

Week 1 – Introduction

COMP90007
Internet Technologies

Chien Aun Chan

Outline

- Computer Networks
- Network Types
- The Internet

Terminologies

- A network device: eg. PC, Router, Switch, Phone
- Server: Provider of a service. Accept requests from clients
- Client: A network device connecting to a server and requesting a service
- Computer Network: A collection of autonomous computers interconnected by a single technology

Terminologies

- Packet: A message send between two network device (more specific definitions will be given during the course)
- IP address: A unique number identifying a network device

What is a Network?

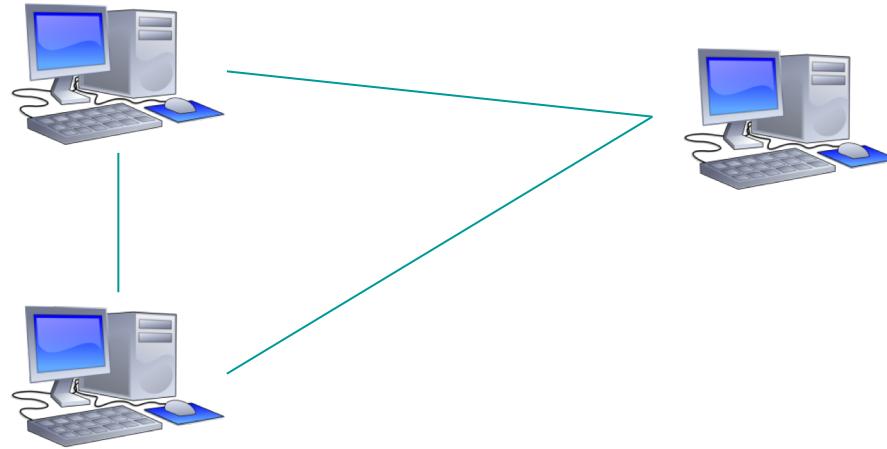
- **Network (Noun):**

- An intricately connected system of things or people
 - An interconnected or intersecting configuration or system of components

- **Computer Network:**

- A data network with computers at one or more of the nodes [Oxford Dictionary of Computing]
 - A collection of autonomous computers interconnected by a single technology

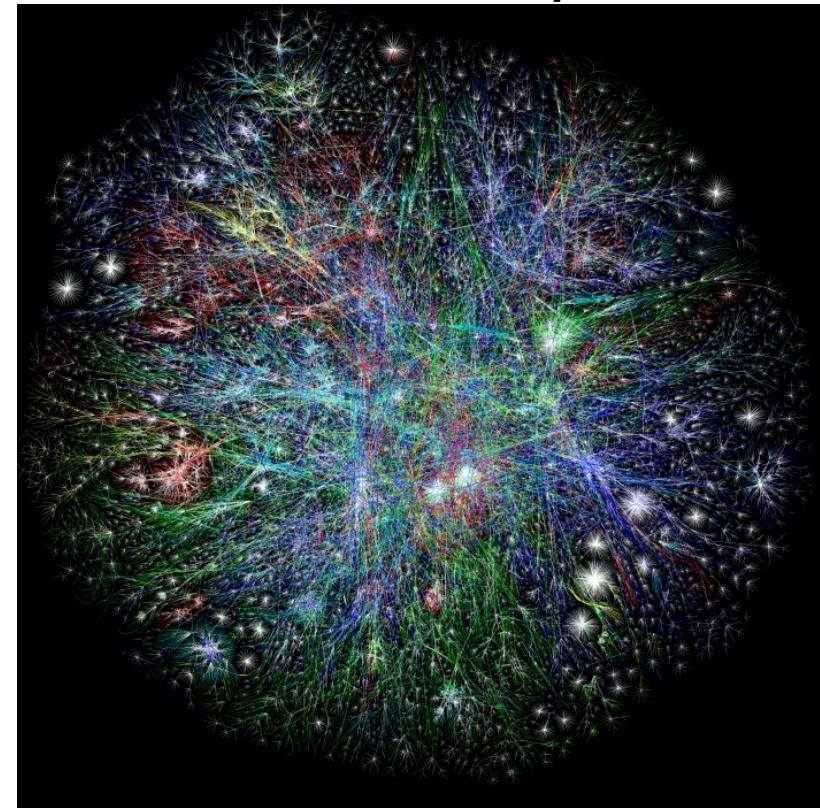
Computer Networks



How does it scale to billions of devices?
What about distances?

What are the Internet and the World Wide Web?

