Fundamentals Of Computer Networking And Internetworking

Prof. Douglas Comer

Purdue University

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

MODULE I

Introductions, Course Overview, Approaches To Networking, Open And Closed Systems, Protocols, And Layering

Introductions

- Professor
- TAs
- Students

Course Overview

Topic And Scope

Computer networks and internets: an overview of concepts, terminology, and technologies that form the basis for digital communication in individual networks and the global Internet

You Will Learn

- Fundamental principles
- Concepts
- Terminology (lots of it)
- Key aspects of networking

The Five Key Aspects Of Networking

- Data communications: signals over wires and bits over signals
- Networks: packets over bits
- Internets: datagrams over packets
- Network programming: application data over the Internet
- Cross-functional concepts and technologies: network configuration, control, and management

Features Of The Course

- Covers *all* of networking and internetworking from media to applications
- Examines each of the underlying technologies
- Focuses on concepts and principles that apply across vendors and products
- Provides perspective and shows how the pieces fit together
- Explains how an Internet is built from heterogeneous networks

What You Will Not Learn

- Commercial aspects
  - Vendors
  - Products
  - Prices
  - Markets and marketing
- How to engineer network equipment
- How to configure/operate networks
- How to design new protocols

Practice Sessions (Aka Labs)

- Form an important part of the course
- You will
  - Build network programs
  - Capture and analyze packets
  - Learn about protocols

Background Expected

- Our goal is breadth rather than depth
- Only a few basics are needed
  - Ability to program in C
  - A glancing acquaintance with data structures and pointers
  - A minor brush with algebra
  - A basic understanding of operating systems
- The major requirement is a desire to learn

Summary Of The Course

- Explores all aspects of networking and internetworking
- Gives concepts and principles
- Focuses on the big picture
- Includes lots of programming exercises
- Moves rapidly and covers lots of vocabulary

Historic Approaches To Networking

How Should A Network Be Structured?

- The early phone company answer
  - Data networking is like telephone calls
  - We will devise and offer various data services
  - Charges will depend on distance and duration
  - You only need 128 Kbps
- The early computer vendor answer
  - A network connects computers in your organization
  - We will devise all the necessary equipment and software
  - You only need to connect *our* computers
  - You only need to run *our* applications

How Should A Network Be Structured? (continued)

- The early network equipment vendor answer
  - The network is independent of computers
  - We will create network equipment and interface hardware that connects computers to our network
  - We will build device drivers for your operating system
  - You only need to use *our* network

Some Resulting Commercial Network Systems

- Apple Computer *Appletalk*
- Banyan *Vines*
- Digital Equipment Corporation *DECNET*
- IBM *SNA*
- Novell *Netware*
- Ungermann Bass *NET/One*
- Xerox *XNS*

The Researcher's Answer

- Although we have computers at multiple sites, we reject the phone company's approach
- Because we use diverse computer architectures, we reject the computer vendors' approach
- Because a variety of network technologies are possible, we reject the network vendors' approach
- A variety of applications are possible
- Let's experiment with new technologies (short distance as well as long distance) and new applications

Some Resulting Research Projects

- Xerox Palo Alto Research Center
  - *Ethernet*
- MIT and elsewhere
  - *Token passing ring networks*
- Department of Defense
  - *ARPANET*
  - *SATNET*
  - *Packet radio net*
  - The *global Internet*

Open Vs. Closed Networking

- Closed networks
  - Vertical approach
  - Each vendor designs/builds their own
  - Given technology owned by vendor
  - Vendor may license technology to other groups
- Open networks
  - Competitive approach
  - Multiple groups collaborate to define a technology
  - To insure interoperability, specifications written in *standards* documents that are available to everyone
  - Companies build products according to standards


Questions?

Protocol Standards
And Protocol Design

Why Standardize?

- Networking supports communication among multiple entities
- Agreement needed to make communication correct, efficient, and meaningful

Which Organizations Issue Standards?

- IEEE (*Institute of Electrical and Electronics Engineers*)
- IETF (*Internet Engineering Task Force*)
- ITU (*International Telecommunications Union*)
- ISO (*International Organization for Standardization*)
- W3C (*World Wide Web Consortium*)
- ...and many others

Standards And Standardization

- Joke: why is networking so difficult?
- Because there are so many standards from which to choose

Protocol

- Each *protocol* specifies how to handle one aspect of communication
- A protocol can specify
  - Low-level details such as voltage and frequency
  - High-level details such as format visible to a user
- Many individual communication protocol standards exist
- Set of protocols designed to work together is known as a *suite*
  - Example: TCP/IP Internet protocol suite

Two Key Properties That Protocols Specify

- Syntax
  - Format of each message
  - Representation of data items
  - Encoding of bits in electromagnetic signals
- Semantics
  - Meaning of each message
  - Procedures used to exchange messages
  - Actions to take when an error occurs

Steps In Protocol Design

- Look at the facilities the underlying hardware provides
- Imagine an abstract communication mechanism as a user would like it to work
- Design an efficient implementation of the abstraction
- The key to success: *choose a good abstraction*

Why Protocol Design Is Difficult

- Multiple *implementations* of a protocol will exist
- Implementations will be created by a multiple individuals/organizations
- There are many details to consider
- Key tradeoff
  - A specification that dictates all possible details restricts implementations
  - A specification that does not specify enough details is ambiguous and leads to incompatible implementations

Maximizing Interoperability

- Design principle that maximizes interoperability (due to Postel)

Be conservative in what you send

and be liberal in what you accept.

Protocol Layering and Layering Models

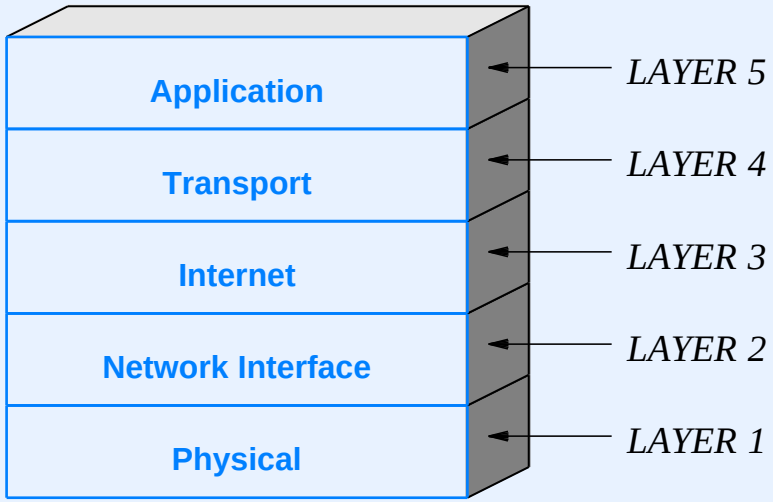
Protocol Layering

- Needed because communication is complex
- Intended primarily for protocol designers
- Divides communication into intellectually manageable pieces
- Provides a conceptual framework that can help us understand protocols
- Ideally, layering is invisible once protocols have been designed
- Notes:
  - Layering gives a guideline, not a rigid framework
  - Optimizations may violate strict layering

Two Layering Models

- Internet protocols use a 5-layer reference model
- ISO and the ITU defined a 7-layer model

Internet Reference Model



- Descriptive model formed after TCP/IP protocols were devised
- Used in practice

Physical Layer

- Underlying transmission media
- Electromagnetic energy and its use
- Representation of information in signals
- Electrical properties such as radio frequencies and voltage
- Associated hardware

Network Interface Layer

- Communication between a computer and network hardware
- Also called *data link* or *MAC* layer
- Mechanisms for gaining access to shared media
- Hardware (MAC) addressing
- Packet (frame) formats
- Packet (frame) types and demultiplexing
- Error detection

Internet Layer

- Communication between a pair of computers across the Internet
- Internet packet format (datagram)
- Internet addressing model and address assignment
- Forwarding of Internet packets
- Dividing an Internet packet into smaller packets for transmission
- Error detection and reporting

Transport Layer

- Communication between a pair of applications
- Demultiplexing among multiple destinations on a computer
- Reliable delivery and retransmission
- Mechanisms to control data rate and avoid congestion

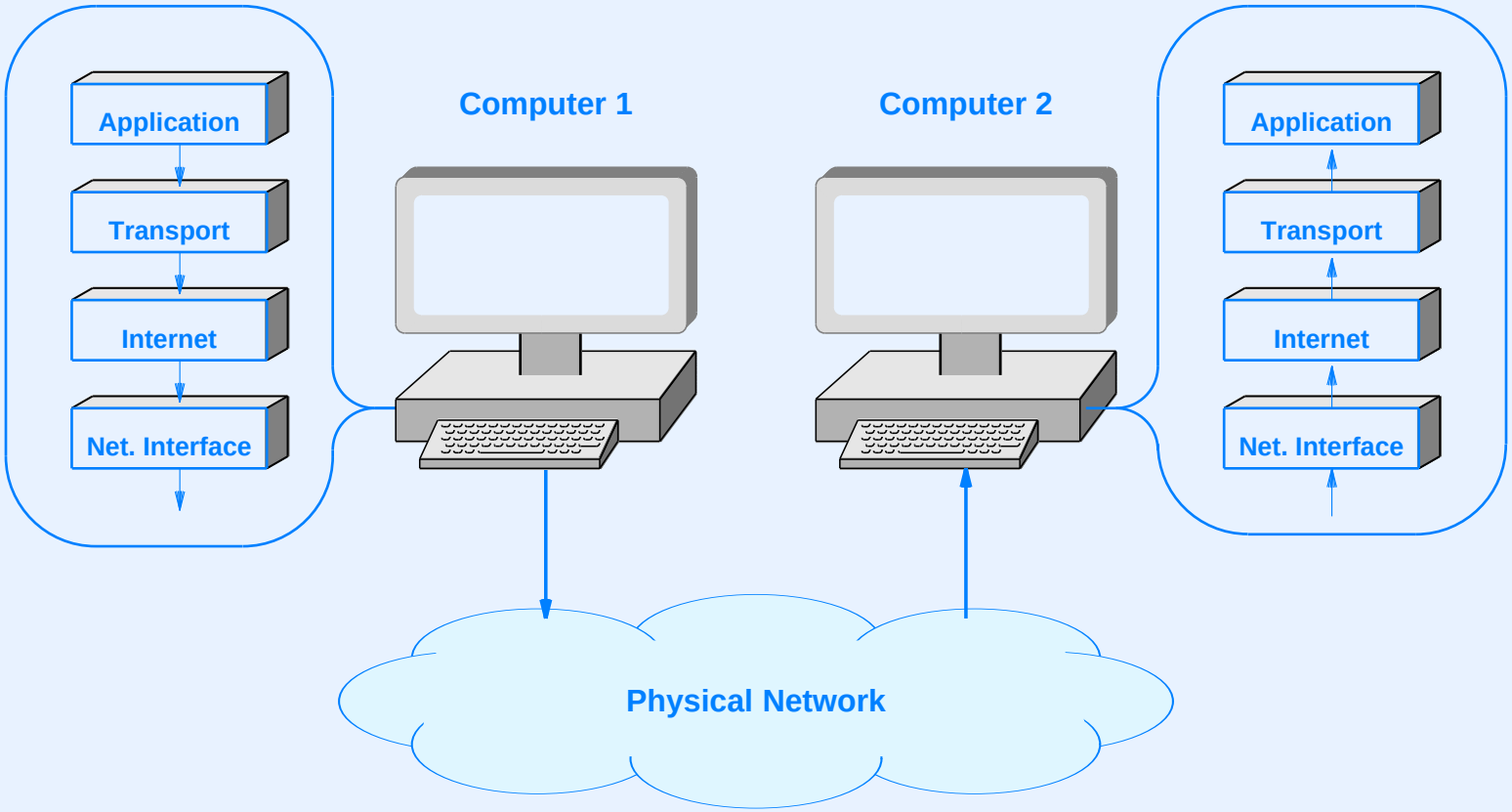
Application Layer

- Format and representation of data and messages
- Procedures applications follow to
  - Transfer data
  - Handle errors or unexpected conditions
- Meaning of messages exchanged
- Internet infrastructure such as routing and DNS

General Idea

- Each computer contains an entire set of layered protocols
- When an application sends a message
  - The message passes down through the layered protocols
  - A given layer adds information and forms a packet
  - The computer transmits the final packet
- When a packet arrives
  - The packet passes up through the protocol layers
  - A given layer performs processing and passes the packet up to the next layer
  - The application receives the message that was sent

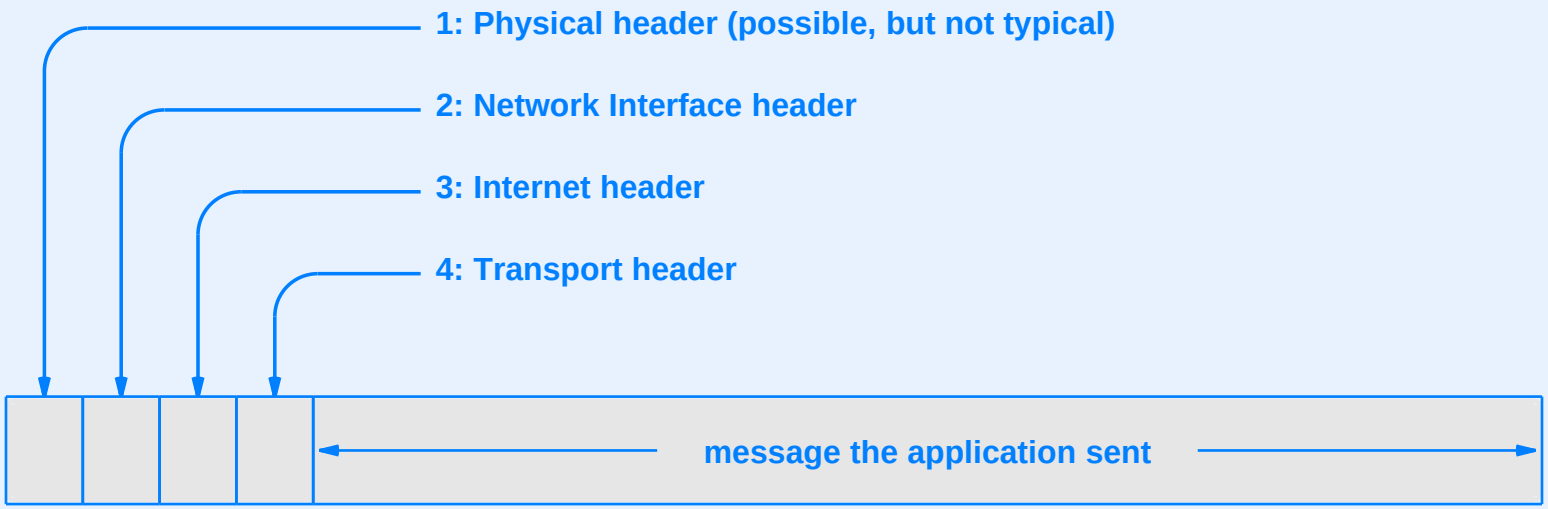
Illustration Of Protocol Software On A Computer



- Protocols on a computer arranged in a conceptual *stack*

Packet Headers As A Packet Passes Across The Internet

- One header prepended by each layer when message sent
- Result: headers are *nested* with lowest-layer header appearing first



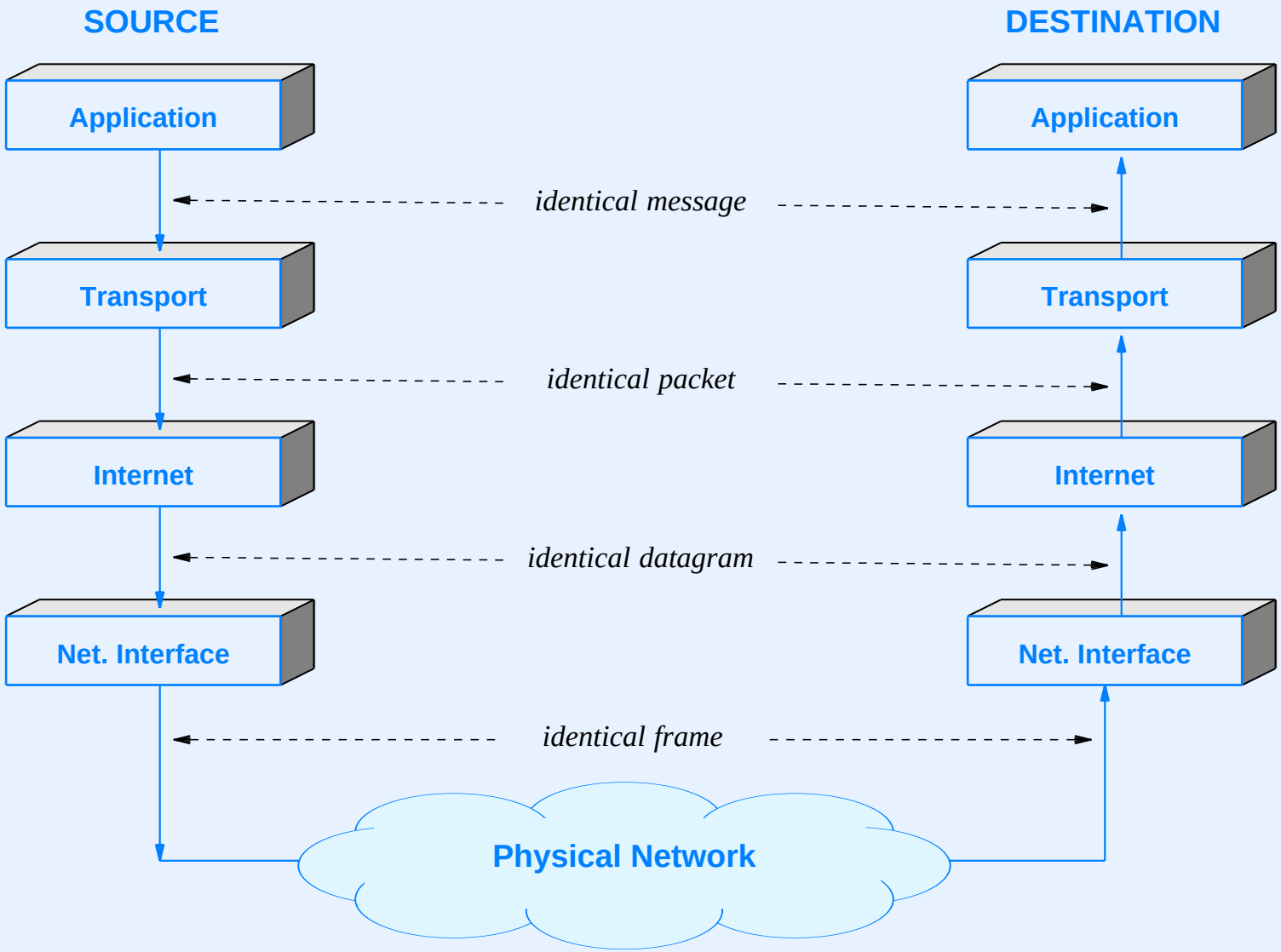
Layering Principle

- Layered protocols enforce an invariant:

Layer N at the destination receives an exact copy of the message sent by layer N at the source. All headers and other modifications added by lower layers at the source must be removed by lower layers at the destination.

- Allows protocol designer to focus on one layer at a time

Illustration Of The Layering Principle



A Few Subtle Complications Of Layering

- Layering diagrams are abstract and simplistic
- Details and exceptions complicate practical systems
- Four examples
  - Cross-layer communication
  - Multiple protocols per layer
  - Layering in an Internet
  - Technologies that intertwine layers

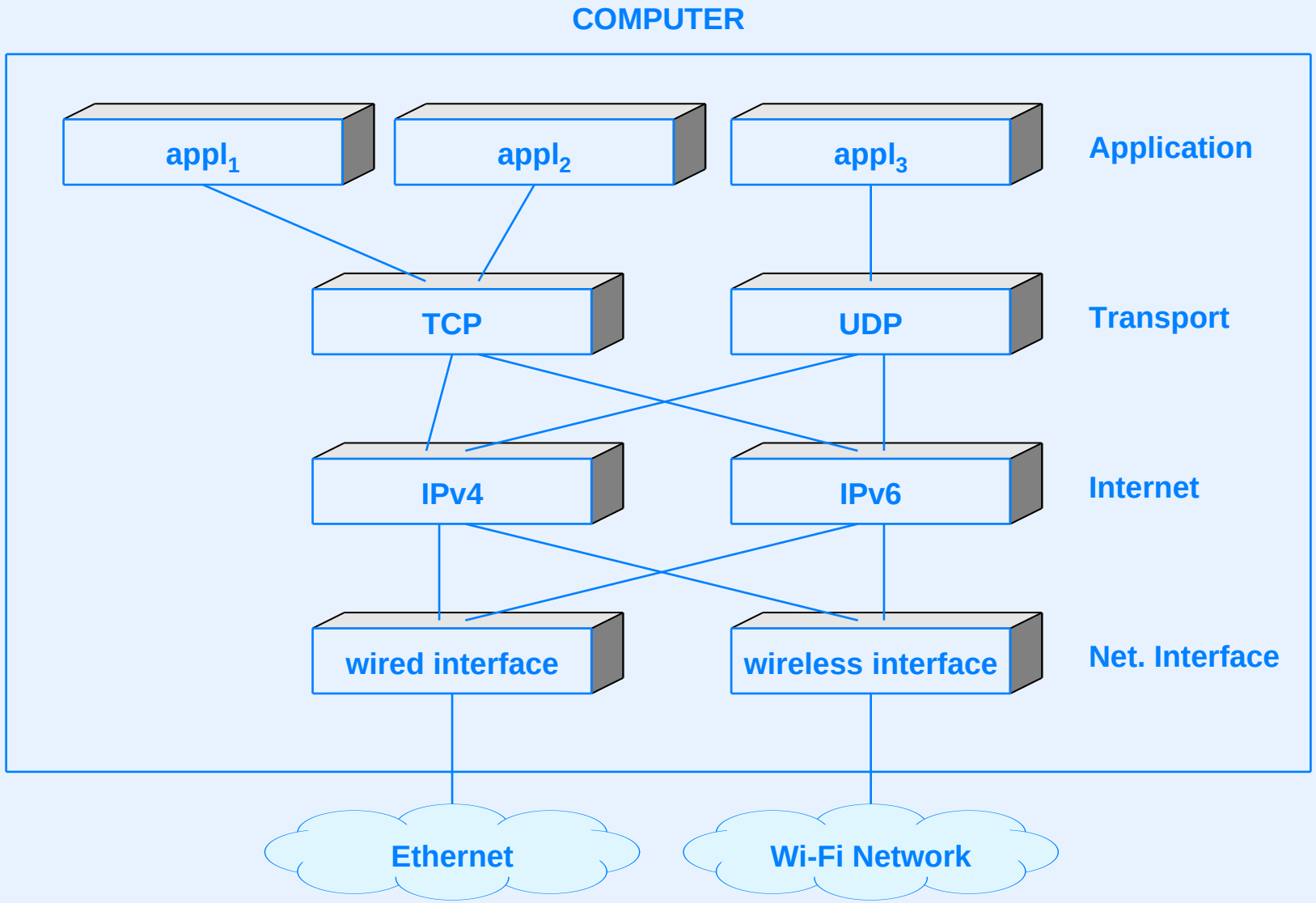
Example Of Cross-Layer Communication

- Facts
  - A transport protocol selects amount of data to send in each packet
  - To optimize performance, ensure packets are full
- Unfortunately
  - To find maximum packet size, transport protocol must interact with a lower layer

Multiple Protocols Per Layer

- Consider a typical computer
- User can run multiple applications simultaneously
  - Email
  - Web browser
- Computer can connect to multiple physical networks
  - Wired Ethernet
  - Wi-Fi wireless network
- Other layers have multiple protocols as well

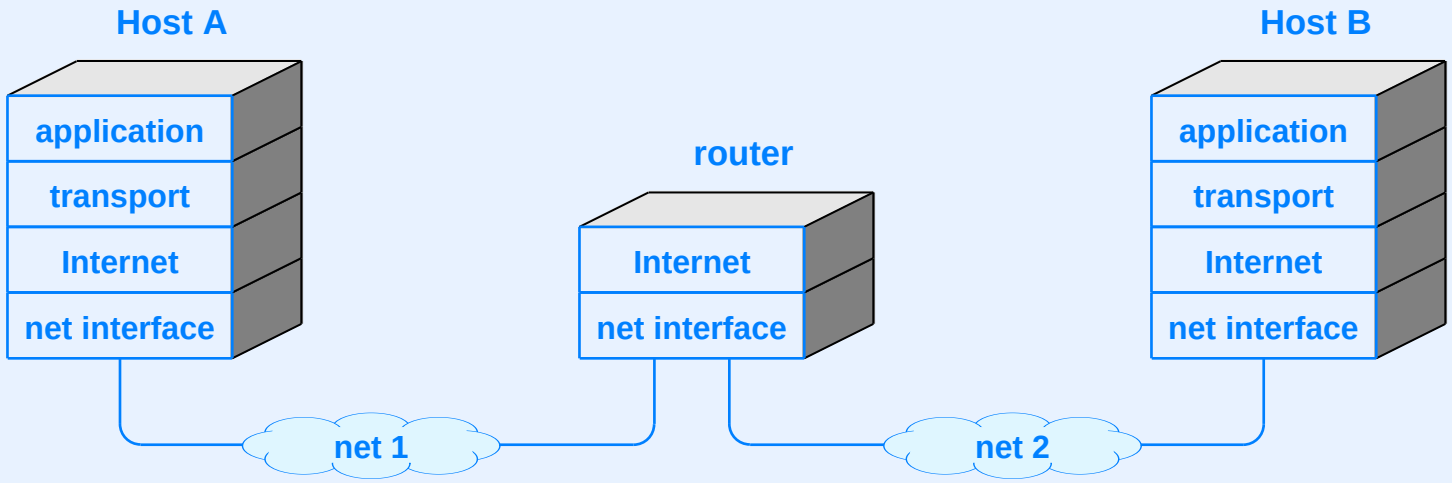
Illustration Of Multiple Protocols At Each Layer



Layering In An Internet

- Our layering diagrams only show two computers connected to a network
- The Internet contains multiple networks interconnected by routers
- Routers only need layer 2 and layer 3 software to forward packets across the Internet

Illustration Of Layers Used To Forward Packets Across The Internet

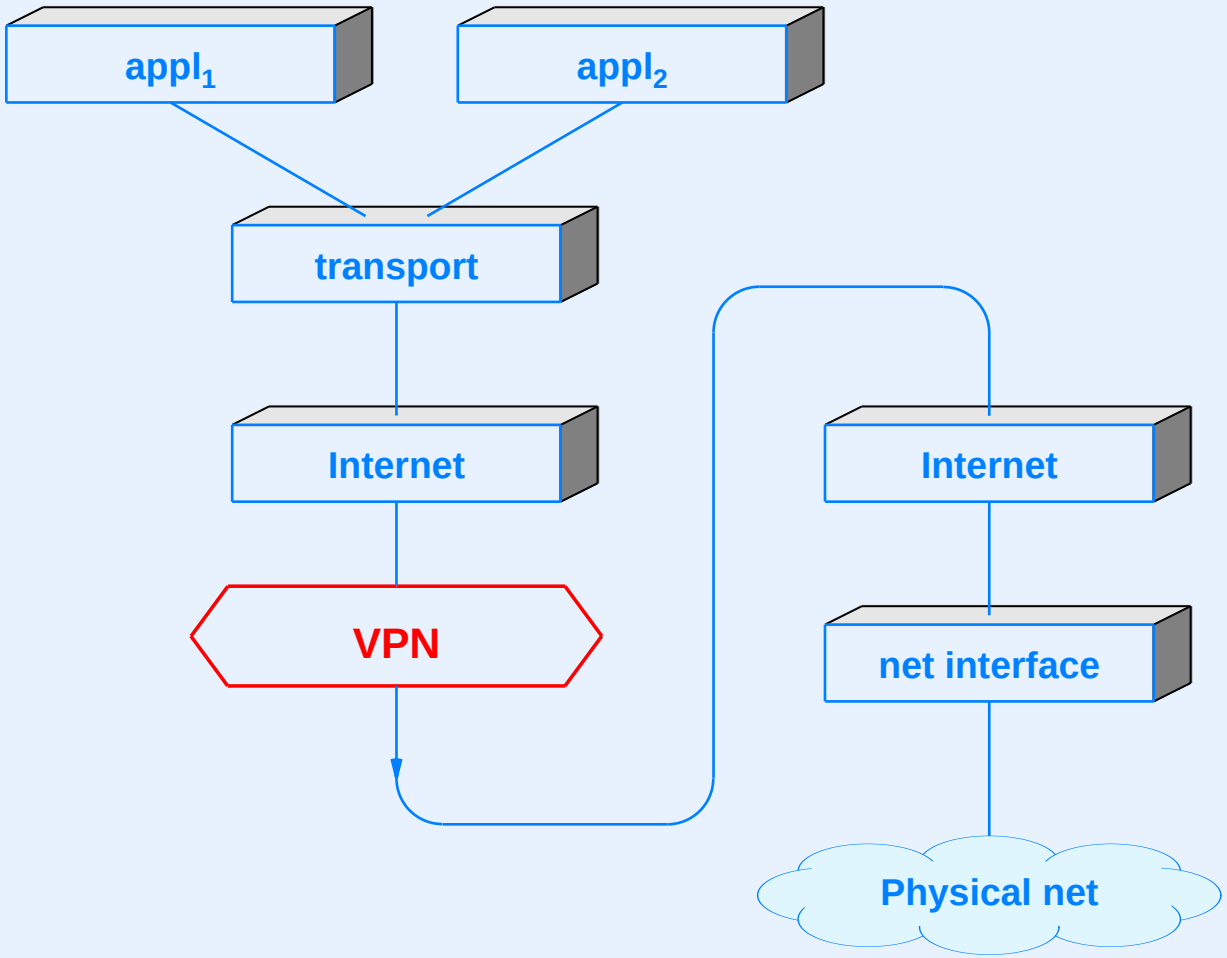


- In practice, routers do more than forward packets
- We will learn more later in the course

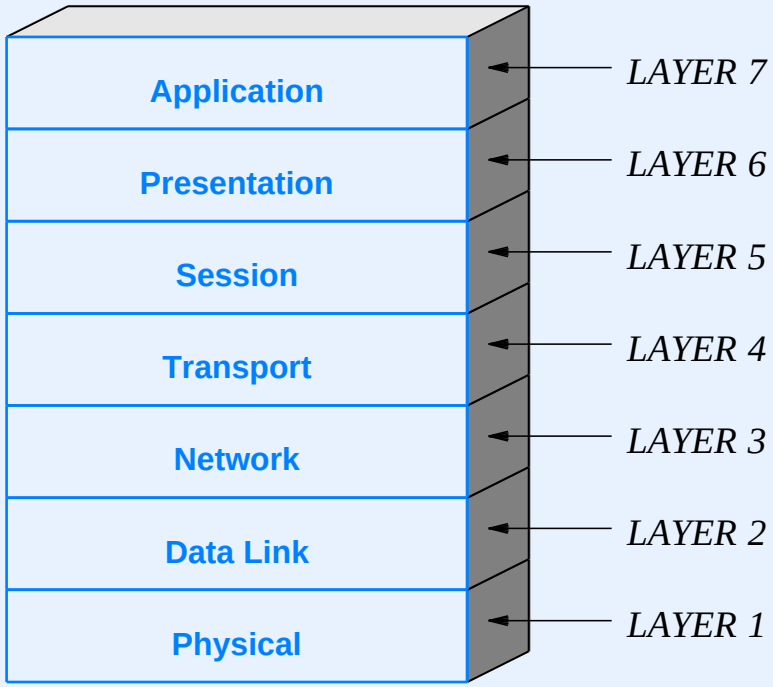
Technologies That Intertwine Layers

- Cross-layer functions
  - Routing protocols operate at layer 5 but change layer 3 forwarding tables
  - Address resolution maps layer 3 addresses to layer 2 addresses
- Layer circularities
  - Tunneling can be used to send IPv6 (a layer 3 protocol) over IPv4 (another layer 3 protocol)
  - Virtual Private Networks (VPNs) send IP over IP

Illustration Of Layering Used By A VPN



ISO 7-Layer Reference Model



- Prescriptive model formed before protocols were devised
- Created by committee vote

ISO 7-Layer Reference Model
(continued)

- Model was defined when data networks connected dumb terminals to large mainframes
- Session layer
  - Handled details of login and control of send/receive
  - Provided opportunity for billing and accounting
- Presentation layer
  - Defined data representation
  - Primary intention was to map character sets
- Both layers now superfluous

Unfortunately

- Marketing organizations decided seven is better than five
- Many textbooks and vendors claim to use “all seven layers”

Summary

- Network systems can be open or closed
  - Closed systems are created and owned by a single company
  - Open systems require that technology be specified in standards documents that allow multiple companies to build products
- A protocol standard can specify data and message representation, rules for message exchange, error handling, or low-level details such as voltage

Summary
(continued)

- A layering model provides a conceptual framework that helps protocol designers create a suite of protocols
- Implementation of layered protocols known as a *stack*
- Internet uses a 5-layer reference model
- Remainder of the course explores each layer

MODULE II

Network Programming And Applications

Topics

- Internet services and communication paradigms
- Client-server model and alternatives
- Network programming with a simplified API
- The socket API
- Application layer protocols
- Examples of standard application protocols

Internet Services And Communication Paradigms

General Principle: Intelligence At The Edge

The Internet does not provide services. Instead, the Internet only provides communication, and application programs provide all services.

- Consequence
  - Every Internet communication, including voice and video teleconferencing, involves communication among application programs

Communication Paradigms

- The Internet offers two communication paradigms

Stream Paradigm	Message Paradigm
Connection-oriented	Connectionless
1-to-1 communication	Many-to-many communication
Sequence of individual bytes	Sequence of individual messages
Arbitrary length transfer	Each message limited to 64 Kbytes
Used by most applications	Used for multimedia applications
Built on TCP protocol	Built on UDP protocol

- Each paradigm has surprising characteristics

Stream Paradigm (TCP)

- Transfers a sequence of bytes
- Connection-oriented: data sent between two applications
- Bidirectional (one stream in each direction)
- No meaning attached to data and no boundaries inserted in data
- Surprising characteristic:

Although it delivers all bytes in sequence, the stream paradigm does not guarantee that the chunks of bytes passed to a receiving application correspond to the chunks of bytes transferred by the sending application.

Message Paradigm (UDP)

- Connectionless: network accepts and delivers individual messages
- If the sender places *N* bytes in a message, a receiver will find exactly *N* bytes in the incoming message
- Paradigm allows unicast, multicast, or broadcast delivery (one destination, multiple destinations, or all destinations)
- Surprising characteristic:

Although it preserves boundaries, the message paradigm allows messages to be lost, duplicated, or delivered out-of-order; neither the sender nor receiver is informed when such errors occur.

Stream Transport And Data Chunks

- The protocol system may
  - Divide the data from the sender into multiple segments and deliver a few bytes at a time to the receiver
  - Combine data from multiple transmissions into a single large chunk and deliver it to the receiver all at once
- Consequence: receiving application cannot know exactly which pieces were sent

Example #1

- Assume a stream connection between two applications
- Sender
  - Places 1000-byte message in buffer *buf*
  - Makes a single request to send all 1000 bytes
- Receiver
  - Allocates a buffer *b* with 1000 bytes
  - Reads 1000 bytes from the stream into buffer *b*
- The OS may return between 1 and 1000 bytes
- Application *must* make repeated calls until all 1000 bytes have been acquired

Example #2

- Assume a stream connection between two applications
- Sender transmits a sequence of four messages that are each 100 bytes long
- Receiver allocates a large buffer *b* of 1000 bytes and requests that up to 1000 bytes from stream be read into buffer *b*
- The OS may choose to return all four messages (400 bytes) with a single read request
- Receiving application *must* be able to separate received data into four separate messages

Programming Hints

- When using the stream paradigm
  - Devise a way that a receiver knows where a message ends
  - Read from a socket until the entire message has been acquired
- When considering using the message paradigm
  - Don’t (at least not yet)

Identifying Individual Messages In A Stream

- Possibilities
  - Send exactly one message followed by *end of file (EOF)*
  - Send multiple messages with an integer length before each message
  - Send multiple messages with a termination character (or sequence) following each message
- Notes
  - Any technique can be used as long as both sides agree
  - If sending a multi-byte length value or multi-byte termination sequence, remember that the application may need multiple calls receive all bytes

Questions

- In a realistic setting
  - Is division of a message likely to occur?
  - Is aggregation of multiple messages likely to occur?
- Answers yes! (depending on the size of the messages)
  - Messages larger than 1400 characters are usually divided into multiple packets for transmission, and *may* be delivered together or separately
  - The stream service is designed to aggregate small messages before making them available to a receiving application

Buffering In The Stream Paradigm

- Aggregation, which makes bulk transfer more efficient, can occur on the sending or receiving side
- The stream paradigm includes a *push* operation that an application can use to force transmission and delivery
- Unix convention: automatically *push* for each individual *write* call
- Programming hints
  - To ensure a small message is transmitted and delivered without delay, use a separate *write*
  - Even with *push*, network delays mean applications must be written to tolerate aggregation
- More details later in the course

Client-Server Model
And Alternatives

Client-Server Model Of Interaction

- Used by applications to establish communication
- One application acts as a *server*
  - Starts execution first
  - Awaits contact
- The other application becomes a *client*
  - Starts after server is running
  - Initiates contact
- Important concept: once communication has been established, data (e.g., requests and responses) can flow in either direction between a client and server

Characteristics Of A Client

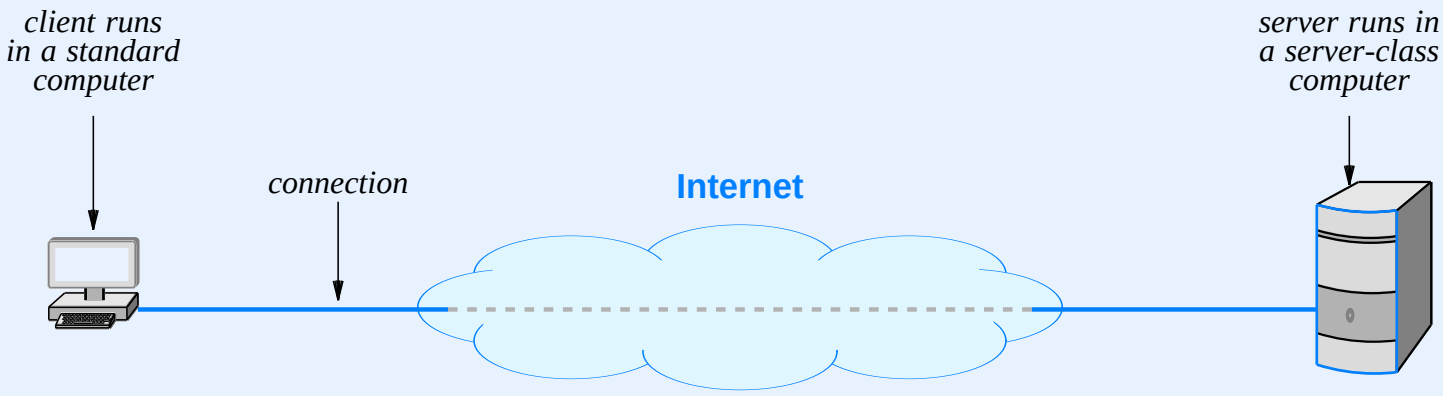
- Arbitrary application program that becomes a client temporarily
- Usually invoked directly by a user, and usually executes only for one session
- Actively initiates contact with a server, exchanges messages, and then terminates contact
- Can access multiple services as needed, but usually contacts one remote server at a time
- Runs locally on a user's personal computer or smart phone
- Does not require especially powerful computer hardware

Characteristics Of A Server

- Special-purpose, privileged program dedicated to providing a service
- Usually designed to handle multiple remote clients at the same time — complicates the design
- Invoked automatically when a system boots, and continues to execute through many client sessions
- Waits passively for contact from arbitrary remote clients and then exchanges messages
- Requires powerful hardware and a sophisticated operating system
- Runs on a large, powerful computer

Server Programs And Server-Class Computers

- Confusion exists between scientific and marketing terminology
- Scientific: a *client* and a *server* are each programs
- Marketing: a *server* is a powerful computer



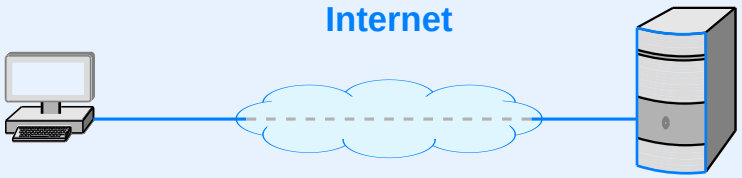
Summary Of Client-Server Interaction

Server Application	Client Application
Starts first	Starts second
Does not need to know which client will contact it	Must know which server to contact
Waits passively and arbitrarily long for contact from a client	Initiates a contact whenever communication is needed
Communicates with a client by sending and receiving data	Communicates with a server by sending and receiving data
Stays running after servicing one client, and waits for another	May terminate after interacting with a server

Illustration Of Steps Taken By Client And Server

Client Side

- Agree a priori on a port number, N
- Start after server is already running
- Obtain server name from user
- Use DNS to translate name to IP address
- Contact server using IP address and port N
- Interact with server and then exit



Server Side

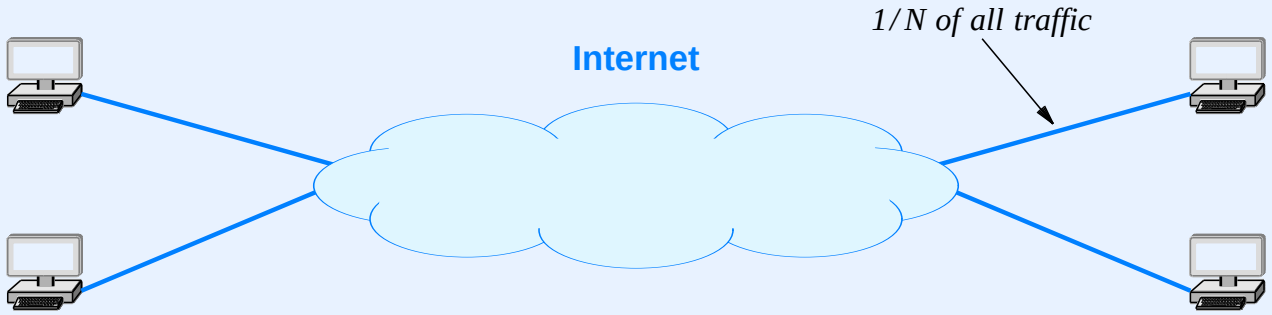
- Agree a priori on a port number, N
- Start before any of the clients
- Register port N with the local system
- Wait for contact from a client
- Interact with client until client finishes
- Wait for contact from the next client…

Alternatives To Client-Server

- Broadcast
  - Sender broadcasts message and all stations receive it
  - Does not scale well (becomes inefficient)
  - Difficult to restrict data access
- Rendezvous point
  - Intermediary connects communicating applications
  - In essence, there are two clients and a server
  - Rendezvous point becomes a bottleneck

Alternatives To Client-Server (continued)

- Peer-To-Peer Interaction
  - Designed to avoid central server bottleneck
  - Data divided among *N* computers
  - Each computer acts as a server for its data and as a client for other data
  - Given computer receives *1 / N* of the traffic



Network Programming With A Simplified API

Network Programming

- General term that refers to the creation of client and server applications that communicate over a network
- Programmer uses an *Application Program Interface* (*API*)
  - Set of functions
  - Include control as well as data transfer functions (e.g., establish and terminate communication)
- Defined by the operating system; not part of the Internet standards
- *Socket API* has become a de facto standard

A Simplified API

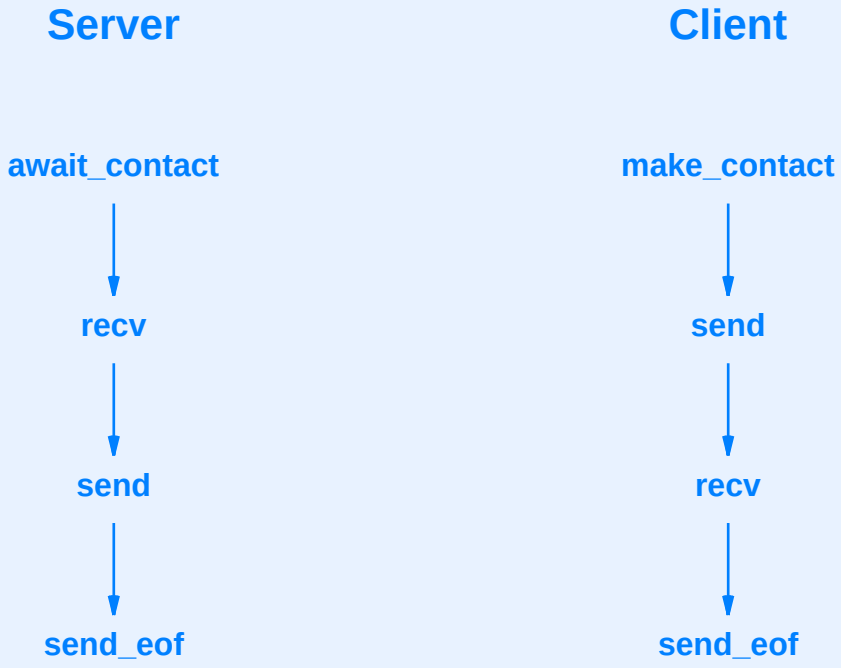
- Will help you get started
- General idea
  - Server is identified by pair (computer, application)
  - Server starts first and waits for contact
  - Client specifies server's location
  - Once a connection is established, client and server can exchange data
- Only seven functions in the simplified API

Our Simplified API

Operation	Meaning
await_contact	Used by a server to wait for contact from a client
make_contact	Used by a client to contact a server
appname_to_appnum	Used to translate a program name to an equivalent internal binary value
cname_to_comp	Used to translate a computer name to an equivalent internal binary value
send	Used by either client or server to send data
recv	Used by either client or server to receive data
send_eof	Used by both client and server after they have finished sending data

Client And Server Using The API

- Sequence of calls for a trivial exchange in which a client sends a single request and the server responds



- Both sides must call *send_eof* because communication is bidirectional

Data Types For Our Simplified API

Type Name	Meaning
appnum	A binary value used to identify an application
computer	A binary value used to identify a computer
connection	A value used to identify the connection between a client and server

An Extra Function For Convenience

- Simplified API includes an extra function, *recvln*
- Not required, but convenient
- Similar to *recv*
  - Receives data from a connection
  - Places data in a buffer
- Difference
  - Reads *exactly* the amount requested
  - Technique: repeatedly call *recv* until specified length has been acquired

Argument Types For Our API

Function Name	Type Returned	Type of arg 1	Type of arg 2	Type of args 3–4
await_contact	connection	appnum	–	–
make_contact	connection	computer	appnum	–
appname_to_appnum	appnum	char *	–	–
cname_to_comp	computer	char *	–	–
send	int	connection	char *	int
recv	int	connection	char *	int
recvln	int	connection	char *	int
send_eof	int	connection	–	–

- You will learn more in the PSOs

The Socket API

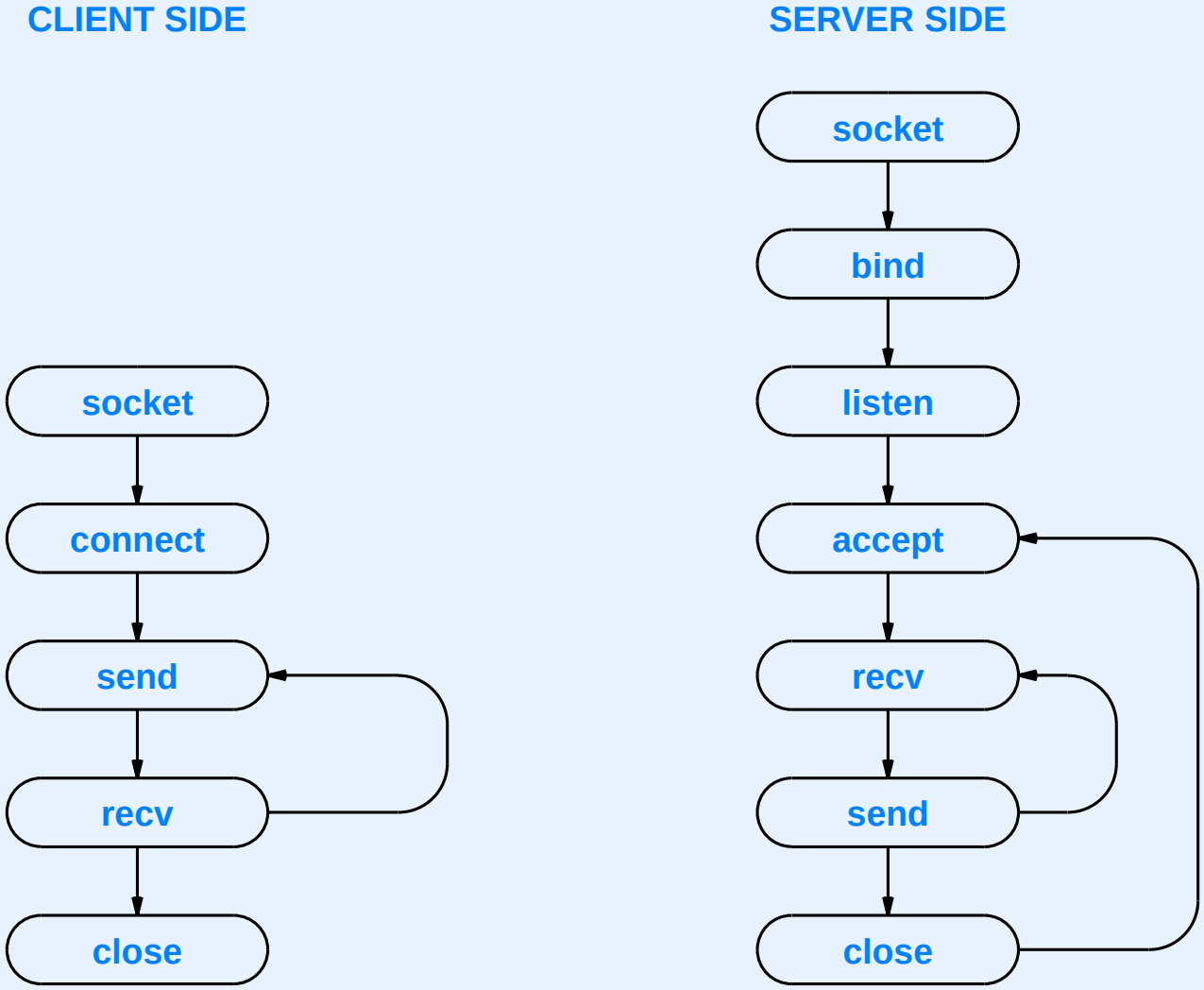
Sockets

- Originally part of BSD Unix
- Now standard in the industry
- AT&T defined an alternative named *TLI* (*Transport Layer Interface*), but TLI is now extinct
- Almost every OS includes an implementation
- MS Windows chose to make minor changes (annoying)

Socket Characteristics

- Socket can be used for
  - Connectionless communication (UDP message)
  - Connection-oriented communication (TCP stream)
- Many functions in the API
- Approach
  - Create a socket
  - Make many function calls to specify type of communication, remote computer's address, port number to be used, etc.
  - Use socket to send / receive data
  - Close the socket (terminate use)

Example Socket Calls For Stream Communication



Application Layer Protocols

Terminology

- Availability of an application protocol
  - *Closed* — vendor defines a protocol for their products
  - *Open* — standardized and available for all vendors
- Basic protocol types
  - *Data representation* — message and data formats
  - *Data transfer* — procedures for exchanging messages and handling unexpected / error conditions
- Notes
  - Application may define separate protocol for each type
  - Term *Transfer* in a protocol title indicates the latter

Defining An Application Layer Protocol

- Programmer specifies representation
  - Format of each message and each data item
  - Meaning of each item in a message
- Programmer specifies transfer
  - Which side sends first
  - Which side closes the connection first
  - What to do if one side crashes unexpectedly

State In An Application Protocol

- Big decision: should state information be kept?
- Stateful protocol assumes previous requests have been honored
- Stateless protocol assumes each request is independent
- Example of stateful interaction
  - Request 1 specifies “read from file X”
  - Request 2 specifies “read next 128 bytes”
- Example of stateless interaction
  - Request 1 specifies “read bytes 0-127 from file X”
  - Request 2 specifies “read bytes 128-255 from file X”

Examples Of Standard Application Protocols

Application Protocol Examples

- Web browsing
- Email
- File transfer
- Remote login and remote desktop
- Domain Name System (name lookup)

Application-Layer Protocols For The Web

Standard	Purpose
HyperText Markup Language (HTML)	A representation standard used to specify the contents and layout of a web page
Uniform Resource Locator (URL)	A representation standard that specifies the format and meaning of a web page identifier
HyperText Transfer Protocol (HTTP)	A transfer protocol that specifies how a browser interacts with a web server to transfer data

- Reminder: keyword *Transfer* in the name of a protocol means the protocol specifies message exchange

HyperText Markup Language (HTML)

- Representation standard for multimedia documents
- Specifies document is entirely in printable text
- Uses declarative rather than procedural approach
- Document includes *metadata* that can link to arbitrary item
- Document contains markup guidelines rather than precise, detailed formatting or typesetting instructions
  - Page can be displayed on arbitrary device
  - Appearance depends on device
- Embedded *tags* control display
  - Form is <*tag_name*> and </*tag_name*>

Uniform Resource Locator (URL)

- Representation standard
- A text string with punctuation characters separating the string into (optional) subfields
- General form is:

protocol : // *computer_name* : *port* / *document_name* ? *parameters*

- Example where protocol, port, and parameters are omitted:

www . cs . purdue . edu / people / comer

HyperText Transfer Protocol (HTTP)

- Transfer protocol used with the Web
- Specifies format and meaning of messages
- Each message represented as text
- Transfers arbitrary binary data
- Can download or upload data
- Incorporates caching for efficiency
- Browser sends *request* to server

Four Major HTTP Request Types

Request	Description
GET	Requests a document; server responds by sending status information followed by a copy of the document
HEAD	Requests status information; server responds by sending status information, but does not send a copy of the document
POST	Sends data to a server; the server appends the data to a specified item (e.g., a message is appended to a list)
PUT	Sends data to a server; the server uses the data to completely replace the specified item (i.e., overwrites the previous data)

- GET request has the form:

GET /*item* *version* *CRLF*

- Version is HTTP/1.0 or HTTP/1.1

HTTP Response

- Response begins with a header in text, optionally followed by an item (which can be binary)
- Header uses *keyword: information* form like email header
- Header ends with a blank line

HTTP Header Format

- General form

`HTTP/1.0` *status_code status_string CRLF*
`Server:` *server_identification CRLF*
`Last-Modified:` *date_document_was_changed CRLF*
`Content-Length:` *datasize CRLF*
`Content-Type:` *document_type CRLF*
CRLF
... item begins here and contains datasize bytes ...

Telnet Example (Apache Web Server)

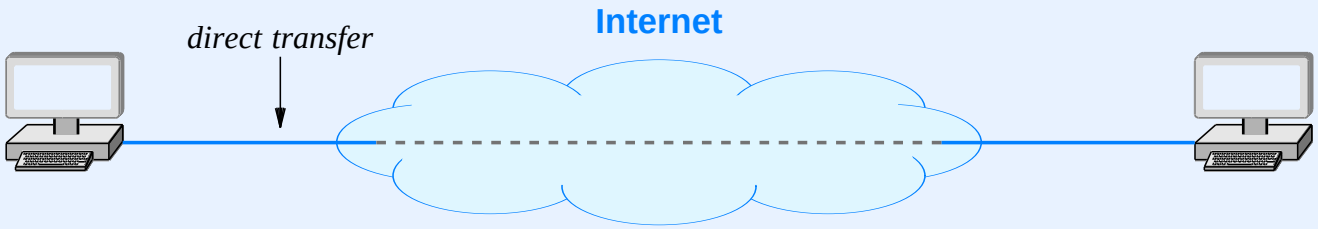
```
$  telnet www.cs.purdue.edu 80
Trying 128.10.19.20...
Connected to lucan.cs.purdue.edu.
Escape character is '^]'.
GET /homes/comer/ HTTP/1.0
HTTP/1.1 200 OK
Date: Sun, 10 Nov 2013 11:38:27 GMT
Server: Apache/2.2.11 (Unix) mod_ssl/2.2.11 OpenSSL/0.9.8r
Last-Modified: Mon, 17 Oct 2011 22:21:41 GMT
ETag: "bafb0-a50-4af8607f7c740"
Accept-Ranges: bytes
Content-Length: 2640
Connection: close
Content-Type: text/html
```

...data from the web page follows here

Application Protocol Examples

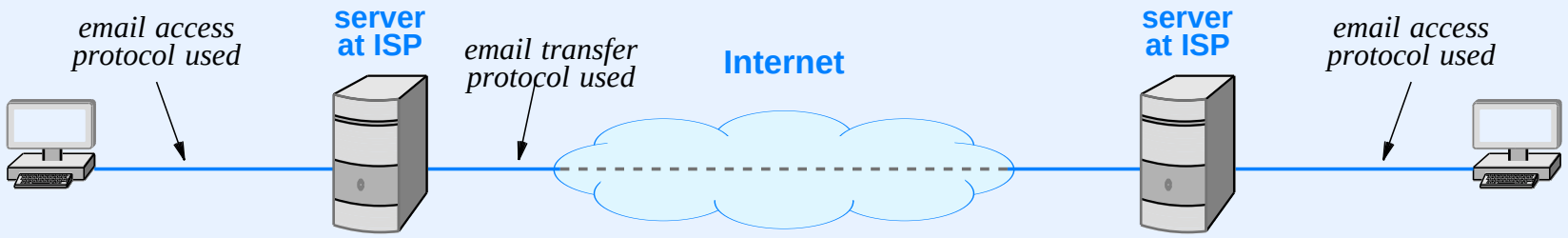
- Web browsing
- Email
- File transfer
- Remote login and remote desktop
- Domain Name System (name lookup)

Original End-To-End Email Paradigm



- Each computer runs
  - Email server to accept incoming email
  - Email client to send outgoing email
- Incoming mail deposited in user's mailbox
- Outgoing mail placed in queue
- User interface to read or compose messages separate from transfer applications

Current Email Paradigm



- User's mailbox located on separate computer (usually at an ISP)
- Mail transfer application deposits message in mailbox
- User interface application accesses remote mailbox
  - A web browser may be used as an access mechanism
  - Special-purpose applications also exist

Simple Mail Transfer Protocol (SMTP)

- Standard for email transfer
- Follows a stream paradigm
- Uses textual control messages
- Only transfers text messages
- Terminates message with *<CR> <LF>* **.** *<CR> <LF>*
- Allows a sender to specify recipients' names and checks each name
- Sends only one copy of a message to a computer, even if destined to multiple recipients on the computer

Example SMTP Session

```
S:  220 somewhere.com Simple Mail Transfer Service Ready
C:  HELO example.edu
S:  250 OK
C:  MAIL FROM:<John_Q_Smith@example.edu>
S:  250 OK
C:  RCPT TO:<Mathew_Doe@somewhere.com>
S:  550 No such user here
C:  RCPT TO:<Paul_Jones@somewhere.com>
S:  250 OK
C:  DATA
S:  354 Start mail input; end with <CR><LF>.<CR><LF>
C:  ...sends body of mail message, which can contain
C:  ...arbitrarily many lines of text
C:  <CR><LF>.<CR><LF>
S:  250 OK
C:  QUIT
S:  221 somewhere.com closing transmission channel
```

Mail Access Protocols

- Two standard protocols
  - Post Office Protocol version 3 (POP3)
  - Internet Mail Access Protocol (IMAP)
- Functionality
  - Provide access to a user's mailbox
  - Permit user to view headers, download, delete, or send individual messages
  - Client runs on user's personal computer
  - Server runs on a computer that stores user's mailbox

RFC2822 Mail Message Format

- Email representation standard
- Name derived from the Internet standard in which it is defined
- Specifies
  - Email message consists of text file
  - Blank line separates *header* from *body*
  - Header lines have the form:

Keyword: information

RFC2822 Mail Message Format (continued)

- Some keywords have defined meanings:
  - From:
  - To:
  - Subject:
  - Cc:
- Keywords starting with uppercase X have no effect
- Examples:

X-Best-networking-Course: CS422 at Purdue

X-Spam-Check-Results: bulk spam 90% likely

X-Worst-TV-Shows: any reality show

Multimedia Email

- Observe
  - Email was standardized when computers only had character-oriented (textual) interfaces
  - SMTP is limited to transferring plain text messages
  - Users want to email photos, spreadsheets, messages with special fonts and color
- Question: can SMTP be used to transfer such email?
- Answer: it is possible because one can encode arbitrary binary items in plain text (think of a hex dump)

Sending Non-Text Email

- Standard is *MIME* (*Multimedia Internet Mail Extensions*)
- Backward compatible with RFC2822 mail and SMTP
- Sender
  - Encodes arbitrary binary item in plain text
  - Adds lines to email header to specify MIME
  - Places additional headers before each item in the message (including plain text items)
- Sender can specify content type and encoding
- Standard includes *Base64* encoding

Examples Of Mime Headers

- MIME header lines added to other RFC2822 headers

```
MIME-Version: 1.0
Content-Type: Multipart/Mixed; Boundary=xyz123
```

- Each part of the message has a MIME header that starts with the separator and specifies content type and encoding
- Example

```
--xyz123
Content-Type: image/jpeg
```

←*blank line ends header*

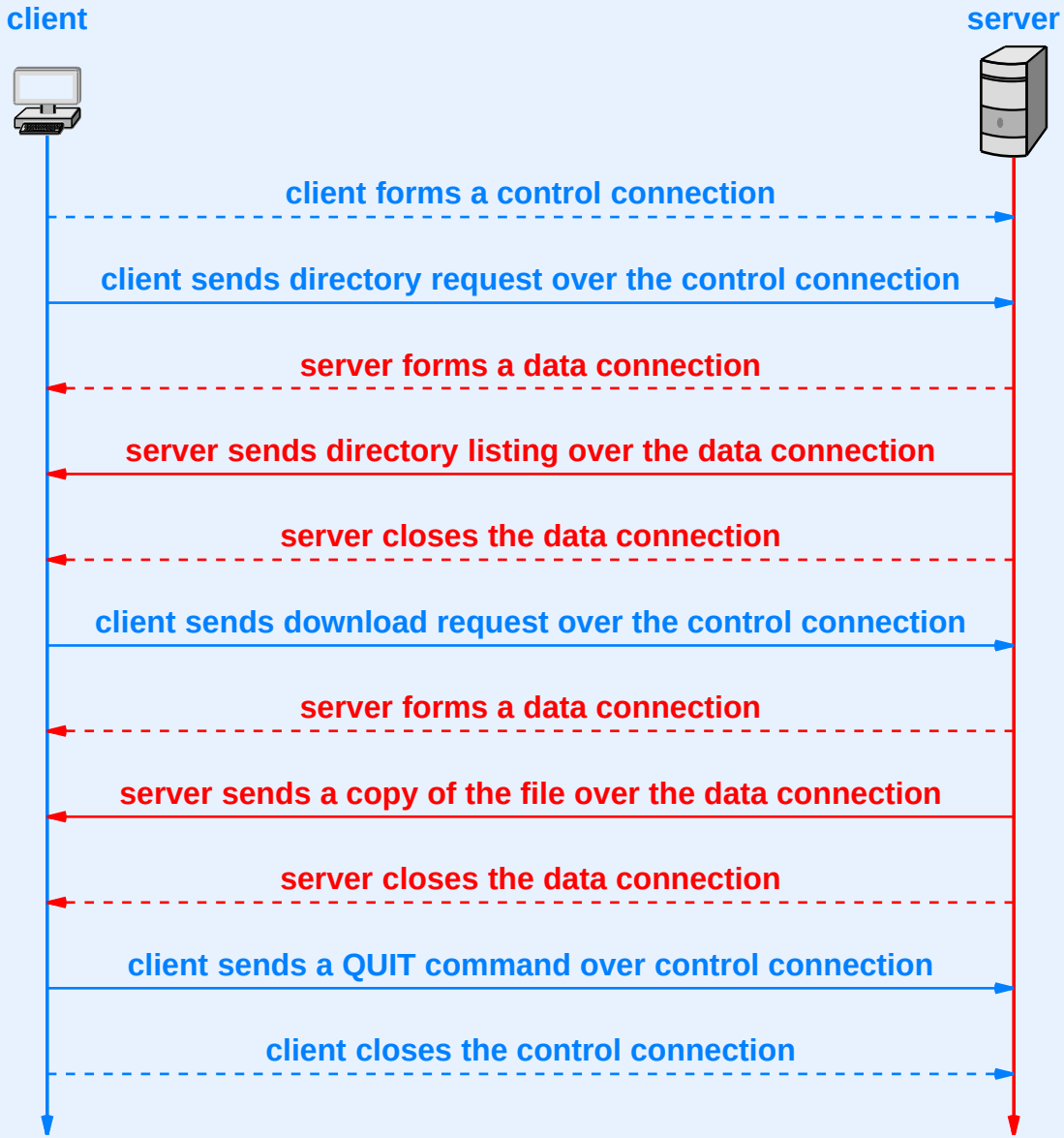
Application Protocol Examples

- Web browsing
- Email
- File transfer
- Remote login and remote desktop
- Domain Name System (name lookup)

File Transfer

- Standard is the *File Transfer Protocol* (*FTP*)
- Once accounted for the most packets on the Internet
- Interesting communication paradigm
  - Client forms a control connection to send requests
  - Server forms data connection for each file transferred
  - Server closes data connection after transfer complete
- Notes
  - Using a separate connection allows arbitrary data transfer
  - For data connections, the server becomes a client and the client becomes a server (important for NAT)

Illustration Of FTP Communication



Application Protocol Examples

- Web browsing
- Email
- File transfer
- Remote login and remote desktop
- Domain Name System (name lookup)

Remote Login And Remote Desktop

- Remote login
  - Intended for systems with command-line interface
  - Internet standard is TELNET
  - Secure shell (ssh) encrypts transfers
  - To appreciate the complexity of application protocols look at the TELNET standard
- Remote desktop
  - Intended for systems that have a Graphical User Interface (GUI)
  - No Internet standards
  - Move to *thin client* has revived interest

Application Protocol Examples

- Web browsing
- Email
- File transfer
- Remote login and remote desktop
- Domain Name System (name lookup)

Domain Name System (DNS)

- Important piece of Internet infrastructure
- Runs at the application layer
- Translates human-readable names into the binary addresses used by the Internet Protocol
- Example
  - Computer www.cs.purdue.edu
  - Has the IP address 128.10.19.20

DNS Terminology

- Names are *hierarchical*
- Each name divided into *segments* by period character, which is read “dot”
- Most significant segment is on the right
- Rightmost segment known as a *top-level domain* (*TLD*)
- Client program known as a *resolver*
  - Used by web browser, email, etc

Top-Level Domains

Domain Name	Assigned To
aero	Air transport industry
arpa	Infrastructure domain
asia	For or about Asia
biz	Businesses
com	Commercial organizations
coop	Cooperative associations
edu	Educational institutions
gov	United States government
info	Information
int	International treaty organizations
jobs	Human resource managers
mil	United States military
mobi	Mobile content providers

Top-Level Domains (continued)

Domain Name	Assigned To
museum	Museums
name	Individuals
net	Major network support centers
org	Non-commercial organizations
pro	Credentialed professionals
travel	Travel and tourism
xxx	Adult entertainment (porn)
country code	A sovereign nation

- In 2014, ICANN decided to allow many new TLDs

Domain Registration

- Organization
  - Applies under a specific top-level domain
  - Can choose an internal hierarchy
  - Assigns each computer a name
- Geographic registration is possible

  `cnri.reston.va.us`
- Some countries impose conventions
  - Universities in Great Britain register under

    `ac.uk`

Domains With Most Hosts (July 2013)

Domain	Hosts	Explanation
net	366592151	Networks
com	163634309	Commercial
jp	74461142	Japan
de	34904481	Germany
br	33691951	Brazil
it	26136473	Italy
cn	19976554	China
mx	17658991	Mexico
fr	17437386	France
au	16900586	Australia
ru	15122103	Russian Federation
nl	14011944	Netherlands
pl	14011944	Poland
ar	13335042	Argentina
edu	12251571	Educational
ca	9004861	Canada
uk	8116718	United Kingdom
in	7429638	India
tr	7146979	Turkey
tw	6429021	Taiwan

- See domain survey at www . isc . org for details

Host Names and Services Offered

- Many organizations choose a host name to match the service a computer offers

mail.foobar.com

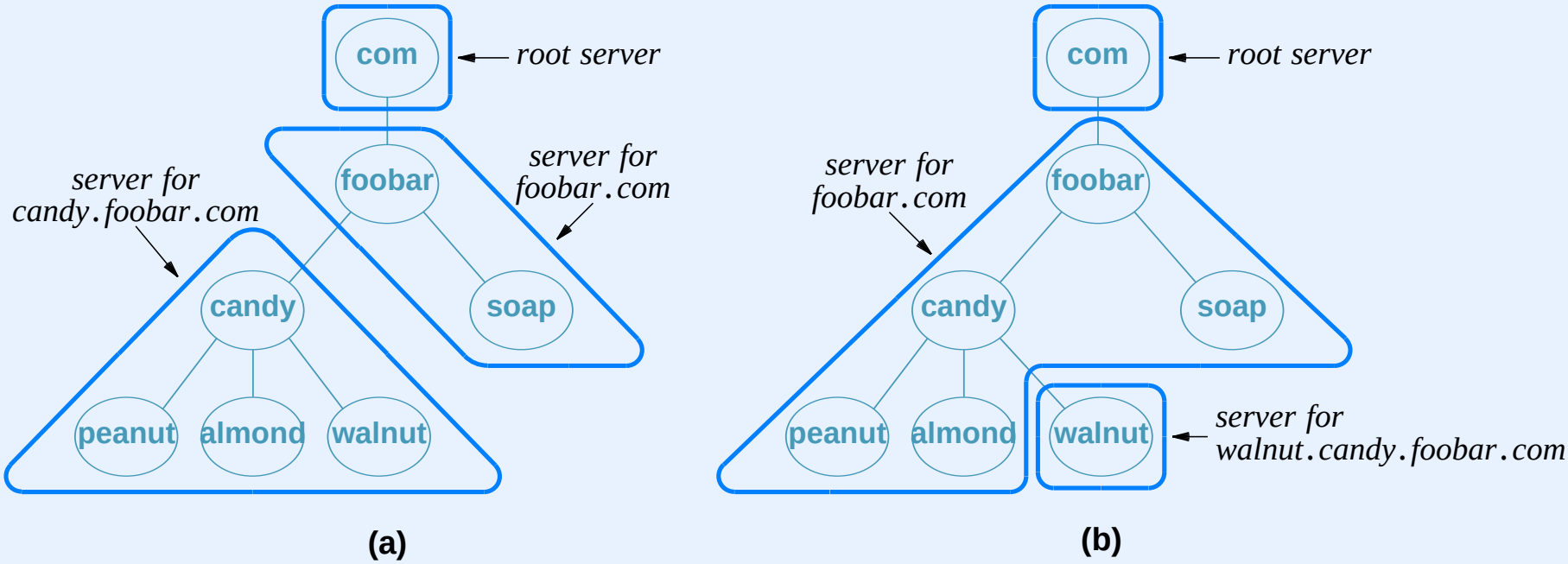
ftp.foobar.com

www.foobar.com

- Although convenient for humans, a host name does not specify which servers are running (e,g., a computer named *mail* could run a web server)

DNS Servers

- Names divided into a hierarchy of servers
- Multiple groupings possible
- Hypothetical example



Name Resolution And Caching

- Resolver
  - Acts as a client
  - Is configured with address of local DNS server
  - Contacts local server first
  - Socket library resolver is *gethostbyname*
- Caching
  - Follows locality of reference principle
  - Each DNS server caches results
  - Cached item never kept when stale

DNS Server Algorithm Part 1

Given:

A request message from a DNS name resolver

Provide:

A response message that contains the address

Method:

Extract the name, N, from the request
if (server is an authority for N) {
    Form and send an *authoritative* response
    to the requester;
else if (answer for *N* is in the cache) {
    Form and send a *nonauthoritative* response
    to the requester;

DNS Server Algorithm Part 2

```
else  {   /* Need to look up an answer */
     if  ( authority server for N is known )  {
          Send request to authority server;
     }  else  {
          Send request to root server;
     }
     Receive response and place in cache;
     Form and send a response to the requester;
}
```

Summary

- Applications provide all Internet services
- Internet offers connection-oriented stream communication or connectionless message communication
- Most applications follow client-server approach
  - Server starts first and awaits client
  - Client contacts server
- Socket API is a de facto standard
- Application-layer protocol can define
  - Data and message formats (representation)
  - Rules for message exchange (transfer)

Summary
(continued)

- Applications reviewed include
  - Web (URL, HTML, HTTP)
  - Email (SMTP, RFC2822, MIME)
  - File transfer (FTP)
  - Remote login and remote desktop (TELNET)
  - Domain Name System (DNS)

MODULE III

Foundations Of Data Communications And The Physical Layer

Topics

- Motivation and model
- Information sources and signals
- Transmission media
- Reliability and channel coding
- Transmission modes
- Modulation and demodulation
- Multiplexing and demultiplexing (channelization)

Motivation And Model

What Is Data Communications?

