Items For Negotiation

```
Accept-Encoding:
Accept-Charset:
Accept-Language:
```

Conditional Request

- Allows browser to check cached copy for freshness
- Eliminates useless latency
- Sends *If-Modified-Since* in header of GET request
- Example

  If-Modified-Since: Wed, 31 Dec 2003 05:00:01 GMT

Proxy Servers

- Browser can be configured to contact proxy
- Permits caching for entire organization
- Server can specify maximum number of proxies along path (including none)

Caching Of Web Pages

- Caching essential to efficiency
- Server specifies
  - Whether page can be cached
  - Maximum time page can be kept
- Intermediate caches and browser cache web pages
- Browser can specify maximum age of page (forces intermediate caches to revalidate)

Summary

- Web is major application in the Internet
- Standard for representation is HTML
- Standard for transfer is HTTP
  - Request-response protocol
  - Header precedes item
  - Version 1.1 permits persistent connections
  - Server specifies length of time item can be cached
  - Browser can issue conditional request to validate cached item


Questions?

PART XXVIII

APPLICATIONS:
VOICE AND VIDEO OVER IP
(VOIP, RTP, RSVP)

TCP/IP Protocols

- Designed for data
- Can also handle voice and video
- Industry excited about *Voice Over IP* (*VOIP*)

Representation

- Voice and video must be converted between analog and digital forms
- Typical device is *codec* (*coder / decoder*)
- Example encoding used by phone system is *Pulse Code Modulation* (*PCM*)
  - Note: 128 second audio clip encoded in PCM requires one megabyte of memory
- Codec for voice, known as *vocodec*, attempts to recognize speech rather than just waveforms

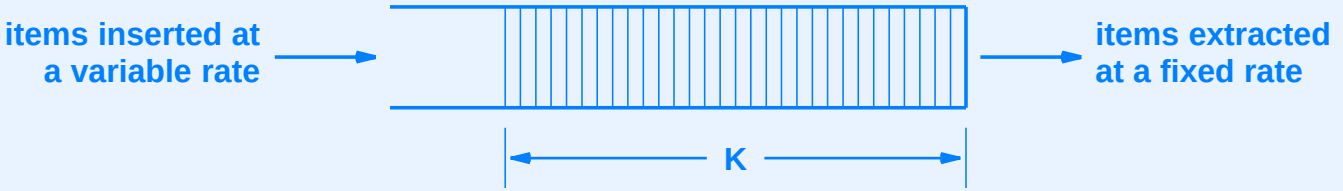
Playback

- Internet introduces burstiness
- Jitter buffer used to smooth bursts
- Protocol support needed

Requirements For Real-Time

Because an IP Internet is not isochronous, additional protocol support is required when sending digitized real-time data. In addition to basic sequence information that allows detection of duplicate or reordered packets, each packet must carry a separate timestamp that tells the receiver the exact time at which the data in the packet should be played.

Illustration Of Jitter Buffer



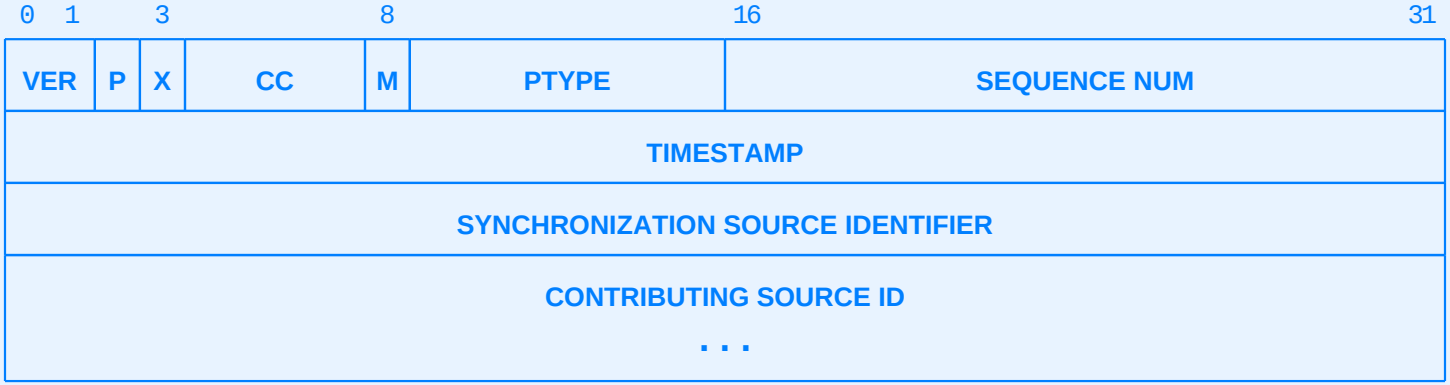
- Data arrives in bursts
- Data leaves at steady rate

Real-Time Transport Protocol (RTP)

- Internet standard
- Provides playback *timestamp* along with data
- Allows receiver to playback items in sequence

RTP Message Format

- Each message begins with same header



Terminology And Layering

- Name implies that RTP is a transport-layer protocol
- In fact
  - RTP is an application protocol
  - RTP runs over UDP

Mixing

- RTP can coordinate multiple data streams
- Intended for combined audio and video
- Up to 15 sources
- Header specifies mixing

RTP Control Protocol (RTCP)

- Required part of RTP
- Allows sender and receiver to exchange information about sessions that are in progress
- Separate data stream
- Uses protocol port number one greater than port number of data stream

RTCP Message Types

Type	Meaning
200	Sender report
201	Receiver report
202	Source description message
203	Bye message
204	Application specific message

RTCP Interaction

- Receivers generate *receiver report* messages
- Inform sender about reception and loss
- Senders generate *sender report*
- Provide absolute timestamp and relate real time to relative playback timestamp

VOIP

- RTP used for encoding and transfer
- Also need signaling protocol for
  - Dialing
  - Answering a call
  - Call forwarding
- Gateway used to connect IP telephone network to *Public Switched Telephone Network* (*PSTN*)
- PSTN uses SS7 for signaling

Standards For IP Telephony

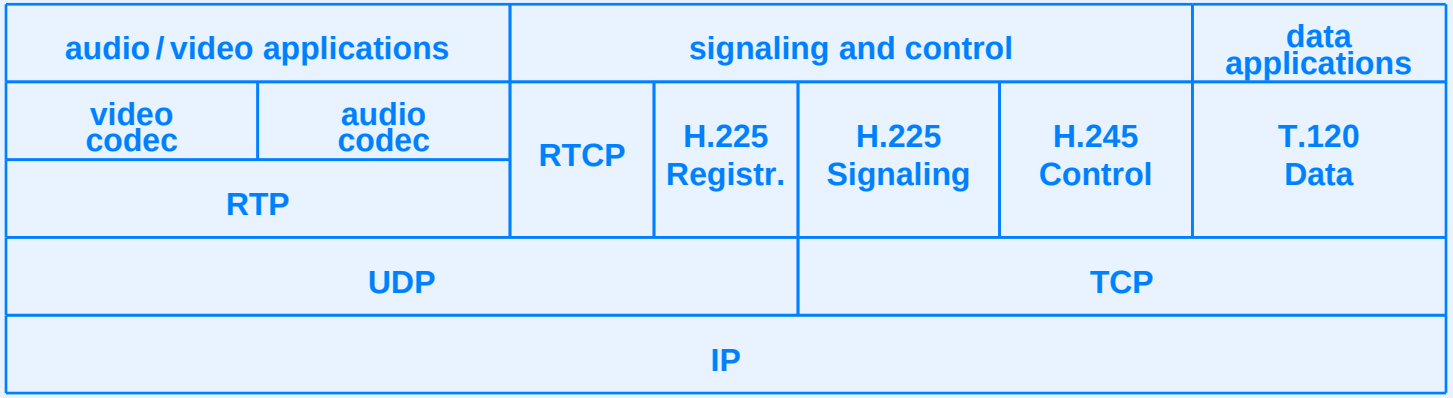
- H.323
- SIP

H.323

- ITU standard
- Set of many protocols
- Major protocols specified by H.323 include

