MODULE V

Internetworking: Concepts, Addressing, Architecture, Protocols, Datagram Processing, Transport-Layer Protocols, And End-To-End Services

Topics

- Internet concept and architecture
- Internet addressing
- Internet Protocol packets (datagrams)
- Datagram forwarding
- Address resolution
- Error reporting mechanism
- Configuration
- Network address translation

Topics
(continued)

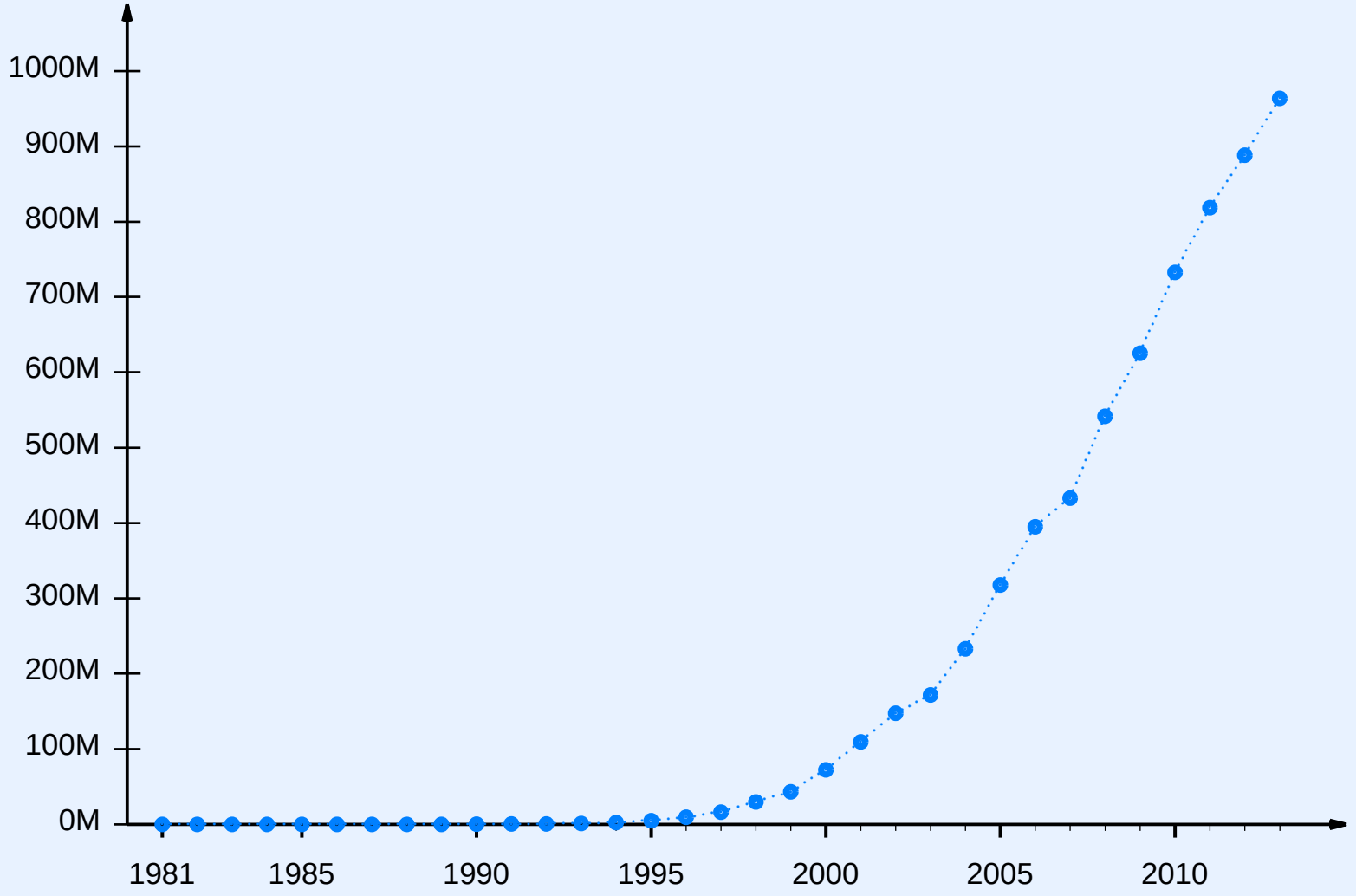
- Transport layer protocol characteristics and techniques
- Message transport with the User Datagram Protocol (UDP)
- Stream transport with the Transmission Control Protocol (TCP)
- Routing algorithms and protocols
- Internet multicast and multicast routing

Internet Concept
And Internet Architecture

What Is The Internet?

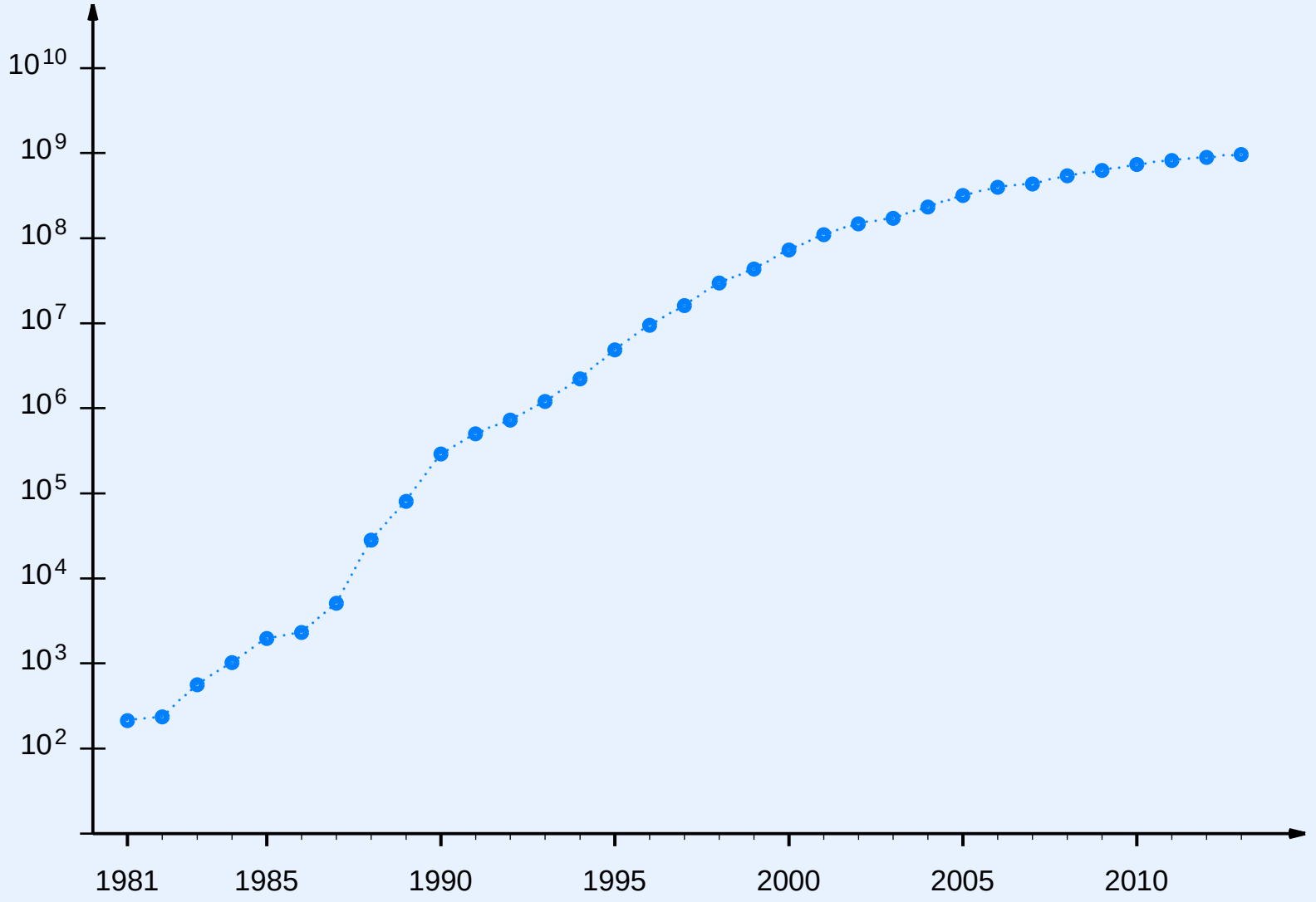
- Users see it as services and applications
  - Web and e-commerce
  - Email, texting, instant messenger
  - Social networking and blogs
  - Music and video download (and upload)
  - Voice and video teleconferencing
- Networking professionals see it as infrastructure
  - Platform on which above services run
  - Grows rapidly

Growth Of The Internet



- Plot shows number of computers on the Internet each year

Growth Of The Internet (log scale)



- Plot shows number of computers on the Internet each year

Actual Size Of The Internet

- Previous plots are somewhat misleading
  - Derived by walking the Domain Name System
  - Only report hosts with IP addresses
- Since around 2000, many Internet devices
  - Do not have a fixed IP address
  - Connect behind a NAT box (e.g., wireless router)
- Actual size is difficult to measure

Internet Architecture And Design

- If one were to design a global communication system from scratch
  - How should it be organized?
  - Which technology or technologies should be used?
- The challenges
  - Which applications should it support?
  - Which network technologies should it use
    * PANs / LANs / MANs / WANs
    * Wired / wireless
    * Terrestrial / satellite

Internet Architecture And Design (continued)

- Key principles
  - Internet is designed to accommodate extant services plus new services that will be invented
  - Internet is designed to accommodate *any* network technology, allowing each technology to be used where appropriate

Internet Philosophy

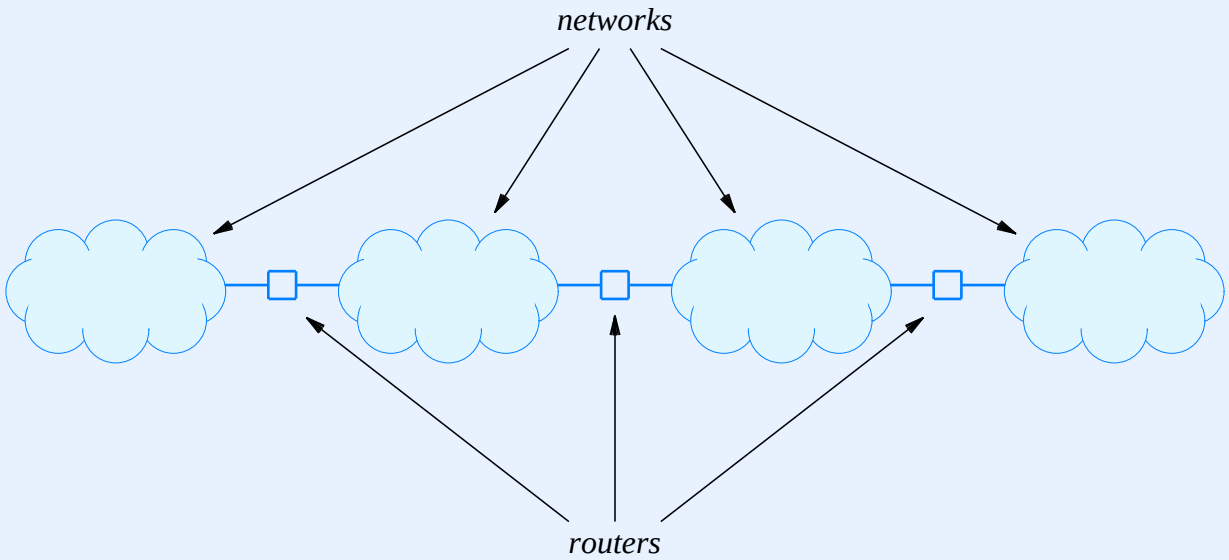
- Infrastructure
  - Provides a packet communication service
  - Treats all attached endpoints as equal (any endpoint can send a packet to any other endpoint)
  - Does not restrict or dictate packet contents
  - Does not restrict or dictate underlying network technologies
- Attached endpoints
  - Run applications that use the network to communicate with applications on other endpoints
  - Control all content and provide all services

Advantages Of The Internet Philosophy

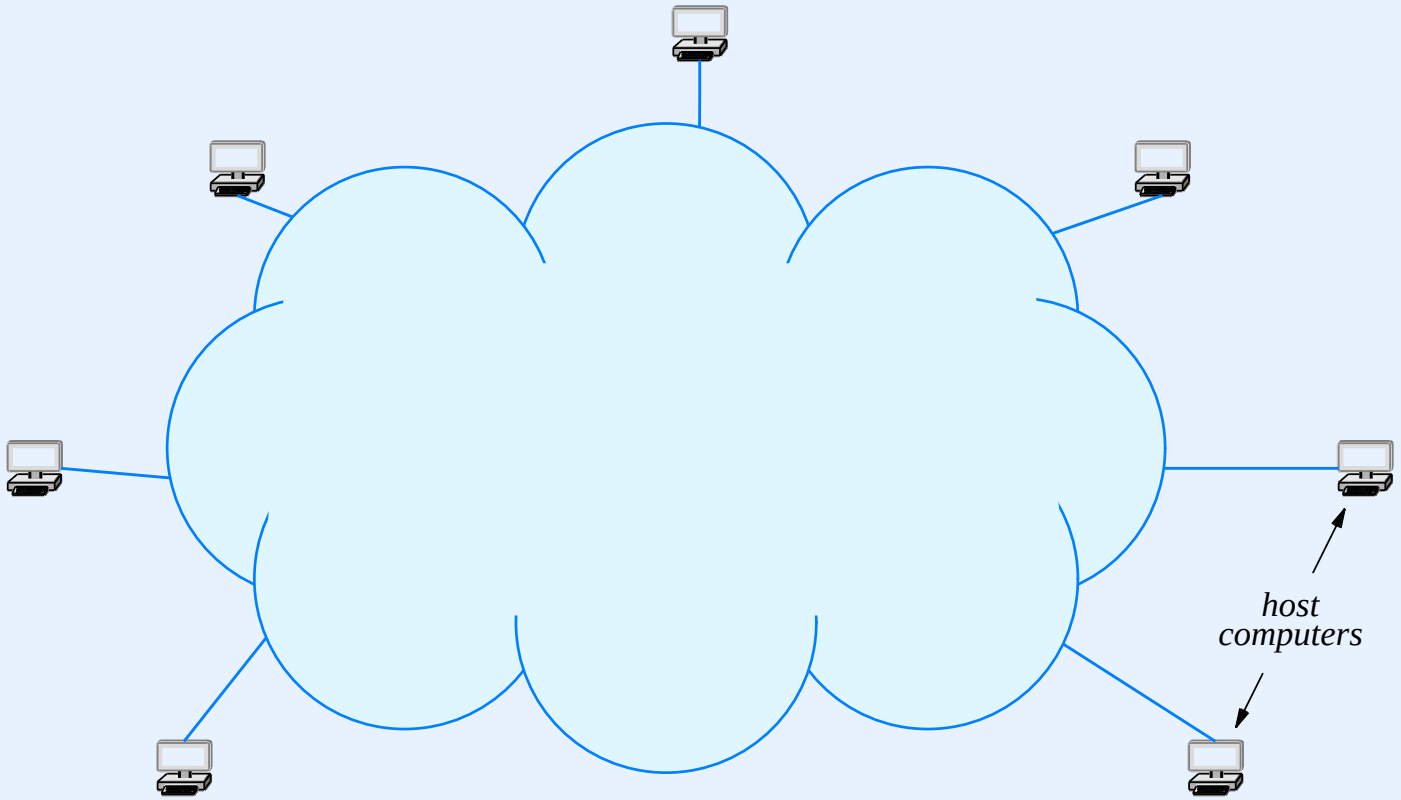
- Accommodates heterogeneous underlying networks
- Accommodates arbitrary applications and services
- Separates communication from services

Internet

- Follows a *network of networks* approach
- Allows arbitrary networks to be included
- Uses *IP routers* to interconnect individual networks
- Permits each router to connect two or more networks

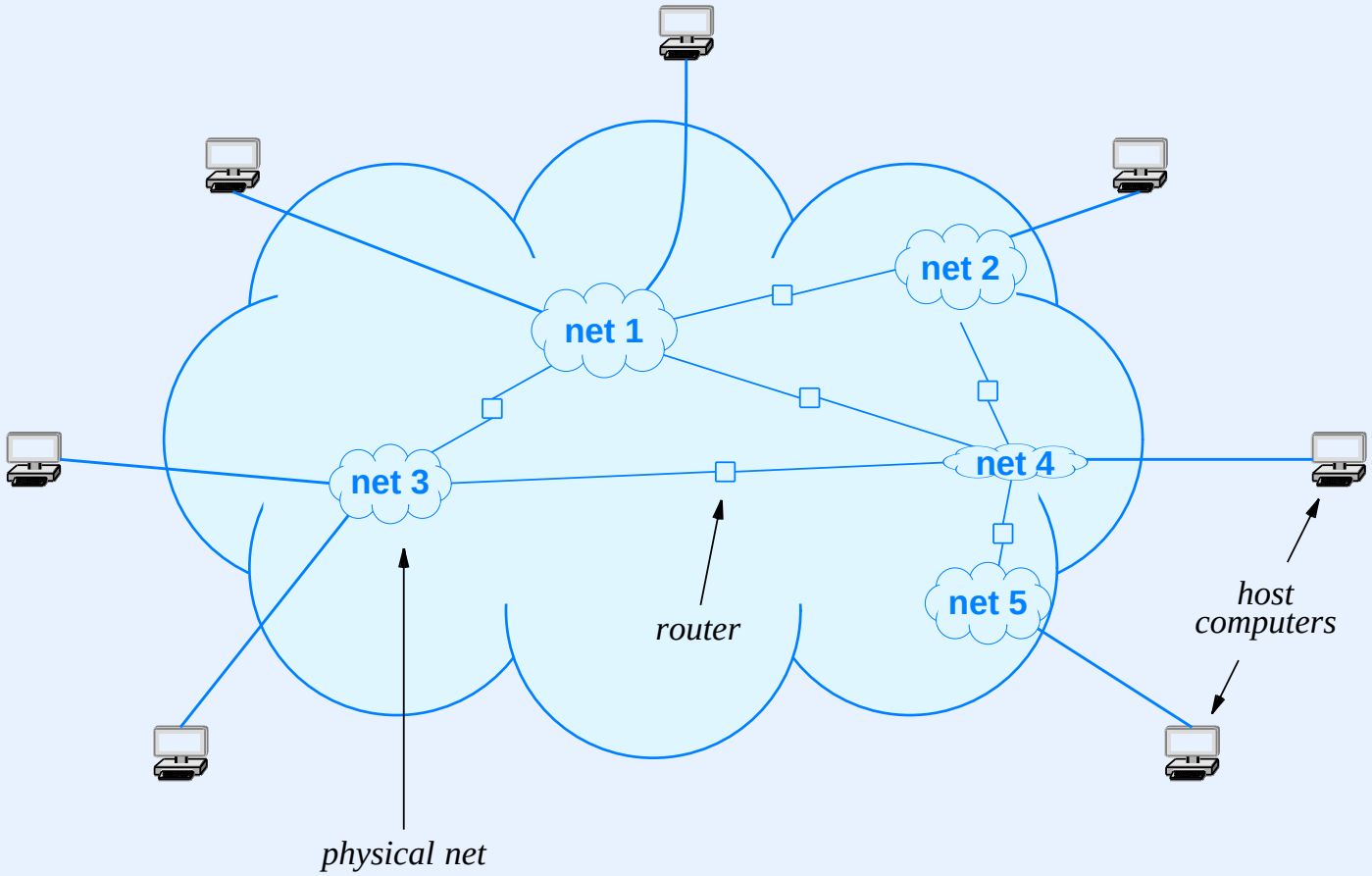


Internet Architecture: Logical View



- Computers attached to Internet known as *host* computers
- To a host, Internet appears to be one giant network

Internet Architecture: Physical View



- Network of heterogeneous networks connected by routers
- Each host attaches to a network

Before We Discuss Internet Addressing

The Situation

- Internet addressing is defined by the *Internet Protocol* (*IP*)
- IP is changing
  - Current version is 4 (*IPv4*)
  - New version is 6 (*IPv6*)

History Of The Internet Protocol

- IP separated from TCP in 1978
- Version 1-3 discarded quickly; version 4 was the first version used by researchers
- By early 1990s, a movement started that clamored for a new version of IP because the 32-bit address space would run out “soon”
- In 1993, the IETF received proposals, and formed a working group to find a compromise
- By 1995, a new version had been proposed and documents written

Background Of The New Version Of IP

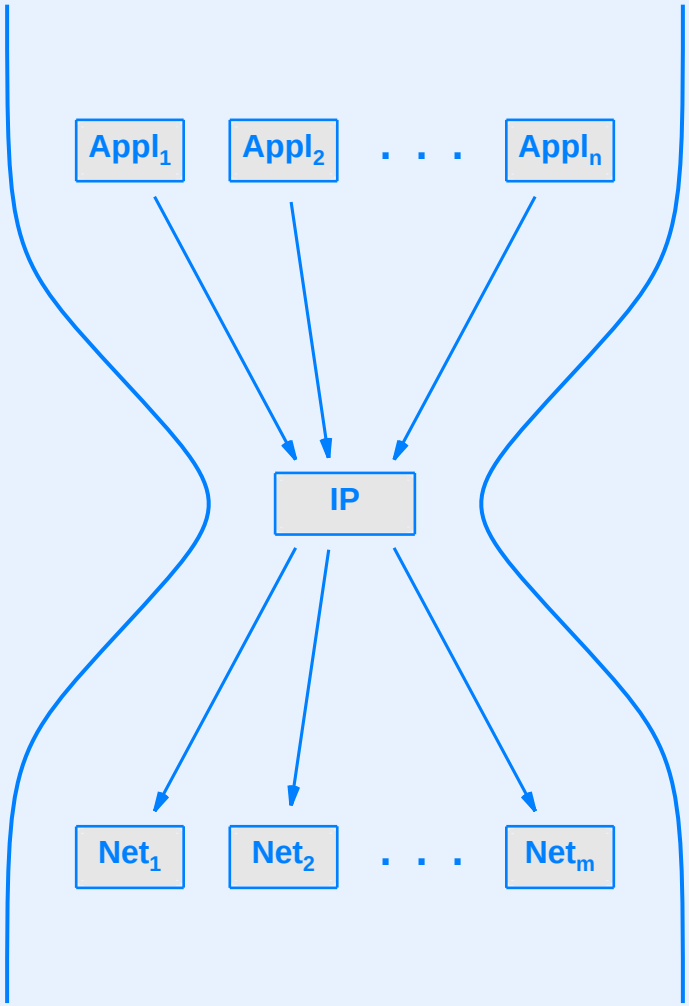
- Various groups offered opinions about the features
  - Cable companies wanted support for broadcast delivery
  - Telephone companies argued that everyone would soon be using a connection-oriented network technology (ATM)
  - Several groups wanted mobility
  - The military pushed for better security
- A compromise was reached: IP version 6 includes all the above

The Uphill Battle To Change IPv4

- IP is difficult to change because
  - IP lies at the heart of the Internet protocols
  - Version 4 of IP has a proven track record

The success of the current version of IP is incredible — the protocol has accommodated changes in hardware technologies, heterogeneous networks, and extremely large scale.