- Broad field of study
- Usually associated with the Physical Layer
- Touches on
  - Physics
  - Mathematics
  - Engineering
- Includes
  - Transmission of signals
  - Encoding data
  - Modulation and multiplexing

Motivation

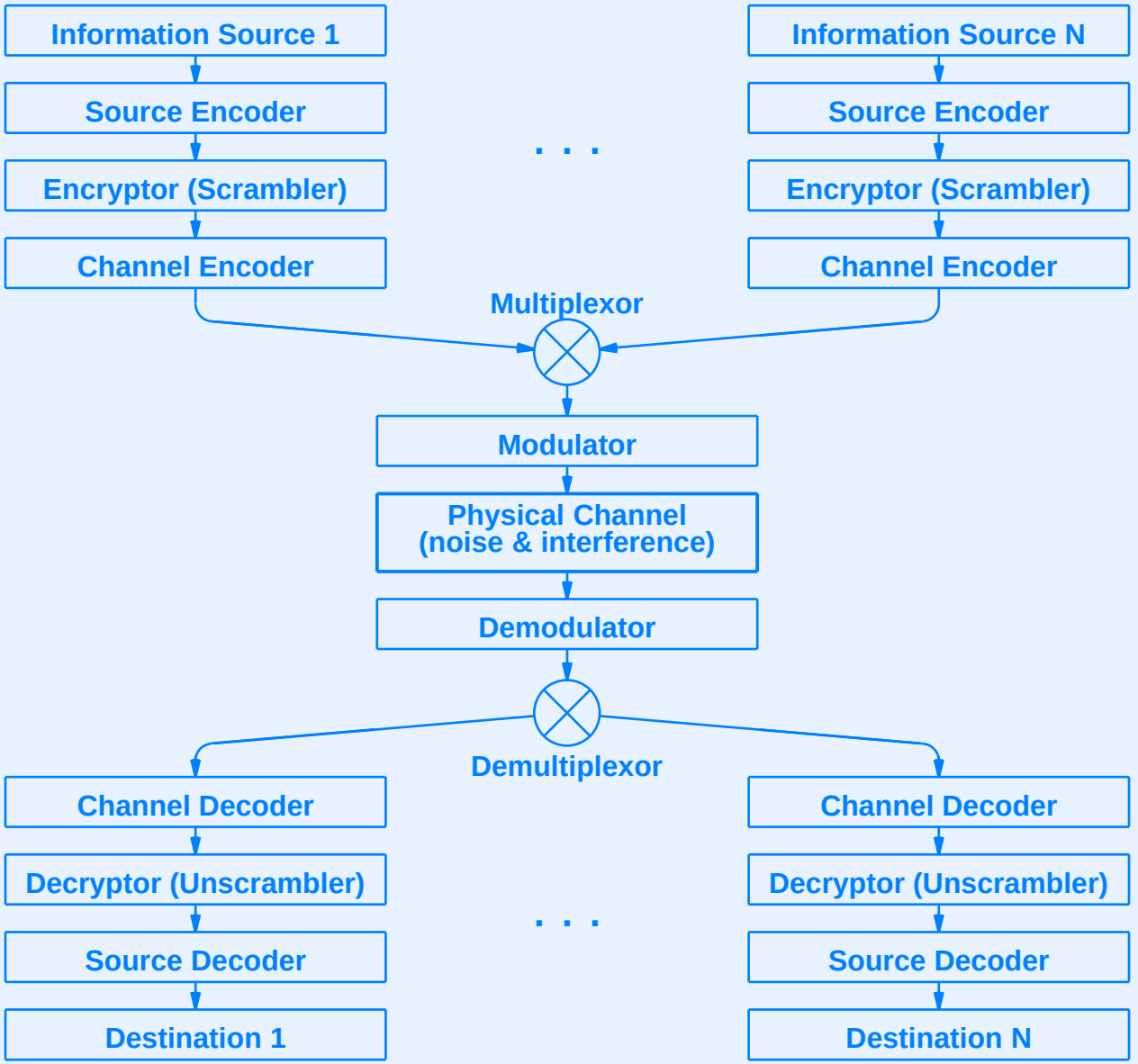
- Find ways to transmit analog and digital information
  - Using natural phenomena (e.g., electromagnetic radiation)
  - Allow multiple senders to share a transmission medium
- Data communications provides
  - A conceptual framework
  - Mathematical basis

Key Concept

Although we tend to think of analog and digital communication separately, ultimately, all communication uses the same physical phenomena, usually electromagnetic energy.

- Differences lie in the way the physical phenomena are used
  - Analog: use all values in a continuous range
  - Digital: restrict use to a fixed set of values, usually two
- Data communications covers both analog and digital

Conceptual Framework For Data Communications

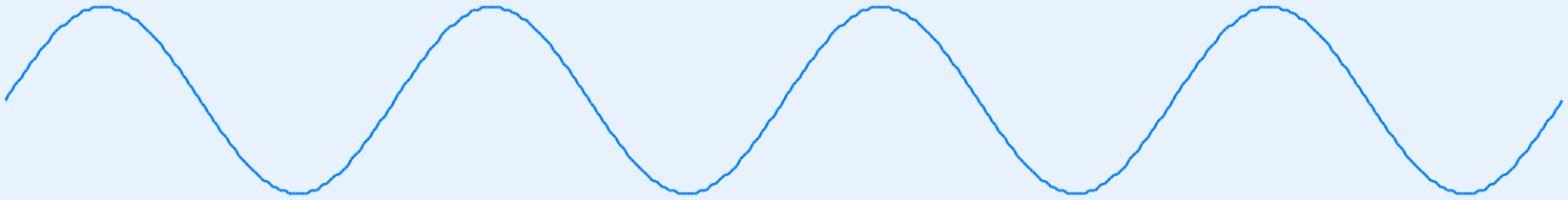


Information Sources And Signals

Sources Of Information

- An input signal can arise from
  - Transducer such as a microphone
  - Receiver such as an Ethernet interface
- We use the term *signal processing* to describe the recognition and transformation of signals

Sine Waves



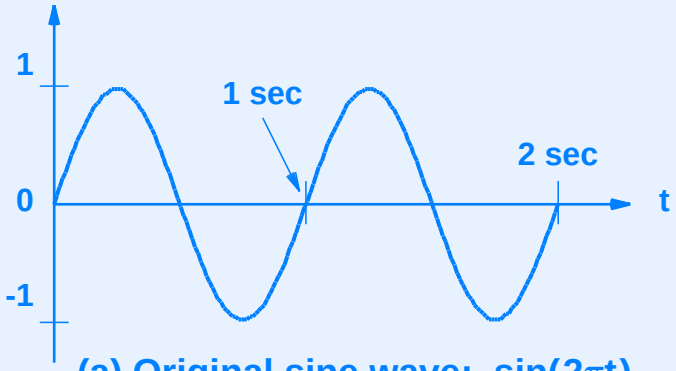
- Fundamental because sine waves characterize many natural phenomena
- Examples
  - Audible tones
  - Radio waves
  - Light energy

Fourier Analysis

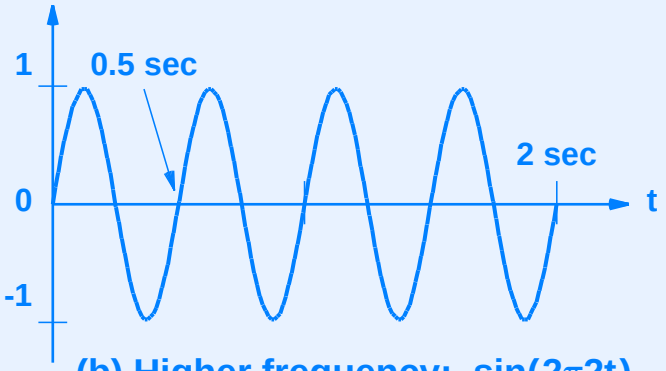
- Multiple sine waves can be added together
  - Result is known as a *composite* wave
  - Corresponds to combining multiple signals (e.g., playing two musical tones at the same time)
- Mathematician named Fourier discovered how to decompose an arbitrary composite wave into individual sine waves
- Fourier analysis provides the mathematical basis for *signal processing*
- Bad news: according to Fourier, a digital wave decomposes into an infinite set of sine waves

Sine Wave Characteristics

- Three important characteristics are used in networks: frequency, amplitude, and phase



(a) Original sine wave: sin(2πt)



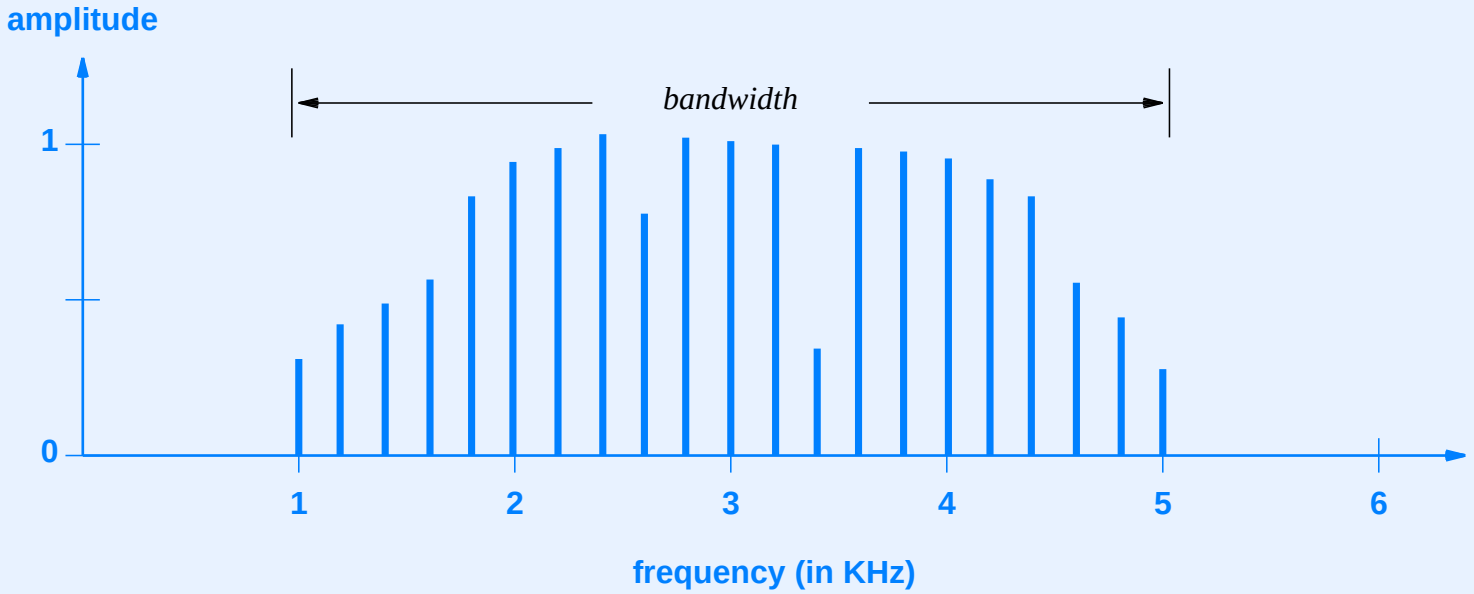
(b) Higher frequency: sin(2π2t)

(c) Lower amplitude: 0.4 sin(2πt)

(d) New phase: sin(2πt+1.5π)

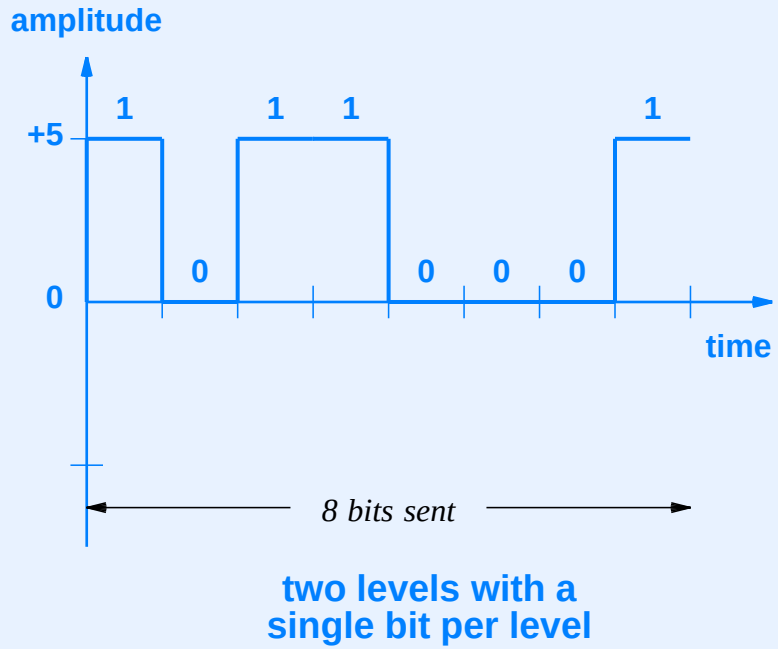
Definition Of Analog Bandwidth

- Decompose a signal into a set of sine waves and take the difference between the highest and lowest frequency
- Easy to compute from a *frequency domain plot*
- Example signal with bandwidth of 4 Kilohertz (KHz):

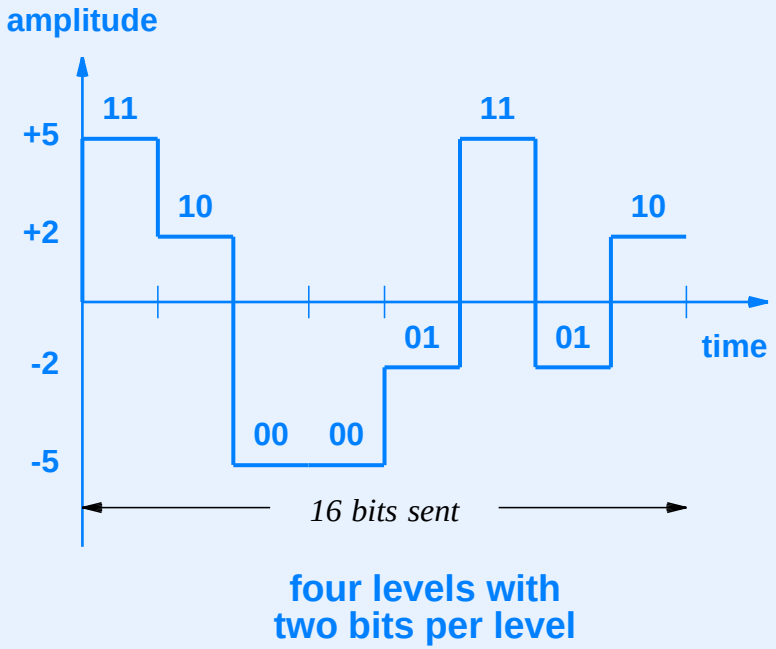


Digital Signals And Signal Levels

- A digital signal level can represent multiple bits
- Example



two levels with a
single bit per level



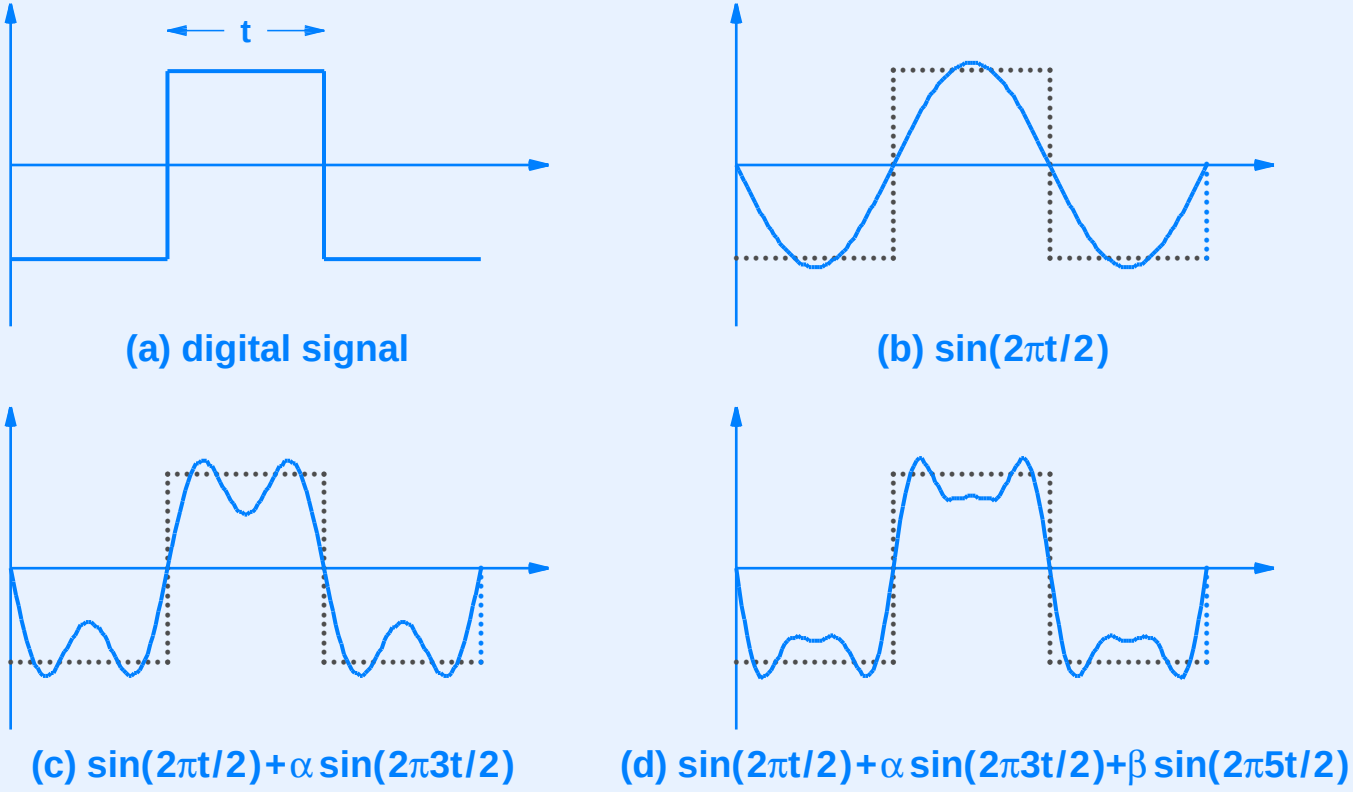
four levels with
two bits per level

- *Baud rate* is number of times signal changes per second;

$$\text{data rate in bits per second} = \textit{baud} \times \left\lfloor \log_2(\textit{levels}) \right\rfloor$$

Converting Digital To Analog

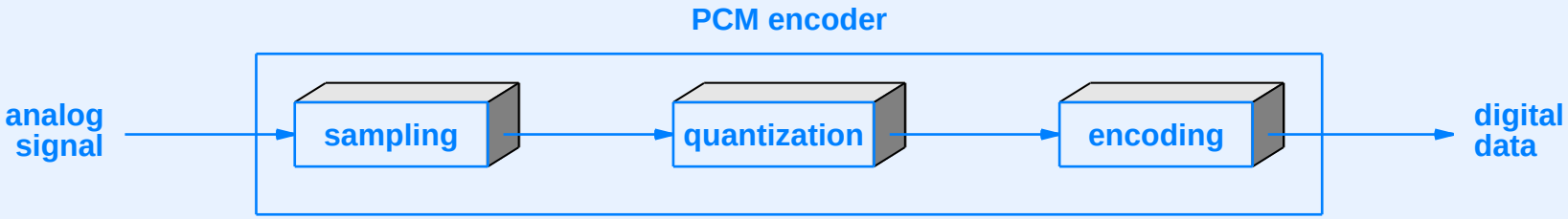
- Approximate digital signal with a composite of sine waves:



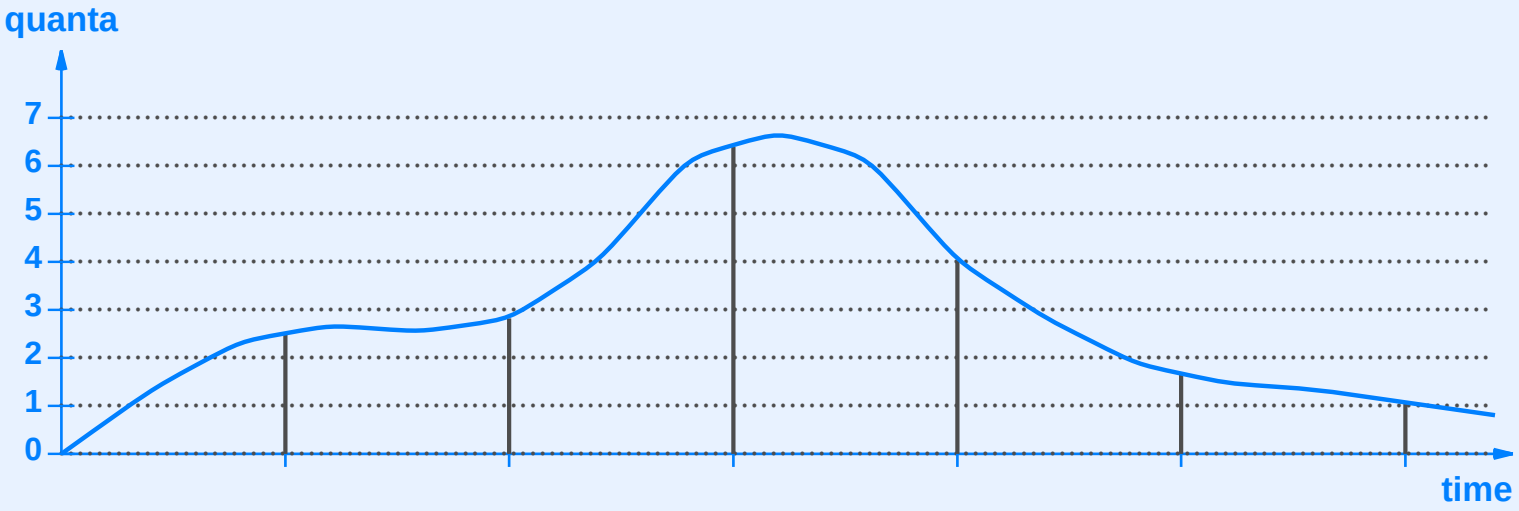
- Mathematically, the bandwidth of a digital signal is infinite

Converting Analog To Digital

- Three steps taken during conversion



- Example sampling using eight levels



Sampling Rate And Nyquist Theorem

- How many samples should be taken per second?
- Mathematician named Nyquist discovered the answer:

$$sampling\ rate = 2 \times f_{\max}$$

where $f_{\max}$ is highest frequency in the composite signal

- Example: to capture audio frequencies up to 4000 Hertz, a digital telephone system samples at 8000 samples per second
- Amount of data generated by a single digitized voice call:

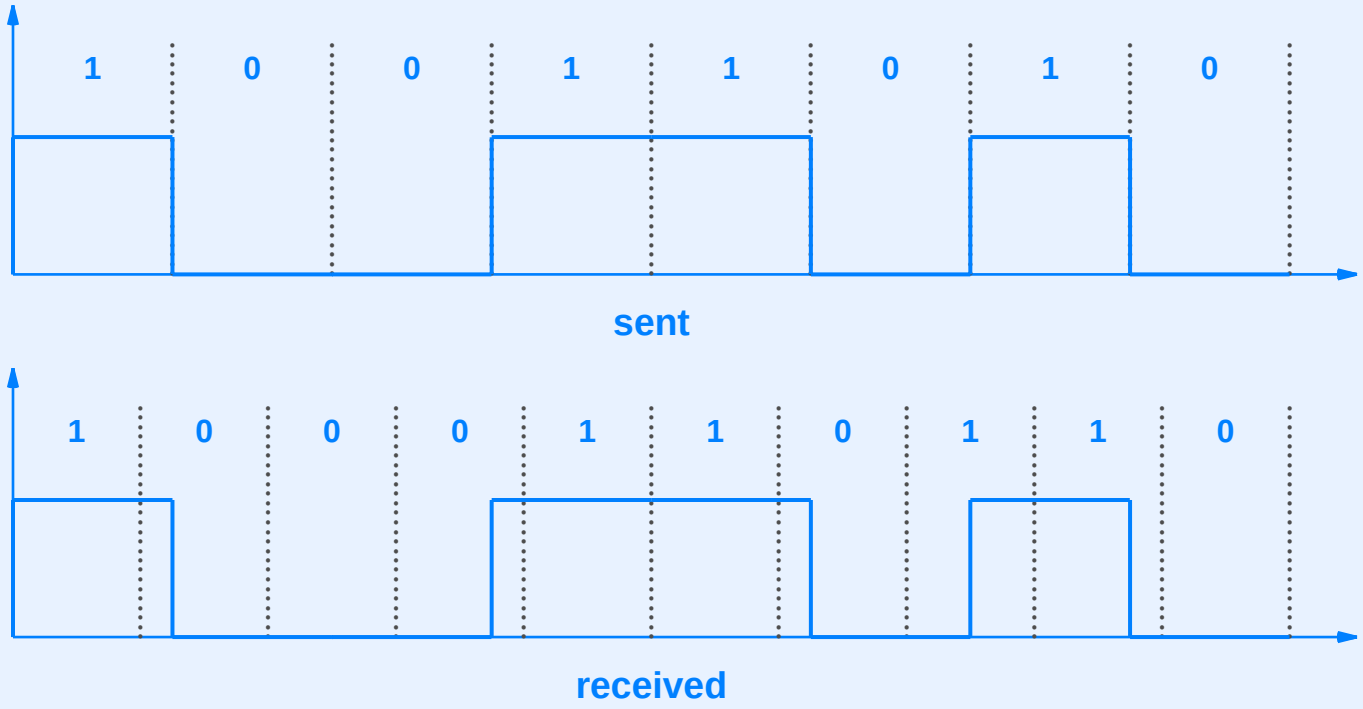
$$data\ rate = 8000 \frac{samples}{second} \times 8 \frac{bits}{sample} = 64{,}000 \frac{bits}{second}$$

Nonlinear Encoding

- Linear sampling does not work well for voice
- Researchers created nonlinear sampling that modify dynamic range to reproduce sounds to which the human ear is sensitive
- Mu-law (μ-law)
  - Used in North America and Japan
  - More dynamic range, but more sensitive to noise
- A-law
  - Used in Europe
  - Less sensitive to noise, but less dynamic range

Synchronization Errors And Line Coding

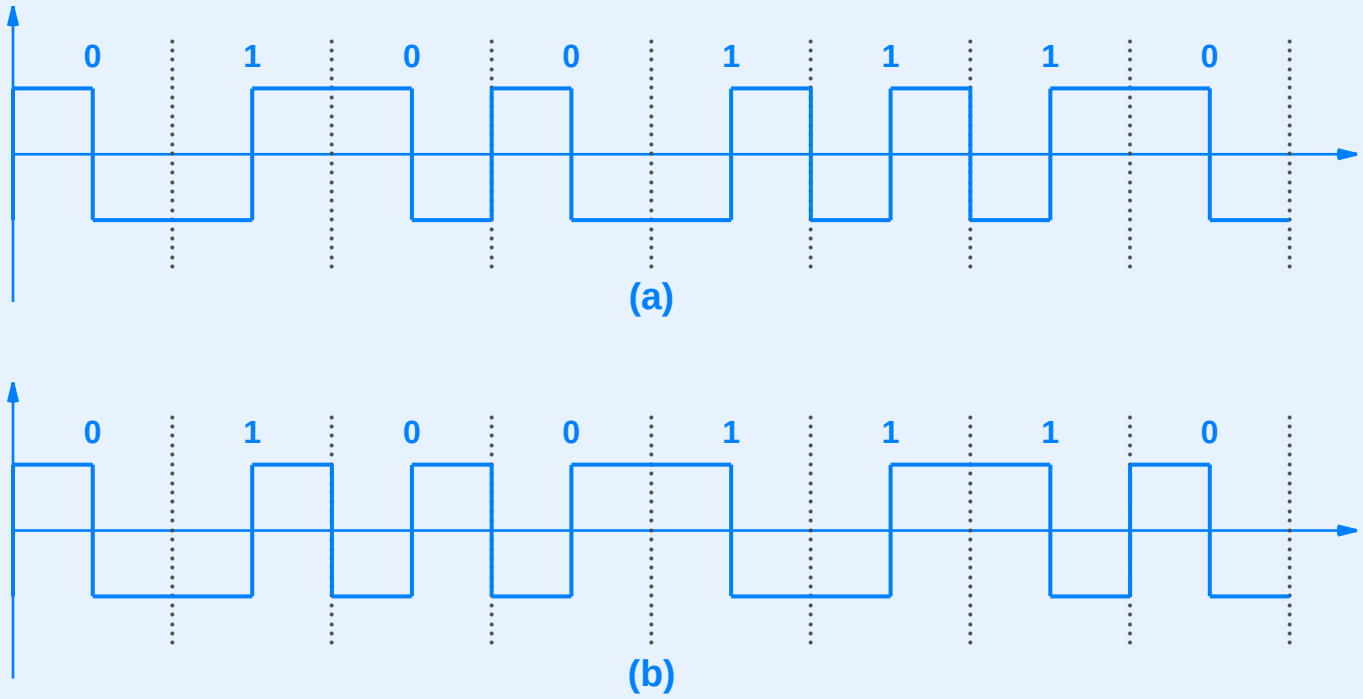
- Synchronization error occurs when receiver and sender disagree about bit boundaries (clocks differ)



- Line coding techniques prevent synchronization errors

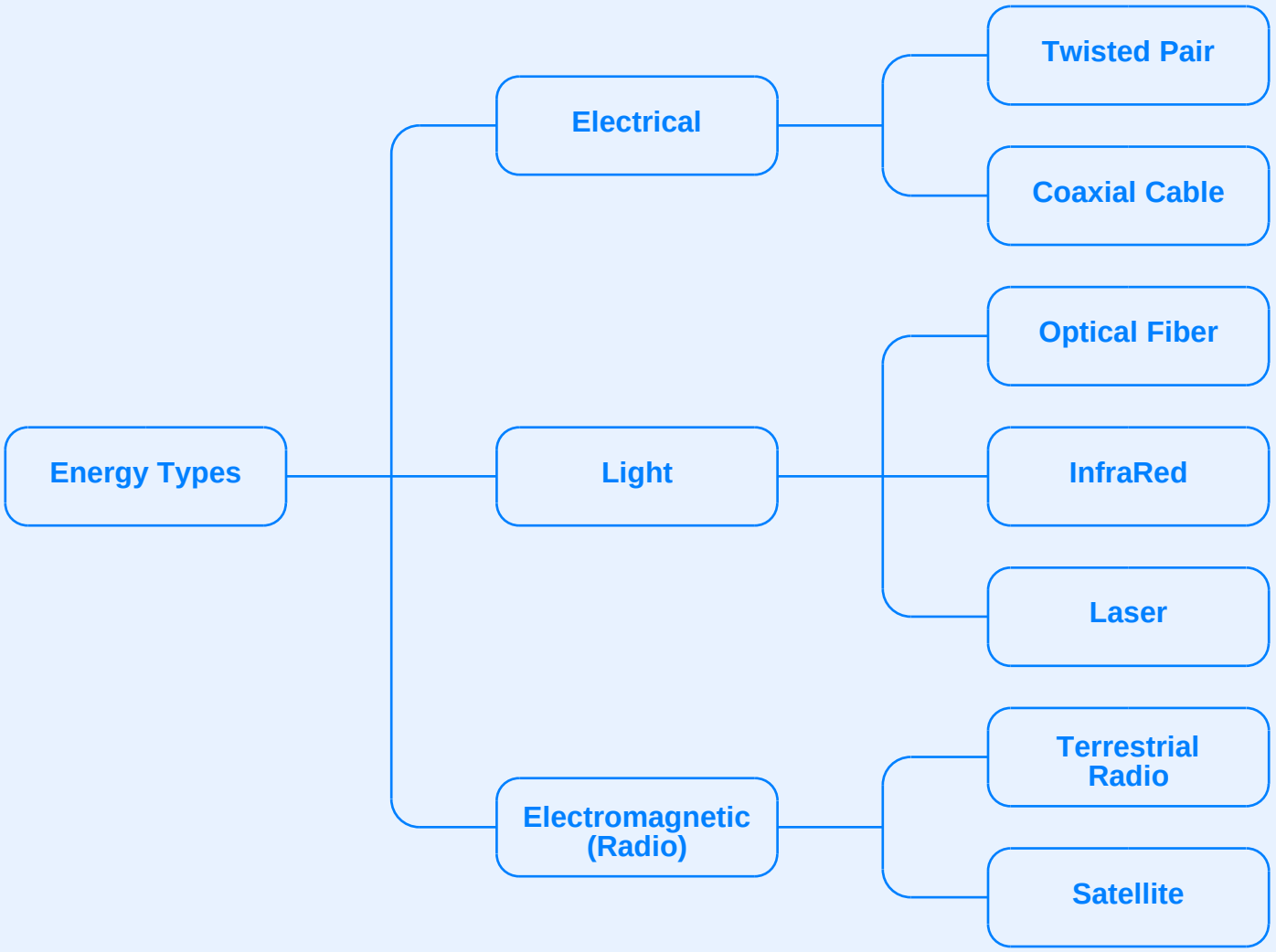
Example Line Coding: Manchester Encoding

- Used with Ethernet
- Synchronizes receiver with sender (transition represents bit)
- Example of (a) Manchester Encoding, and (b) differential Manchester Encoding:



Transmission Media

A Taxonomy Of Transmission Media



- Is anything omitted?

Some Really Bad News

- In the real world, entropy rules
- Transmission is plagued with problems

Loss, Interference, And Electrical Noise

- Problems in the electrical and electromagnetic worlds
  - Resistance (leads to loss)
  - Capacitance (leads to distortion)
  - Inductance (leads to interference)
- Random electromagnetic radiation is called *noise*
  - Can be generated by specific sources such as electric motor
  - Background radiation is an inescapable feature of the universe

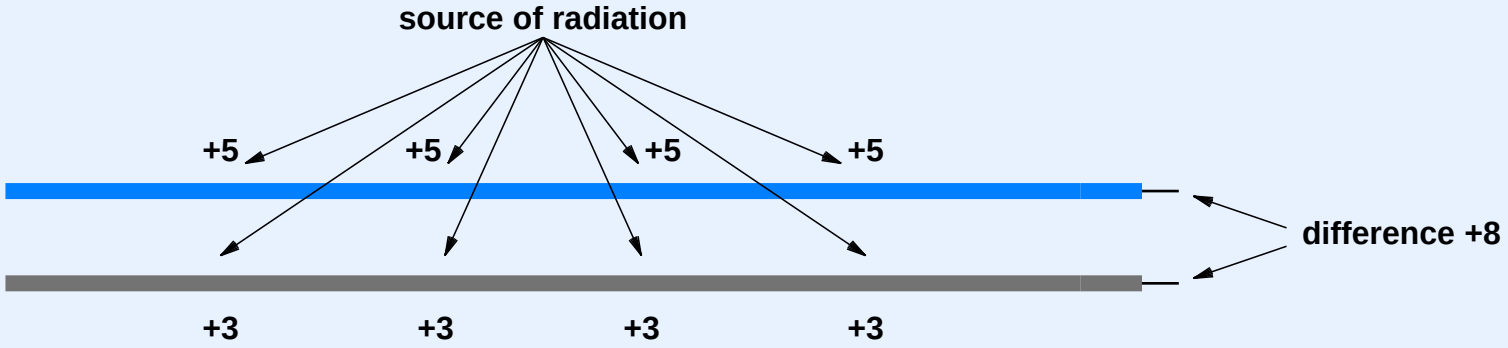
Examples

- When electrical signals propagate down a wire, electromagnetic energy is radiated (i.e., the wire acts like an antenna)
- When electromagnetic radiation encounters metal, a small electrical current is induced that can interfere with signals being carried on the wire
- When an electrical pulse is sent down an unterminated wire, reflection comes back
- When a signal passes across the connection between two wires, reflection and loss occur
- Note: a network diagnostic tool uses reflection to find the distance to the point where a cable has been cut

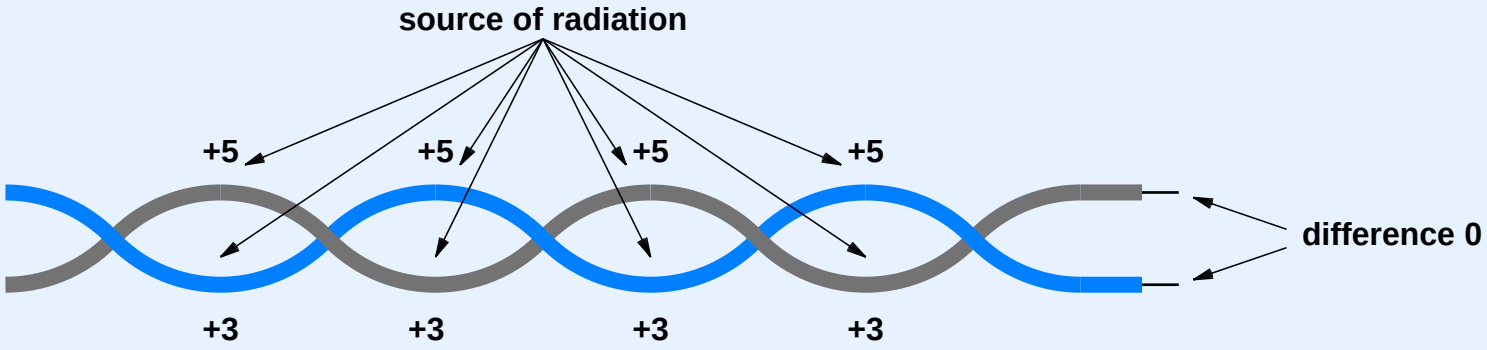
How Can We Reduce The Effect Of Noise On Copper Wiring

- Several techniques have been invented
  - Unshielded Twisted Pair (UTP)
  - Coaxial cable
  - Shielded Twisted Pair (STP)
- All are used in computer networks

How Twisted Pair Helps



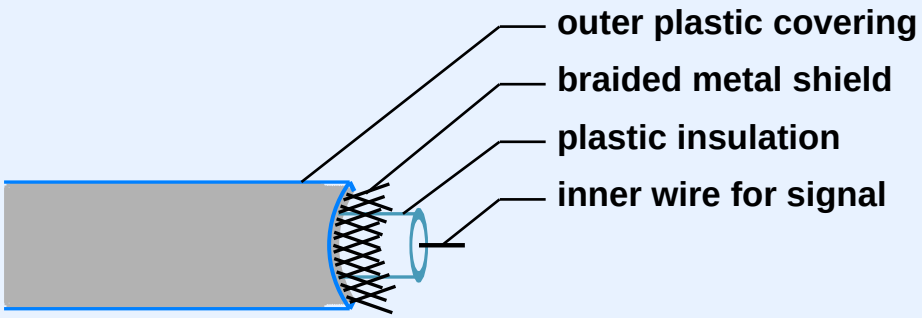
- In an untwisted pair of wires, more current is generated in first wire the interference hits



- Twisting exposes each wire equally

Coaxial Cable And Shielding

- Better protection: wrap a metal shield around the wire



- Shielding can be added to twisted pair
  - Around entire cable containing many pairs
  - Around each pair as well as around cable
- Shielding determines maximum data rate

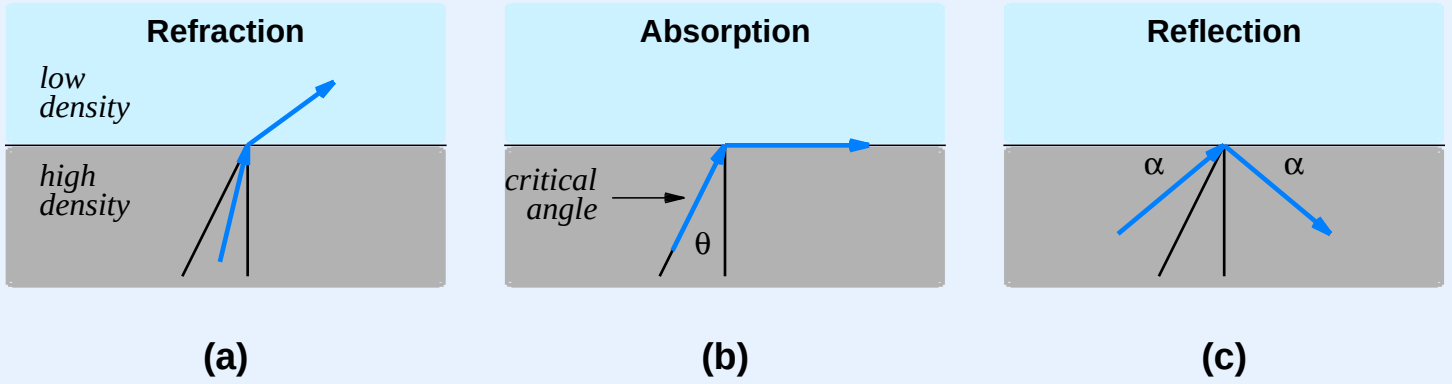
Wiring Standards And Data Rates

Category	Description	Data Rate (in Mbps)
CAT 1	Unshielded twisted pair used for telephones	< 0.1
CAT 2	Unshielded twisted pair used for T1 data	2
CAT 3	Improved CAT2 used for computer networks	10
CAT 4	Improved CAT3 used for Token Ring networks	20
CAT 5	Unshielded twisted pair used for networks	100
CAT 5E	Extended CAT5 for more noise immunity	125
CAT 6	Unshielded twisted pair tested for 200 Mbps	200
CAT 7	Shielded twisted pair with a foil shield around the entire cable plus a shield around each twisted pair	600

- What common data rate is missing from the list?

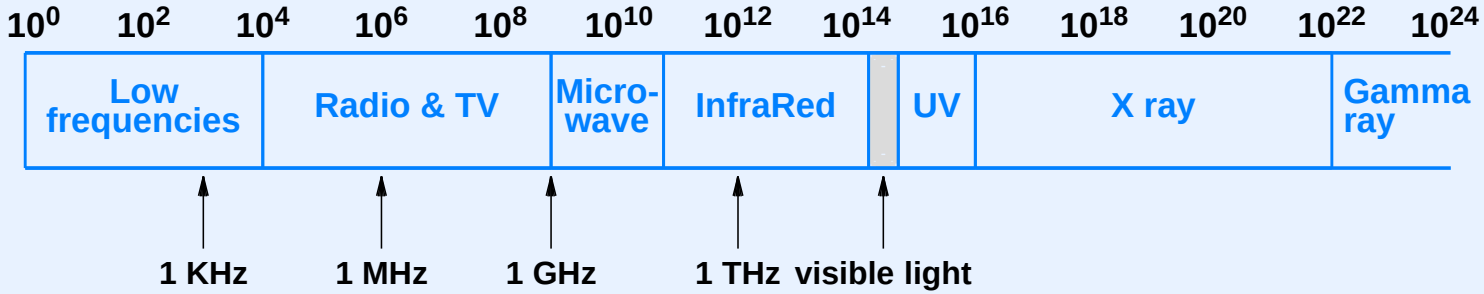
Media Using Light Energy

- InfraRED transmission (short range and low data rate)
- Point-to-point lasers (useful between buildings)
- Optical fiber (high data rate and long distance)
- Why light stays in a fiber:



(a) (b) (c)

Electromagnetic Spectrum And Properties



Classification	Range	Type Of Propagation
Low Frequency	< 2 MHz	Wave follows earth's curvature, but can be blocked by unlevel terrain
Medium Frequency	2 to 30 MHz	Wave can reflect from layers of the atmosphere, especially the ionosphere
High Frequency	> 30 MHz	Wave travels in a direct line, and will be blocked by obstructions

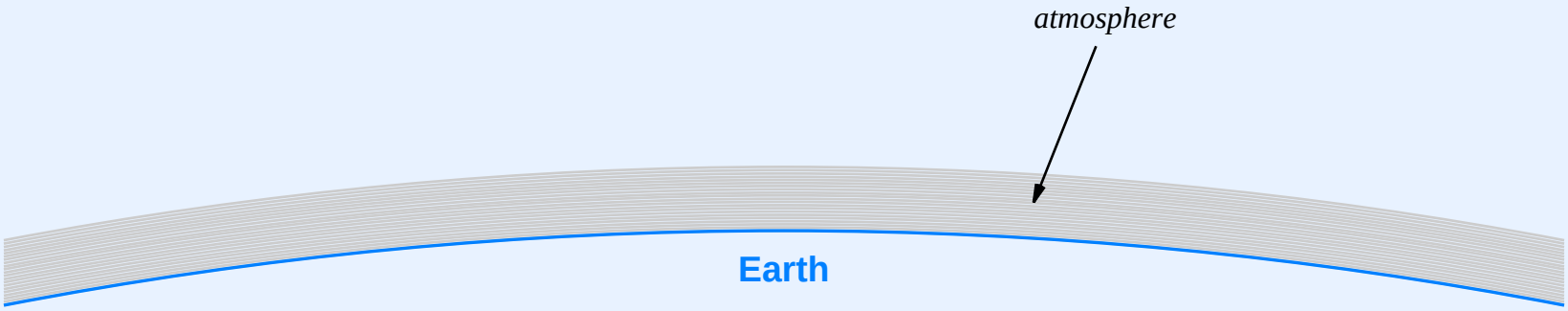
Satellite Communication

- Three types of communication satellites

Orbit Type	Description
Low Earth Orbit (LEO)	Has the advantage of low delay, but the disadvantage that from an observer's point of view on the earth, the satellite appears to move across the sky
Medium Earth Orbit (MEO)	An elliptical (rather than circular) orbit primarily used to provide communication at the North and South Poles
Geostationary Earth Orbit (GEO)	Has the advantage that the satellite remains at a fixed position with respect to a location on the earth's surface, but the disadvantage of being farther away

GEO Satellites

- Figure below shows the earth's atmosphere drawn to scale
- Where would a GEO satellite be in the figure?



GEO Satellites
(continued)

- Distance to GEO satellite is 35,785 km or 22,236 miles
- Approximately 3 times earth's diameter or one-tenth of the distance to the moon
- In other words: the satellite is far off the page
- A consequence for networking: a long round-trip time, even at the speed of light:

$$Round\ trip\ time\ =\ \frac{2\ \times\ 35.8 \times 10^{6} meters}{3\ \times\ 10^{8}\ meters/sec}\ =\ 0.238\ sec$$

Measures Of Transmission Media

- *Propagation delay* - time required for a signal to traverse a medium
- *Channel capacity* - maximum data rate

Channel Capacity

- Nyquist's Theorem gives theoretical bound on maximum data rate for hardware bandwidth B and K signal levels

$$D = 2 B \log_2 K$$

- Mathematical result known as *Shannon's Theorem* gives the maximum channel capacity, C, in the presence of noise

$$C = B \log_2 (1 + S/N)$$

- Quantity S/N is known as the *signal-to-noise ratio*

Assessment

- Nyquist's Theorem gives us hope: using more signal levels can increase the data rate
- Shannon's Theorem is sobering: electrical noise in the universe limits the effective channel capacity of any practical communication system

Reliability And Channel Coding

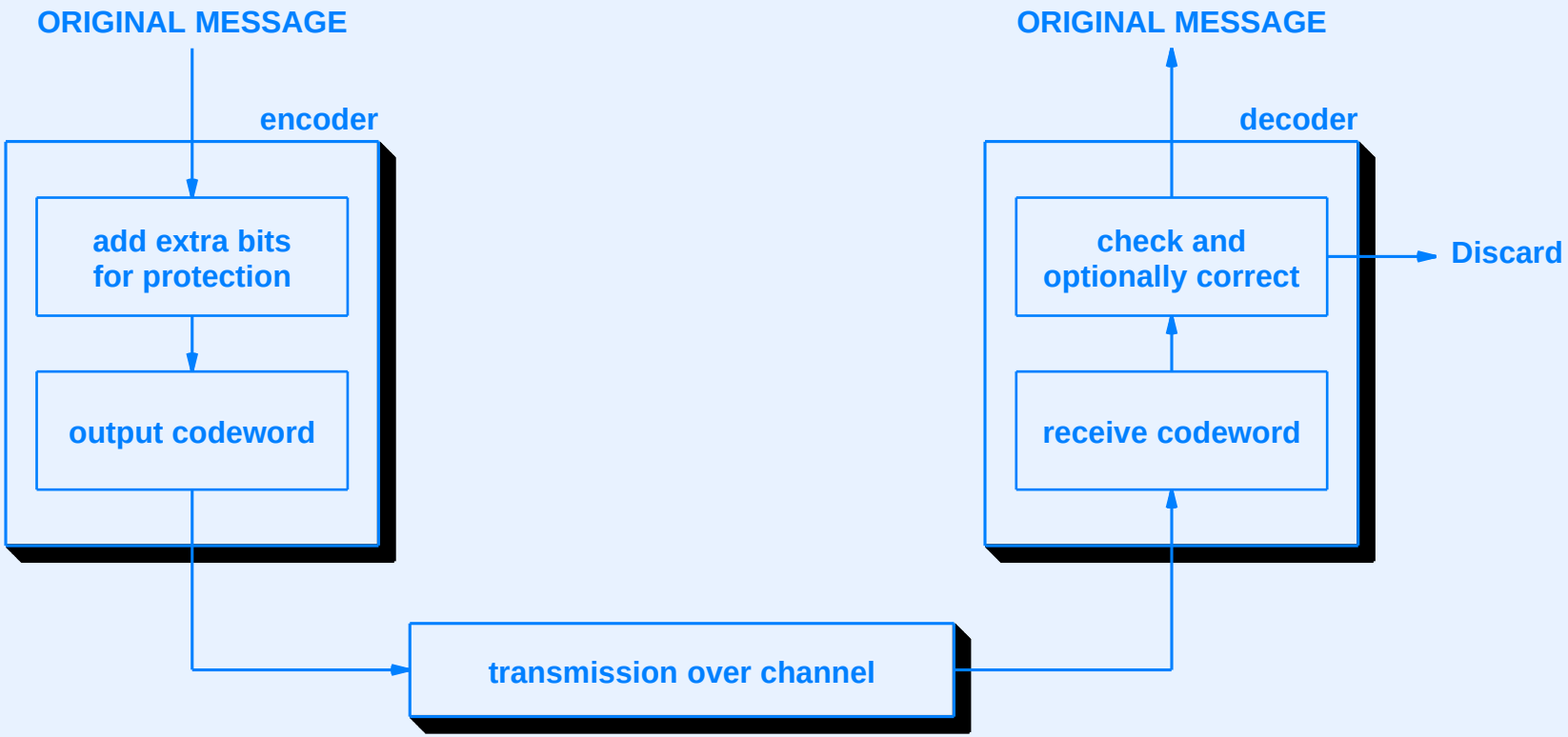
Sources Of Errors And Types

- Error sources: interference, distortion, and attenuation
- Resulting error types:

Type Of Error	Description
Single Bit Error	A single bit in a block of bits is changed and all other bits in the block are unchanged (often results from very short-duration interference)
Burst Error	Multiple bits in a block of bits are changed (often results from longer-duration interference)
Erasure (Ambiguity)	The signal that arrives at a receiver is ambiguous (does not clearly correspond to either a logical 1 or a logical 0; can result from distortion or interference)

- Channel coding used to detect and correct errors

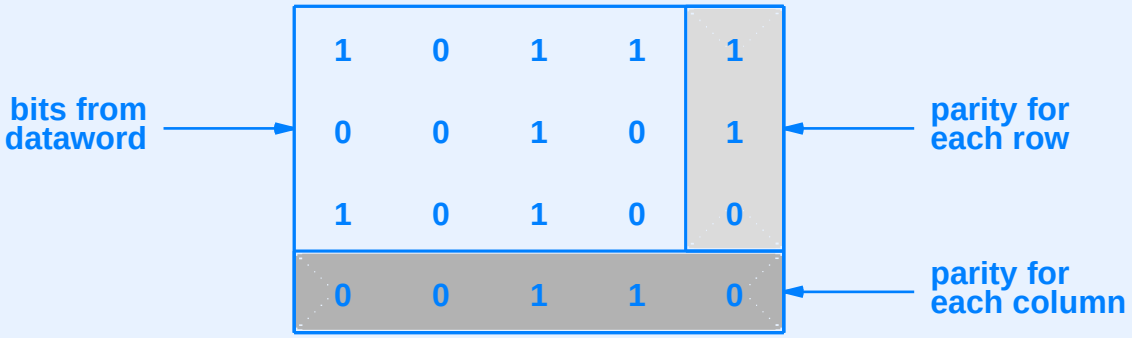
Concept Of Forward Error Correction (FEC)



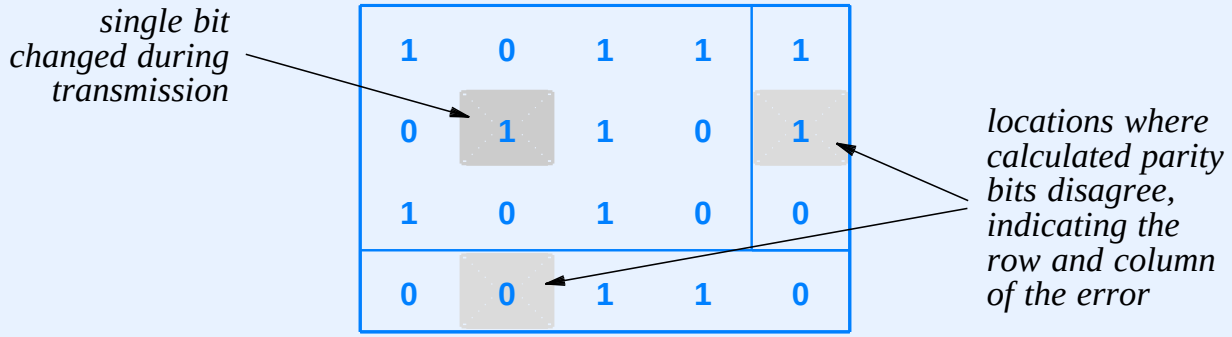
- Examples:
  - Single parity bit
  - Row And Column (RAC)
  - Cyclic Redundancy Check (CRC)

Example: Row And Column Code

- To send 12 bits, arrange the bits in a matrix, compute a parity for each row and column, and send 20 bits



- Receiver computes same parity for the 12 bits and compares to the parity bits received



Hamming Distance

- Used to assess code's resistance to errors
- Defined to be number of bit changes to transform bit string S_1 into bit string S_2
- Can be computed as number of 1 bits in the *exclusive or* of S_1 and S_2
- To assess code's strength, compute Hamming distance among all possible pairs of codewords, and take the minimum
- If minimum Hamming distance is *n*, an error that changes fewer than *n* bits will be detected

Internet Checksum Computation

Given:

A message, M, of arbitrary length

Compute:

A 16-bit 1s complement checksum, C

Method:

```
Pad M to an exact multiple of 16 bits;
Set a 32-bit checksum integer, C, to zero;
for (each 16-bit group in M) {
    Treat the 16 bits as an integer and add to C;
}
Extract high-order 16 bits of C and add to C;
Checksum is inverse of the low-order 16 bits;
If the checksum is zero, substitute all 1s;
```

Cyclic Redundancy Code (CRC)

- Used with Ethernet and other high-speed networks
- Properties:

Arbitrary Length Message	**As with a checksum, the size of a dataword is not fixed, which means a CRC can be applied to an arbitrary length message**
Excellent Error Detection	**Because the value computed depends on the sequence of bits in a message, a CRC provides excellent error detection capability**
Fast Hardware Implementation	**Despite its sophisticated mathematical basis, a CRC computation can be carried out extremely fast by hardware**

Explanation Of CRC

- Mathematicians explain CRC computation as the remainder from polynomial division
- Theoretical computer scientists explain CRC as the remainder from a division of binary numbers
- Cryptographers explain CRC as an operation in a Galois field of order 2
- Computer programmers explain CRC as an algorithm that iterates through a message and uses table lookup
- Hardware architects explain CRC computation as a small hardware pipeline unit that uses *exclusive or*

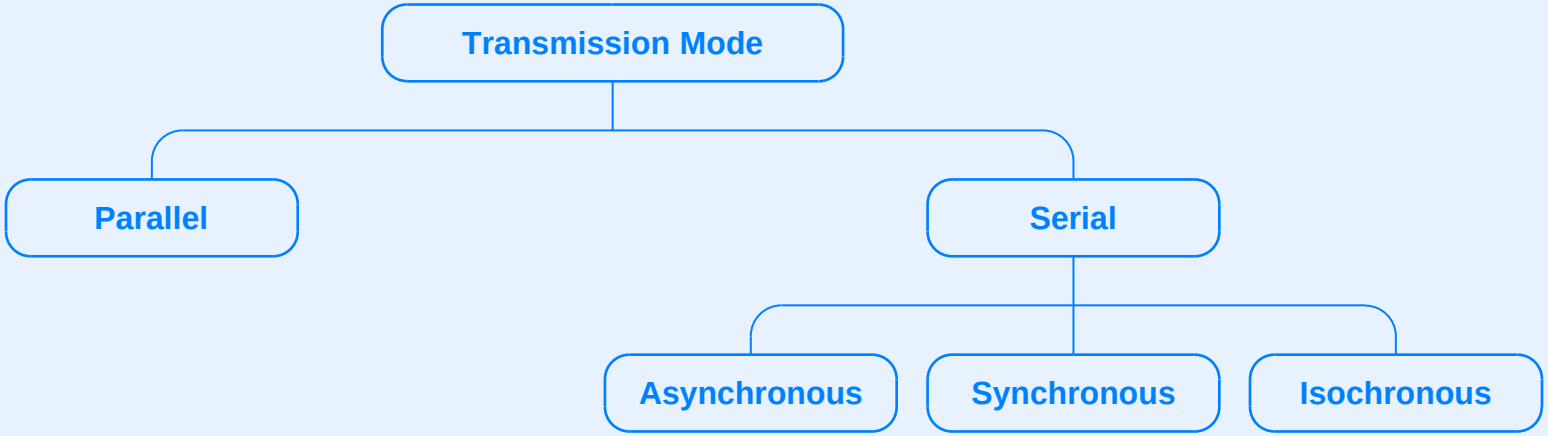
Question

- Can you explain the following?
  - Fact 1: it is possible to write a function that computes the 32-bit CRC used with Ethernet
  - Fact 2: commercial Ethernet products use hardware instead of software to compute a CRC

Transmission Modes

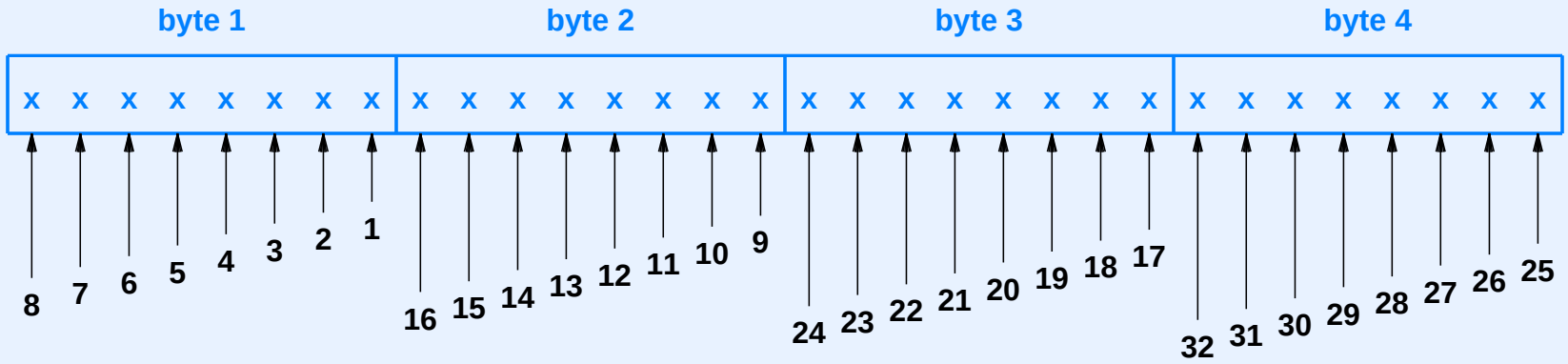
Terminology

- Serial - one bit at a time
- Parallel - multiple bits at a time
- Taxonomy of transmission methods:



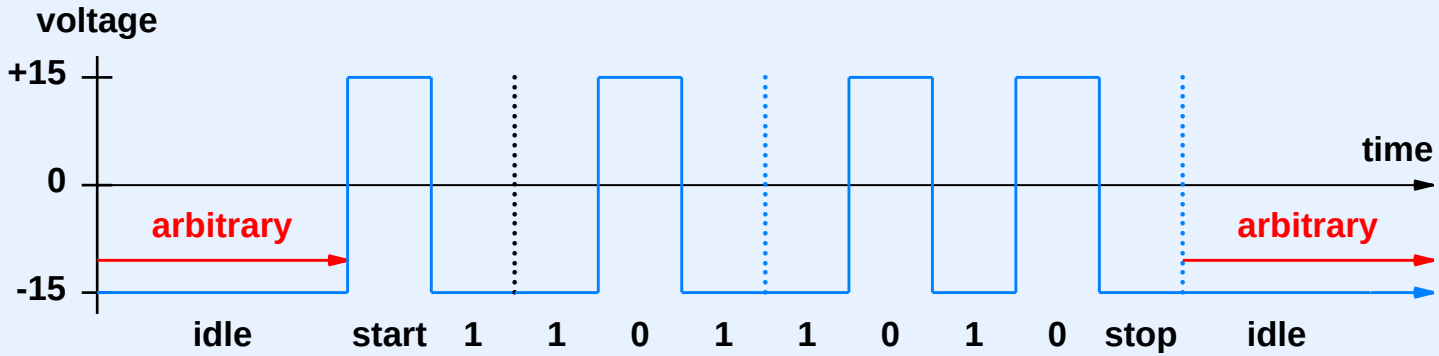
Serial Ordering Of Bits And Bytes

- Both sides must agree on order in which bits are transmitted
- Two approaches known as *big-endian* and *little-endian*
- Example: Ethernet uses byte big-endian and bit little-endian order



Asynchronous And Synchronous Transmission

- Asynchronous: line idle when not in use; data starts at arbitrary time



- Synchronous: each bit slot used

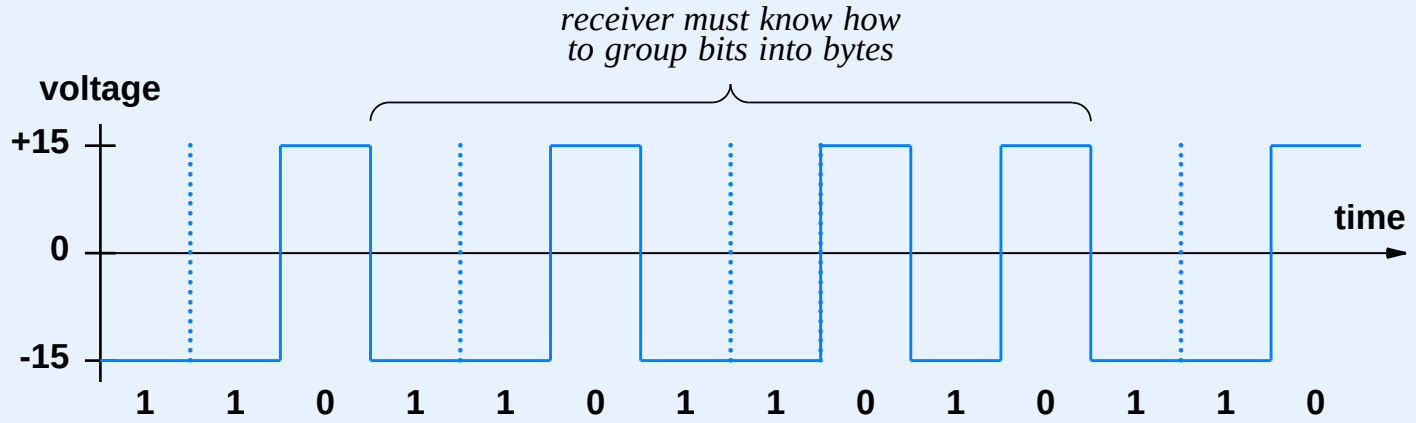
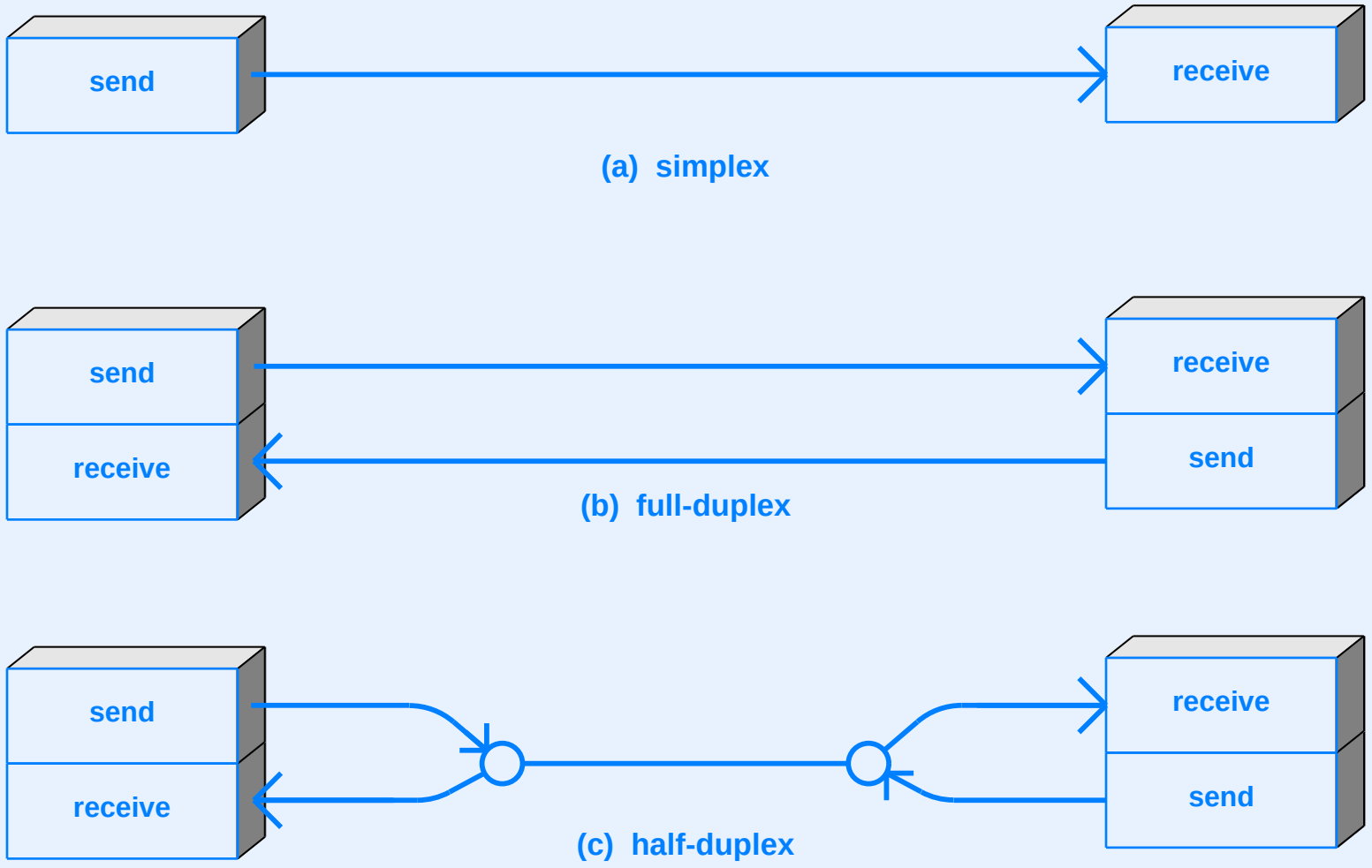