Protocol	Purpose
H.225.0	Signaling used to establish a call
H.245	Control and feedback during the call
RTP	Real-time data transfer (sequence and timing)
T.120	Exchange of data associated with a call

How H.323 Protocols Fit Together



Session Initiation Protocol (SIP)

- IETF standard
- Alternative to H.323
  - Less functionality
  - Much smaller
- Permits SIP telephone to make call
- Does not require RTP for encoding

Session Description Protocol (SDP)

- Companion to SIP
- Specifies details such as
  - Media encoding
  - Protocol port numbers
  - Multicast addresses

Quality Of Service (QoS)

- Statistical guarantee of performance
- Requires changes to underlying Internet infrastructure
- Proponents claim it is needed for telephony
- Others claim only larger bandwidth will solve the problem

Resource ReSerVation Protocol (RSVP)

- IETF response to ATM
- End-to-end QoS guarantees
- Abstraction is unidirectional flow
- Initiated by endpoint

RSVP Requests

An endpoint uses RSVP to request a simplex flow through an IP internet with specified QoS bounds. If routers along the path agree to honor the request, they approve it; otherwise, they deny it. If an application needs QoS in two directions, each endpoint must use RSVP to request a separate flow.

Note About RSVP

- RSVP defines
  - Messages endpoint sends to router to request QoS
  - Messages routers send to other routers
  - Replies
- RSVP does not specify how enforcement done
- Separate protocol needed

Common Open Policy Services (COPS)

- Proposed enforcement protocol for RSVP
- Known as *traffic policing*
- Uses policy server
- Checks data sent on flow to ensure the flow does not exceed preestablished bounds

Summary

- Codec translates between analog and digital forms
- RTP used to transfer real-time data
- RTP adds timestamp that sender uses to determine playback time
- RTCP is companion protocol for RTP that senders and receivers use to control and coordinate data transfer
- Voice Over IP uses
  - RTP for digitized voice transfer
  - SIP or H.323 for signaling
- RSVP and COPS provide quality of service guarantees


Questions?

PART XXIX

APPLICATIONS: INTERNET MANAGEMENT (SNMP)

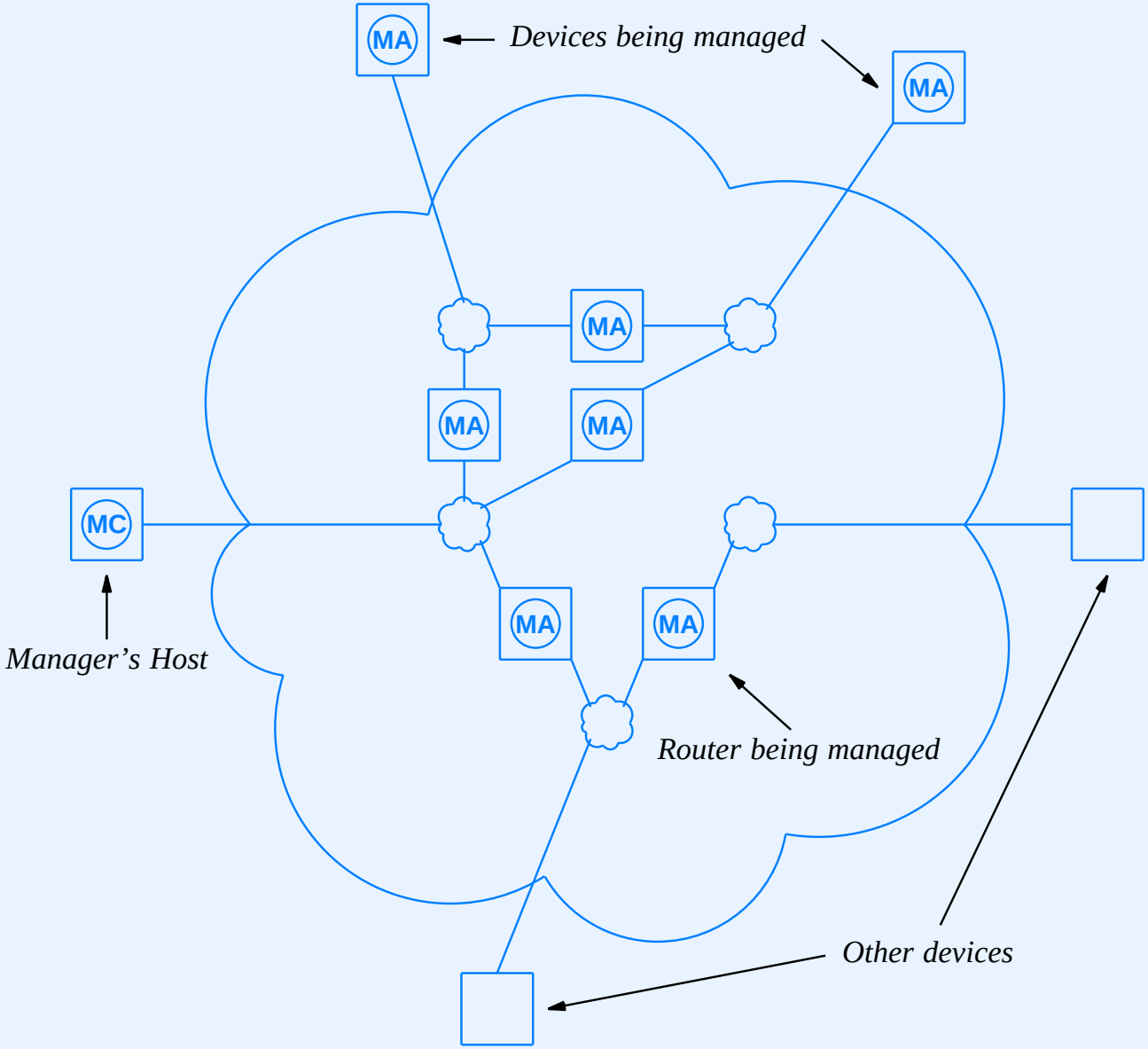
Management Protocols

- Early network systems used two approaches
  - Separate, parallel management network
  - Link-level management commands
- TCP/IP pioneered running management protocols at the application layer
  - Motivation: provide internet-wide capability instead of single network capability

The Point About Internet Management

In a TCP/IP internet, a manager needs to examine and control routers and other network devices. Because such devices attach to arbitrary networks, protocols for internet management operate at the application level and communicate using TCP/IP transport-level protocols.