- Neither the Internet nor the WWW is a computer network!
- Simple answers:
 - The Internet is not a single network but a network of networks!
 - The WWW is a distributed system that runs on top of the Internet



<https://mountpeaks.wordpress.com/>

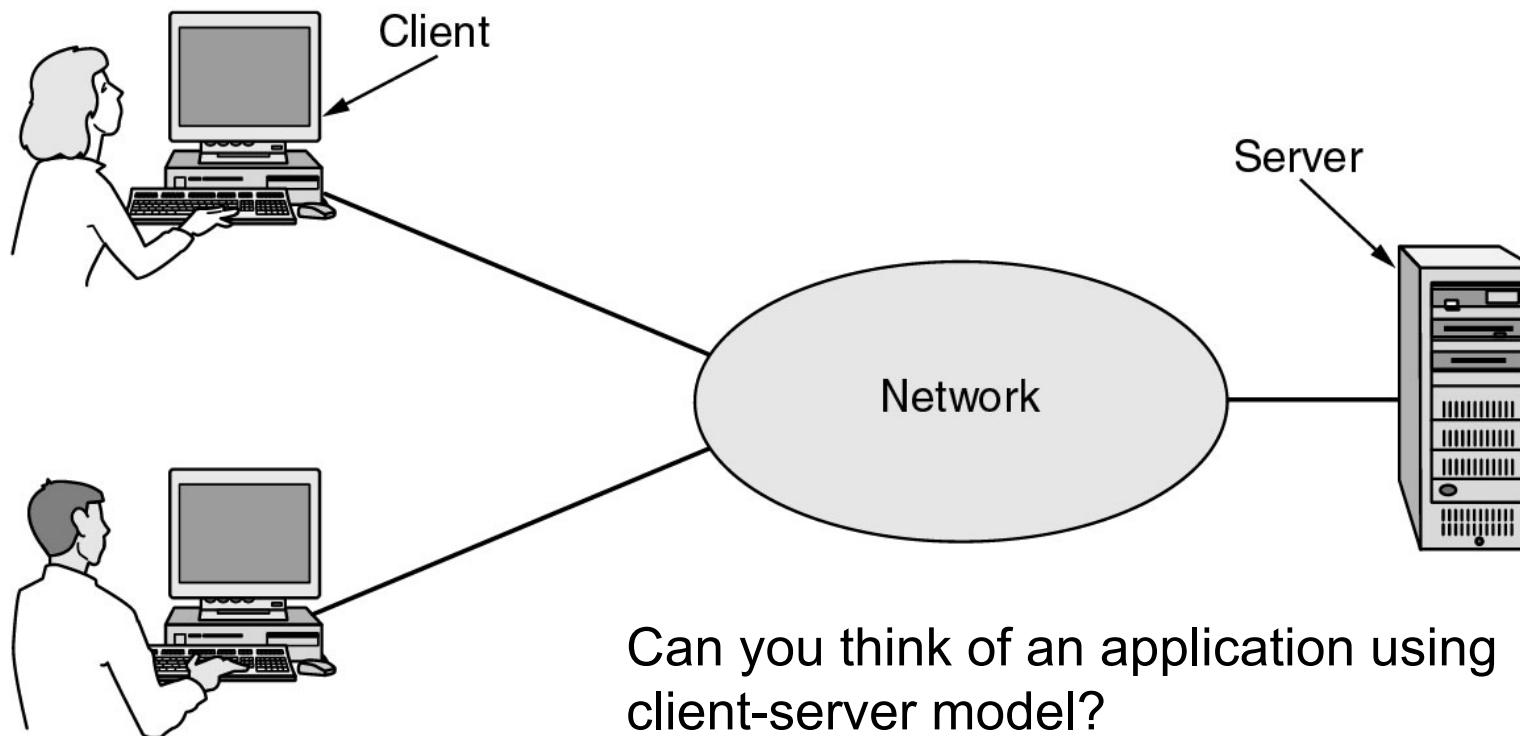
Uses of Computer Networks

- Business Applications
 - Resource sharing (e.g., printer, scanner)
- Home Applications
 - Access to remote information
 - Interactive entertainment
 - E-commerce
- Mobile Users
 - Mobility
 - Internet-of-things (e.g., parking, smart-meter, vending machines)
- Social Interactions

How many different types of networks have you used?