Illustration Of Simplex And Duplex Modes



Modulation And Demodulation

Illustration Of Amplitude Modulation

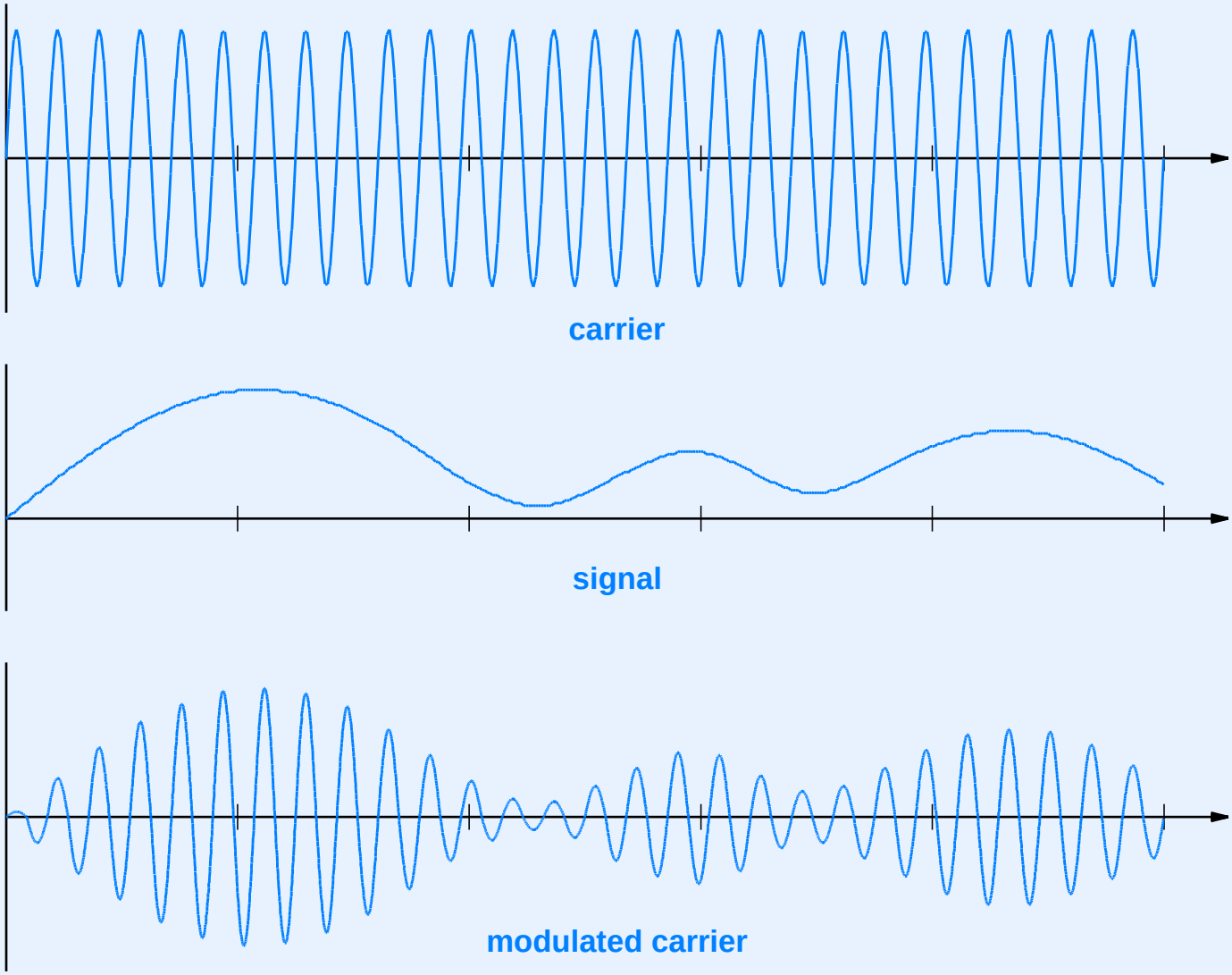
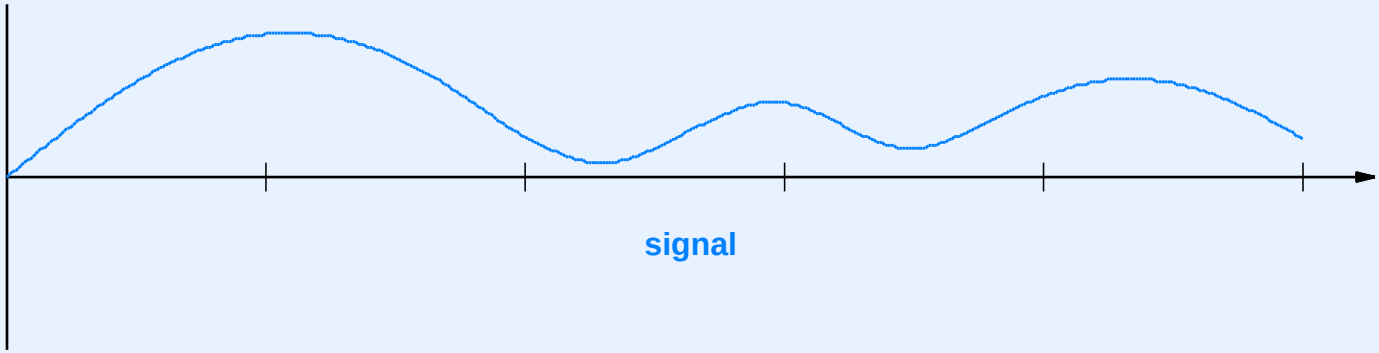
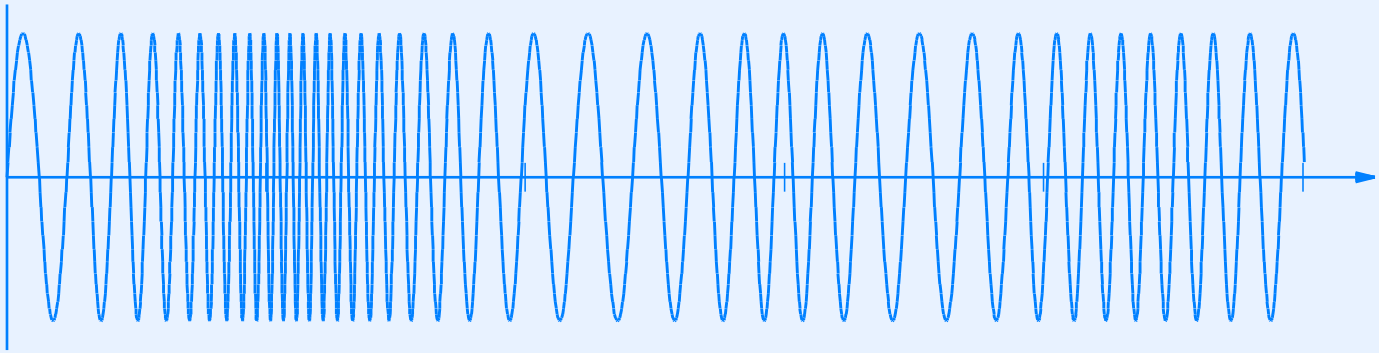


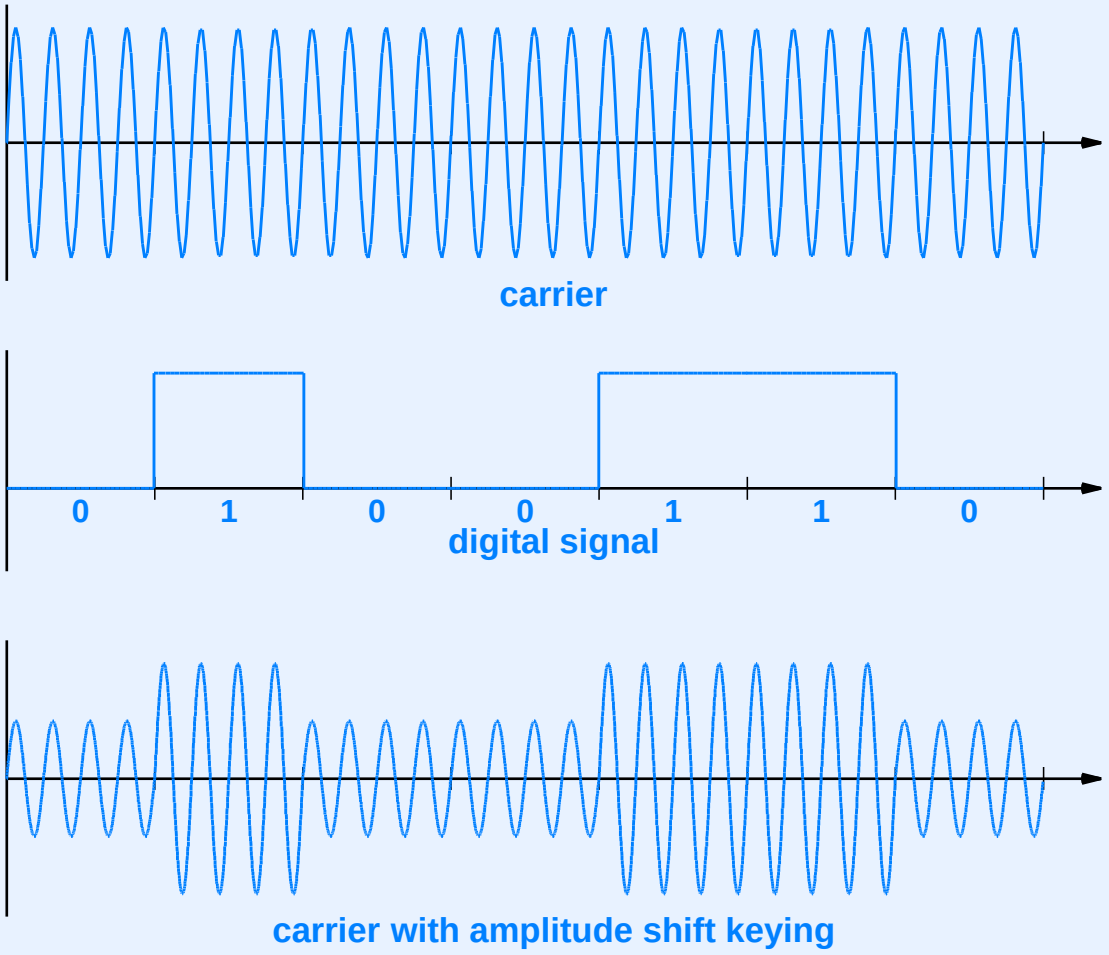
Illustration Of Frequency Modulation





Shift Keying

- Like modulation except signal is digital



A Challenge

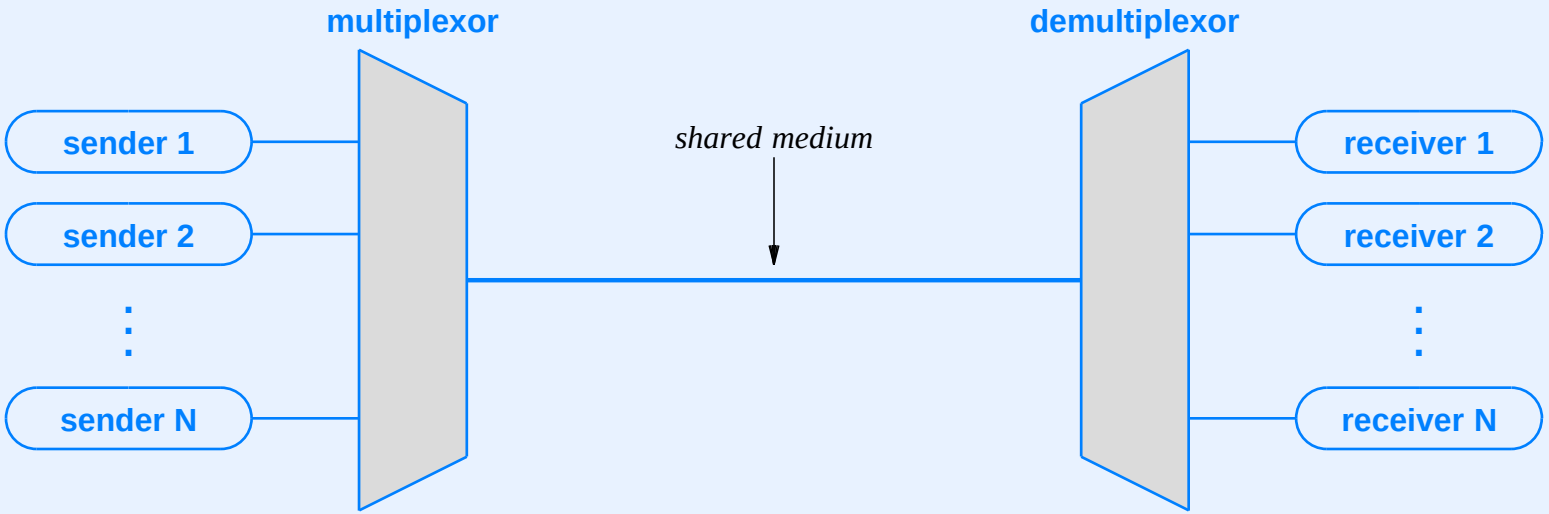
Write a computer program that takes as input a series of points defining a signal and produces plots of sine waves that show amplitude and frequency modulation as in the previous diagrams

Other Modulation Topics

- Phase shift modulation
- Increasing bits per second by combining amplitude and phase shift (QAM techniques)
- Constellation diagrams to represent combinations
- Modems (modulator / demodulator)

Multiplexing And Demultiplexing (Channelization)

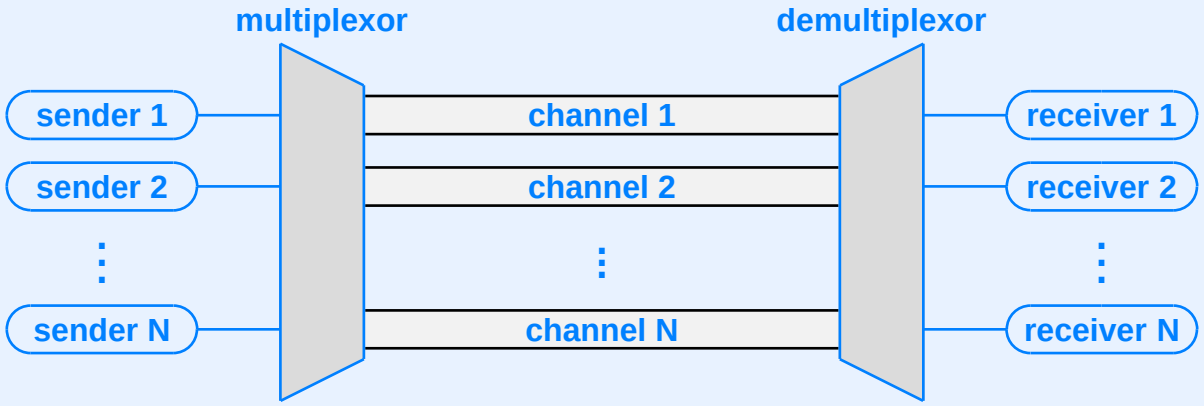
Concept Of Multiplexing And Types



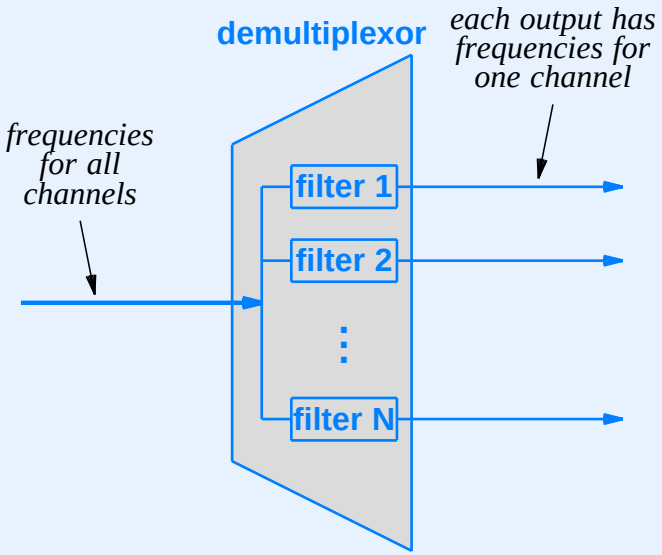
- Types:
  - Frequency division multiplexing
  - Wavelength division multiplexing
  - Time division multiplexing
  - Code division multiplexing

Frequency Division Multiplexing (FDM)

- Used in broadcast radio and cable TV



- Demultiplexing implemented with sets of filters

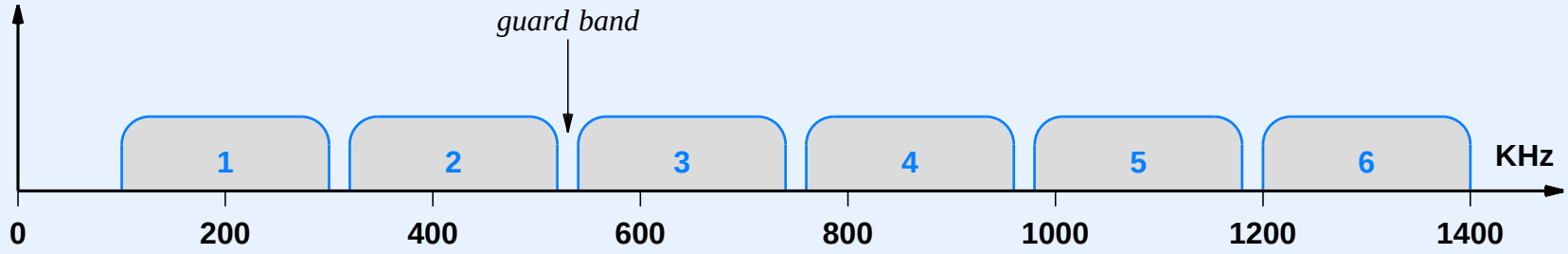


FDM In Practice

- Each channel assigned a range of frequencies

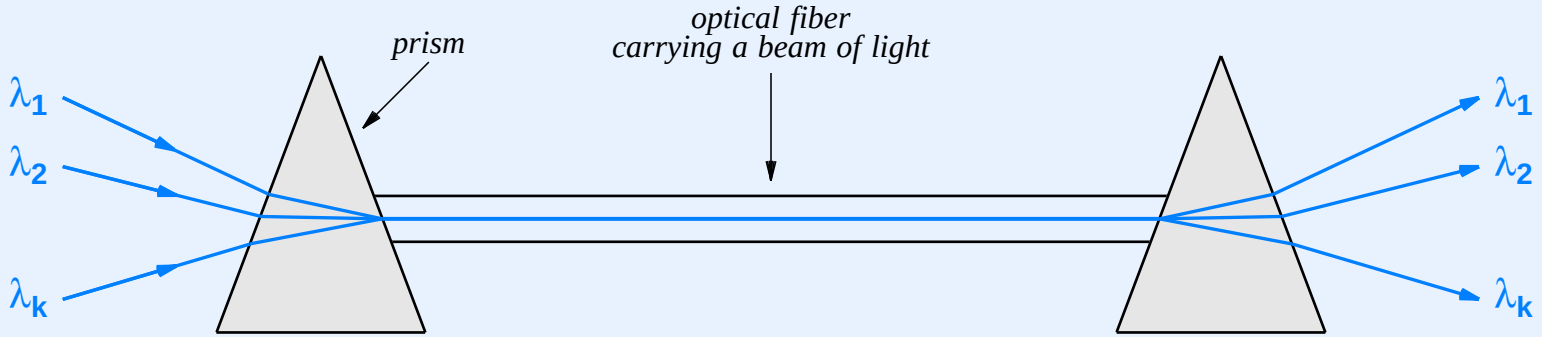
Channel	Frequencies Used
1	100 KHz - 300 KHz
2	320 KHz - 520 KHz
3	540 KHz - 740 KHz
4	760 KHz - 960 KHz
5	980 KHz - 1180 KHz
6	1200 KHz - 1400 KHz

- A *guard band* separates adjacent channels



Wavelength Division Multiplexing (WDM)

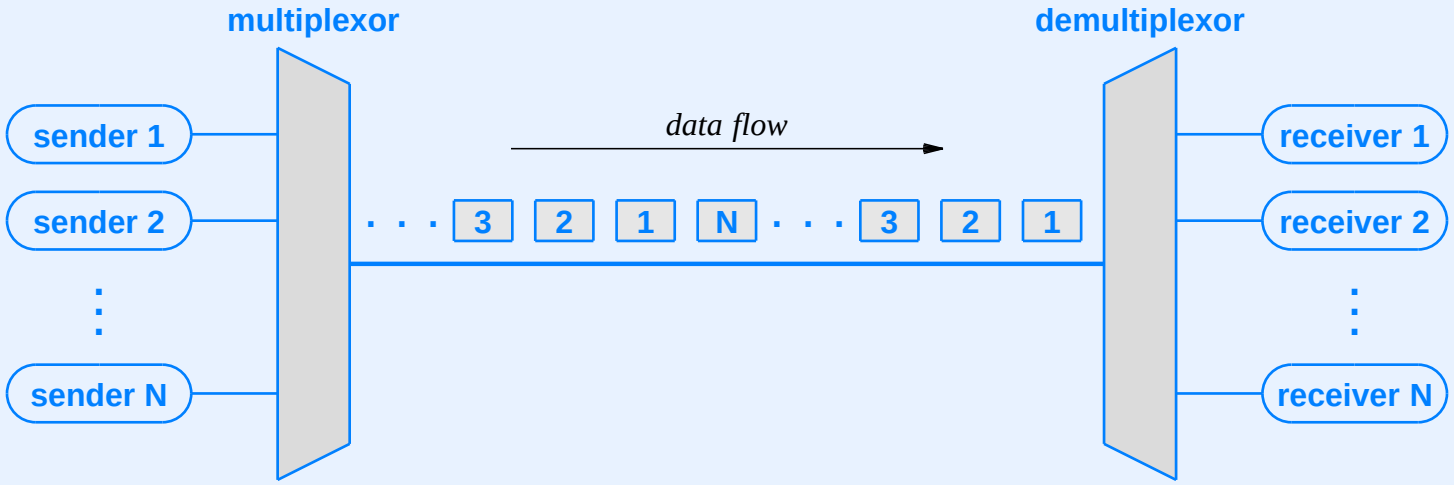
- Form of FDM used with light (i.e., on an optical fiber)
- Separate frequencies called *colors* or *lambdas*
- Prisms used to separate frequencies



- Current technology is *Dense WDM* (*DWDM*); an individual channel can provide 10 Gbps

Time Division Multiplexing

- Senders take turns transmitting



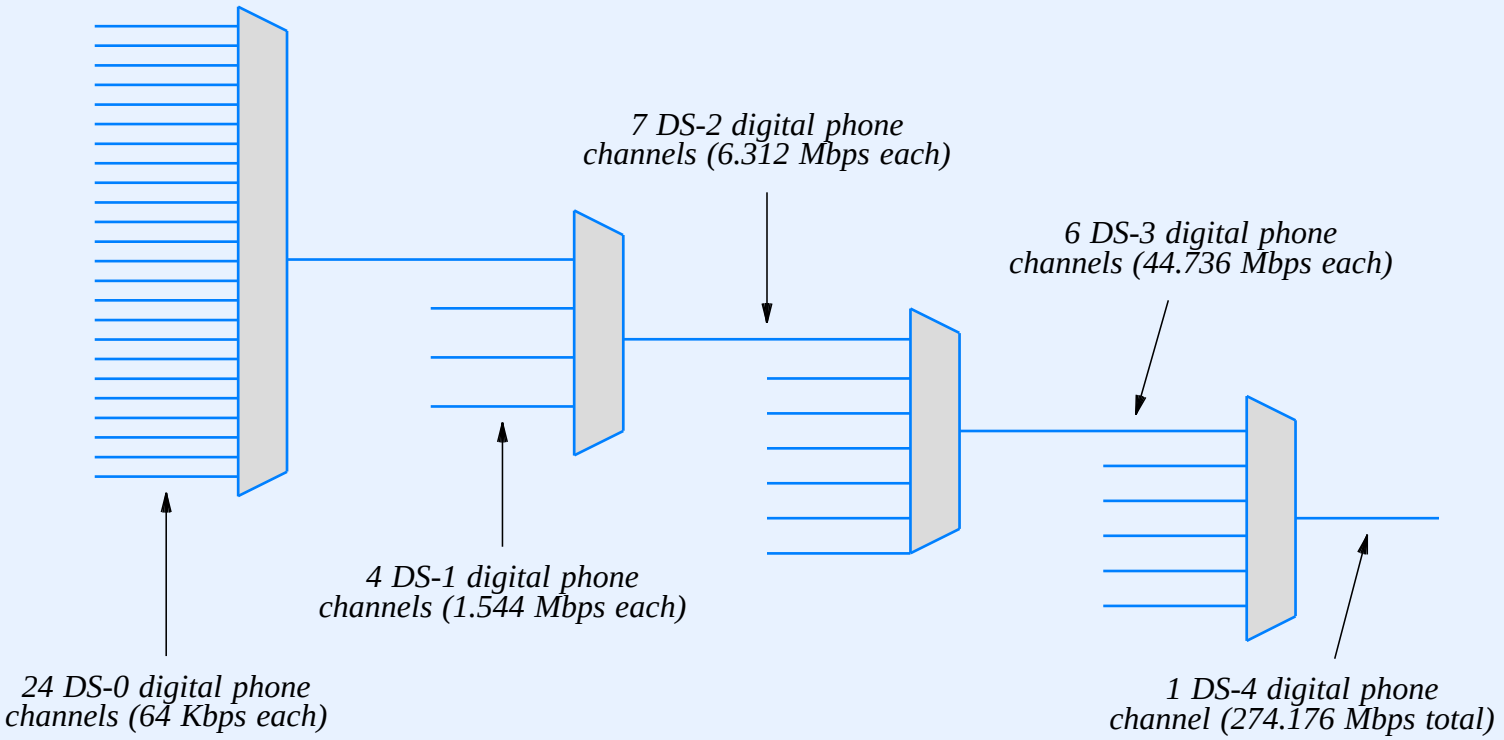
- Synchronous TDM
  - Each sender assigned a slot (typically round-robin)
  - Used by the telephone company
- Statistical TDM
  - Sender only transmits when ready (e.g., Ethernet)

Code Division Multiplexing

- Mathematical form of multiplexing used with cell phones
- Algorithm
  - Each sender/receiver pair is assigned a unique number called a *chip sequence*
  - Senders multiply the data value by their chip sequence (orthogonal vector spaces)
  - Transmitted value is a sum of all senders
  - Each receiver multiplies incoming value by its chip sequence to extract data
- Advantage over statistical TDM: lower delay when network loaded

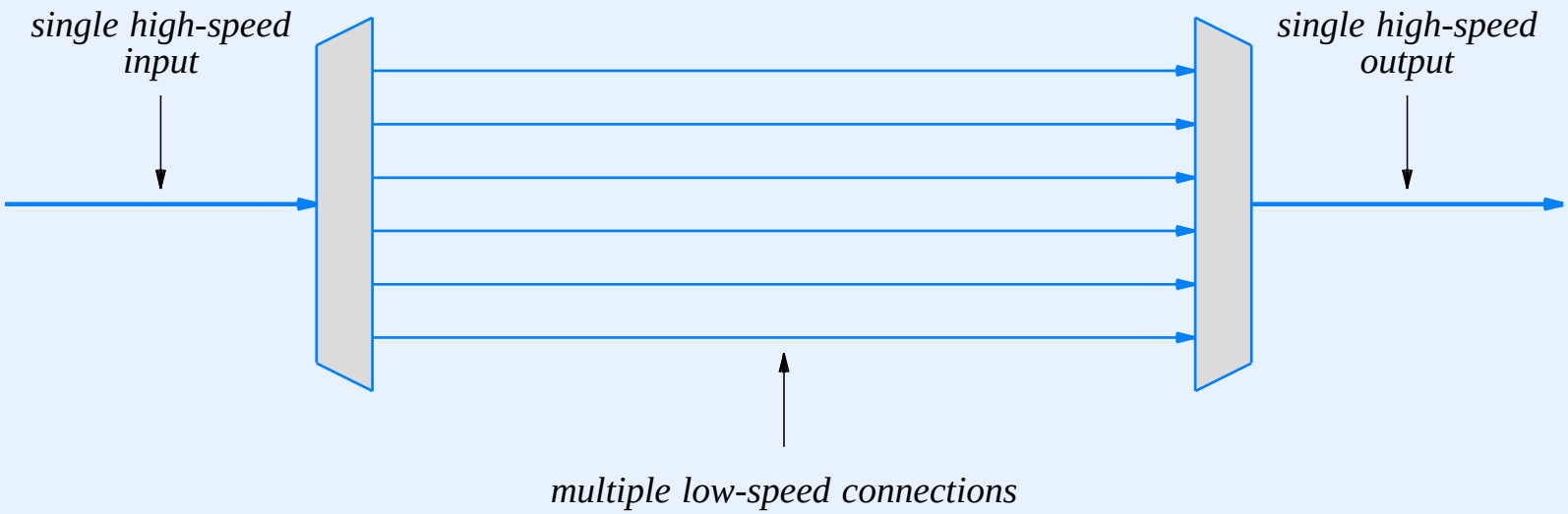
Hierarchical Multiplexing

- Hierarchies used with FDM and TDM to combine multiple lower-capacity channels
- Example of TDM hierarchy used by the phone system



Inverse Multiplexing

- Divides data from a single channel into several lower-speed channels
- Used when high-speed channel is unavailable or too expensive
- Some ISPs use inverse multiplexing to combine several 10 Gbps channels into a higher-speed channel



Summary

- Data communications deals with the Physical Layer and data transmission
- Concepts include
  - Signals and conversion between digital and analog
  - Transmission media
  - Reliability and channel coding
  - Modulation and demodulation
  - Multiplexing and demultiplexing

MODULE IV

Computer Network Technologies: Access, Wired And Wireless LANs, Extensions, Bridging, And Layer 2 Switching

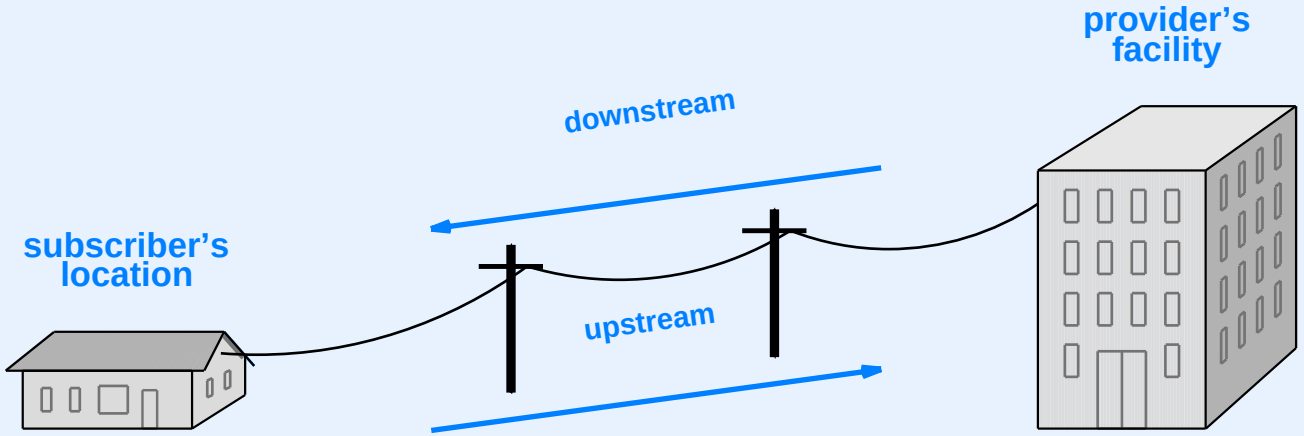
Topics

- Access technologies
- Interconnection technologies
- Local area network packets, frames, and topologies
- Media access mechanisms and the IEEE MAC sub-layer
- Wired LAN technologies (Ethernet and 802.3)
- Wireless Networking Technologies
- LAN Extensions
- Switches and switched networks

Access Technologies

Definition Of Access

- Used in the “last mile” between a provider and a subscriber
- Informally classified as either *narrowband* or *broadband*
- May not be the bottleneck
- Many are asymmetric with higher data rate *downstream*



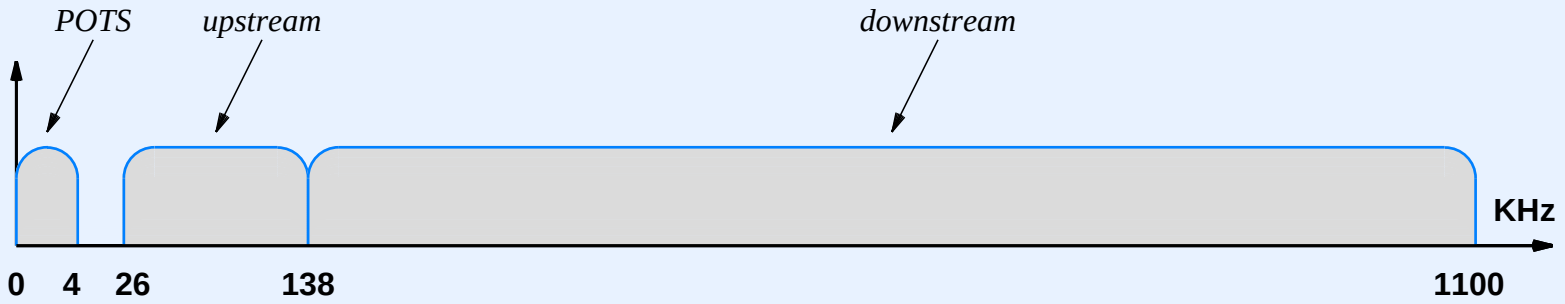
- Note: party that is downstream pays a fee for service

Access Technology Types

- Narrowband (less than 128 Kbps)
  - Dialup
  - Integrated Services Digital Network (ISDN)
  - Is disappearing
- Broadband (more than 128 Kbps)
  - Digital Subscriber Line (DSL)
  - Cable modems
  - Wireless (e.g., Wi-Fi and 4G)

Digital Subscriber Line (DSL) Technologies

- Use frequency-division multiplexing to share *local loop* between data and *POTS*
- Head-end equipment is DSL Access Multiplexor (DSLAM)
- Asymmetric Digital Subscriber Line (ADSL)
  - 255 downstream carrier frequencies, 31 upstream
  - Maximum downstream data rate is 8.45 Mbps
  - Adaptive selection of carrier frequencies



Cable Modem Technology

- Sends data over CATV coaxial cable system
- Standard is DOCSIS (Data-Over-Cable Service Interface Specification)
- Head-end equipment known as Cable Modem Termination System (CMTS)
- Version 1.x uses frequency-division multiplexing
- Maximum downstream data rate is 52 Mbps
- Bandwidth shared among multiple subscribers
  - Each subscriber receives $\frac{1}{N}$ of the bandwidth
  - Cable company chooses N

Other Access Technologies

- Hybrid systems include optical fiber plus copper
  - Fiber To The Curb (FTTC)
  - Fiber To The Building (FTTB)
  - Fiber To The Premises (FTTP)
  - Fiber To The Home (FTTH)
- Key question: how much capacity is needed at each point downstream?
- Answer: it depends on whether endpoints have traffic in common
  - Broadcasts are shared
  - Individual communications are not

Other Access Technologies (continued)

- Wireless
  - Wi-Fi
  - WIMAX
  - Satellite
  - 3G and 4G cellular services
- Leased point-to-point circuits (e.g., T1 or fractional T1)

Interconnection Technologies

Interconnections At The Core Of The Internet

- Typically needed by large ISPs
- Circuits leased from common carriers (phone companies)
- Terminated with a *Data Service Unit/Channel Service Unit* (*DSU/CSU*)
- Upstream interface aggregates many lower-speed access connections
- Key idea: data rates based on voice
  - Basic data rate: single digital voice channel (64 Kbps)
  - Higher data rate circuits created from multiples of voice channels
- SONET encoding and framing used

Example Data Rates Of Leased Circuits

Name	Bit Rate	Voice Circuits	Location
basic rate	0.064 Mbps	1	
T1	1.544 Mbps	24	North America
T2	6.312 Mbps	96	North America
T3	44.736 Mbps	672	North America
E1	2.048 Mbps	30	Europe
E2	8.448 Mbps	120	Europe
E3	34.368 Mbps	480	Europe

- T-standards used in North America
- E-standards used in Europe
- Note: T prefix specifies encoding as well as data rate; data rate alone is given by Digital Signal Level (DS) standards

High Capacity Data Circuits

Copper Name	Optical Name	Bit Rate	Voice Circuits
STS-1	OC-1	51.840 Mbps	810
STS-3	OC-3	155.520 Mbps	2430
STS-12	OC-12	622.080 Mbps	9720
STS-24	OC-24	1,244.160 Mbps	19440
STS-48	OC-48	2,488.320 Mbps	38880
STS-192	OC-192	9,953.280 Mbps	155520

- STS standards specify copper interface
- OC standards specify optical fiber interface
- Suffix C on OC-standards means single channel

Local Area Networks:
(Packets, Frames, Topologies)

Networks

- Distinct from physical communication systems
- Attach multiple endpoints
- Two broad categories
  - *Circuit switched*
  - *Packet switched*

Circuit Switched Networks

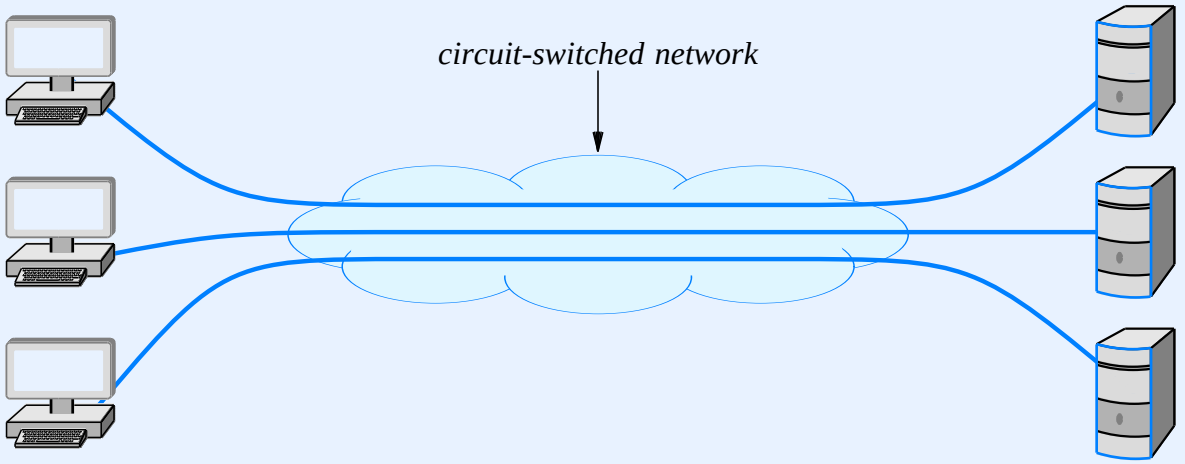
- Provide point-to-point communication between pairs of endpoints
- Establish path between sender and receiver
- Separate steps for circuit creation, use, and termination
- Performance equivalent to an isolated physical path
- Circuit can be
  - Permanent/ provisioned (left in place for long periods)
  - Switched (created on demand)
- Concept: user leases piece of underlying infrastructure for a time period

Packet Switched Networks

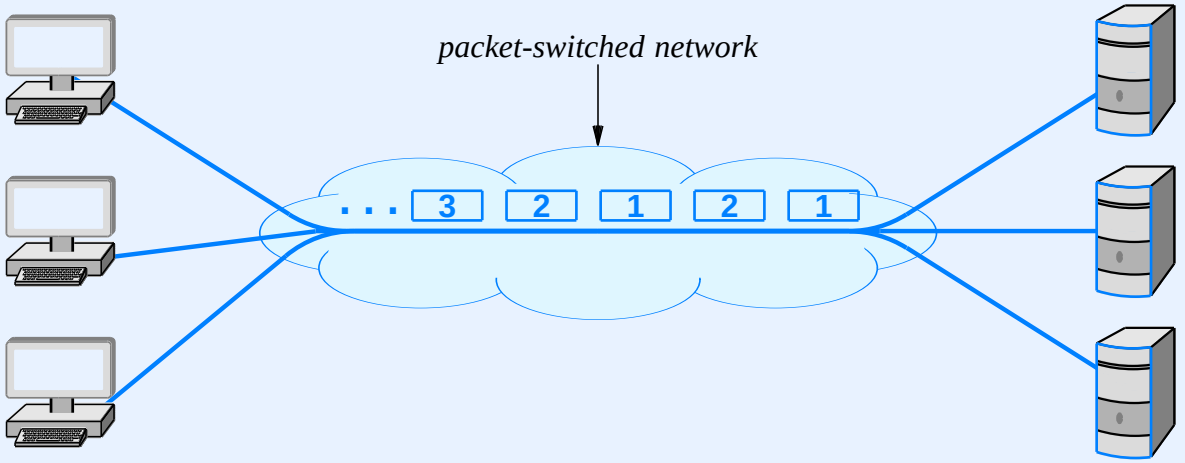
- Form the basis for the Internet
- Multiplex communication over shared media
- All data divided into packets (maximum size fixed)
- After sending one packet, sender allows others a chance to transmit before sending a second packet
- Arbitrary, asynchronous communication
- No set-up required before communication begins
- Performance varies due to statistical multiplexing
- Concept: underlying infrastructure is shared among users

Illustration Of Circuit And Packet Switching

- Circuit switching provides 1-to-1 dedicated connections



- Packet switching provides statistical TDM sharing



Categories Of Packet Switched Networks

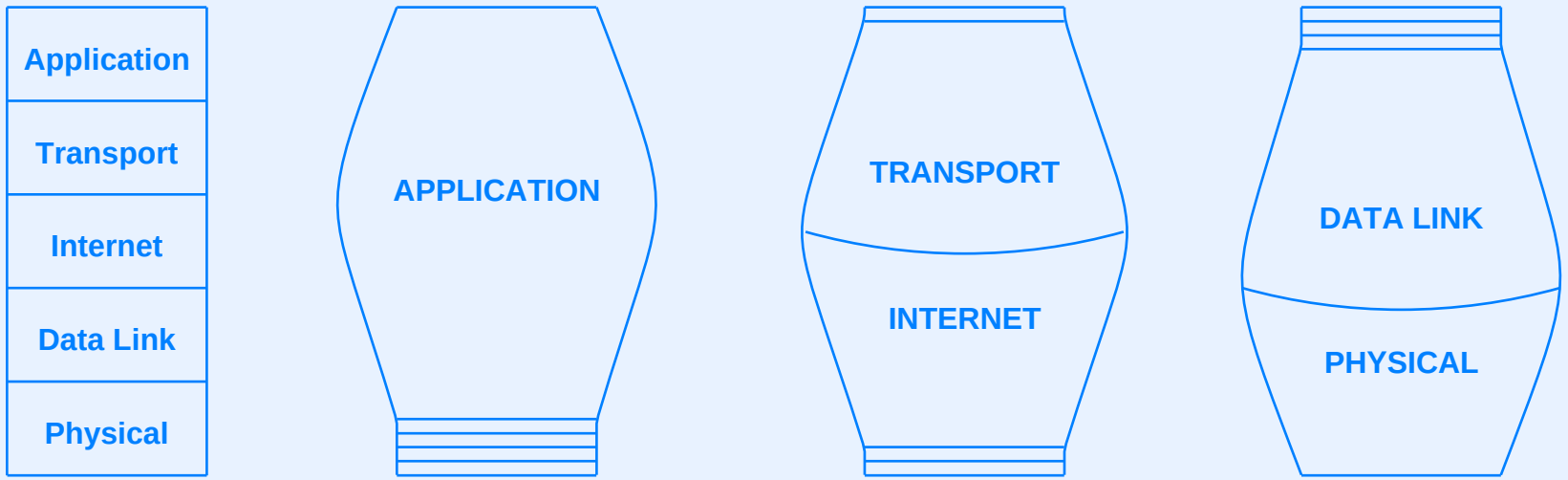
Name	Expansion	Description
LAN	Local Area Network	Least expensive; spans a single room or a single building
MAN	Metropolitan Area Network	Medium expense; spans a major city or a metroplex
WAN	Wide Area Network	Most expensive; spans sites in multiple cities

- Everyone loves names that end in "AN"

Name	Expansion	Description
PAN	Personal Area Network	Spans the area around an individual used for earphones
SAN	Storage Area Network	Spans the distance between a disk farm and processors in a data center
CAN	Chip Area Network	Spans a single chip and connects processor, memories, etc.

Standards Bodies And Their Bias

- Standards bodies and academic departments each emphasize certain layers of a protocol stack, leading to the following views



IEEE 802 Model And Standards

- IEEE (*Institute of Electrical and Electronics Engineers*)
  - Professional society of engineers
  - Standardizes vendor-independent technologies
- Project 802
  - LAN/MAN standards committee
  - Organized in 1980
  - Focuses on layer 1 and layer 2 standards
  - Divides layer 2 into two sublayers
    - *Logical Link Control* (*LLC*)
    - *Media Access Control* (*MAC*)

Example IEEE Standards

ID	Topic
802.1	Higher layer LAN protocols
802.2	Logical link control
802.3	Ethernet
802.4	Token bus (disbanded)
802.5	Token Ring
802.6	Metropolitan Area Networks (disbanded)
802.7	Broadband LAN using Coaxial Cable (disbanded)
802.9	Integrated Services LAN (disbanded)
802.10	Interoperable LAN Security (disbanded)
802.11	Wireless LAN (Wi-Fi)
802.12	Demand priority

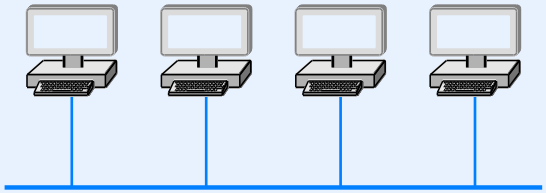
More Example IEEE Standards

ID	Topic
802.13	Category 6 - 10Gb LAN
802.14	Cable modems (disbanded)
802.15	Wireless PAN 802.15.1 (Bluetooth) 802.15.4 (ZigBee)
802.16	Broadband Wireless Access 802.16e (Mobile) Broadband Wireless
802.17	Resilient packet ring
802.18	Radio Regulatory TAG
802.19	Coexistence TAG
802.20	Mobile Broadband Wireless Access
802.21	Media Independent Handoff
802.22	Wireless Regional Area Network

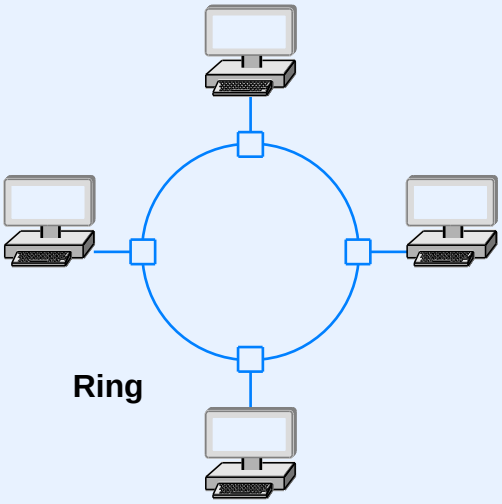
Standards Define

- Network topology (shape)
- Endpoint addressing scheme
- Frame (packet) format
- Media access mechanism
- Physical layer aspects and wiring

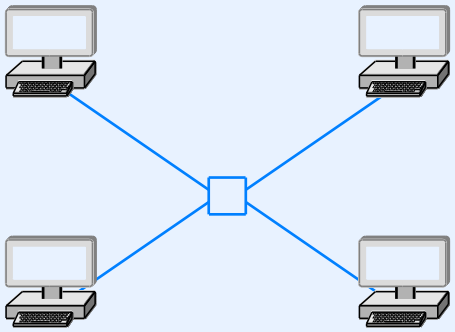
Illustration Of The Four LAN Topologies



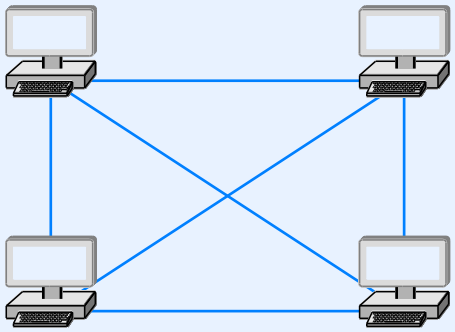
Bus



Ring



Star



Mesh

- Each topology has advantages and disadvantages

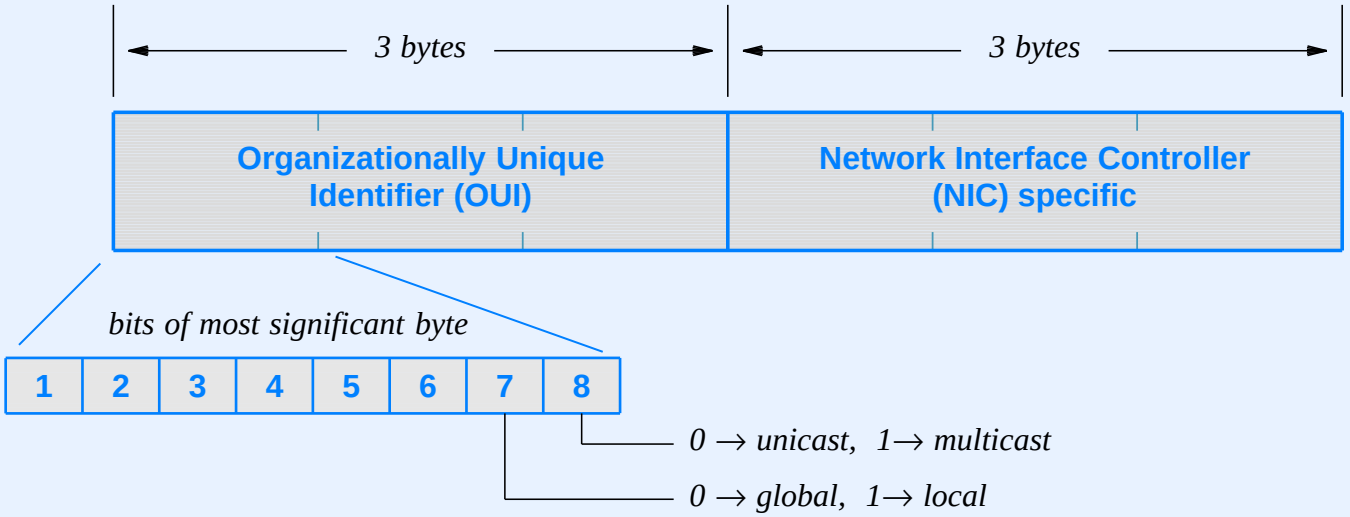
Endpoint Addressing Scheme

- Each station on a LAN is assigned a unique *address*
- Each packet specifies a destination address
- LAN hardware uses the address in a packet to determine which station(s) receive a copy

IEEE Standard For Addressing

- Formal name: *IEEE Media Access Control address* (*MAC address*)
- Informally called an *Ethernet address*
- Each address is 48 bits long
- Assigned to *Network Interface Card* (*NIC*) when device manufactured
- Divided into subfields
  - 3-byte *Organizationally Unique ID* (*OUI*)
  - 3-byte *Network Interface Controller* (*NIC*)

Illustration Of Fields In An IEEE 48-Bit Address



- Address types

Address Type	Meaning And Packet Delivery
unicast	Destination is a single computer; only that computer should receive a copy of the packet
broadcast	Destination is all computers on a network; they should each receive a copy of the packet
multicast	A subset of the computers on a network should receive a copy of the packet

Algorithm For Processing An Incoming Packet

Purpose:

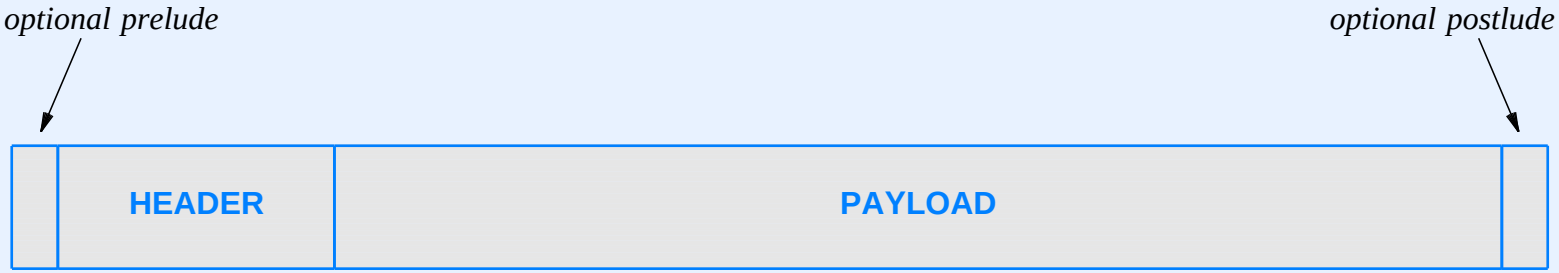
Handle a packet that has arrived over a LAN

Method:

```
Extract destination address, D, from the packet;
if (D matches “my address”) {
     accept and process the packet;
} else if (D matches the broadcast address) {
     accept and process the packet;
} else if (D matches one of the multicast addresses for a
   multicast group of which I am a member) {
     accept and process the packet;
} else {
     ignore the packet;
}
```

Frame Format

- Layer 2 packet is called a *frame*
- General layout of a frame



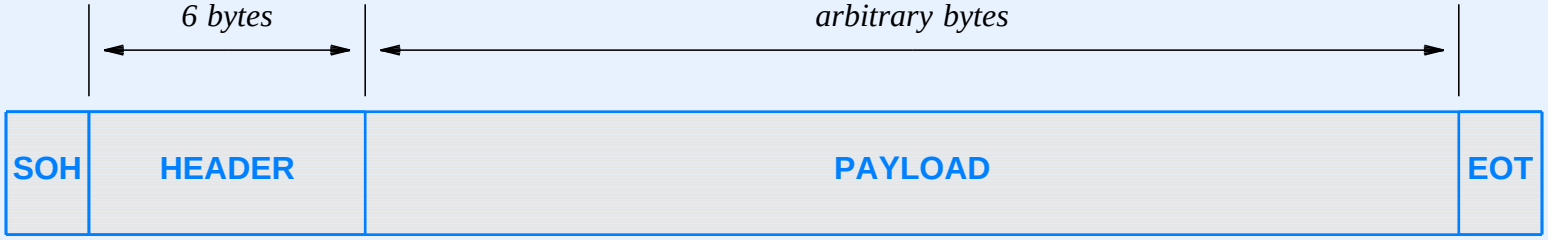
- Header usually has fixed fields
- Each technology imposes a maximum payload size
- Note: we will see specific frame formats later

Framing And Serial Communications Systems

- Consider sending packets over a leased circuit
- Circuit hardware either provides a stream of bits or a stream of bytes (characters)
- We will consider hardware that provides a byte stream
  - No frame boundaries
  - Any 8-bit value can appear in the data
- How can we send packets over such a system?
- Answer: sender and receiver must agree on framing

Example Framing Used With A Leased Circuit

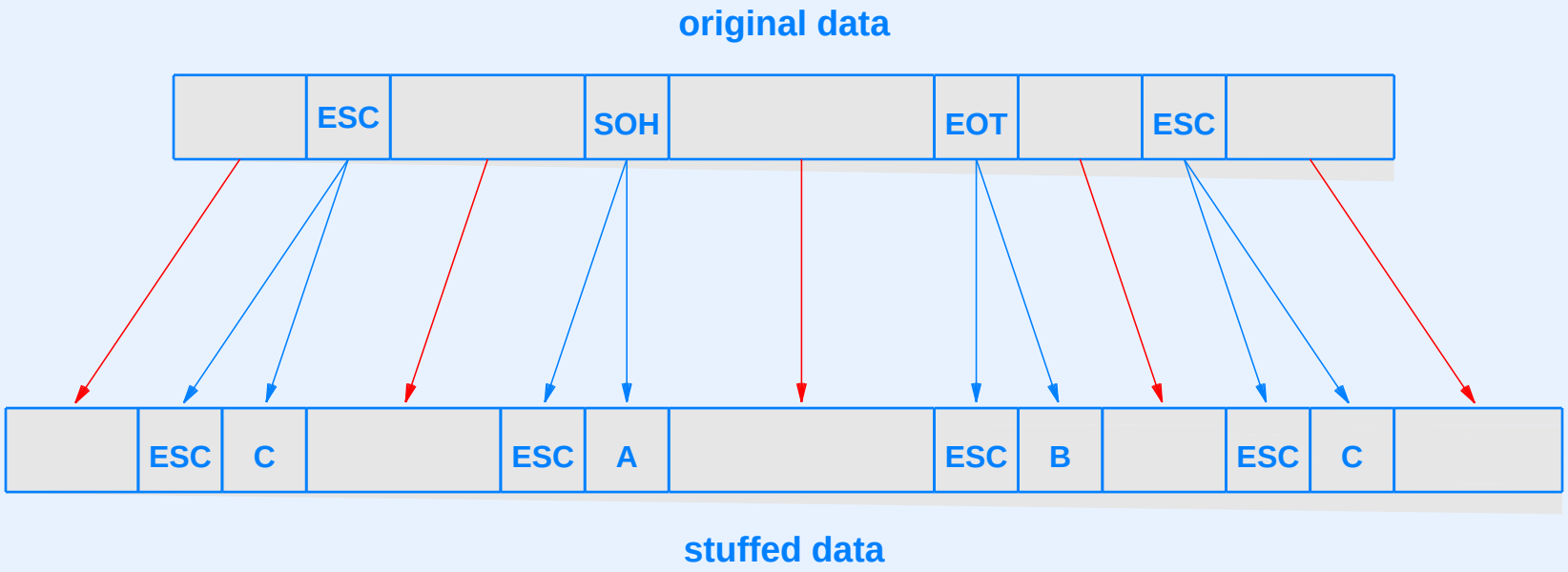
- Use SOH and EOT characters to mark the start and end of a frame



- Use byte stuffing within the payload

Byte In Payload	Sequence Sent
SOH	ESC A
EOT	ESC B
ESC	ESC C

Illustration Of Byte Stuffing



- Internet uses SLIP or PPP (standards) for transmission over serial circuits
- Bit stuffing techniques are also available for systems that transfer a stream of bits

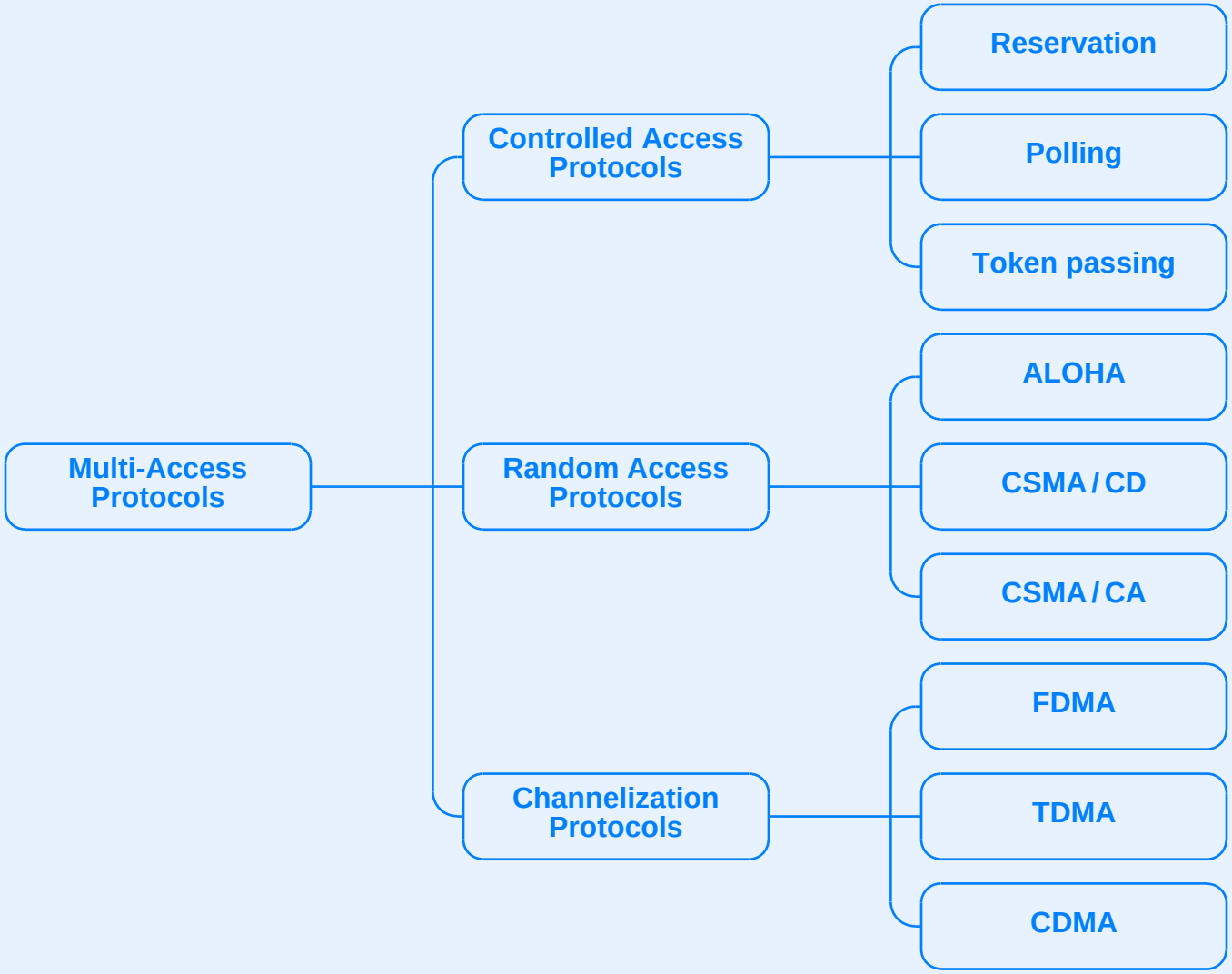
Media Access Mechanisms (IEEE MAC Sublayer)

MAC Protocols

- Control access to shared medium
- Two types of channel allocation
  - Static
  - Dynamic
- General principle:

Static channel allocation suffices when the set of communicating entities is known in advance and does not change; most networks require a form of dynamic channel allocation.