Architectural Model



Terminology

- Agent
  - Runs on arbitrary system (e.g., a router)
  - Responds to manager's requests
- Management software
  - Runs on manager's workstation
  - Sends requests to agents as directed by the manager

TCP/IP Network Management Protocols

- *Management Information Base* (*MIB*)
- *Structure Of Management Information* (*SMI*)
- *Simple Network Management Protocol* (*SNMP*)

Management Information Base (MIB)

- All management commands are encoded as fetch or store operations on ‘‘variables’’
- Example: to reboot, store a zero in a variable that corresponds to the time until reboot.
- A MIB is a set of variables and the semantics of fetch and store on each

MIB Categories

MIB category	Includes Information About
system	The host or router operating system
interfaces	Individual network interfaces
at	Address translation (e.g., ARP mappings)
ip	Internet Protocol software
icmp	Internet Control Message Protocol software
tcp	Transmission Control Protocol software
udp	User Datagram Protocol software
ospf	Open Shortest Path First software
bgp	Border Gateway Protocol software
rmon	Remote network monitoring
rip-2	Routing Information Protocol software
dns	Domain Name System software

Examples of MIB Variables

MIB Variable	Category	Meaning
sysUpTime	system	Time since last reboot
ifNumber	interfaces	Number of network interfaces
ifMtu	interfaces	MTU for a particular interface
ipDefaultTTL	ip	Value IP uses in time-to-live field
ipInReceives	ip	Number of datagrams received
ipForwDatagrams	ip	Number of datagrams forwarded
ipOutNoRoutes	ip	Number of routing failures
ipReasmOKs	ip	Number of datagrams reassembled
ipFragOKs	ip	Number of datagrams fragmented
ipRoutingTable	ip	IP Routing table
icmpInEchos	icmp	Number of ICMP Echo Requests received
tcpRtoMin	tcp	Minimum retransmission time TCP allows
tcpMaxConn	tcp	Maximum TCP connections allowed
tcpInSegs	tcp	Number of segments TCP has received
udpInDatagrams	udp	Number of UDP datagrams received

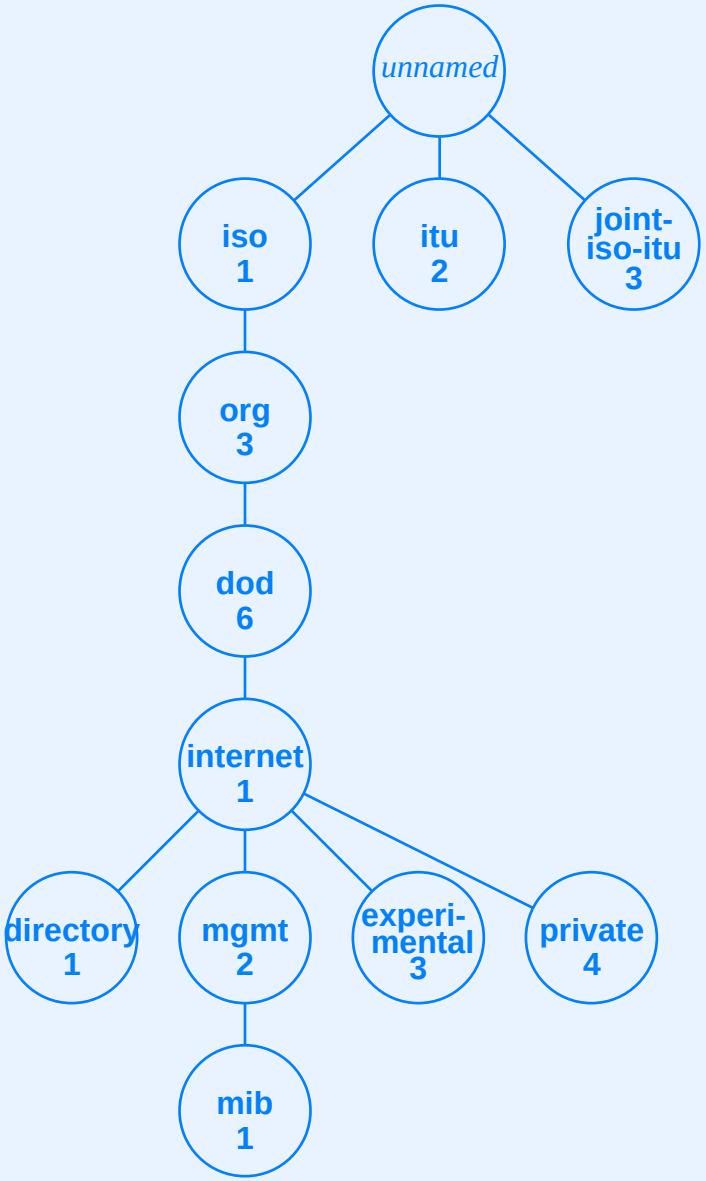
Structure of Management Information (SMI)

- Set of rules for defining MIB variable names
- Includes basic definitions such as
  - Address (4-octet value)
  - Counter (integer from 0 to $2^{32} - 1$)
- Specifies using *Abstract Syntax Notation 1* (*ASN.1*)

ASN.1

- ISO standard
- Specifies
  - Syntax for names (user-readable format)
  - Binary encoding (format used in a message)
- Absolute, global, hierarchical namespace