The Hourglass Model



- IP lies in the middle — changing it means changing all hosts and routers in the Internet

Our Approach

- In the current Internet, both IPv4 and IPv6 are relevant and important
- Throughout the course, we will
  - Discuss general concepts
  - See how IPv4 and IPv6 implement the concepts

Internet Addressing

Addressing In The Internet

- Can we use MAC addresses across an internet?
- No: heterogeneity means
  - Multiple *types* of MAC addresses
  - MAC address meaningful on one network not meaningful on another
- Solution
  - Create new addressing scheme that is independent of MAC addresses

The Two Forms Of Addresses

- Identity
  - Unique number assigned to each endpoint
  - Analogous to Ethernet address
- Locator
  - Endpoint address encodes location information, such as
    - Geographic location
    - Location relative to a service provider
    - Computer on a given physical network

Two Principles To Keep In Mind

Both identify and locator forms have advantages in some situations; no form is best in all cases

Addressing is inherently linked to routing; the choice of an addressing scheme affects the cost of computing and maintaining routes

The IPv4 Addressing Scheme

- Unique number is assigned to each Internet host
- 32-bit binary value known as *IPv4 address*
- Virtual address, not derived from MAC address
- Divided into two parts
  - Prefix identifies physical network (locator)
  - Suffix identifies a host on the network (identity)

Dotted Decimal Notation (IPv4)

- Convenient for humans
- Divides IPv4 address into *octets* of eight bits each
- Represents each octet in decimal separated by dots
- Examples

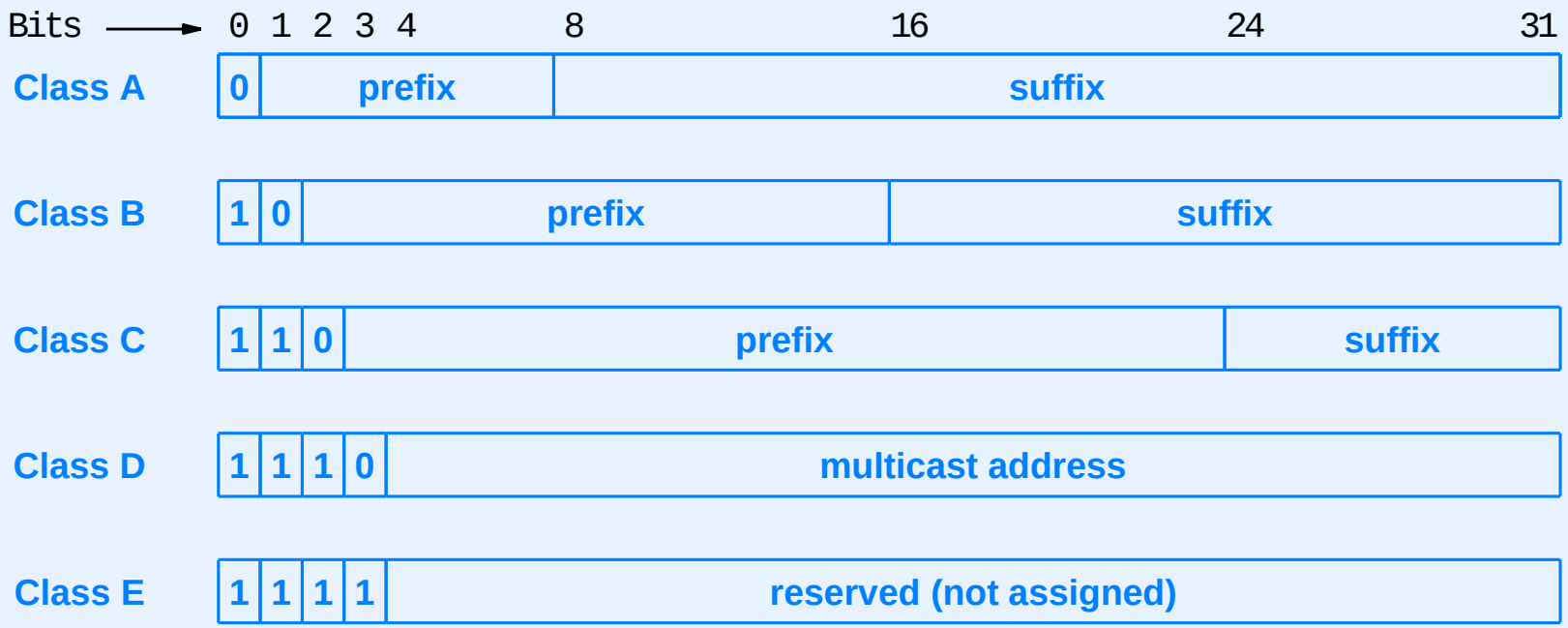
32-bit Binary Number				Equivalent Dotted Decimal
10000001	00110100	00000110	00000000	129.52.6.0
11000000	00000101	00110000	00000011	192.5.48.3
00001010	00000010	00000000	00100101	10.2.0.37
10000000	00001010	00000010	00000011	128.10.2.3
10000000	10000000	11111111	00000000	128.128.255.0

Division Between Prefix And Suffix

- Original scheme (*classful addressing*)
  - Each address divided on octet (8-bit) boundary
  - Division could be computed from the address
- Current scheme (*classless addressing*)
  - Formal name *Classless Inter-Domain Routing* (*CIDR*)
  - Division permitted at arbitrary bit position
  - Boundary must be specified external to the address

Classful Addressing

- Now historic
- Explains IPv4 multicast range



Address Mask

- Required with classless addressing
- Associated with a network
- Specifies division of addresses into network prefix and host suffix for that network
- 32-bit binary value
  - 1-bits correspond to prefix
  - 0-bits correspond to suffix
- Example mask that specifies six bits of prefix

11111100 00000000 00000000 00000000

CIDR Notation

- Used by humans to enter address mask
- Avoids dotted decimal errors
- Follows address with slash and integer *X*, where *X* is the number of prefix bits
- Example
  - In dotted decimal, a 26-bit mask is

    255 . 255 . 255 . 192

  - CIDR merely writes

    /26

Table Of CIDR And Dotted Decimal Equivalences

Length (CIDR)	Address Mask	Notes
/0	0 . 0 . 0 . 0	All 0s (equivalent to no mask)
/1	128 . 0 . 0 . 0	
/2	192 . 0 . 0 . 0	
/3	224 . 0 . 0 . 0	
/4	240 . 0 . 0 . 0	
/5	248 . 0 . 0 . 0	
/6	252 . 0 . 0 . 0	
/7	254 . 0 . 0 . 0	
/8	255 . 0 . 0 . 0	1-octet boundary
/9	255 . 128 . 0 . 0	
/10	255 . 192 . 0 . 0	
/11	255 . 224 . 0 . 0	
/12	255 . 240 . 0 . 0	
/13	255 . 248 . 0 . 0	
/14	255 . 252 . 0 . 0	
/15	255 . 254 . 0 . 0	
/16	255 . 255 . 0 . 0	2-octet boundary

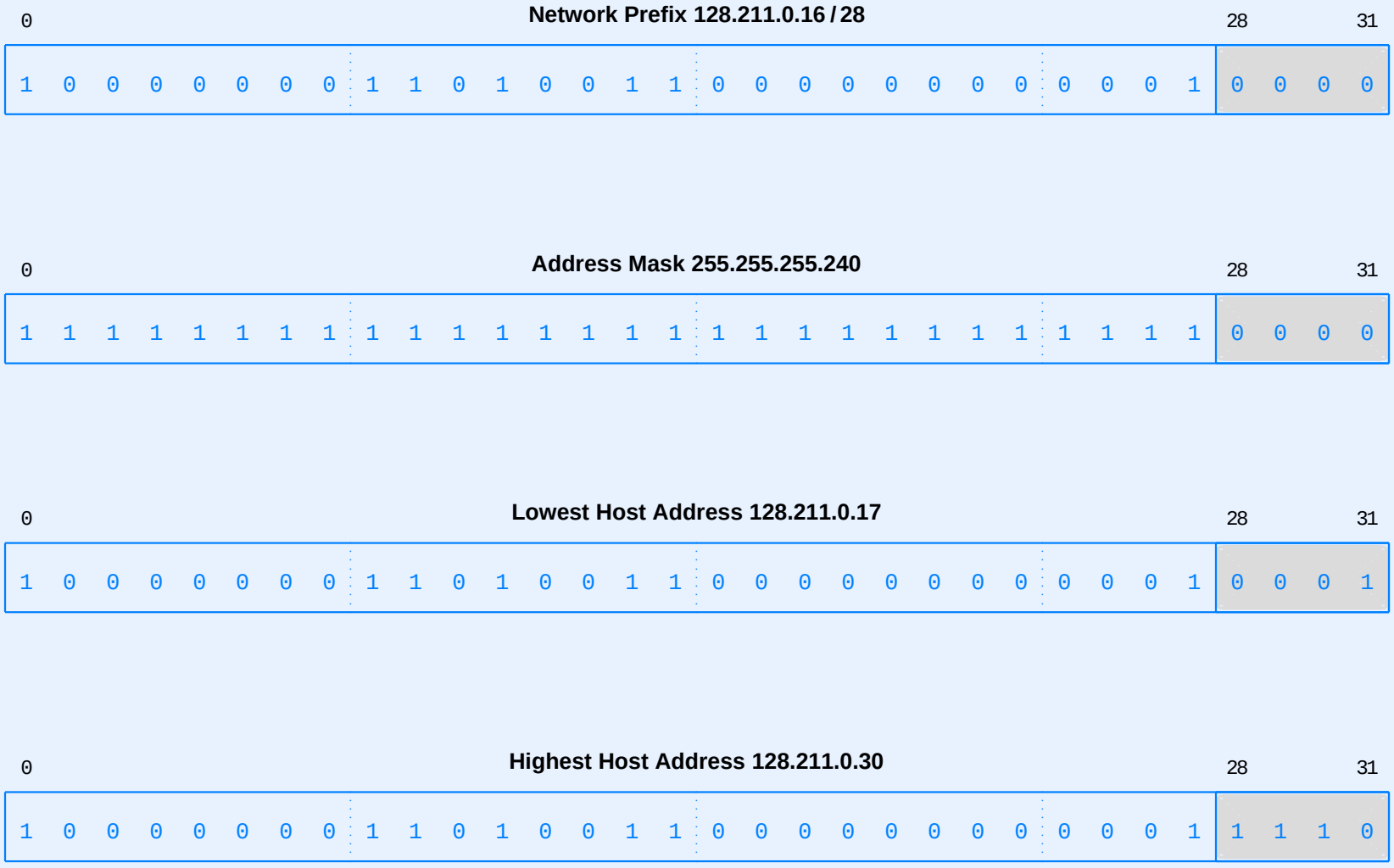
Table Of CIDR And Dotted Decimal Equivalences

Length (CIDR)	Address Mask	Notes
/17	255 . 255 . 128 . 0	
/18	255 . 255 . 192 . 0	
/19	255 . 255 . 224 . 0	
/20	255 . 255 . 240 . 0	
/21	255 . 255 . 248 . 0	
/22	255 . 255 . 252 . 0	
/23	255 . 255 . 254 . 0	
/24	255 . 255 . 255 . 0	3-octet boundary
/25	255 . 255 . 255 . 128	
/26	255 . 255 . 255 . 192	
/27	255 . 255 . 255 . 224	
/28	255 . 255 . 255 . 240	
/29	255 . 255 . 255 . 248	
/30	255 . 255 . 255 . 252	
/31	255 . 255 . 255 . 254	
/32	255 . 255 . 255 . 255	All 1s (host specific mask)

Why CIDR Is Useful

- ISPs assign IP addresses
- Corporate customer with *N* computers needs *N* addresses
- CIDR permits ISP to round to nearest power of two
- Example
  - Assume ISP owns address block 128.211.0.0/16
  - Customer has 12 computers
  - ISP assigns 4 bits of suffix to customer
  - Mask used is /28
  - Example: customer is assigned 128.211.0.16/28
  - Each computer at customer site has unique final 4 bits

Example Of A /28 Address Block



Special IPv4 Addresses

- Some address forms are reserved

Prefix	Suffix	Type Of Address	Purpose
all-0s	all-0s	this computer	used during bootstrap
network	all-0s	network	identifies a network
network	all-1s	directed broadcast	broadcast on specified net
all-1s	all-1s	limited broadcast	broadcast on local net
127/8	any	loopback	testing

- *Loopback address* (127.0.0.1) used for testing
  - Packets never leave the local host
- Addresses 240.0.0.0/8 and above are *multicast*

Host Address Count

- For a given network prefix, the all-0s and all-1s suffixes have special meaning
- Consequence: if a suffix has N bits, $2^N - 2$ hosts can be present

IP Addressing Principle

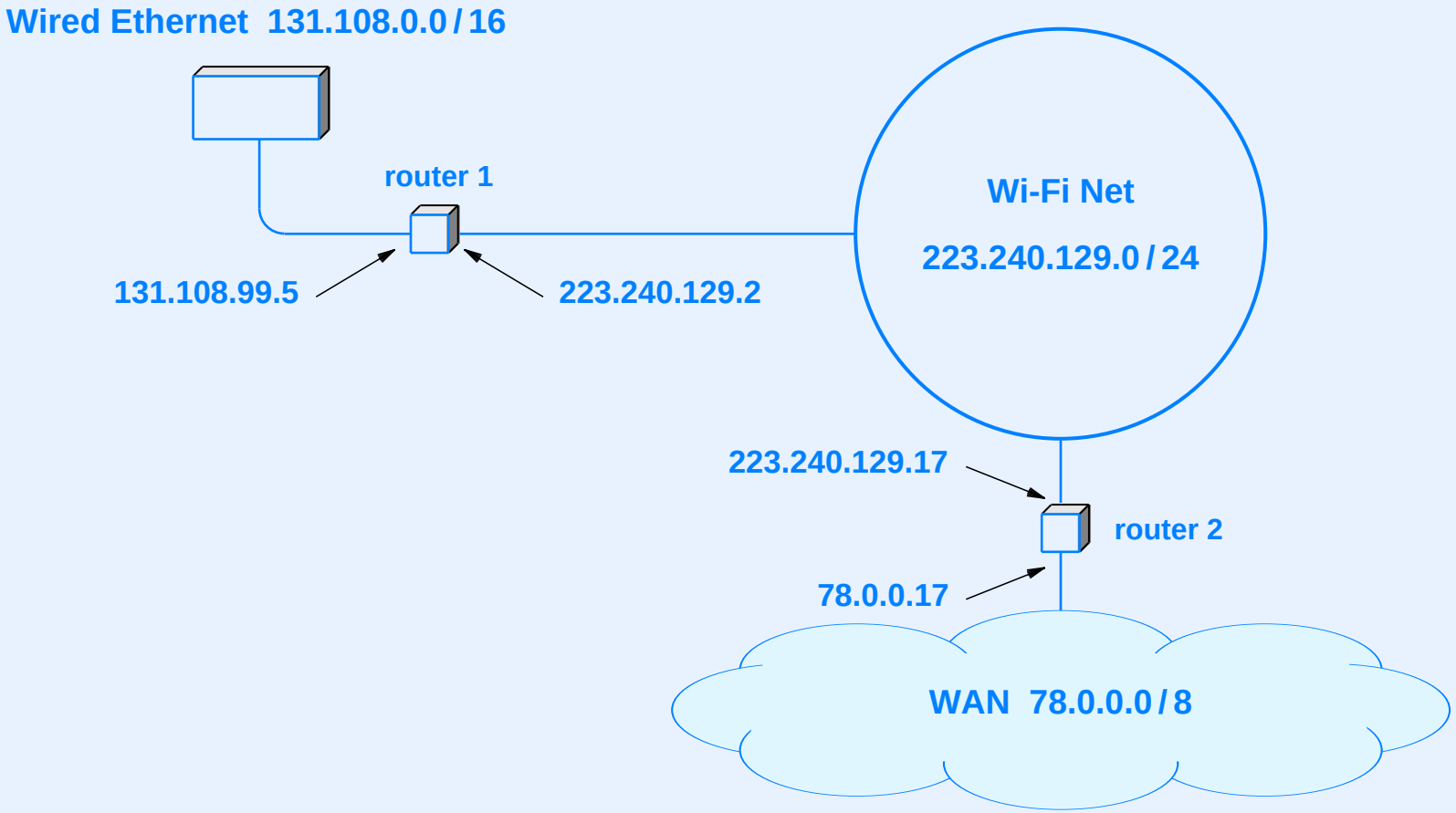
An IP address does not identify a specific computer. Instead, each IP address identifies a connection between a computer and a network.

- Consequence

A router or a host with multiple network connections must be assigned one IP address for each connection.

- Note: host with multiple network connections is called a *multi-homed host*

Illustration Of IPv4 Address Assignment

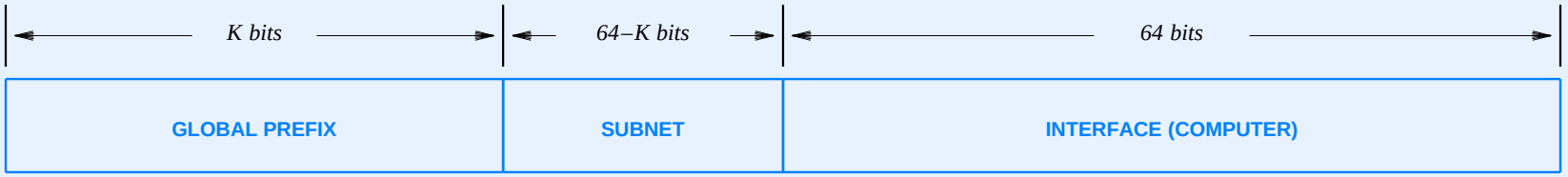


- Each network assigned a unique prefix
- Each host on a network assigned a unique suffix

IPv6 Host Addresses

- Like IPv4
  - Binary value
  - Divided into locator prefix and unique ID suffix
  - Identifies a connection to a network
- Unlike IPv4
  - 128 bits long
  - Suffix can be derived from MAC address
  - 3-level address hierarchy

The IPv6 3-Level Hierarchy



- Prefix size chosen by ISP
- Subnet area allows organization to have multiple networks

IPv6 Address Types

Type	Purpose
unicast	The address corresponds to a single computer. A datagram sent to the address is routed along a shortest path to the computer.
multicast	The address corresponds to a set of computers, and membership in the set can change at any time. IPv6 delivers one copy of the datagram to each member of the set.
anycast	The address corresponds to a set of computers that share a common prefix. A datagram sent to the address is delivered to exactly one of the computers (e.g., the computer closest to the sender).

Colon Hex Notation

- Syntactic form used by humans to enter addresses
- Replacement for IPv4's dotted decimal
- Expresses groups of 16 bits in hexadecimal separated by colons
- Example:

```
105.220.136.100.255.255.255.255.0.0.18.
            128.140.10.255.255
```

becomes

```
69DC:8864:FFFF:FFFF:0:1280:8C0A:FFFF
```

Colon Compression

- Many IPv6 addresses contain long strings of zeroes
- Successive zeros can be replaced by two colons
- Example

```
FF0C:0:0:0:0:0:0:B1
```

can be written:

```
FF0C::B1
```

Two Major Reasons To Adopt IPv6

- More addresses
  - Eventually, IPv4 addresses will be depleted
  - IPv6 provides more addresses than we will ever need

    $340,282,366,920,938,463,463,374,607,431,768,211,456$
  - 10^{24} addresses per square meter of the Earth's surface!
- Hype and excitement
  - Researchers view IPv6 as an opportunity to be part of the action
  - Industries view IPv6 as an opportunity for revenue enhancement

Internet Protocol Packets
(IP datagrams)

Internet Packets

Because it includes incompatible networks, the Internet cannot adopt a particular hardware packet format. To accommodate heterogeneity, the Internet Protocol defines a hardware-independent packet format.

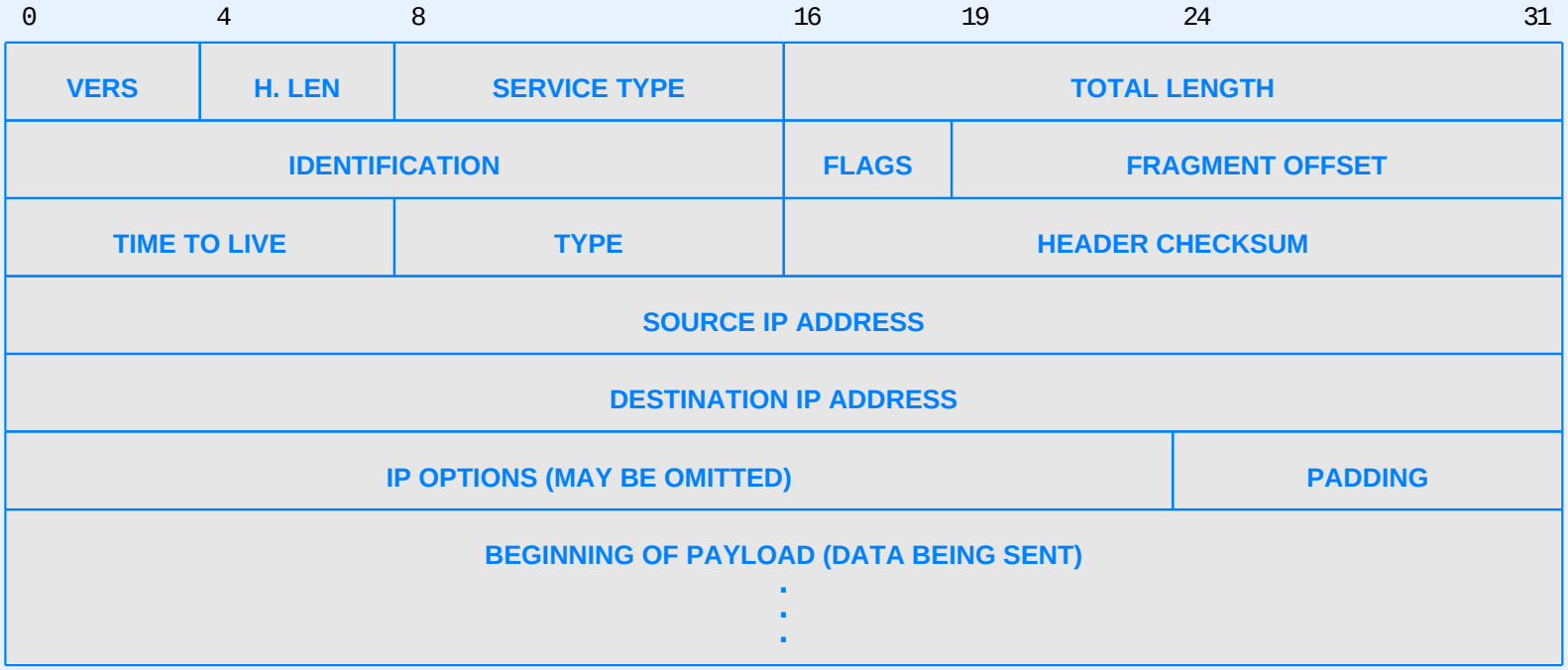
IP Datagram

- Virtual packet format used in the Internet
- Same general layout as a network frame

- Format of header determined by protocol version (*IPv4* or *IPv6*)
- Size of payload determined by application
  - Maximum payload is almost 64K octets
  - Typical datagram size is 1500 octets

IPv4 Datagram Header

- Most header fields have fixed size and position
- Header specifies source, destination, and content type

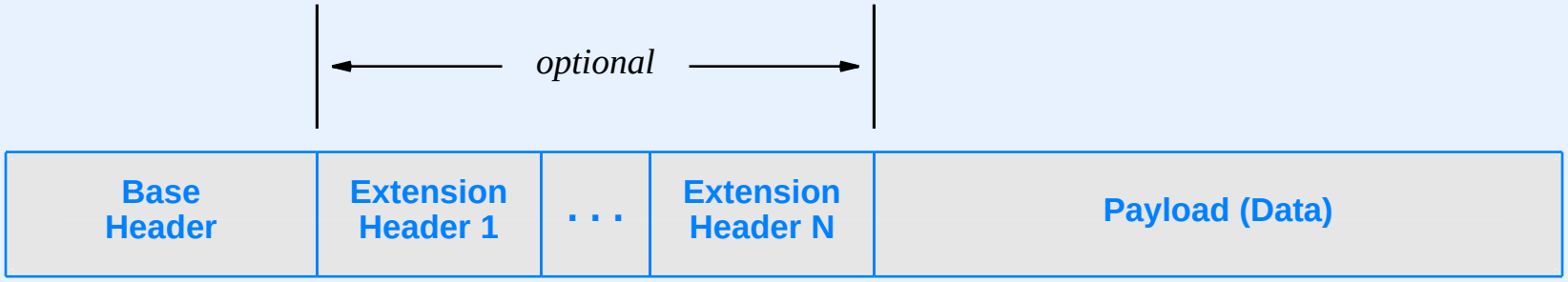


A Few Details

- *SOURCE IP ADDRESS* field gives the IPv4 address of the original source
- *DESTINATION IP ADDRESS* field gives the IPv4 address of the ultimate destination
- Intermediate router addresses do not appear in header
- Header size
  - Almost no Internet datagrams contain options
  - Therefore header length is usually 20 octets

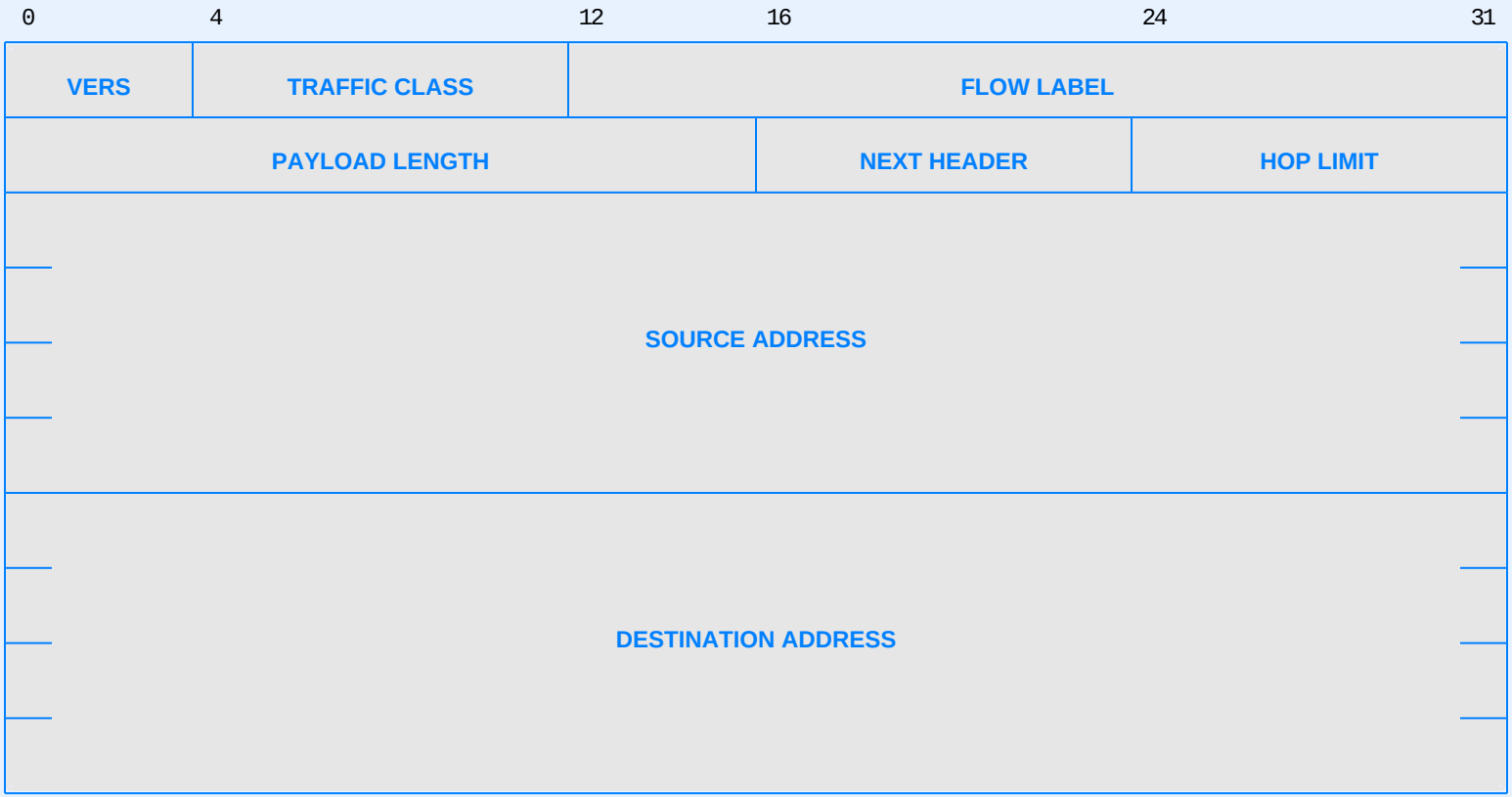
IPv6 Header Arrangement

- Multiple headers used: base plus zero or more extension(s)