Taxonomy Of Media Access Mechanisms



Channelization Protocols

- Employ and extend basic multiplexing techniques
- May be static or dynamic
- Three basic types

Protocol	Expansion
FDMA	Frequency Division Multi-Access
TDMA	Time Division Multi-Access
CDMA	Code Division Multi-Access

Controlled Access Protocols

- Three principal forms

Type	Description
Polling	Centralized controller repeatedly polls stations and allows each to transmit one packet
Reservation	Stations submit a request for the next round of data transmission
Token Passing	Stations circulate a token; each time it receives the token, a station transmits one packet

- All three have been used in practice

Algorithm For Polled Access

Purpose:

Control transmission of packets through polling

Method:

```
Controller repeats forever {
    Select a station, S, and send a polling message to S;
    Wait for S to respond by sending a packet or passing;
}
```

Algorithm For Reservation-Based Access

- Often used with satellite systems
- Stations inform a controller if they have data to send

Purpose:

Control transmission of packets through reservation

Method:

```
Controller repeats forever {
    Form a list of stations that have a packet to send;
    Allow each station on the list to transmit;
}
```

Algorithm For Token Passing Access

- Special packet known as a *token* passed among senders
- Station sends one packet each time token arrives

Purpose:

Control transmission of packets through token passing

Method:

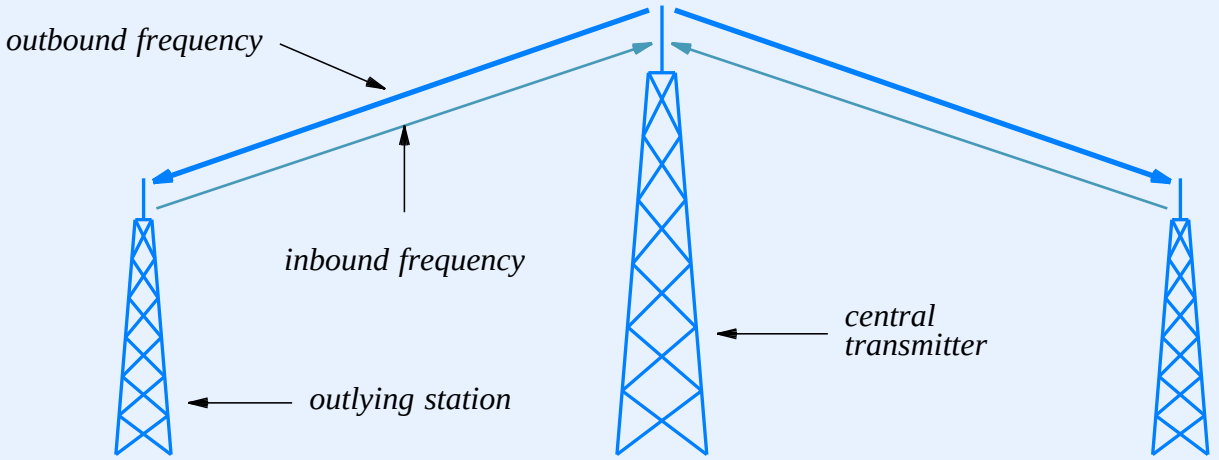
```
Each computer on the network repeats {
    Wait for the token to arrive;
    Transmit a packet if one is waiting to be sent;
    Send the token to the next station;
}
```

Example Random Access Protocols

Type	Description
ALOHA	Historic protocol used in an early radio network in Hawaii; popular in textbooks and easy to analyze, but not used in real networks
CSMA / CD	Carrier Sense Multi-Access with Collision Detection The basis for the original Ethernet, and the most widely used random access protocol
CSMA / CA	Carrier Sense Multi-Access with Collision Avoidance The basis for Wi-Fi wireless networks

Aloha

- Used in early network in Hawaii (ALOHAnet)
- Two carrier frequencies, *inbound* and *outbound*
- Central transmitter rebroadcast each incoming packet



- If inbound packets *collide*, each sender waits a random time and retransmits
- Channel utilization under 20%

CSMA/CD

- Used in original Ethernet (1973)
- Provides access to shared medium
- Principle features
  - Carrier Sense (CS)
  - Multiple Access (MA)
  - Collision Detection (CD)
- Uses binary exponential backoff

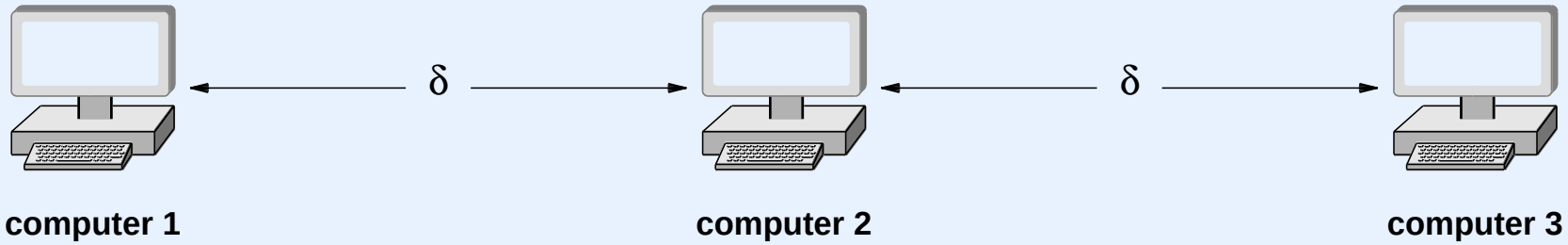
CSMA / CD Algorithm

Method:

When a packet is ready, perform CS (wait for access);
Delay for the interpacket gap;
Set variable x to the standard backoff range, d ;
Attempt to transmit the packet and perform CD;
While (collision occurred during transmission) {
    Choose q to be a random delay between 0 and x ;
    Delay for q microseconds;
    Double x in case needed for the next round;
    Attempt to retransmit the packet and perform CD;
}

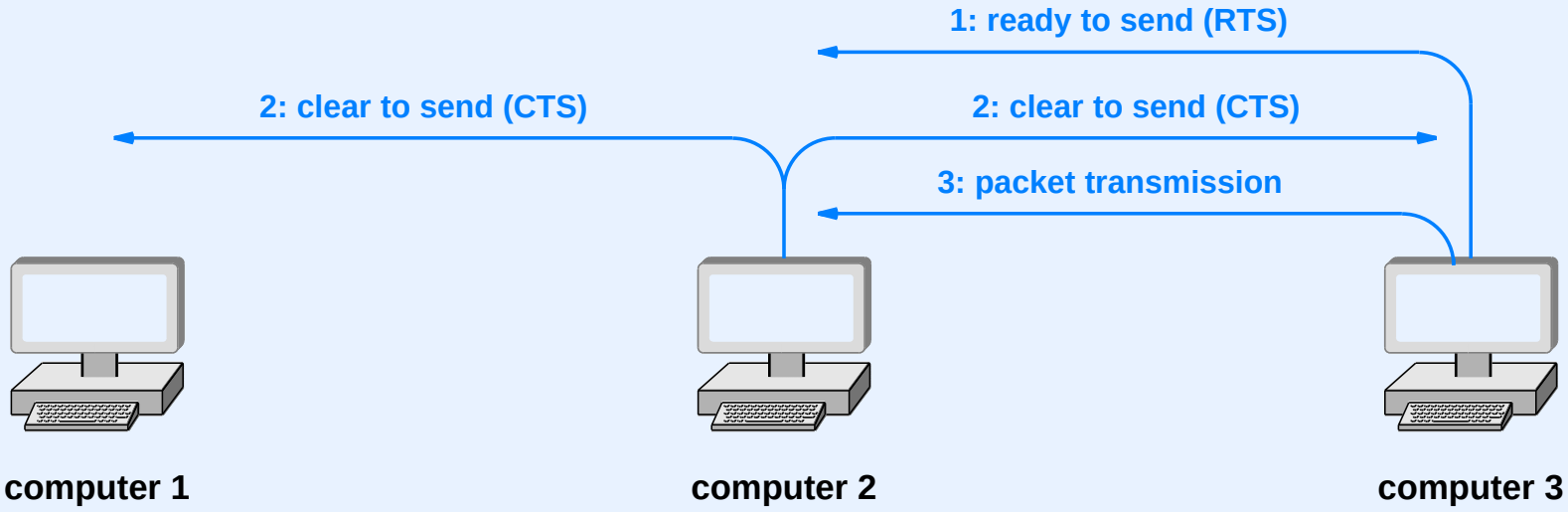
CSMA / CA

- Alternative to CSMA / CD
- Used in wireless networks (Wi-Fi)
- Needed because signals have limited distance, δ
- Example: computer 1 cannot receive transmission when computers 2 and 3 communicate



- All computers in range of computers 2 and 3 must be informed that a transmission will occur

Illustration Of CSMA / CA



- Communicating pair exchange *RTS* and *CTS* before packet transmission
- Any computer less than δ away from either computer 2 or 3 hears at least one of the RTS / CTS messages

Wired LAN technologies (Ethernet and 802.3)

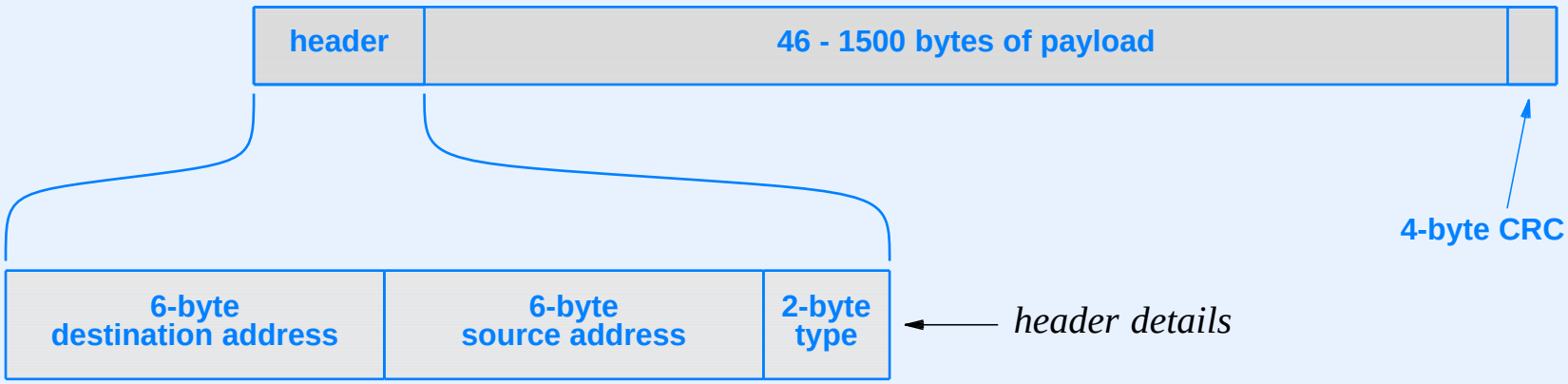
Wired LAN Technologies

- Explosion of technologies and products during 1980s
- Consolidation during the 1990s
- Currently: one de facto wired LAN standard

Ethernet

Ethernet Technology

- Invented at Xerox PARC in 1973
- Standardized by Digital, Intel, and Xerox (DIX) in 1978
- Frame has a 14-byte header followed by payload of 46 to 1500 bytes
- Frame format and addressing have survived virtually unchanged



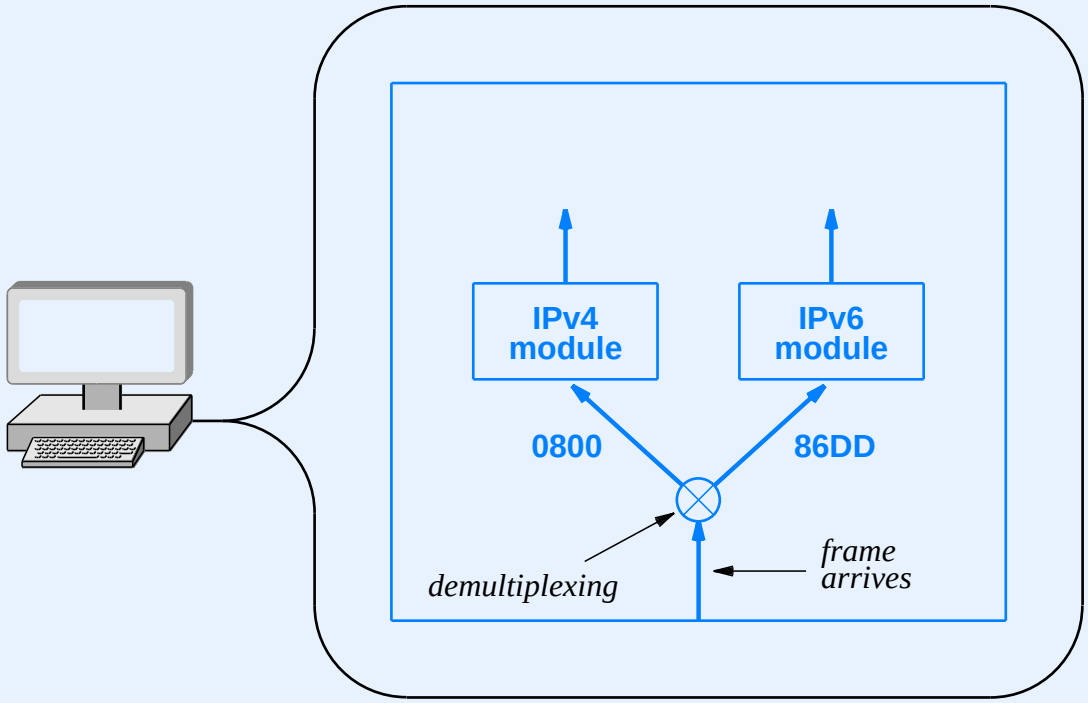
Ethernet Address Filtering

- Recall: station accepts a copy of the frame if *destination address* matches
  - The station's unicast address
  - The broadcast address (all 1s)
  - A multicast address to which station is listening
- Other frames are ignored
- *Promiscuous mode* allows a station to receive all frames regardless of address
  - Basis of protocol analyzer software such as *Wireshark*

Frame Type Field

- 2-octet field in frame header
- Set by sender to identify contents of frame
- Used by receiver to determine how to process the frame
- Values are standardized
- Examples:
  - Type 0x0800 used for IPv4 datagram
  - Type 0x86DD used for IPv6 datagram
  - Type 0x0806 used for ARP

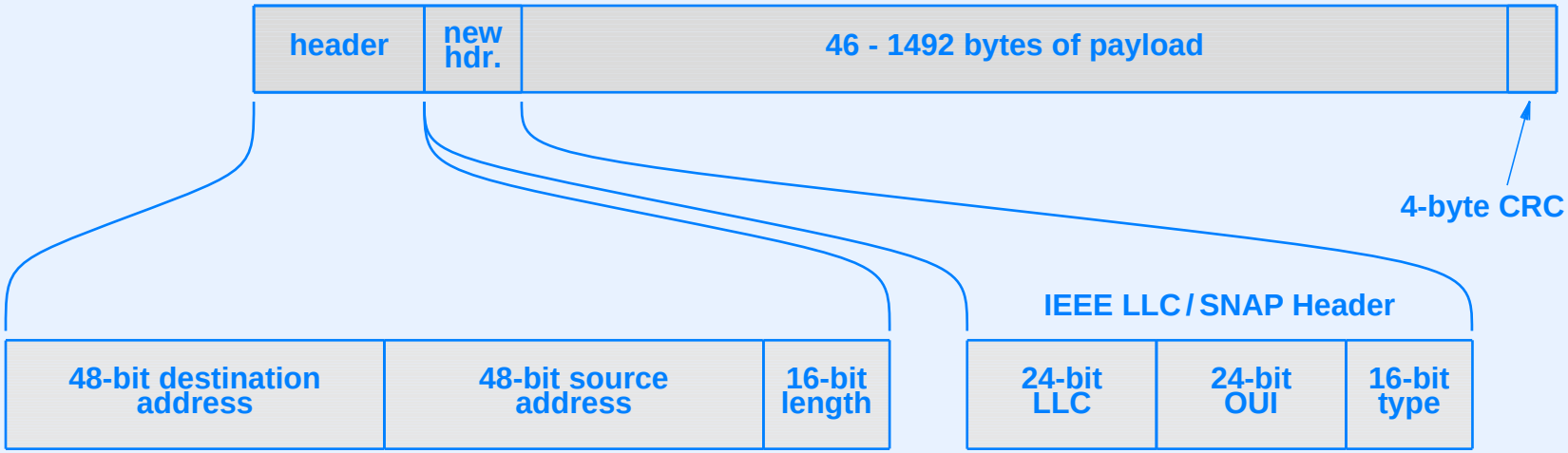
Illustration Of Frame Demultiplexing



- Performed when frame arrives
- Usually handled by protocol software
- Frame type field examined and frame passed to appropriate protocol module; unrecognized types are discarded

IEEE's Version Of Ethernet

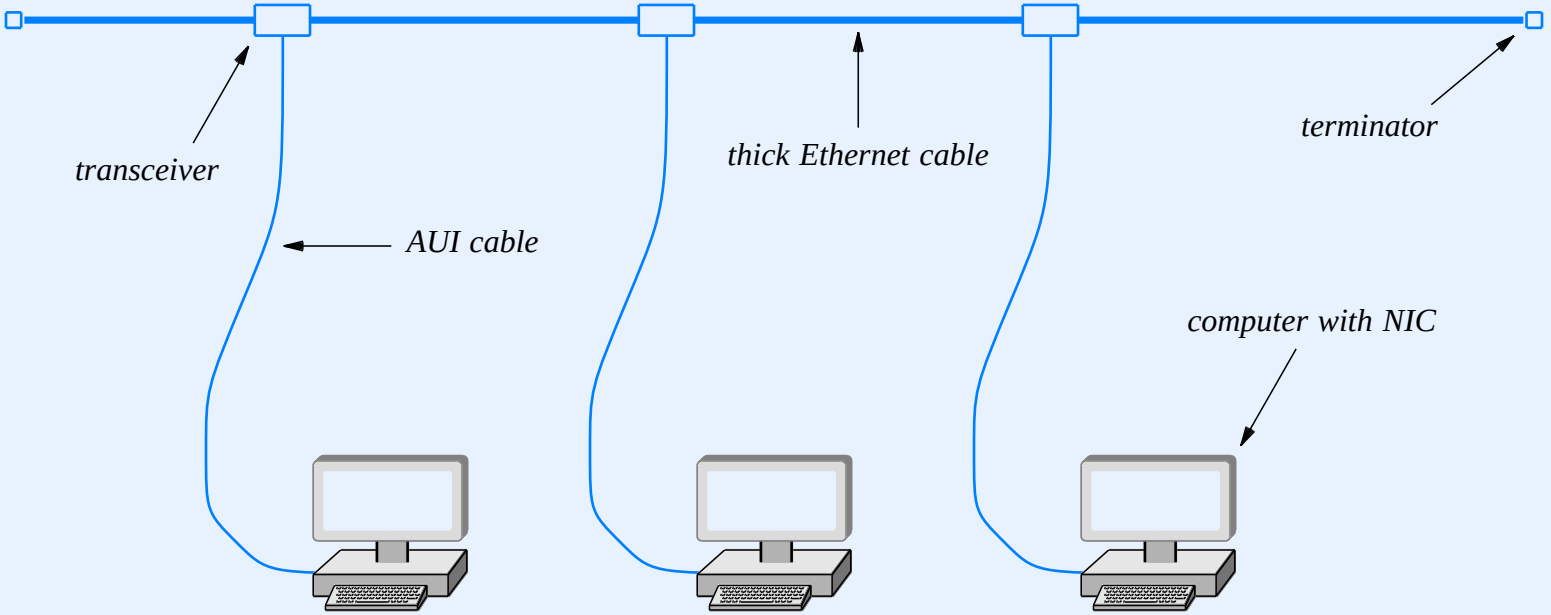
- Standardized in 1983 as IEEE standard 802.3
- Not widely adopted
- Header *type* field reinterpreted as a *frame length*
- Eight bytes of payload occupied by *LLC / SNAP header*



Ethernet Wiring

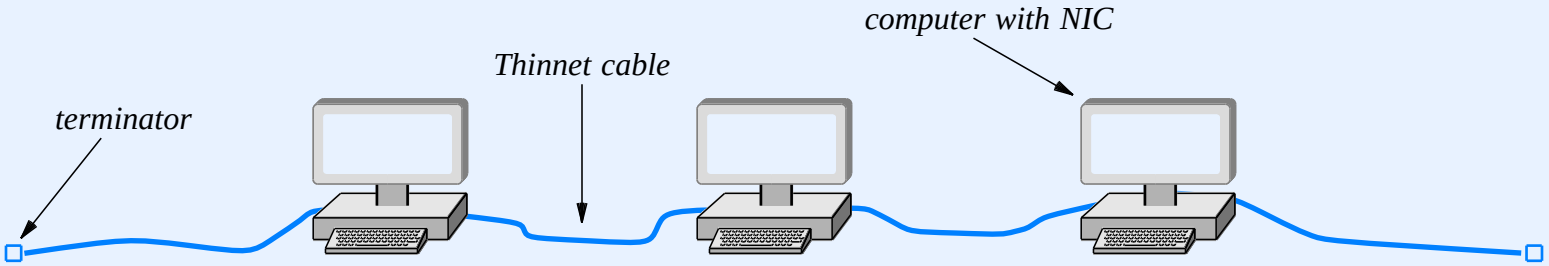
- Evolved through three generations
  - *Thicknet*
  - *Thinnet*
  - *Twisted pair*
- Illustrate a range of possible network wiring schemes

Illustration Of Thicknet Wiring



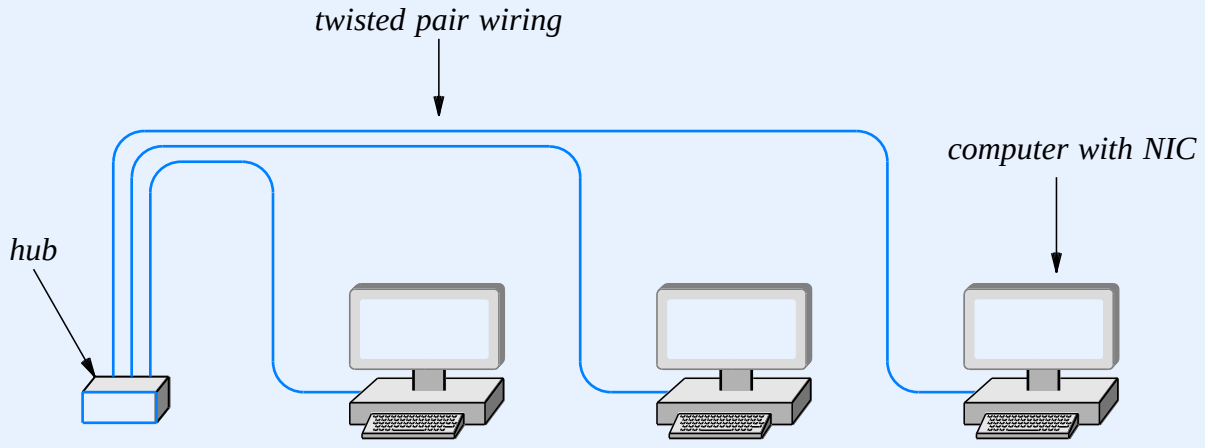
- Heavy coaxial cable typically in the ceiling
- Each computer attached to the cable

Illustration Of Thinnet Wiring



- Flexible coaxial cable
- Connections run point-to-point among computers
- Disadvantage: user can disconnect the network

Illustration Of Twisted Pair Ethernet Wiring



- Unshielded or shielded twisted pairs using RJ45 connectors
- Multiple pairs allows full-duplex operation
- Each computer connects to central *hub*
- Topology is physical star, but logical bus
- Hub is known as “bus in a box”

Evolution Of Twisted Pair Ethernet Technologies

- Several variants of twisted pair Ethernet have been created
- Variants differ in data rate and wiring required

Designation	Name	Data Rate	Cable Used
10BaseT	Twisted Pair Ethernet	10 Mbps	Category 5
100BaseT	Fast Ethernet	100 Mbps	Category 5E
1000BaseT	Gigabit Ethernet	1 Gbps	Category 6

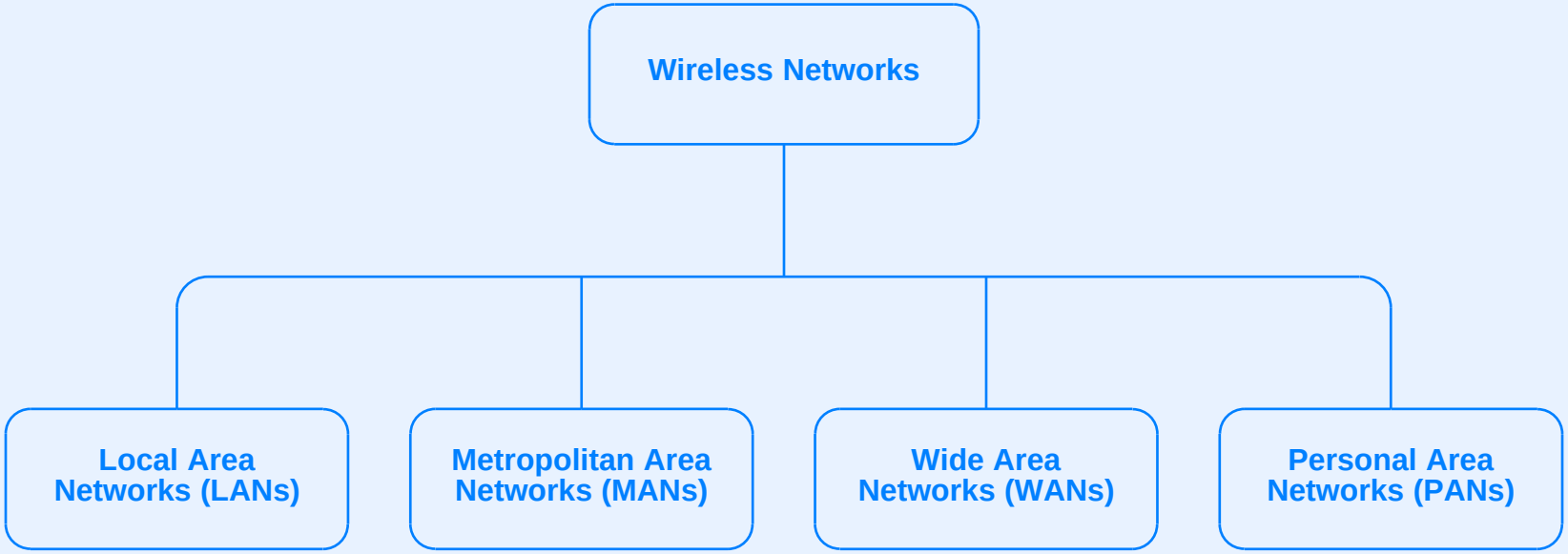
Wireless Networking Technologies

Wireless Networks

- Many types exist
- Technologies differ in
  - Distance spanned
  - Data rates
  - Physical characteristics of electromagnetic energy
    - Ability to permeate obstructions like walls
    - Susceptibility to interference
  - Isolated channel vs. shared channel

A Taxonomy Of Wireless Networks

- We use a basic taxonomy to help classify wireless technologies



- Note: the terminology is qualitative because some technologies span multiple categories

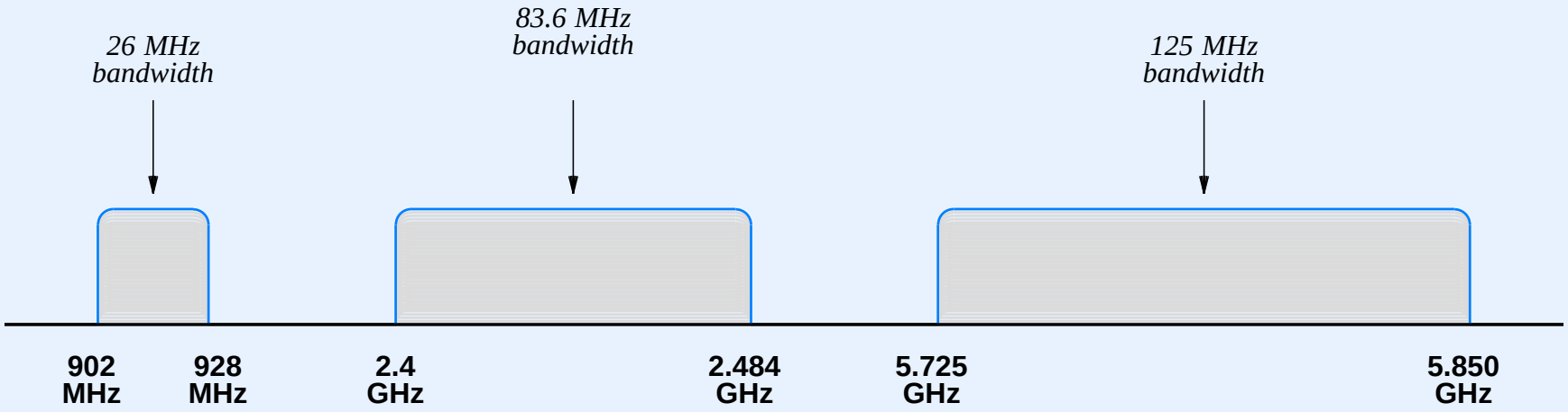
Personal Area Network (PAN)

- Terminology used primarily with wireless networks
- Spans short distance
- Dedicated to a single user (not shared)
- Example PAN technologies

Type	Purpose
Bluetooth	Communication over a short distance between a small peripheral device such as a headset or mouse and a system such as a cell phone or a computer
InfraRed	Line-of-sight communication between a small device, often a hand-held controller, and a nearby system such as a computer or entertainment center
ZigBee	Communication over distances about as large as a residence, which allows electrical appliances to connect to the Smart Grid

ISM Wireless Bands

- ISM stands for *Industrial, Scientific, and Medical*
- Region of the electromagnetic spectrum available for use without license
- Used for wireless LANs and PANs (e.g., cordless phones)
- Three separate bands



Unlicensed does not mean unregulated.

Wireless LANs And Wi-Fi

- Variety of wireless LANs have been created
- Vendors moved to open standards in 1990s, with IEEE providing most of the standards under 802.11
- In 1999, vendors formed *Wi-Fi Alliance*
- Example IEEE wireless standards

IEEE Standard	Frequency Band	Data Rate	Modulation Technique	Multiplexing Technique
original 802.11	2.4 GHz	1 or 2 Mbps	FSK	DSSS
	2.4 GHz	1 or 2 Mbps	FSK	FHSS
	InfraRed	1 or 2 Mbps	PPM	– none –
802.11b	2.4 GHz	5.5 and 11 Mbps	PSK	DSSS
802.11g	2.4 GHz	22 and 54 Mbps	various	OFDM
802.11n	2.4 GHz	54 to 600 Mbps	various	OFDM

Spread Spectrum Transmission

- Uses multiple frequencies for a single channel
- Can increase performance or provide immunity to noise
- Major spread spectrum techniques

Name	Expansion	Description
DSSS	Direct Sequence Spread Spectrum	Similar to CDMA where a sender multiplies the outgoing data by a sequence to form multiple frequencies and the receiver multiplies by the same sequence to decode
FHSS	Frequency Hopping Spread Spectrum	A sender uses a sequence of frequencies to transmit data, and a receiver uses the same sequence of frequencies to extract data
OFDM	Orthogonal Frequency Division Multiplexing	A frequency division multiplexing scheme where the transmission band is divided into many carriers in such a way that the carriers do not interfere

More IEEE Wireless LAN Standards

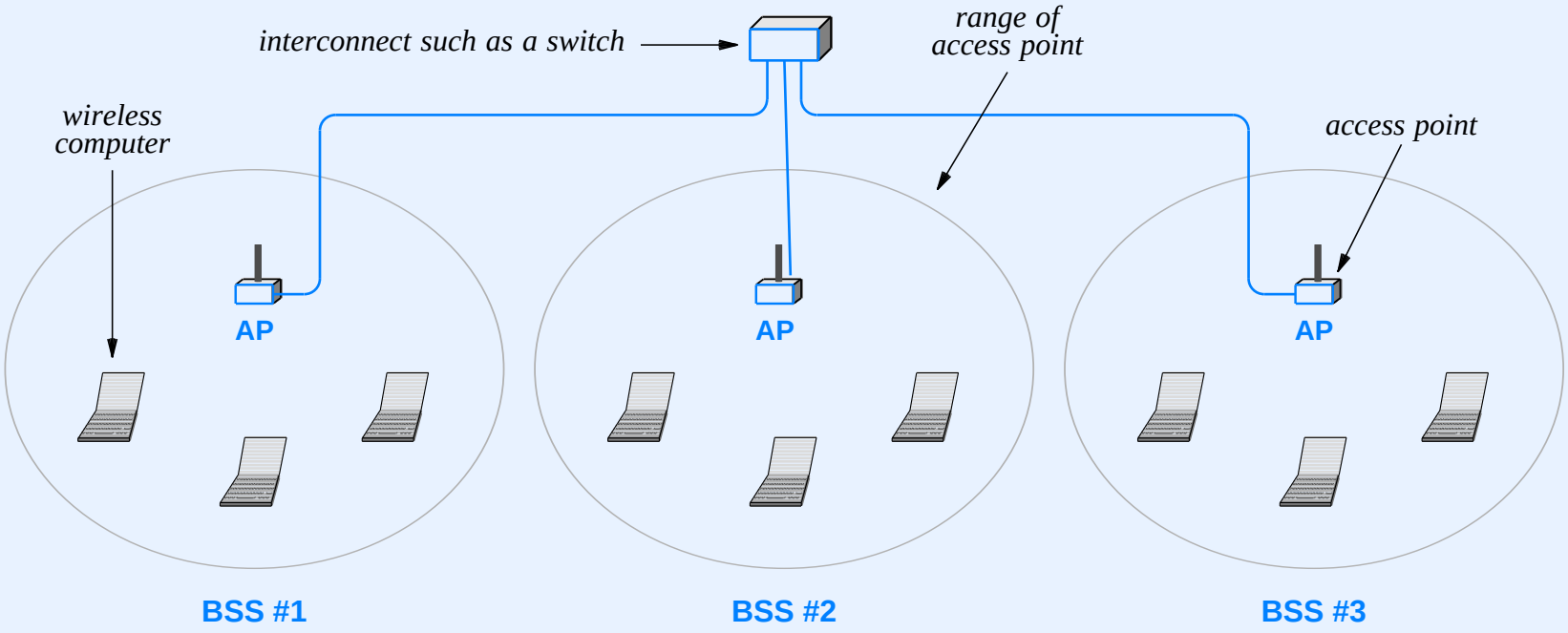
Standard	Purpose
802.11e	Improved quality of service, such as a guarantee of low jitter
802.11h	Like 802.11a, but adds control of spectrum and power (primarily intended for use in Europe)
802.11i	Enhanced security, including Advanced Encryption Standard; the full version is known as WPA2
802.11k	Will provide radio resource management, including transmission power
802.11n	Data rate over 100 Mbps to handle multimedia (video) applications (may be 500 Mbps)
802.11p	Dedicated Short-Range Communication (DSRC) among vehicles on a highway and vehicle-to-roadside
802.11r	Improved ability to roam among access points without losing connectivity
802.11s	Proposed for a mesh network in which a set of nodes automatically form a network and pass packets

Wireless LAN Architecture

- IEEE defines two possible modes for wireless LAN communication
- *Infrastructure* mode
  - Wireless devices communicate through an *access point* (*AP*)
  - APs connect to each other and (usually) the Internet
  - Typical uses: corporate wireless LAN, Internet cafe
- *Ad hoc* mode
  - Direct communication among wireless devices
  - Forwarding possible
  - Seldom used

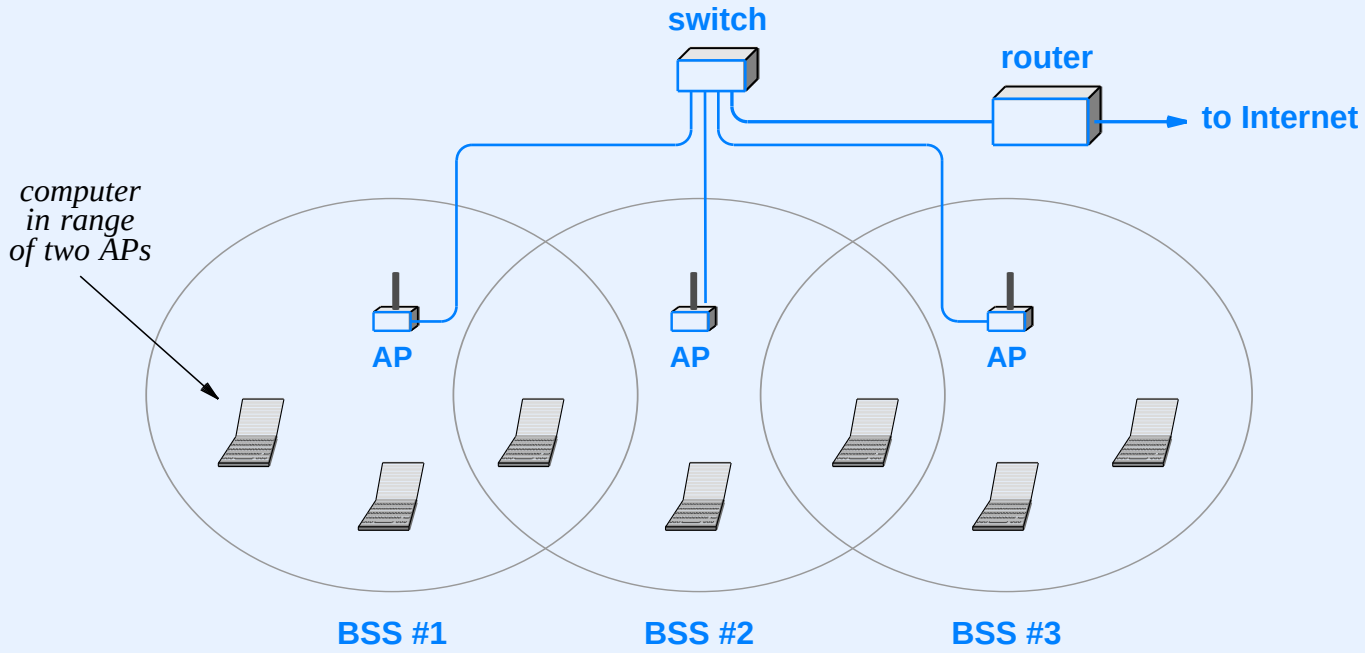
Illustration Of Infrastructure Mode Wireless LAN

- *Basic Service Set* (*BSS*) for an AP is defined as set of devices that can hear the AP
- APs interconnect through wired network



Practical Considerations And Association

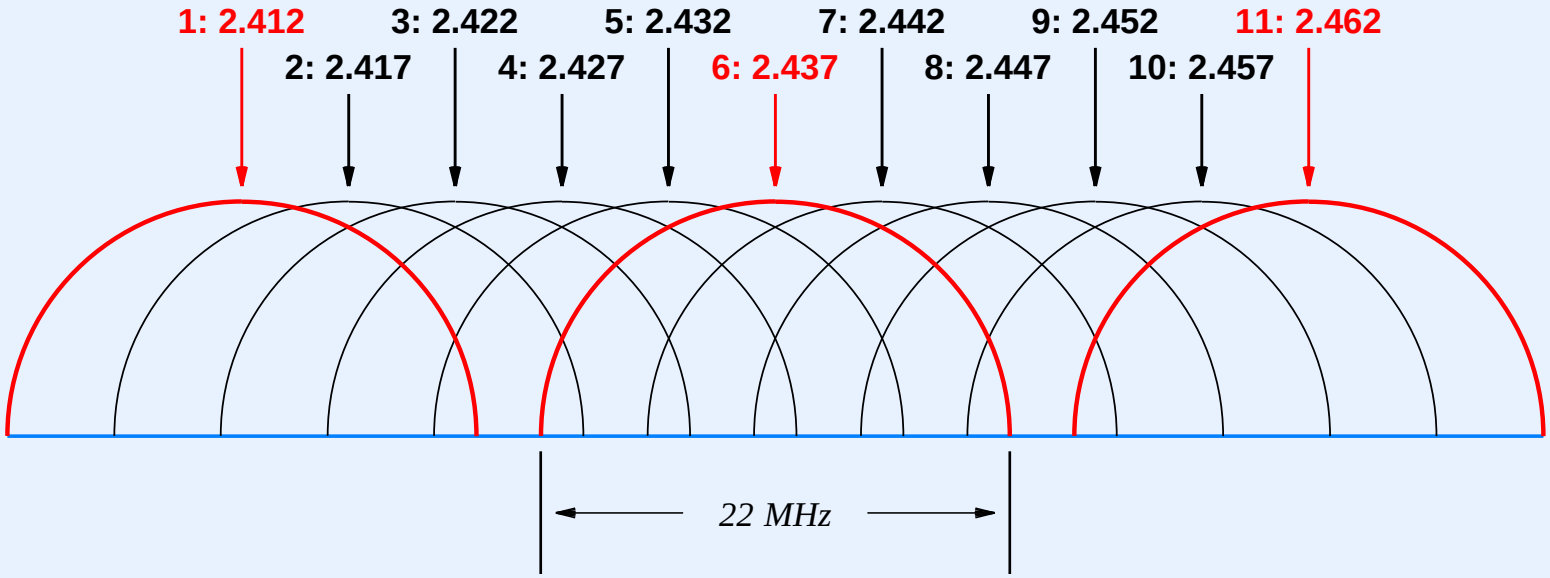
- In practice BSSs can overlap (given wireless device can hear more than one AP)



- To solve the problem each device *associates* with one AP at any time

Practical Considerations: Wi-Fi Channels

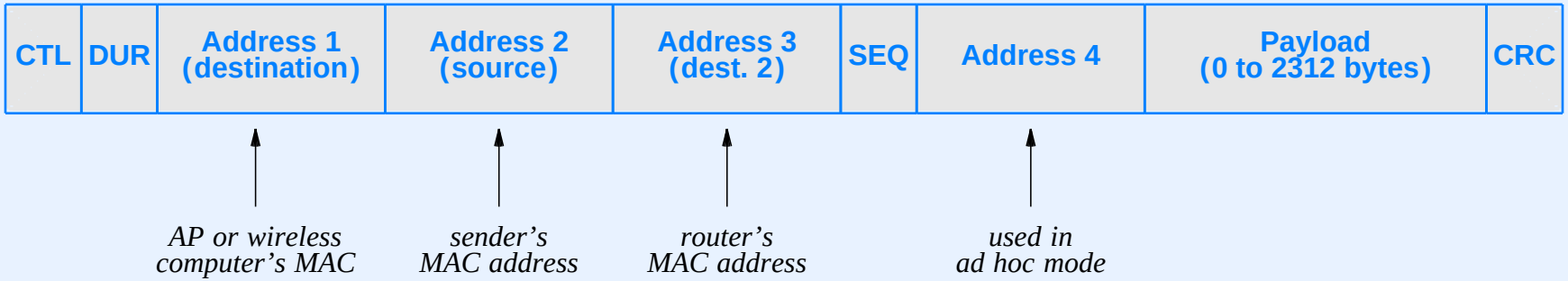
- 11 channels defined for North America in 2.4 GHz range
- Bad news: 22 MHz bandwidth means channels overlap



- Good news: channels 1, 6, and 11 can operate simultaneously with no interference

Addresses In 802.11 Frame Format

- 802.11 frame is *not* the same as an Ethernet frame
- Each 802.11 frame includes four MAC addresses
  - Source (e.g., wireless device)
  - Destination AP (associated AP)
  - Router along the path to the Internet
  - Extra address for ad hoc mode



Coordination Among Access Points

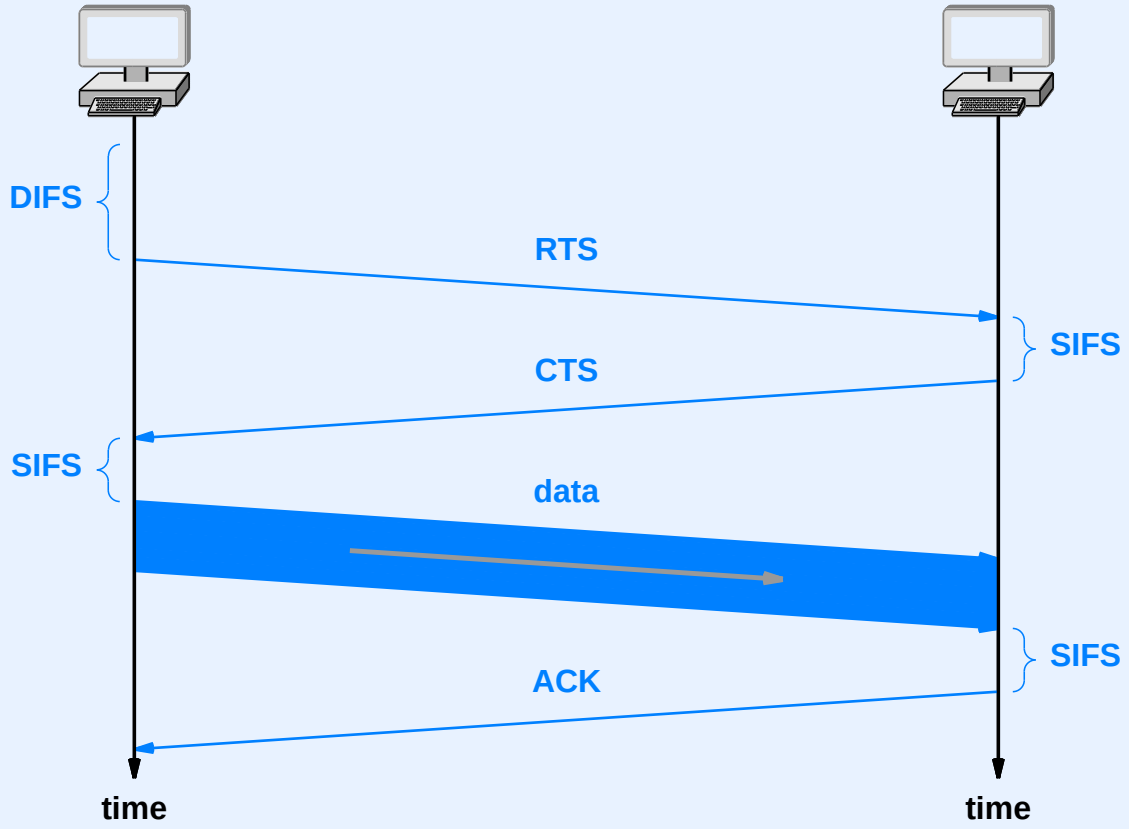
- Coordinated approach
  - Initial design
  - Similar to cellular telephone
  - APs communicate to achieve smooth handoff
- Uncoordinated approach
  - Later alternative
  - APs do not communicate
  - Wireless device changes association when communication with an AP lost
  - Lower overall cost

CSMA/CA Protocol (Review)

- Alternative to CSMA/CD used in wireless LANs
- Allows stations within range of communicating pair to know when communication starts
- Requires exchange of Ready-To-Send (RTS) and Clear-To-Send (CTS) messages
- Delay associated with each message to ensure protocol is efficient and correct

CSMA/CA Protocol Details

- SIFS — Short Inter-Frame Space of 10 μsec
- DIFS — Distributed Inter-Frame Space of 50 μsec
- Slot Time of 20 μsec



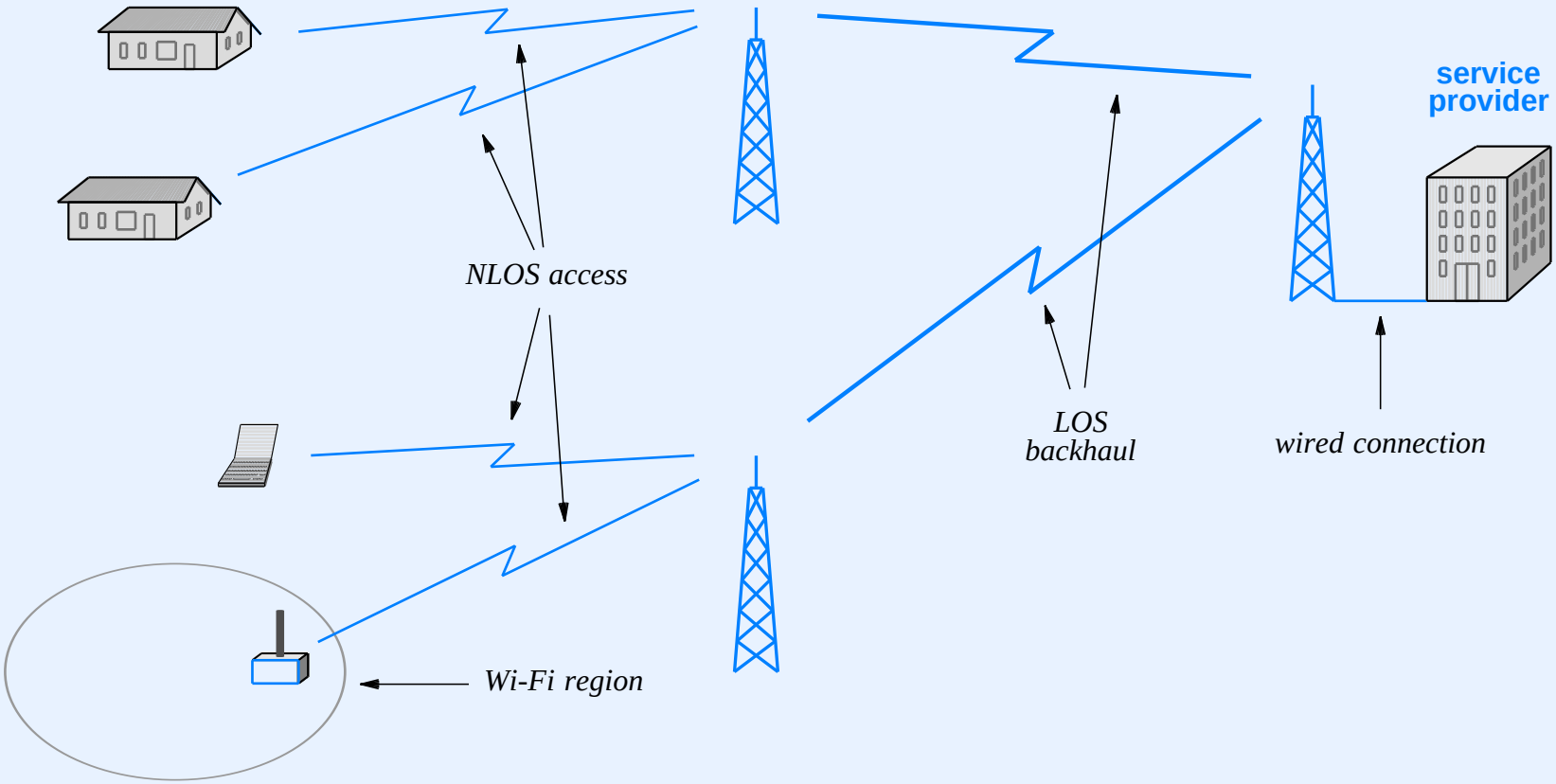
Wireless MAN Technology (WiMax)

- WiMax standard, IEEE 802.16, provides two types
  - Fixed (802.16-2004) — endpoint does not move
  - Mobile (802.16e-2005) — endpoint moves
- Uses

Access
– Last-mile alternative to DSL or cable modems – High-speed interconnection for nomadic users – Unified data and telecommunications access – As a backup for a site's Internet connection
Interconnect
– Backhaul from Wi-Fi access points to a provider – Private connections among sites of a company – Connection between small and large ISPs

Illustration Of WiMax Uses

- Fixed type of WiMax used for high-capacity backhaul requires Line-Of-Sight (LOS)



Standards For Wireless PANs

- Used in industrial as well as consumer products
- Remote control protocols optimized for short commands (do not need high data rate)

Standard	Purpose
802.15.1a	Bluetooth technology (1 Mbps; 2.4 GHz)
802.15.2	Coexistence among PANs (noninterference)
802.15.3	High rate PAN (55 Mbps; 2.4 GHz)
802.15.3a	Ultra Wideband (UWB) high rate PAN (110 Mbps; 2.4 GHz)
802.15.4	ZigBee technology – low data rate PAN for remote control
802.15.4a	Alternative low data rate PAN that uses low power

Other Short-Distance Wireless Technologies

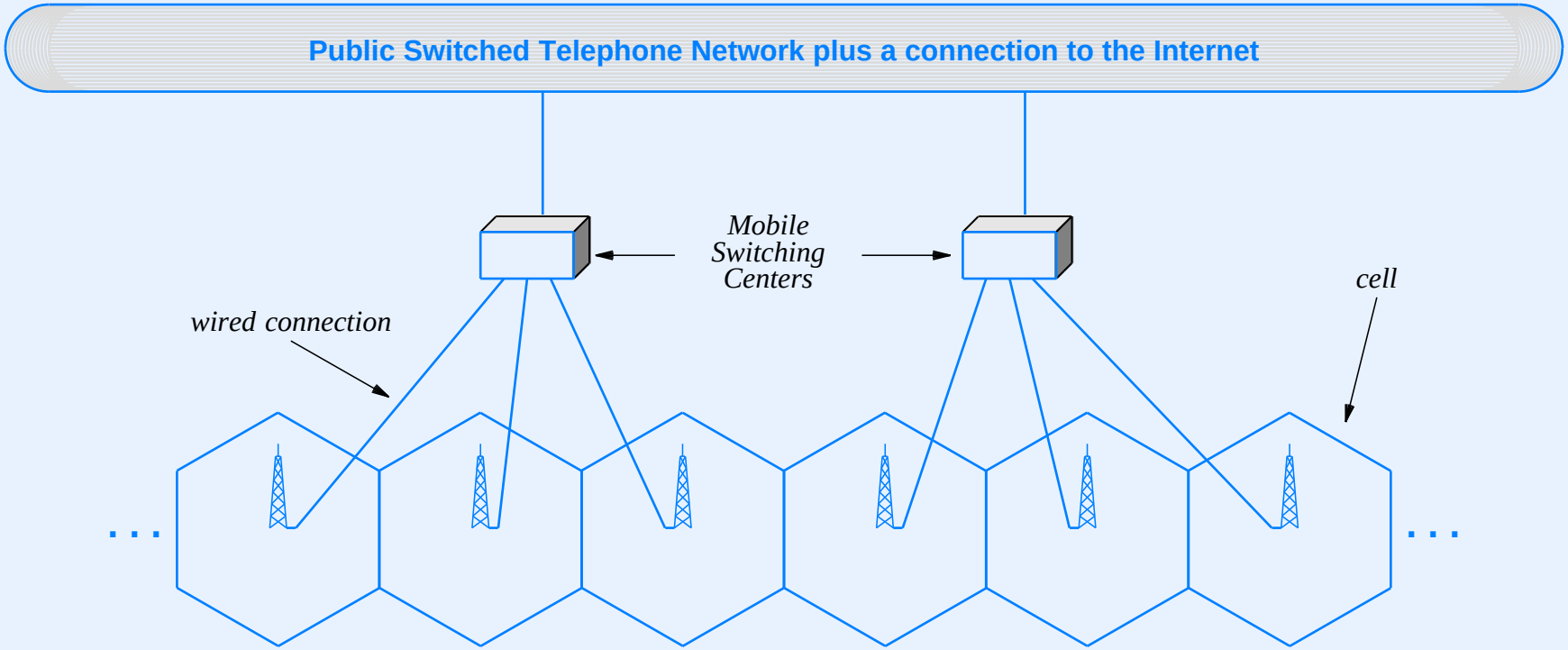
- Infrared Data Association (IrDA)
  - Family of standards (data rate of 2.4 Kbps to 16 Mbps)
  - Range of several meters
  - Directional transmission with cone covering 30 degrees
  - Generally low power consumption
- Radio Frequency IDentification (RFID) tags
  - Over 140 RFID standards exist
  - Passive RFID tags draw power from reader's signal
  - Active RFID tags contain a multi-year battery
  - Frequencies from less than 100 MHz to 868-954 MHz

Wireless WAN Technologies

- Cellular communication systems
- Satellite communication systems

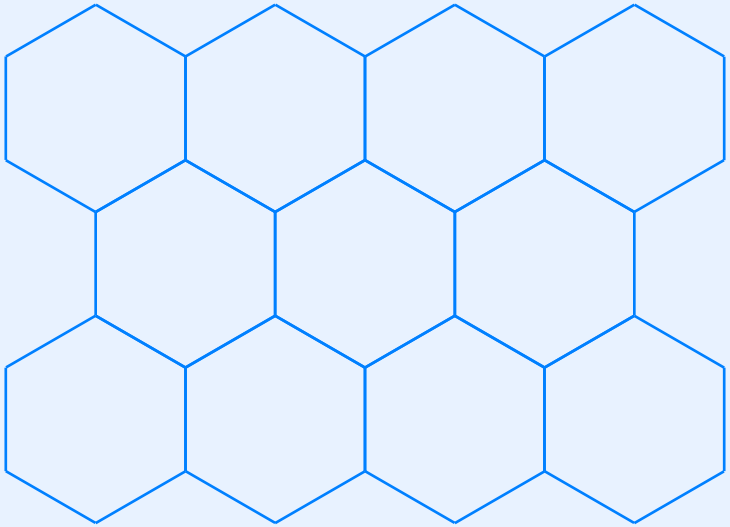
Current Cellular System Architecture

- Cell has a tower that connects to *mobile switching system*
- Each mobile switching system connects to PSTN or Internet

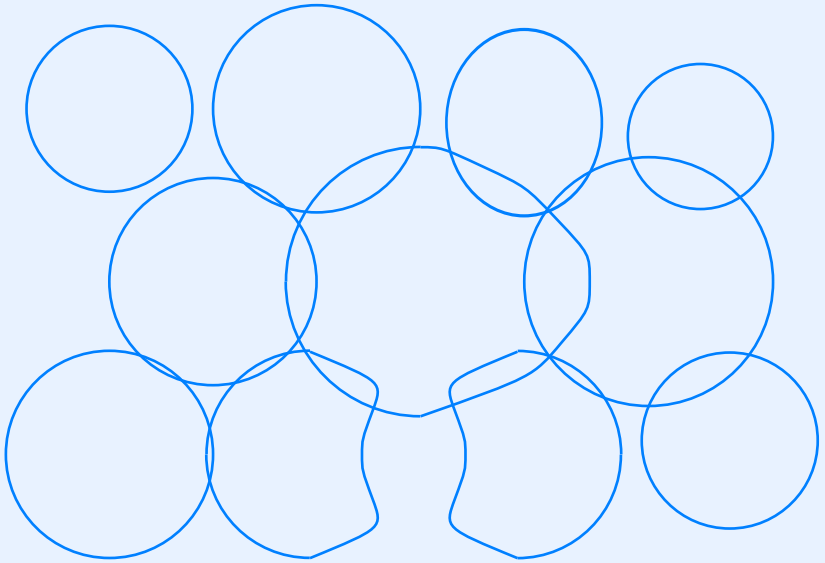


- Handoff decision made by infrastructure

Theoretical And Actual Cells



theoretical



actual

- Problems include: overlap and gaps

Cell Size And Expected Cell Phone Density

- Textbook diagrams show equal-size cells
- In practice, cell size related to expected number of cell phones
- Smaller cells used in high-population areas
- Larger cells used in rural areas

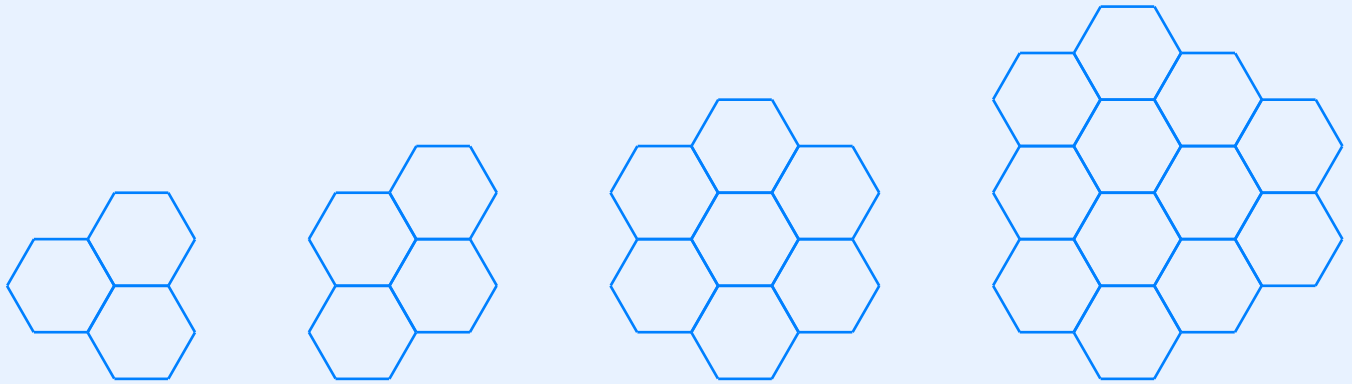
Frequency Assignment

- Goal: minimize interference
- Principle

Interference can be minimized if an adjacent pair of cells do not use the same frequency.

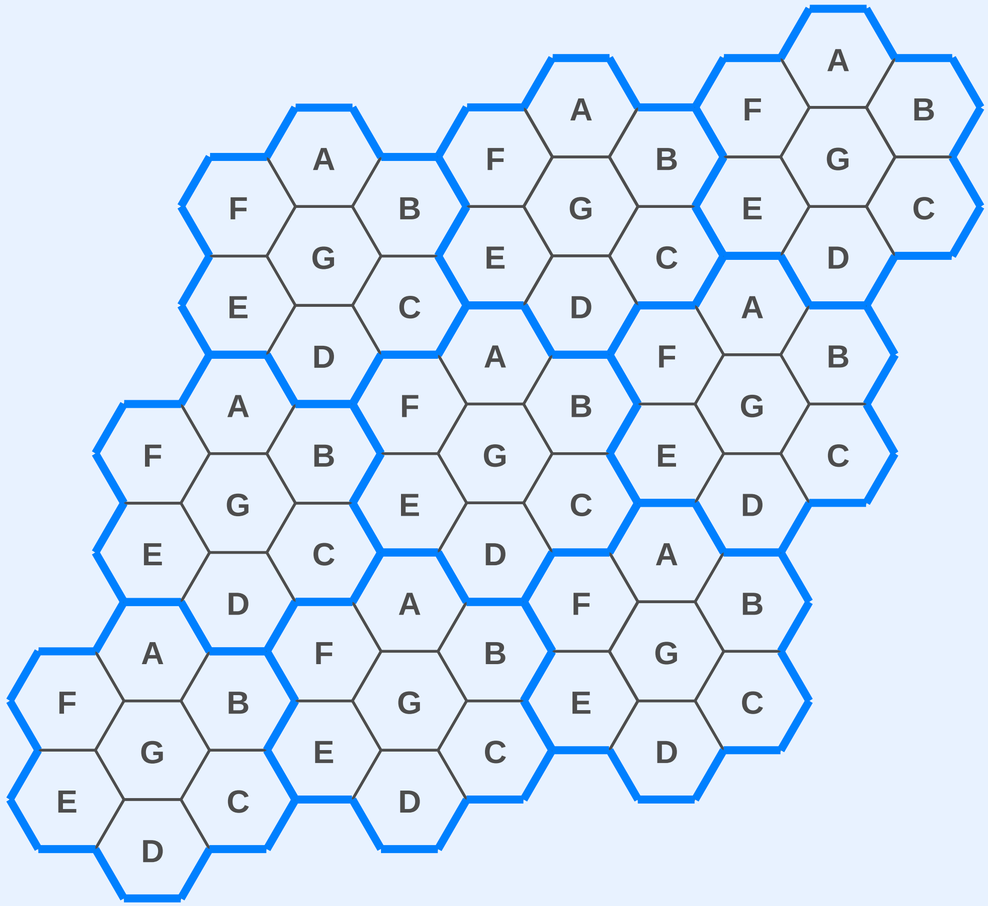
- Method: devise an assignment of frequencies such that two adjacent cells are not assigned the same frequency
- Technique: create a pattern that can be repeated
- Known as a *cluster* approach

Example Clusters That Are Used



- Each cell in cluster assigned a unique frequency
- When replicated, clusters cover 2-dimensional surface
- Mathematically, the concept is *tiling the plane*

Illustration Of Cluster Replication



- No pair of adjacent cells assigned the same frequency

Four Generations Of Cellular Networks

- 1G used analog (1970s - 1980s)
- 2G and 2.5G use digital signals for voice (1990s-)
- 3G and 3.5G also include data transfer at rates of 400 Kbps through 2 Mbps (2000s-)
- 4G offers higher data rates and support for real-time multimedia such as television (2008-)

Cellular Technologies

- Many competing standards
- European Conference Of Postal and Telecommunications Administrators chose a TDMA technology known as *Global System for Mobile Communications* (*GSM*) for Europe
- In US, each carrier created its own standards
  - Motorola created iDEN using TDMA
  - Others adopted IS-95A, which uses CDMA
- Japan chose PDC, which uses TDMA

Summary Of 2G Wireless Standards

Approach	Standard	Generation
GSM	GSM	2G
	GPRS	2.5G
	EDGE (EGPRS)	2.5G
	EDGE Evolution	2.5G
	HSCSD	2.5G
CDMA	IS-95A	2G
	IS-95B	2.5G
TDMA	iDEN	2G
	IS-136	2G
	PDC	2G

- Note: 2.5G standards extend 2G standards by adding some features of 3G

Third Generation Standards

- 2G standards were consolidated and extended:

Approach	Standard	Successor To
WCDMA	UMTS	IS-136, IS-95A, EDGE, PDC
	HSDPA	UMTS
CDMA 2000	1xRTT	IS-95B
	EVDO	1xRTT
	EVDV	1xRTT

- EVDO and EVDV data transfer standards evolved at approximately the same time to deliver data at 2.4 Mbps or 3.1 Mbps
- HSDPA can achieve 14 Mbps

Fourth Generation Standards

- Initially, the ITU insisted on high performance before using the term 4G
- Eventually, the ITU allowed intermediate technologies “to be advertised” as 4G

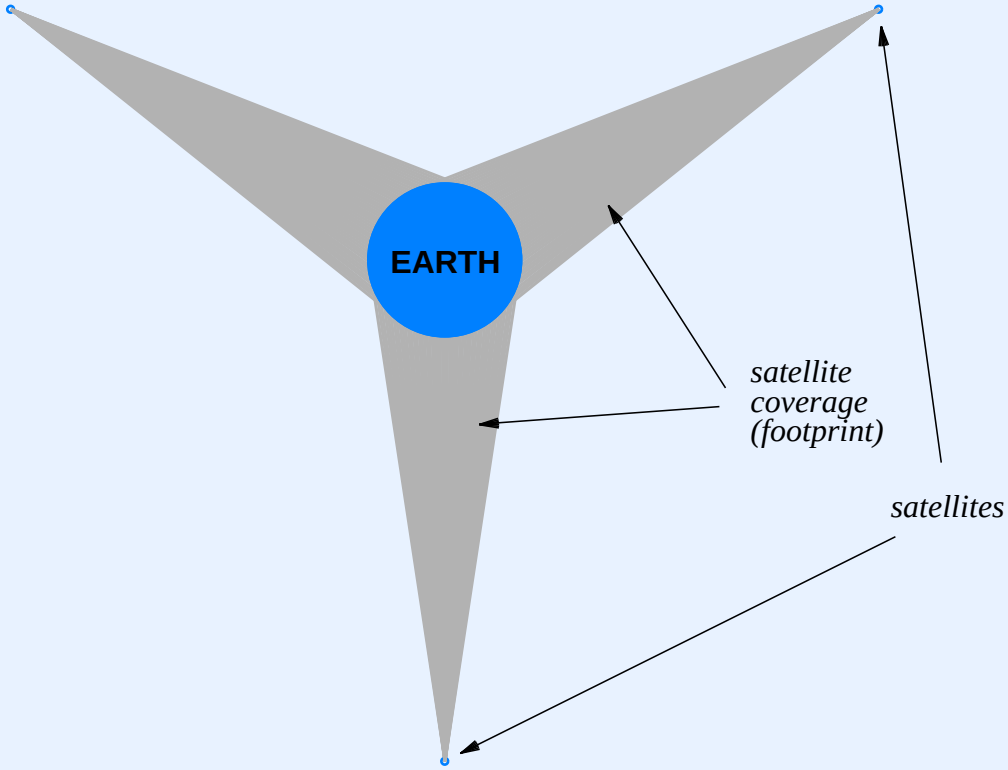
Classification	Standard
Can be advertised as 4G	HSPA+, HTC Evo 4G, LTE, WiMAX
Adheres to IMT-Advanced	LTE Advanced, WiMAX Advanced

Review Of Satellite Types

- Low Earth Orbit (LEO)
  - Appears to move across the sky
  - Requires a cluster of 66 satellites to cover the earth surface
- Medium Earth Orbit (MEO)
  - Covers the poles
  - Seldom used for general communication
- Geostationary Earth Orbit (GEO)
  - Appears to remain stationary in the sky
  - Requires only three satellites to cover the earth's surface

GEO Coverage Of The Earth's surface

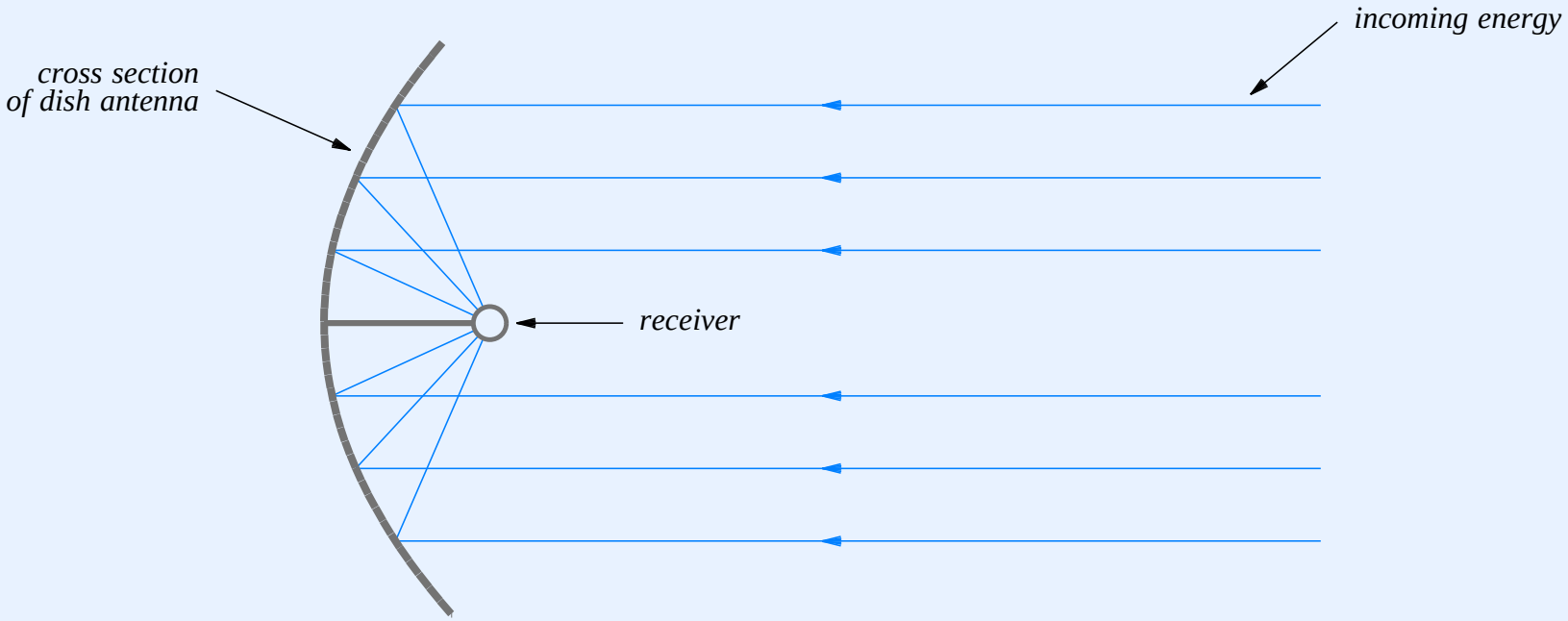
- In the best case, only three satellites needed



- Surface area covered known as *footprint*
- Ratio of distance to earth's diameter approximately to scale

VSAT Satellite Technology

- Stands for *Very Small Aperture Terminal*
- Parabolic antenna focuses incoming signal



- Example use: connect a company's retail stores

Frequency Bands Used With VSAT Technology

- Multiple bands available
- Each band has disadvantages

Band	Frequency	Footprint	Signal Strength	Effect Of Rain
C Band	3 - 7 GHz	Large	Low	Medium
Ku	10 - 18 GHz	Medium	Medium	Moderate
Ka	18 - 31 GHz	Small	High	Severe

Global Positioning System (GPS)

- 24 satellites
- Arranged in 6 orbital planes
- Civilian version has accuracy between 20 and 2 meters
- Relevance to data networking
  - Provides accurate time
  - Can be used to synchronize remote points in a data network (needed by some protocols)

Software Defined Radio

- Also known as a *software programmable radio*
- New approach emerging from research
- Exciting possibilities
- Replaces fixed radio components with mechanism that can be controlled by a programmable processor
- Can make better use of spectrum
- Potential downside: user might choose parameters that interfere with police or emergency vehicles

Features Controlled In A Software Radio

Feature	Description
Frequency	The exact set of frequencies used at a given time
Power	The amount of power the transmitter emits
Modulation	The signal and channel coding and modulation
Multiplexing	Any combination of CDMA, TDMA, FDMA and others
Signal Direction	Antennas can be tuned for a specific direction
MAC Protocol	All aspects of framing and MAC addressing

- Enabling technologies
  - Tunable analog filters to select frequencies and control power
  - Multiple antenna management to select direction

Multiple Antenna Management

- Needed because
  - No single antenna handles all frequencies
  - Directional signals important in focusing communication
- *Multiple-Input Multiple-Output (MIMO)* technology can aim transmission or reception

LAN Extensions

Network Design Tradeoffs

- Network technology engineered for
  - Distance spanned
  - Maximum data rate
  - Cost
- LAN technologies maximize data rate and minimize cost
- General principle

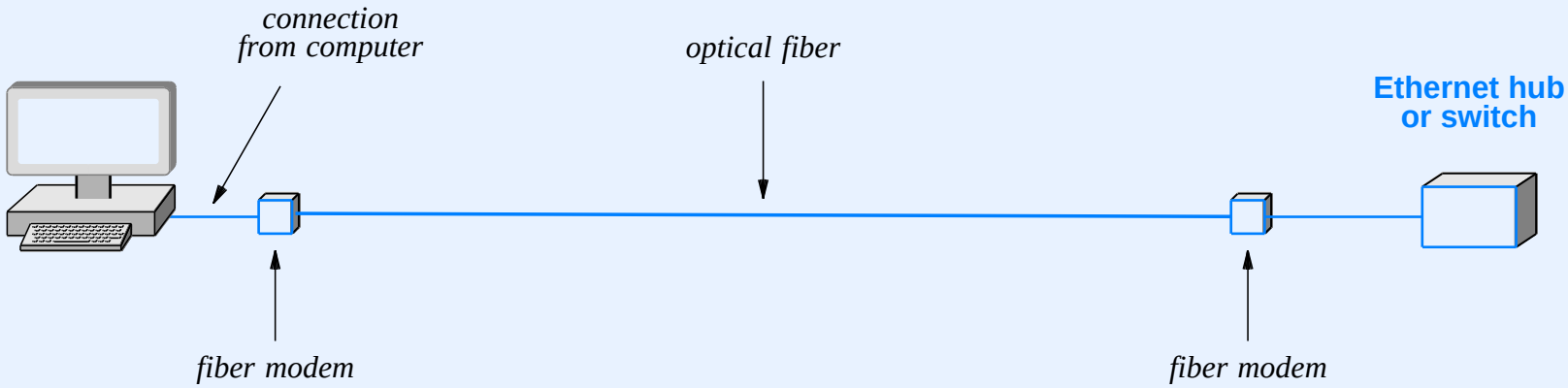
A maximum length specification is a fundamental part of LAN technology; LAN hardware will not work correctly over wires that exceed the bound.