Position of MIB In The ASN.1 Hierarchy

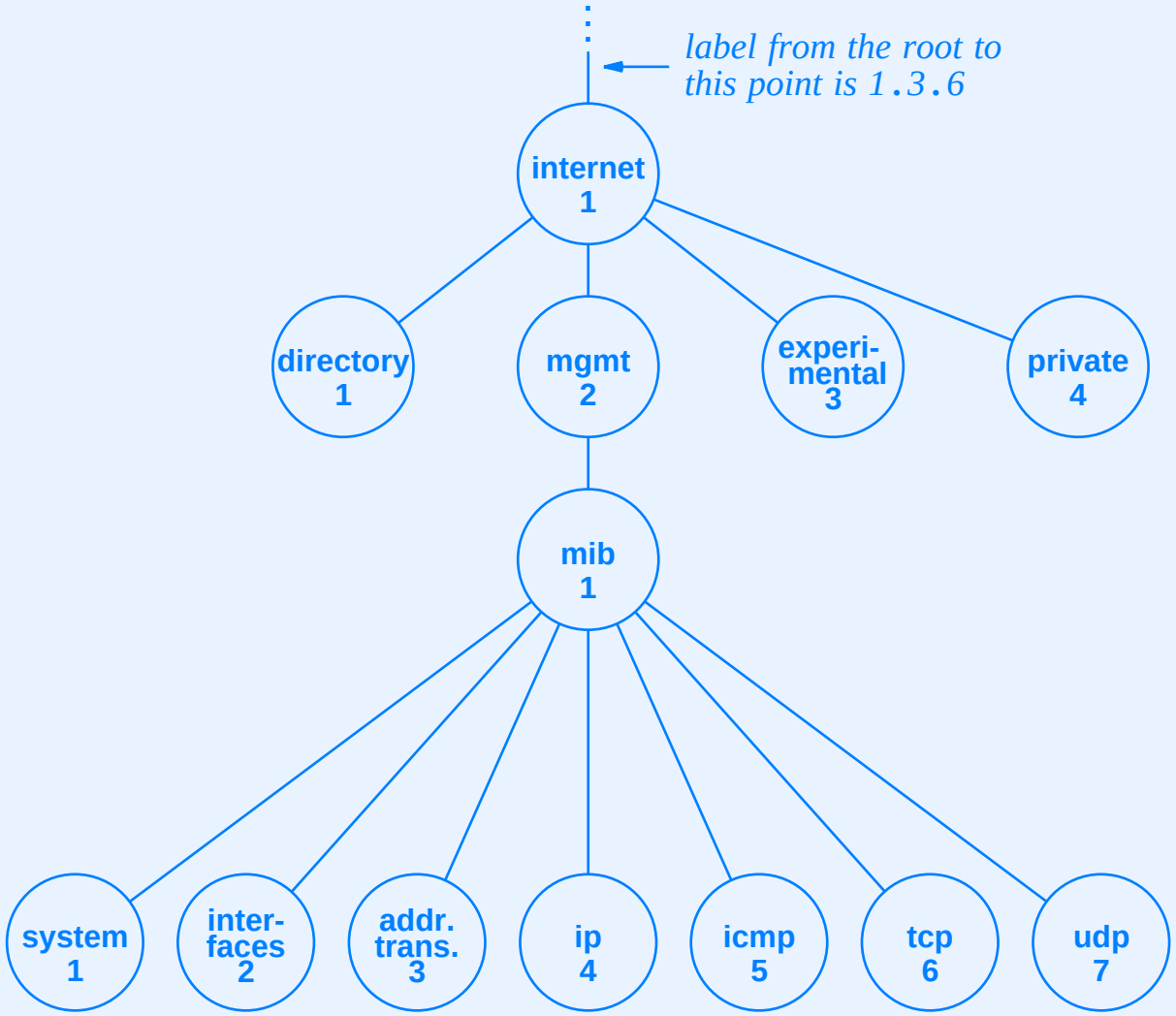


Syntactic Form

- Variable name written as sequence of labels with dot (period as delimiter)
- Numeric encoding used in messages
- Example: prefix for mgmt node is

1.3.6.1.2.1

ASN.1 Hierarchy For TCP/IP



Example MIB Variables

- Prefix for variable *ipInReceives* is

  iso . org . dod . internet . mgmt . mib . ip . ipInReceives

- Numeric value is

  1 . 3 . 6 . 1 . 2 . 1 . 4 . 3

MIB Tables

- Correspond to data structures programmers think of as arrays or structs
- ASN.1 definition uses keyword *SEQUENCE*
- Array index is appended to MIB variable name

Example Of SEQUENCE Definition

```
IpAddrEntry  ::=  SEQUENCE  {
    ipAdEntAddr
            IpAddress,
    ipAdEntIfIndex
            INTEGER,
    ipAdEntNetMask
            IpAddress,
    ipAdEntBcastAddr
            IpAddress,
    ipAdEntReasmMaxSize
            INTEGER (0..65535)
}
```

Simple Network Management Protocol (SNMP)

- Specifies communication between manager's workstation and managed entity
- Uses fetch-store paradigm

Operations That SNMP Supports

Command	Meaning
get-request	Fetch a value from a specific variable
get-next-request	Fetch a value without knowing its exact name
get-bulk-request	Fetch a large volume of data (e.g., a table)
response	A response to any of the above requests
set-request	Store a value in a specific variable
inform-request	Reference to third-part data (e.g., for a proxy)
snmpv2-trap	Reply triggered by an event
report	Undefined at present

SNMP Message Format

- Defined using ASN.1 notation
- Similar to BNF grammar

Example ASN.1 Definition

```
SNMPv3Message  ::=
    SEQUENCE  {
        msgVersion  INTEGER (0..2147483647),
          -- note: version number 3 is used for SNMPv3
        msgGlobalData  HeaderData,
        msgSecurityParameters  OCTET STRING,
        msgData  ScopedPduData
    }
```

Definition Of HeaderData Area In SNMP Message

```
HeaderData ::= SEQUENCE {
    msgID  INTEGER (0..2147483647),
      --  used to match responses with requests
    msgMaxSize  INTEGER (484..2147483647),
      --  maximum size reply the sender can accept
    msgFlags  OCTET STRING (SIZE(1)),
      --  Individual flag bits specify message characteristics
      --  bit 7 authorization used
      --  bit 6 privacy used
      --  bit 5 reportability (i.e., a response needed)
    msgSecurityModel  INTEGER (1..2147483647)
      -- determines exact format of security parameters that follow
}
```

Discriminated Union

- ASN.1 uses *CHOICE* keyword for a discriminated union
- Example

```
ScopedPduData ::= CHOICE {
    plaintext  ScopedPDU,
    encryptedPDU  OCTET STRING  -- encrypted ScopedPDU value
}
```

Summary

- TCP/IP management protocols reside at application layer
- Management Information Base (MIB) specifies set of variables that can be accessed
- Structure Of Management Information (SMI) specifies rules for naming MIB variables
- Simple Network Management Protocol (SNMP) specifies format of messages that pass between a manager's workstation and managed entity
- Variables named using ASN.1 (absolute, global, hierarchical)
- Message format defined with ASN.1 (similar to BNF grammar)


Questions?

PART XXX

INTERNET SECURITY AND FIREWALL DESIGN (IPsec, SSL)

Network Security

- Refers in broad sense to confidence that information and services available on a network cannot be accessed by unauthorized users
- Implies
  - Safety
  - Freedom from unauthorized access or use
  - Freedom from snooping or wiretapping
  - Freedom from disruption of service
  - Assurance that outsiders cannot change data
- Also called *information security*

A Crucial Point

Just as no physical property is absolutely secure against crime, no network is completely secure.

Aspects Of Protection

- Data integrity
- Data availability
- Privacy or confidentiality
- Authorization
- Authentication
- Replay avoidance

Information Policy

- Defines what is allowed
- Special note:

Humans are usually the most susceptible point in any security scheme. A worker who is malicious, careless, or unaware of an organization's information policy can compromise the best security.

Internet Security

- Especially difficult
- Data travels across many networks owned by many groups from source to destination
- Computers in the middle can change data

A Point About Authentication

An authorization scheme that uses a remote machine's IP address to authenticate its identity does not suffice in an unsecure internet. An imposter who gains control of an intermediate router can obtain access by impersonating an authorized client.