- The figure is not to scale: extension headers and/or the payload can be much larger than the base header

IPv6 Base Header Format



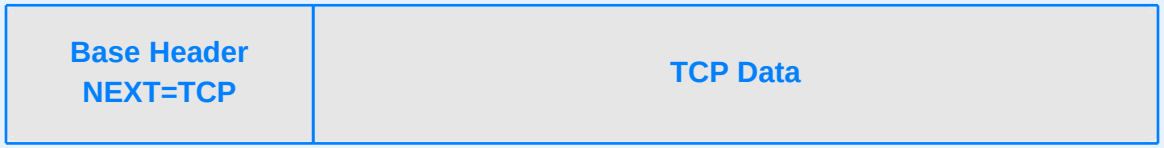
- *Flow Label* field allows datagram to be associated with a flow

Identifying Headers

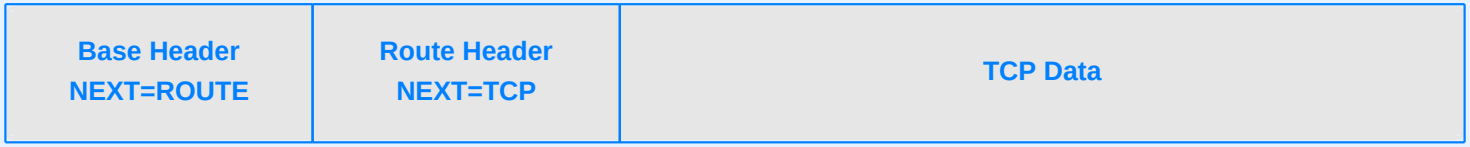
- Each header contains a *NEXT HEADER* field
- Value specifies the type of the next item
- Each layer 4 protocol (UDP, TCP, etc) is also assigned a type

Example Use Of Next Header Field

- Illustration of headers when a datagram contains a base header and transport protocol

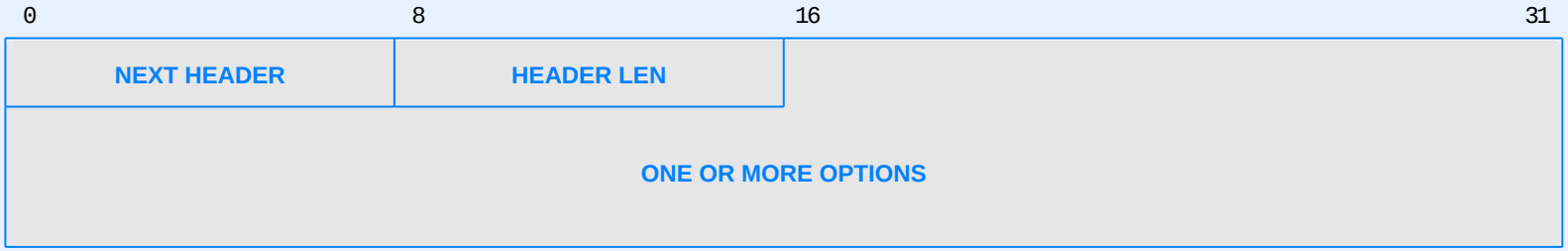


- Illustration of headers when a datagram also contains an optional *route header*



The Size Of An Extension Header

- Fixed length headers
  - Size is specified in the standards document
  - Protocol software contains size constant
- Variable length headers
  - Size is determined by sender
  - Header contains an explicit length field



Consequences For Packet Processing

- Consider a host or router that receives an IPv6 datagram
- The datagram contains a set of extension headers
- Each extension header can contain an explicit length field
- To parse the datagram, IP software must iterate through headers
- Conclusion: processing IPv6 can entail extra overhead

Datagram Forwarding

Internet Communication Paradigm

- Each datagram handled independently
- Datagram formed on source computer
- Source sends datagram to nearest router
- Router forwards datagram to next router along path to destination
- Final router delivers datagram to destination
- Datagram passes across a single physical network at each step

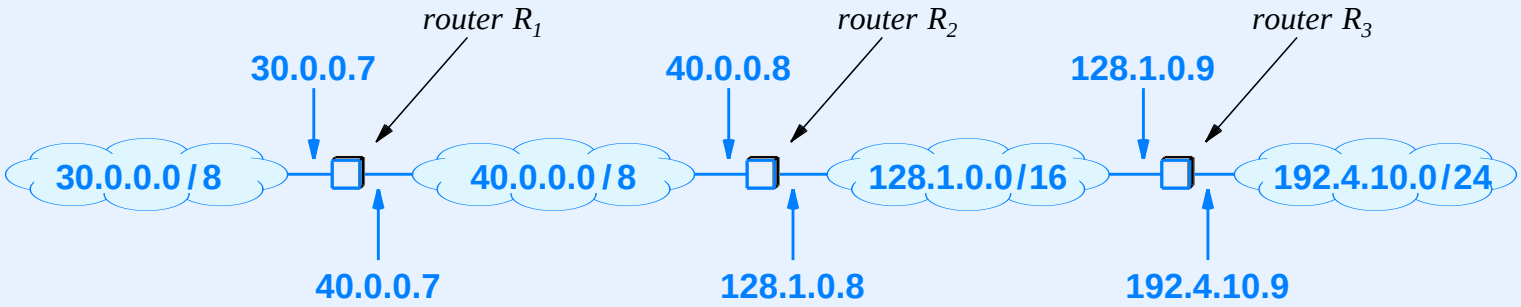
Datagram Forwarding

- Performed by initial host and each router along path
- Selects *next hop* for the datagram as either
  - Next router along the path
  - Ultimate destination
- Uses a *forwarding table* with one entry per network
- Important point: size of forwarding table proportional to number of networks in the Internet

Forwarding Table Entry

- Uses IP addresses only (no MAC addresses)
- Contains
  - Destination network IP prefix
  - Address mask for the destination network
  - IP address of next hop

Illustration Of An IPv4 Forwarding Table



(a)

Destination	Mask	Next Hop
30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	deliver direct
128.1.0.0	255.255.0.0	deliver direct
192.4.10.0	255.255.255.0	128.1.0.9

(b)

- In practice, table usually contains a *default* entry

Prefix Extraction

- Forwarding paradigm
  - Use network prefix when forwarding
  - Use host when delivering
- Conceptual forwarding step
  - Compare destination in each forwarding table entry with datagram's destination address, *D*
  - During comparison, only examine network prefix
- Note: mask in forwarding table makes comparison efficient

if ((Mask[i] & D) == Destination[i]) *forward to* NextHop[i];

Longest Prefix Match

- Classless addressing means forwarding table entries can be ambiguous
- Example: consider destination 128.10.2.3 and a table that includes the following two entries:

128.10.0.0/16 next hop A

128.10.2.0/24 next hop B

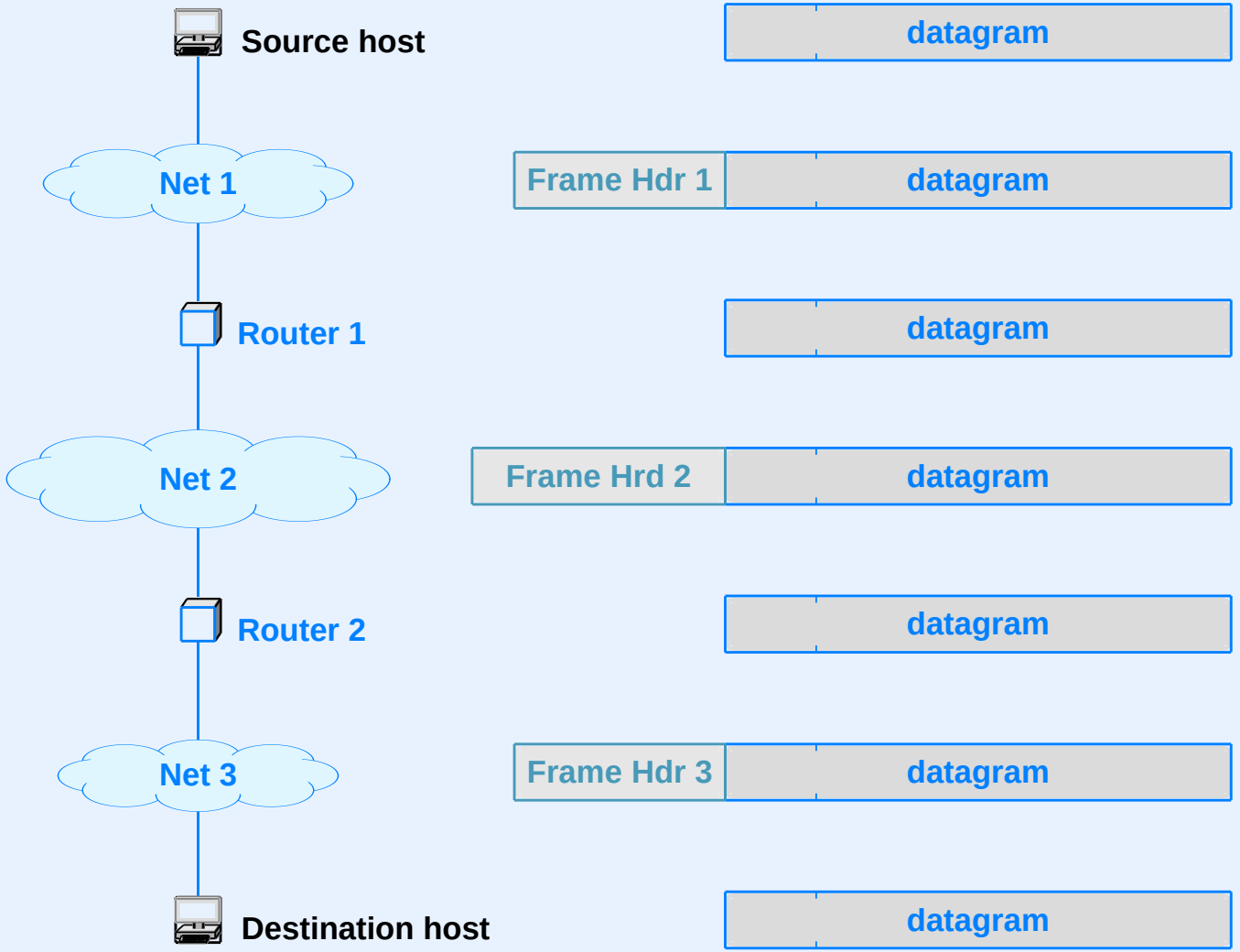
- The destination matches both of them!
- Solution: select the match that has the longest prefix (in the example, take next hop B)
- Known as *longest prefix match*

Datagram Encapsulation

- Needed because underlying network hardware does not understand datagrams
- Entire datagram travels in payload area of frame

- Frame header contains MAC address of *next hop*
- Frame only used for trip across one network: when frame arrives at next hop, datagram is extracted and frame is discarded
- Datagram remains intact *end-to-end*

Illustration Of Encapsulation

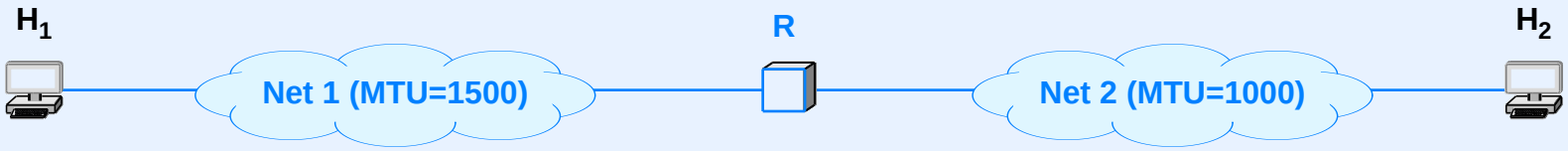


Semantics Of Internet Communication

- IP uses *best effort delivery* semantics
- IP attempts to deliver each datagram, but specifies that a datagram can be
  - Lost
  - Duplicated
  - Delayed
  - Delivered out-of-order
  - Delivered with bits scrambled
- Motivation: accommodate *any* underlying network
- Note: in practice, IP works and it works well

MTU And Network Heterogeneity

- Each network technology specifies a *Maximum Transfer Unit* (*MTU*) that is the largest amount of data that can be sent in a packet
- Example: Ethernet MTU is 1500 octets
- Datagram can be as large as the network MTU
- Consider a 1500-octet datagram set from H_1 to H_2 in the following network



- Datagram can reach router R, but cannot traverse Net 2

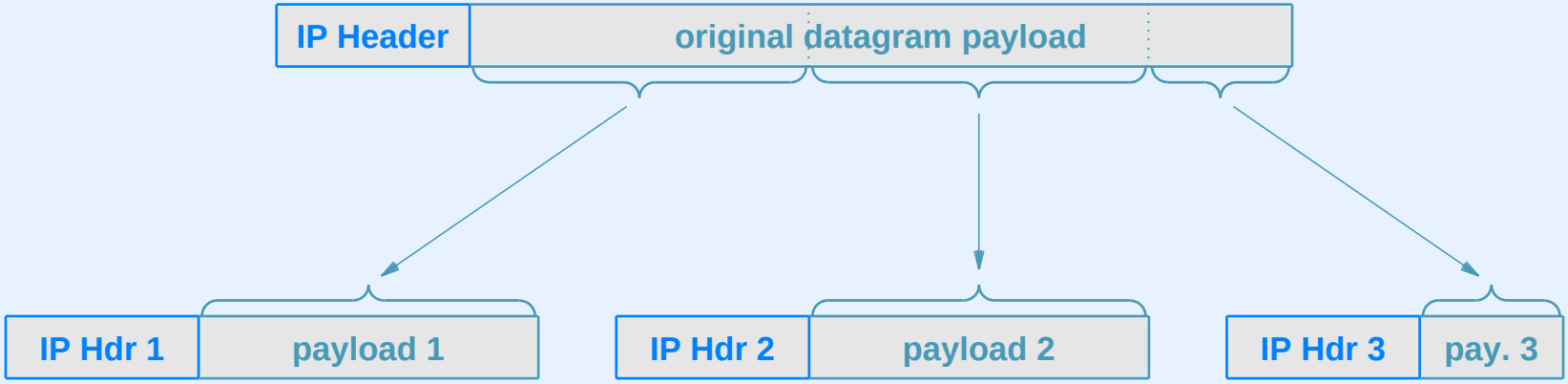
Datagram Fragmentation

- Technique for accommodating heterogeneous MTUs
- Needed if datagram exceeds MTU
- Original datagram divided into smaller datagrams called *fragments*
- Header of fragment derived from original datagram header
- Each fragment is forwarded independently
- IPv4 allows routers to perform fragmentation
- IPv6 requires sending host to perform fragmentation
- Important principle for both IPv4 and IPv6:

The ultimate destination *reassembles* fragments.

The General Idea Of Fragmentation

- Divide the payload into a series of datagrams



- Note: the tail fragment may be smaller than the others

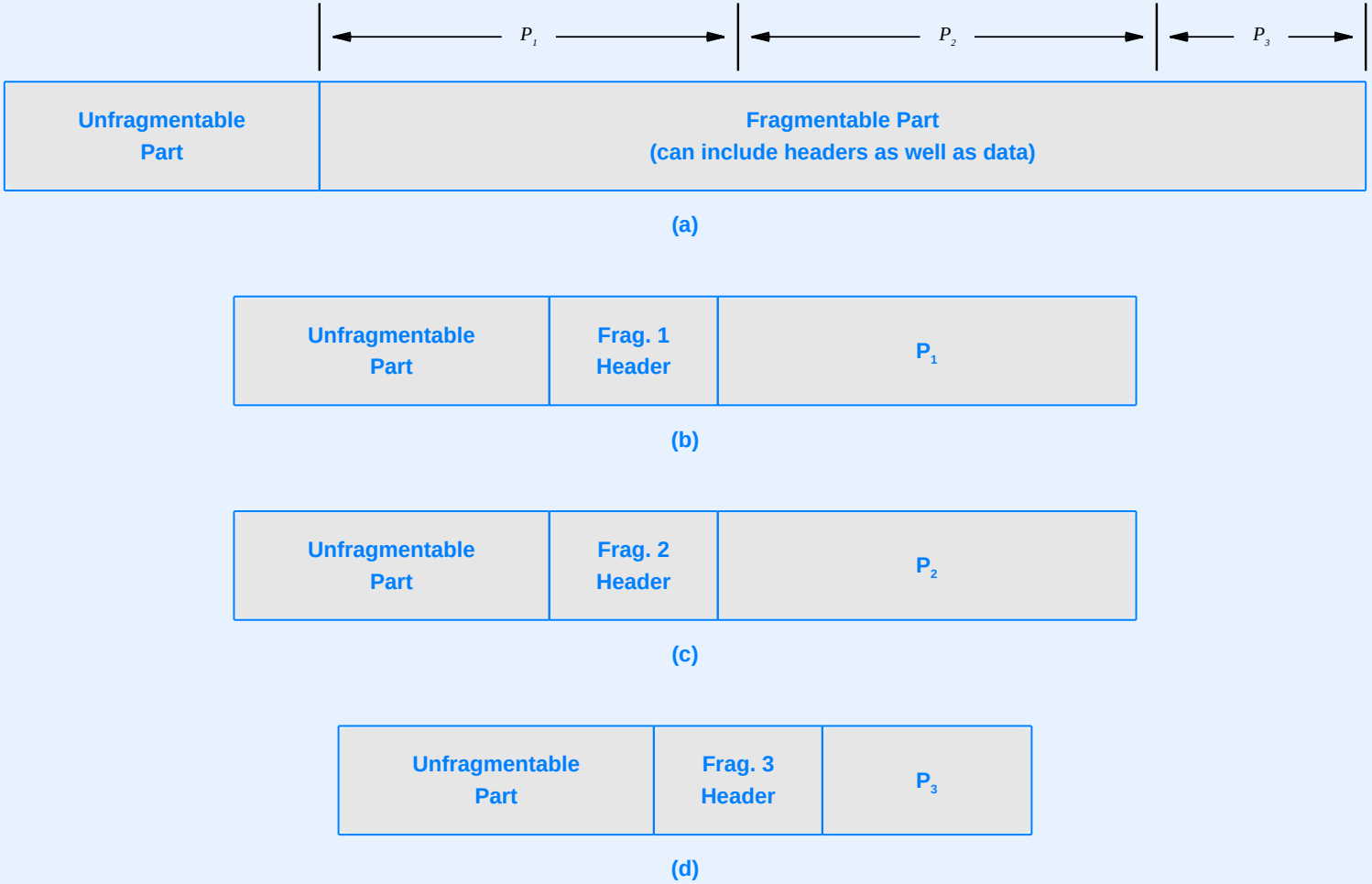
IPv4 Fragmentation Details

- Datagram header contains fixed fields that control fragmentation
- A bit in *FLAGS* field specifies whether given datagram is a fragment or complete datagram
- An additional *FLAGS* bit specifies whether the fragment carries the tail of the original datagram
- *OFFSET* field specifies where the payload belongs in the original datagram

IPv6 Fragmentation Details

- Always performed by source, never by routers
- Rule: no header changes as an IPv6 datagram traverses the Internet
- Consequences
  - Source must discover *path MTU*
  - Separate extension header contains fragmentation information (same items as IPv4)
- Fragmentable part of datagram may include some extension headers

Illustration of IPv6 Fragmentation



- A datagram (a) divided into fragments (b through d)

Collecting Fragments

- Destination collects incoming fragments
- *IDENTIFICATION* field used to group related fragments
- *OFFSET* field allows receiver to recreate the original payload
- *LAST FRAGMENT* bit allows receiver to know when all fragments have arrived
- If a fragment fails to arrive within a timeout period, entire datagram is discarded
- Note: if an IPv4 fragment is divided into subfragments, reassembly does not require reassembling subfragments

Address Resolution

Review Of Datagram Transmission

- Host or router has datagram to send
- IP uses longest-prefix match to look up datagram's destination address in forwarding table and obtains
  - IP address of next hop
  - Network over which to send (in case there is more than one network connection)
- IP encapsulates datagram in frame (entire datagram placed in payload area of frame)
- Is the resulting frame ready to send to the next hop?

No!

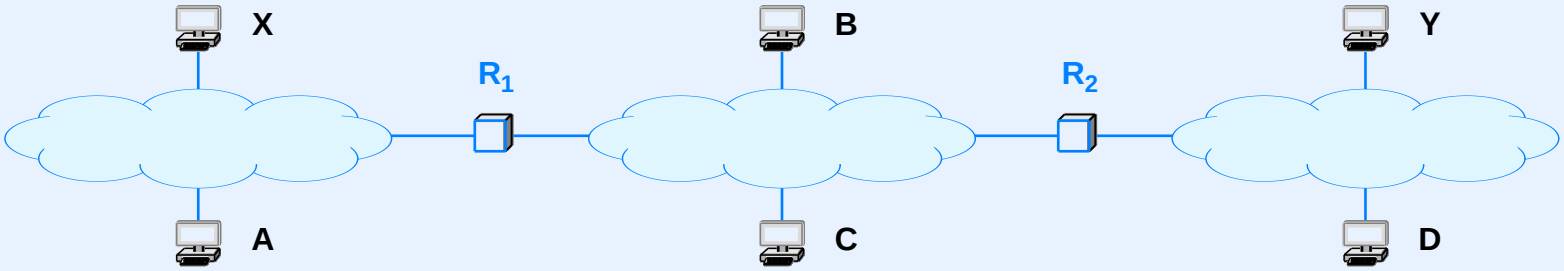
Hardware And Protocol Addressing

- Underlying network hardware
  - Only understands MAC addresses
  - Requires each outgoing frame to contain the MAC address of the next hop
- IP forwarding
  - Deals only with (abstract) IP addresses
  - Computes the IP address of the next hop
- Conclusion

The IP address of the next hop must be translated to a MAC address before a frame can be sent.

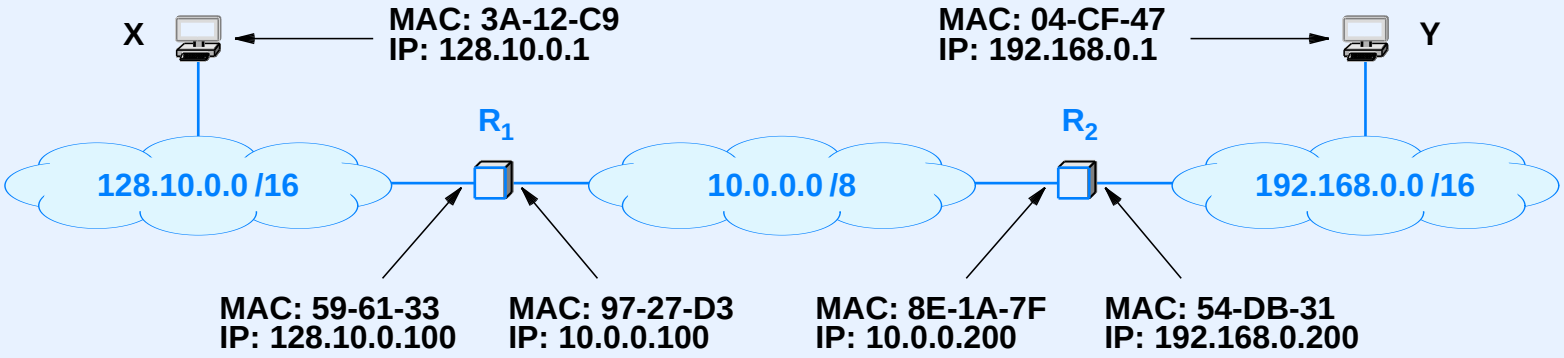
Address Resolution

- Translates IP address to equivalent MAC address that the hardware understands
- IP address is said to be *resolved*
- Restricted to a single physical network at a time
- Example: consider computer X sending to computer Y



- A MAC address is needed at each hop

An Example With MAC Addresses



Sender	NEXT-HOP	SRC MAC	DST MAC	SRC IP	DST IP
X	128.10.0.100	3A-12-C9	59-61-33	128.10.0.1	192.168.0.1
R_1	10.0.0.200	97-27-D3	8E-1A-7F	128.10.0.1	192.168.0.1
R_2	192.168.0.1	54-DB-31	04-CF-47	128.10.0.1	192.168.0.1

- How can a host or router find the MAC address of the next hop?