Technologies That Extend LANs

- Variety of techniques have been invented to extend LANs
- Three key extension technologies
  - Fiber modems
  - Repeaters
  - Bridges

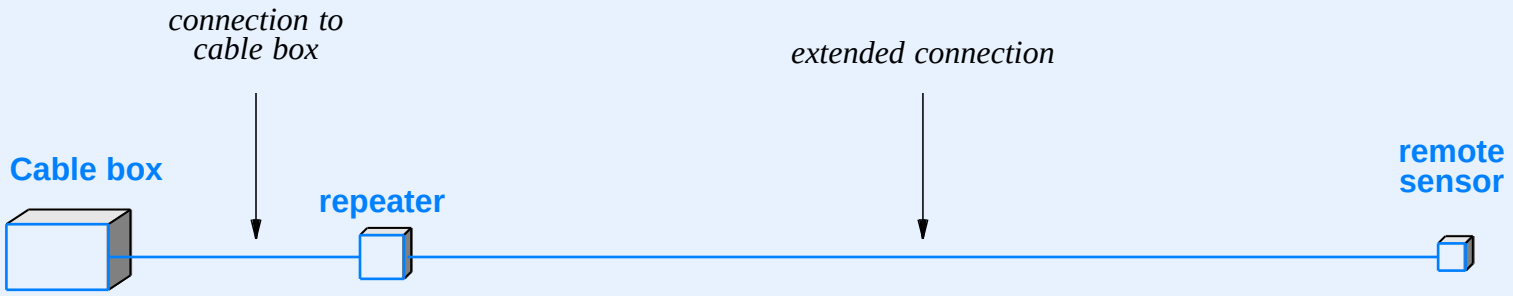
Fiber Modems

- Communicate over an optical fiber
- Can span long distance
- Provide standard network interface (e.g., Ethernet)
- Can be used to extend connection between computer and network
- Illustration of an extended network connection



Repeaters

- Operate at layer 1 (do not understand packets)
- Repeat and amplify signals
- Low cost
- Example use: extended infrared sensor on a cable box



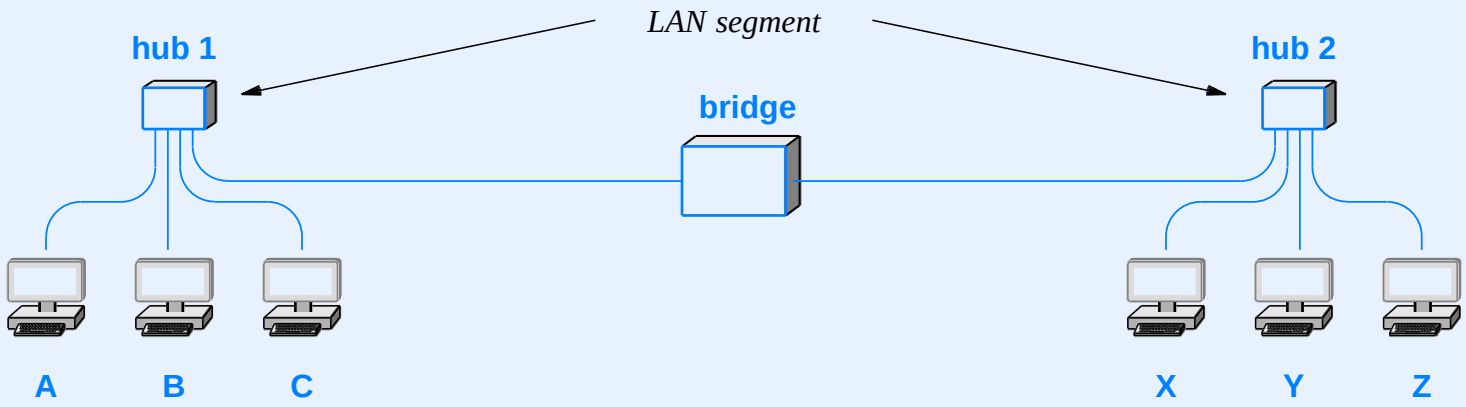
- Disadvantage: amplifies and repeats noise

Switches and Switched Networks

Bridge

- Originally sold as stand-alone device to extend two LAN segments
- Operates at layer 2
- Can connect two or more segments
- Listens in *promiscuous mode* on each segment and sends copy of each frame to other segments
- Does not copy noise, collisions, or frames that are incorrectly formed
- Makes connected segments appear to be a single, large LAN
- Uses *source MAC address* in frames to learn computer locations automatically, and uses *destination MAC address* to filter frames

Illustration Of A Bridge Learning



Event	Segment 1	Segment 2	Frame Sent
Bridge boots	–	–	–
A sends to B	A	–	Both Segments
B sends to A	A, B	–	Segment 1 only
X broadcasts	A, B	X	Both Segments
Y sends to A	A, B	X, Y	Both Segments
Y sends to X	A, B	X, Y	Segment 2 only
C sends to Z	A, B, C	X, Y	Both Segments
Z sends to X	A, B, C	X, Y, Z	Segment 2 only

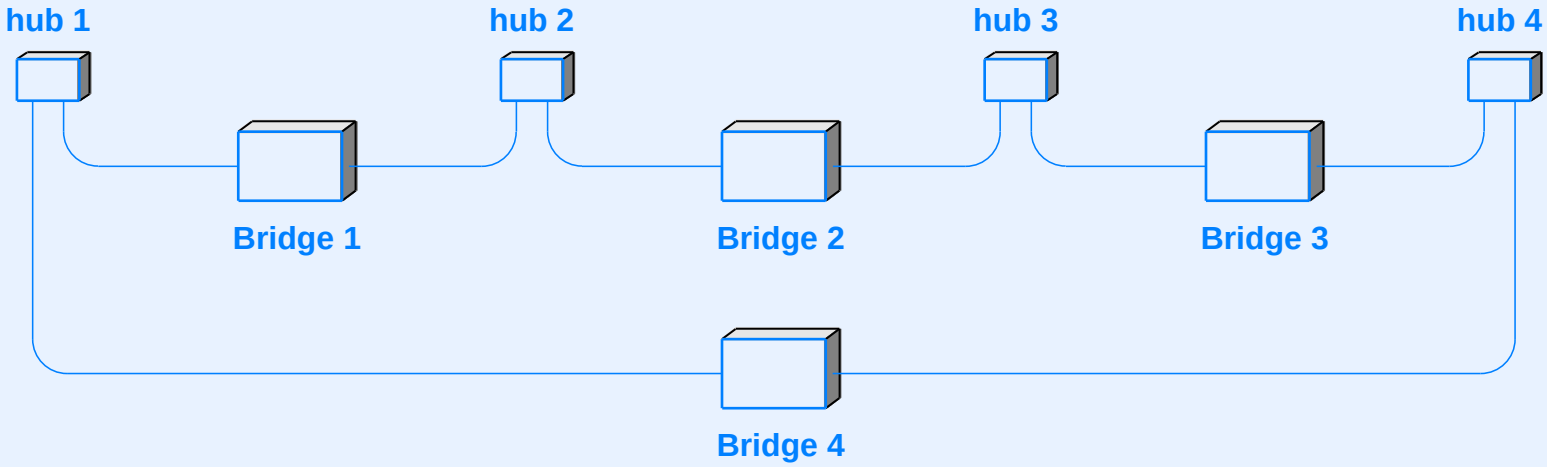
General Principle

Because a bridge permits simultaneous activity on attached segments, a pair of computers on one segment can communicate at the same time as a pair of computers on another segment.

- Each segment forms a separate *collision domain*

A Problem With Bridges

- A bridge *always* forwards broadcast and multicast frames
- Consider four bridges used to connect four LAN segments in a *loop*



- What happens if a computer attached to one of the segments sends a broadcast frame?

Copies of the frame cycle around the bridges forever!

Distributed Spanning Tree

- Prevents a packet from circulating around a cycle of bridges
- Initial protocol developed by Perlman at *Digital Equipment Corporation*
- Executed by each bridge when the bridge boots
- Allows bridges to break a forwarding cycle
- Name *Spanning Tree Protocol* (*STP*) applies to basic protocol
- Many variants have been created with extended names

How STP Works

- Executed at startup
- Distributed algorithm
  - Each bridge runs it independently
  - No central coordination
- Algorithm guaranteed to converge quickly
- No data packets forwarded until STP finishes

Steps Taken By STP

- Bridges exchange a series of STP messages (frames) that are used to
  - Elect a *root bridge*
  - Select a shortest path to the root
- Each bridge disables forwarding broadcast or multicast except along the selected path
- Result is a tree

Bridging Is Alive And Well

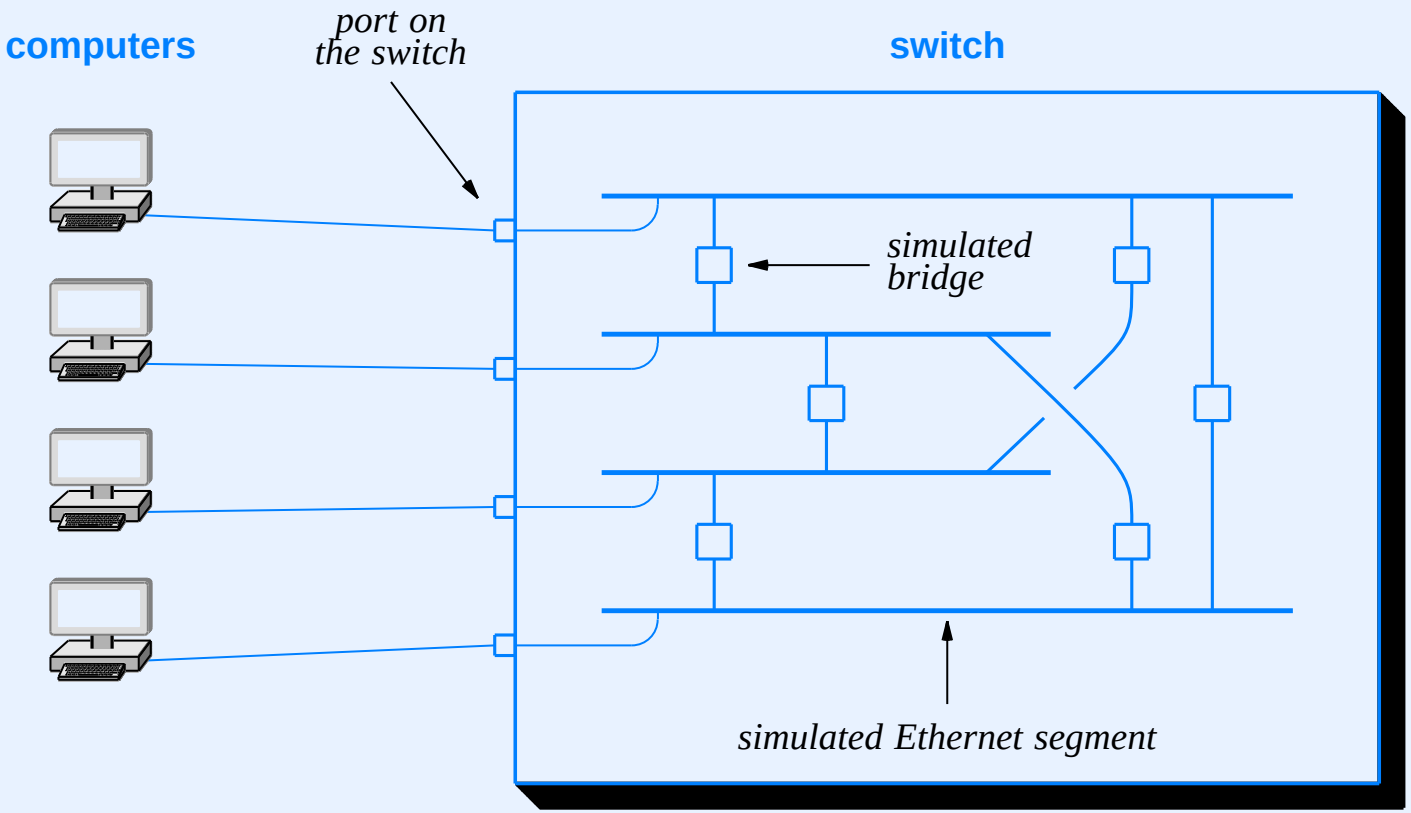
- Stand-alone bridge devices are seldom used
- Bridge technology is now incorporated into other devices
  - DSL modems
  - Cable modems
  - Wi-Fi “repeaters”
  - Satellite systems

Switching

Layer 2 Switch

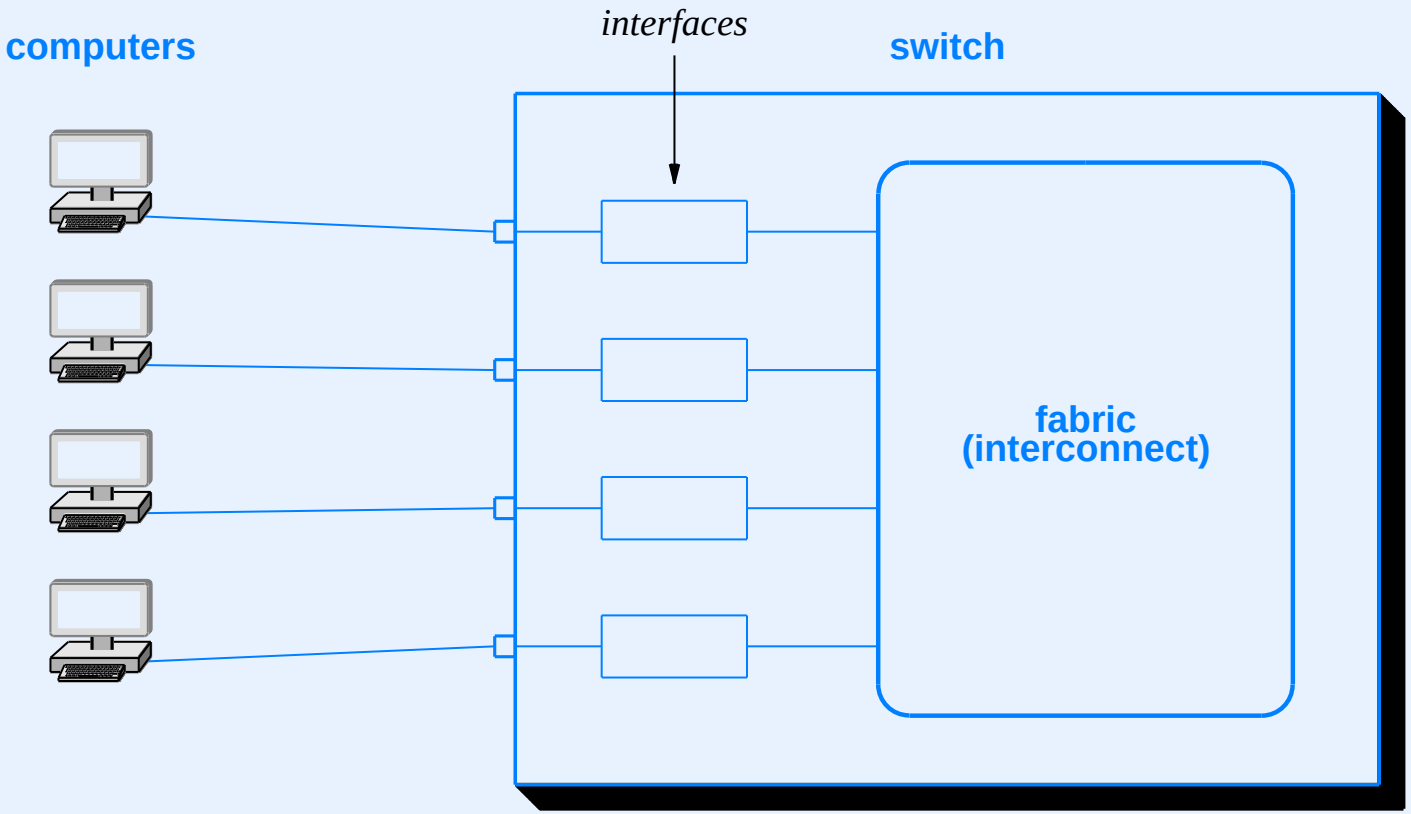
- Physically similar to a layer 2 hub
  - Network device
  - Connects multiple computers
  - Computers appear to be attached to a LAN segment
- Logically similar to a set of bridged networks
  - Switch understands packets, not just signals
  - No contention, and no need for CSMA / CD
  - Ports operate in parallel
  - Switch can include services that examine packets

Logical Function Of A Switch



- Switch offers same advantage as bridged networks: multiple transfers can occur simultaneously

Actual Switch Architecture



- Switching fabric used for high throughput

Thought Problem

Suppose a computer is unplugged from a port on a Layer 2 switch and plugged into another port. Suppose the computer does not send any packets. Will the computer continue to receive unicast frames that are sent to it? Why or why not?

Virtual Local Area Network (VLAN) Switch

- Physically
  - Similar to a conventional Layer 2 switch
  - Has ports to which computer can connect
- Logically
  - Manager can configure one or more *broadcast domains*
  - Each port assigned to one broadcast domain
- Frame sent to broadcast or multicast address only propagated to ports in the same broadcast domain

Networking Technologies: Past And Present

A Wide Variety of Networking Technologies

- LAN technologies
  - Token ring (esp., IBM Token Ring)
  - FDDI/CDDI
- WAN technologies
  - X.25
  - Frame Relay
  - ATM
  - ISDN
  - MPLS
- See Chapter 19 for a longer list

Asynchronous Transfer Mode (ATM)

- Created by phone companies in 1990s
- Intended as replacement for the Internet
- Paradigm was connection-oriented
- Used small cells (53 octets)
- Network guaranteed per-connection *Quality of Service* (*QoS*)
  - Throughput
  - Bound on delay
  - Bound on jitter

Asynchronous Transfer Mode (ATM) (continued)

- QoS in ATM
  - Specified for each transfer (i.e., each TCP connection)
  - Required setup time
  - Meant each switch maintained state
  - Was difficult/impossible to enforce at high speed
- Despite the failure of ATM, proponents still argue that Internet needs QoS

Summary

- Packet switching divides data into small packets
- Each packet (frame) specifies destination
- Access technologies are used in the last mile
- Media access can be controlled, random, or channelized
- IEEE specifies Local Area Network standards
- Topologies used with LANs: bus, star, ring, and mesh
- Ethernet is the de facto standard for wired LANs
- Current Ethernets use twisted pair wiring

Summary

- Wireless networks include PANs, LANs, and WANs,
- Cellular telephones are using packet technology
- Satellite can deliver data through a dish antenna
- Software-defined radio adds flexibility to wireless devices
- LAN extensions include repeaters and bridges
- Once stand-alone devices, bridges are now incorporated into other devices
- Layer 2 switch acts like bridged networks

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)

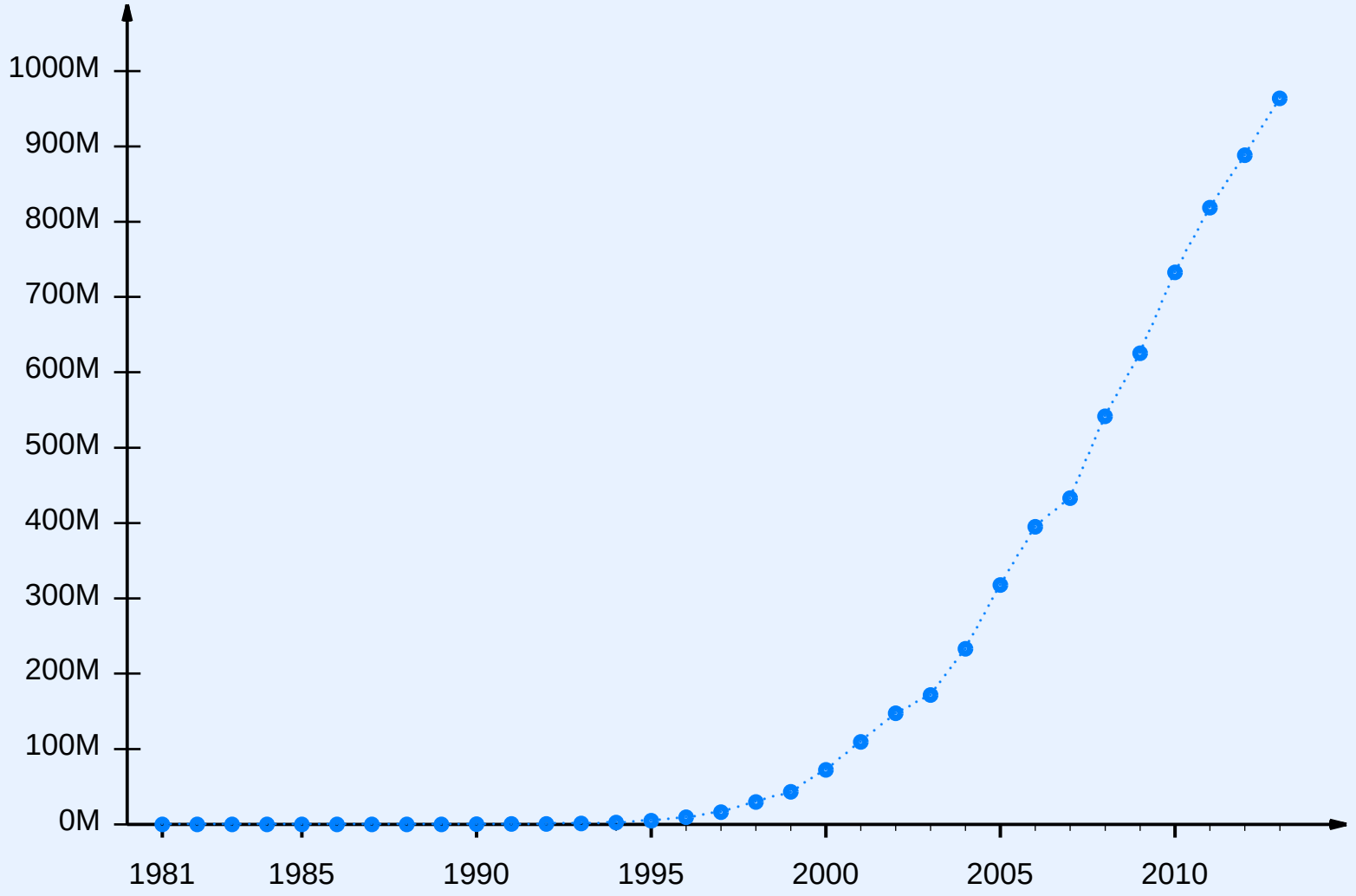
- Transport layer protocol characteristics and techniques
- Message transport with the User Datagram Protocol (UDP)
- Stream transport with the Transmission Control Protocol (TCP)
- Routing algorithms and protocols
- Internet multicast and multicast routing

Internet Concept
And Internet Architecture

What Is The Internet?

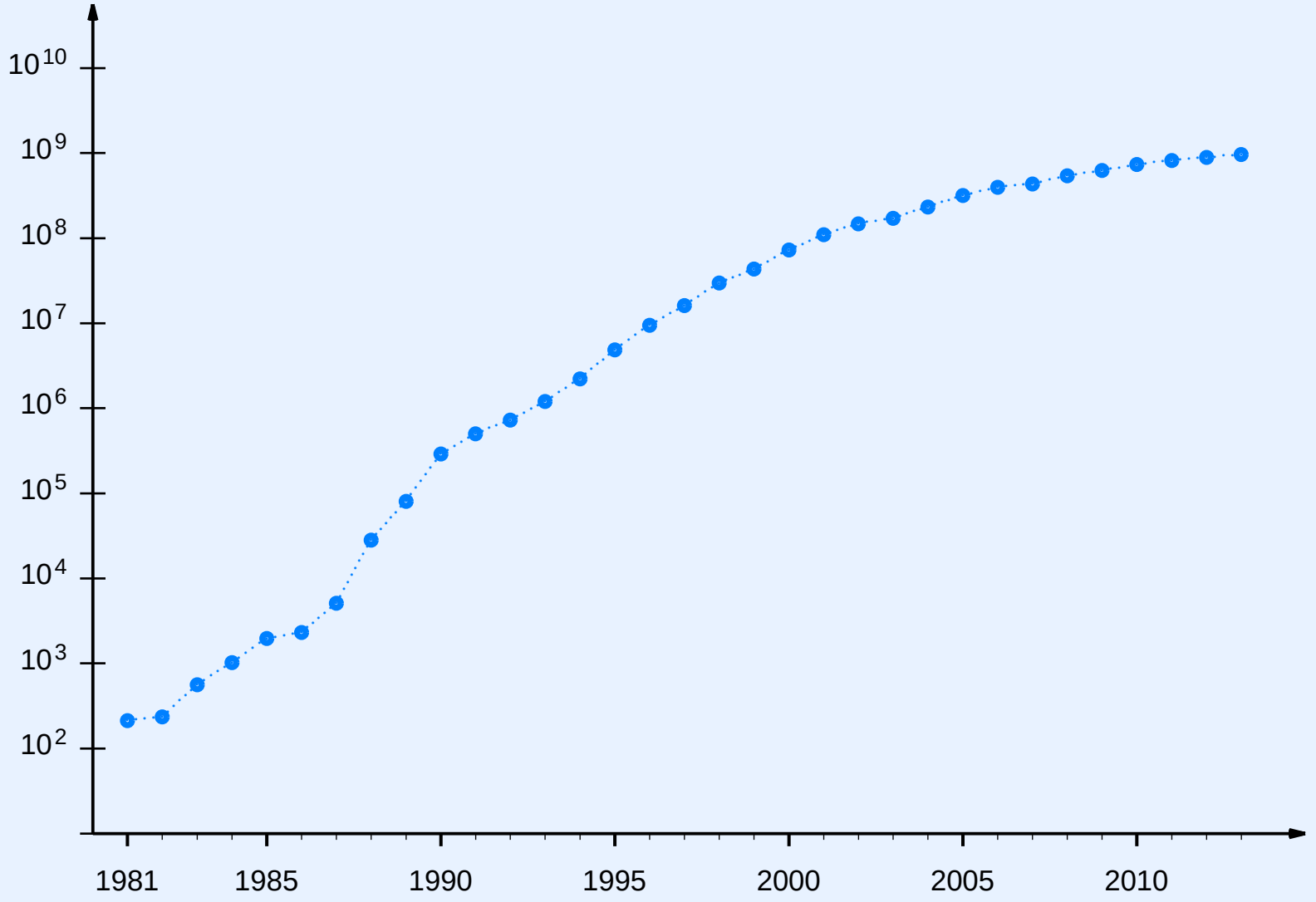
- Users see it as services and applications
  - Web and e-commerce
  - Email, texting, instant messenger
  - Social networking and blogs
  - Music and video download (and upload)
  - Voice and video teleconferencing
- Networking professionals see it as infrastructure
  - Platform on which above services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

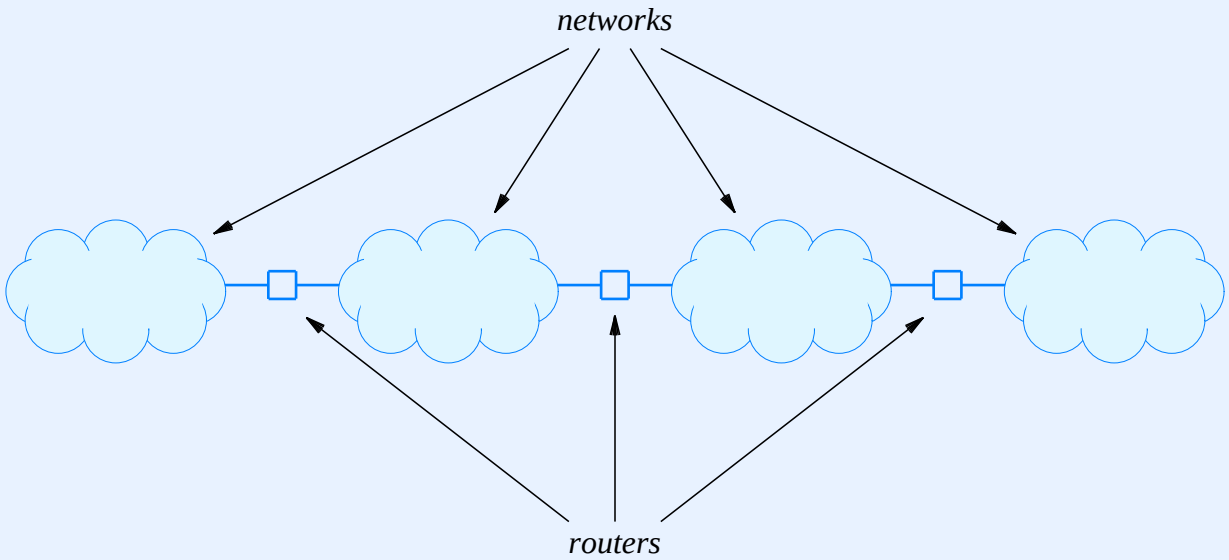
- Infrastructure
  - Provides a packet communication service
  - Treats all attached endpoints as equal (any endpoint can send a packet to any other endpoint)
  - Does not restrict or dictate packet contents
  - Does not restrict or dictate underlying network technologies
- Attached endpoints
  - Run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

Advantages Of The Internet Philosophy

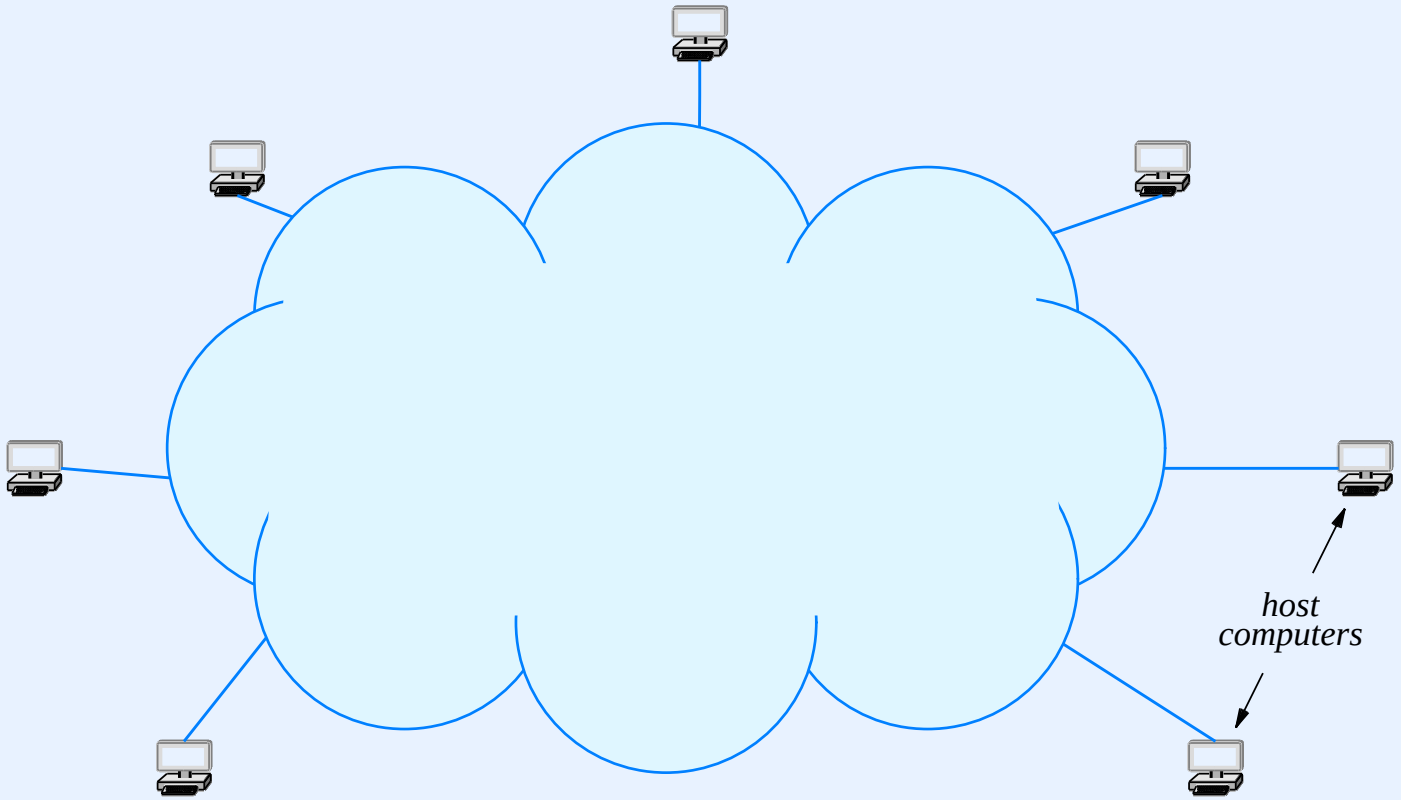
- Accommodates heterogeneous underlying networks
- Accommodates arbitrary applications and services
- Separates communication from services

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

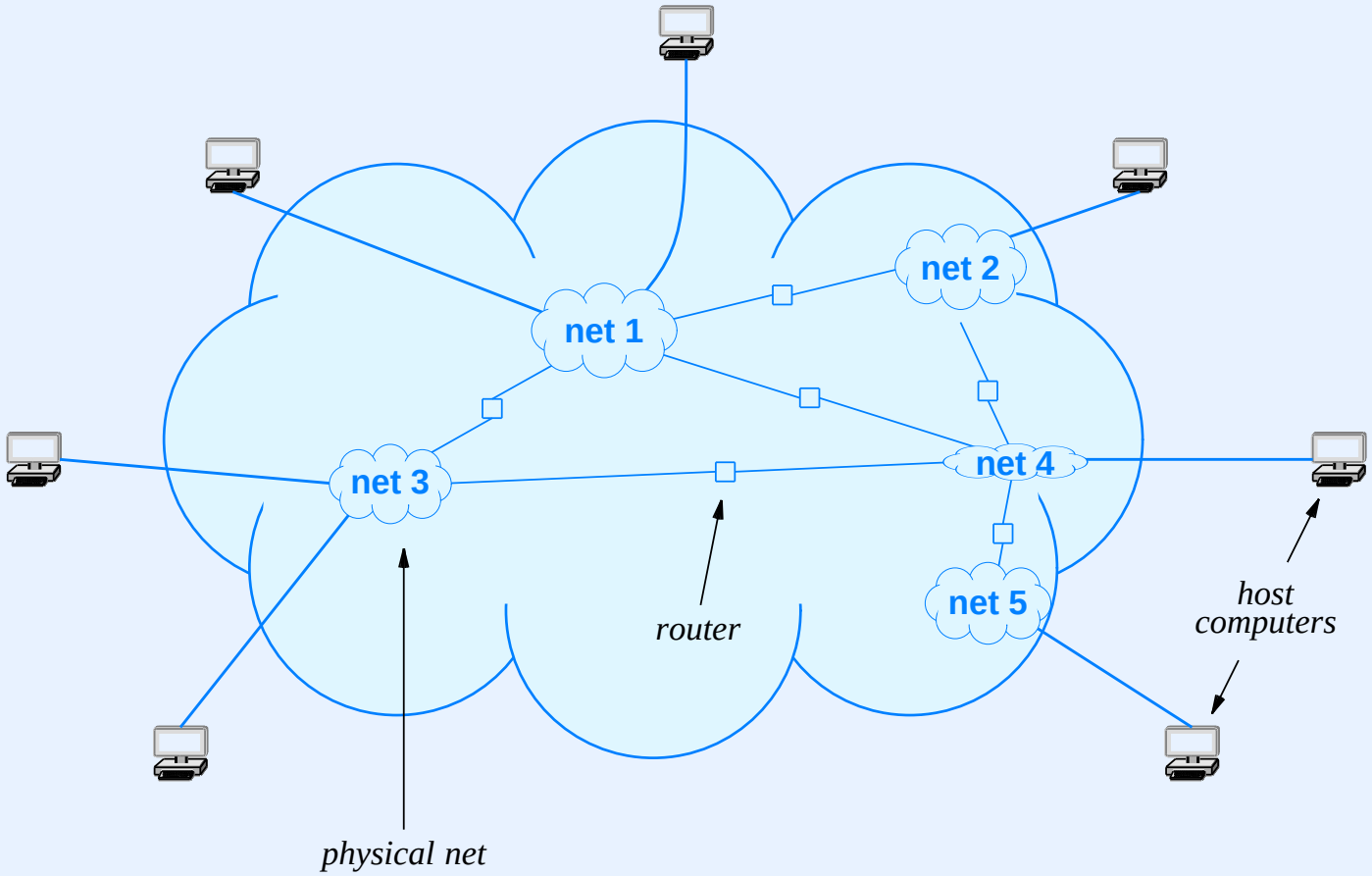


Internet Architecture: Logical View



- 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

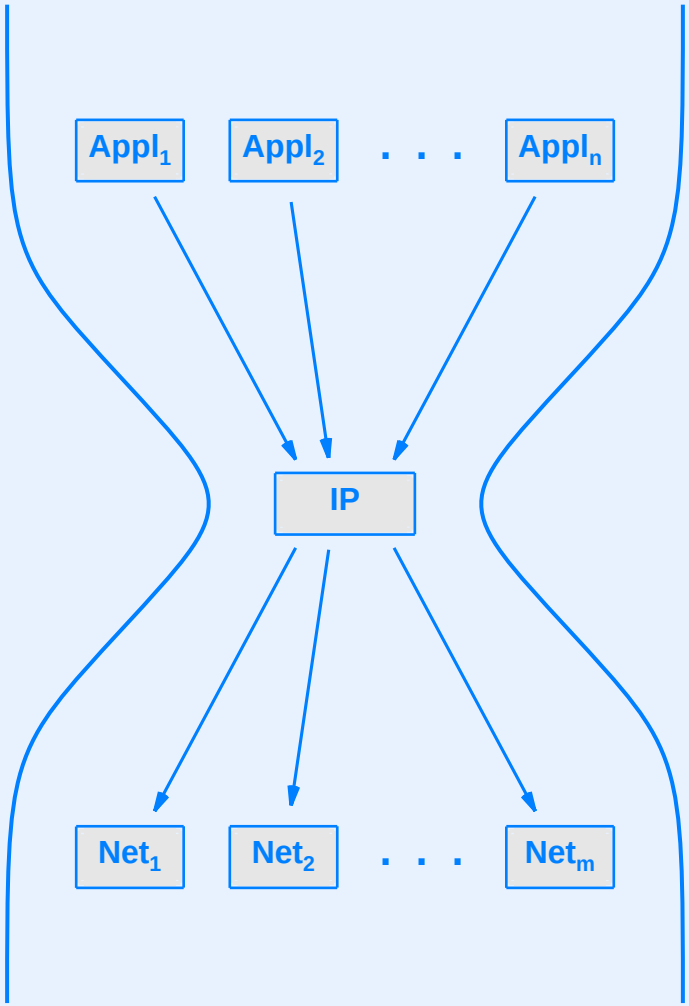
- Various groups offered opinions about the features
  - Cable companies wanted support for broadcast delivery
  - Telephone companies argued that everyone would soon be using a connection-oriented network technology (ATM)
  - Several groups wanted mobility
  - The military pushed for better security
- 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

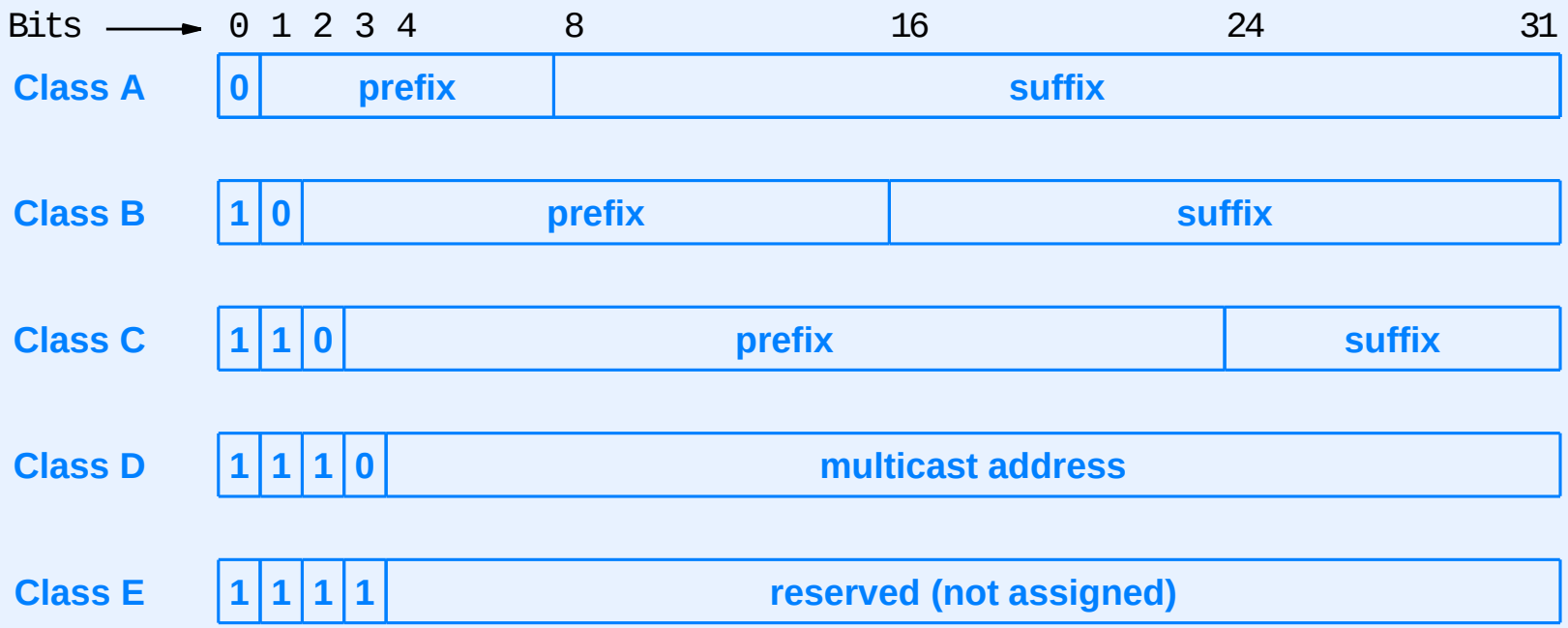
32-bit Binary Number				Equivalent Dotted Decimal
10000001	00110100	00000110	00000000	129.52.6.0
11000000	00000101	00110000	00000011	192.5.48.3
00001010	00000010	00000000	00100101	10.2.0.37
10000000	00001010	00000010	00000011	128.10.2.3
10000000	10000000	11111111	00000000	128.128.255.0

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

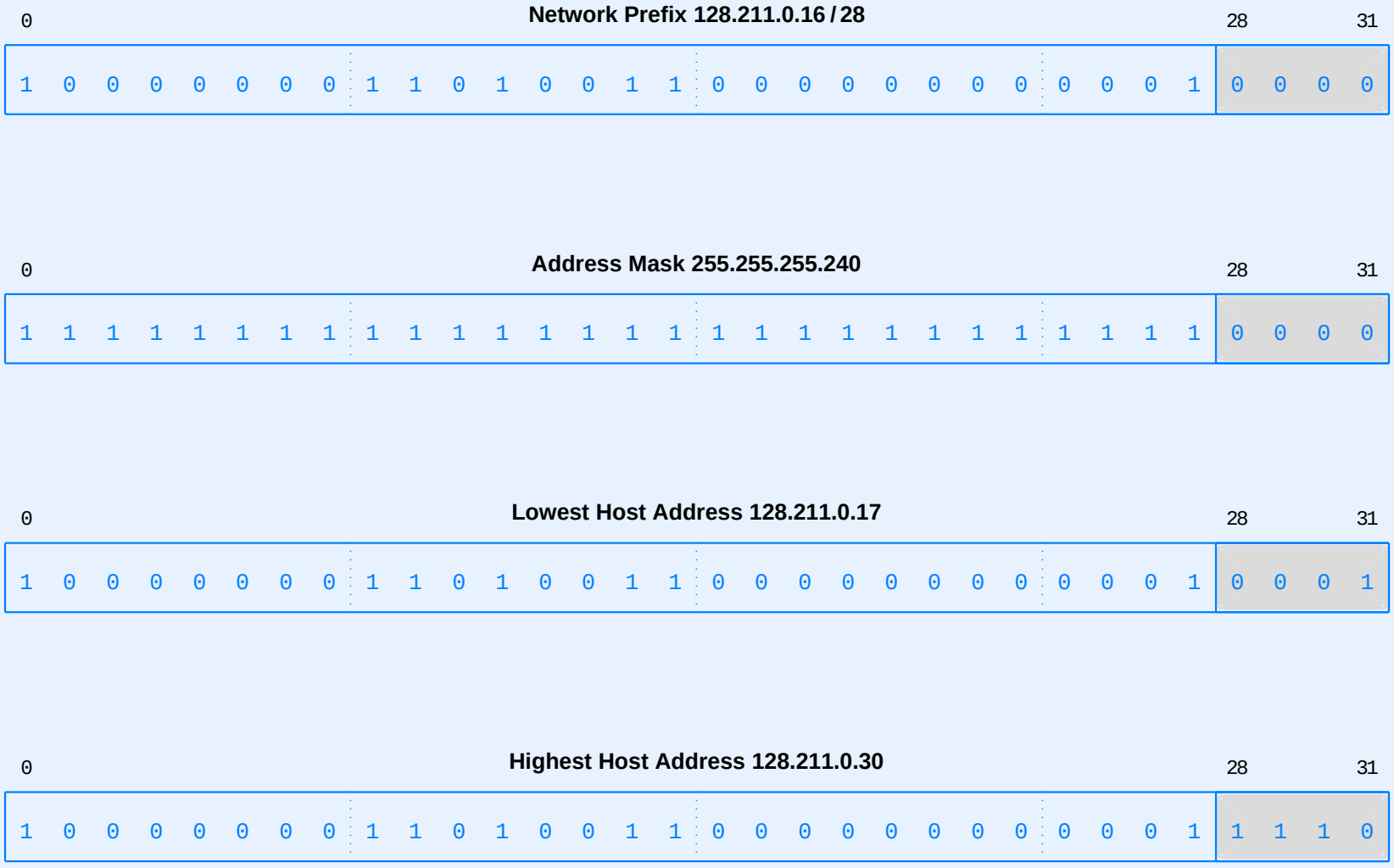
Table Of CIDR And Dotted Decimal Equivalences

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

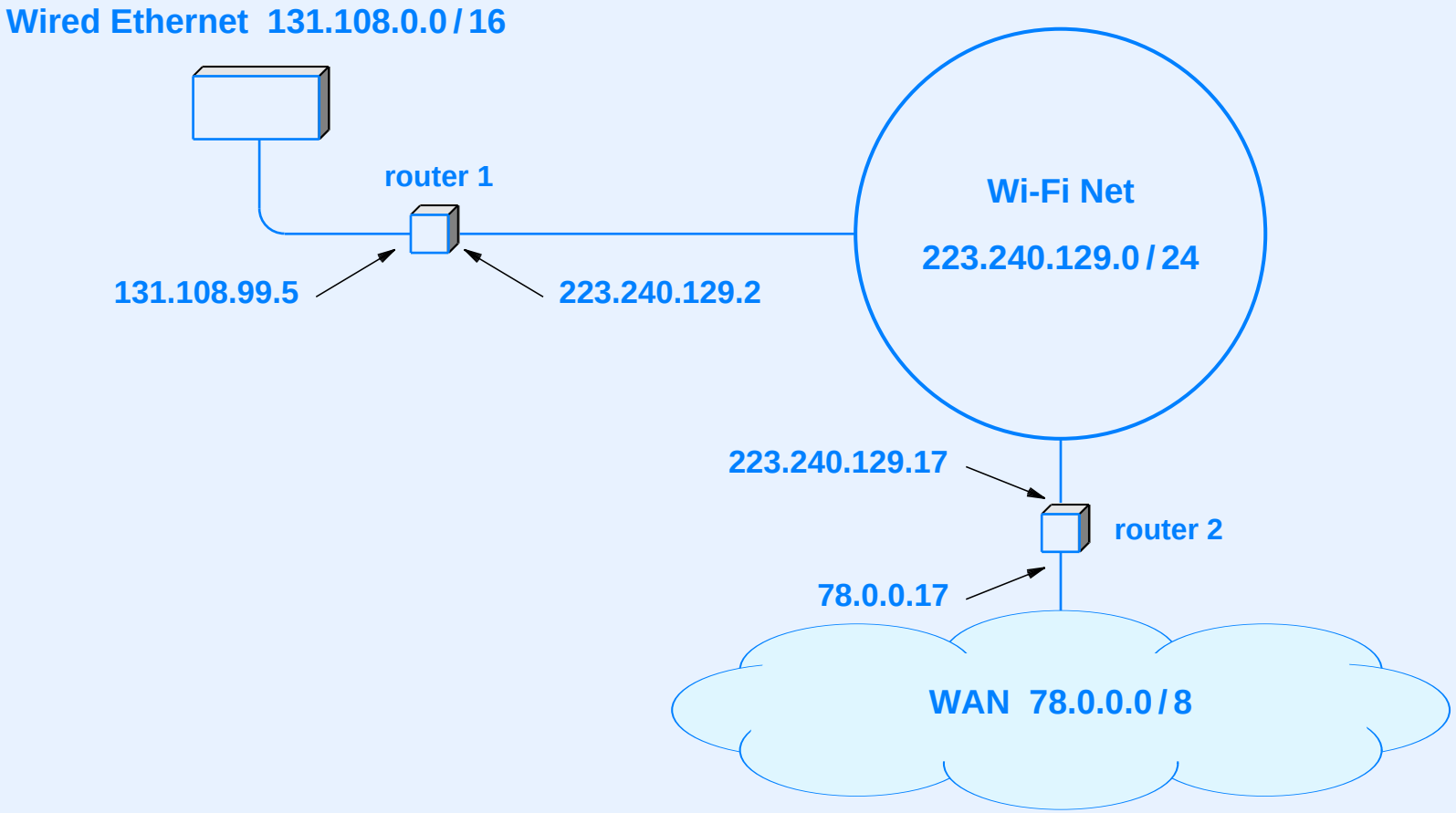
An IP address does not identify a specific computer. Instead, each IP address identifies a connection between a computer and a network.

- 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

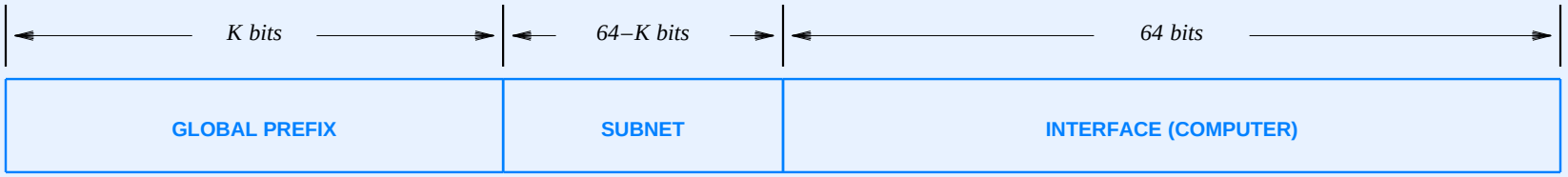


- Each network assigned a unique prefix
- Each host on a network assigned a unique suffix

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.

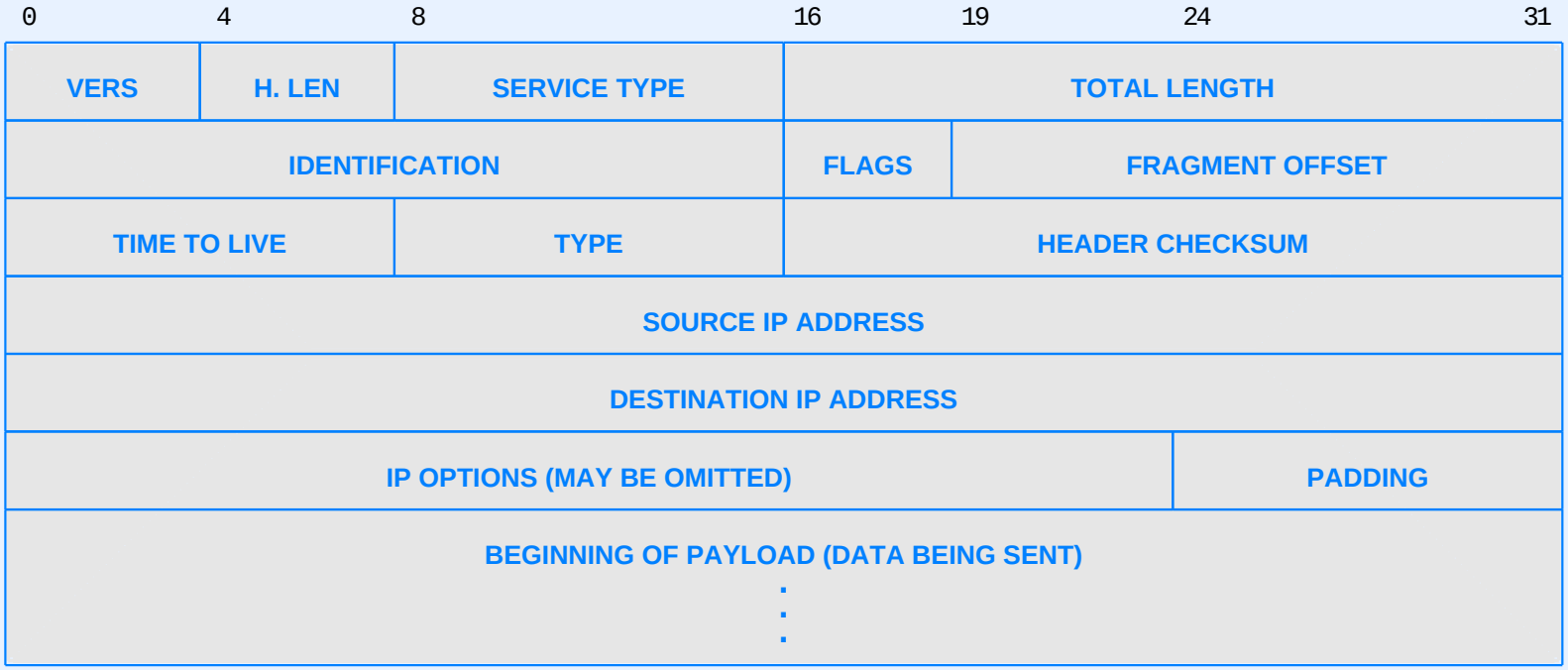
IP Datagram

- 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

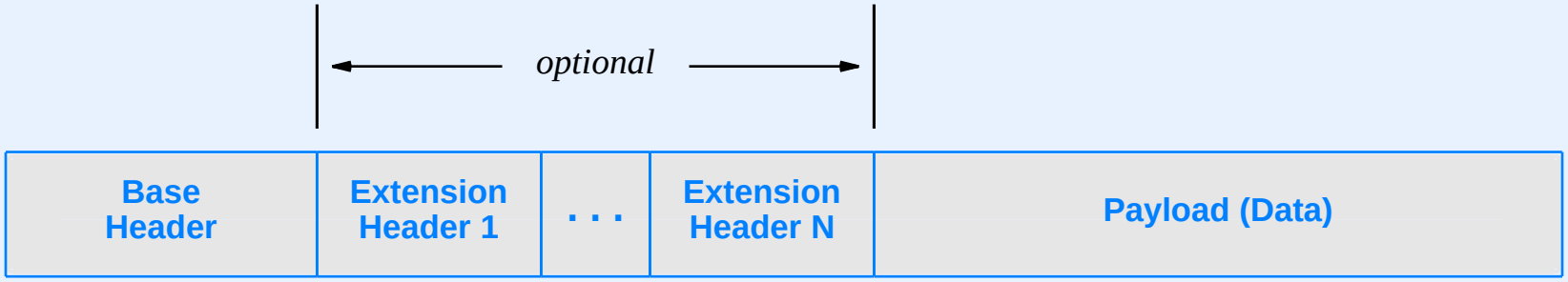


A Few Details

- *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

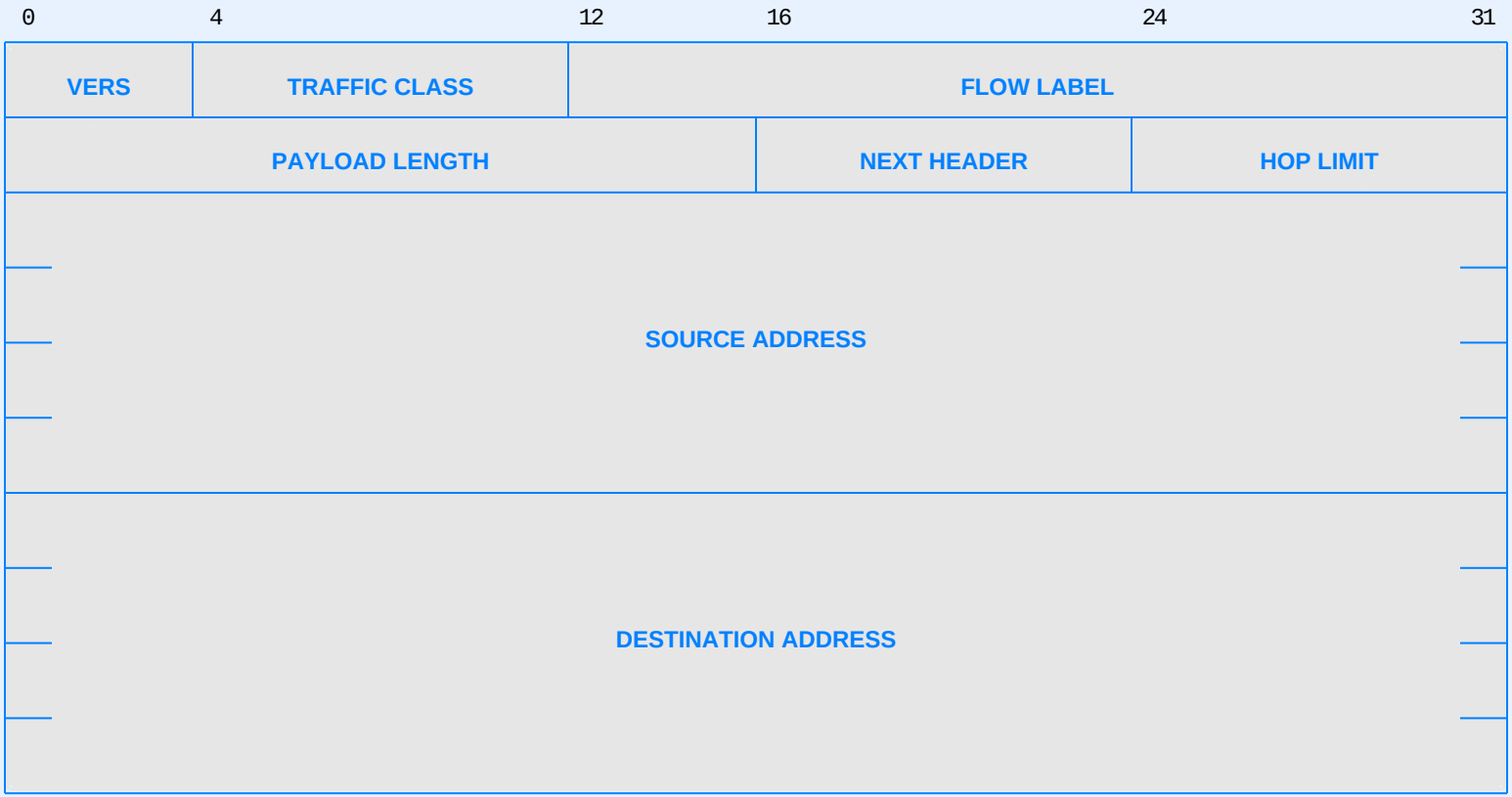
IPv6 Header Arrangement

- 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



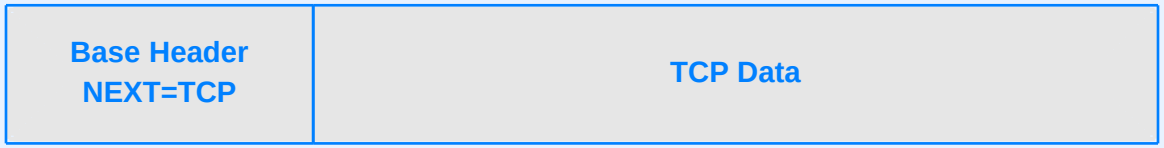
- *Flow Label* field allows datagram to be associated with a flow

Identifying Headers

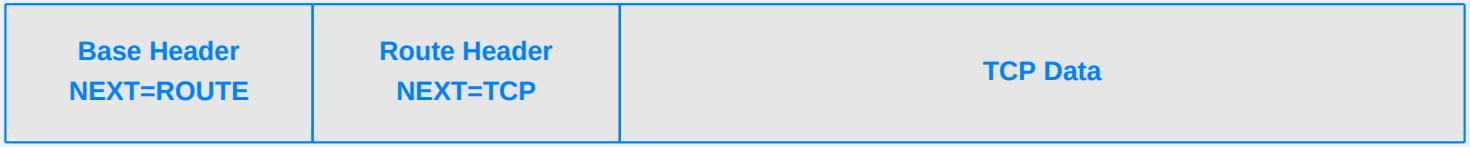
- Each header contains a *NEXT HEADER* field
- Value specifies the type of the next item
- Each layer 4 protocol (UDP, TCP, etc) is also assigned a type

Example Use Of Next Header Field

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

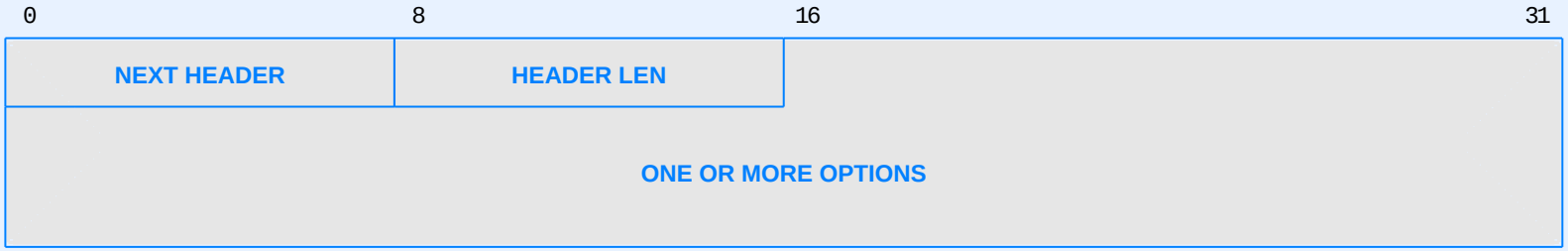


- Illustration of headers when a datagram also contains an optional *route header*



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

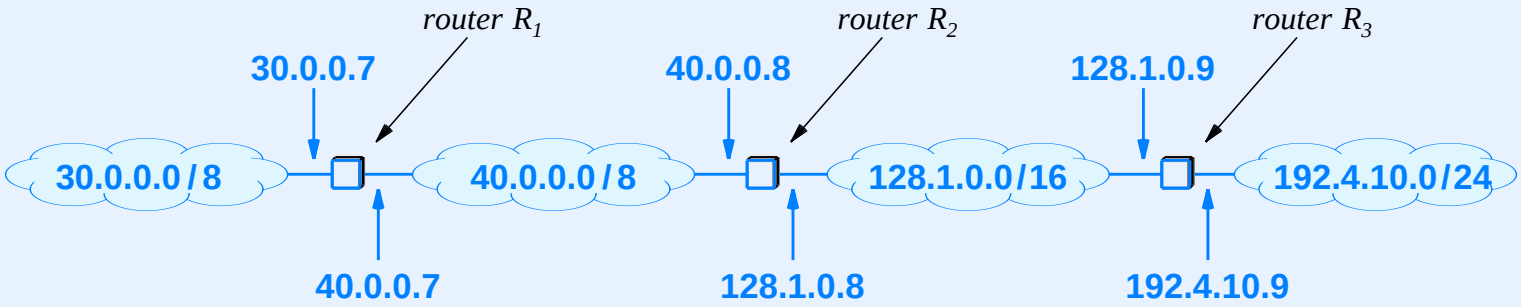
Datagram Forwarding

- 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

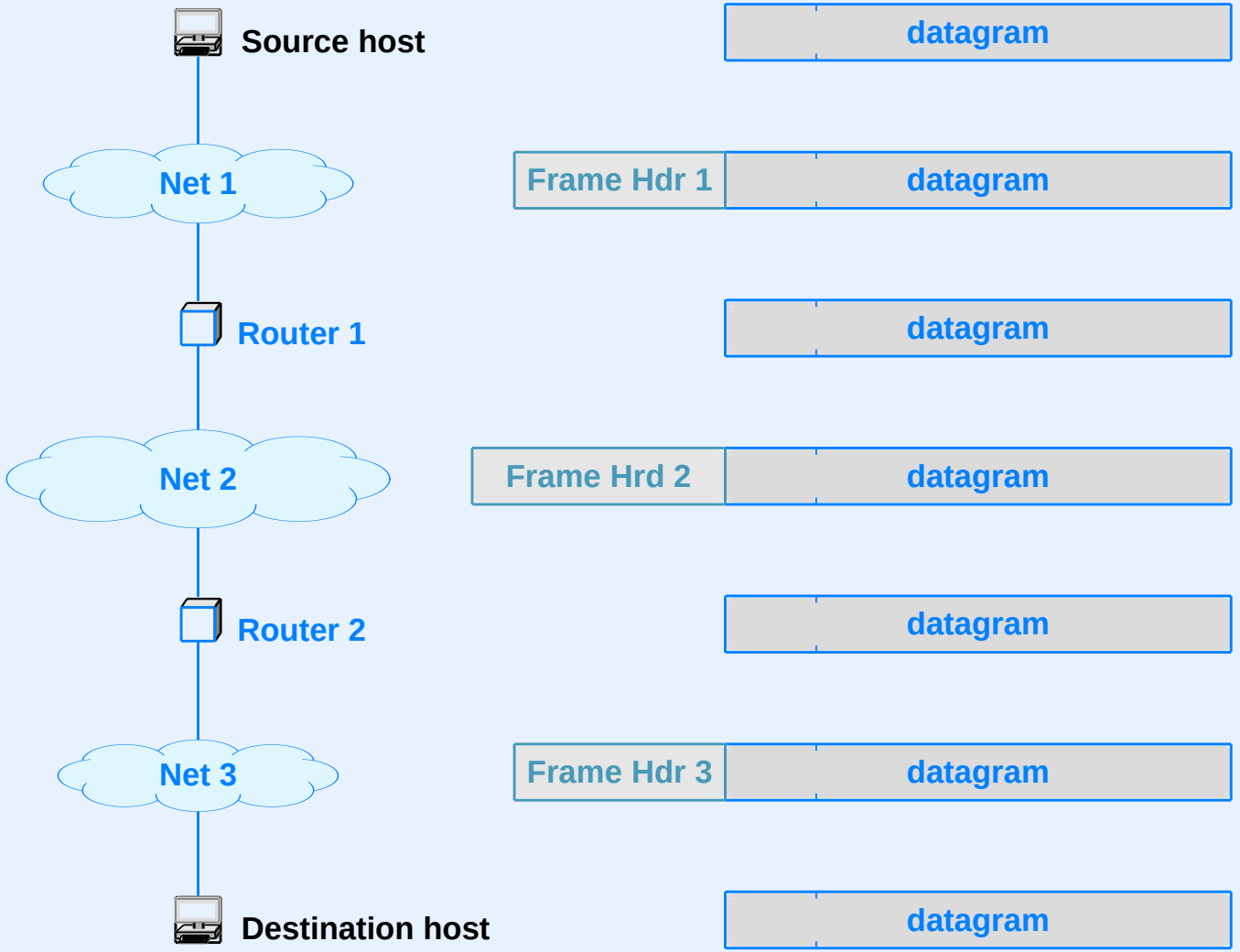
- The destination matches both of them!
- Solution: select the match that has the longest prefix (in the example, take next hop B)
- Known as *longest prefix match*

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

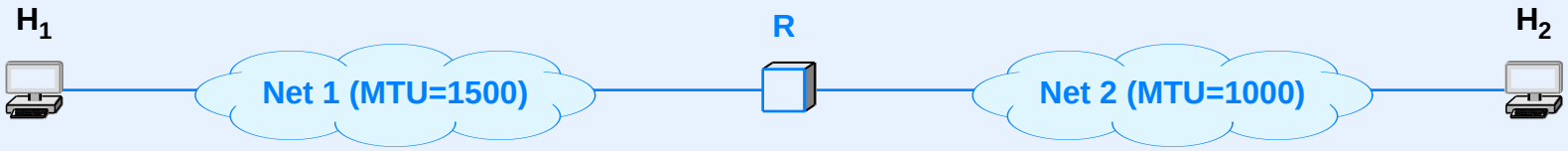


Semantics Of Internet Communication

- 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

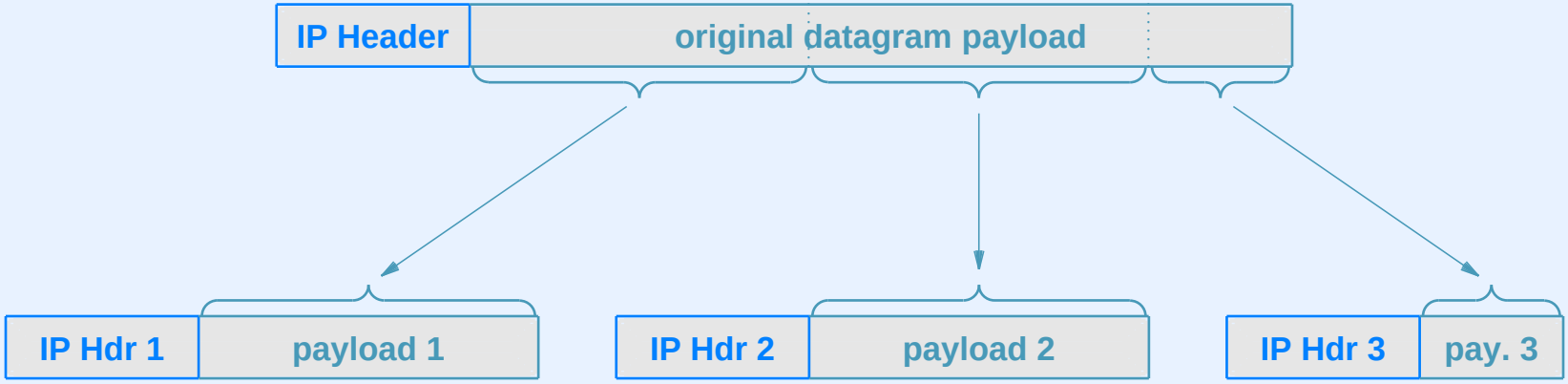
Datagram Fragmentation

- 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

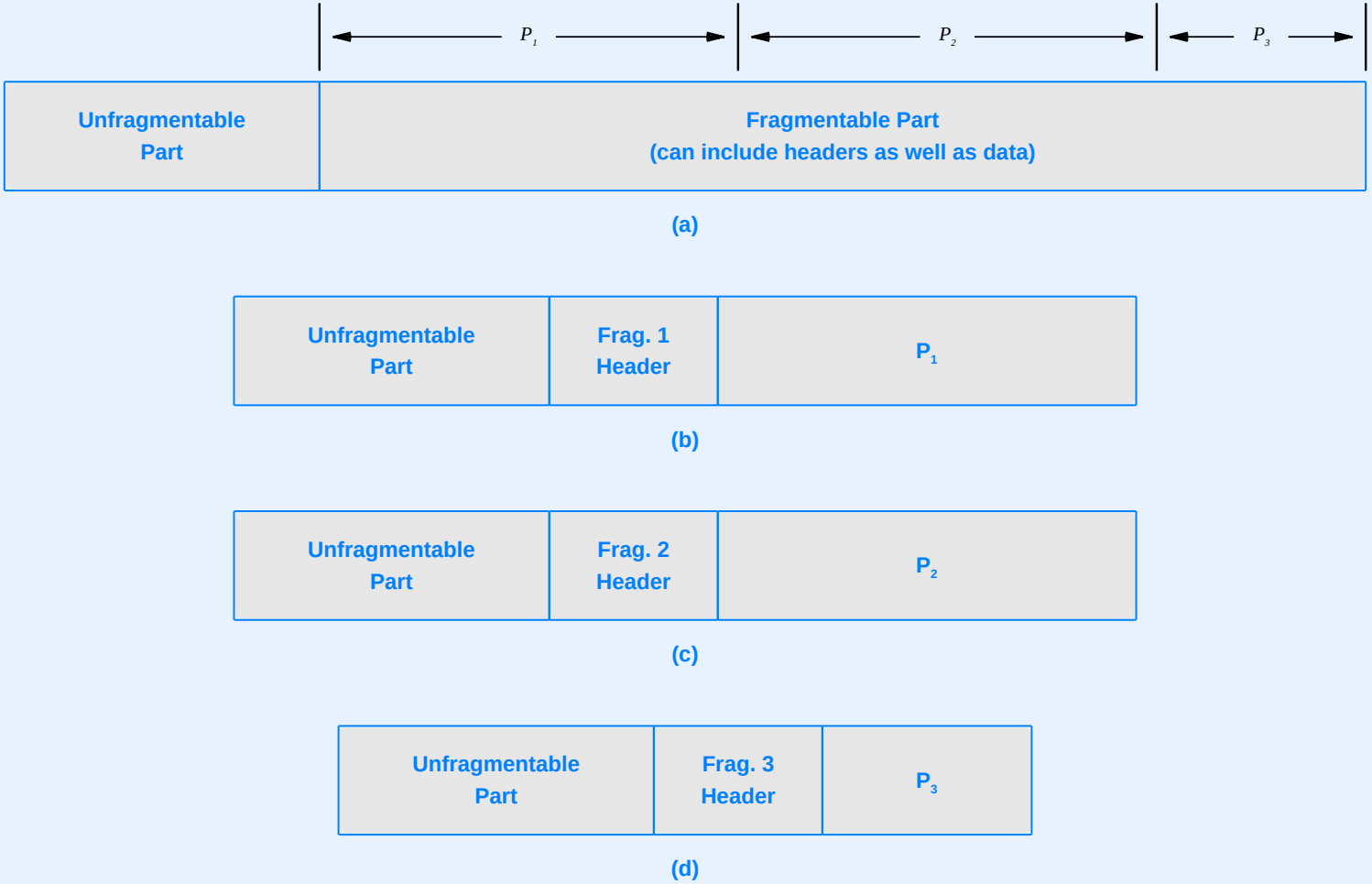
IPv4 Fragmentation Details

- 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 the original source, never by routers
- Rule: no header changes are allowed 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!