Two Basic Techniques For Internet Security

- Encryption
- Perimeter Security

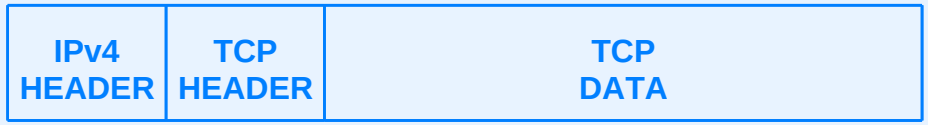
IP Security Protocol (IPsec)

- Devised by IETF
- Actually a set of protocols
- Name *IPsec* applies collectively
- Works with IPv4 or IPv6
- Gives framework, but does not specify exactly which encryption or authentication algorithms to use
- Choice between authentication and encryption

IPsec Authentication Header (AH)

- Not an IP option
- Added after IP header
- Follows IPv6 format (more on IPv6 later in the course)

Illustration of Authentication Header Insertion



(a)

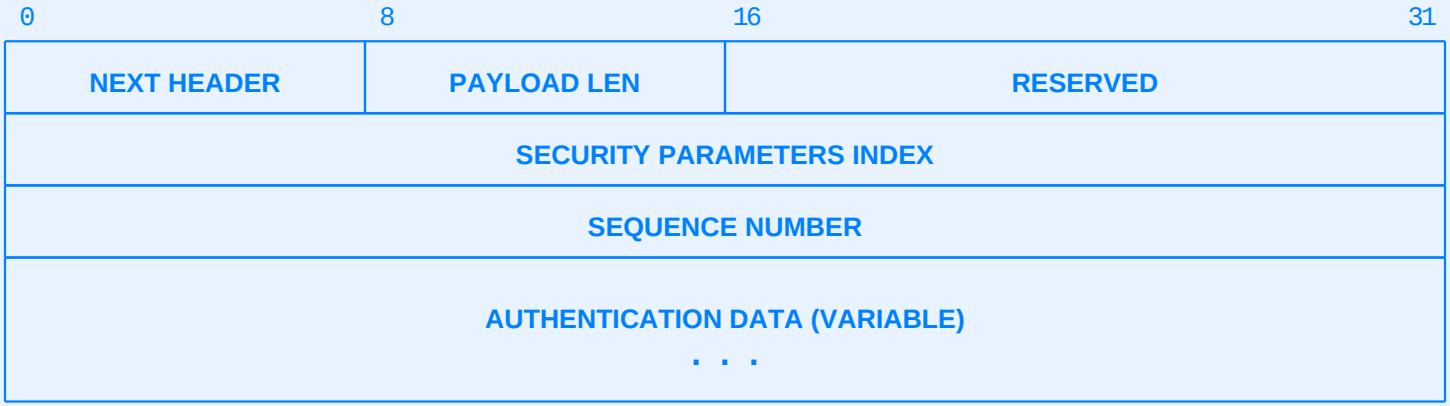
(b)

- (a) shows datagram and (b) shows same datagram after header has been inserted

Type Information

- IPv4 *PROTOCOL* field is modified so the type is IPsec
- Authentication header contains *NEXT HEADER* field that specifies original type

Illustration Of Type Information With Authentication



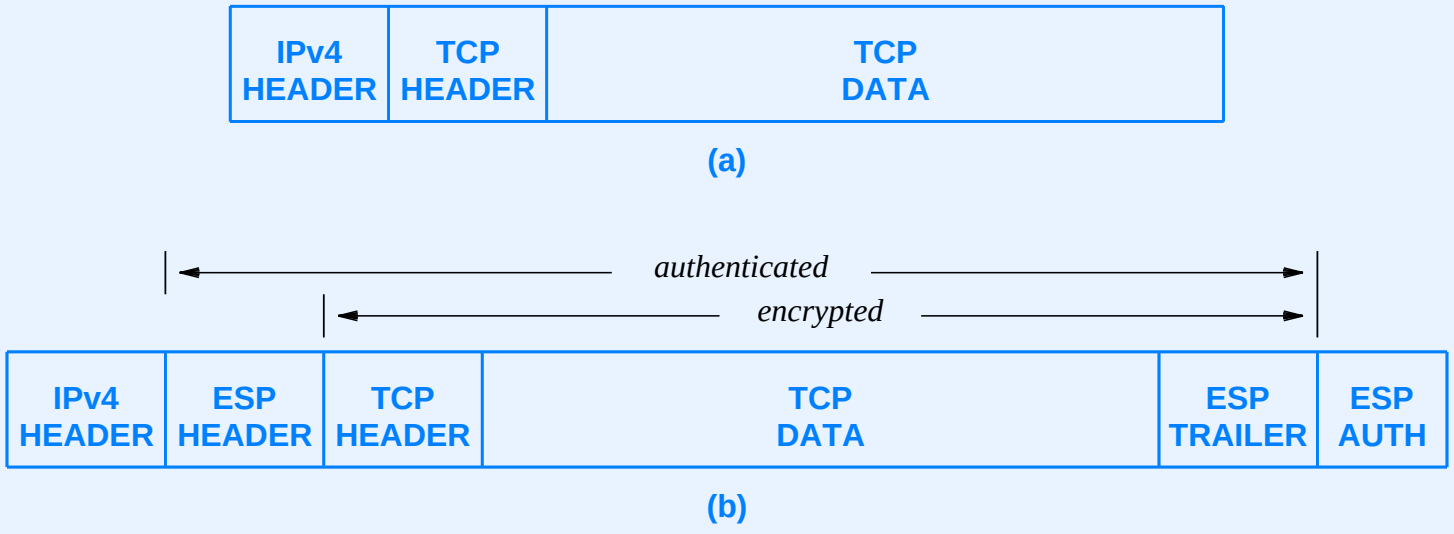
Security Association

- Not all information related to security can fit in header
- Sender and receiver communicate, agree on security parameters, assign small index to each parameter, and then use index values in headers

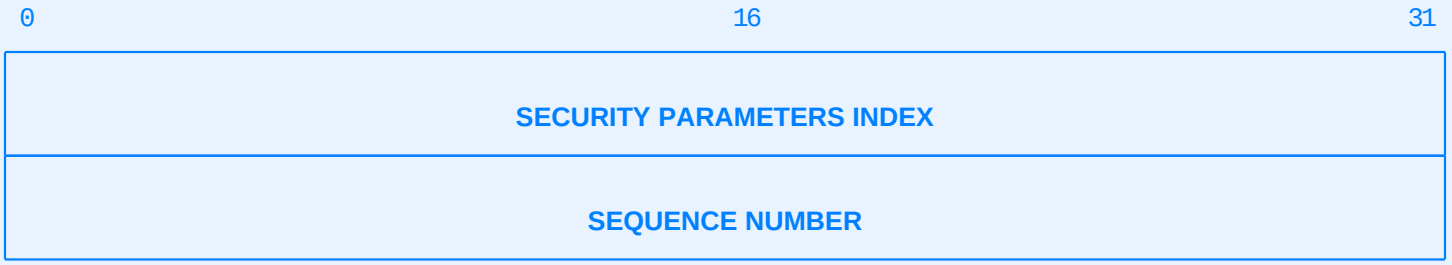
IPsec Encapsulating Security Payload (ESP)