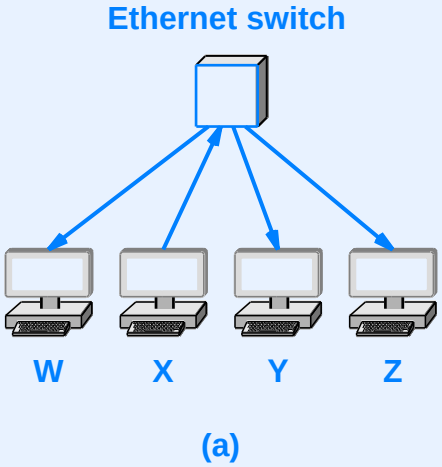
Address Resolution Protocol (ARP)

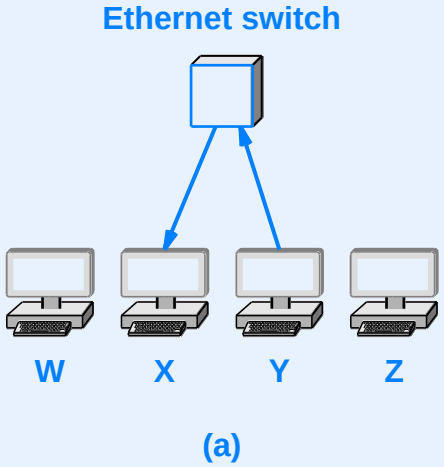
- Designed for IPv4 over Ethernet
- Used by two computers on the same physical network
- Allows a computer to find the MAC address of another computer
- Operates at layer 2
- Uses network to exchange messages
- Computer seeking an address sends request to which another replies

Example Of ARP Exchange

- Assume
  - Four computers attached to an Ethernet
  - Computer B has a datagram to send
- Computer B
  - Uses forwarding table to find next-hop address I_C
  - Broadcasts an ARP request: "I'm looking for a computer with IP address I_C"
- Computer C
  - Receives the request and replies; "I'm the computer with IP address I_C"

Illustration Of The ARP message Exchange

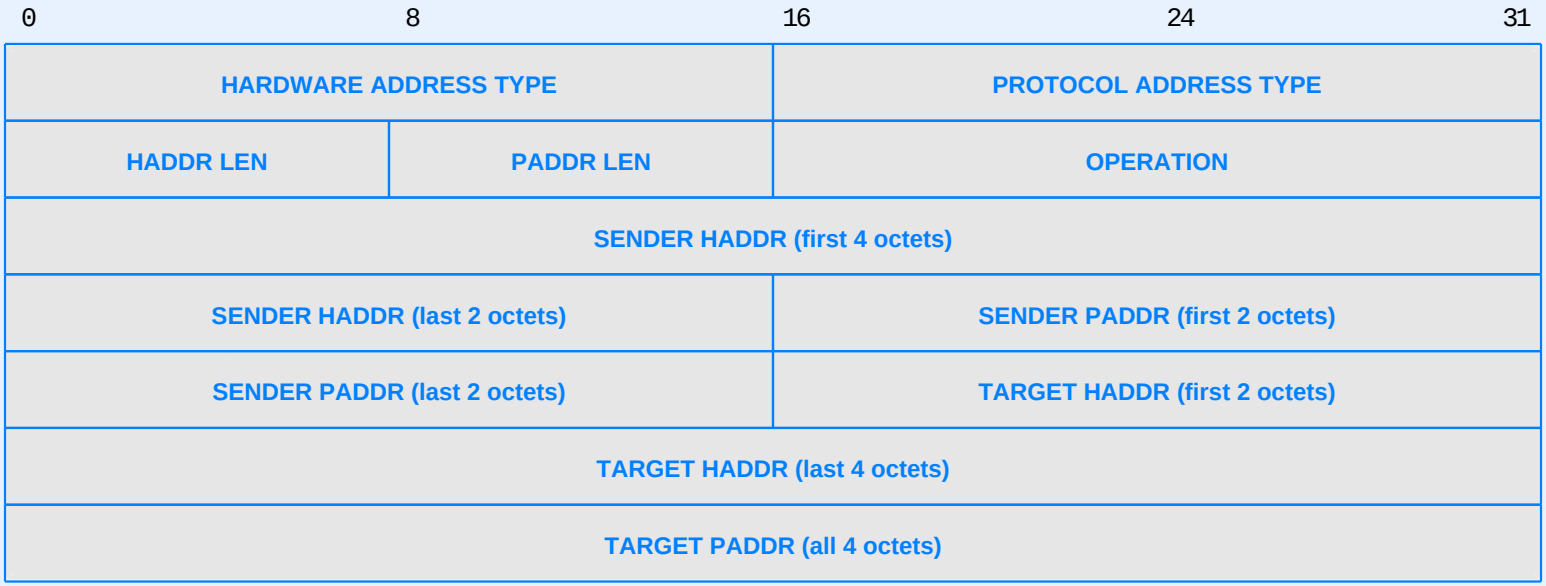




- Request is broadcast to all computers
- Only the intended recipient replies
- Reply is sent unicast

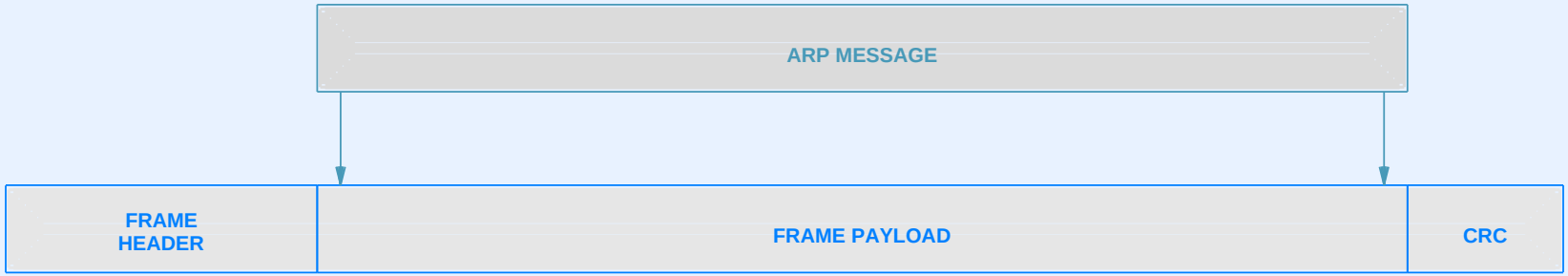
ARP Message Format

- Sufficiently general to permit
  - Arbitrary high-level protocol address
  - Arbitrary hardware address
- In practice, only used with IP and 48-bit Ethernet addresses



ARP Encapsulation

- ARP message is placed in payload area of hardware frame
- When used with Ethernet, type is 0x0806
- Source and destination MAC addresses must be added to frame header before sending



ARP Algorithm And Caching

Given:

An incoming ARP request or response

Purpose:

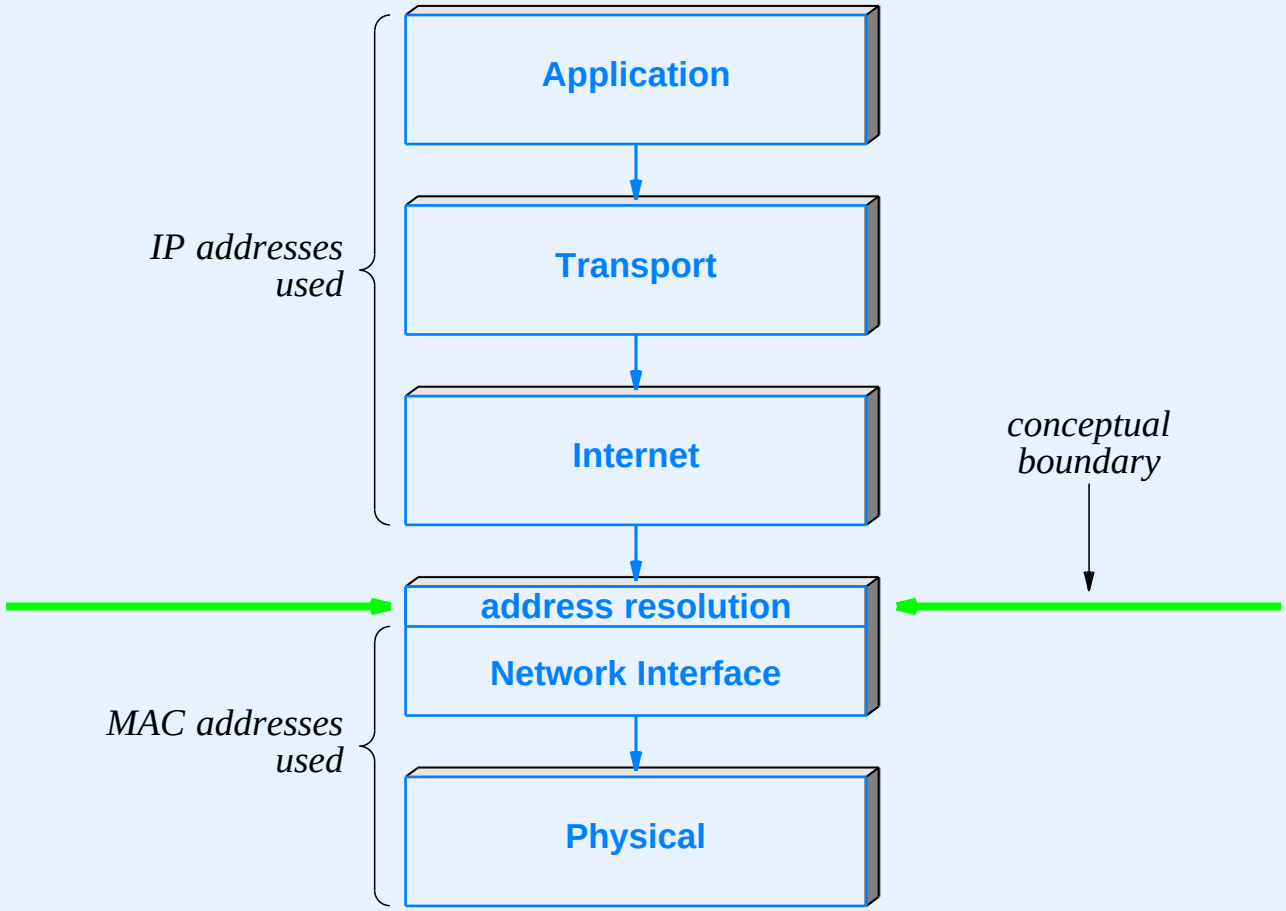
Process the message and update the ARP cache

Method:

```
Extract sender's IP address, I, and MAC address, M
If (address I is already in the ARP cache) {
    Replace corresponding MAC address with M;
}
if (message is a request and target is "me") {
    Add sender's entry to the ARP cache providing
        no entry exists;
    Generate and send a response;
}
```

Boundary Between Protocol And MAC Addressing

- ARP isolates hardware addresses, allowing layers above to use only IP



Thought Problem

- ARP is sometimes cited as a security weakness
- If someone gains access to a given network, how can they exploit ARP to intercept packets?

Address Binding With IPv6

- IPv6 does not use ARP
- Instead, IPv6 defines a new address binding mechanism known as *IPv6 Neighbor Discovery* (*IPv6-ND*)
- Sends a multicast request to find neighbors
- Polls neighbors periodically, even if no datagrams are being sent

Error Reporting Mechanism

IP Error Detection And Reporting

- Recall that IP allows datagrams to be
  - Lost
  - Duplicated
  - Delayed
  - Delivered out-of-order
- Why is error reporting needed?
- Answer: *best-effort* does not mean "careless" — the design is intended to tolerate errors in the underlying networks, not to introduce them
- IP reports problems when they are detected

General Error Detection

- A variety of basic error detection mechanisms exist
- Examples
  - Parity bits and other forward error codes can detect transmission errors
  - A CRC can detect an incorrect frame
  - The IP header checksum can detect an incorrect datagram header
  - IP's TTL (hop limit) can detect a routing loop
  - A reassembly timer can detect lost fragments
- Only some types of errors can be reported

Internet Control Message Protocol (ICMP)

- Required and integral part of IP
- Reports errors back to the original source
- Uses IP to carry messages
- Defines many types of messages, each with a specific format and contents
- Includes information messages as well as error reports
- ICMPv4 and ICMPv6 share many messages

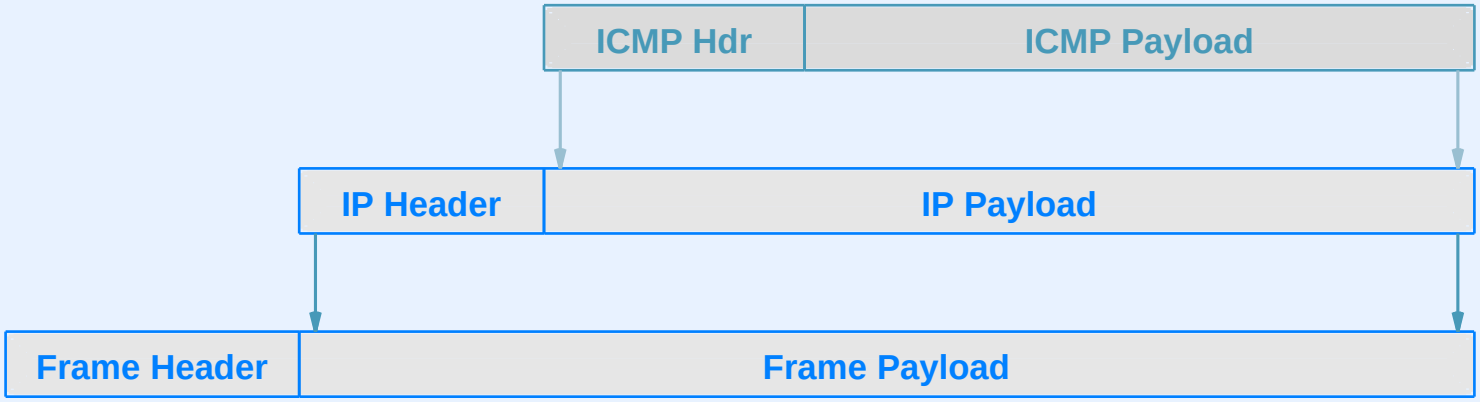
Example ICMP Messages

Number	Type	Purpose
0	Echo Reply	Used by the ping program
3	Dest. Unreachable	Datagram could not be delivered
5	Redirect	Host must change a route
8	Echo Request	Used by the ping program
11	Time Exceeded	TTL expired or fragments timed out
12	Parameter Problem	IP header is incorrect
30	Traceroute	Used by the traceroute program

- Most heavily-used ICMP messages are 8 and 0, which are sent and received by the *ping* program

ICMP Encapsulation

- Two levels of encapsulation
  - ICMP message encapsulated in an IP datagram
  - IP datagram encapsulated in a network frame



Example Of An ICMP Error Report

- Host *S* creates a datagram for destination *D*
- *S* sets the TTL to 255 and sends the datagram
- Datagram reaches a loop in the middle of the Internet
- Datagram circulates around the loop until the TTL reaches zero
- Router that decrements the TTL to zero
  - Sends a type 11 ICMP message to *S*
  - Discards the datagram that caused the problem

Configuration

Protocol Configuration

- Many items must be set before protocols can be used
  - IP address of each network interface
  - Address mask for each network
  - Initial values in the forwarding table
- Process is known as *protocol configuration*
- Usually occurs when operating system boots
- Two basic approaches
  - Manual
  - Automatic

Manual Configuration

- Used for IP routers or host that has a permanent IP address
- Manager
  - Enters configuration once
  - Specifies that the configuration be saved in non-volatile storage
  - Interfaces include *Command Line Interface* (*CLI*) and web
- OS
  - Fetches values from non-volatile storage whenever the device boots

Automatic Configuration

- Used primarily for hosts
- Initially created for diskless workstations
- Basic idea
  - Use network to obtain configuration information
  - Configure protocol software, and then start to run applications
- A seeming paradox

> Automatic configuration requires a computer to be able to use a network before the computer's protocol parameters have been configured.

Ways To Solve The Paradox

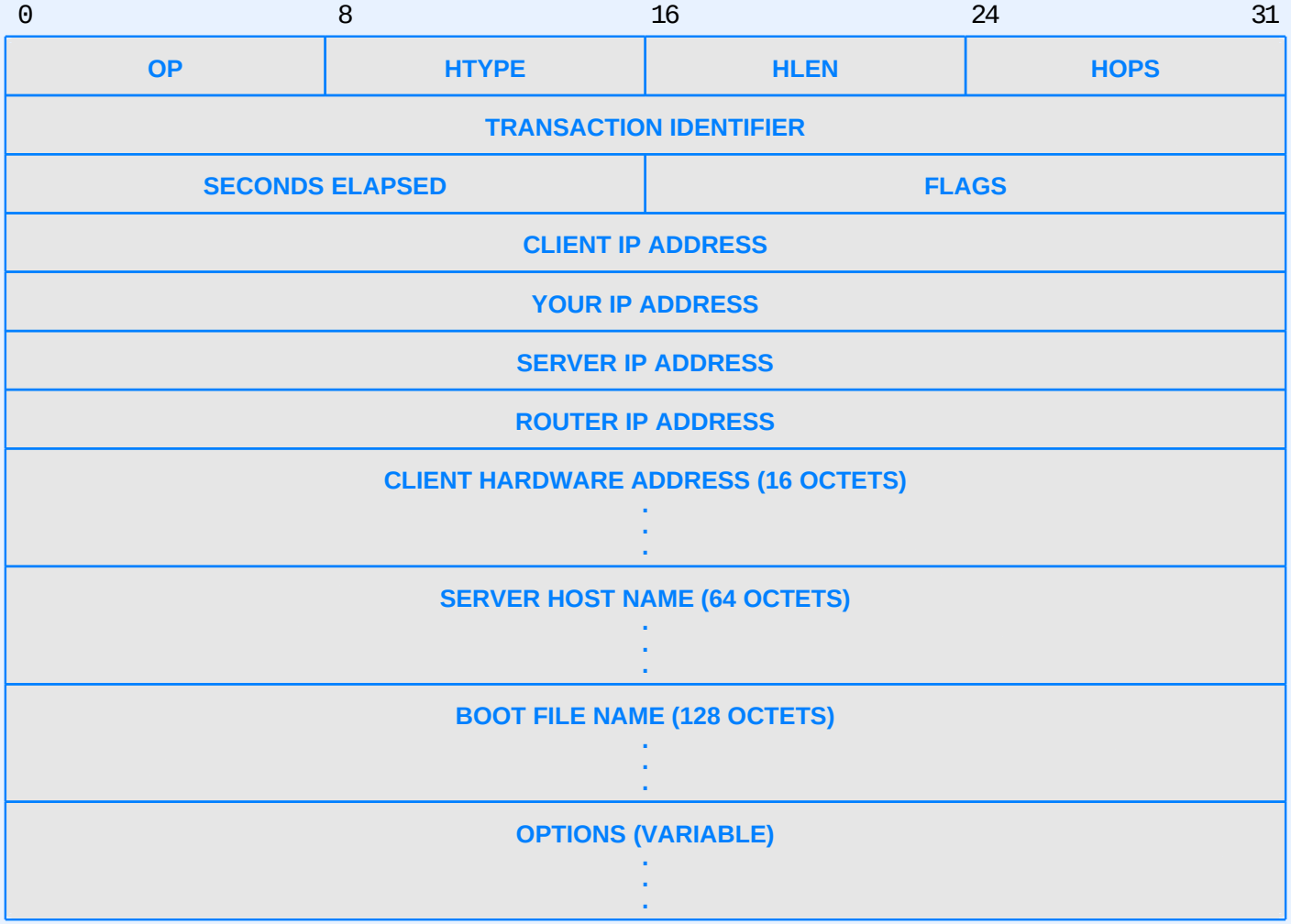
- Use layer 2 protocols to obtain layer 3 parameters, then use layer 3 to obtain higher layers
  - Historic approach
  - Relied on Ethernet broadcast
  - One computer on a network responded to requests
- Use layer 3 to obtain all parameters
  - Current approach
  - Relies on IP broadcast (IPv4) or multicast (IPv6)
  - Means routers can forward requests to a remote server

Dynamic Host Configuration Protocol (DHCP)

- The standard protocol for automatic configuration
- Popular in private enterprises as well as with service providers
- Host broadcasts/multicasts a request and receives a reply
- Single message exchange allows a host to obtain
  - An IP address and address mask to use
  - The IP address of a default router
  - The address of a DNS server
  - A DNS name
  - The location of an image to boot (optional)

DHCP Message Format

- Same message format used for requests and responses



DHCP Protocol

- Significant features of the protocol
  - Recovers from loss or duplication
  - Avoids synchronized flooding of requests after a power-failure and restart
  - Host discovers DHCP server once and caches server address for future interaction
- Derived from BOOTstrap Protocol (BOOTP), but adds dynamic address assignment

Address Lease Paradigm

- DHCP server
  - Owns a set of IP addresses
  - Chooses an address from the set when a request arrives
  - Issues a *lease* for the address for specified time, *T*
- Client
  - Obtains an address and starts a timer for *T* time units
  - Uses the address to communicate
  - When the timer expires, requests the server *renew* the lease
  - Either receives a renewal and restarts timer or stops using the address

Thought Problem

- Consider how addresses are assigned
- An ISP using DHCP can choose which IP address to assign to a customer at a given time
- There are two approaches
  - The ISP can remember which address was previously assigned to each customer and use the same address
  - The ISP can assign addresses at random, meaning the customer will not retain the same address
- Many ISPs try to change the address frequently
- Why?

IPv6 Configuration

- DHCPv6 has been defined, but...
- IPv6 prefers a new procedure known as

IPv6 autoconfiguration

- General idea: host can generate an address without using a server
- Motivation: allow two hosts to communicate without further infrastructure

Steps In IPv6 Autoconfiguration

- Obtain a network prefix
  - Convention is to use a /64 prefix
  - Globally-valid prefix can be obtained from a router
  - Local-scope prefix created if no router available
- Generate a unique suffix
- Verify that no one else on the network is using the resulting address

IPv6 Autoconfiguration in Practice

- Need a unique host suffix
- For /64 network, a 64-bit host suffix is needed
- Recommended approach
  - Start with MAC address (globally unique, but only 48 bits)
  - Create a 64-bit value
- IEEE standard EUI-64 specifies how 48 bits of an IEEE MAC address are placed in a 64-bit host suffix

Network Address Translation (NAT)

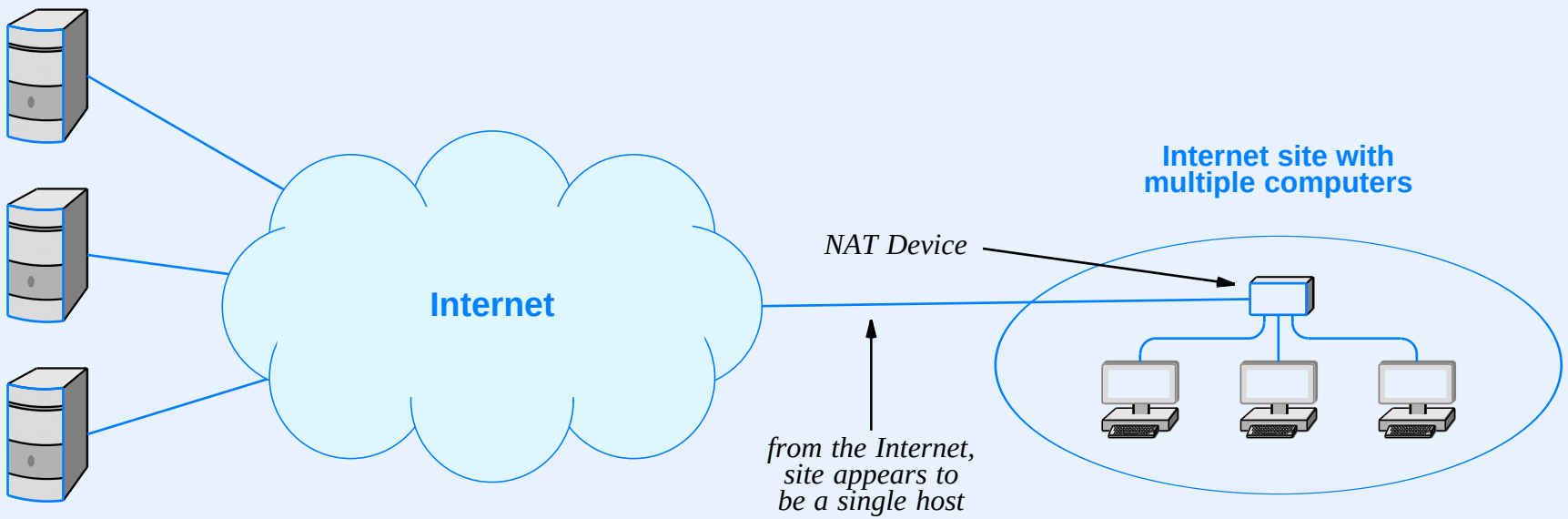
NAT Motivation

- IPv4 was running out of addresses
- ISPs only want to limit a customer to one IP address at any time, but customers want multiple devices to be online
- Engineers invented *Network Address Translation* (*NAT*) as a way to solve both problems

NAT Operation

- Conceptually, NAT device is located between computers at a site and the rest of the Internet
- Site
  - Only needs one globally-valid IP address
  - Can have multiple local hosts using the Internet
- Local host has full Internet access
- Service is *transparent*
  - No change in protocols on local hosts
  - No change in protocols on Internet servers

Conceptual Organization Of NAT



- NAT is said to be *in-line*
- From the Internet, site appears to be a single computer
- From within the site, each computer appears to have an independent connection to the Internet

Addresses Used by NAT

- NAT device runs a DHCP server to hand out IP addresses to computers at the site
- Addresses assigned are IPv6 *link-local* or IPv4 *private*

Block	Description
10.0.0.0/8	Class A private address block
169.254.0.0/16	Class B private address block
172.16.0.0/12	16 contiguous Class B blocks
192.168.0.0/16	256 contiguous Class C blocks

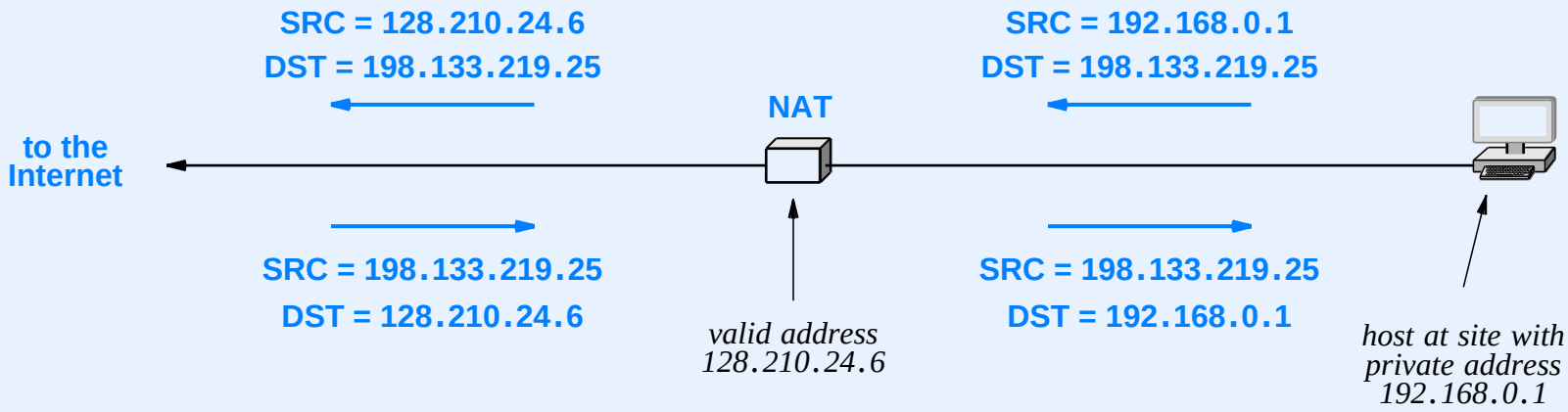
  - NAT translates source and/or destination addresses in datagrams that pass between the site and the Internet