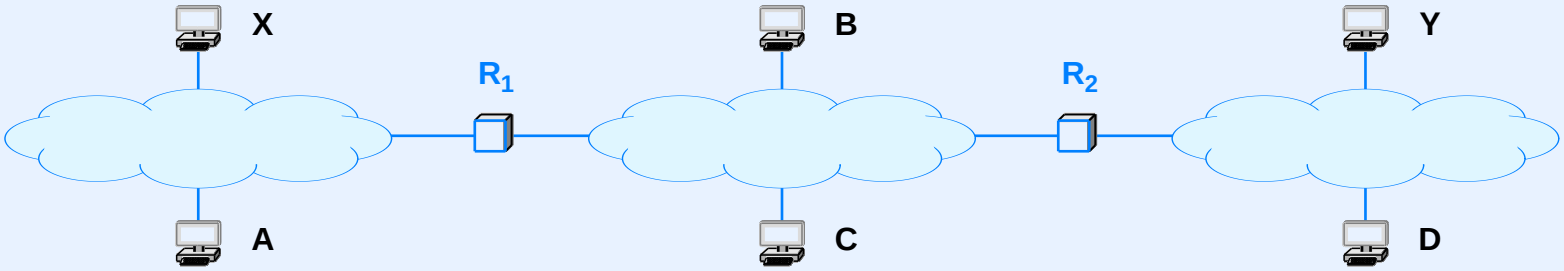
Hardware And Protocol Addressing

- 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.

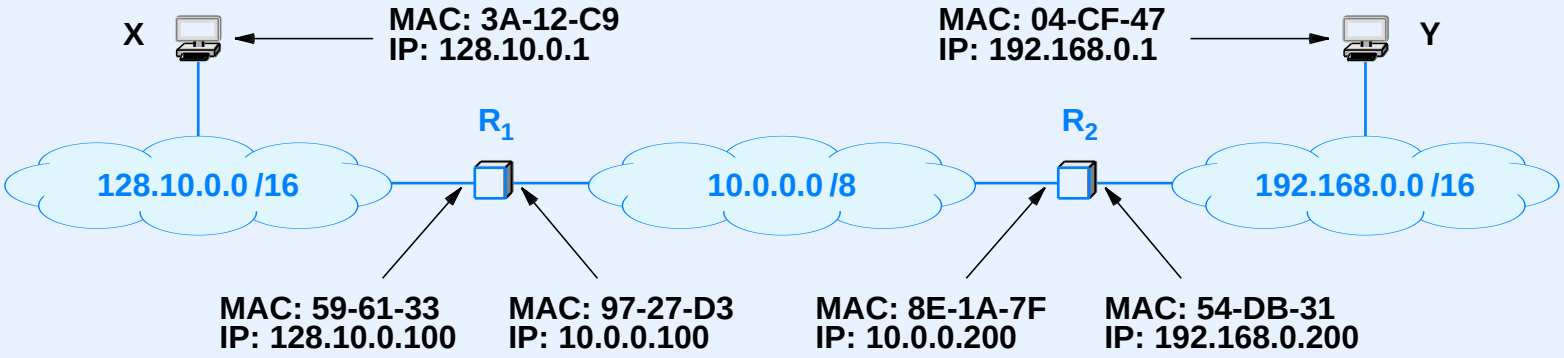
Address Resolution

- 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?

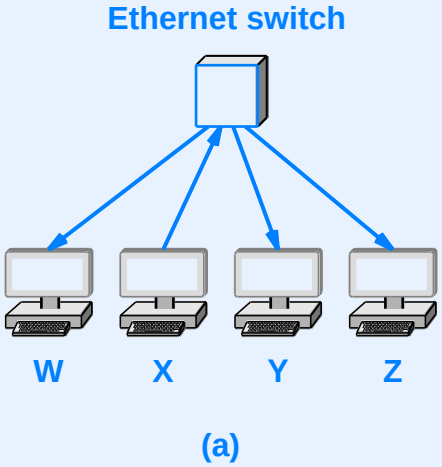
Address Resolution Protocol (ARP)

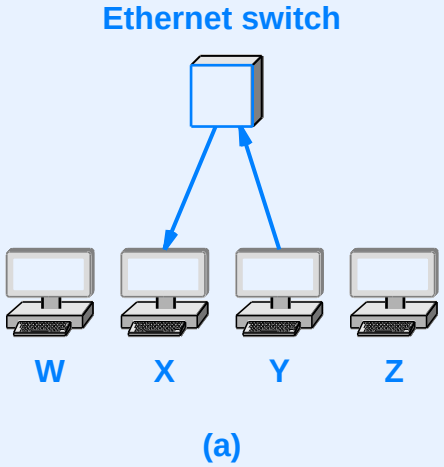
- Designed for IPv4 over Ethernet
- Used by two computers on the same physical network
- Allows a computer to find the MAC address of another computer
- Operates at layer 2
- Uses network to exchange messages
- Computer seeking an address sends request to which another replies

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

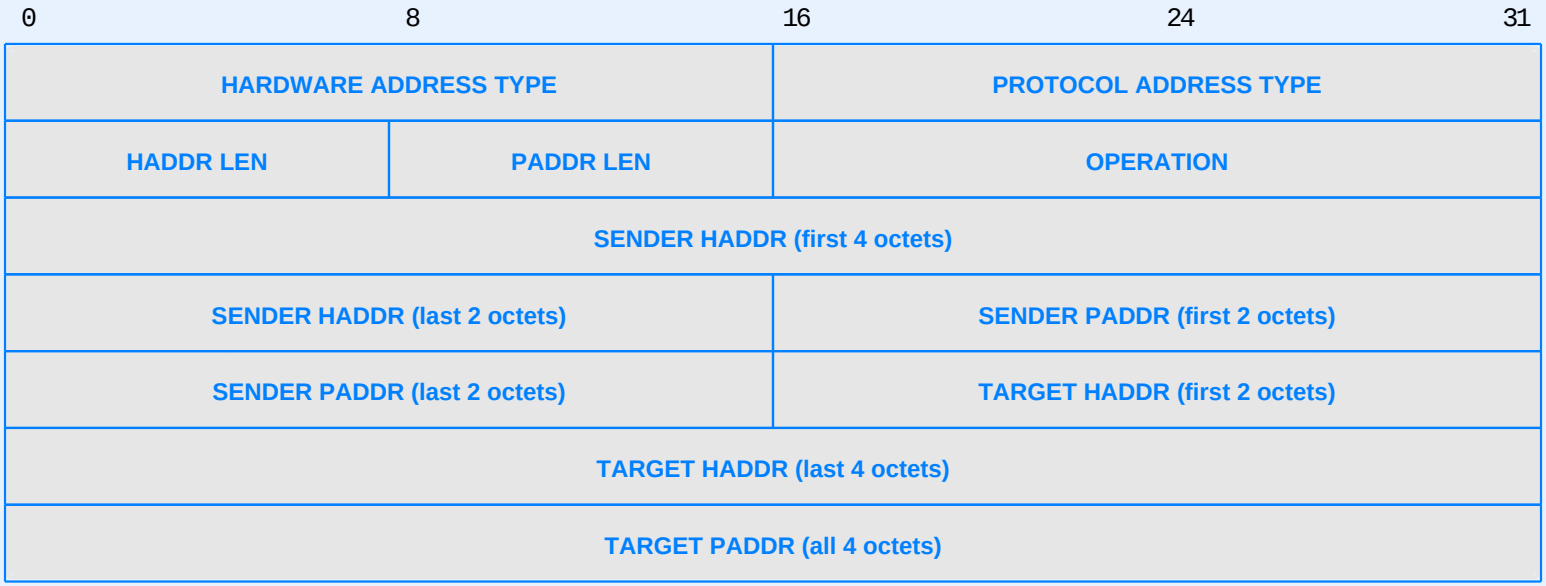




- Request is broadcast to all computers
- Only the intended recipient replies
- Reply is sent unicast

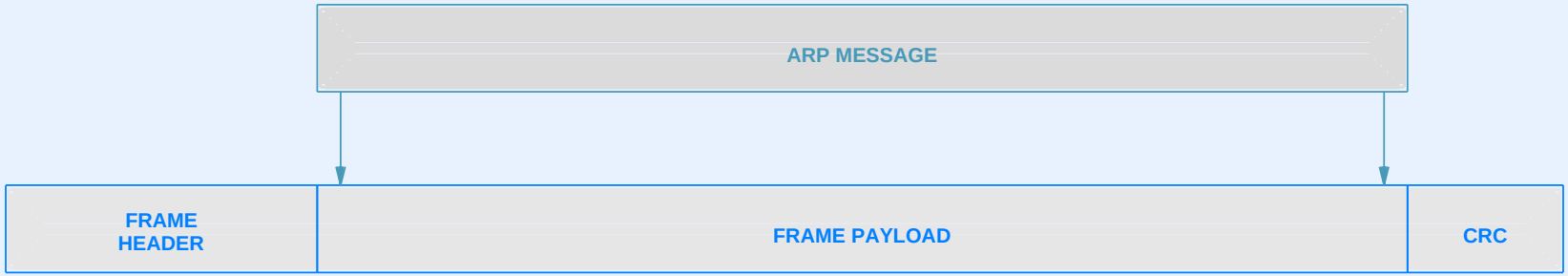
ARP Message Format

- 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:

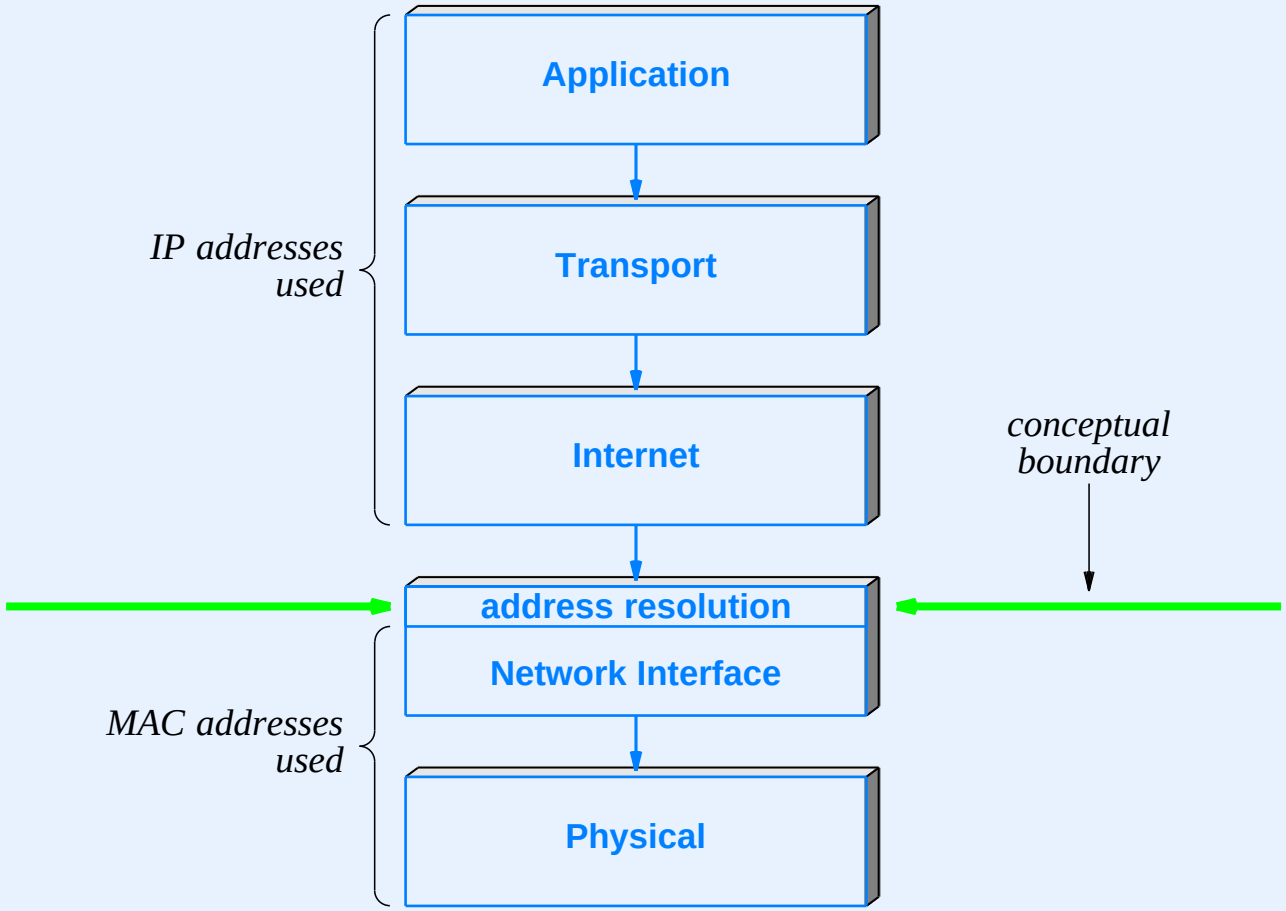
Process the message and update the ARP cache

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*)
- IPv6-ND
  - Maintains a neighbor cache
  - Keeps the cache up-to-date at all times
- IPv6-ND operation
  - Sends a multicast request to find neighbors and populate the cache
  - Polls neighbors periodically, even if no datagrams are being sent to the neighbor

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

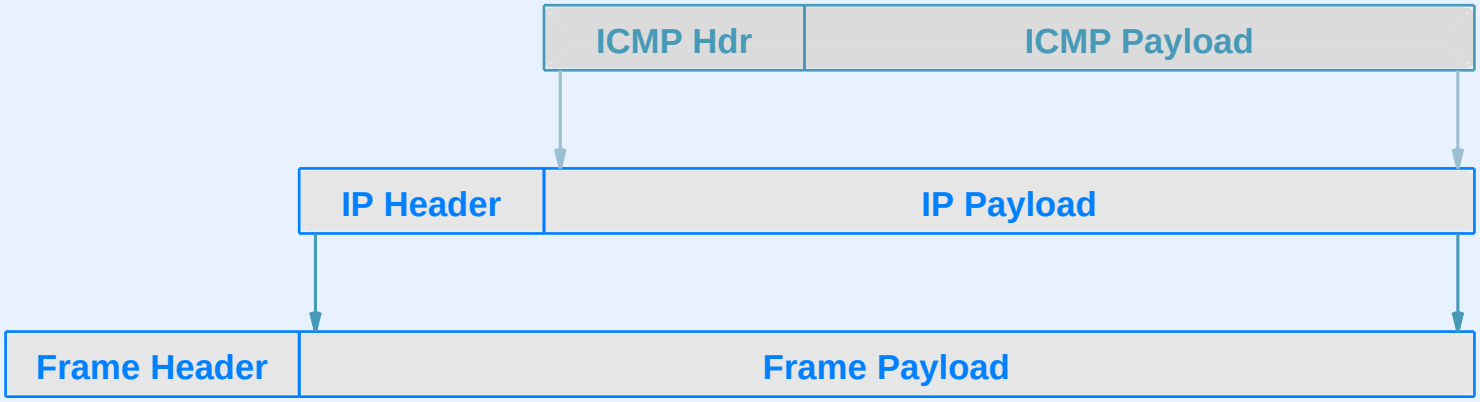
Example ICMP Messages

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

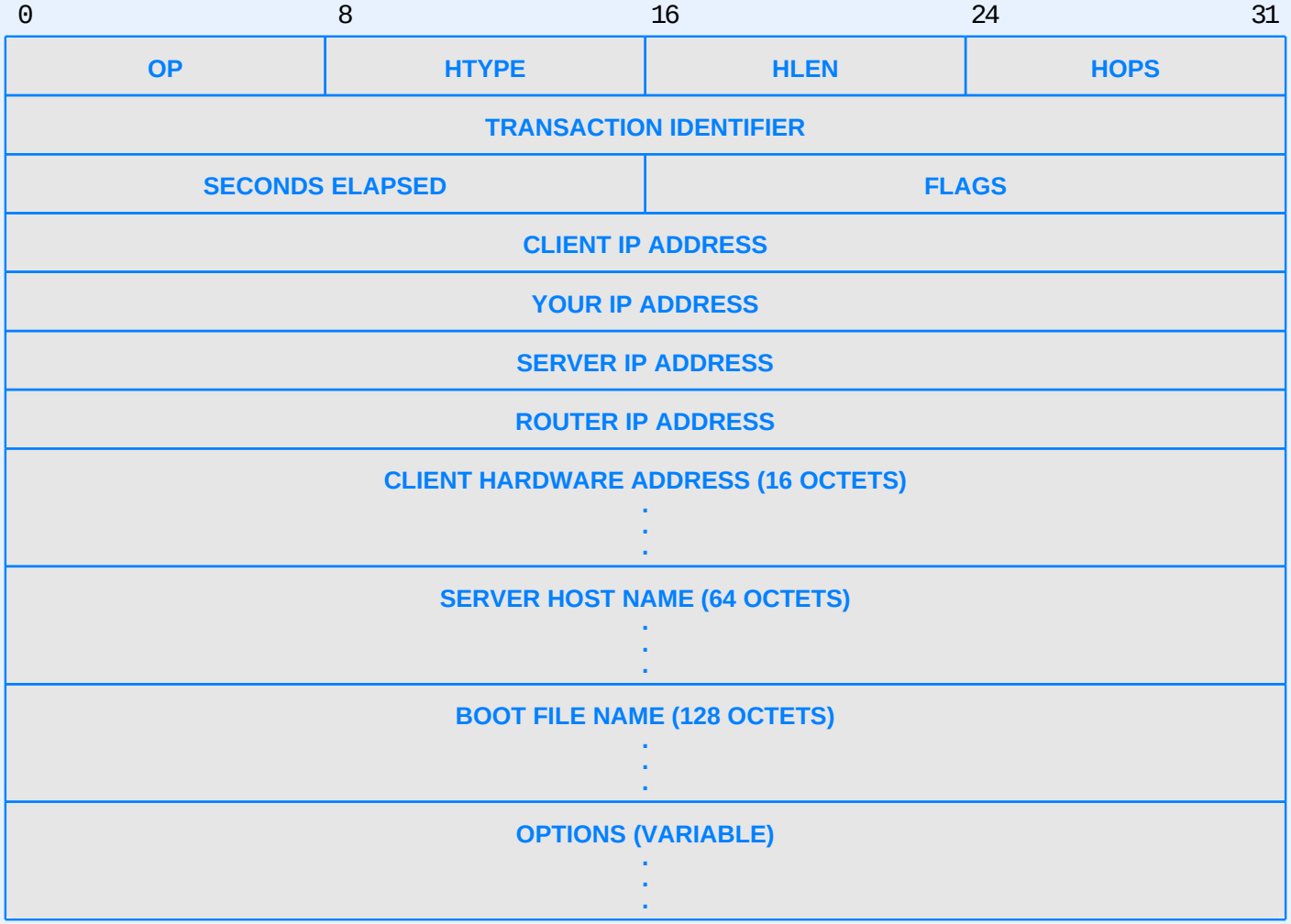
- Use layer 2 protocols to obtain layer 3 parameters, then use layer 3 to obtain higher layers
  - Historic approach
  - Relied on Ethernet broadcast
  - One computer on a network responded to requests
- Use layer 3 to obtain all parameters
  - Current approach
  - Relies on IP broadcast (IPv4) or multicast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

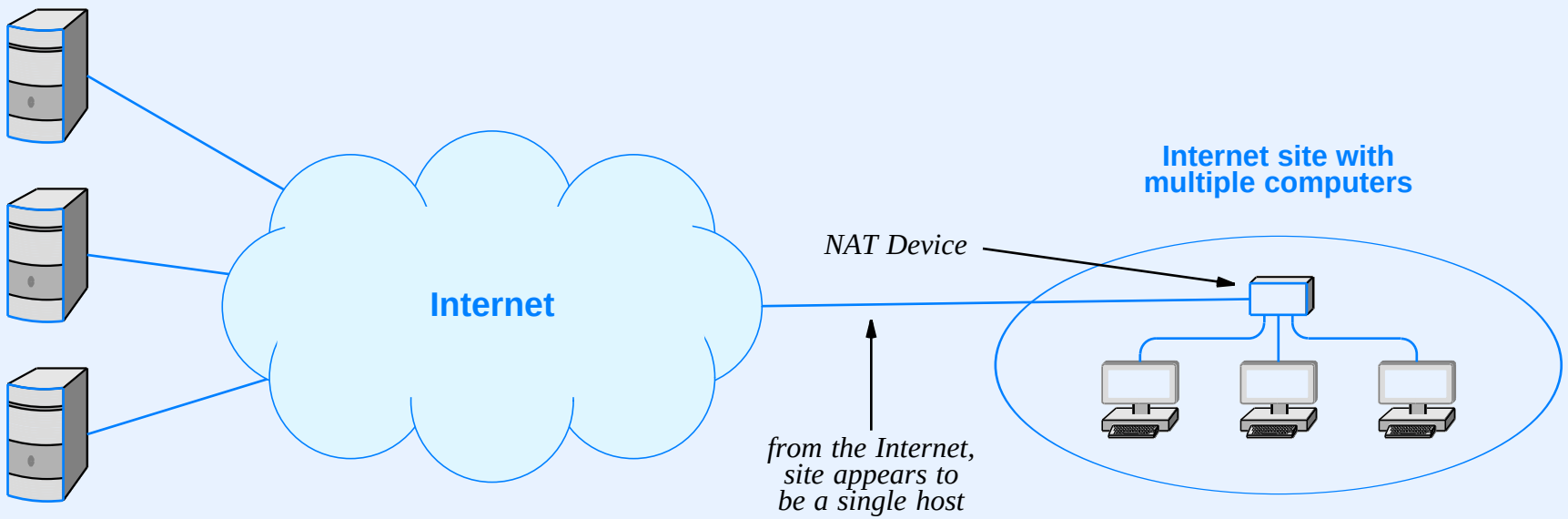
NAT Motivation

- 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

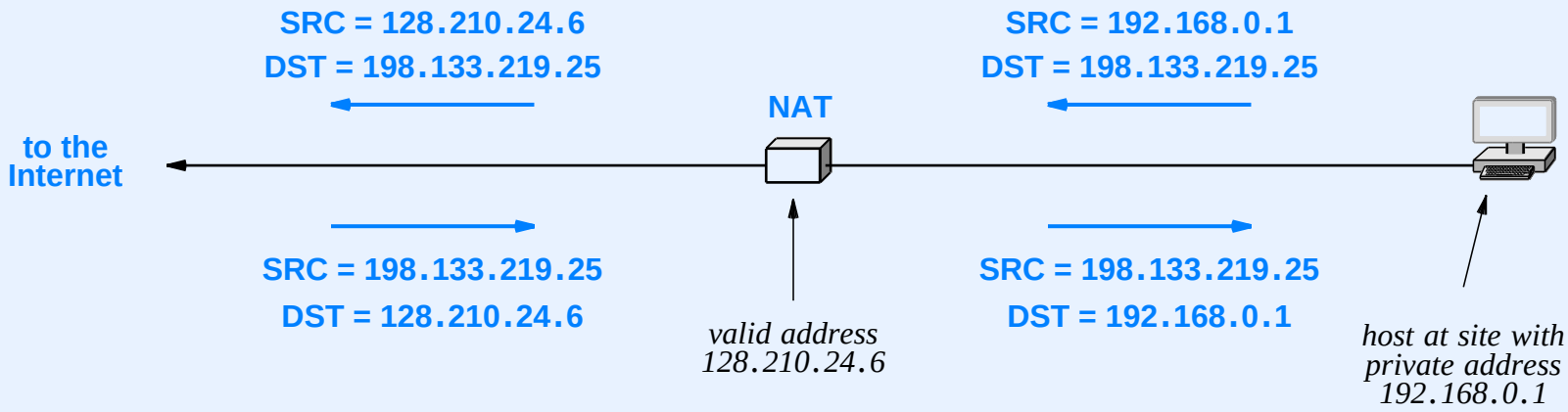
  - NAT translates source and/or destination addresses in datagrams that pass between the site and the Internet

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

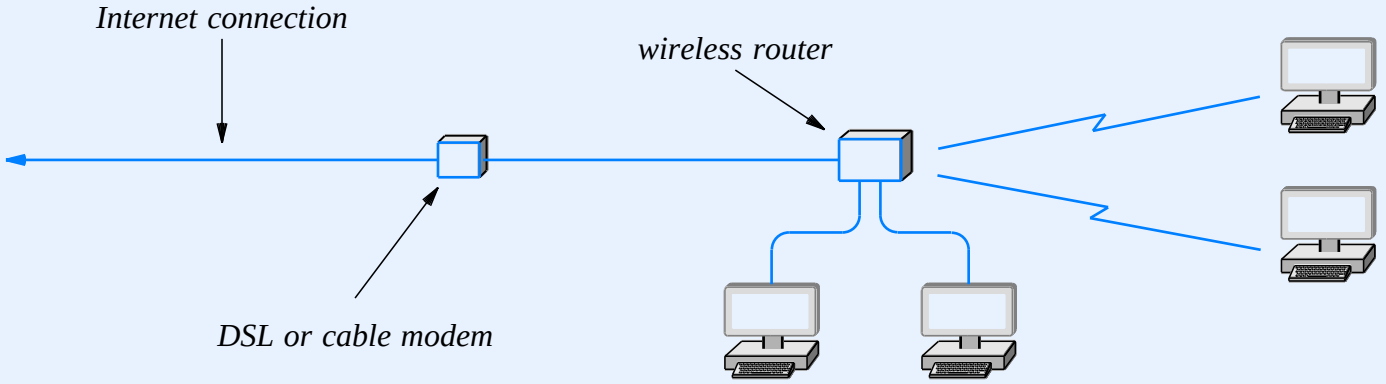
Example Of NAPT Translation

- 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*

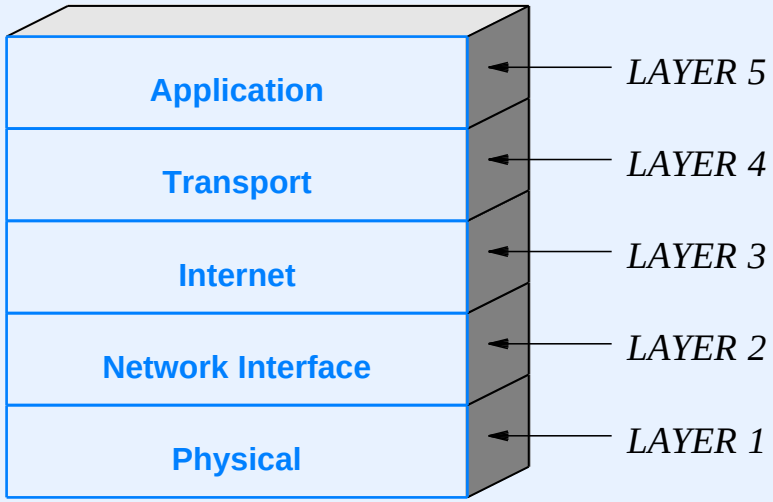
End-To-End Principle

- 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

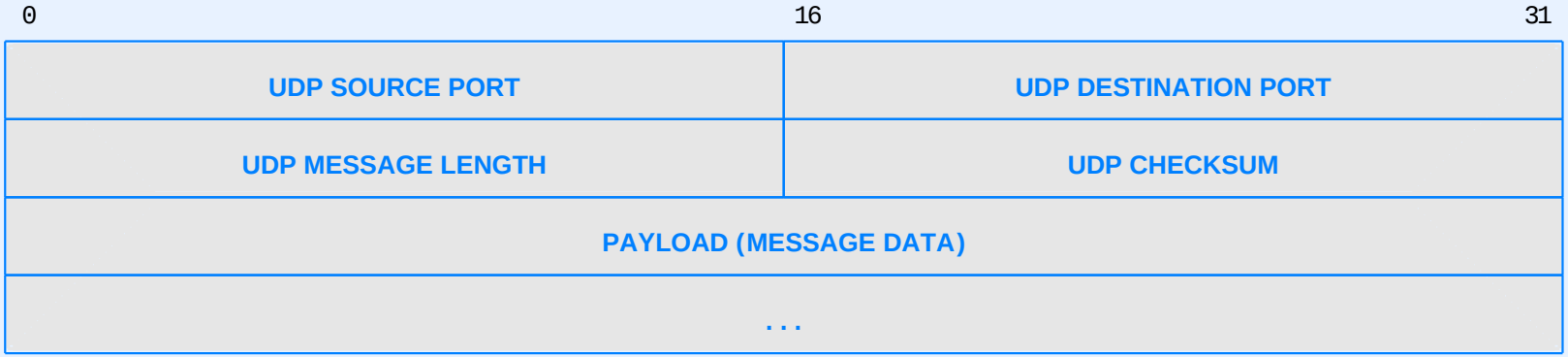
- Request an unused local port number from the local operating system
- Obtain the IP address of the local computer from the operating system
- Look up the port number of the service to be contacted
- Obtain the domain name of a computer that runs the service and map to an IP address
- Form a UDP datagram 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

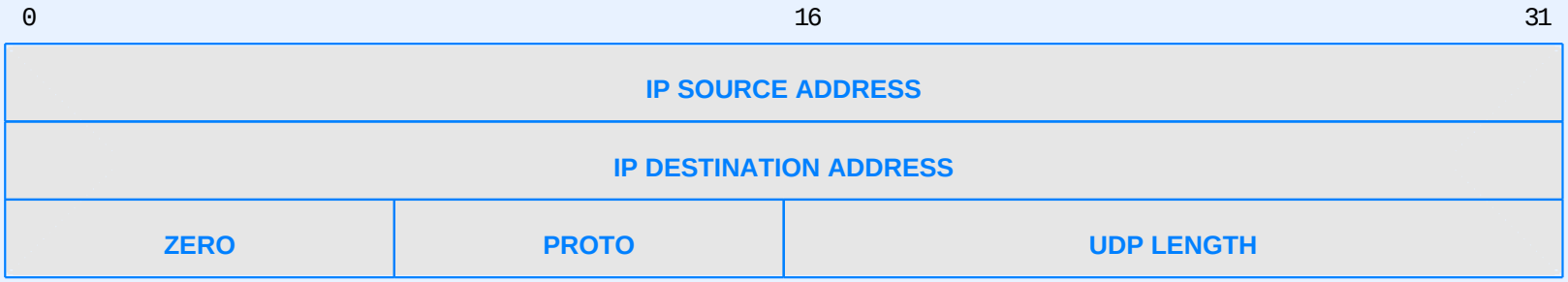
- Extremely thin layer
- User Datagram is divided into *header* and *payload*
- Header contains only 8 octets:



- 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:

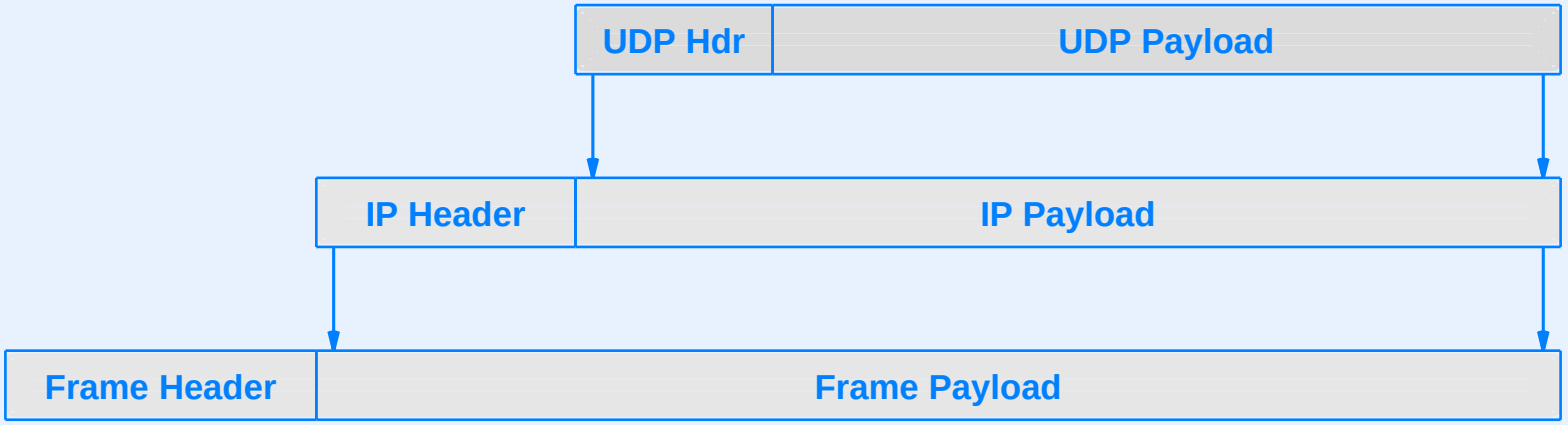


Purpose Of A Pseudo Header

- 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

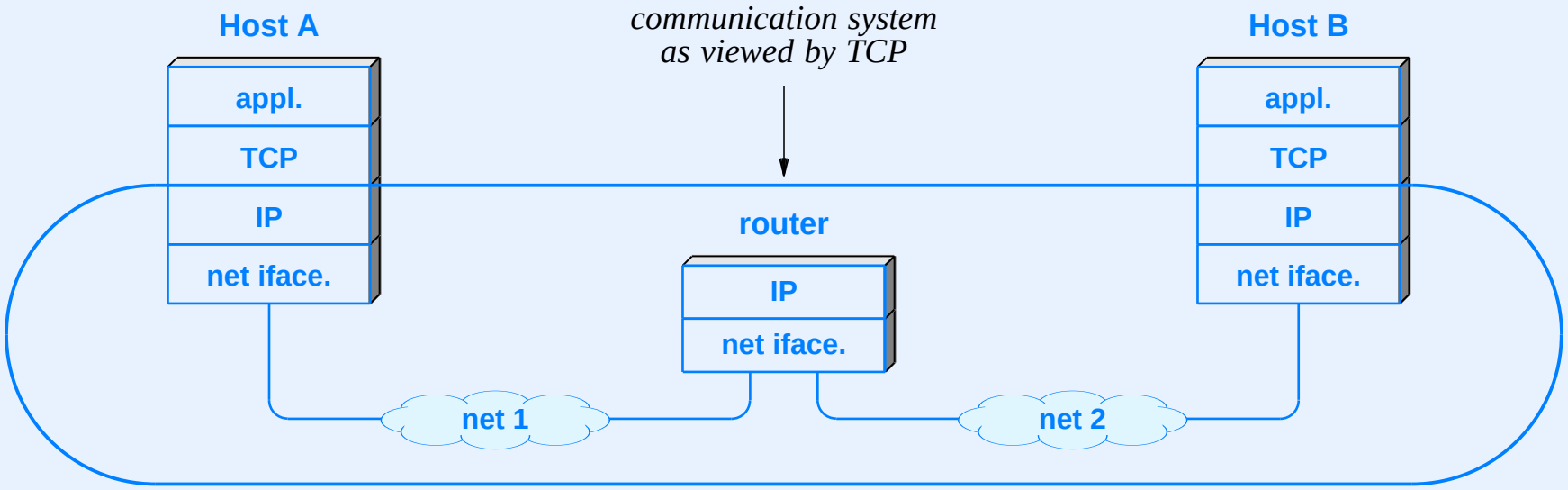
- End-to-end communication
- Connection-oriented paradigm
- Point-to-point connections
- Complete reliability
- Full-duplex communication
- Stream interface
- Reliable connection startup
- Graceful connection shutdown

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

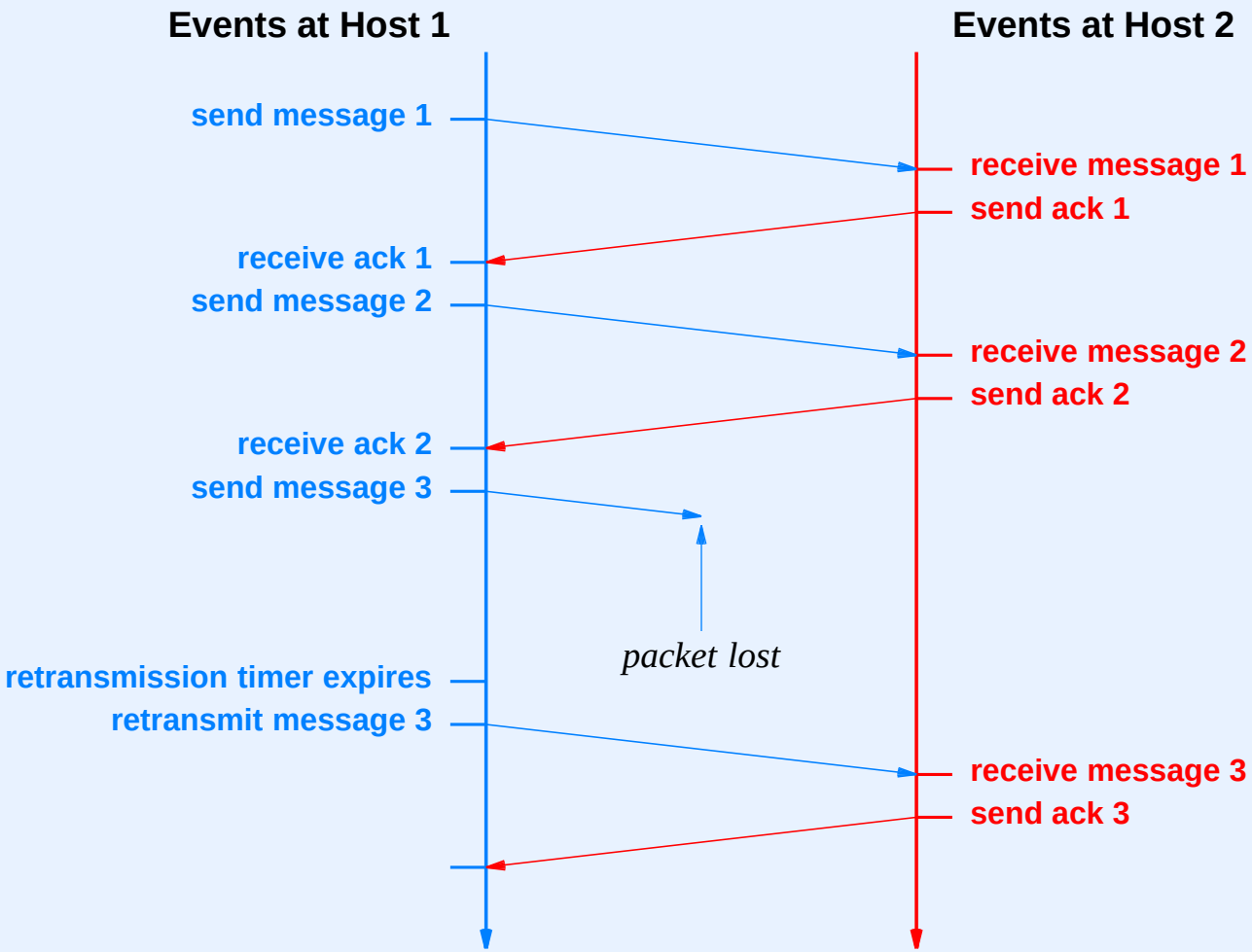
- TCP provides full reliability
- Compensates for
  - Loss
  - Duplication
  - Delivery out of order
- Does so without overloading the underlying networks and routers
- TCP makes the following guarantee

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

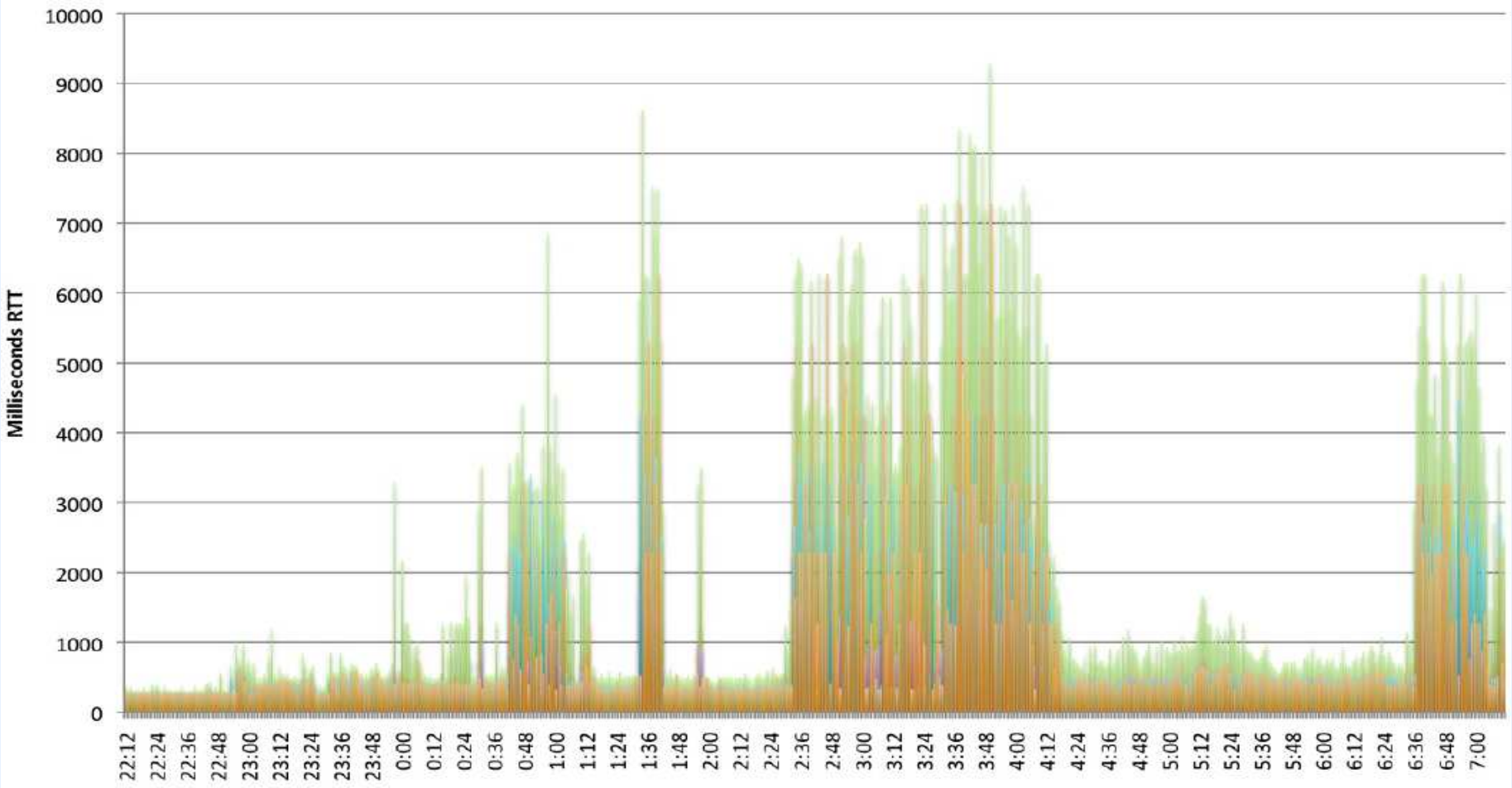


Why TCP Retransmission Is Hard

- 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?

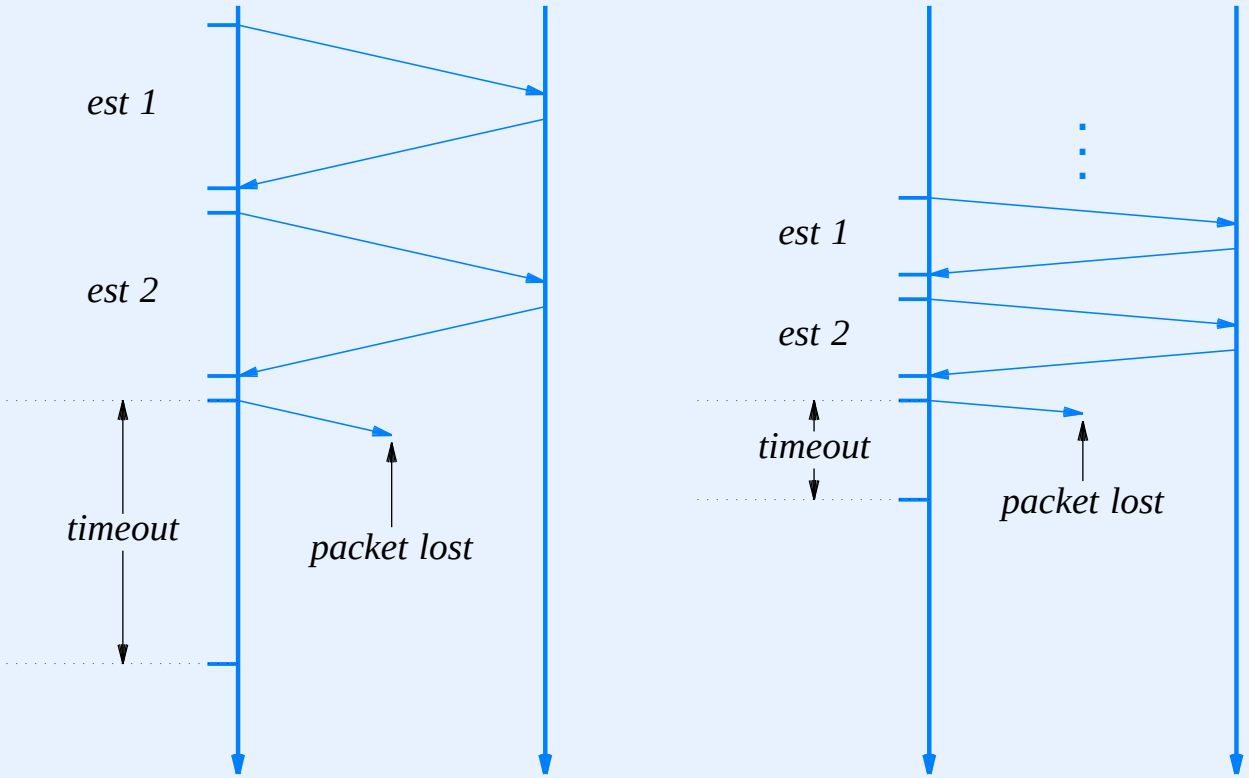
- In the old days: delays in seconds, high variability
- Now: delays in seconds, high variability



Example round-trip measurements from Ireland to California, 2009

Adaptive Retransmission

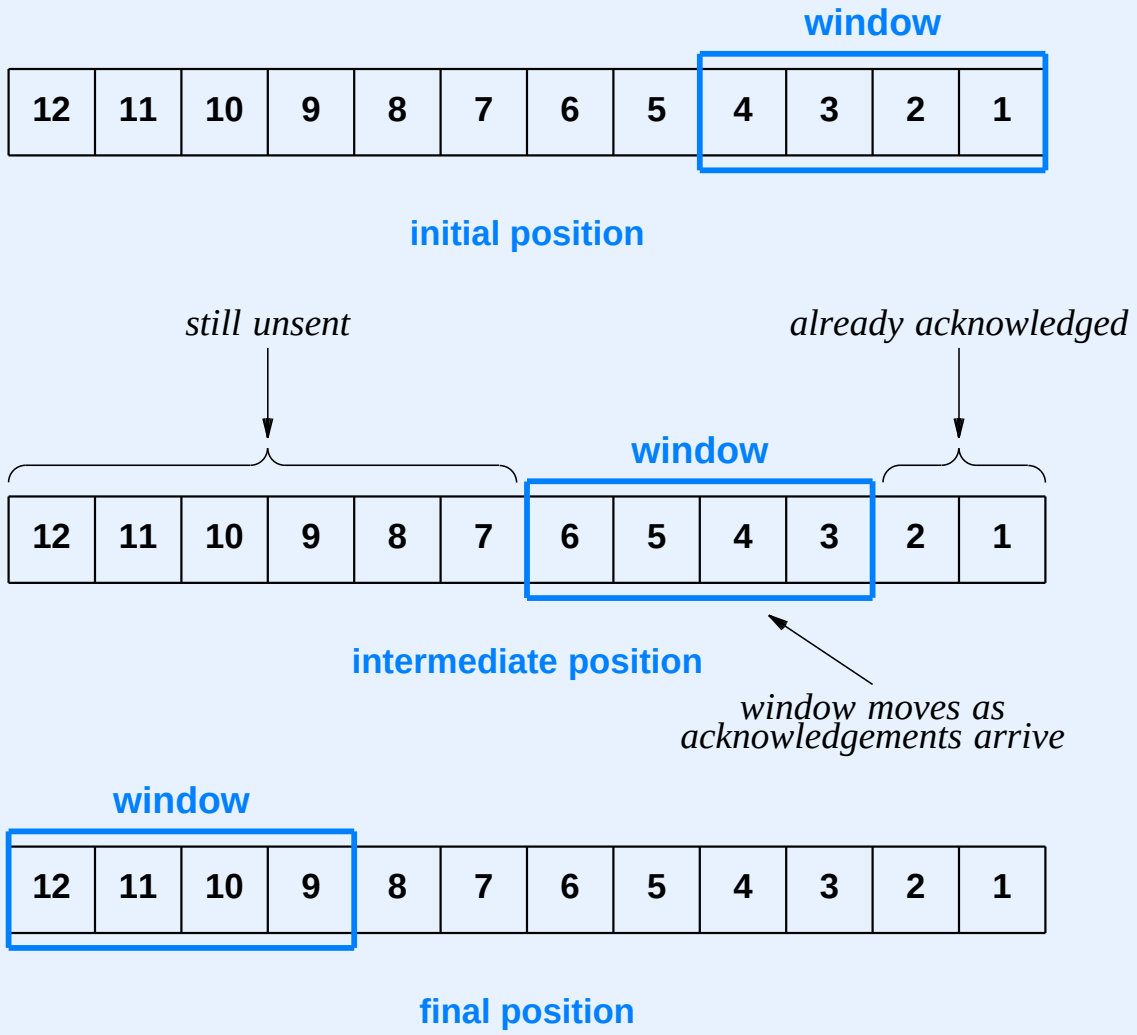
- Continually estimate round-trip time of each connection
- Set retransmission timer from round-trip estimate
- Illustration of timeout on two connections:



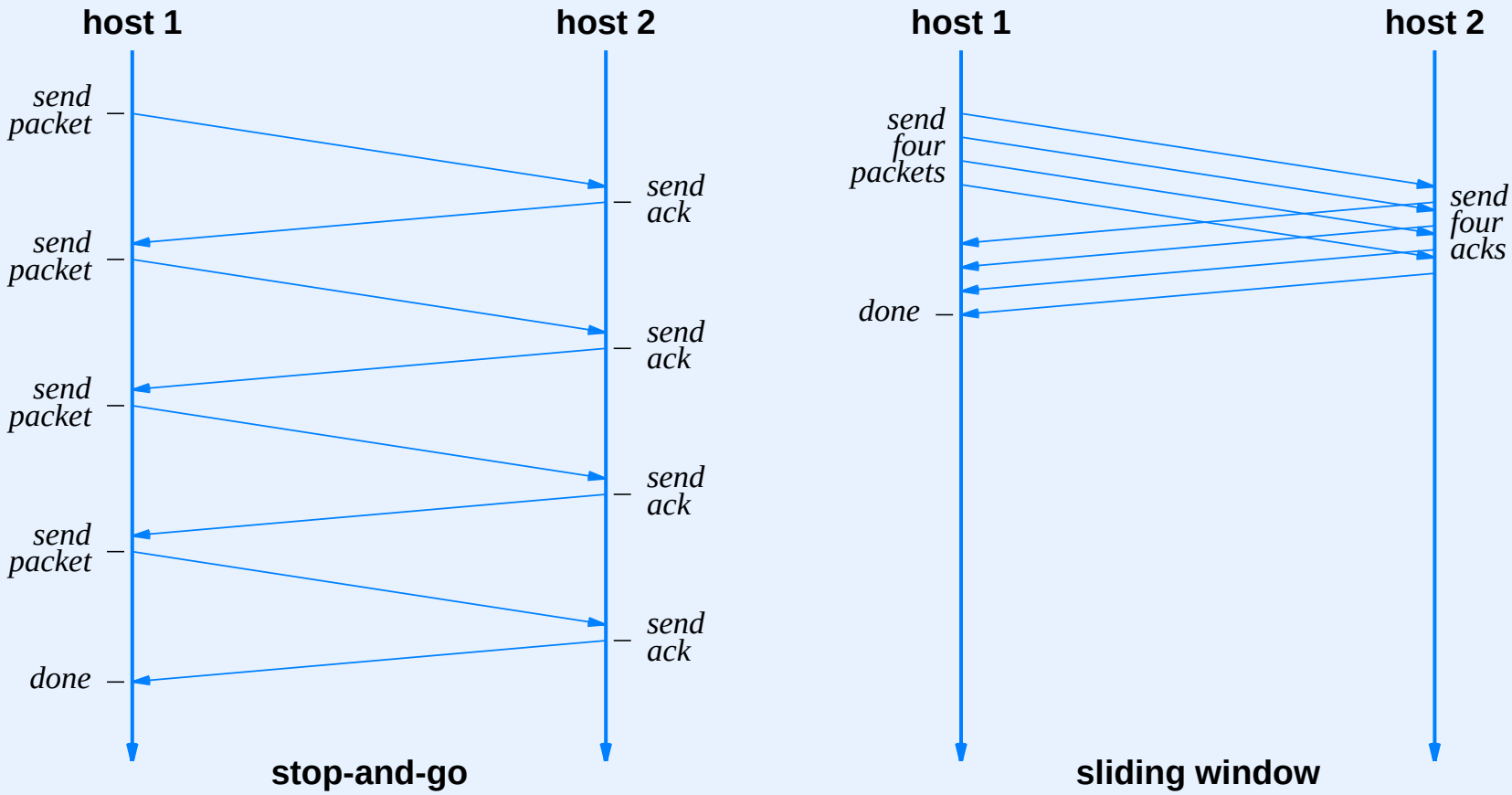
Review Of Sliding Window

- 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

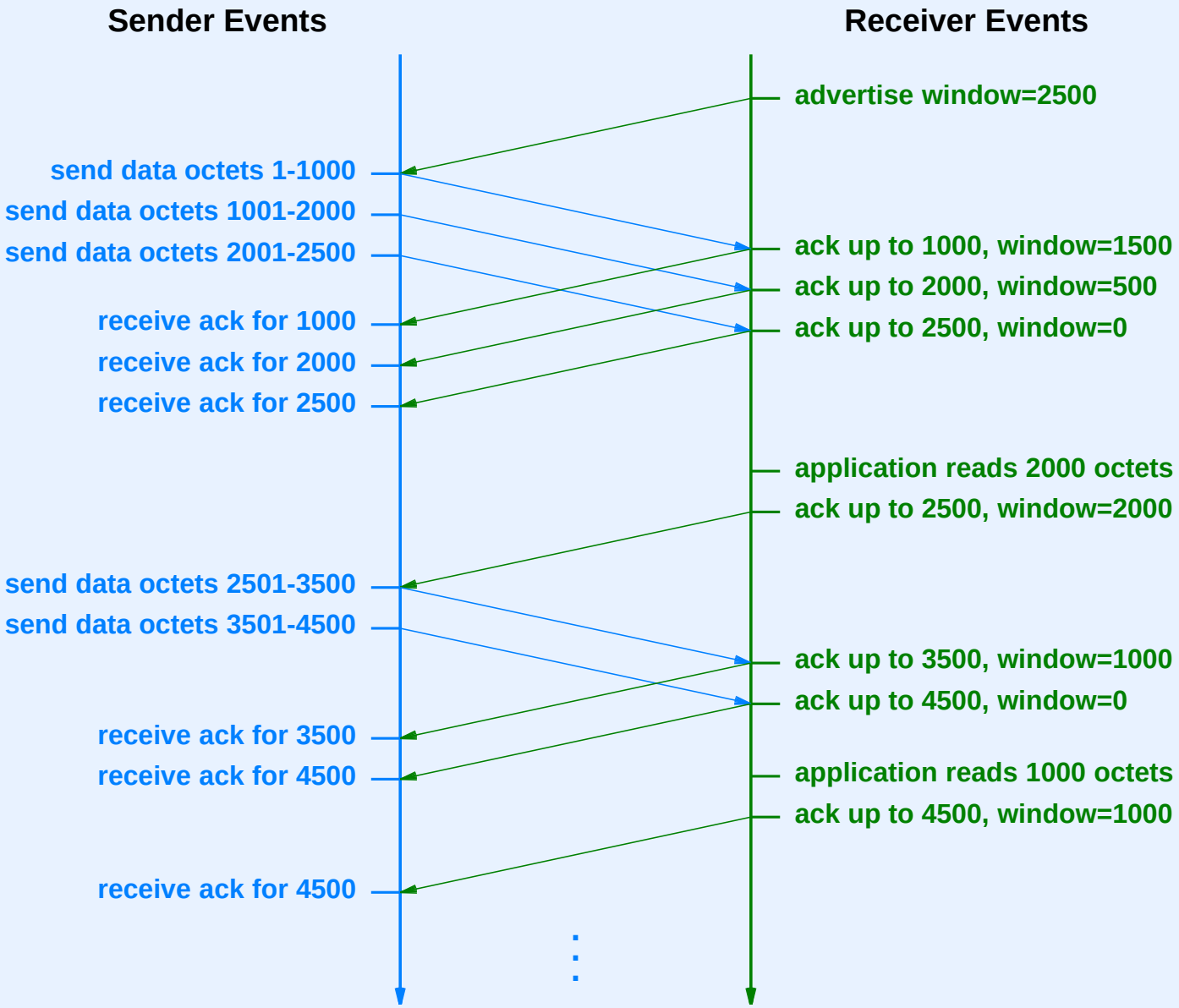


- Window size of *K* improves data rate by a factor of *K*

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

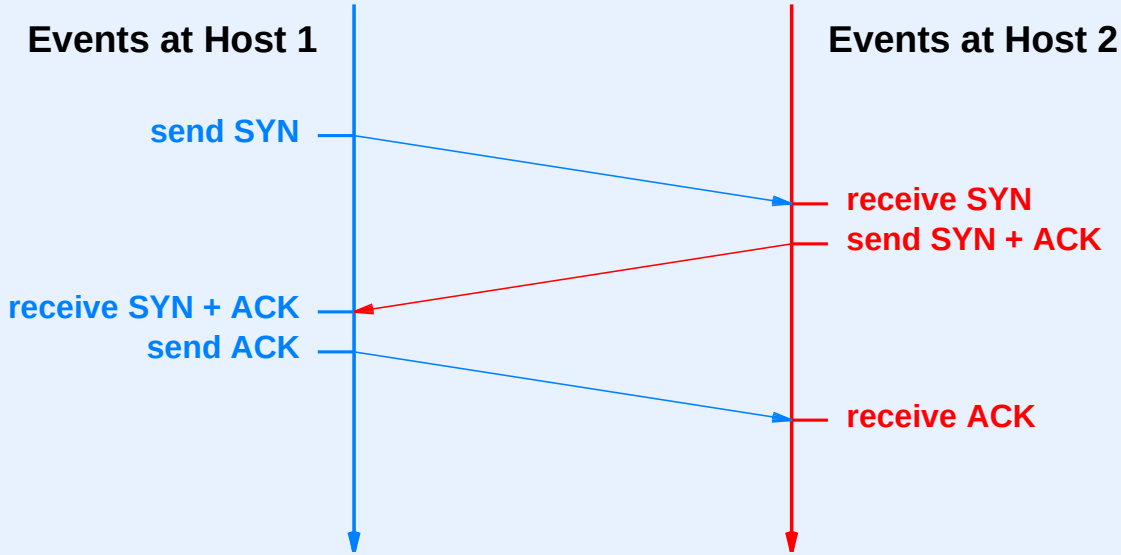
Data may be passed to a receiving application in chunks that differ from the chunks that the sending application generated.