- Used to encrypt packet contents
- More complex than authentication header

Illustration Of ESP

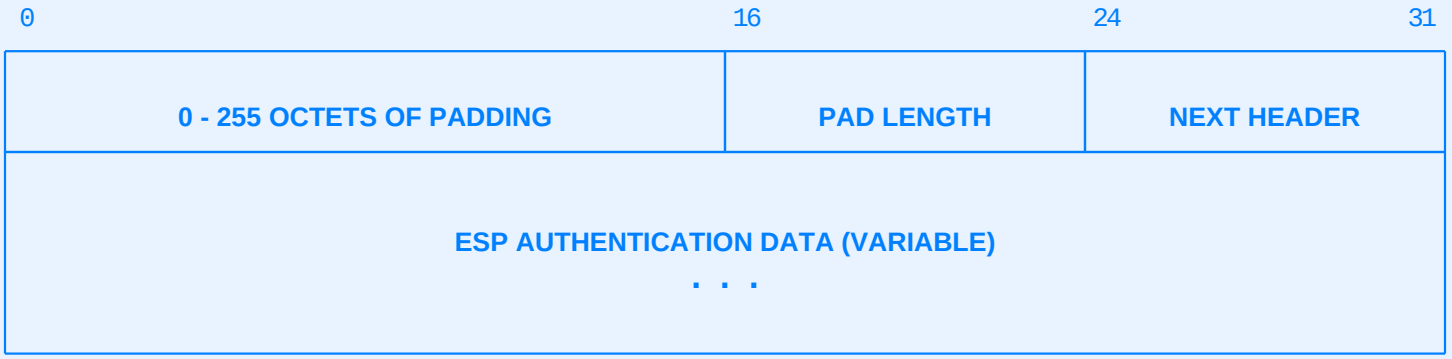


ESP Header



- Eight octets
- Precedes payload

ESP Trailer

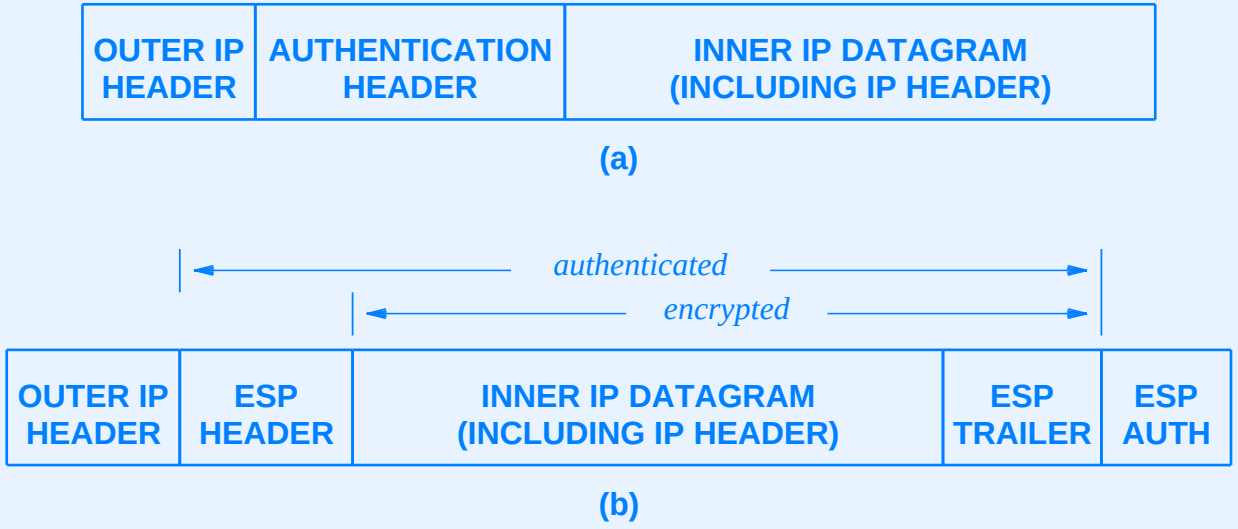


- Authentication data variable size
- Padding optional

Mutable Header Fields

- Some IP header fields change (e.g., TTL)
- IPsec designed to ensure end-to-end integrity
- One possibility: IPsec tunneling
  - Place IPsec datagram inside normal datagram
  - Often used in VPNs

Illustration Of IPsec Tunneling



- (a) when used with authentication
- (b) when used with encapsulated security payload

Mandatory Security Algorithms For IPsec

Authentication

HMAC with MD5	RFC 2403
HMAC with SHA-1	RFC 2404

Encapsulating Security Payload

DES in CBC mode	RFC 2405
HMAC with MD5	RFC 2403
HMAC with SHA-1	RFC 2404
Null Authentication	
Null Encryption	

Secure Sockets Layer (SS)

- Created by Netscape, Inc.
- Widely used
- Not formally adopted by IETF
- Same API as sockets
- Provides authentication and encryption
- De facto standard for web browsers

Transport Layer Security (TLS)

- Created by IETF
- So closely related to SSL that the same protocol port is used
- Most implementations of SSL also support TLS

Perimeter Security

- Form of access control
- Mechanism is *Internet firewall*
- Firewall placed at each connection between site and rest of Internet
- All firewalls use coordinated policy
- Blocks unwanted packets

Firewall Implementation

- Basic technique is *packet filter*
- Typically runs in a router
- Manager specifies restrictions on incoming packets
- Filter drops packets that are not allowed

Illustration Of Packet Filter

ARRIVES ON INTERFACE	IP SOURCE	IP DEST.	PROTOCOL	SOURCE PORT	DEST. PORT
2	*	*	TCP	*	21
2	*	*	TCP	*	23
1	128.5.0.0 / 16	*	TCP	*	25
2	*	*	UDP	*	43
2	*	*	UDP	*	69
2	*	*	TCP	*	79

Effective Filtering

To be effective, a firewall that uses datagram filtering should restrict access to all IP sources, IP destinations, protocols, and protocol ports except those computers, networks, and services the organization explicitly decides to make available externally. A packet filter that allows a manager to specify which datagrams to admit instead of which datagrams to block can make such restrictions easy to specify.

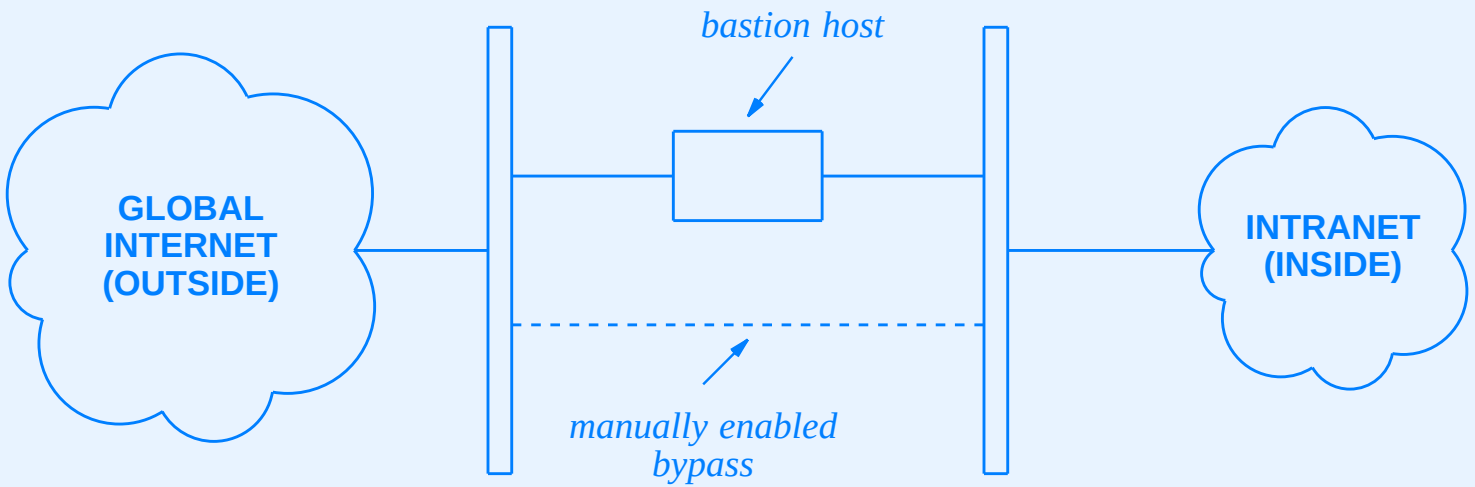
Consequences Of A Restrictive Filter

If an organization's firewall restricts incoming datagrams except for ports that correspond to services the organization makes available externally, an arbitrary application inside the organization cannot become a client of a server outside the organization.