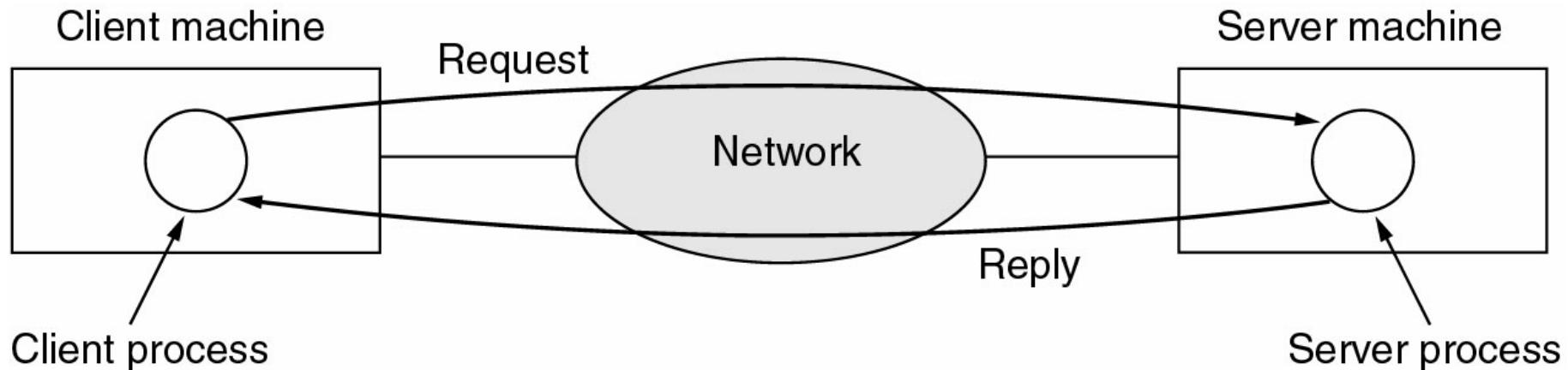
Business Applications of Networks

- A Simple Client-Server Network
- A network with two clients and one server



Business Applications of Networks (2)

- The client-server model involves requests and replies



Differentiating Factors of Networks

■ Types of transmission technology

□ Broadcast link

- Broadcast networks have a single communication channel shared by all machines on a network. Packets sent by any machine are received by all others, an address field in the packet specifies the intended recipient. Intended recipients process the packet contents, others simply ignore it.
- Broadcasting is a mode of operation which allows a packet to be transmitted that every machine in the network must process.

Differentiating Factors of Networks

■ Types of transmission technology

□ Point-to-point links

- Data from sender machine is not seen and process by other machines
- Point to point networks consist of many connections between individual pairs of machines. Packets travelling from source to destination must visit intermediate machines to determine a route - often multiple routes of variant efficiencies are available and optimisation is an important principle.
- Unicasting is the term used where point-to-point networks with a single sender and receiver pair can exchange data

□ Multicasting

- Transmission to a subset of the machines

Differentiating by Scale

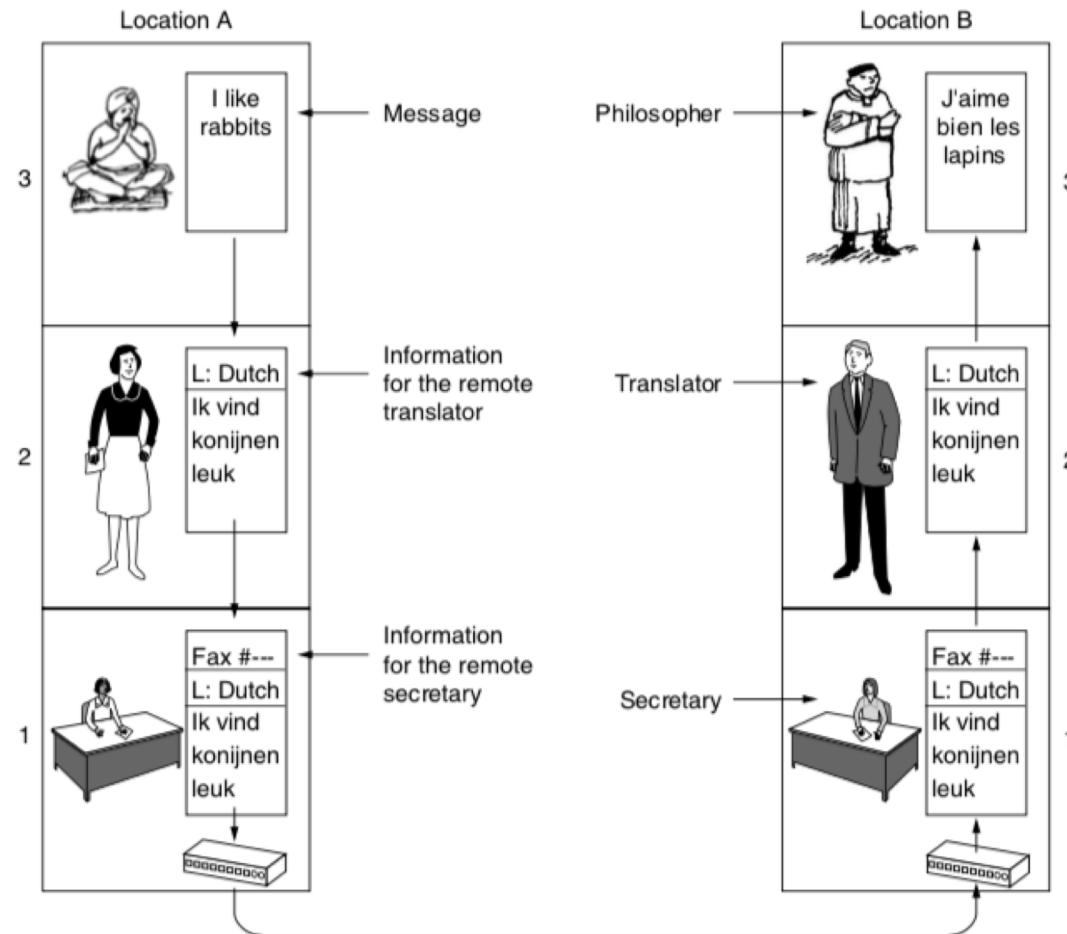
- Classification of interconnected processors by scale.