NAT Variants

- Basic NAT
  - Only translates IP addresses
  - Seldom used in practice
- NAPT
  - Translates IP address and transport-layer port numbers
  - Most widely-used type of NAT
- Twice NAT
  - Works with DNS server
  - Provides NAPT plus ability to accept incoming communication

Example Of Basic NAT

- Suppose
  - NAT box has globally-valid IP address of 128.210.24.6
  - Computer at a site has private address 192.168.0.1
  - Computer contacts Internet site 198.133.219.25
- Resulting translation is:



Implementation Of NAT

- NAT device keeps an internal translation table
- Table stores translations for both outgoing and incoming datagrams
- Values filled in automatically when computer at site first sends datagram to the Internet
- Translation table for previous example

Direction	Field	Old Value	New Value
out	IP Source	192.168.0.1	128.210.24.6
	IP Destination	198.133.219.25	-- no change --
in	IP Source	198.133.219.25	-- no change --
	IP Destination	128.210.24.6	192.168.0.1

Transport-Layer NAT (NAPT)

- Handles TCP, UDP, and ICMP
- Translates TCP/UDP protocol port numbers as well as IP addresses
- Permits multiple computers at a site to contact the *same Internet service* simultaneously without interference
- Examples:
  - Two computers at a site download songs from iTunes
  - Three computers at a site contact Google simultaneously

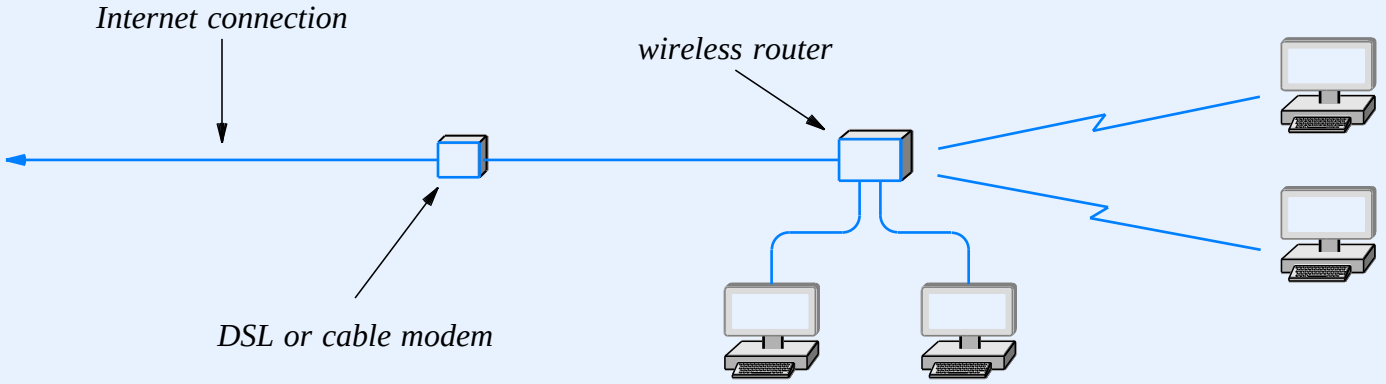
Example Of NAPT Translation

- Suppose
  - Computers at site have private addresses assigned from private address block 192.168/16
  - Two computers at the site each contact TCP port 30000 on computer 128 210.19.20
- NAPT chooses a new port number for each and translates

Dir.	Fields	Old Value	New Value
out	IP SRC:TCP SRC	192.168.0.1:30000	128.10.24.6:40001
out	IP SRC:TCP SRC	192.168.0.2:30000	128.10.24.6:40002
in	IP DEST:TCP DEST	128.10.24.6:40001	192.168.0.1:30000
in	IP DEST:TCP DEST	128.10.24.6:40002	192.168.0.2:30000

NAT In Practice

- Many consumer products have NAT built in
- Examples:
  - Cable and DSL modems
  - Wireless routers
- Note that most wireless routers provide both wired and wireless network connections; they provide NAT on all connections



Transport Layer Protocols: Characteristics And Techniques

What Should A Network Provide?

- One possibility: network centric
  - Network offers all services, such as email, web, etc
  - Host accesses services
  - Network authenticates user, handles reliability
  - Know as *customer-provider communication*
- Another possibility: network provides communication
  - Network only transfers packets
  - Applications handle everything else, including reliability, flow control, and authentication
  - Known as *end-to-end communication*

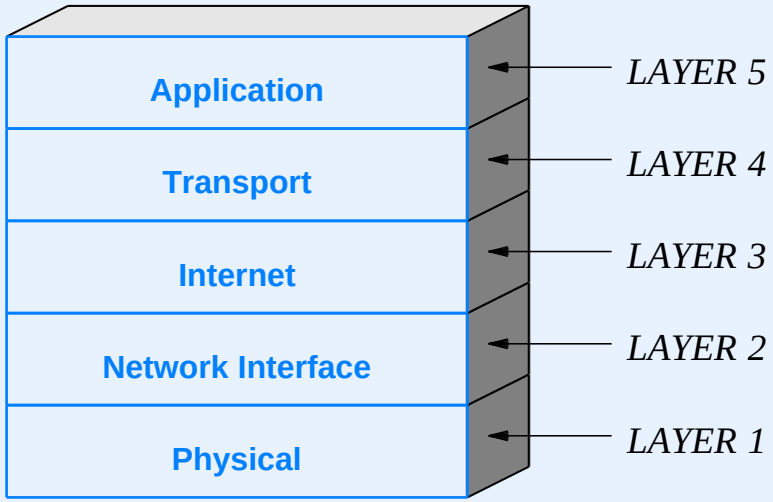
End-To-End Principle

- Fundamental concept in the Internet
- Network provides best-effort packet transport
- Endpoints
  - Control communication
  - Provide all reliability
- Consequence

Some of the most complex protocols in the Internet protocol suite run in hosts rather than in routers.

Transport Layer

- Layer between applications and IP



- Allows multiple applications on a given host to communicate with applications on other hosts
- Uses IP to carry messages

Problems A Transport Protocol Can Handle

- Accommodate speed mismatch between sender and receiver
- Detect and recover from datagram loss
- Eliminate duplicate packets
- Guarantee that messages arrive in order
- Respond to congestion in the Internet
- Prevent delayed packets from being misinterpreted
- Verify that data was not corrupted during transit
- Ensure that each party has agreed to communicate
- Note: a given transport protocol may not handle all problems

Techniques Transport Protocols Use

- Application demultiplexing
  - Sender places a value in each outgoing packet that identifies an application on the receiving host
  - Receiver uses the value to determine which application should receive the packet
- Flow-control mechanisms
  - Receiver informs sender of acceptable data rate
  - Sender limits rate to prevent overrunning the receiver

Techniques Transport Protocols Use (continued)

- Congestion control mechanisms
  - Receiver or network informs sender about congestion in the network
  - Sender reduces data rate (packet rate) until congestion subsides
- Sequence numbers
  - Sender places a *sequence number* in each packet
  - Receiver uses the sequence numbers to ensure no packets are missing and that packets are delivered in the correct order

Techniques Transport Protocols Use (continued)

- Positive acknowledgement with retransmission
  - Receiver sends *acknowledgement* to inform sender when a packet arrives
  - Sender *retransmits* packet if acknowledgement fails to arrive within a specified time
- Sliding window
  - Instead of transmitting a packet and waiting for an acknowledgement, a sender transmits *K* packets and each time an acknowledgement arrives, transmits another

Transport Protocols Used In The Internet

- Two primary transport protocols used in the Internet
  - User Datagram Protocol (UDP)
  - Transmission Control Protocol (TCP)
- Choice determined by application protocol
  - Many applications specify the use of a single transport (e.g., email transfer uses TCP)
  - Some applications allow the use of either (e.g., DNS queries can be sent via UDP or TCP)
- Recall: each transport protocol has some surprising characteristics

Message Transport With The User Datagram Protocol

User Datagram Protocol (UDP)

- Used
  - During startup
  - For VoIP and some video applications
- Accounts for less than 10% of Internet traffic
- Blocked by some ISPs

UDP Characteristics

- End-to-end
- Connectionless communication
- Message-oriented interface
- Best-effort semantics
- Arbitrary interaction
- Operating system independence
- No congestion or flow control

End-To-End Communication

- UDP provides communication among applications
- Sending UDP
  - Accepts outgoing message from application
  - Places message in a User Datagram
  - Encapsulates User Datagram in an IP datagram and sends
- Receiving UDP
  - Accepts incoming User Datagram from IP
  - Extracts message and delivers to receiving application
- Note: message is unchanged by the network

Connectionless Communication

- An application using UDP can
  - Send a message to any receiver (universal)
  - Send at any time (asynchronous)
  - Stop sending at any time (unterminated)
- That is, a sender does not
  - Inform the network before sending (i.e., does not establish a communication channel)
  - Inform the other endpoint before sending
  - Inform the network or other endpoint that no more messages will be sent

Message-Oriented Interface

- UDP
  - Accepts and delivers messages (blocks of data)
  - Does not require all messages to be the same size, but does define a maximum message size
  - Places each outgoing User Datagram in a single IP datagram for transmission
  - Always delivers a complete message to receiving application
- Sending application must divide outgoing data into messages; UDP sends what it is given (or reports an error if the message is too large)

UDP Message Size

- UDP allows up to 64K octet messages
- As a practical limit, the size of a User Datagram is limited by payload area in IP datagram
- Maximum IP payload is 64K octets minus size of IP header
- Therefore, the maximum UDP payload is 64K octets minus size of IP and UDP headers (usually 64K octets minus 28)
- Application can choose any message size up to the maximum UDP payload

Large And Small Messages

- What happens if an application sends a 10K octet message?
- The message fits into an IP datagram, but... network frames have a smaller MTU (typically 1500 octets)
- So, the result of sending a large message is

IP Fragmentation!

- What happens if an application chooses a small message size, such as 20 octets?

Inefficiency!

Choosing An Optimal Message Size

- What size messages should an application send?
- Optimal UDP message size is $S = M - H$
  - M is the path MTU (i.e., minimum MTU on the path)
  - H is the size of IP and UDP headers
- Finding M requires an application to
  - Violate layering and obtain forwarding information from IP
  - Note: for IPv4, only the local MTU is known
- Bottom line: it may be difficult/impossible for an application to compute S

UDP Semantics

- UDP uses IP for delivery and offers the same semantics!
- UDP packet can be
  - Lost
  - Duplicated
  - Delayed
  - Delivered out of order
  - Delivered with data bits altered
- Note 1: UDP does not introduce such errors; the errors arise from the underlying networks
- Note 2: UDP does include an *optional* checksum to protect the data (but the checksum may be disabled)

Using Best-Effort Semantics

- Questions
  - Do best-effort semantics make any sense for applications?
  - Why would a programmer choose UDP?
- Answers
  - Retransmitting a lost message does not make sense for real-time audio and video applications because a retransmitted packet arrives too late to be used
  - Additional real-time protocols can be added to UDP to handle out-of-order delivery (we will cover later in the course)

Arbitrary Interaction

- UDP permits arbitrary interaction among applications

  1-to-1

  1-to-many

  Many-to-1

  Many-to-many

- Application programmer chooses interaction type
- Ability to send a single message to multiple recipients can be valuable

Efficient Implementation Of Interaction

- Key point: UDP can use IP broadcast or multicast to deliver messages
- Provides efficient delivery to a set of hosts
- Example: UDP packet sent to IPv4 destination address *255.255.255.255* is delivered to all hosts on the local network (IPv6 has an *all nodes* multicast address)
- No need for sender to transmit individual copies
- Allows application to find a server without knowing the computer on which the server runs
- Broadcast is a significant advantage of UDP over TCP for some applications

Operating System Independence

- Goal is to allow applications on heterogeneous computers to interact
- Must avoid OS-specific identifiers, such as
  - Process IDs
  - Task names
- Instead, create application identifiers that are not derived from any OS

UDP Application Identifiers

- 16-bit integer known as *UDP protocol port number*
- Each application using UDP must obtain a port number
- Sending UDP
  - Places a port number in UDP header to identify destination application on receiving host
  - Also includes port number of sending application
- Receiving UDP
  - Uses value in header to select appropriate application

UDP protocol port numbers are universal across all computers, and do not depend on the operating system.

Identifying An Application

- Both sending and receiving applications need a port number
- Assignment of port numbers depends on the type of application
- Application that offers a standardized service (server)
  - Uses a *well-known port number* for the service
  - Value is less than 1024
  - Example: TFTP service uses UDP port 69
- Other applications (client)
  - Request a port number from the local operating system
  - Value is greater than 49151

Steps Taken To Contact A Service

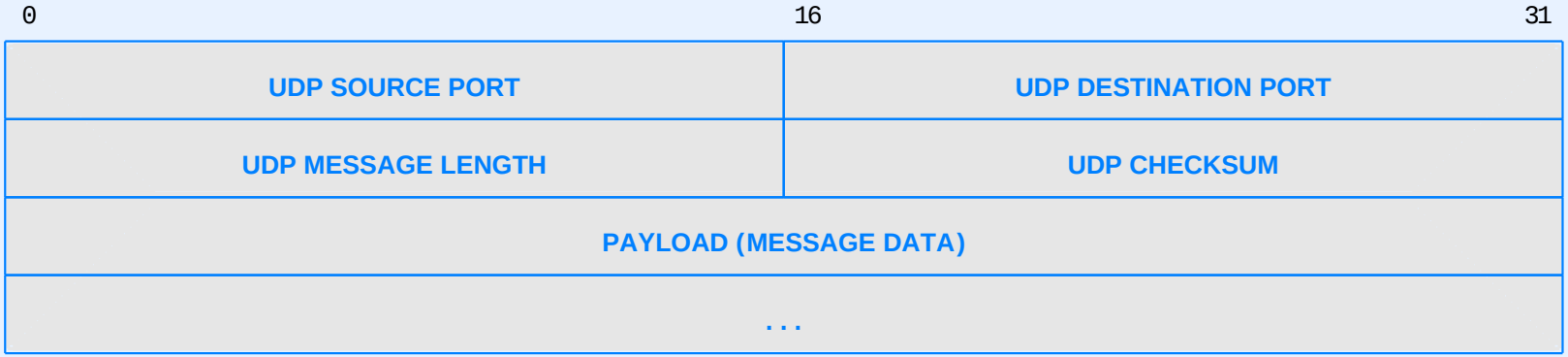
- Request an unused local port number from the local operating system
- Obtain the IP address of the local computer from the operating system
- Look up the port number of the service to be contacted
- Obtain the domain name of a computer that runs the service and map to an IP address
- Form a UDP datagram with a *source port* field set to the local port number and the *destination port* field set to the port number of the service
- Request that the UDP datagram be encapsulated in an IP datagram and sent using the source and destination IP addresses obtained above

Examples Of Well-Known UDP Ports

Port Number	Description
0	Reserved (never assigned)
7	Echo
9	Discard
11	Active Users
13	Daytime
15	Network Status Program
17	Quote of the Day
19	Character Generator
37	Time
42	Host Name Server
43	Who Is
53	Domain Name Server
67	BOOTP or DHCP Server
68	BOOTP or DHCP Client
69	Trivial File Transfer
88	Kerberos Security Service
111	Sun Remote Procedure Call
123	Network Time Protocol
161	Simple Network Management Protocol
162	SNMP Traps
514	System Log

UDP Datagram Format

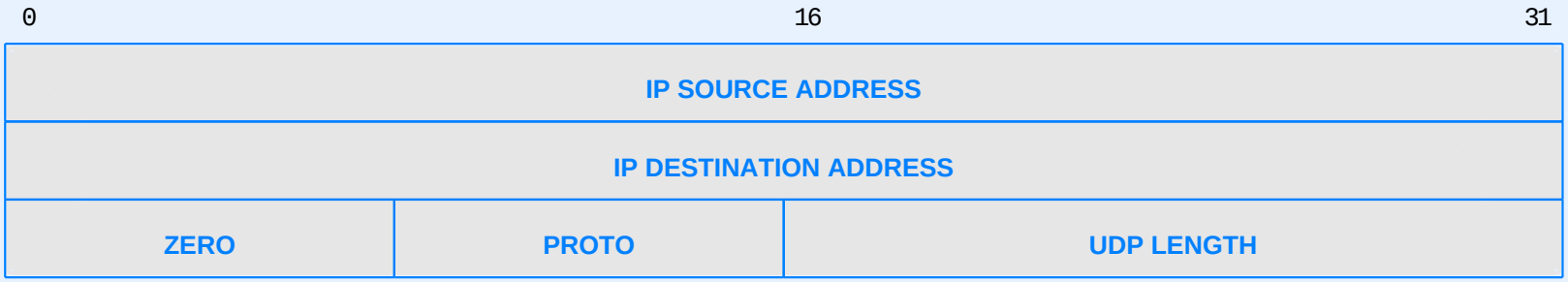
- Extremely thin layer
- User Datagram is divided into *header* and *payload*
- Header contains only 8 octets:



- Question: why is length needed?

UDP Checksum

- 16-bit 1s-complement checksum
- Covers entire UDP packet, including data (recall: IP does not checksum the payload)
- Is optional: value of zero means sender did not compute a checksum
- Includes extra *pseudo header* that contains IP addresses
- Example of IPv4 pseudo header:

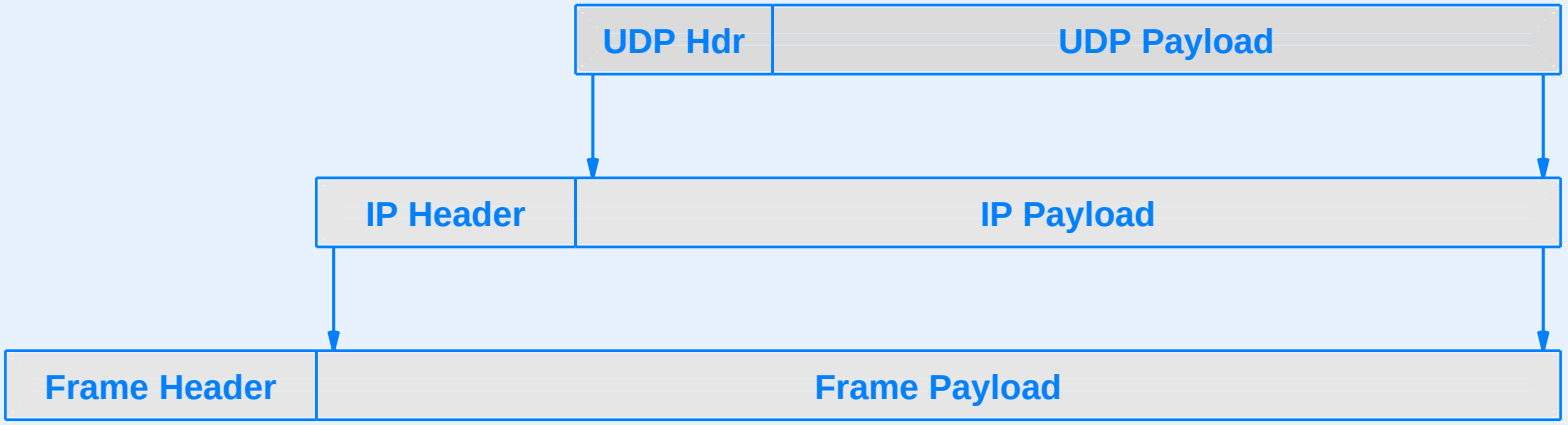


Purpose Of A Pseudo Header

- Receiver can verify that message arrived at correct computer as well as correct application on that computer
- Consequence for NAT: if it changes the IP source or destination address, NAT must recompute UDP checksum
- Note: pseudo headers provide another example of layering violations

UDP Encapsulation

- User Datagram travels in IP datagram
- Two levels of encapsulation occur



- Note: the message the application places in the UDP Payload field may also have header and payload fields

Transmission Control Protocol (Stream Transport)

Transmission Control Protocol (TCP)

- The primary transport-layer protocol used in the Internet
- Accounts for about 90% of all Internet traffic (some estimates are higher)
- Provides reliability
- Appeals to programmers

TCP Characteristics

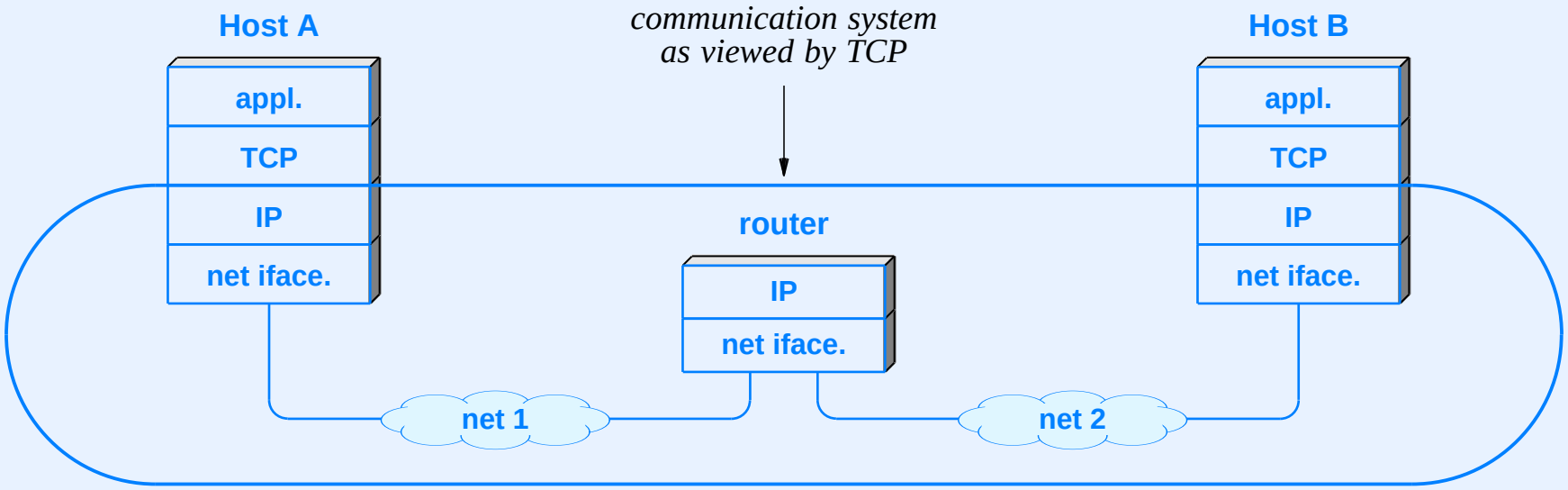
- End-to-end communication
- Connection-oriented paradigm
- Point-to-point connections
- Complete reliability
- Full-duplex communication
- Stream interface
- Reliable connection startup
- Graceful connection shutdown

End-To-End Communication

- TCP provides communication among pairs of applications
- Allows an application on one host to communicate with an application on another host
- Permits multiple applications on a given computer to communicate simultaneously without interference
- Uses *protocol port numbers* to distinguish among applications
- Note: TCP ports are completely independent of UDP ports

End-To-End Principle And Transport Protocols

- Transport protocols operate in end systems, and view the underlying Internet as a virtual network



- IP does not read or interpret TCP packets
- When forwarding datagrams, router only processes layers 1 through 3

TCP Protocol Port Numbers

- 16-bit integers used to identify applications
- Each application needs a port number
- TCP well-known port assignments are independent of UDP assignments
- However, to help humans, the same value chosen if service available via either transport
- Examples
  - Both UDP and TCP assign port 53 to the Domain Name System (DNS)
  - Both UDP and TCP assign port 7 to the echo service

Protocol Ports, The Four-Tuple, And Flows

- Key concept: because a TCP connection corresponds to a pair of endpoints, the connection is identified by four items
  - IP source address
  - TCP source port
  - IP destination address
  - TCP destination port
- Commonly called the *four-tuple*
- Explains how an application such as a web server can communicate with multiple clients at the same time
- Interestingly, more than four values must be extracted from a frame to identify a TCP flow

TCP's Connection-Oriented Paradigm

- Analogous to a telephone call
- Pair of applications must
  - Establish a TCP *connection* before communicating
  - Terminate the connection when finished
- Important insights
  - A TCP connection is *virtual* because only the two endpoints know a connection is in place
  - TCP does not have keep-alive messages: no packets are exchanged unless applications are sending data

Limited Interaction

- A TCP connection only provides communication between a pair of applications
- Known as a *point-to-point* communication
- TCP connection does *not* support
  - Reception from an arbitrary set senders
  - Multi-point connections with more than two endpoints
  - Broadcast or multicast delivery

The TCP Reliability Guarantee

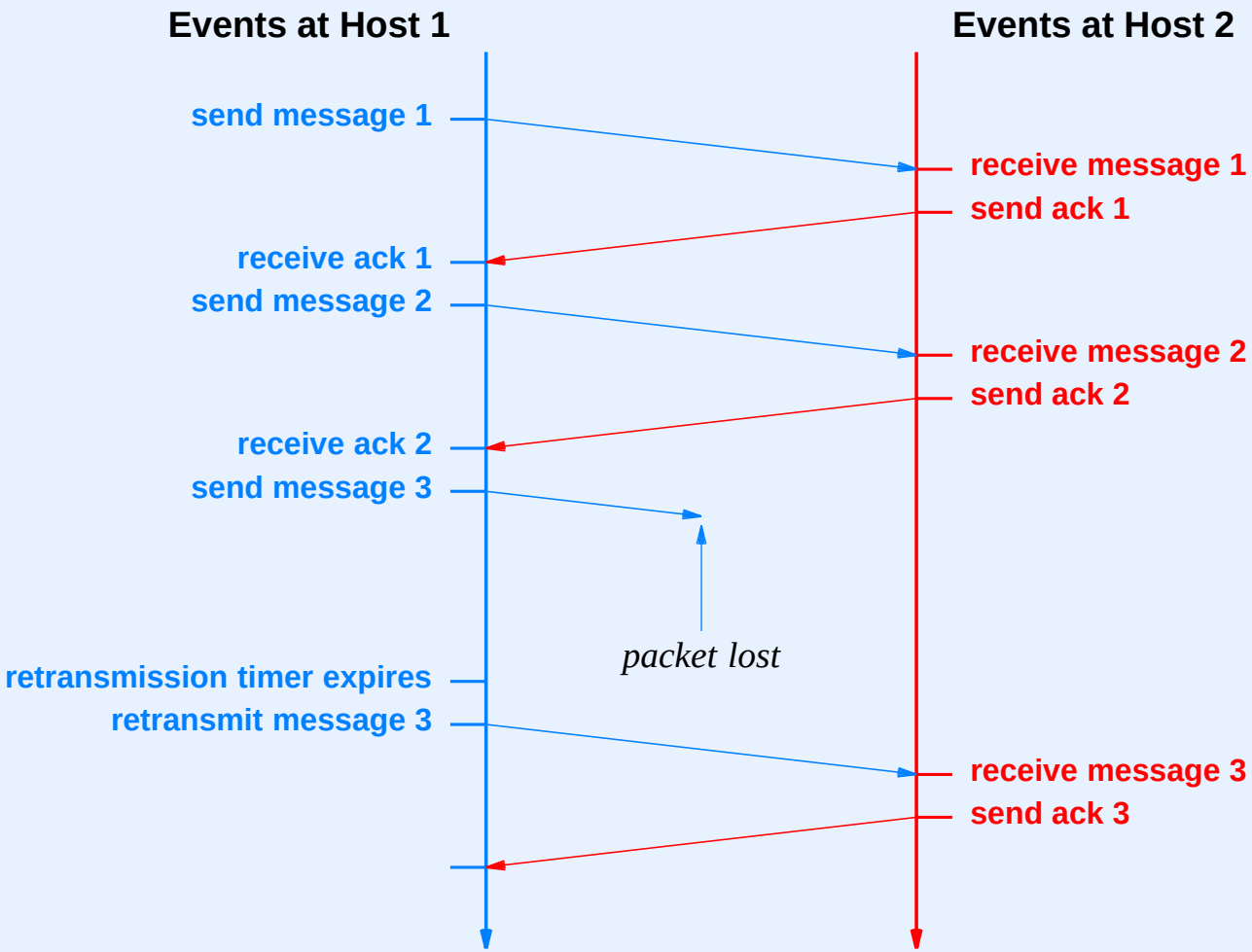
- TCP provides full reliability
- Compensates for
  - Loss
  - Duplication
  - Delivery out of order
- Does so without overloading the underlying networks and routers
- TCP makes the following guarantee

Data will be delivered or sender will (eventually) be notified.