Proxy Access

- Allows specific clients to access specific services
- Handles problems like virus detection on incoming files
- Uses *bastion host*

Illustration Of Proxy Access



- Two firewall filters restrict
  - Incoming packets from Internet to proxy
  - Outgoing packets from site to proxy

Stateful Firewalls

- Allow clients inside an organization to contact servers in the Internet
- Firewall
  - Watches outgoing packets
  - Records source and destination information
  - Uses recorded information when admitting packets
- Communication still subject to policies

Managing Firewall State

- Connection tracking
  - Uses FIN to remove state for TCP connection
  - Does not work well with UDP
- Soft state
  - Timer set when entry created
  - Idle entry removed after timeout

Content Protection With Proxies

- Firewall only operates at packet level
- Mechanism known as *application proxy* protects against incoming
  - Viruses
  - Other illicit content
- Proxy can examine entire content (e.g., mail message)

Summary

- Two basic techniques used for Internet security
  - Encryption
  - Perimeter security
- IETF has defined IPsec as a framework for security
- IPsec offers choice of
  - Authentication header (AH)
  - Encapsulated Security Payload (ESP)

Summary
(continued)

- Firewall is mechanism used for perimeter security
- Packet filter specified by manager
- Firewall rejects packets except those explicitly allowed
- Stateful firewall allows clients in organization to initiate communication
- Application proxy can be used to check content


Questions?

PART XXXI

THE FUTURE OF TCP/IP (IPv6)

Current Version

- TCP/IP has worked well for over 25 years
- Design is flexible and powerful
- Has adapted to
  - New computer and communication technologies
  - New applications
  - Increases in size and load

Most Significant Technical Problem

- Address space limitation
- IPv4 address space may be exhausted by the year 2020

History Of The New Version

- Developed by IETF
- Started in early 1990s
- Input from many groups, including: computer manufacturers, hardware and software vendors, users, managers, programmers, telephone companies, and the cable television industry

History Of The New Version
(continued)

- Three main proposals
- Eventually new version emerged
- Assigned version number 6, and known as *IPv6*
- RFC in 1994
- Defined over 10 years ago!

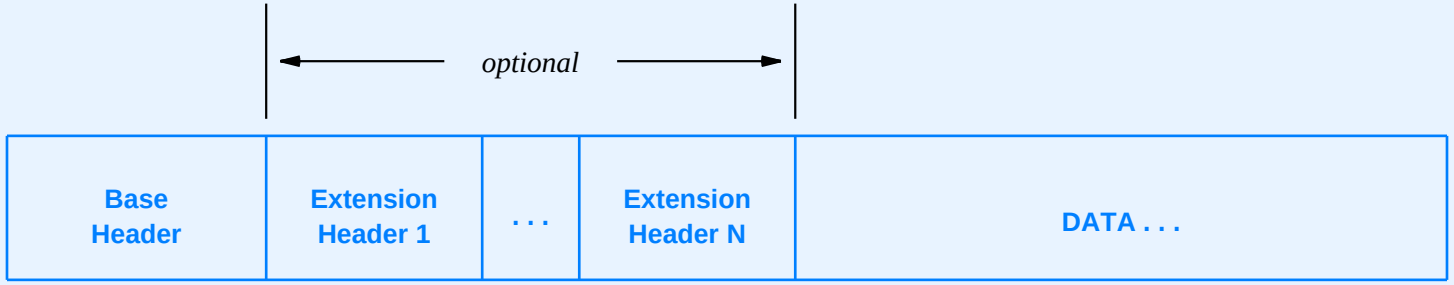
Major Changes From IPv4

- Larger addresses
- Extended address hierarchy
- Variable header format
- Facilities for many options
- Provision for protocol extension
- Support for autoconfiguration and renumbering
- Support for resource allocation

IPv6 Address Size

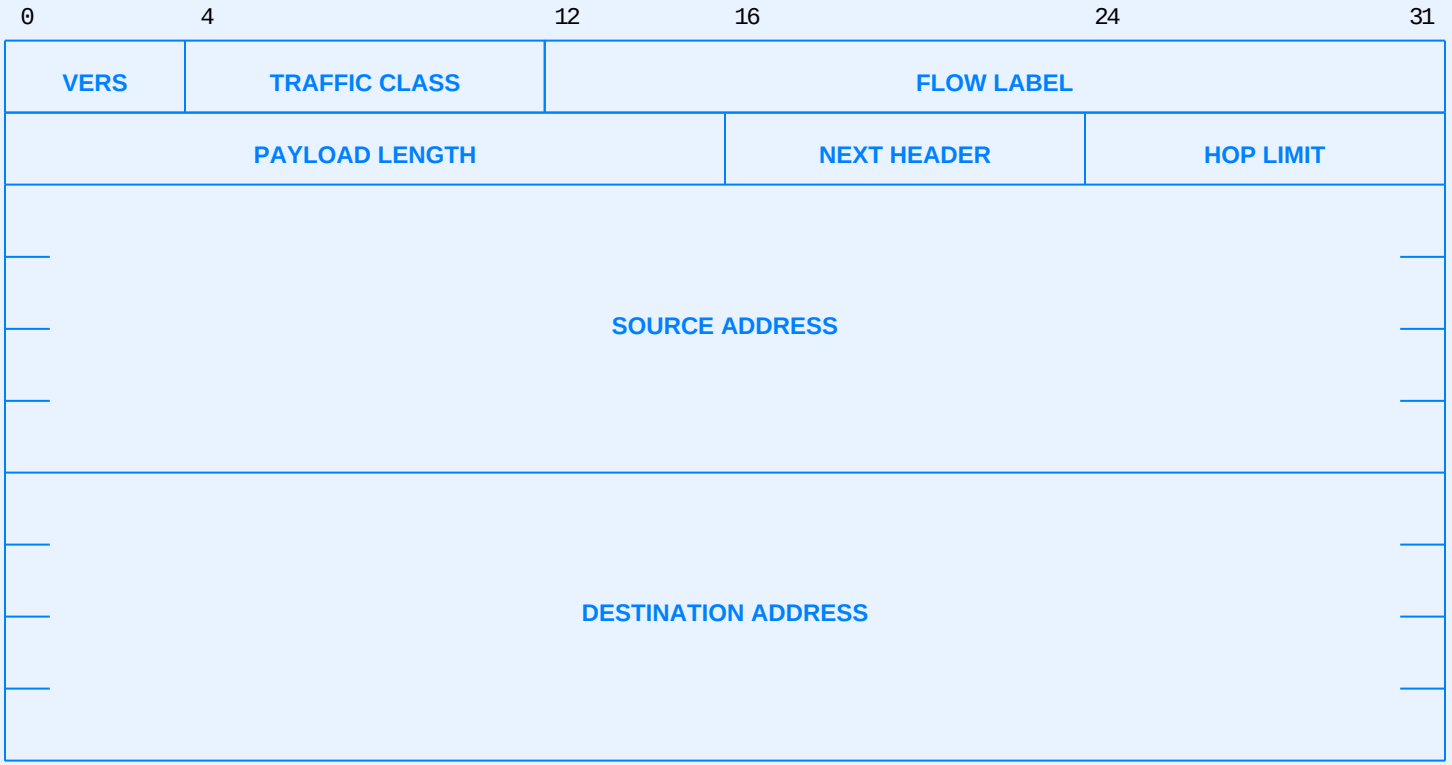
- 128 bits per address
- Absurd increase in capacity
- IPv6 has 10^{24} addresses per square meter of the Earth's surface!