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

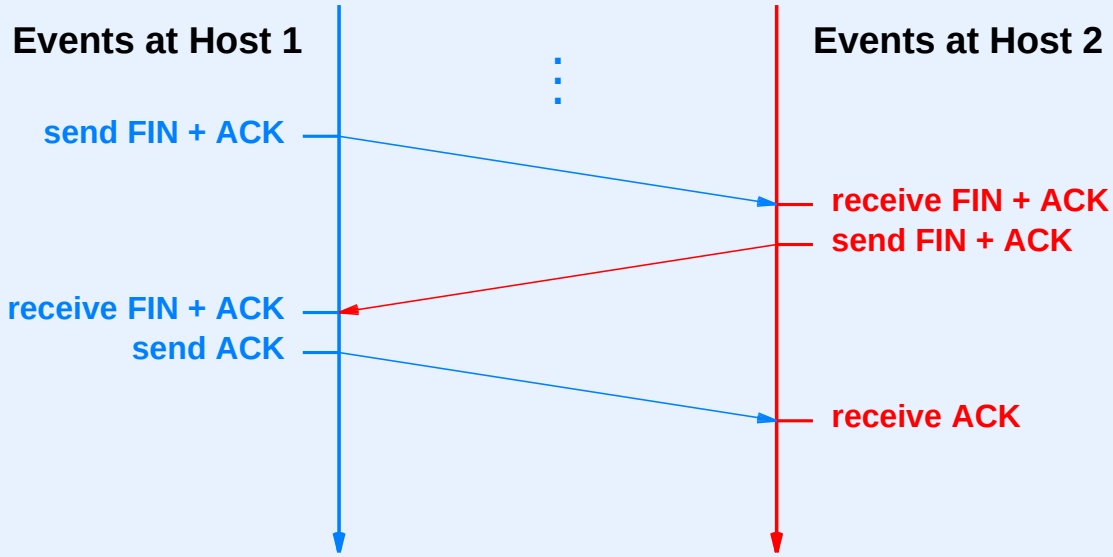
- TCP guarantees reliable connection startup that avoids *replay* problems
- Performed with 3-way handshake



- Each side chooses starting sequence number at random

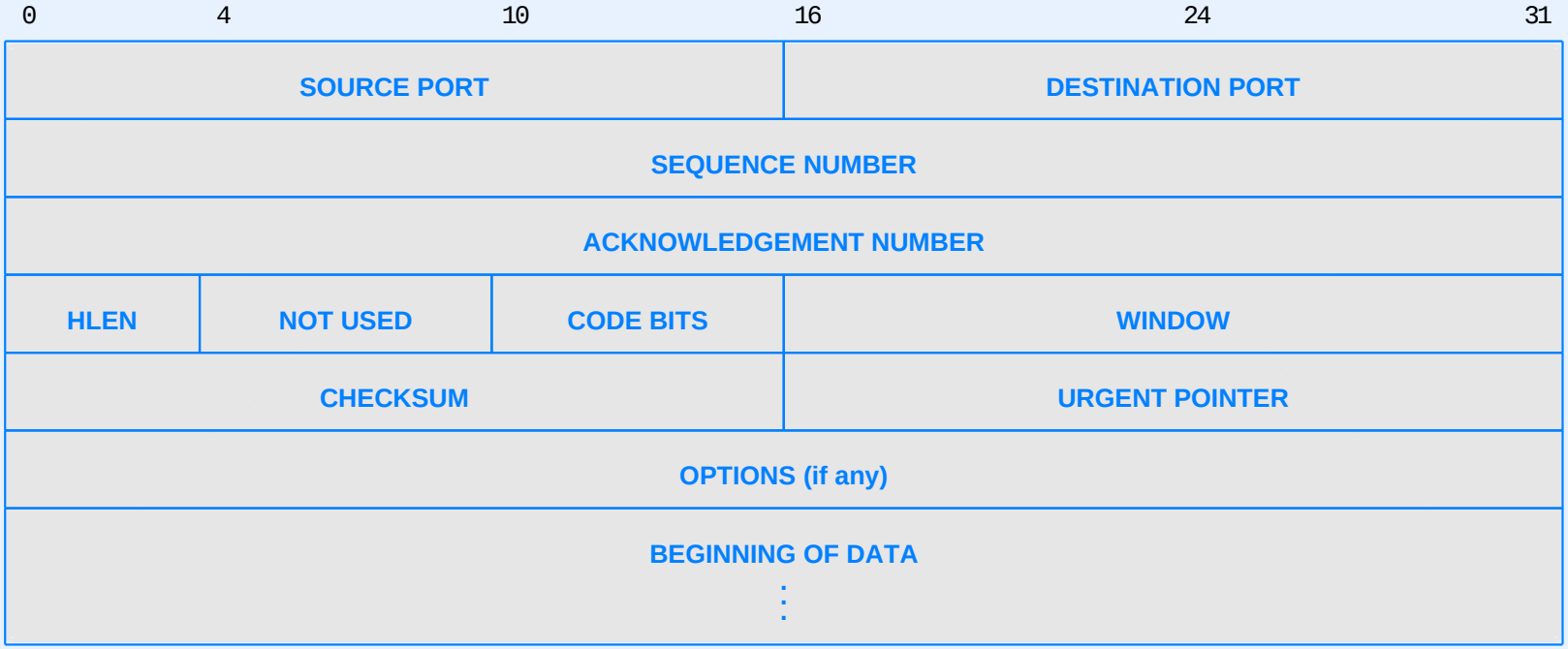
Graceful Connection Shutdown

- 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

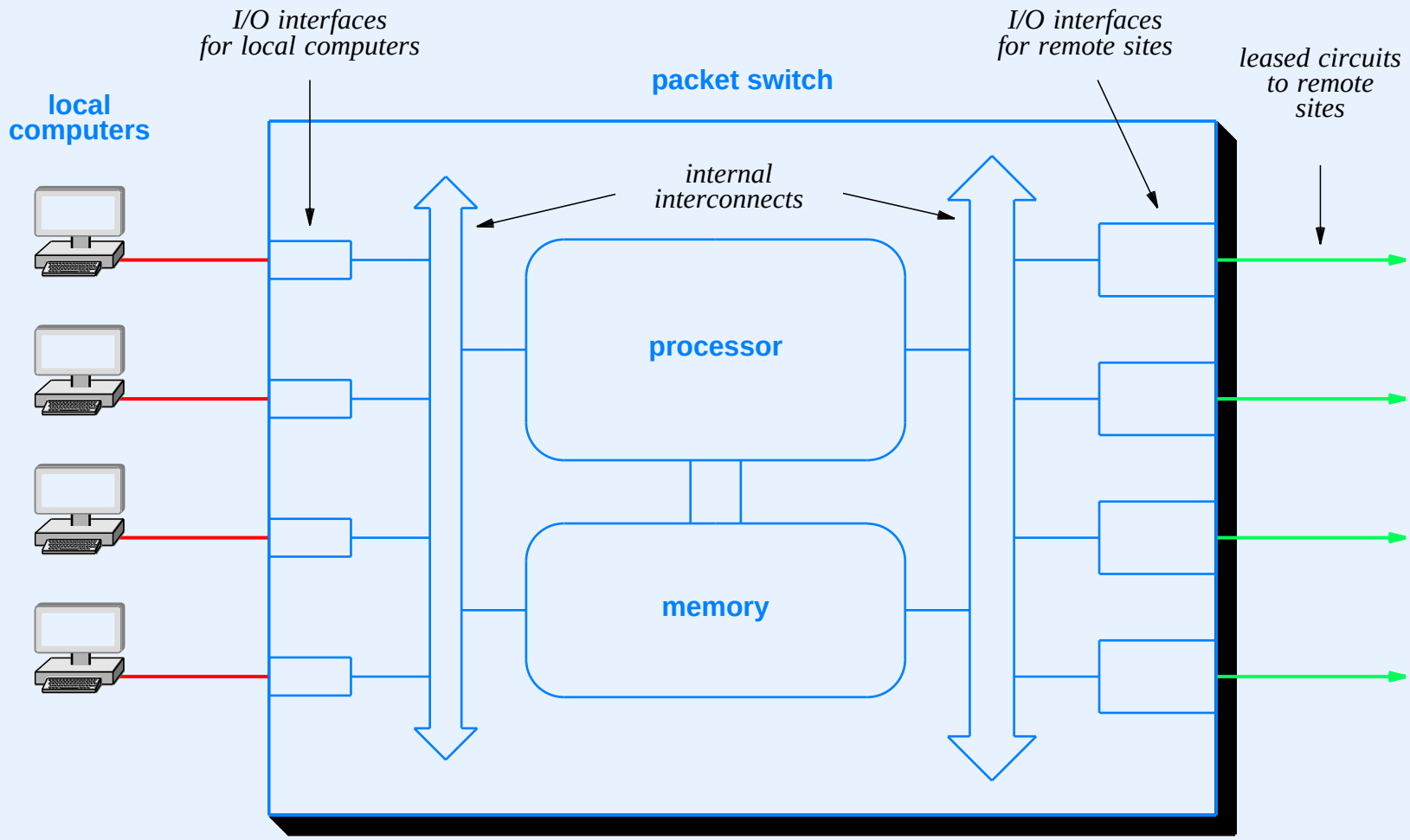
Traditional WAN Architecture

- 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

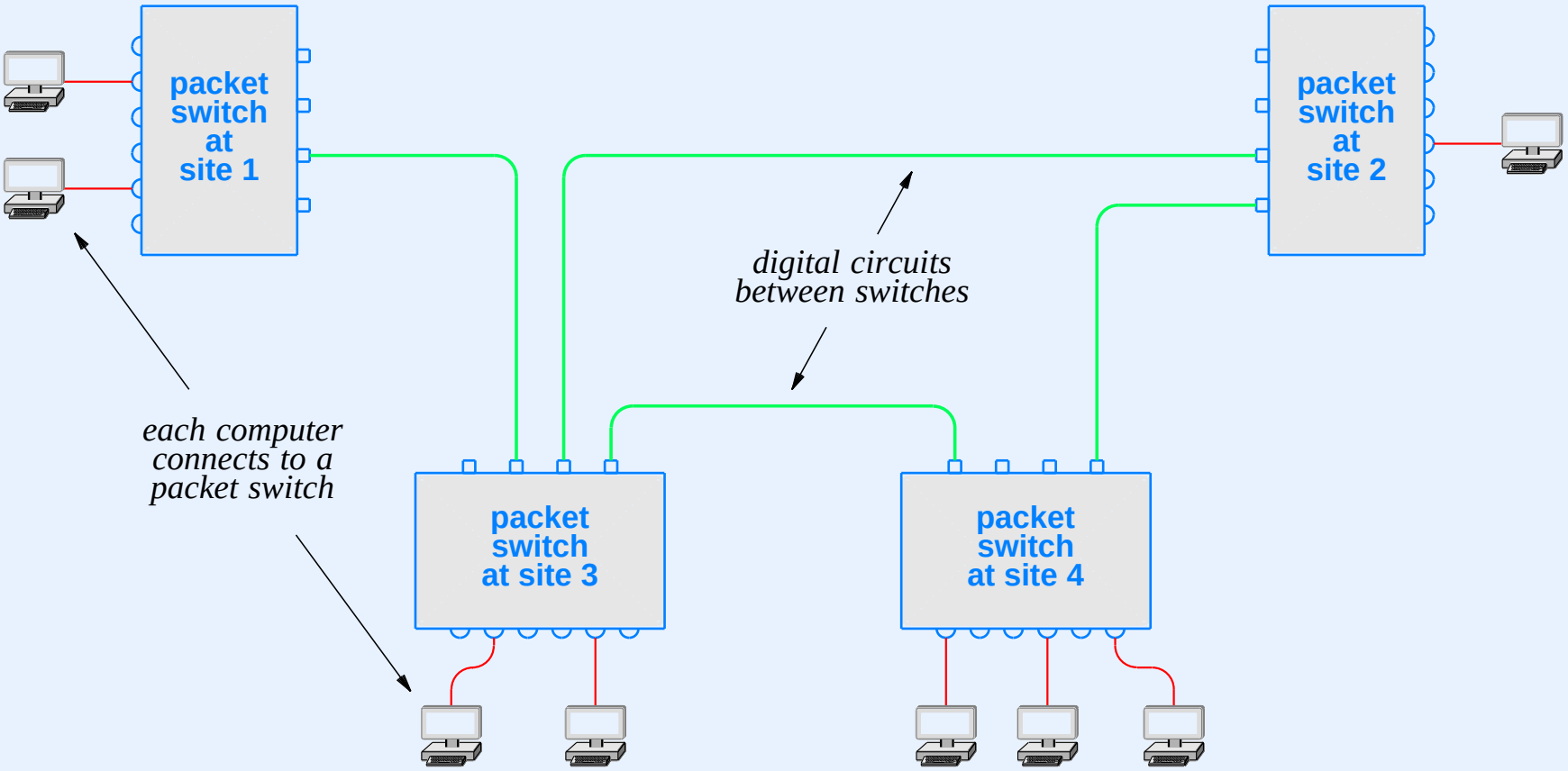
Store And Forward Paradigm

- 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

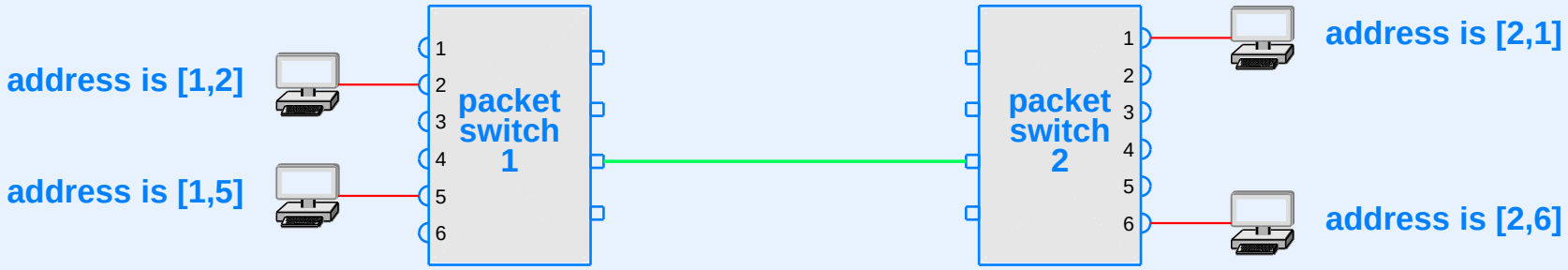
- Hierarchical model analogous to Internet addressing
- Conceptual two-level hierarchy

(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:

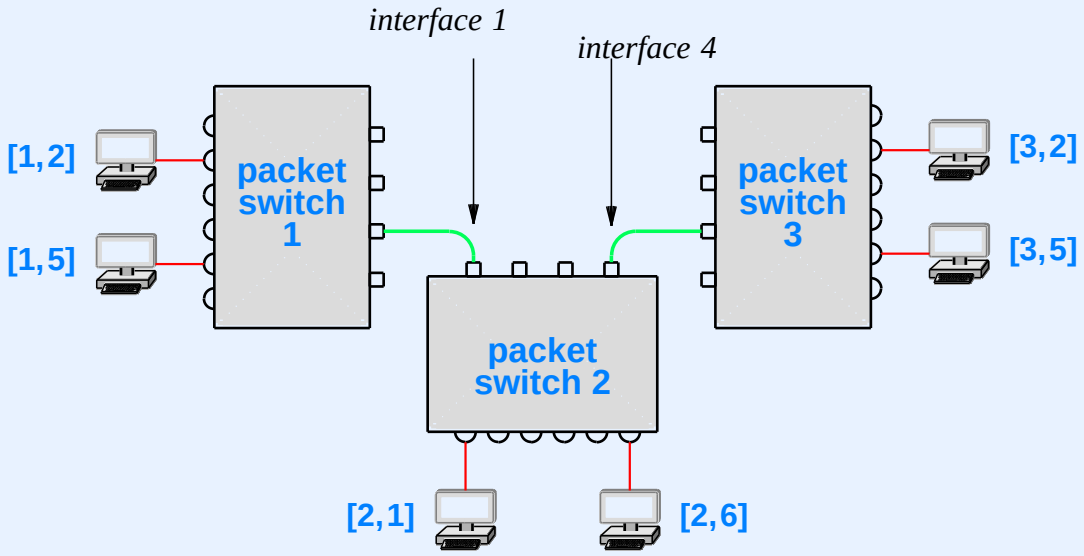
The next-hop forwarding step

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

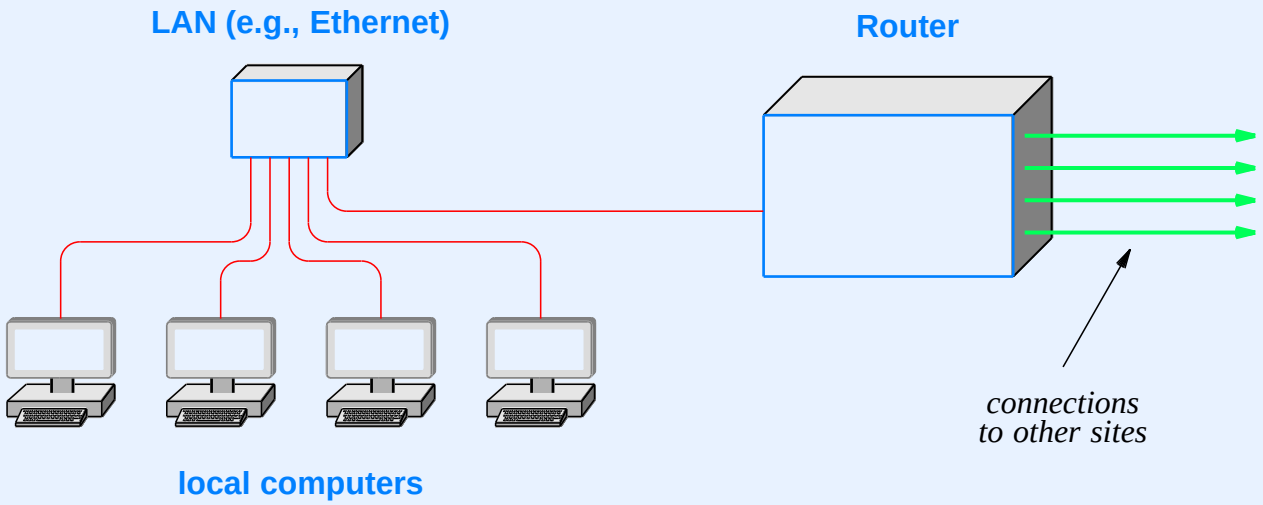
to reach	send to
switch 1	interface 1
switch 2	local delivery
switch 3	interface 4

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

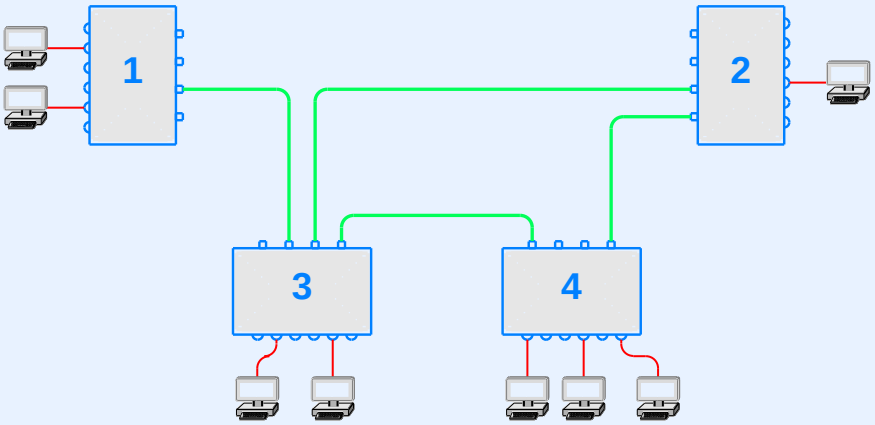
- Two basic approaches
- Static routing
  - Used in Internet hosts
  - Entries inserted when system boots and do not change
- Dynamic routing
  - Used in packet switches and IP routers
  - Initial entries inserted when system boots
  - Routing software continually monitors network, computes shortest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

Static Routing

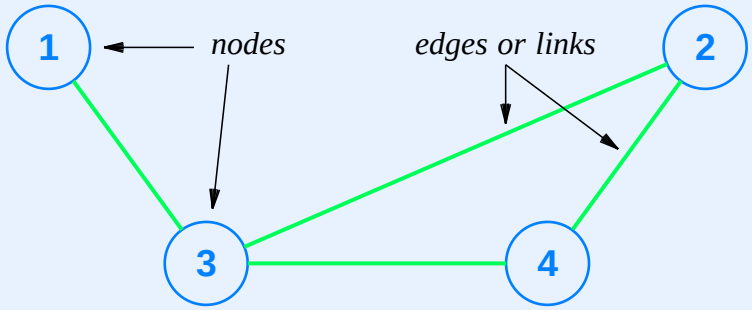
- Used in most hosts
- Only K+1 entries in forwarding table if host has K network connections
- K entries, one per network connection
  - IP prefix for the network
  - Address mask for the network
  - Interface for the network
- Final entry: default route
  - default IP router address
  - Interface for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

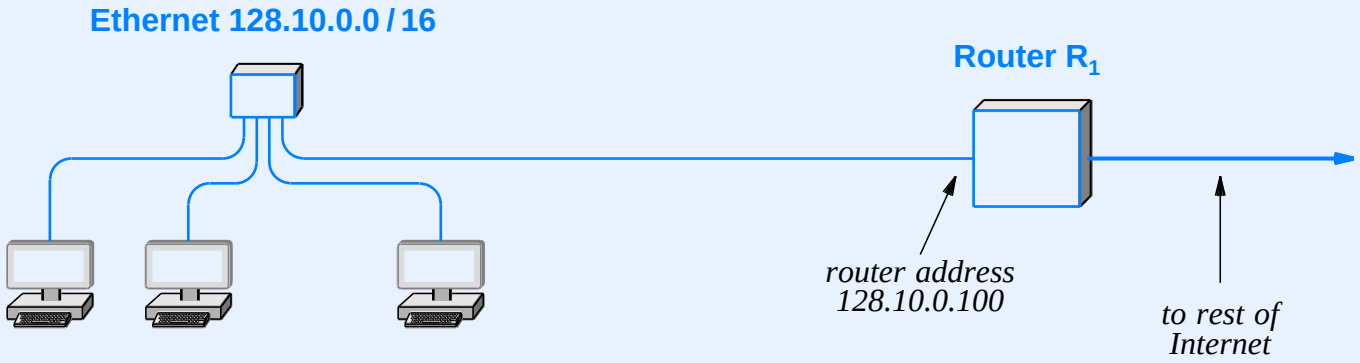
    The link between X and Y is down.

- 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

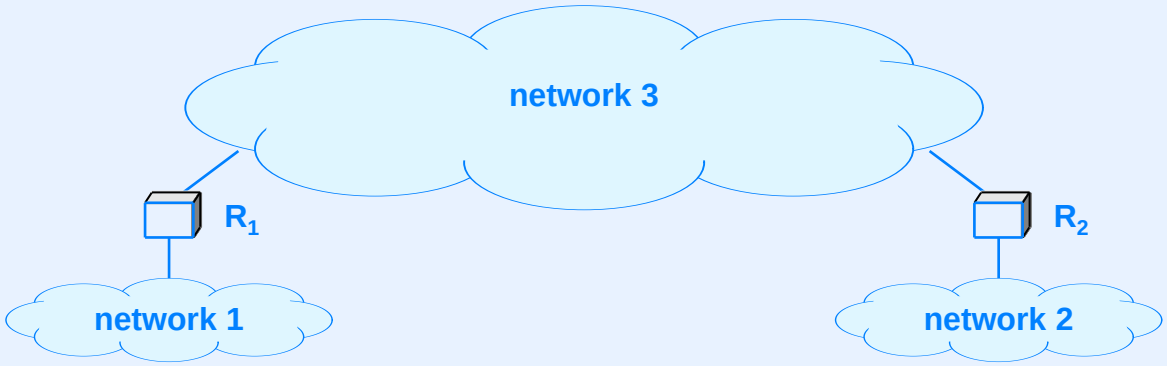
Net	Mask	Next hop
128.10.0.0	255.255.0.0	direct
default	0.0.0.0	128.10.0.100

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.

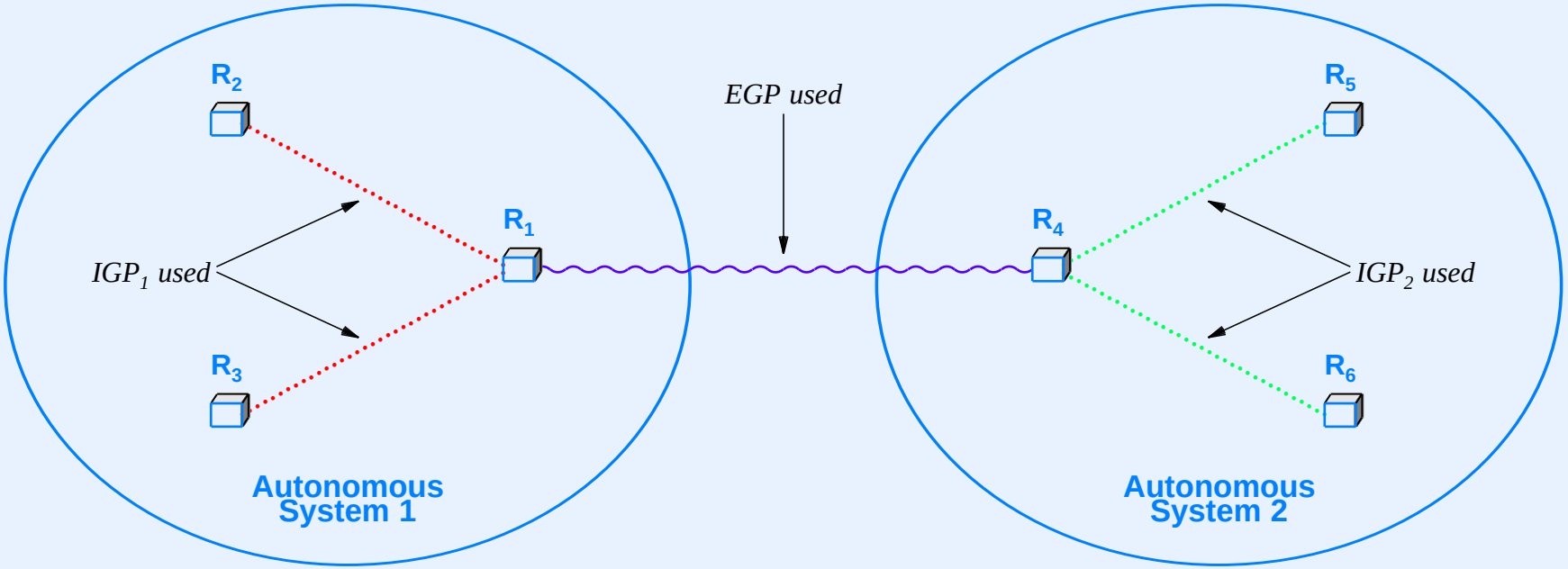
Autonomous System Concept

- 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

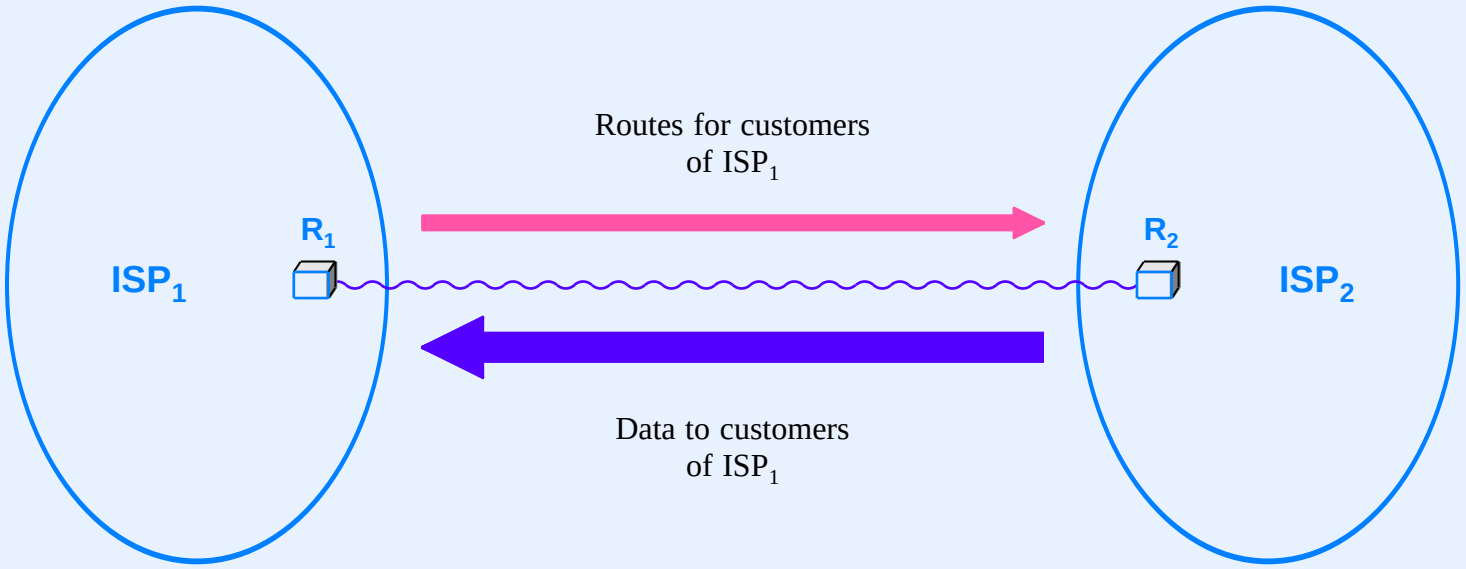
Illustration Of IGPs and EGPs



- 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

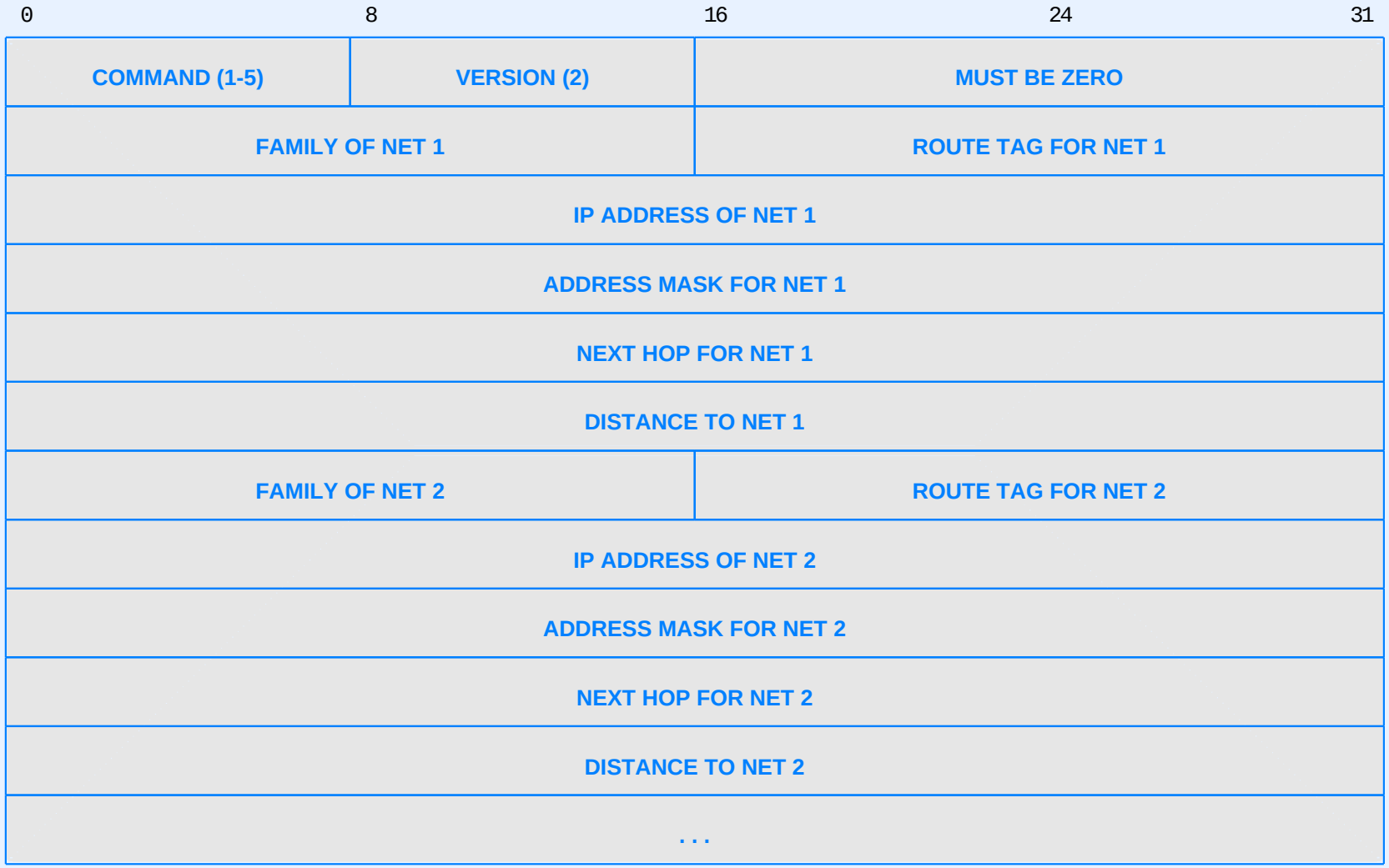
To reach network X, I send along path Z, Y, W,...

- 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

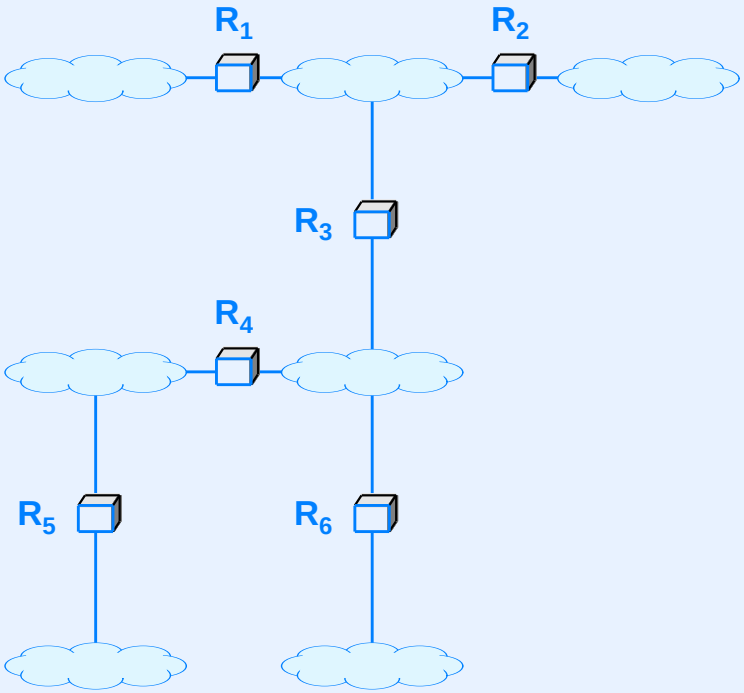


- Note: routing protocols run at application layer (layer 5)

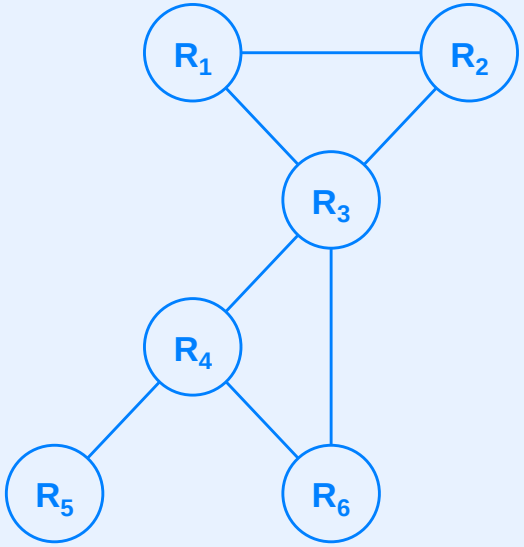
Open Shortest Path First Protocol (OSPF)

- 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

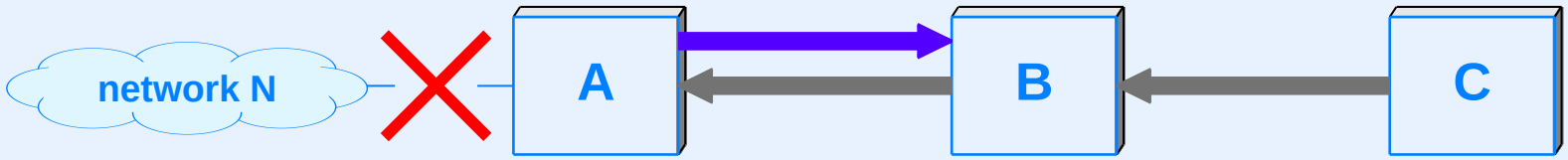
- Routing is *not* like water flowing through pipes or traffic on highways
  - Multi-path routing is difficult
  - Capacity can go unused if not along shortest path
- Fewest hops may not always be best
  - Compare two Ethernet hops and one satellite hop
- Routing around congestion is *not* straightforward, 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

MODULE VI

Other Topics

Topics

- Measuring network performance
- Quality of Service (QoS) and provisioning
- Multimedia and IP telephony
- Network security
- Traffic engineering and MPLS
- Network management (SNMP)

Measuring
Network Performance

Why Measure Network Performance?

- Optimization
- Planning (anticipating future needs)
- Assessing and understanding traffic
  - Trends in applications and network use
  - Detecting anomalous traffic patterns
- Contract (SLA) enforcement
- Bragging rights
  - IT staff in an organization
  - Marketing department in an equipment vendor

Qualitative Terminology And Marketing

- Marketing seems to love qualitative terms
  - High-speed
  - Fast
  - Powerful
  - High bandwidth
- Unfortunately
  - Qualitative terminology is vague
  - Networking technologies change rapidly

Qualitative Terminology That Faded

- A *high-speed leased line*
  - Was once defined to run at 9.6 Kbps
- The Internet's *Very high-speed Backbone Network System (VBNS)*
  - Used OC-12 links, that are no longer considered very high speed
- *Fast Ethernet*
  - Runs at 100 Mbps and is only one-tenth as fast as Gigabit Ethernet technology
- *Broadband*
  - Was once defined by the FCC to start at 128 Kbps

Quantitative Measures

- Quantifiable measurement is surprisingly difficult
- Routes and data rates can be asymmetric, making measurements in one direction differ from measurements in the other
- Inserting measurement probes can affect the performance of the system being measured
- Conditions can change rapidly

Aggregate Traffic Analysis

- Short-term variation
  - Packets tend to arrive in clumps called *bursts*
- Long-term variation
  - Diurnal and annual patterns exist
- Interestingly, data traffic is unlike voice traffic
  - Aggregate of voice telephone calls is smooth average
  - Aggregate of data traffic is bursty

Self-Similarity

Unlike voice telephone traffic, data traffic is bursty. Data traffic is said to be self-similar because aggregates of data traffic exhibit a pattern of burstiness that is statistically similar to the burstiness on a single link.

The point: data traffic is not easy to analyze

Practical Measures Of Network Performance

- Three primary quantitative measures

Measure	Description
Latency (delay)	The time required to transfer a bit across a network from one end to another
Throughput (capacity)	The amount of data that can be transferred over a network per unit time
Jitter (variability)	The changes in delay that occur and the duration of the changes

- We will see that the three are *not* completely independent

Latency Or Delay

- Time required for data to travel “across” a network
- Think of latency as the time required for a single bit to traverse a network
- Depends on
  - Physical properties of the universe (the speed of light)
  - Traffic on the network

Latency And Perceived Response Time

- Users are interested in *response time*
- Several components of delay contribute to overall response time a user perceives

Type	Explanation
Access Delay	The time needed to obtain access to a transmission medium (e.g., a cable)
Propagation Delay	The time required for a signal to travel across a transmission medium
Switching Delay	The time required to forward a packet
Queuing Delay	The time a packet spends in the memory of a switch or router waiting to be selected for transmission
Server Delay	The time required for a server to respond to a request and send a response

Bottlenecks

- Any part of a communication system can be a bottleneck that causes the most delay
- Examples
  - Access delay: acquiring a wireless channel
  - Propagation delay: a satellite transmission
  - Switching delay: deep packet inspection
  - Server delay: a news agency web site overloaded during a crisis
  - Queuing delay: packets arriving faster than they depart

Assessing Delay

- Make multiple measurements over an interval
- Report minimum, maximum, mean, and standard deviation
- Divide delay into constituent components if possible
- Choose small intervals to look for repeated patterns

Throughput

- Maximum amount of data a network can transport per unit time
- Expressed as data rate in *bits per second* (e.g., *100 megabits per second*)
- Mistakenly cited as network “speed”, but really a measure of network *capacity*
- Gives an upper-bound on performance, not a guarantee

Assessing Throughput

- Several possible measures
  - Capacity of a single communication channel
  - Capacity along a path through the network
  - Aggregate capacity of all channels
  - Capacity among pairs of ingress and egress points when used simultaneously

The Concept Of Goodput

- Invented to provide meaningful assessment of network performance
- Defined as the effective rate at which an application receives data
- Can differ from throughput for any of the following reasons
  - Application protocol overhead
  - Channel coding overhead
  - Packet header overhead
  - Receiver buffer limitations
  - Congestion avoidance mechanisms
  - Packet retransmission

Assessing Goodput

- Measure data that arrives successfully, and compute the amount of data per unit time
- Goodput measurements also include the overhead introduced by
  - Operating system
  - Transport protocol
  - Lower layer encodings and protocols
  - Application protocol and implementation
- Note: although they use the term *throughput*, most measurement tools report goodput

Jitter

- Another prominent measure of network performance
- Especially important in transmission of streaming audio and video
- Measures variation in delay
- Example
  - Suppose network has average delay D
  - If each packet takes exactly D time units to traverse the network, jitter is zero
  - If packets alternate between delays of $D+\varepsilon$ and $D-\varepsilon$, average delay remains D, but jitter increases

Key Observation

In the Internet, congestion is the single most significant cause of packet loss, high jitter, and long delays.

Handling Jitter

- Replace the Internet with an *isochronous* network
  - Approach used in the original telephone network
  - All parallel paths have exactly the same delay
- Change the Internet to reserve capacity
  - Discussed later in the module
- Keep the current Internet design and add protocols that compensate for jitter
  - Basic technique is a jitter buffer
  - Discussed later in the module

Understanding Throughput And Delay

- An analogy
  - Think of a network as a road between two locations
  - Propagation delay determines how long it takes a single car to traverse the road
  - Throughput determines how many cars can enter the road per unit time
- Observe
  - Adding a lane doubles the throughput (i.e., capacity), but leaves the delay unchanged
  - It is possible to have arbitrarily high throughput, even if the delay is long (imagine a long road with hundreds of lanes)

Understanding Throughput And Delay
(continued)

- The analogy helps us understand network measures

Propagation delay specifies the time a single bit remains in transit in a network. Throughput, which specifies how many bits can enter the network per unit time, measures network capacity.

- The key consequence is incorporated in an aphorism

You can always buy more throughput, but you cannot buy lower delay.

Delay-Throughput Product

- Specifies the maximum amount of data “in flight”

$$\textit{Bits present in a network} = D \times T$$

  where

  – D is delay measured in seconds

  – T is throughput measured in bits per second

- Specifies how many bits can be transmitted before the first bit arrives at the receiver

- Often incorrectly labeled the *delay-bandwidth* product

Delay-Throughput Terminology And Examples

- Ethernet
  - Although it has high throughput, the short delay limits the delay-throughput product
- Satellite link
  - Usually has a high delay-throughput product because delay is long and throughput is high
- Informally, we use an analogy
  - A network with a long delay is called a *long pipe*
  - A network with high throughput is called a *fat pipe*
  - A satellite is known as a *long, fat pipe*

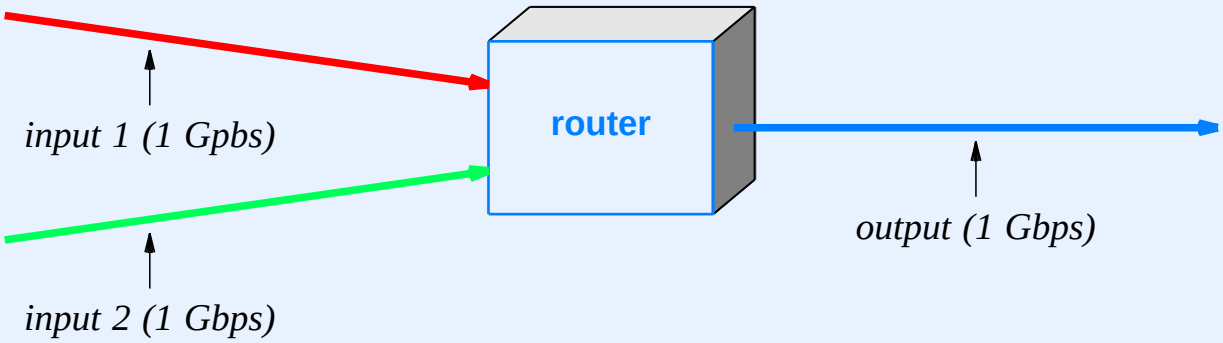
Delay, Throughput, and Utilization

Relationship Between Delay And Throughput

- In theory, delay and throughput are independent
- In practice, they are related
- Reason
  - Throughput determines rate at which traffic can pass across a communication link
  - A switch or router queues packets until they can be sent
  - If data arrives at a switch or router faster than it leaves, queue length grows, which means increased delay (congestion)

Illustration Of How Congestion Occurs

- Consider a router with three 1 Gbps connections, and assume that traffic is arriving over two connections destined for the third



- If the capacity of the red link is doubled, all links can experience more congestion, which increases delay

Utilization

- Measure of the current *load* on a network link
- Given as a percentage of capacity being used, and expressed as a real value between 0.0 and 1.0
- Example: if a link capable of 1 Gbps has traffic of 500 Mbps, link utilization is 0.5
- Because utilization changes over time, it is reported over an interval by giving
  - Peak (i.e., maximum)
  - Average (i.e., mean)

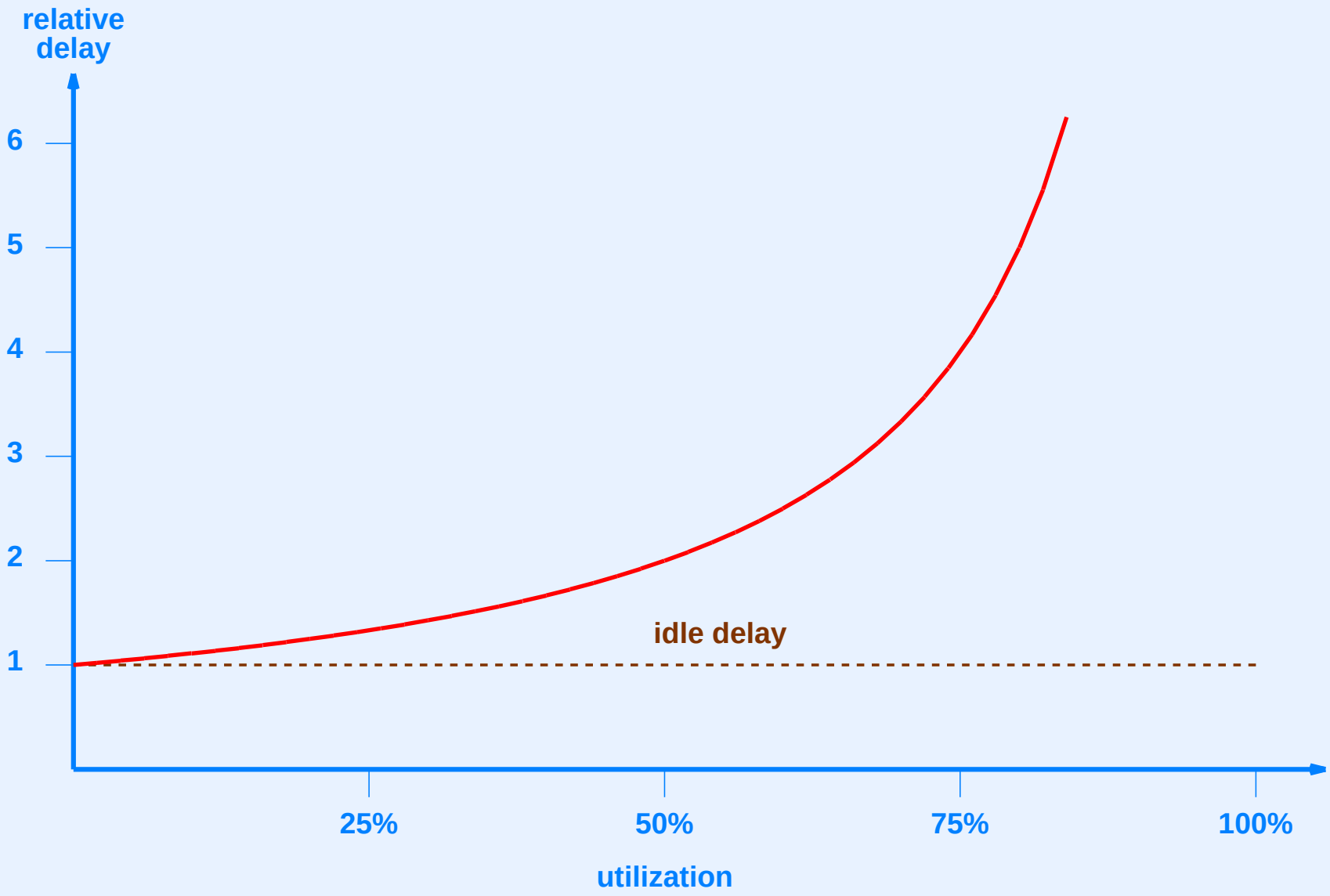
Utilization As Estimate Of Delay

- Packet traffic is *bursty*
- Key discovery: the effective queuing delay can be estimated from the utilization as follows:

$$D = \frac{D_0}{(1 - U)}$$

- Where
  - D_0 is delay when the network is idle
  - U is current utilization between *0* and *1*

Delay As A Function Of Utilization



Practical Interpretation Of Utilization

- Delay increases rapidly as utilization climbs
- When utilization reaches 50%, delay is double
- When utilization reaches 80%, delay is five times higher than average

The 50-80 Rule

- Heuristic managers follow
  - When utilization reaches 50%, plan an upgrade
  - When utilization reaches 80%, an upgrade is overdue
- Note: alternative consists of partitioning a network (e.g., separating VLANs)

Line Speed And Packets Per Second

- Networking equipment is said to operate at *line speed* if the equipment can handle a sequence of back-to-back packets
- Observe
  - Per-packet overhead is often the bottleneck in equipment
  - For a given data rate, equipment processes
    - Fewer packets per second if packets are large
    - More packets per second if packets are small
- Conclusion: line speed is meaningless without a specification of packet size

Quality of Service (QoS) and Provisioning

Quality of Service (QoS)

- Set of technologies that can be used to provide service guarantees
  - Bound on latency
  - Guarantee on throughput
  - Bound on jitter
- Marketing
  - Tries to equate QoS and “quality”
  - Implies that lack of QoS means lack of quality

QoS In The Internet

- Motivation
  - Make it possible to run applications such as streaming video with no interruptions
  - Allow service providers to charge (much) more for better service
- Three approaches have been proposed and studied
  - Priority
  - Fine-grain QoS
  - Coarse-grain QoS

Priority Approach

- Each packet assigned a *priority*, and multiplexing selects packets in priority order
- Popular among ISPs, and used by some corporations to give voice and video traffic priority
- Advantages
  - Easy to implement
  - Can assign priority to a "customer" rather than to a specific type of data
- Disadvantages
  - No quantitative guarantees
  - Can lead to *starvation*

Fine-grain QoS Approach

- Pursued by the IETF under the name *Integrated Services (IntServ)* and adopted in ATM networks
- QoS parameters negotiated for each flow (e.g., each TCP connection)
  - Maximum delay
  - Minimum throughput
  - Maximum jitter
- Difficult/impossible to implement

After many years of research and standards work, the fine-grain approach to QoS has been relegated to a few special cases.

QoS Terminology That Has Survived

- Derived from ATM

Abbreviation	Expansion	Meaning
CBR	Constant Bit Rate	Data enters the flow at a fixed rate, such as data from a digitized voice call entering at exactly 64 Kbps
VBR	Variable Bit Rate	Data enters the flow at a variable rate within specified statistical bounds
ABR	Available Bit Rate	The flow agrees to use whatever data rate is available at a given time
UBR	Unspecified Bit Rate	No bit rate is specified for the flow; the application is satisfied with best-effort service

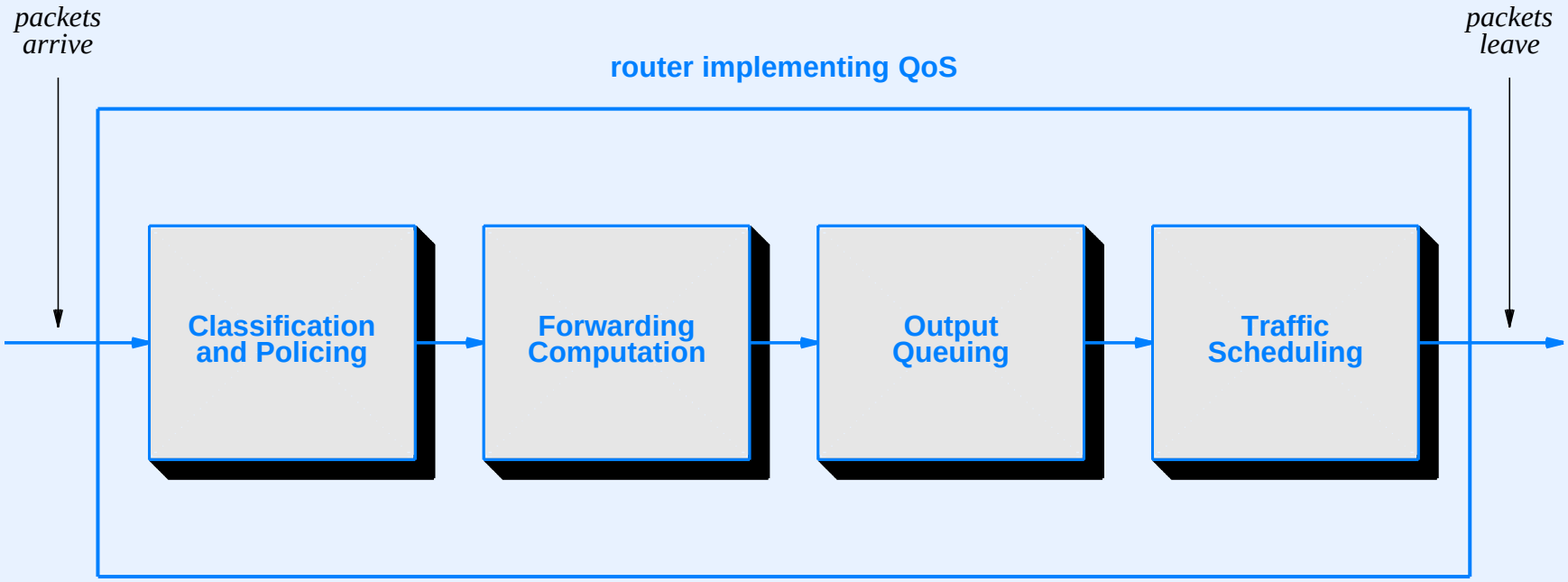
- Bounds specified statistically (e.g., average and peak throughput and burst size)

Coarse-grain QoS Approach

- Current approach approved by the IETF under the name *Differentiated Services (DiffServ)*
- Divides traffic into *classes*
- Service guaranteed for each class rather than per flow
- Easier to implement than fine-grain approach
- Usually implemented as a proportional guarantee rather than absolute quantities
- Example policy

At least 10% of the underlying network capacity is reserved for voice traffic

Steps A Router Takes To Implement QoS



- Policing enforces rules on incoming traffic
- Forwarding can select among multiple paths (router may have many output queues)
- Queuing may use *Random Early Discard* (*RED*)

Traffic Scheduling

- Algorithm used to select packets from queues
- Principal types

Algorithm	Description
Leaky Bucket	Allows a queue to send packets at a fixed rate by incrementing a packet counter periodically and using the counter to control transmission
Token Bucket	Allows a queue to send data at a fixed rate by incrementing a byte counter periodically and using the counter to control transmission
Weighted Round Robin	Selects packets from a set of queues according to a set of weights that divide the capacity into fixed percentages, assuming a uniform packet size
Deficit Round Robin	A variant of the round-robin approach that accounts for bytes sent rather than packets transferred, and allows a temporary deficit caused by a large packet

Traffic Engineering (MPLS)

Traffic Engineering

- An approach to networking that allows a manager to establish and control routes through a network and assign specific types of data to each
- Implies
  - Non-standard forwarding mechanism
  - All traffic of a given type sent along a specified path
- Most popular technology: MPLS

Multi-Protocol Label Switching (MPLS)

- Widely deployed among tier-1 ISPs
- Requires participating routers to have MPLS module
- MPLS *tunnel* created by configuring routers along a path
- Router may allow manager to assign a portion of link capacity to each tunnel
  - Term *multi-protocol* arises because an MPLS packet can contain arbitrary content

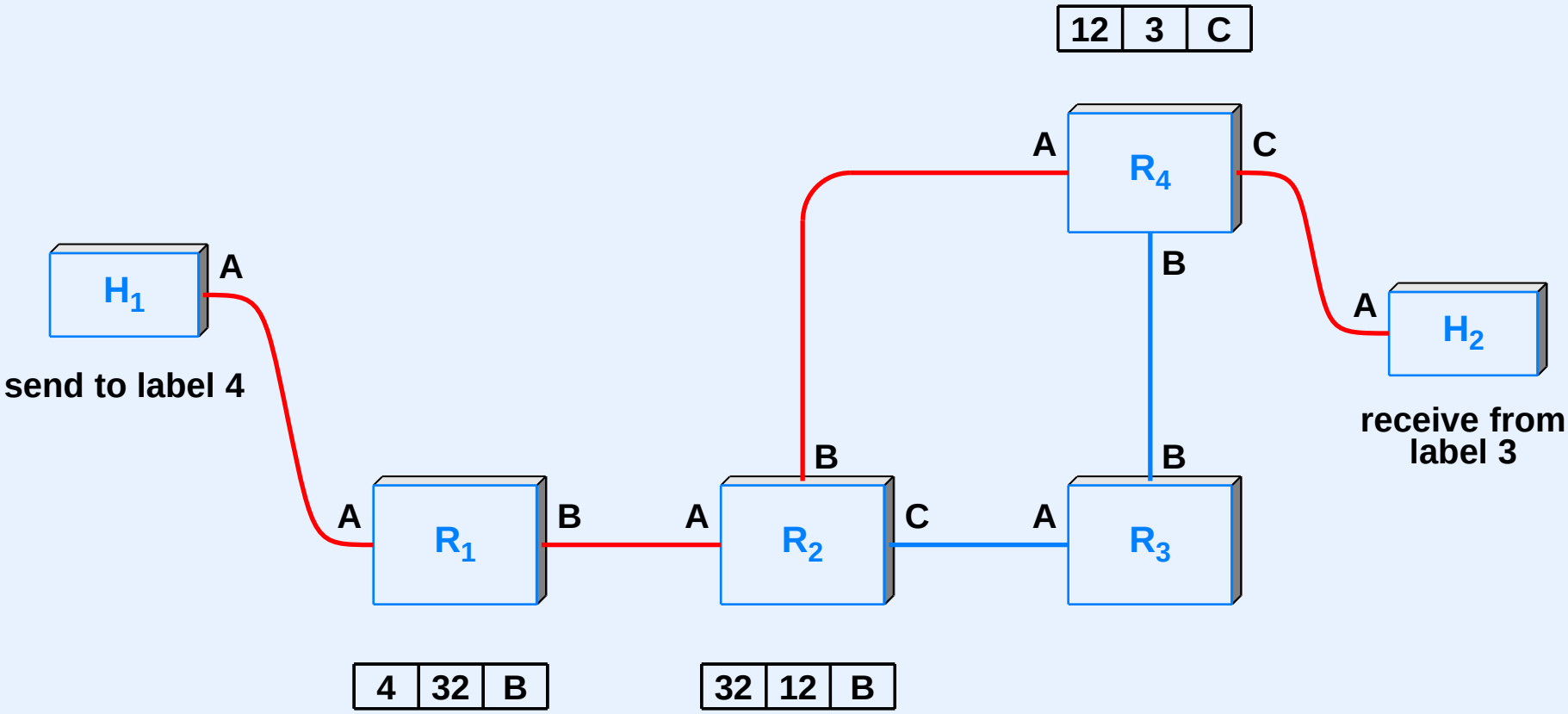
Label Rewriting

- Concept came from ATM and is used in MPLS
- Each link in path has different integer label
- Router rewrites label in MPLS datagram before forwarding to next hop
- Known as *label switching*
- Motivation: avoid global coordination and allow local assignment of labels
- Disadvantages:
  - No general protocol available to configure an MPLS path
  - Can be difficult to debug

How MPLS Works

- Datagram encapsulated in MPLS header by router at the start of a tunnel
- MPLS datagram tagged with *label* of path over which it must pass
- Each router along the path
  - Uses label to make forwarding decision
  - Replaces label with value used on next hop
- MPLS encapsulation removed when datagram reaches end of tunnel

Illustration Of Label Rewriting



- Labels along the path are: 4, 32, 12, 3

Multimedia

A Few Definitions

- *Multimedia* combines two or more forms of information, such as
  - Photos and music
  - Audio and video
- *Real-time* refers to information that must be presented in a predetermined timed sequence, such as
  - Audio
  - Video
- An *individual source* provides one particular sequence of real-time information

A Few Definitions
(continued)

- *Playback* refers to the output of real-time information for a user (e.g., video display or audio output)
- *Sample rate* refers to the rate at which real-time information has been converted to digital form (e.g., audio sampled 8000 times per second)
- *Synchronization* refers to the coordination of playback information from multiple sources (e.g., a movie requires synchronization between audio and video)

Real-Time Sample Rates

- Each source of real-time data can choose a *sample rate* and encoding
- Examples
  - A video stream might contain 30 frames per second, with an encoding that uses compression
  - An audio stream might contain 8000 audio samples per second using a PCM encoding
- Important concept

Because each source of real-time information can choose a sample rate, playback and synchronization must know the sample rate and encoding that was selected.