TCP Reliability

- Uses timeout-and-retransmission
- Receiver returns an acknowledgement (ACK) to sender when data arrives
- Sender waits for acknowledgement and retransmits data if no acknowledgement arrives

Illustration Of TCP Retransmission

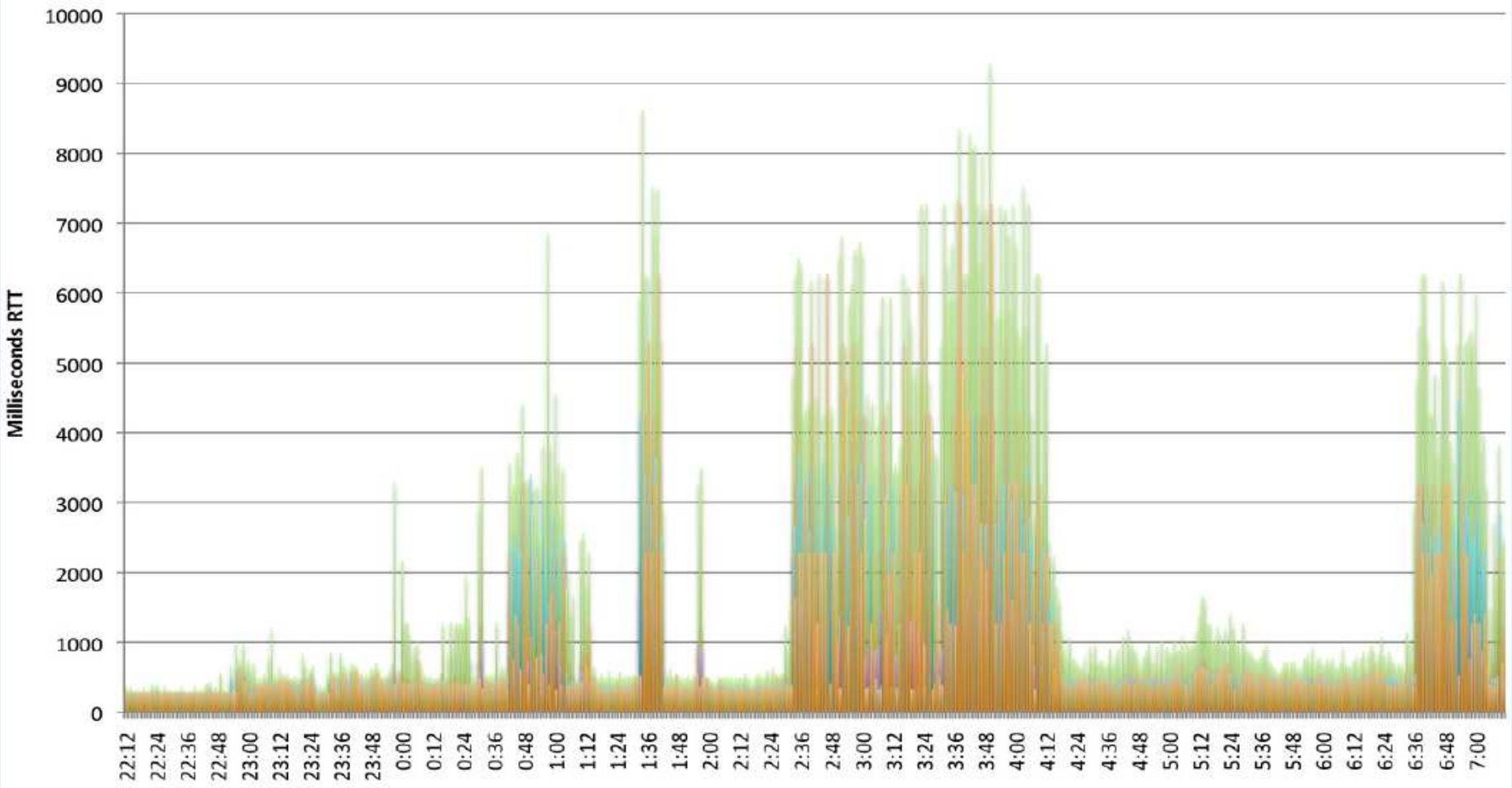


Why TCP Retransmission Is Hard

- TCP designed for Internet
  - Round-trip delays differ among connections
  - Round-trip delays vary over time
- Waiting too long introduces unnecessary delay
- Not waiting long enough sends unnecessary copies
- Key to TCP's success: *adaptive* retransmission

How Bad Is The Internet?

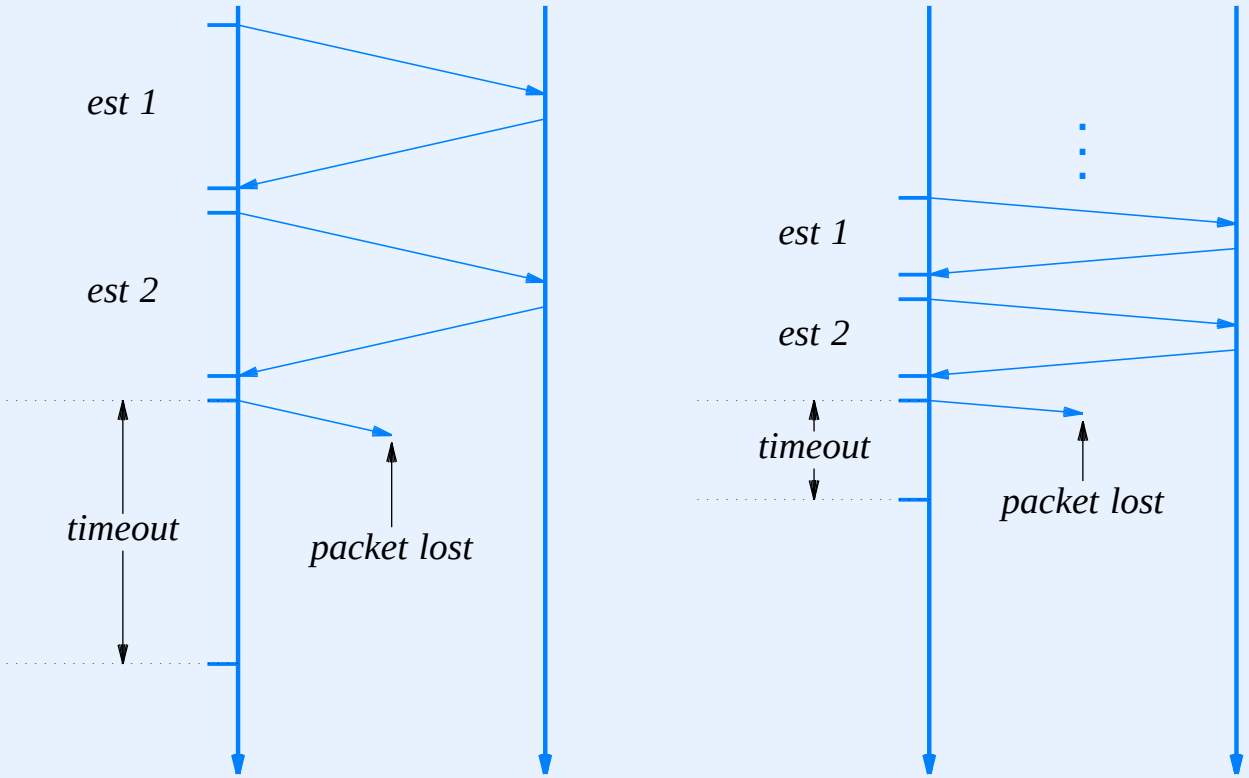
- In the old days: delays in seconds, high variability
- Now: delays in seconds, high variability



Example round-trip measurements from Ireland to California, 2009

Adaptive Retransmission

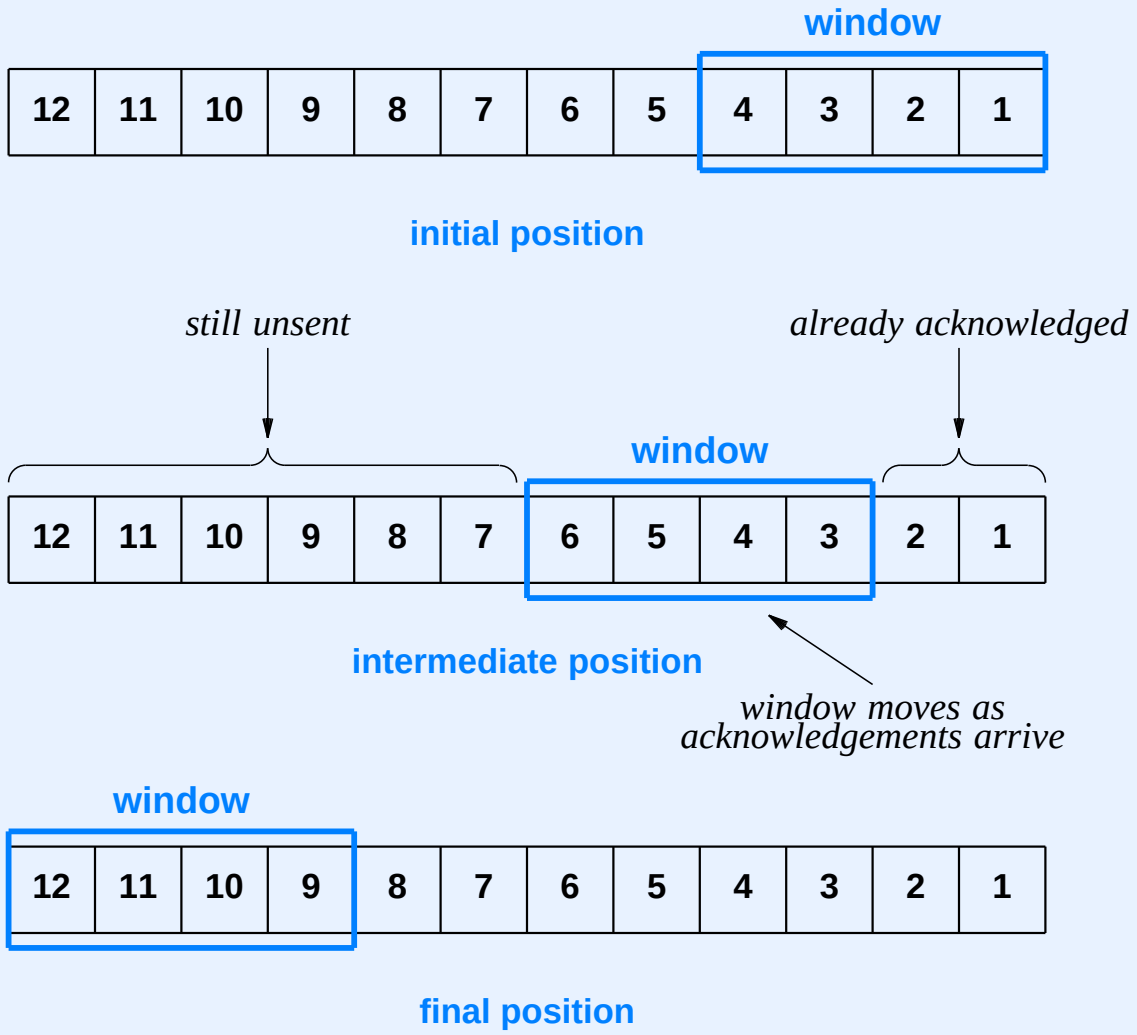
- Continually estimate round-trip time of each connection
- Set retransmission timer from round-trip estimate
- Illustration of timeout on two connections:



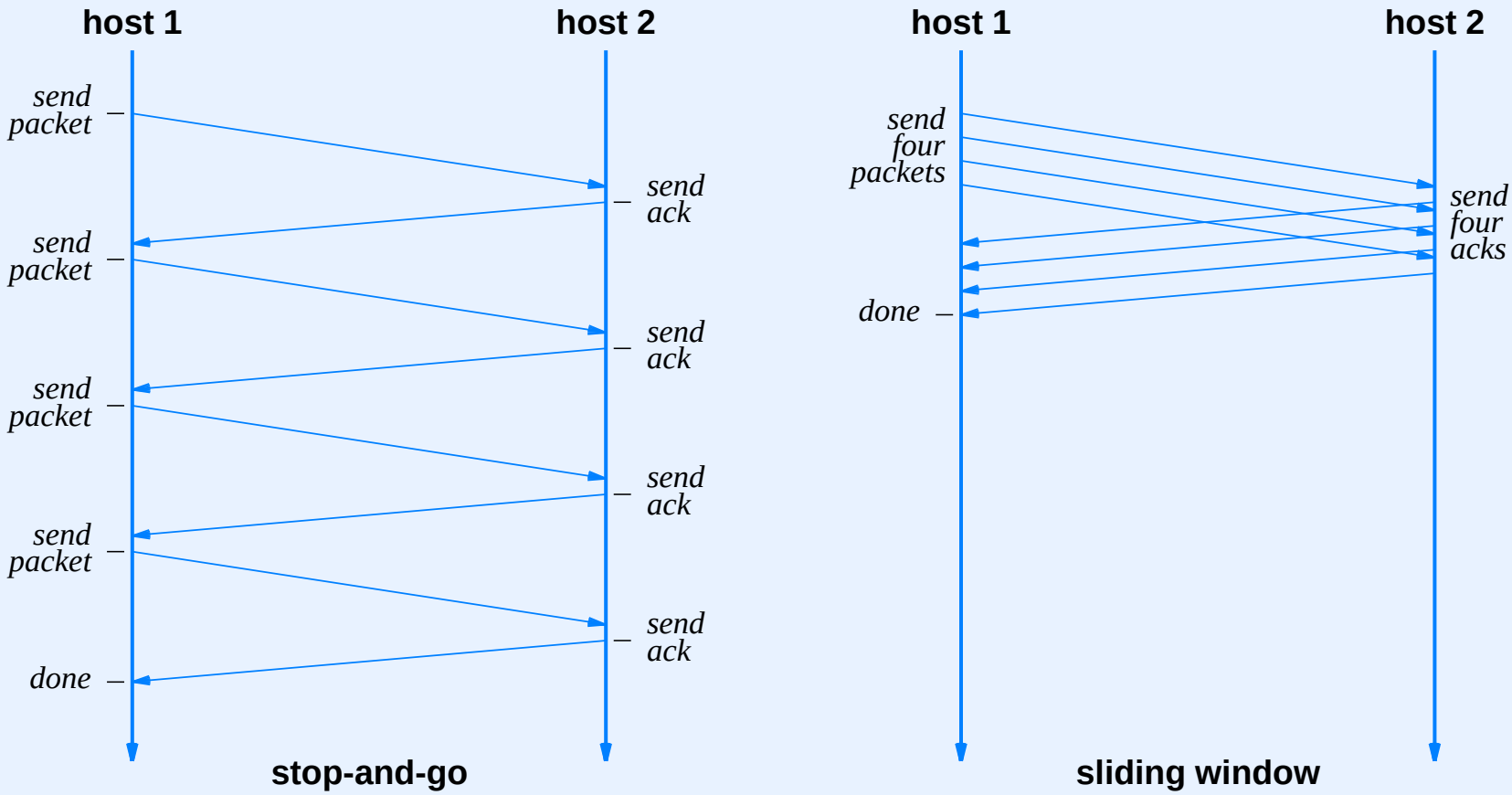
Review Of Sliding Window

- Transport protocols use *sliding window* mechanism
- Idea is to send multiple packets before waiting for an acknowledgment
- Window size is relatively small (tens of packets, not millions)
- Motivation is to increase throughput

Illustration Of TCP's Sliding Window



How Sliding Window Improves Data Rate

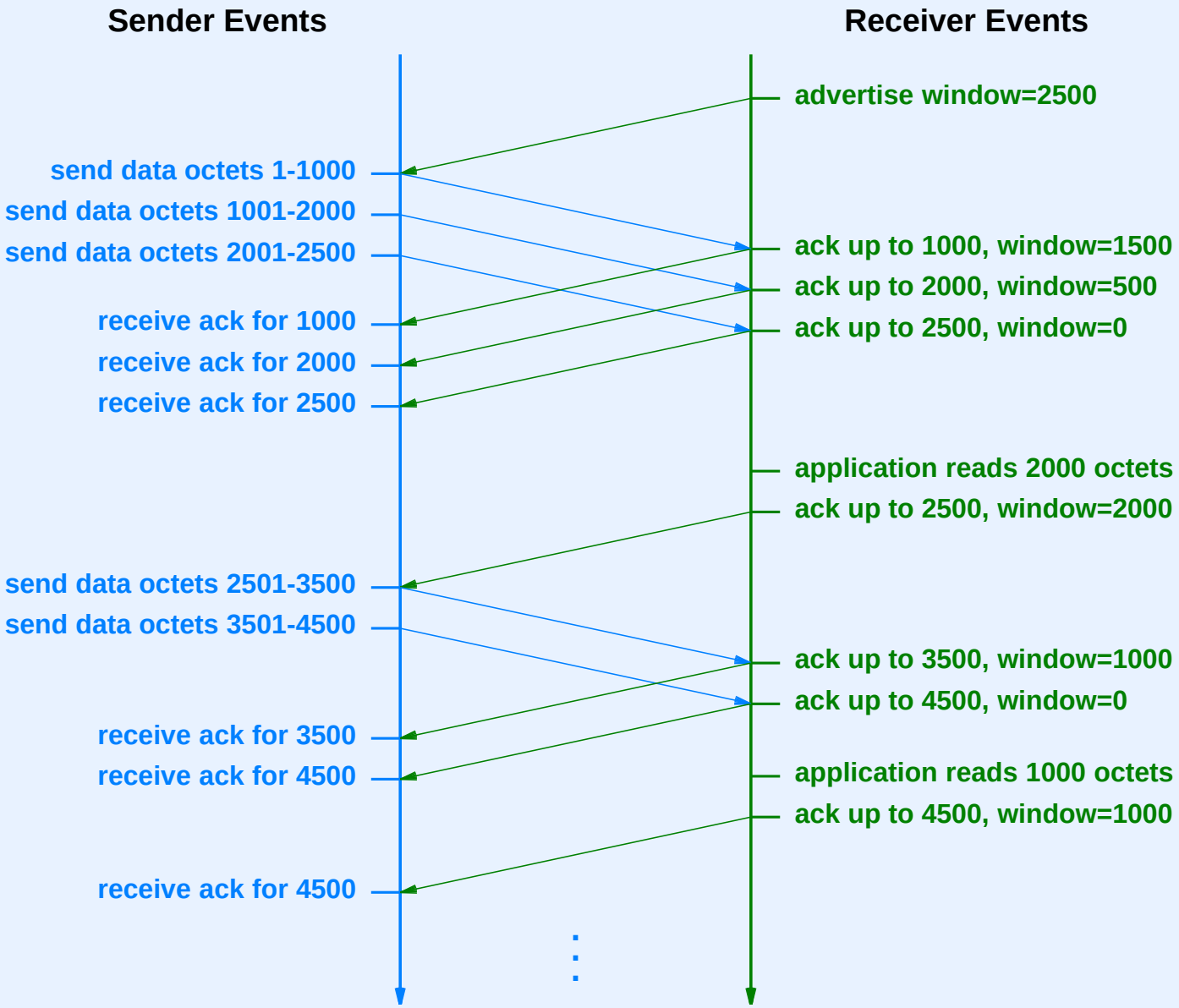


- Window size of *K* improves data rate by a factor of *K*

TCP Flow Control And TCP Window

- *Flow control* mechanism coordinates data being sent with receiver's speed
- Buffer size used instead of data rate
- Receiver tells sender size of initial buffer
- Each acknowledgement specifies space remaining in buffer
- Known as *window advertisement*

Illustration Of TCP Flow Control



TCP Congestion Control And Slow Start

- TCP uses loss or changes in delay to infer congestion in the network
- When congestion is detected, sending TCP temporarily reduces the size of the window
- When a packet is lost, TCP temporarily reduces the effective window to one half its current value
- Later, TCP slowly increases the window again
- Congestion avoidance also used when a connection starts
  - Temporarily use a window size of one segment
  - Double the window size when ACK arrives
  - Known as *slow start*

Full-Duplex Communication

- TCP connection between *A* and *B* provides two independent data streams, one from *A* to *B* and the other from *B* to *A*
- Each side
  - Has a receive buffer
  - Advertises a window size for incoming data
  - Uses sequence numbers to number outgoing data bytes
  - Implements timeout-and-retransmission for data it sends
- Application can choose to shut down communication in one direction

Full-Duplex Communication (continued)

- Each TCP packet contains fields for both forward and reverse data streams
  - Sequence number for data being sent in the forward direction
  - Acknowledgement number for data that has been received

Stream Interface

- After connection is established, TCP accepts a stream of data bytes from the sending application and transfers them
- Sending application can choose amount of data to pass on each request
- Surprise: TCP decides how to group bytes into packets
- Known as *stream* interface
- Consequence

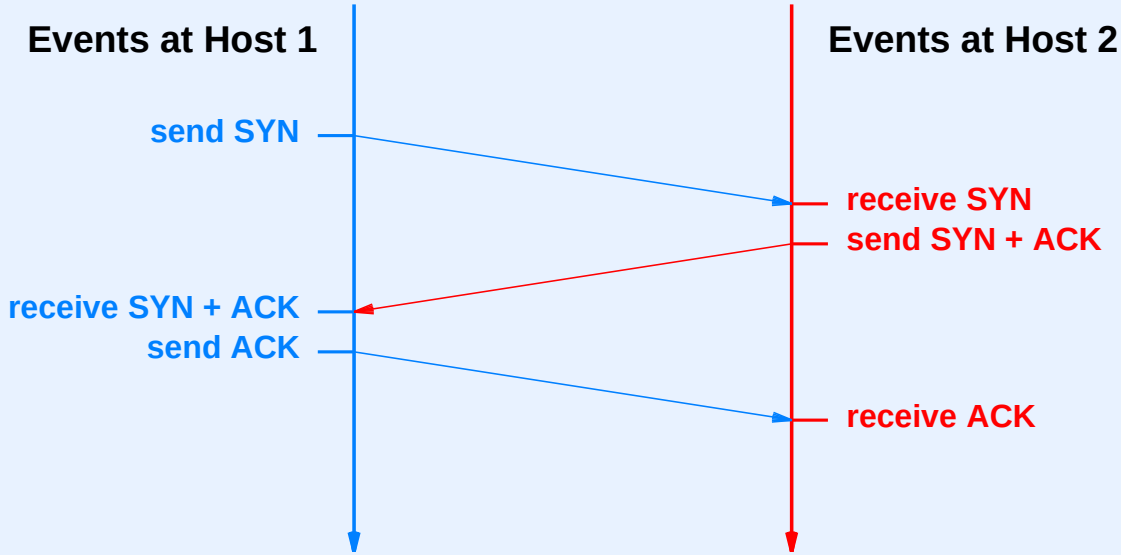
Data may be passed to a receiving application in chunks that differ from the chunks that the sending application generated.