General Form Of IPv6 Datagram



- Base header required
- Extension headers optional

IPv6 Base Header Format



- Alignment is on 64-bit multiples
- Fragmentation in extension header
- Flow label intended for resource reservation

Size Of Base Header

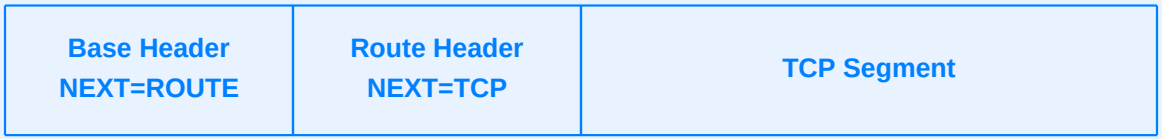
Each IPv6 datagram begins with a 40-octet base header that includes fields for the source and destination addresses, the maximum hop limit, the traffic class, the flow label, and the type of the next header. Thus, an IPv6 datagram must contain at least 40 octets in addition to the data.

IPv6 Extension Headers

- Sender chooses zero or more extension headers
- Only those facilities that are needed should be included

Parsing An IPv6 Datagram

(a)



(b)

(c)

- Each header includes *NEXT HEADER* field
- *NEXT HEADER* operates like type field

IPv6 Fragmentation And Reassembly

- Like IPv4
  - Ultimate destination reassembles
- Unlike IPv4
  - Routers avoid fragmentation
  - Original source must fragment

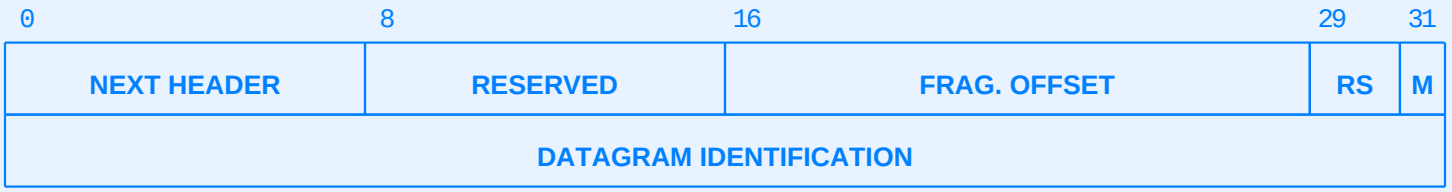
How Can Original Source Fragment?

- Option 1: choose minimum guaranteed MTU of 1280
- Option 2: use path MTU discovery

Path MTU Discovery

- Guessing game
- Source sends datagram without fragmenting
- If router cannot forward, router sends back ICMP error message
- Source tries smaller MTU

Fragmentation Details



- Fragmentation information carried in extension header

Discussion Questions

- Is fragmentation desirable?
- What are the consequences of the IPv6 design?

IPv6 Colon Hexadecimal Notation

- Replaces dotted decimal
- Example: dotted decimal value

104.230.140.100.255.255.255.255.0.0.17.128.150.10.255.255

- Becomes

68E6:8C64:FFFF:FFFF:0:1180:96A:FFFF

Zero Compression

- Successive zeroes are indicated by a pair of colons
- Example

FF05:0:0:0:0:0:0:B3

- Becomes

FF05::B3

IPv6 Destination Addresses

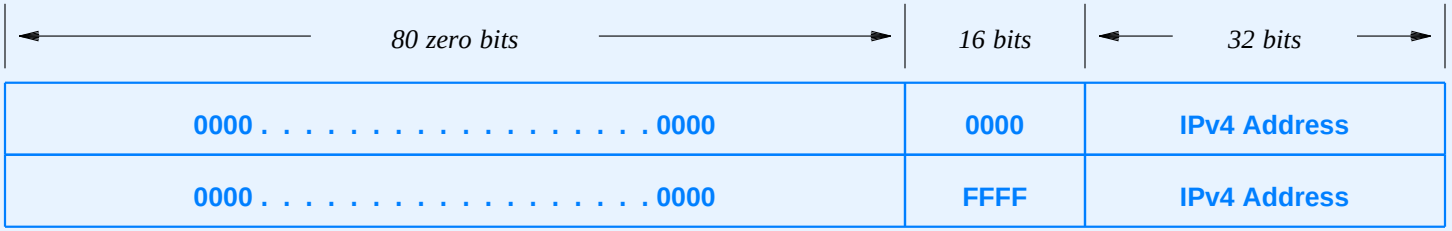
- Three types
  - Unicast (single host receives copy)
  - Multicast (set of hosts each receive a copy)
  - Anycast (set of hosts, one of which receives a copy)
- Note: no broadcast (but special multicast addresses (e.g., ‘‘all hosts on local wire’’)

Proposed IPv6 Address Space

Binary Prefix	Type Of Address	Part Of Address Space
0000 0000	Reserved (IPv4 compatibility)	1/256
0000 0001	Unassigned	1/256
0000 001	NSAP Addresses	1/128
0000 01	Unassigned	1/64
0000 1	Unassigned	1/32
0001	Unassigned	1/16
001	Global Unicast	1/8
010	Unassigned	1/8
011	Unassigned	1/8
100	Unassigned	1/8
101	Unassigned	1/8
110	Unassigned	1/8
1110	Unassigned	1/16
1111 0	Unassigned	1/32
1111 10	Unassigned	1/64
1111 110	Unassigned	1/128
1111 1110 0	Unassigned	1/512
1111 1110 10	Link-Local Unicast Addresses	1/1024
1111 1110 11	IANA - Reserved	1/1024
1111 1111	Multicast Addresses	1/256

Backward Compatibility

- Subset of IPv6 addresses encode IPv4 addresses
- Dotted hex notation can end with 4 octets in dotted decimal



Myths About IPv6
According To Geoff Huston

- IPv6 is
  - More secure
  - Required for mobility
  - Better for wireless networks
- IPv6 offers better QoS
- Only IPv6 supports auto-configuration
- IPv6 solves route scaling

Myths About IPv6 According To Geoff Huston (continued)

- IPv6 provides better support for
  - Rapid prefix renumbering
  - Multihomed sites
- IPv4 has run out of addresses

Source: G. Huston, ‘‘The Mythology Of IP Version 6,’’ *The Internet Protocol Journal* vol. 6:2 (June, 2003)

Summary

- IETF has defined next version of IP to be IPv6
- Addresses are 128 bits long
- Datagram starts with base header followed by zero or more extension headers
- Sender performs fragmentation
- Many myths abound about the advantages of IPv6
- No strong technical motivation for change


Questions?


STOP