Transfer Of Streamed Real-Time Data

- Source
  - Samples information at regular intervals
  - Generates data continuously
  - Prepares data for transmission
- Ideal transmission channel
  - Accepts input at rate source produces
  - Delivers output at same rate as input

Quantitative Network Performance Needed For Real-Time Streaming

- QoS type: Constant Bit Rate (CBR)
- Throughput sufficient to accommodate sender's data rate (known in advance)
- Latency within a specified bound, usually 200 msec
- Jitter of zero or near-zero

Buffering

- Especially important in a packet transmission system
- Combines multiple samples into a single transmission
- Advantage
  - Increases transmission efficiency
- Disadvantage
  - Introduces delay

Buffering Example

- Consider PCM audio
- One eight-bit audio sample taken every 125 μseconds
- Ethernet has 1500 octet payload
- Waiting to fill an entire frame takes

$$125 \times 10^{-6}\ \textit{seconds/byte}\ \times\ 1500\ \textit{bytes}\ =\ 0.188\ \textit{seconds}$$

- Filling a packet incurs delay at the source

Buffering Compromise

- Choose buffer size according to application
- Example: send 128 audio samples in each packet
- Tradeoffs
  - Packet size is larger than one sample per packet, but generates more packets than absolutely necessary
  - Header overhead is a smaller percentage of total bits than with one sample per packet, but a greater percentage than for larger packets
  - Latency is better than with many samples per packet, but not as good as with one sample per packet

Jitter Buffers

Streaming Of Real-Time Data Across The Internet

- Must handle
  - Lost packets
  - Duplicated packets
  - Packets delivered out of order
  - Variance in delay (jitter)
- Key facts
  - Conventional retransmission is useless
  - Jitter is unavoidable

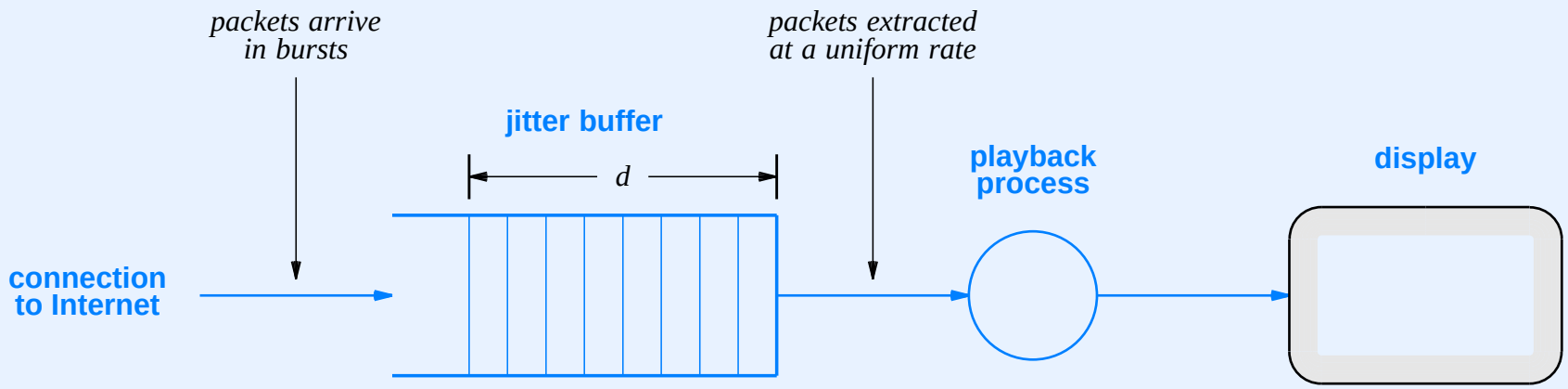
Two Useful Techniques

- *Timestamps*
  - Provided by sender
  - Assigned to each piece of data
  - Allow receiver to know when data should be played
  - Use relative values to avoid need for clock synchronization
- *Jitter buffer*
  - Used by receiver
  - Accommodates small variance in delay

Jitter Buffer

- Used by receiver to assemble incoming real-time data
- Timestamp on an item determines where item is placed in the playback sequence
- General principle: ensure information will be available in time to play without delay
- Trick: to compensate for maximum jitter of *d*, delay playback for *d* time units
- Result: jitter buffer holds just enough data so playback can proceed uninterrupted

Illustration Of A Jitter Buffer



- During normal operation, playback can continue for *d* time units while waiting for delayed packets

Real-Time Transport Protocol (RTP)

- Widely used for voice and video
- Despite the name, not really a *transport* protocol
- Does *not* contain a jitter buffer and does *not* control playback
- Provides three basic mechanisms
  - *Sequence number* on each packet that allows a receiver to handle loss and out-of-order delivery
  - *Timestamp* used for playback of the data
  - Series of *source identifiers* that tell a receiver the source(s) of the data

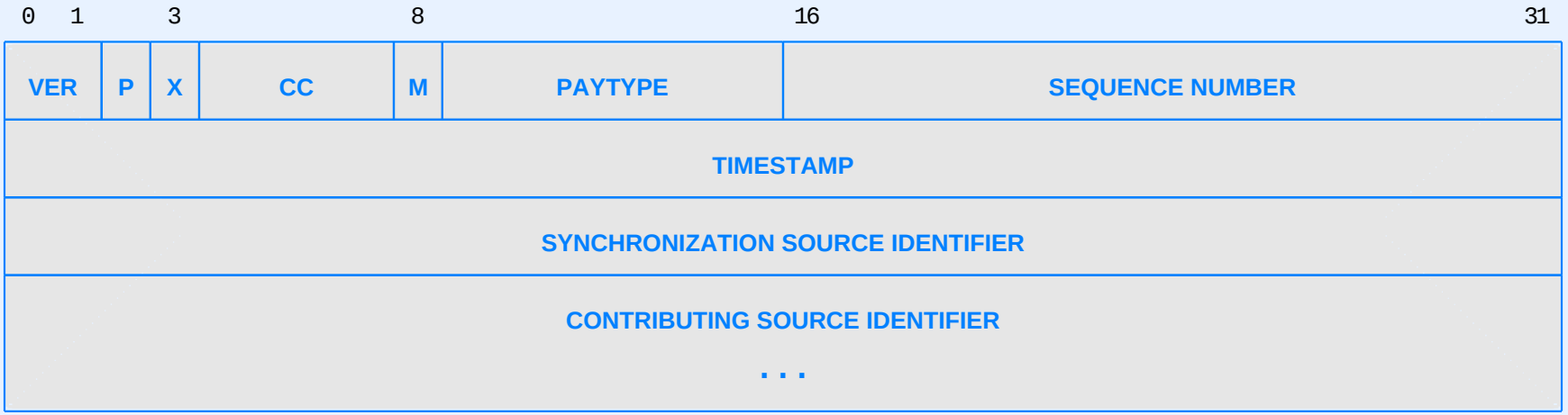
RTP Details

- Allows sender and receiver to choose sample rate and encoding
- Specifies a header for each message transferred
- Uses UDP for transport
- Separates timestamp from packet sequence number
- Includes a *marker* bit that allows some frames to be marked
- Companion protocol allows receivers to inform sender about transfer

Motivation For RTP Design

- Marking
  - Permits differential encoding with a full frame followed by incremental changes
  - Example use: video I-frame followed by B-frames
- Separation of timestamp and packet sequence
  - Means timestamps do not need to be linearly related to packets
  - Allows compression schemes that vary the rate at which data is sent

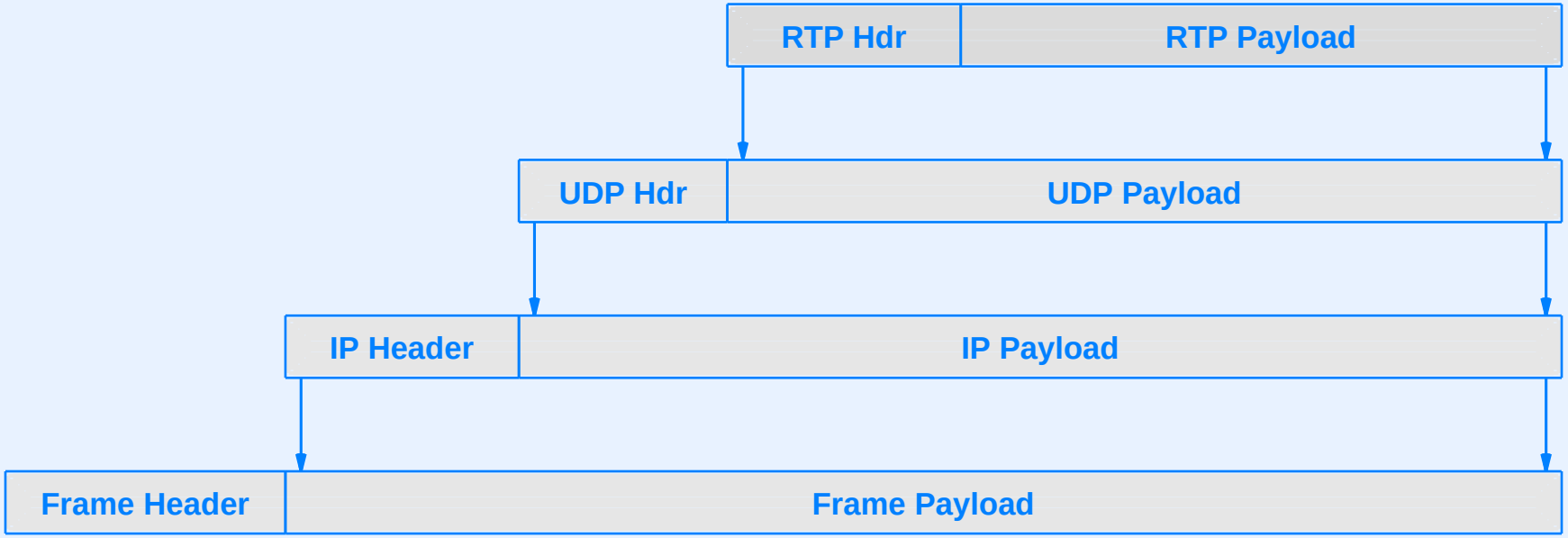
RTP Header Format



- TIMESTAMP is interpreted by sender and receiver
- PAYTYPE specifies the payload type
- Initial SEQUENCE NUMBER chosen at random
- CONTRIBUTING SOURCE IDENTIFIERS allow sender to mix streams from multiple sources

RTP Encapsulation

- Three levels of encapsulation



- Use of UDP permits sending one multicast instead of multiple unicast copies

IP Telephony (VoIP)

IP Telephony

- Known as *Voice over IP (VoIP)*
- Two groups have created standards
  - International Telecommunications Union (ITU)
  - Internet Engineering Task Force (IETF)
- Standards agree on two basics
  - Audio encoded using Pulse Code Modulation (PCM)
  - RTP used to transfer digitized audio
- Standards disagree on
  - *Signaling*
  - Public Switched Telephone Network (PSTN) interaction

Signaling

- Telco term for the process of establishing and terminating a call
- Includes
  - Mapping a phone number to a location
  - Finding a route to the called party
  - Recording information used for accounting and billing
  - Handling functions such as *call forwarding*
- Standard call management facility for the traditional telephone system is known as *Signaling System 7 (SS7)*

IETF Approach

- Known as *Session Initiation Protocol (SIP)*
- Domain Name System used to map a telephone number to an IP address
- SIP signaling system
  - *User agent* makes or terminates calls (e.g., an IP phone)
  - *Location server* consults a database of users, services to which they subscribe, and preferences
  - *Proxy server* forwards requests and optimizes routing
  - *Redirect server* handles tasks such as call forwarding and 800-number connections
  - *Registrar server* allows users to register for service

ITU Approach

- Standard is *H.323*
- Differs substantially from terminology used by SIP
- *Terminal* provides IP telephone functions and may also include facilities for video and data transmission
- *Gatekeeper* provides location and signaling functions, and establishes connections to the PSTN
- *Gateway* interconnects the IP phone system and PSTN, and handles both signaling and media translation
- *Multipoint Control Unit (MCU)* provides services such as multipoint conferencing

International Softswitch Consortium (ISC)

- Formed by vendors to consolidate terminology from multiple standards and create a single conceptual model
- Defined a list of 10 functions that are sufficient to explain all others
- Invented new terms for each function

Summary Of VoIP Protocols And Layering

Layer	Call Process.	User multimedia	User Data	Support	Routing	Signal Transport
5	H.323 Megaco MGCP SIP	RTP	T.120	RTCP RTSP NTP SDP	ENUM TRIP	SIGTRAN
4	TCP UDP	UDP	TCP	TCP UDP		SCTP
3	IP, RSVP, and IGMP					

- Each protocol can be complex
- H.323 is an umbrella

H.323

- Large set of protocols collected together
- Provides voice, video, and data transfer
- Summary of major protocols

<table>
<tr><th>Layer</th><th>Signaling</th><th>Registration</th><th>Audio</th><th>Video</th><th>Data</th><th>Security</th></tr>
<tr><td rowspan="2">5</td><td rowspan="2">H.225.0-Q.931
H.250-Annex G
H.245
H.250</td><td rowspan="2">H.225.9-RAS</td><td>G.711
H.263
G.722
G.723
G.728</td><td>H.261
H.323</td><td rowspan="2">T.120</td><td rowspan="2">H.235</td></tr>
<tr><td colspan="2">RTP, RTCP</td></tr>
<tr><td>4</td><td>TCP, UDP</td><td colspan="3">UDP</td><td>TCP</td><td>TCP, UDP</td></tr>
<tr><td>3</td><td colspan="5">IP, RSVP, and IGMP</td></tr>
</table>

Telephone Number Mapping And Routing

- Two standards proposed by IETF
  - *TRIP* relies on location servers to exchange information
  - *ENUM (E.164 NUMbers)* uses arpa top-level domain in the Domain Name System
- ENUM example
  - Phone number is 1-800-555-1234
  - Domain name is constructed as the string

4.3.2.1.5.5.5.0.0.8.1.e164.arpa

Network Security

Network Security

- Large subject with many aspects
- Major problems include

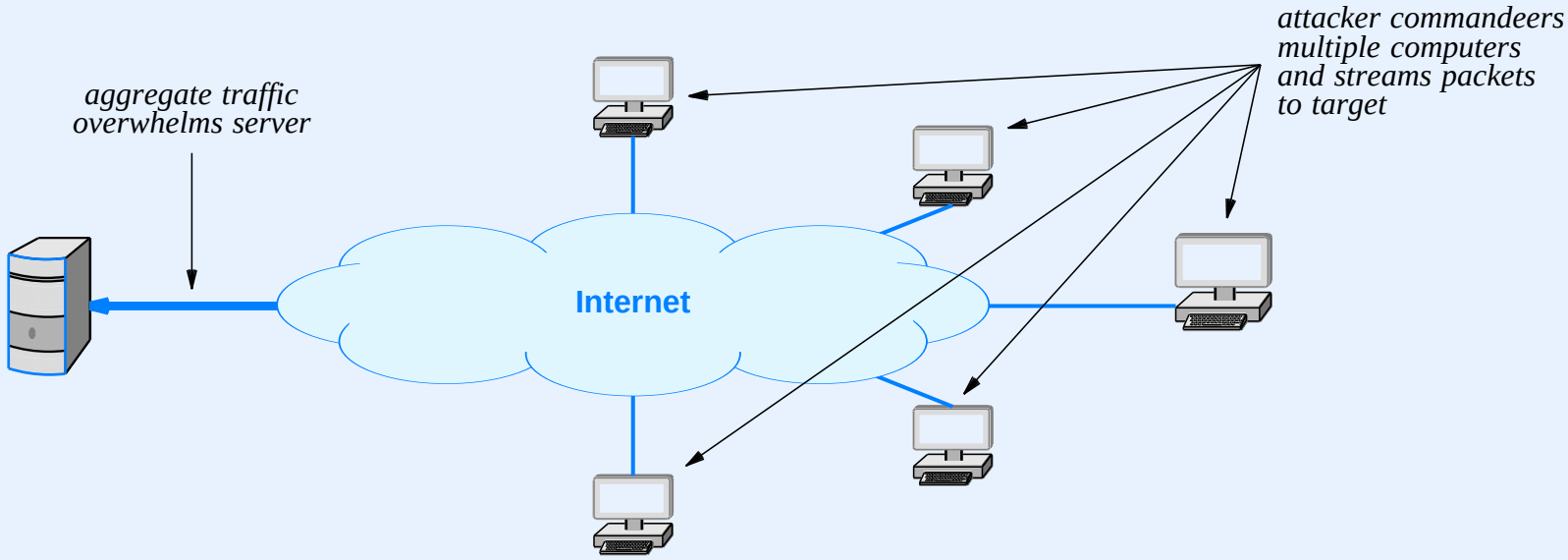
Problem	Description
Phishing	Masquerading as a well-known site such as a bank to obtain a user's personal information, typically an account number and access code
Misrepresentation	Making false or exaggerated claims about goods or services, or delivering fake or inferior products
Scams	Various forms of trickery intended to deceive naive users into investing money or abetting a crime
Denial of Service	Intentionally blocking a particular Internet site to prevent or hinder business activities and commerce
Loss of Control	An intruder gains control of a computer system and uses the system to perpetrate a crime
Loss of Data	Loss of intellectual property or other valuable proprietary business information

Examples Of Techniques Attackers Use

Technique	Description
Wiretapping	Making a copy of packets
Replay	Sending packets captured from a previous session
Buffer Overflow	Overflowing a memory buffer to overwrite values
Address Spoofing	Faking the IP source address in a packet
Name Spoofing	Using a misspelling of a well-known name
DoS and DDoS	Flooding a site with packets to prevent access
SYN Flood	Sending a stream of random TCP SYN segments
Key Breaking	Guessing a decryption key or password
Port Scanning	Probing ports to find a vulnerable application
Packet Interception	Removing a packet from the Internet

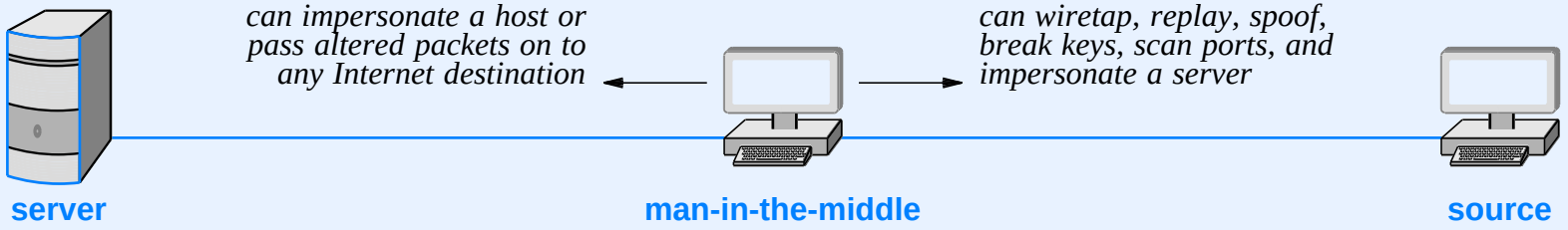
Indirect Attacks

- Attacker commandeers computers of unwitting users
- *Bots* running on commandeered computers launch attack
- Example: *Distributed Denial of Service (DDoS)*



Packet Interception

- Extreme vulnerability
- Can be exploited for many attacks
- Permits *man-in-the-middle* attacks
- Example attacks



Security Policy

- No absolutely secure network exists
- Before security mechanisms are meaningful, organization must define a *security policy*
  - Data integrity (no unauthorized change)
  - Data availability (no disruption of service)
  - Data confidentiality (no unauthorized access)
  - Privacy (no disclosure of sender's identity)
  - Accountability (record keeping and audit trail)
  - Authorization (who is permitted to access information)

Authorization And Authentication

- Authorization is intertwined with authentication
  - Authorization meaningless without authentication
  - Must know identity of a requester
- There is no point in defining a security policy that cannot be enforced

Enforcement Mechanisms

Technique	Purpose
Hashing	Data integrity
Encryption	Confidentiality
Digital Signatures	Message authentication
Digital Certificates	Sender authentication
Firewalls	Site integrity
Intrusion Detection Systems	Site integrity
Deep Packet Inspection & Content Scanning	Site integrity
Virtual Private Networks (VPNs)	Data confidentiality and trusted access

Hash

- Used to guarantee message arrives with no
  - Changes
  - Additions
- Sender and receiver share a *key*
- Sender uses key to compute a small value, H, called a
  - *Message Authentication Code (MAC)*
  - *Hash* of the message
- Sender transmits H with the message
- Receiver uses same key to compute hash of received message and compares to H

Encryption

- Fundamental security technique
- Predates computers and computer networks
- Extensive mathematical analysis
- Definitions
  - *Plaintext*: original, unencrypted message
  - *Cyphertext*: message after encryption
  - *Encryption key*: short bit string used for encryption
  - *Decryption key*: short bit string used for decryption
- Note: in some schemes, the encryption and decryption keys differ; in others, they are identical

Mathematics Of Encryption

- Encryption and decryption viewed as functions
- Encrypt takes key, K_1, and plaintext message, M, as arguments and produces cyphertext, C, as a result

$$C = encrypt(K_1, M)$$

- Decrypt takes a key, K_2, and cyphertext, C, as arguments, and produces a plaintext message, M, as a result

$$M = decrypt(K_2, C)$$

- Mathematically, decrypt is the inverse of encrypt

$$M = decrypt(K_2, encrypt(K_1, M))$$

Two Main Types Of Encryption

- Private or secret key encryption (symmetric)
  - Encryption and decryption use same key
  - Key is a *shared secret*

$$M = decrypt(K, encrypt(K, M))$$

- Public key encryption (asymmetric)
  - Encryption and decryption use different keys
  - *Public key* is widely disseminated
  - *Private key* is known only to one party
  - Knowing a user's public key does not help one guess the corresponding private key

Authentication With Digital Signatures

- Uses encryption (works well with *public key* methods)
- Allows receiver to verify the identity of the sender
- Example
  - Bob sends message to Alice
    * Uses his *private key* to encode message
    * Includes specific information such as Alice's name and a date to avoid a replay attack
  - Alice
    * Uses Bob's public key to decrypt message
    * Knows that only Bob could have sent the message

Authentication With Digital Signatures (continued)

- Can use additional level of encryption to guarantee confidentiality
- Bob signs message and encrypts using Alice's public key

$$X = encrypt(alice\_pub, encrypt(bob\_priv, M))$$

- Alice decrypts message with her private key, and then authenticates the sender by decrypting with Bob's public key

$$M = decrypt(bob\_pub, decrypt(alice\_priv, X))$$

Key Distribution

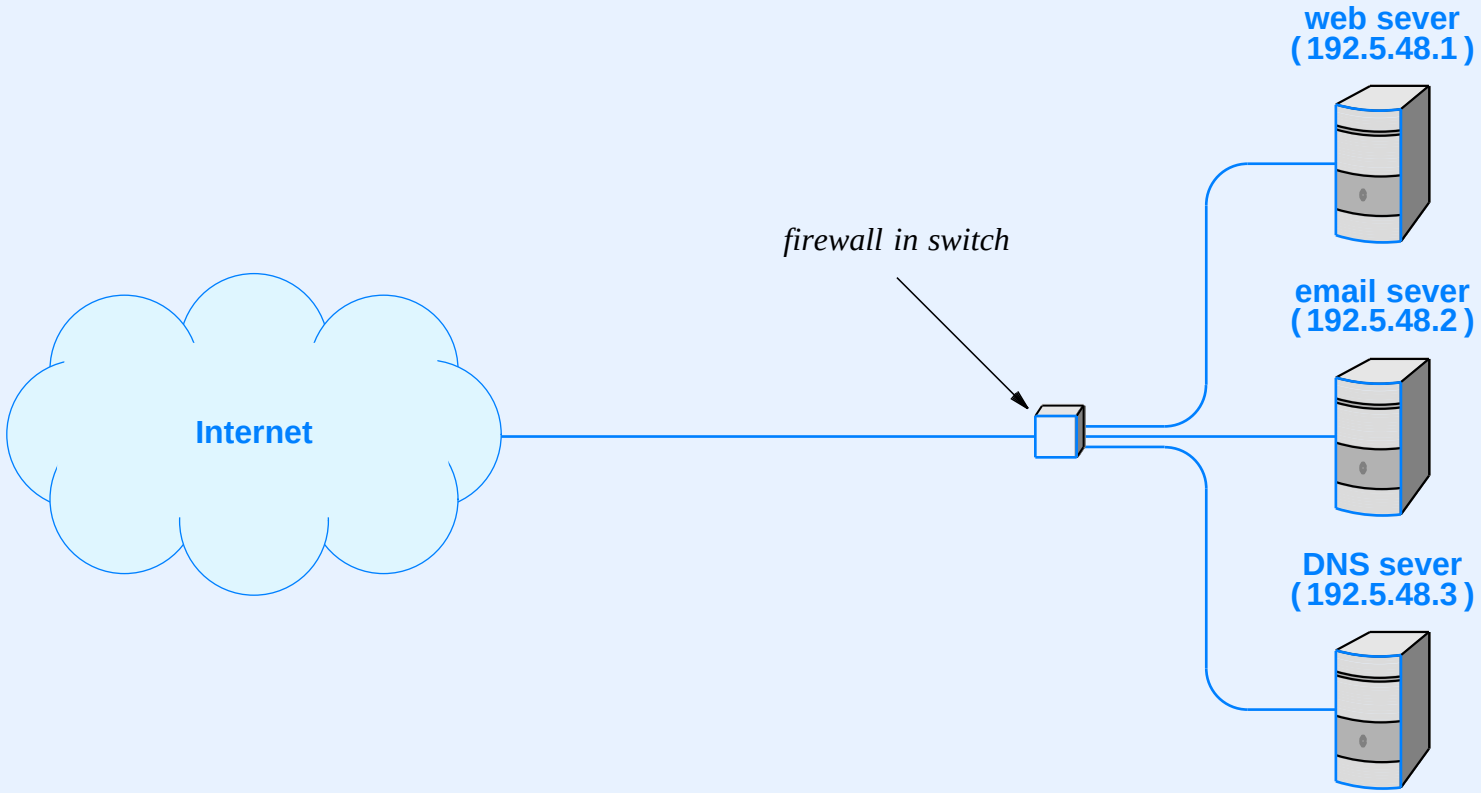
- Everyone needs to obtain a copy of each user's public key
- If an attacker distributes an incorrect key, the entire encryption scheme is compromised
- Question: how can public keys be distributed in a way that guarantees each copy is correct?
- Several solutions have been proposed; most rely on *key authority* organizations that hand out public keys
- Message containing keys signed by well-known authority is a *digital certificate*
- Note: knowing the public key of an authority makes it possible to obtain other public keys securely

Firewall Technology

- Inserted between site and Internet
- Filters packets according to policy
- Controls both incoming and outgoing traffic
- General approach: prevent all communication unless explicitly allowed by policy

Firewall Example

- Consider a site with three servers



- Firewall only allows packets to/ from each server

Firewall Example
(continued)

- Example of firewall rules for the site:

Dir	Frame Type	IP Src	IP Dest	IP Type	Src Port	Dst Port
in	0800	*	192.5.48.1	TCP	*	80
in	0800	*	192.5.48.2	TCP	*	25
in	0800	*	192.5.48.3	TCP	*	53
in	0800	*	192.5.48.3	UDP	*	53
out	0800	192.5.48.1	*	TCP	80	*
out	0800	192.5.48.2	*	TCP	25	*
out	0800	192.5.48.3	*	TCP	53	*
out	0800	192.5.48.3	*	UDP	53	*

Other Network Security Systems

- Intrusion Detection System (IDS)
  - Watches incoming packet stream
  - Attempts to identify unusual activity
- Deep Packet Inspection (DPI)
  - Looks beyond header into packet contents
  - Requires significant processing
- File inspection systems
  - Examine whole data file (e.g., email)
  - Can detect more problems than systems that examine individual packets

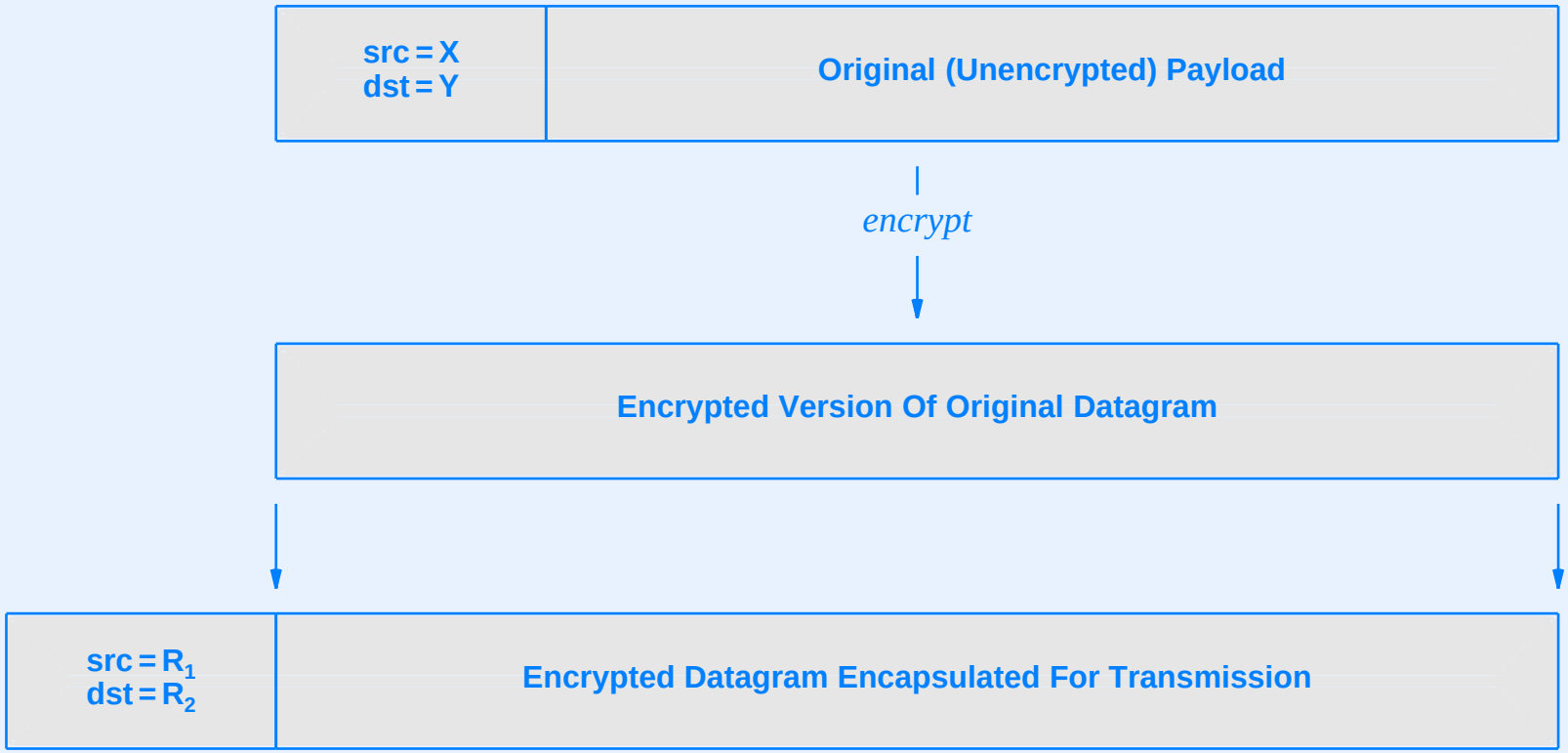
Virtual Private Network (VPN)

- Emulates a dedicated network connection
- Sends traffic across commodity Internet
- Uses encryption to guarantee confidentiality
- Technique known as *tunneling*
- Can be used
  - Among sites of an organization
  - Between individual and organization

Encryption And Tunneling Used In VPNs

- Three basic approaches used
  - Payload encryption
  - IP-in-IP tunneling
  - IP-in-TCP tunneling
- Original data is encrypted in all three
- For additional security, pad datagram length

Illustration Of IP-in-IP Tunneling Used For A Secure VPN



Examples Of Security Technologies

- *PGP* (*Pretty Good Privacy*)
- *SSH* (*Secure Shell*)
- *SSL* (*Secure Socket Layer*)
- *TLS* (*Transport Layer Security*)
- *HTTPS* (*HTTP Security*)
- *IPsec* (*IP security*)
- *RADIUS* (*Remote Authentication Dial-In User Service*)
- *WEP* (*Wired Equivalent Privacy*)
- *WPA* (*Wi-Fi Protected Access*)

Network Management

Terminology

- *Network manager* or *network administrator* is a person responsible for network
  - Planning
  - Installation
  - Operation
  - Monitoring
- *Network* refers to *intranet*
  - Owned and operated by a single organization
  - Contains many managed items such as routers, switches, servers, and hosts
  - May span multiple sites

An Interesting Problem

- Many protocol mechanisms have been created to overcome problems automatically
  - Forward error correction
  - Retransmission
  - Routing protocols
- Consequence: protocols may hide problems from a manager!

The Industry Standard Model

- Derived from ITU recommendation M.3400
- Known by abbreviation, *FCAPS*
- Acronym refers to five aspects of management

Abbreviation	Meaning
F	Fault detection and correction
C	Configuration and operation
A	Accounting and billing
P	Performance assessment and optimization
S	Security assurance and protection

Fault Isolation And Root-Cause Analysis

- Users report high-level symptoms
  - Example: I lost access to a shared file system
- Manager must relate symptoms to underlying cause
  - Cable cut
  - Power supply has failed or disk has crashed
  - Software configuration changed (e.g., file system renamed or moved)
  - Security changed (e.g., password expired)

Network Element

- Generic term for a managed entity
  - Physical device
  - Service (e.g., DNS)
- Examples

Manageable Network Elements	
Layer 2 Switch	IP router
VLAN Switch	Firewall
Wireless Access Point	Digital Circuit (CSU/DSU)
Head-End DSL Modem	DSLAM
DHCP Server	DNS Server
Web Server	Load Balancer

Element Management System

- Management tool that can manage one element at a time
- Typically, supplied by vendor of the network element
- Limitation of element management systems
  - When configuring MPLS tunnel across multiple routers, element management system only allows manager to configure one router at a time
  - If routers sold by multiple vendors, each vendor may have its own element management system
- Unfortunately, many networks only have element management

Types Of Network Management Tools

Physical Layer Testing

Reachability And Connectivity

Packet Analysis

Network Discovery

Device Interrogation

Event Monitoring

Performance Monitoring

Flow Analysis

Routing And Traffic Engineering

Configuration

Security Enforcement

Network Planning

How Should Management Systems Operate?

- Some possibilities
  - Use a parallel physical network
  - Use a parallel logical network
  - Use a special link-layer protocol
  - Use the same links, equipment, and protocols as data
- Surprise: modern network management follows the last approach

Simple Network Management Protocol (SNMP)

- Internet standard
- Allows software in a manager's computer (*manager*) to interact with software that runs in an element (*agent*)
- Specifies format and meaning of messages exchanged
- Runs as an application protocol over TCP or UDP
- Uses *fetch-store* paradigm

SNMP Fetch-Store Paradigm

- Set of conceptual variables defined
- Each variable given a name
- Set of variables known as *Management Information Base (MIB)*
- SNMP offers two basic operations
  - *GET* to read the value of a variable
  - *PUT* to store a value into a variable
- All management functions are defined as side-effects of GET or PUT to a MIB variable
- Example: reboot defined as side-effect of PUT

SNMP Encoding

- SNMP uses a standard known as *Abstract Syntax Notation.1 (ASN.1)*
- Variable-length encoding
- Example: integer encoded as length and value

Decimal Integer	Hexadecimal Equivalent	Length Byte	Bytes Of Value (in hex)
27	1B	01	1B
792	318	02	03 18
24,567	5FF7	02	5F F7
190,345	2E789	03	02 E7 89

MIB Variable Names

- Are hierarchical
- Begin with standard prefix
- Identify a specific protocol and variable
- Example: counter for IP packets received has name

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

- Name is encoded as integers:

1.3.6.1.2.1.4.3

Arrays In A MIB

- ASN.1 does not define an array type
- Many MIB variables correspond to conceptual array
  - Routing table
  - ARP cache
  - Set of network interfaces
- Trick
  - The “index” is appended onto variable name
  - Manager software uses *GET-NEXT* operation to move through array

Example Of Indexing

- IP routing table assigned variable name

standard-prefix.ip.ipRoutingTable

- Each field has a name
- Issuing GET_NEXT operation gets first routing table entry
- For example, name of destination address field variable is

standard-prefix.ip.ipRoutingTable.ipRouteEntry.field.IPdestaddr

A Plethora Of MIBs

- Initially
  - One MIB
  - Defined variables for IP, TCP, UDP, ICMP
- Now
  - Many MIBs
  - Variables for routers, switches, modems, printers, hosts, and other network elements

Summary

- Streamed transfer of real-time data incompatible with Internet's best-effort delivery
- Two approaches
  - Isochronous network
  - Timestamps and jitter buffer
- Real-Time Transport Protocol (RTP) uses timestamps and sequence numbers

Summary
(continued)

- Many IP telephony standards proposed
- Connection to PSTN causes debate
- H.323 and SIP standards are most widely used
- ENUM system uses DNS to convert phone number to IP address

Summary
(continued)

- Quantitative measures of networks include delay, throughput, goodput, and jitter
- Delay increases as utilization increases
- One can purchase more throughput, but not less delay
- Quality of Service (QoS) technologies provide guarantees on performance
- The industry has moved away from fine-grain QoS (per-flow as in ATM and IntServ) to coarse-grain QoS (DiffServ)
- Multi-Protocol Label Switching (MPLS) is used by tier-1 ISPs to provide circuit-oriented networking

Summary
(continued)

- Network security is complex and difficult
- No network is completely secure
- Life goes on anyway
- Network management is complex and difficult
- Current tools are fairly primitive
- Life goes on anyway

MODULE VII

Emerging Technologies

Topics

- Software Defined Networking
- The Internet Of Things
- Other trends in networking

Software Defined Networking (SDN)

What Is Software Defined Networking?

- One of the hottest topics in networking
- According to marketing SDN is
  - A way to eliminate all human error
  - A technology that improves overall routing
  - An approach that eliminates 66% to 80% of operational costs
- In reality SDN is
  - A technology that gives programmers more control over network equipment
  - An approach with the *potential* to make some improvements in network configuration and management

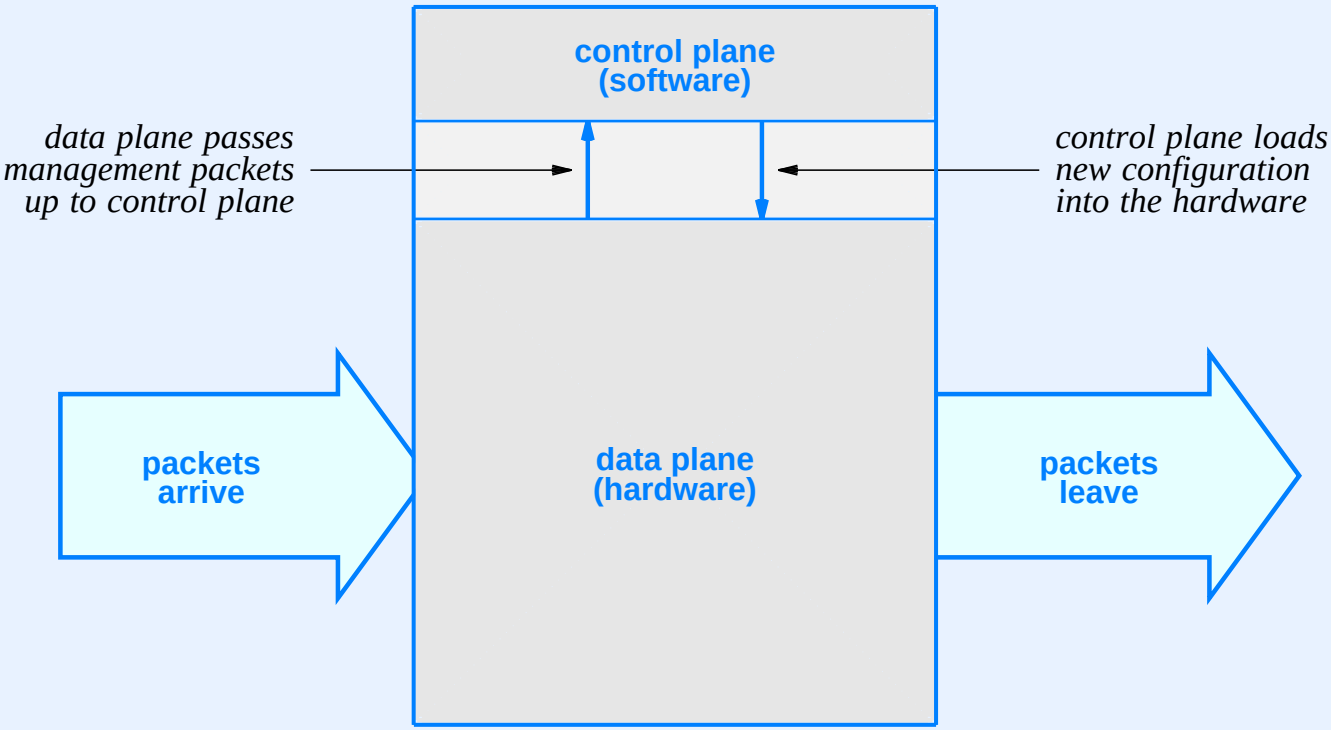
Motivation For SDN

- Switch from element management to network management
- Move from proprietary to open standards
- Automate and unify network-wide configuration
- Change from per-layer to cross-layer control
- Accommodate virtualization used in data centers

Background And Definitions

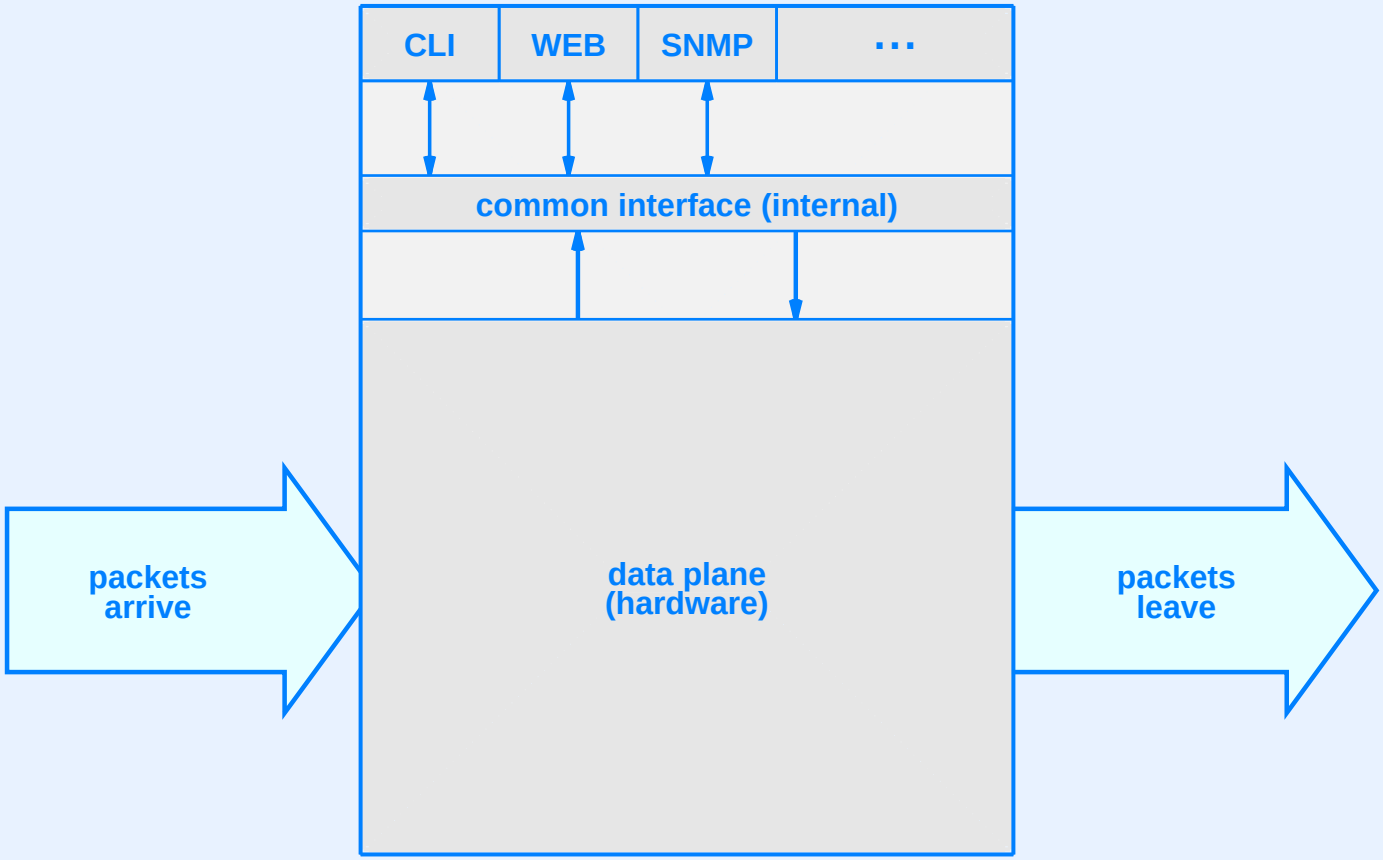
- Terminology adopted from network equipment engineers
- *Data plane*
  - Refers to packet processing mechanisms
  - Typical functions include packet classification and packet forwarding
  - Operates at wire speed
- *Control plane*
  - Refers to management
  - Typical functions include interacting with network manager and modifying forwarding tables
  - Operates slowly and only when changes are needed

Conceptual Organization Of Network Devices



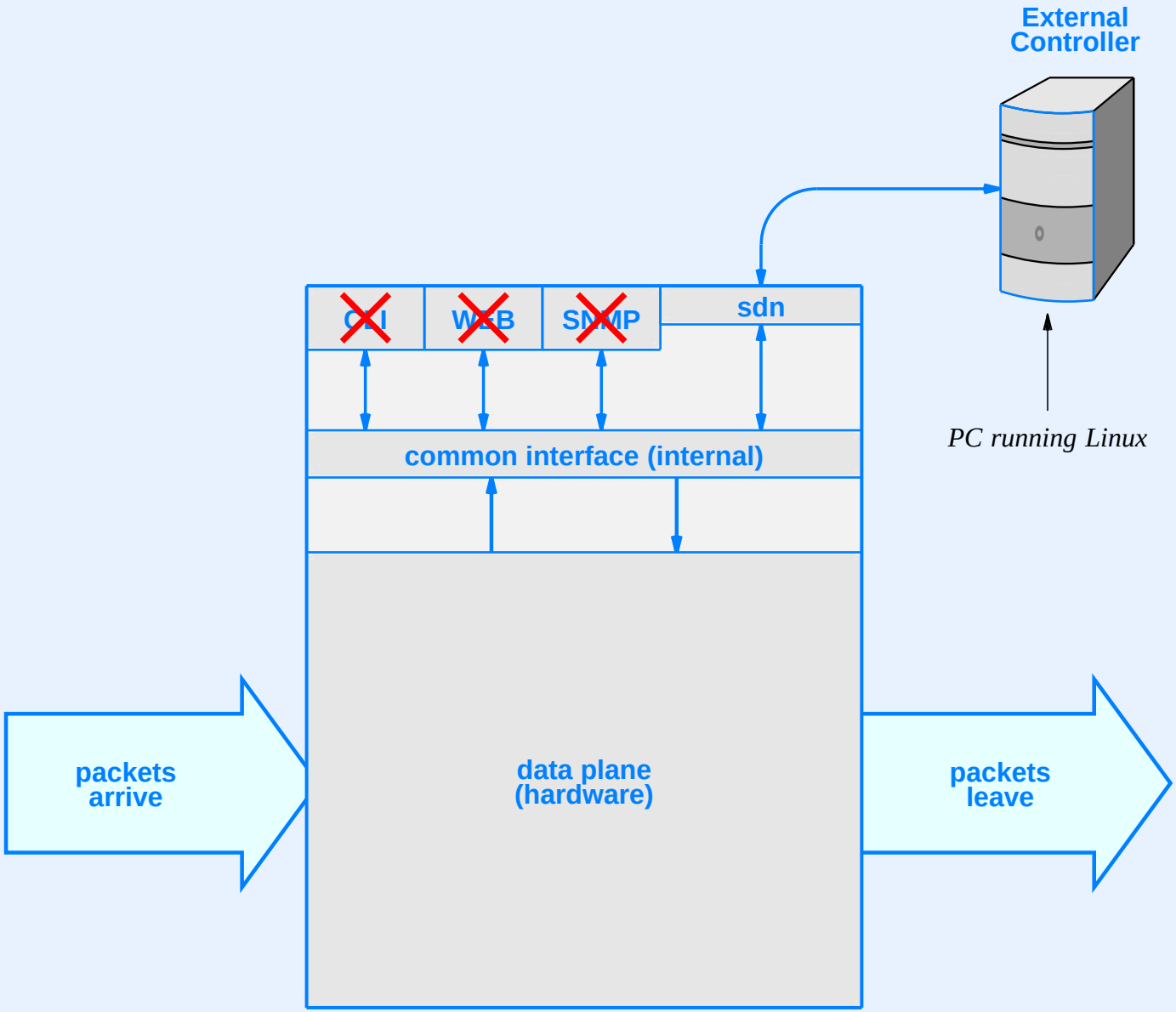
- Data plane may use ASIC hardware for speed
- Control plane includes a TCP/IP stack

Control Plane Interface Modules

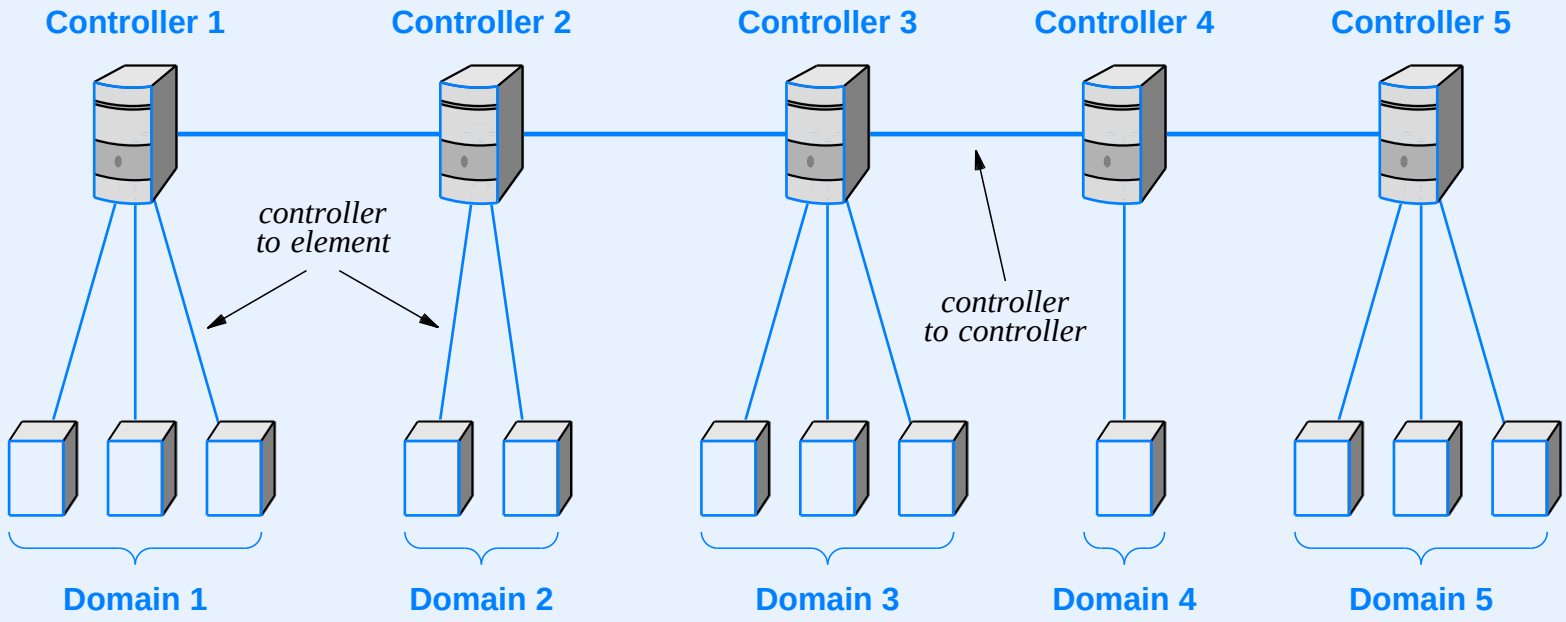


- Managers can choose among command line interface, web interface, and SNMP

The SDN Approach: An External Controller



In Practice



- Each controller can operate multiple devices
- Controllers coordinate to provide consistent configuration

SDN Communication

- Two conceptually separate types
  - Controller to network element
  - Controller to controller
- Protocols used can differ

OpenFlow

- Specification for controller-to-element communication
- Devised at Stanford
- Now a de facto industry standard for SDN
- Defines
  - Secure communication (over SSL)
  - Message format
  - Items to be managed
- Completely unlike SNMP

OpenFlow Model

- Uses *flow table* abstraction
  - Data plane is assumed to have a sequence of flow tables
  - Each flow table specifies how to parse packets and handle them
- OpenFlow allows manager to set values in each flow table
- Important note: flow table model closely matches classification hardware found in Ethernet switches

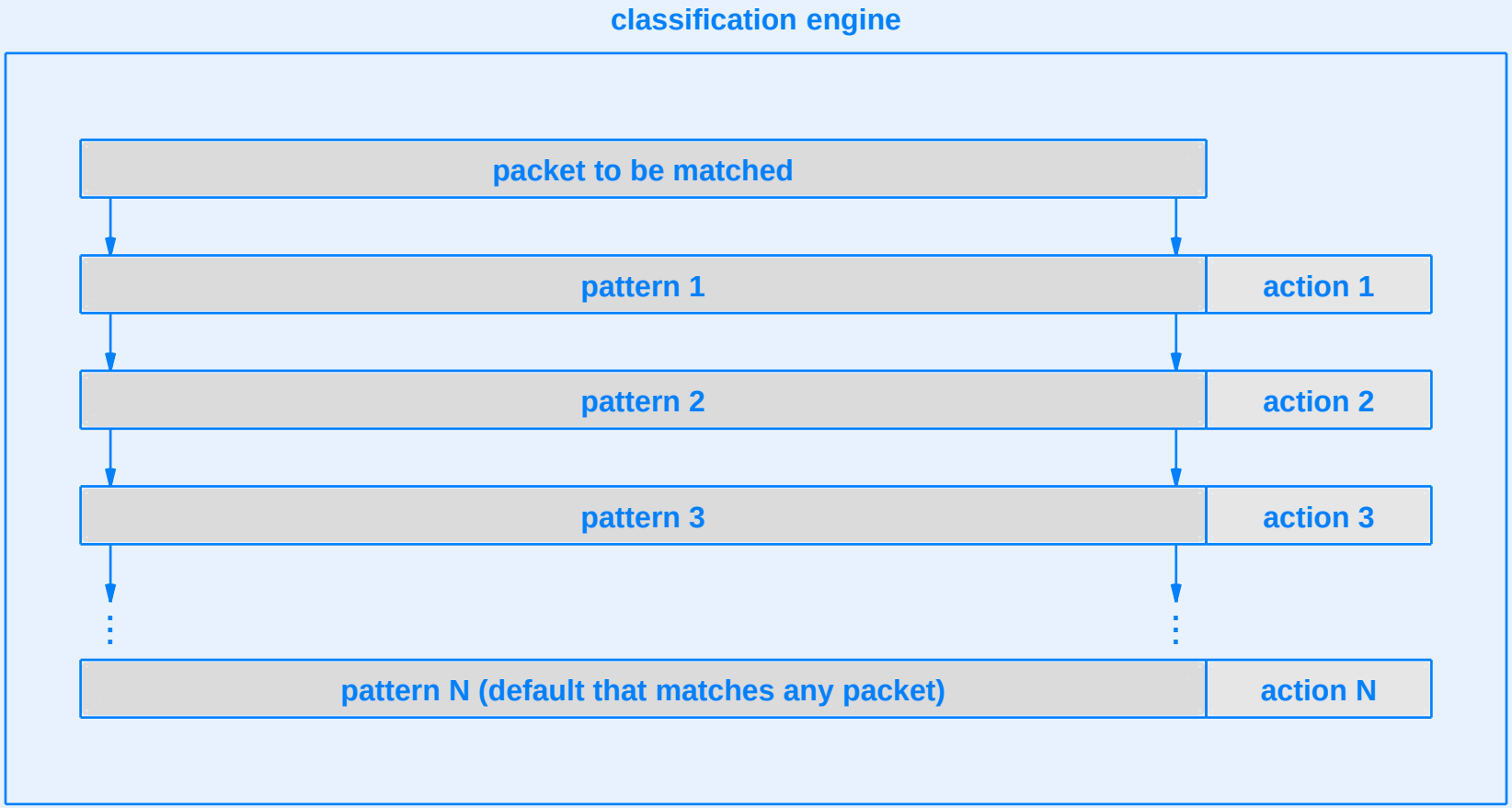
Classification

- Alternative to packet demultiplexing
- Examines headers from multiple layers at the same time
- Uses an array of pairs

(*pattern*, *action*)

- Where
  - *Pattern* is a pattern that is matched against packets
  - *Action* specifies steps to be taken if the match succeeds

Classification Hardware



- Hardware checks all patterns in parallel
- Result is extremely high speed classification

TCAM

- Acronym for *Ternary Content Addressable Memory*
- Hardware technology used for high-speed classification
- Pattern is *ternary* because value for each bit can be 0, 1, or "don't care"
- TCAM matches all patterns at once, and performs the action on the first matching table entry

Example Of IPv4 Classification

- The challenge
  - A frame arrives
  - What is the minimum number of steps needed to determine whether the frame carries an IPv4 datagram destined for a web server?
- The answer
  - Check whether the frame type field specifies IPv4 (0x0800)
  - Check whether the IP protocol field specifies TCP (6)
  - Check whether the TCP destination port specifies a web server (80)

IPv6 Classification

- Simplest case (only a base header)
  - Frame type field specifies IPv6 (0x86DD)
  - Next Header field specifies TCP (6)
  - TCP destination port specifies a web server (80)
- Additional patterns needed for extension headers
- Example: base header plus a route header
  - Frame type field specifies IPv6 (0x86DD)
  - Next Header field specifies Route Header (43)
  - Next Header field specifies TCP (6)
  - TCP destination port specifies a web server (80)

Example Items In An OpenFlow Pattern

Field	Meaning
Layer 2 fields	
Ingress Port	Switch port over which the packet arrived
Metadata	64-bit field of metadata used in the pipeline
Ether src	48-bit Ethernet source address
Ether dst	48-bit Ethernet destination address
Ether Type	16-bit Ethernet type field
VLAN id	12-bit VLAN tag in the packet
VLAN priority	3-bit VLAN priority number
ARP opcode	8-bit ARP opcode
Layer 3 fields	
MPLS label	20-bit MPLS label
MPLS class	3-bit MPLS traffic class
IPv4 src	32-bit IPv4 source address
IPv4 dst	32-bit IPv4 destination address
IPv6 src	128-bit IPv6 source address
IPv6 dst	128-bit IPv6 destination address
IPv4 Proto	8-bit IPv4 protocol field
IPv6 Next Header	8-bit IPv6 next header field
TOS	8-bit IPv4 or IPv6 Type of Service bits

Example Items In An OpenFlow Pattern (continued)

Field	Meaning
Layer 4 fields	
TCP/UDP/SCTP src	16-bit TCP/UDP/SCTP source port
TCP/UDP/SCTP dst	16-bit TCP/UDP/SCTP destination port
ICMP type	8-bit ICMP type field
ICMP code	8-bit ICMP code field

Examples Of SDN Functionality

- End-to-end layer 2 paths
- Forwarding based on source as well as destination
- All traffic from a specific MAC address sent along a specific path
- Segregation of traffic based on application type
- Multipath forwarding based on hash of 4-tuple
- Transport of nonstandard layer 3 protocols


Questions?

The Internet Of Things

Internet Of Things

- Awkward term used for embedded systems on the Internet
  - Generally not operated by humans
  - Can access one another or cloud services
- Examples
  - Scientific sensor systems
  - Home automation systems
  - Smart grid
  - Retail systems

Technology Characteristics

- Low power
  - Energy harvesting (e.g., door latch)
  - Multi-year battery life
- Wireless communication
  - Necessary in many situations
  - Enables mobility

Wireless Mesh Network

- Useful when individual nodes have very low power (limited range)
- Allows a set of nodes to communicate even if some nodes cannot communicate directly
- Each node agrees to forward packets on behalf of neighbors

Example Wireless Mesh Technology

- ZigBee IP
  - Created by ZigBee Alliance
  - Uses IEEE 802.15.4 wireless radios
  - Intended for smart grid
- ZigBee protocol stack
  - Goal is to run IPv6, TCP, and HTTP
  - Includes many other protocols

802.15.4 Wireless Characteristics

- Goal is low power, and result is
  - Extremely low data rate
  - Extremely small MTU
  - Limited distance

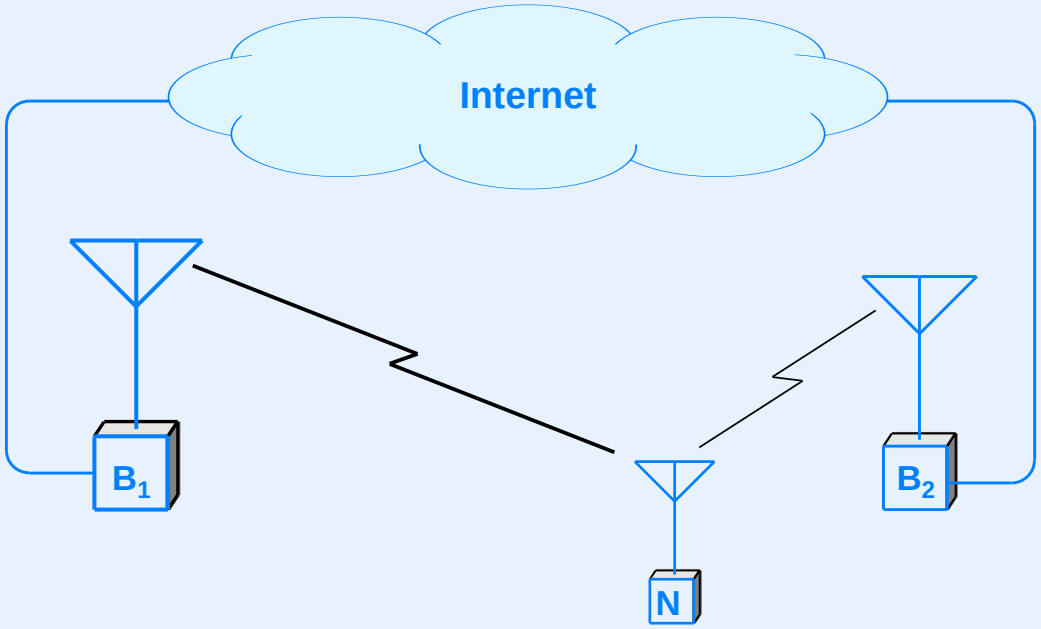
Property	Value
Networking paradigm	Packet switching
Maximum data rate	250 Kbps
Payload size (MTU)	102 octets
Maximum distance	10 meters

ZigBee IP Mesh Routing

- One or more *border routers*
  - Connect to global Internet
  - Are more powerful than other nodes
- Set of *ZigBee IP routers* (*ZIP routers*)
  - Attach to appliances
  - Form a mesh
  - Forward traffic to border router

Choosing A Path

- ZIP router must choose a path to a border router
- Cannot simply choose node with the strongest transmission signal



- Additional protocol used to find which node *receives* strongest signal (MLE)

Running IPv6 Over A ZigBee Network

- IPv6 can only run over networks that offer an MTU of 1280 or higher, but 802.15.4 has an MTU of 102
- Solution
  - Additional protocol named 6LoWPAN
  - Shim layer between IP and device driver

6LoWPAN Operation

- Sending side
  - Divides datagram into series of blocks
  - Transmits each block in a packet
- Receiving side
  - Joins blocks into a datagram
  - Delivers entire datagram to IPv6
- Notes
  - Division into block does *not* use IP fragmentation
  - Unlike fragmentation, division and regrouping is performed at each hop

ZigBee IP Mesh Routing

- ZIP nodes forward packets toward the border router
- Border router
  - Can send outgoing packets to the Internet
  - Forwards other packets across the mesh
- If two ZIP nodes communicate
  - Packet goes to border router first
  - Border router forwards to destination

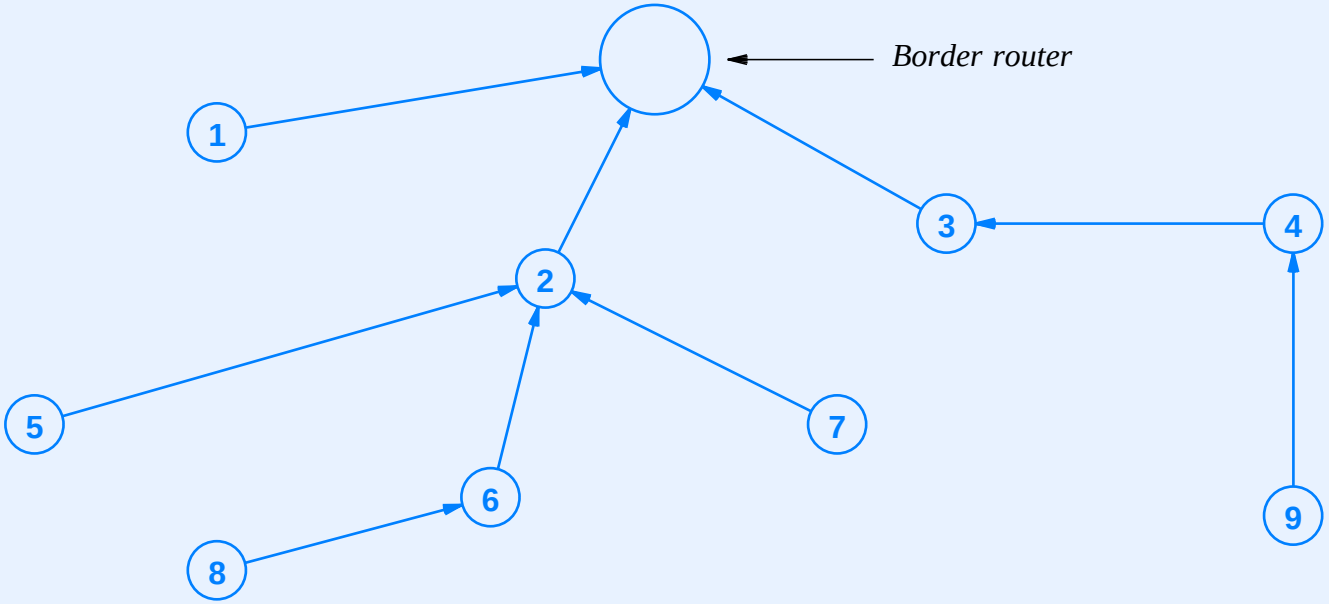
Border Router Operation

- To forward across the mesh the border router
  - Learns the topology of the mesh
  - Computes a path through the mesh to each ZIP node
  - Uses IPv6 source routing
- IPv6 source routing
  - Requires IP-in-IP tunneling (header modification prohibited)
  - Places an extension header on outer datagram with series of hops
  - Each ZIP node only needs to know its neighbors

Computing Source Routes

- All nodes run *Routing Protocol for lossy and Low power networks* (*RPL*)
- Each node reports its parent to the border router
- RPL code on border router creates a *Destination Oriented Directed Acyclic Graph* (*DODAG*)
- DODAG is used to compute source routes

Example DODAG



- Arcs in DODAG point to parent (path toward border router)
- Source route to node X is reverse of the path from X to border router

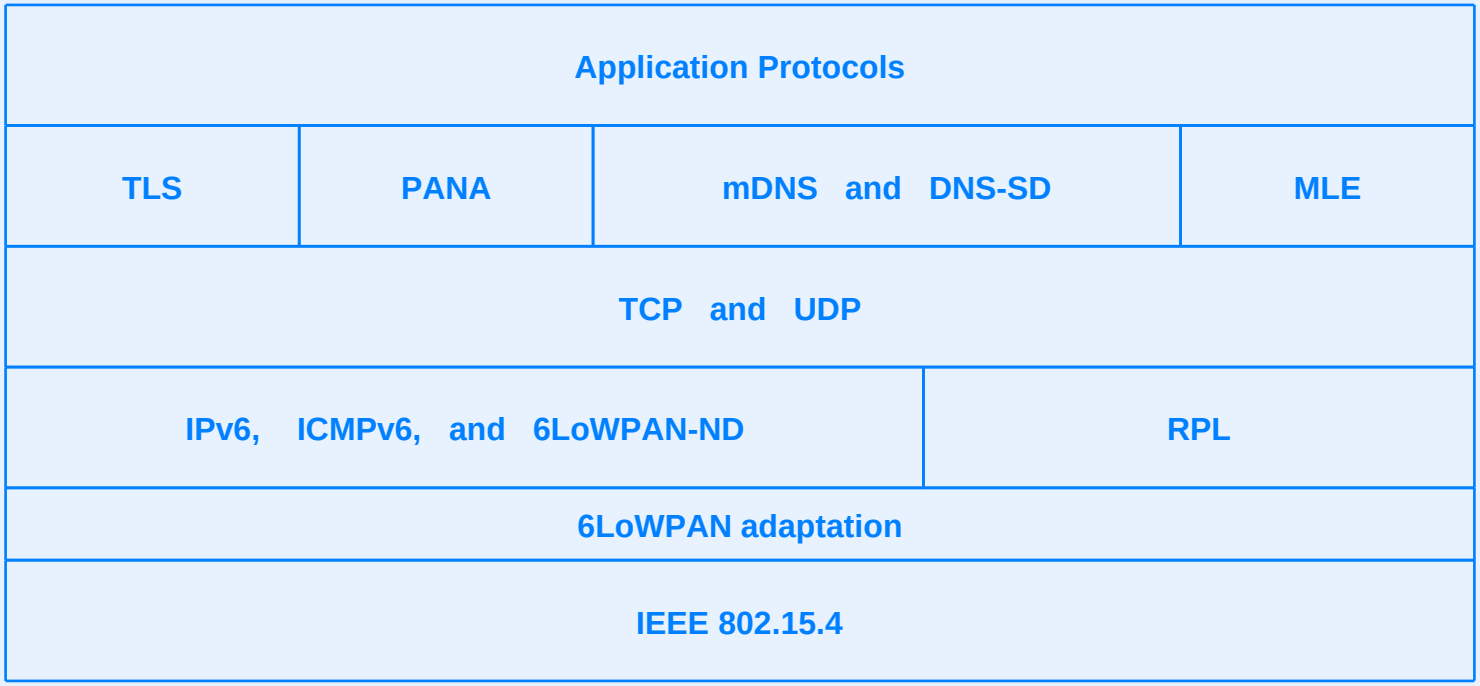
Does ZigBee IP Make Sense?

- Choosing IPv6 instead of IPv4 means
  - Much larger datagram headers
  - The use of 6LoWPAN to divide a datagram into MTU-size pieces
  - Sending more data over a slow network
  - The need for RPL routing protocols
  - Larger memories (and lower battery life)
- Using TCP and HTTP over IPv6 means
  - Using DNS to resolve names
  - Unnecessary overhead
  - Unnecessary memory footprint

But Wait, There's More!

- Smart grid applications must be secure, so ZigBee IP includes security protocols, including TLS
- IPv6 Neighbor Discovery doesn't work in a mesh network, so ZigBee IP includes a modification known as 6LoWPAN-ND
- IEEE 802.15.4 allows short (16-bit) MAC addresses, so ZigBee IP includes a mechanism that allows a border router to prevent address collisions

Major Items In The ZigBee Protocol Stack



- Resulting stack is large
- Design is more general-purpose than necessary
- Technology may be a triumph of politics and economics

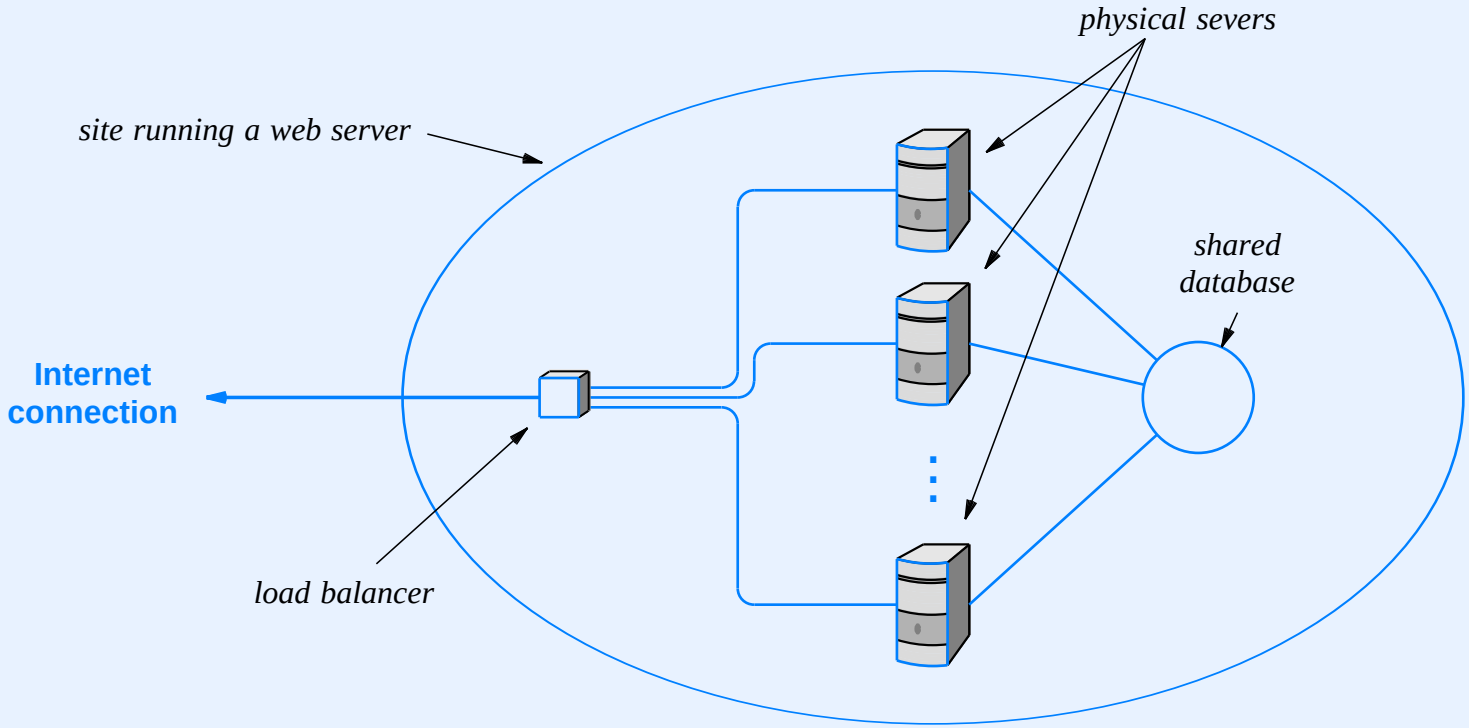

Questions?

Other Trends
In Networking

A Few Key Technologies

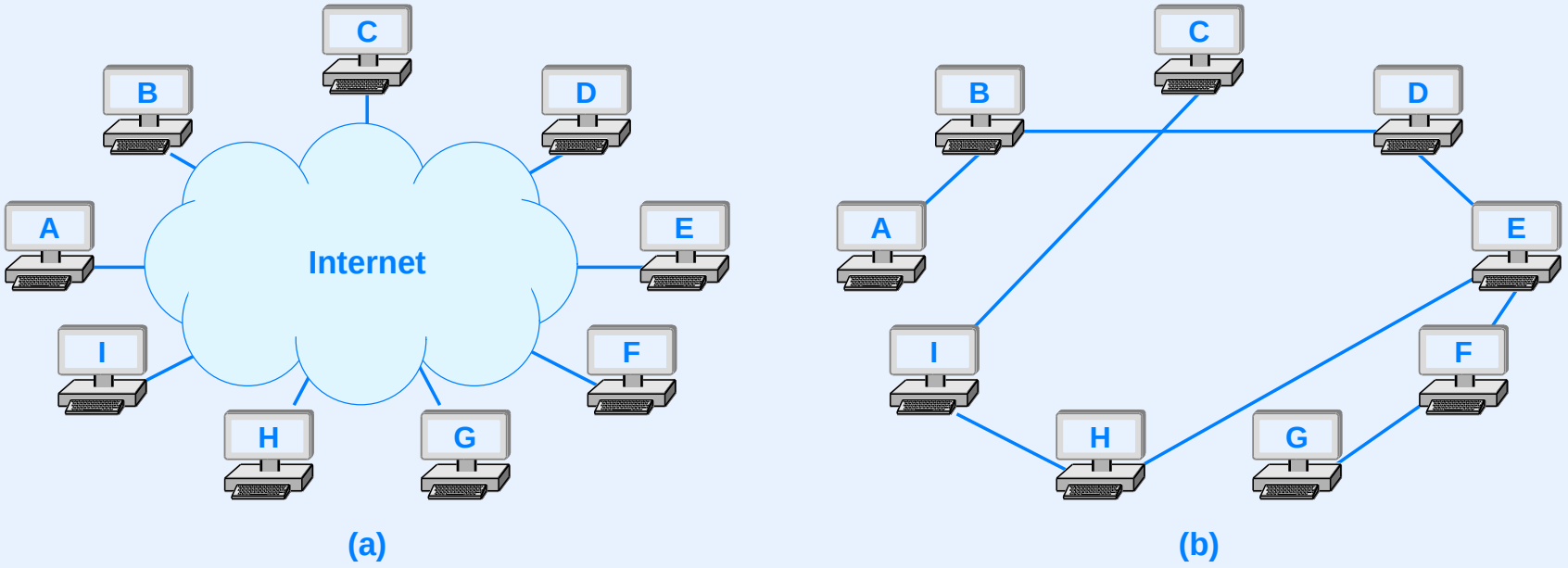
- Content Caching
- Peer-To-Peer Communication
- Universal Representation (XML)
- Wireless networks that support mobility
- Higher-speed access technologies (1 Gbps)
- Cloud computing and cloud data centers

Web Load Balancers



- Load balancer distributes HTTP requests across servers
- Path from servers back to client may be higher speed

Overlay Networking



(a) Physical connection of computers to the Internet

(b) Logical network imposed by overlay routing

Other Trends

- Switch to digital telephony and digital video
- Increased use of social networking and social media
- Distributed data centers and migration


STOP