Connection Startup And Shutdown

- Difficult problem
- Packets can be
  - Lost
  - Duplicated
  - Delayed
  - Delivered out-of-order
- Either end can crash and reboot
- Need to know that both sides have agreed to start/terminate the connection

Reliable Connection Startup

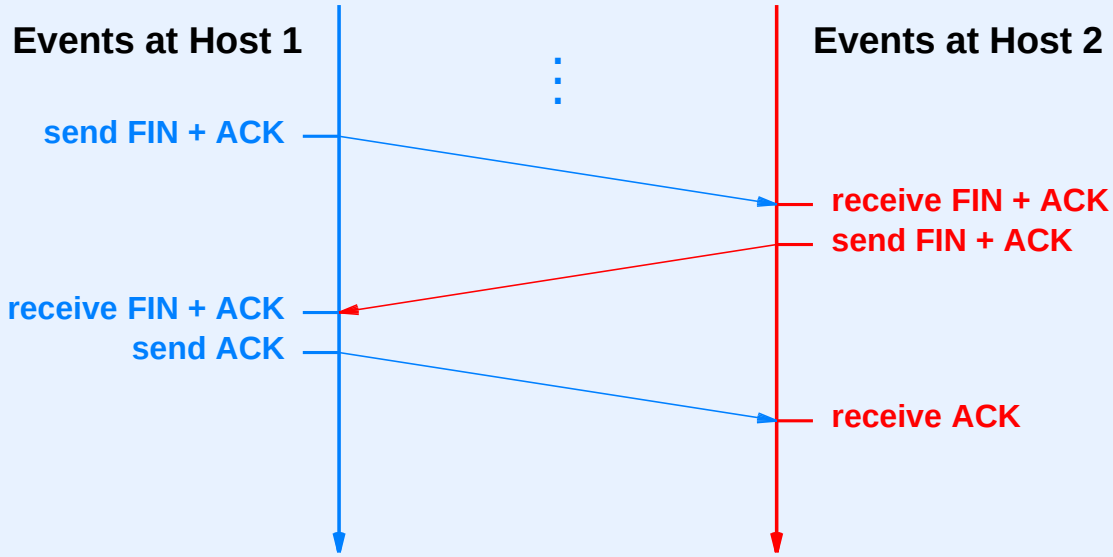
- TCP guarantees reliable connection startup that avoids *replay* problems
- Performed with 3-way handshake



- Each side chooses starting sequence number at random

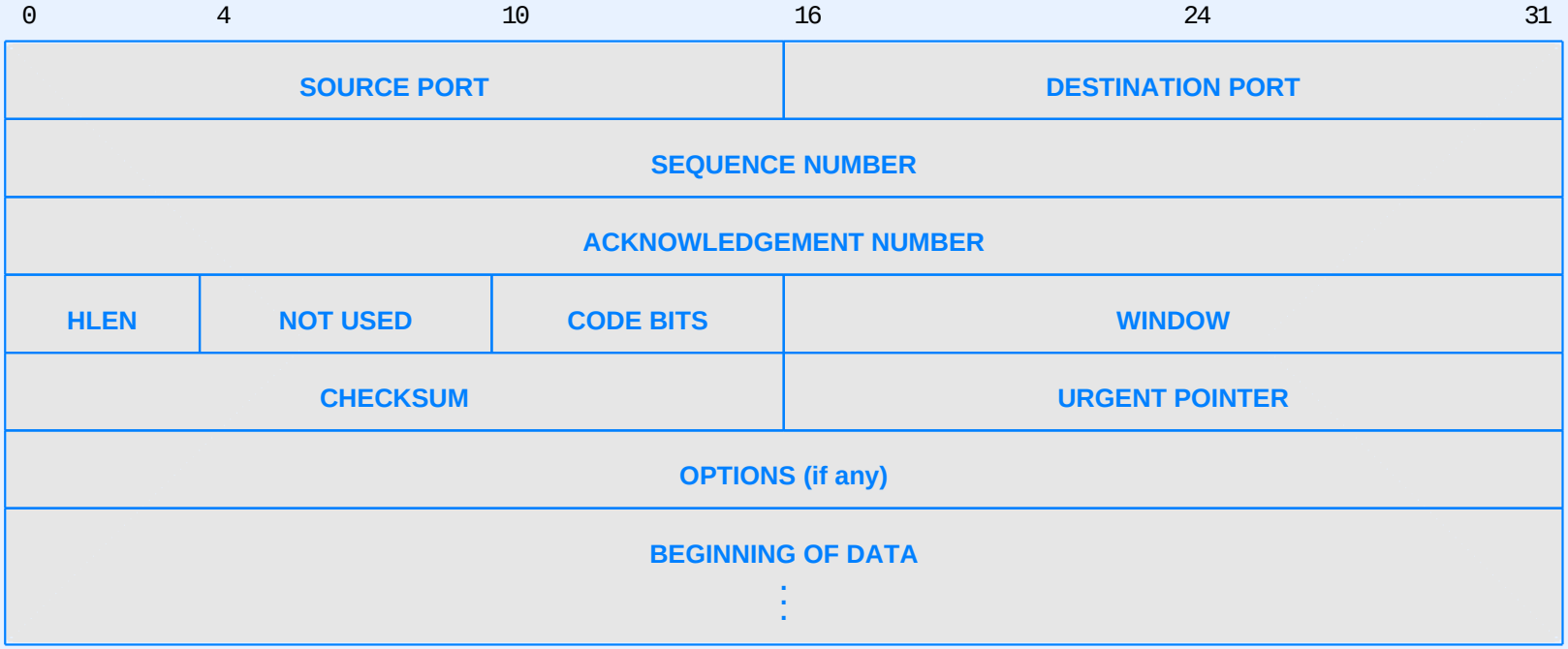
Graceful Connection Shutdown

- Analogous to 3-way handshake for startup
- Guarantees no ambiguity about connection termination



TCP Segment Format

- TCP packet is called a *segment*
- Segment is encapsulated in IP for transmission
- Single format used for SYNs, FINs, ACKs, and data



Routing Algorithms
And Routing Protocols

Historical Perspective

- Computing in the 1960s
  - Mainframes
  - Batch processing with punched cards
  - Usually one computer per organization
- Computing in the 1970s
  - Minicomputers
  - A few computers per organization
  - Dumb terminals

Traditional Wide Area Networks

- Developed during 1960s mainframe era
- Predate
  - LANs
  - PCs
- Basic motivation
  - Interconnect mainframe at one site to mainframes at other sites
  - Allow resource sharing
- First to employ dynamic routing

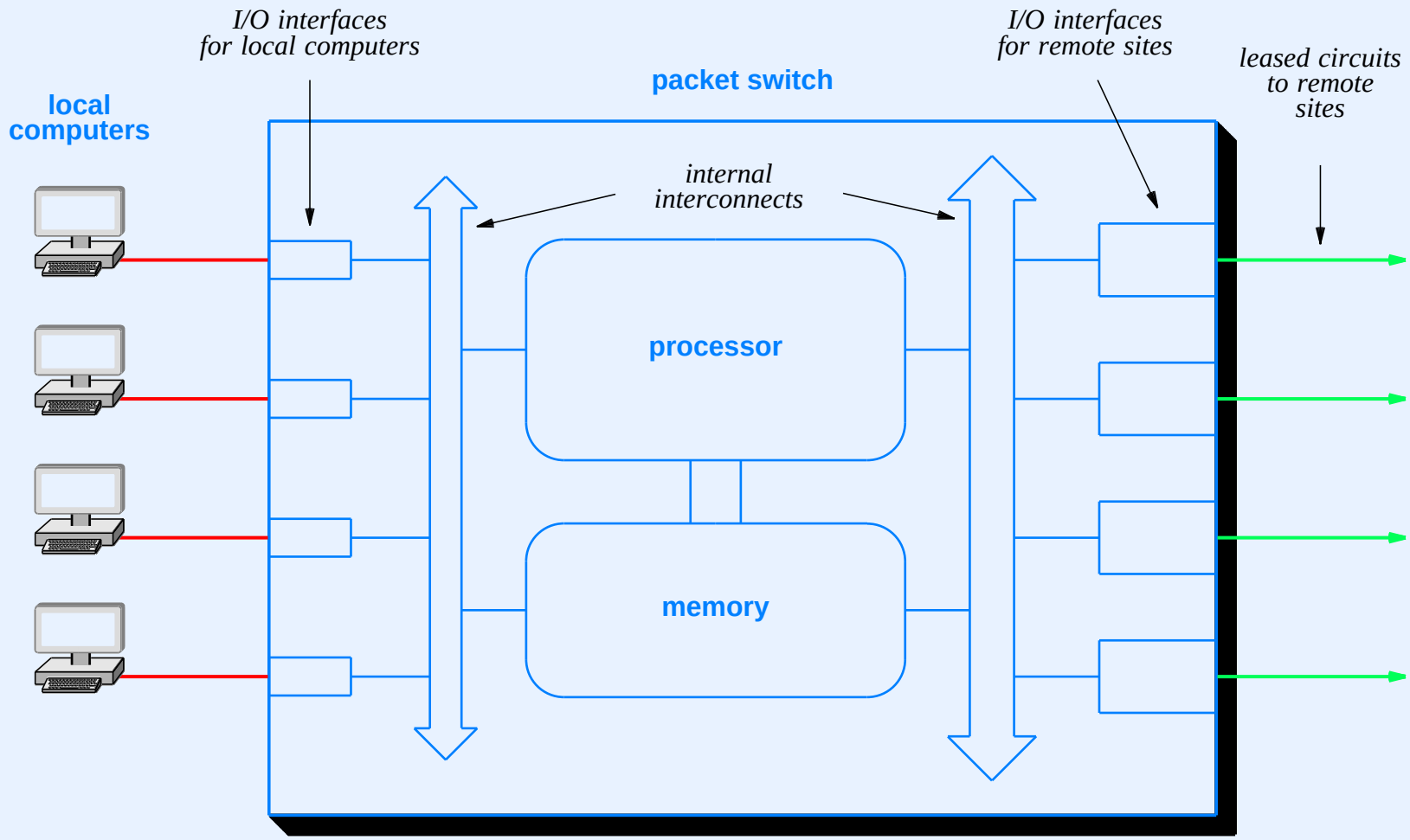
Traditional WAN Architecture

- Dedicated device known as *packet switch* placed at each site
- Packet switch provides
  - Local connections for host computer(s) at the site
  - Long-distance connections to other sites
- Connection among sites
  - Leased digital circuits
  - Leased raw copper or fiber with customer supplying modems

Packet Switch Used In Traditional WAN

- Special-purpose, stand-alone device
- Dedicated to packet forwarding
- Small computer with
  - Processor
  - Memory
  - Program on stable storage
  - I/O interfaces

Conceptual View Of Traditional Packet Switch



- Memory needed to store packets

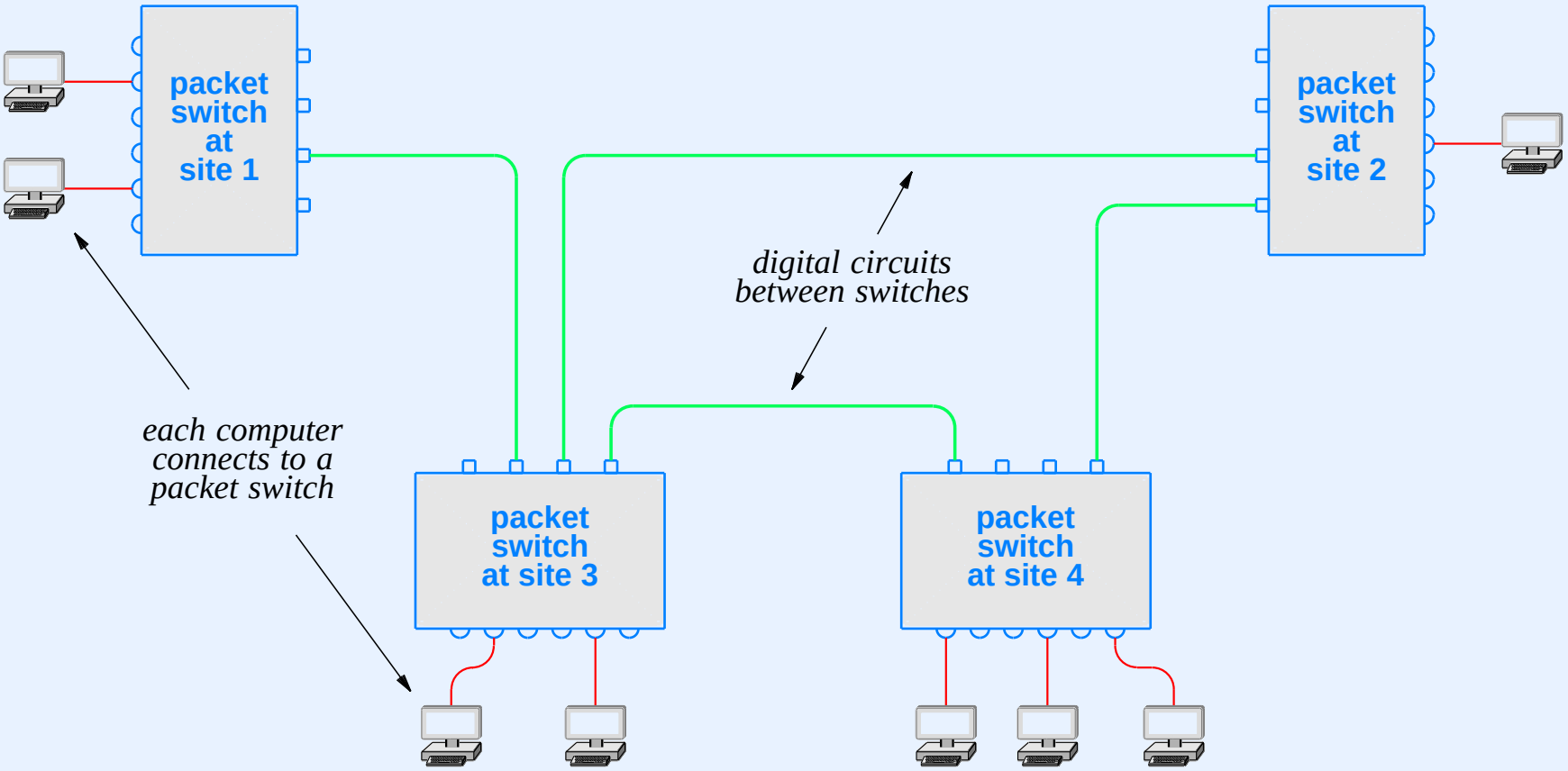
Store And Forward Paradigm

- Key paradigm used in packet switching
- Operation
  - Interface hardware places each arriving packet in a queue in memory
  - Processor continually removes next packet from the queue and forwards toward its destination
- Motivation: memory is a buffer that accommodates a short *burst* of packets that arrive back-to-back

Important point: packet traffic tends to be bursty.

Example Of Traditional WAN Architecture

- Packet switch at each site connects to other sites
- Circuits accommodate traffic and desired robustness



Traditional WAN Addressing

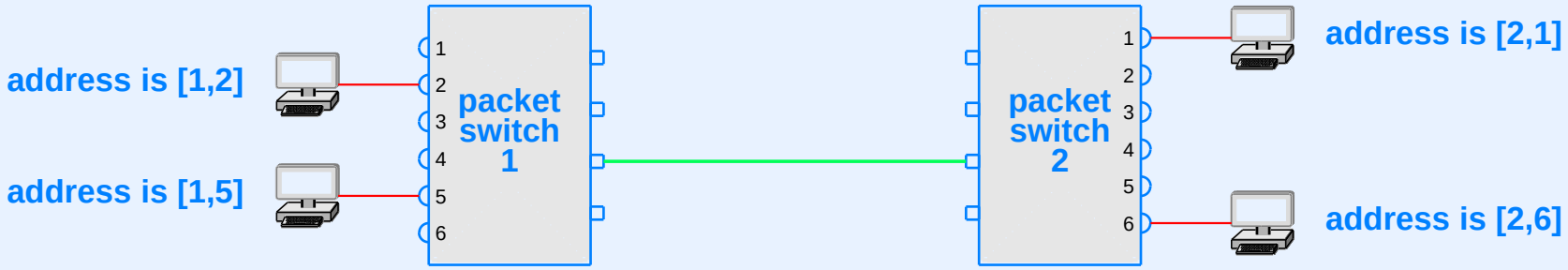
- Hierarchical model analogous to Internet addressing
- Conceptual two-level hierarchy

(site, computer at the site)

- In practice, one packet switch per site and K connections for local computers means the address hierarchy is:

(packet switch, local connection on the switch)

Illustration Of Traditional WAN Addressing



- The two parts of an address are combined to form a single binary number

Next-Hop Forwarding

- Analogous to IP datagram forwarding
- Each packet contains a destination address
- Forwarding uses only the packet switch portion of an address; delivery uses the rest of the address
- If packet has reached the destination packet switch, deliver to locally-connected computer
- Otherwise, forward to another packet switch that is closer to the destination site

Algorithm For Packet Forwarding

Given:

An incoming packet arriving at a packet switch

Perform:

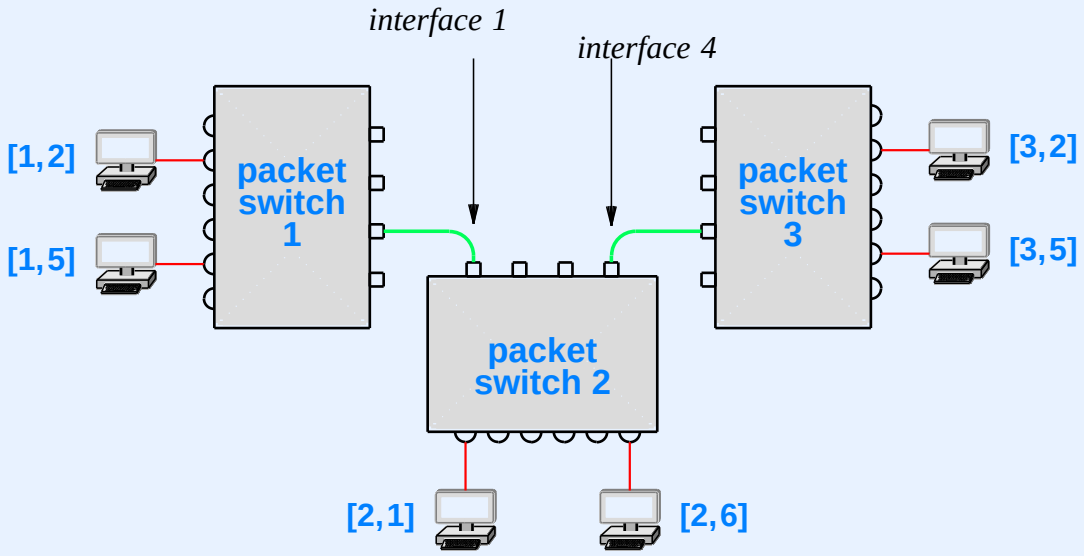
The next-hop forwarding step

Method:

```
Extract the destination address from the packet and
   divide into packet switch, P, and computer, C;
if (P is the same as "my" packet switch number) {
      Deliver the packet to local computer C;
} else {
      Use P to select a next hop, and forward the packet
        over the selected link to the next hop;
}
```

WAN Forwarding Table

- Analogous to IP forwarding table
- Each entry in table refers to a switch, not an individual computer



Example WAN with three packet switches

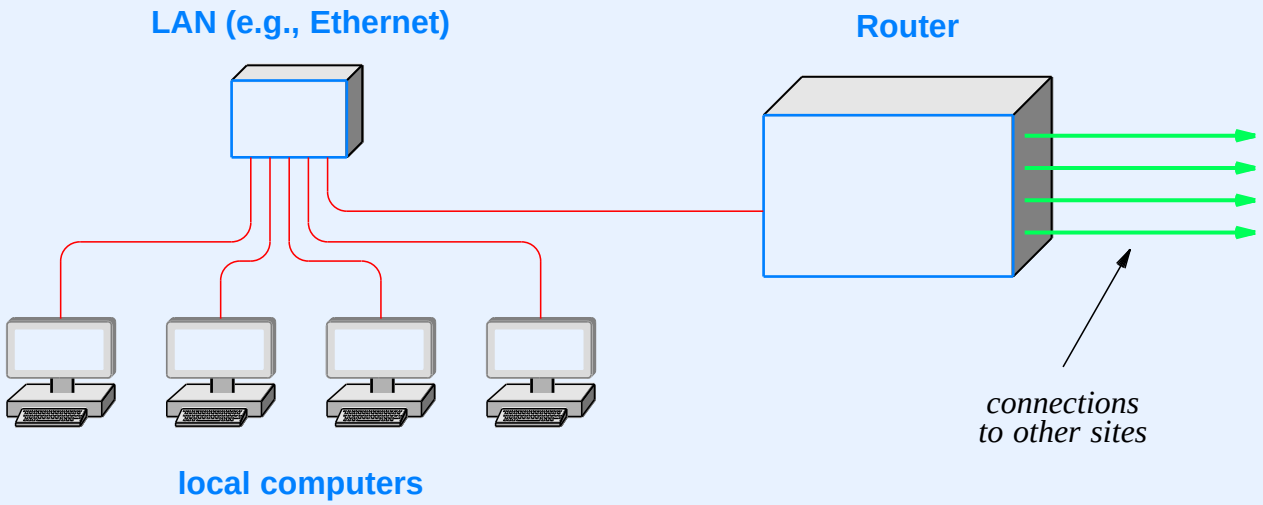
to reach	send to
switch 1	interface 1
switch 2	local delivery
switch 3	interface 4

Forwarding table for switch 2

Modern WAN Architecture

- Uses IP technology
- Router at site has
  - Local connections to networks at the site
  - Long-distance connections to routers at other sites
- Typical use: connect all sites of an organization

Illustration Of Modern WAN Connections



- Uses conventional IP router
- Typical remote connection is a leased data circuit
- Router can also provide connection to the Internet

Routing Algorithms
And Internet Routing

Constructing A Forwarding Table

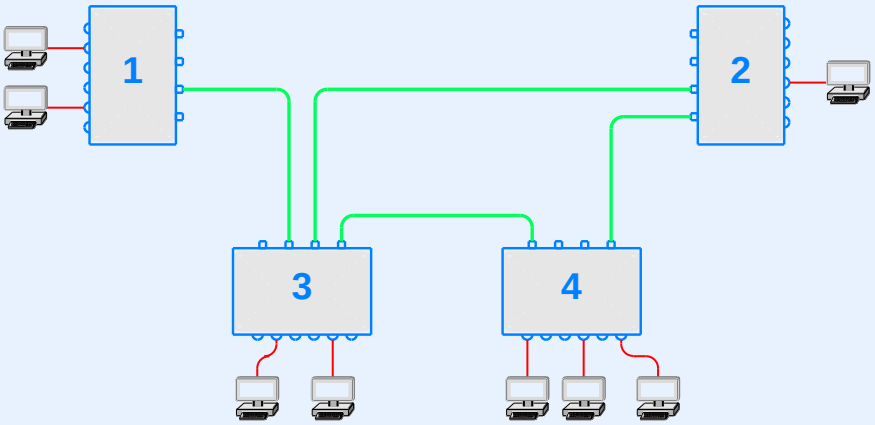
- Two basic approaches
- Static routing
  - Used in Internet hosts
  - Entries inserted when system boots and do not change
- Dynamic routing
  - Used in packet switches and IP routers
  - Initial entries inserted when system boots
  - Routing software continually monitors network, computes shortest paths, and updates forwarding table

Static Routing

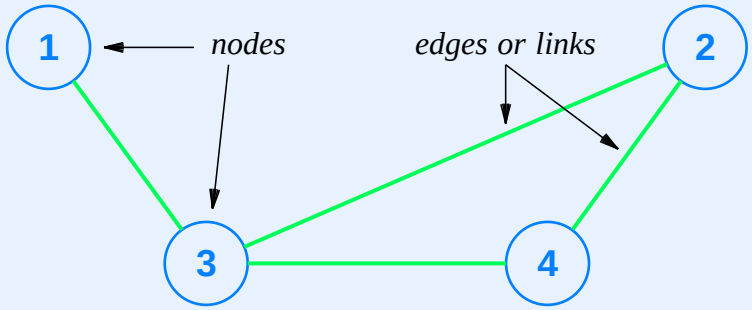
- Used in most hosts
- Only K+1 entries in forwarding table if host has K network connections
- K entries, one per network connection
  - IP prefix for the network
  - Address mask for the network
  - Interface for the network
- Final entry: default route
  - default IP router address
  - Interface for the default router

Dynamic Routing

- Routing Software
  - Runs on each packet switch or router
  - Computes shortest paths and installs entries in local forwarding table
- Models the network as a graph



Example WAN



Equivalent graph

Example Graph And Next-Hop Forwarding Tables

to reach	send over
1	–
2	(1,3)
3	(1,3)
4	(1,3)

node 1

to reach	send over
1	(2,3)
2	–
3	(2,3)
4	(2,4)

node 2

to reach	send over
1	(3,1)
2	(3,2)
3	–
4	(3,4)

node 3

to reach	send over
1	(4,3)
2	(4,2)
3	(4,3)
4	–

node 4

Dynamic Routing

- Goals
  - Consistent, optimal routes
  - Automatic route change to accommodate failures
- Each node (packet switch or router) participates
- Routing software on a node exchanges information with routing software on other nodes
- Distributed computation
- Two basic algorithms employed
  - Distance-Vector (DV)
  - Link-State Routing (LSR)

Distance-Vector (DV) Routing

- Approach used in many early routing protocols
- Also known as *Bellman Ford*
- Node
  - Receives information from neighbors
  - Combines information from all neighbors with local information
  - Sends copy of processed information to all neighbors

How DV Works

- A participant periodically sends *route advertisement* to each neighbor
- Advertisement specifies reachable sites and distance to each

  I can reach site X, and its distance from me is Y.

  I can reach site Z, and its distance from me is W.

  ⋮

- Neighbor receives advertisement and updates its forwarding table
- In next round, neighbors each send advertisements to their neighbors

Distance-Vector Algorithm

- Used when advertisement arrives
- Examine each item in advertisement
  - If neighbor can reach site X and I cannot, add an entry to my forwarding table for X with the neighbor as the next hop
  - If I already have a route to X with the neighbor as the next hop, replace the distance in the route with the advertised distance
  - If I have a route to X that is more expensive than going through the neighbor, change the next hop to the neighbor

Measuring The Distance Of A Route

- Possible measures
  - Hops
  - Delay
  - Throughput
  - Economic or administrative cost
- Many protocols use hops, but routing software often permits a manager to assign administrative hop counts

Link-State Routing (LSR)

- Chief alternative to distance-vector
- Each node
  - Sends link status information
  - Computes shortest paths independently
  - Does not rely on computation performed by others
- Name
  - Formal name is *Link-State* or *Link-Status Routing*
  - Also called *Shortest Path First* (*SPF*), a somewhat misleading term derived from underlying algorithm

How LSR Works

- Each pair of directly-connected nodes periodically
  - Tests connection between them
  - Broadcasts one of the following messages:

    The link between X and Y is up.

    or

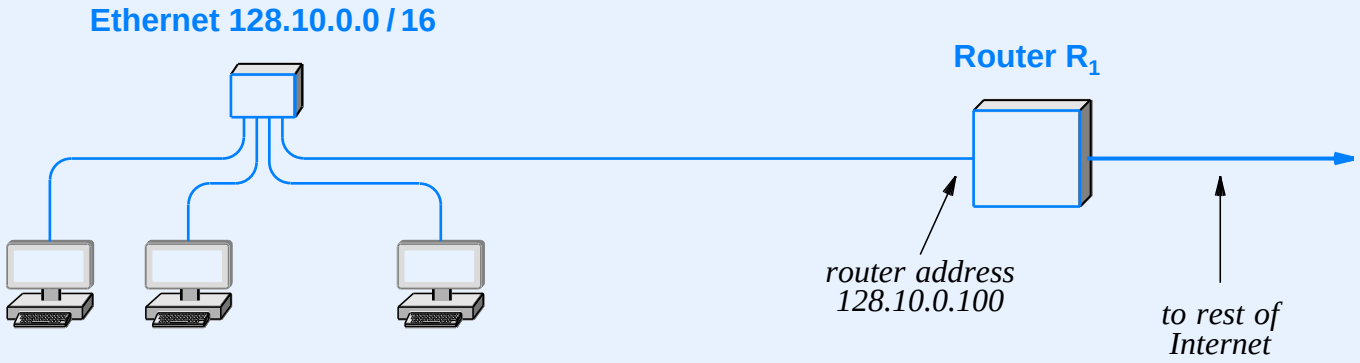
    The link between X and Y is down.

- Each node
  - Collects incoming broadcast messages and creates a graph
  - Uses Dijkstra's SPF algorithm to compute a forwarding table (see text for details and example)

Review Of Internet Forwarding

- Hosts
  - Use static routing
  - Entries placed in forwarding table when system boots and remain unchanged
- Routers
  - Use dynamic routing
  - Initial entries placed in forwarding table when system boots and routing software updates entries continually

Example Of Host Routing



hosts on a network

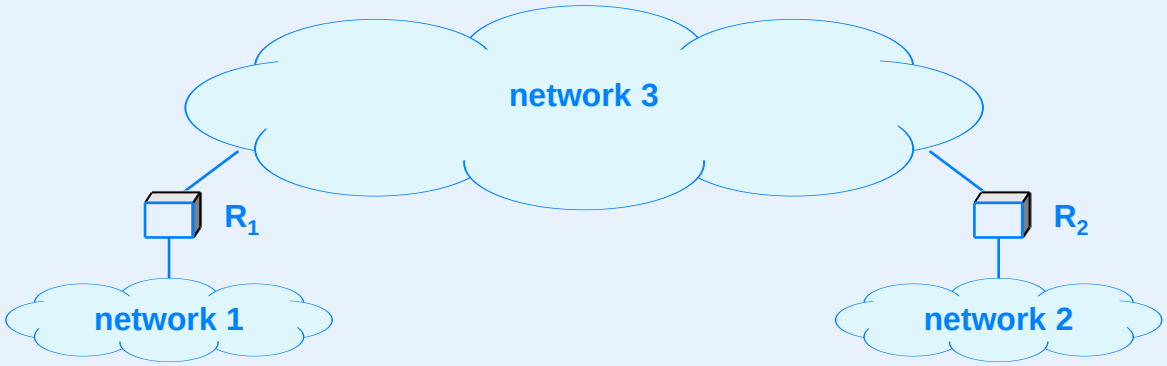
Net	Mask	Next hop
128.10.0.0	255.255.0.0	direct
default	0.0.0.0	128.10.0.100

forwarding table in each host

- Next hop in default route is known as a *default router*

Why Dynamic Internet Routing Is Needed

- Router
  - Only has direct connections to a few networks
  - Must know how to forward datagram to arbitrary destination
- Example



- Router R_1 must learn about network 2 and R_2 must learn about network 1

Important Principle

No single routing protocol can be used across the entire Internet because the overhead is too high.

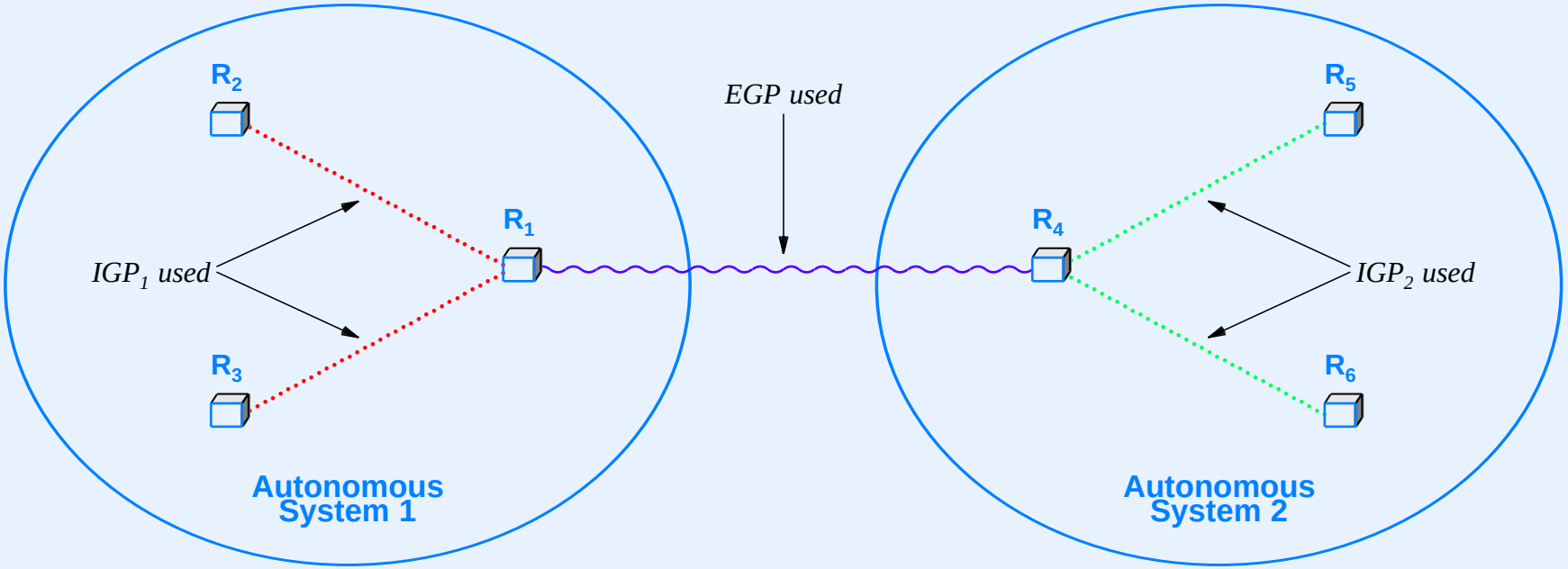
Autonomous System Concept

- Internet divided into a set of routing domains
- Each routing domain is
  - Known as an *autonomous system* (AS)
  - Assigned a unique number
- Generally, an AS is a contiguous set of routers and networks under one *administrative authority*
- No exact definition; think of a large ISP or a large corporation
- AS gathers and summarizes routing information before passing it to another AS

Two Types Of Internet Routing Protocols

- Interior Gateway Protocols (IGPs)
  - Used *within* an autonomous system
  - Choice of IGP is made by each AS
  - Relatively easy to install and manage
- Exterior Gateway Protocols (EGPs)
  - Used *between* autonomous systems
  - More complex to install and configure
  - Include policy constraints that control which information is revealed

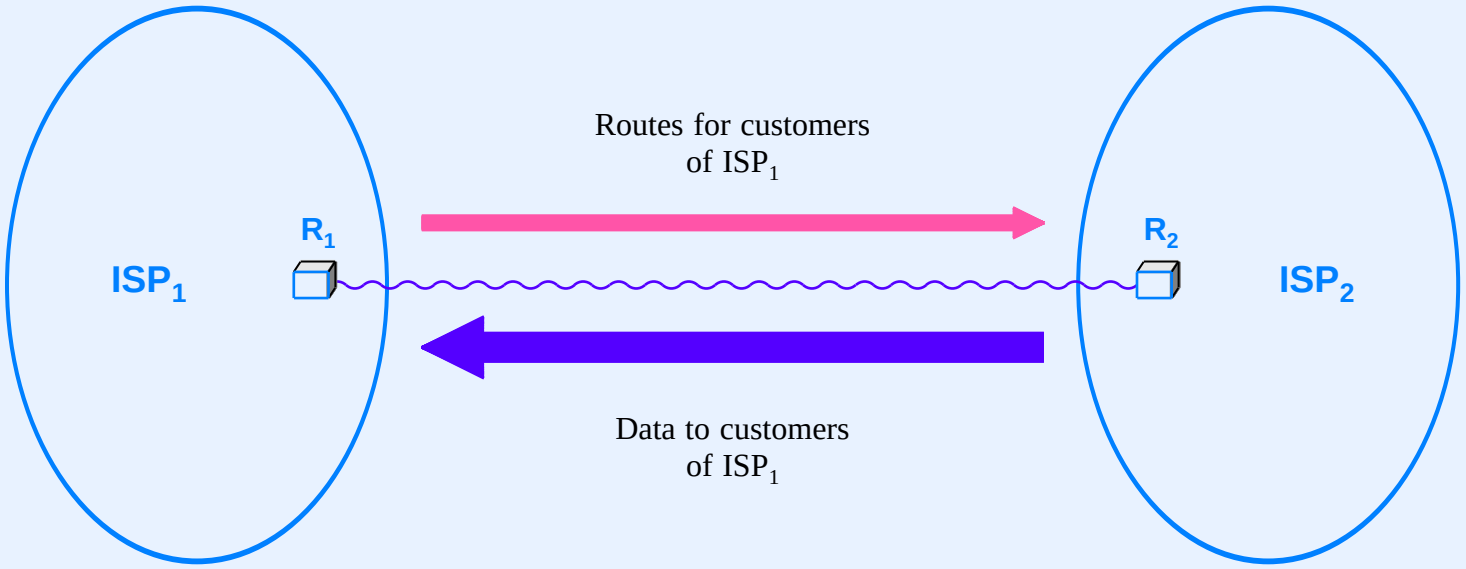
Illustration Of IGPs and EGPs



- Because metrics used in each AS may differ, direct comparison is impossible

Principle Of Route And Data Flow

- Data flows in opposite direction of routes
- Example: ISP_1 advertises route to customer Q and receives traffic for customer Q



Internet Routing Protocols

Border Gateway Protocol (BGP)

- Primary Exterior Gateway Protocol used in the Internet
- Used by Tier 1 ISPs at the center of the Internet
- Current version is 4 (BGP-4)
- Characteristics
  - Provides routing among autonomous systems
  - Includes provisions for policies
  - Distinguishes *transit* routes from *terminal* routes
  - Uses reliable transport (TCP)
  - Sends *path* information

Illustration Of BGP Paths

- Modified Distance-Vector protocol
- Advertisement contains a *path* in place of a distance
- Path lists the autonomous systems to destination
- Example

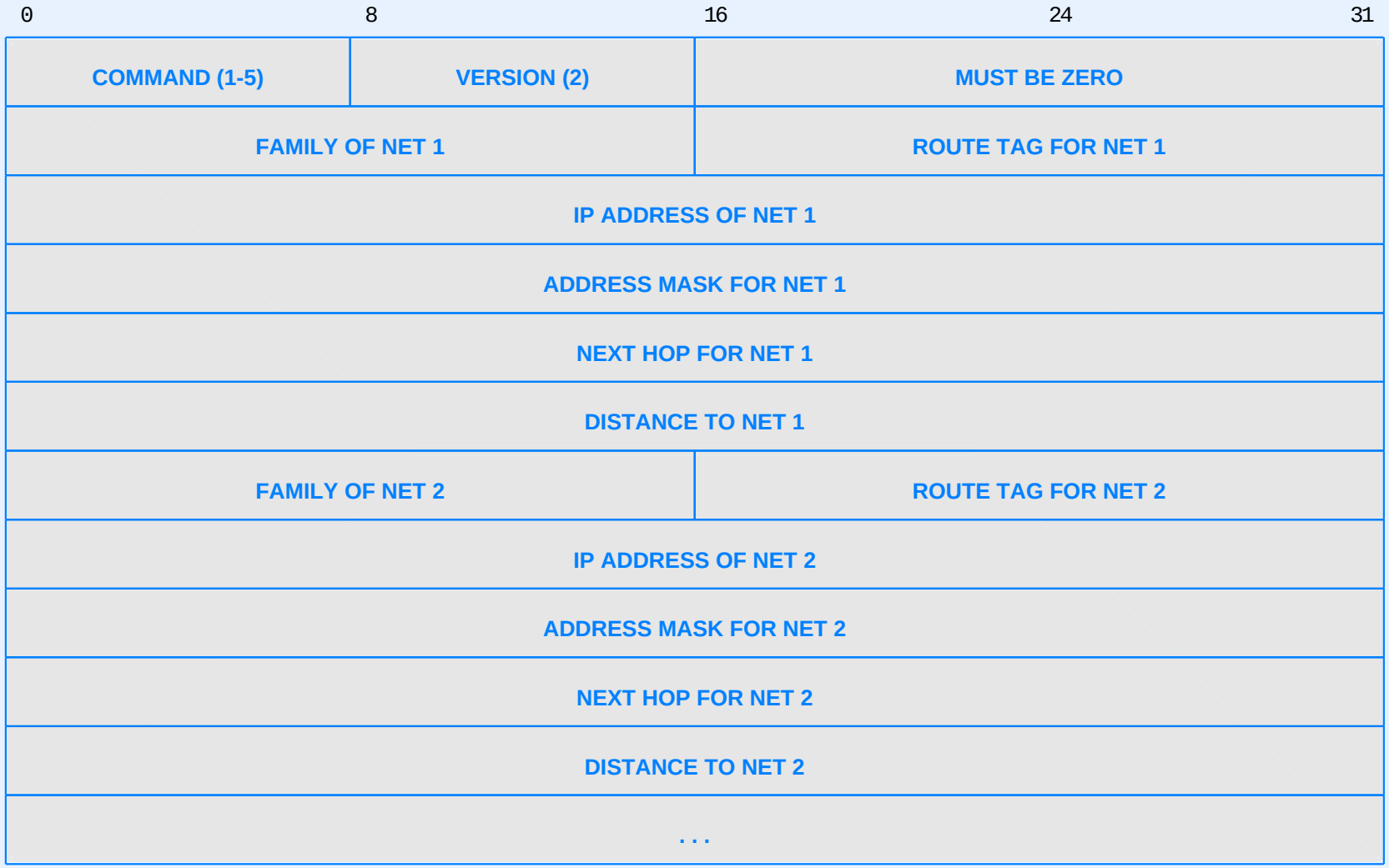
To reach network X, I send along path Z, Y, W,...

- Path information means receiver can apply policies (e.g., receiver can choose to ignore all routes that pass through AS number N)

Routing Information Protocol (RIP)

- Among the earliest Interior Gateway Protocols
- Characteristics
  - Distance-Vector that uses hop-count metric
  - Sent over UDP (unreliable transport)
  - Advertises CIDR prefixes
  - Includes facility for *default route* propagation
  - Broadcast or multicast delivery
- Current version is 2 (RIP2)

RIP2 Packet Format

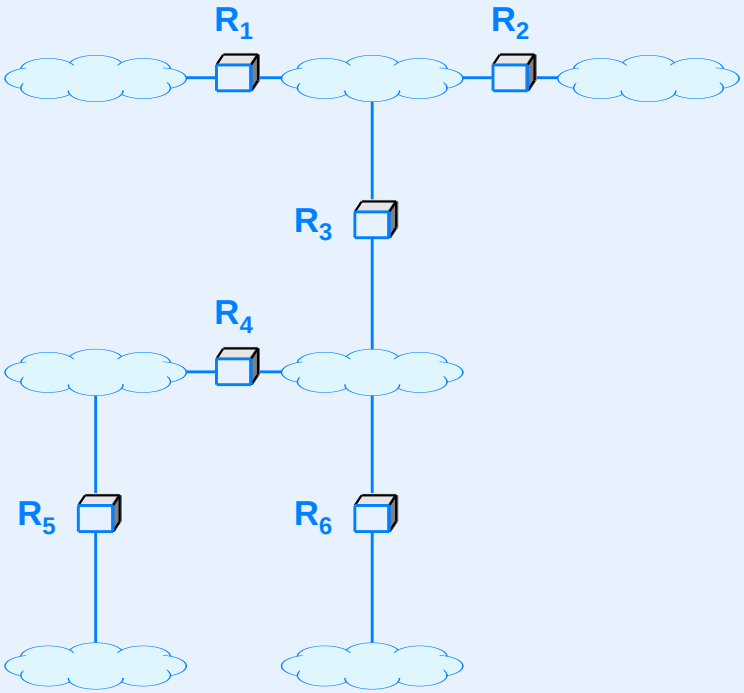


- Note: routing protocols run at application layer (layer 5)

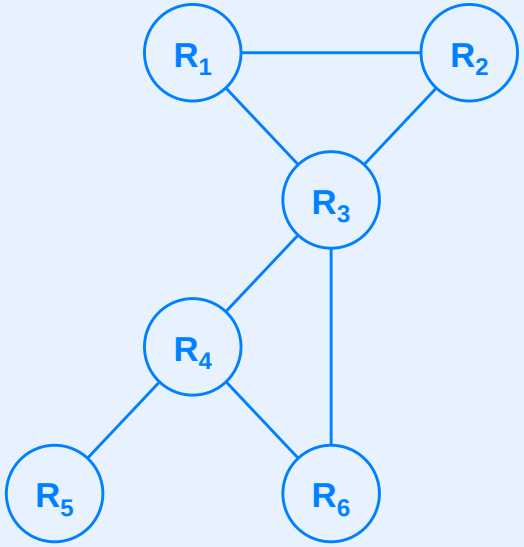
Open Shortest Path First Protocol (OSPF)

- Created by the IETF to be an open standard (reaction to proprietary protocols)
- Characteristics
  - Interior Gateway Protocol
  - Advertises CIDR prefixes
  - Authenticated message exchange
  - Can import routes from BGP
  - Link-state algorithm
  - Provides for multi-access networks
  - Divides large network into *areas*

Illustration Of An OSPF Graph



a network



the OSPF graph

- Graph shows a link between each pair of routers even though some connections cross a shared network

Intermediate System - Intermediate System (IS-IS)

- Originally part of DECNET V protocols
- Uses LSR approach
- Initially
  - Considered somewhat over featured
  - Not widely accepted in the Internet
  - Overshadowed by OSPF
- Eventually
  - OSPF became complex as features were added
  - IS-IS started to gain acceptance

Routing Problems

Where Intuition Fails

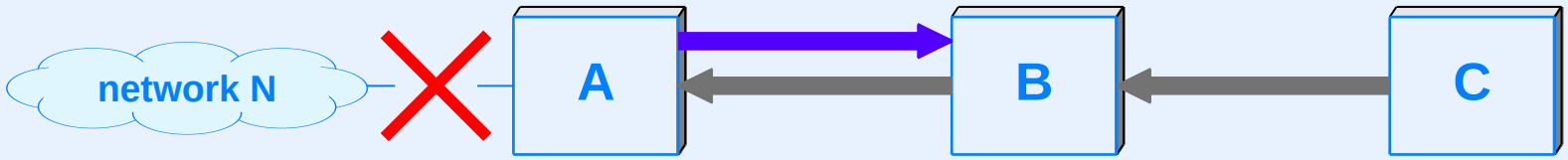
- Routing is *not* like water flowing through pipes or traffic on highways
  - Multi-path routing is difficult
  - Capacity can go unused if not along shortest path
- Fewest hops may not always be best
  - Compare two Ethernet hops and one satellite hop
- Routing around congestion is *not* straightforward, and does *not* always yield a big improvement
  - Can cause out-of-order packets (TCP reacts)
  - Can result in route flapping

Loops And Convergence

- Routing loop
  - Circular routes
  - Can be caused if “good news” flows backward
- Slow convergence (count to infinity) problem arises
  - Routes fail to converge after a change
  - Can cause a routing loop to persist

How Good News Can Backwash

- A story with three routers and a network



- In practice, modern DV protocols employ heuristics that
  - Eliminate backflow
  - Lock down changes after a failure

Other Routing Problems

- Black hole
  - Routing system sends packets for a set of destinations to a location where they are silently discarded
  - Can be caused if routing update packets are lost
- Route flapping (lack of convergence)
  - Routes continue to oscillate
  - Can be caused by equal-length paths

Routing Overhead

- Traffic from routing protocols is “overhead”
- Specific cases
  - DV advertisements tend to be large
  - LSR uses broadcast
- Fundamental tradeoff
  - Decreasing frequency of routing exchanges lowers overhead
  - Increasing frequency of routing exchanges reduces the time between a failure and rerouting around the failure

Internet Multicast
And Multicast Routing

IPv4 Multicast

- Defined early; informally called “Deering multicast”
- Provides Internet-wide multicast dissemination
- Uses IPv4 addresses 224.0.0.0 through 239.255.255.255 (the original Class D address space)
- In theory, any host in the Internet can
  - Join or leave any group at any time
  - Send a datagram to any group at any time

Internet-wide multicast is not widely deployed

IPv6 Multicast

- Fundamental part of IPv6
- IPv6 prohibits broadcast, but defines multicast groups that are equivalent
  - All routers
  - All nodes

Internet Group Multicast Protocol (IGMP)

- Allows a host to join or leave a multicast group
- Restricted to a single network (host talks to local router)
- When first host on a network joins a new group or last host on a network leaves a group, router(s) on the network change multicast routes accordingly

IP Multicast And Ethernet Delivery

- When sending IP multicast across Ethernet
  - Can use Ethernet multicast capability
  - IP multicast address is mapped to an Ethernet multicast address
- Problem
  - Most interface hardware limits the number of Ethernet multicast addresses that can be used simultaneously
  - Trick: use a few multicast addresses and allow software to decide how a given packet should be processed

Multicast Routing Protocols

- Needed to propagate multicast routes throughout the Internet
- Goals
  - Ensure all participants in a group receive packets sent to the group
  - Avoid flooding multicast across a network unless a host is listening
- General approach
  - Form a graph-theoretic tree for each multicast group
  - Forward multicast along links of the tree
- Trick: send a request for group X toward the “center” of the Internet until it reaches a router that knows about group X

Example Multicast Routing Protocols

- Many multicast routing protocols have been proposed
- A few examples

Protocol	Type
DVMRP	Configuration-and-Tunneling
CBT	Core-Based-Discovery
PIM-SM	Core-Based-Discovery
PIM-DM	Flood-And-Prune
MOSPF	Link-State (within an organization)

Summary

- Internet
  - Consists of a network of heterogeneous networks
  - Separates communication from content and services
  - Accommodates arbitrary network technologies and applications
- IPv4 uses 32-bit addresses; IPv6 uses 128-bit addresses
- Internet packet is known as an *IP datagram*
- Datagram is encapsulated for transmission
- Fragmentation and reassembly accommodate heterogeneous MTUs

Summary
(continued)

- IPv4 uses ARP for address resolution and IPv6 uses ND
- ICMP (Internet Control Message Protocol) reports errors back to the original source
- Ping uses ICMP *echo request* and *echo response*
- DHCP allows automatic configuration
- NAT hides multiple computers behind a single address
- Internet follows the end-to-end principle

Summary
(continued)

- Transport protocols that provide end-to-end service run in hosts
- Internet has two main transport protocols
  - UDP provides unreliable, connectionless message delivery
  - TCP provides reliable, stream-oriented delivery
- Dynamic routing was created for WANs and is used in the Internet
- Two basic approaches
  - Distance Vector
  - Link State (also called SPF)

Summary
(continued)

- Internet is divided into Autonomous Systems
- EGPs used between Autonomous Systems
- IGPs used within an Autonomous System
- Internet routing protocols include
  - Border Gateway Protocol (BGP)
  - Routing Information Protocol (RIP)
  - Open Shortest Path First (OSPF)
  - Intermediate System-Intermediate System (IS-IS)
- Multicast routing protocols defined, but are not in wide use