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	
100 m	Building	Local area network
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	
1000 km	Continent	Wide area network
10,000 km	Planet	The Internet

Outline

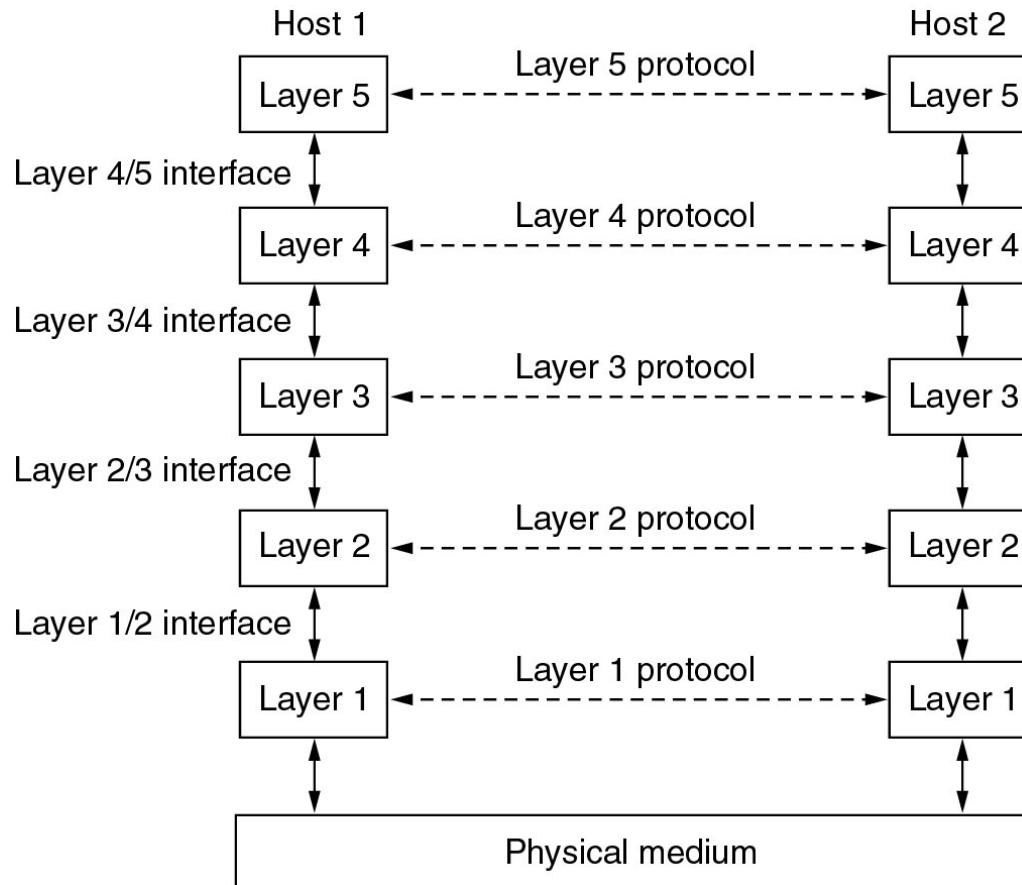
- Protocols, Layers and Services
 - Protocol Hierarchies
 - Design of Layer Models
 - Connection-Oriented and Connectionless Services
 - Services Primitives
 - Services and Protocols
- Network Reference Models
 - Open Systems Interconnect
 - TCP/IP
- Network Standards

The Philosopher-translator-secretary Architecture



Network Software

Protocol Hierarchies



Consider the network as a stack of layers

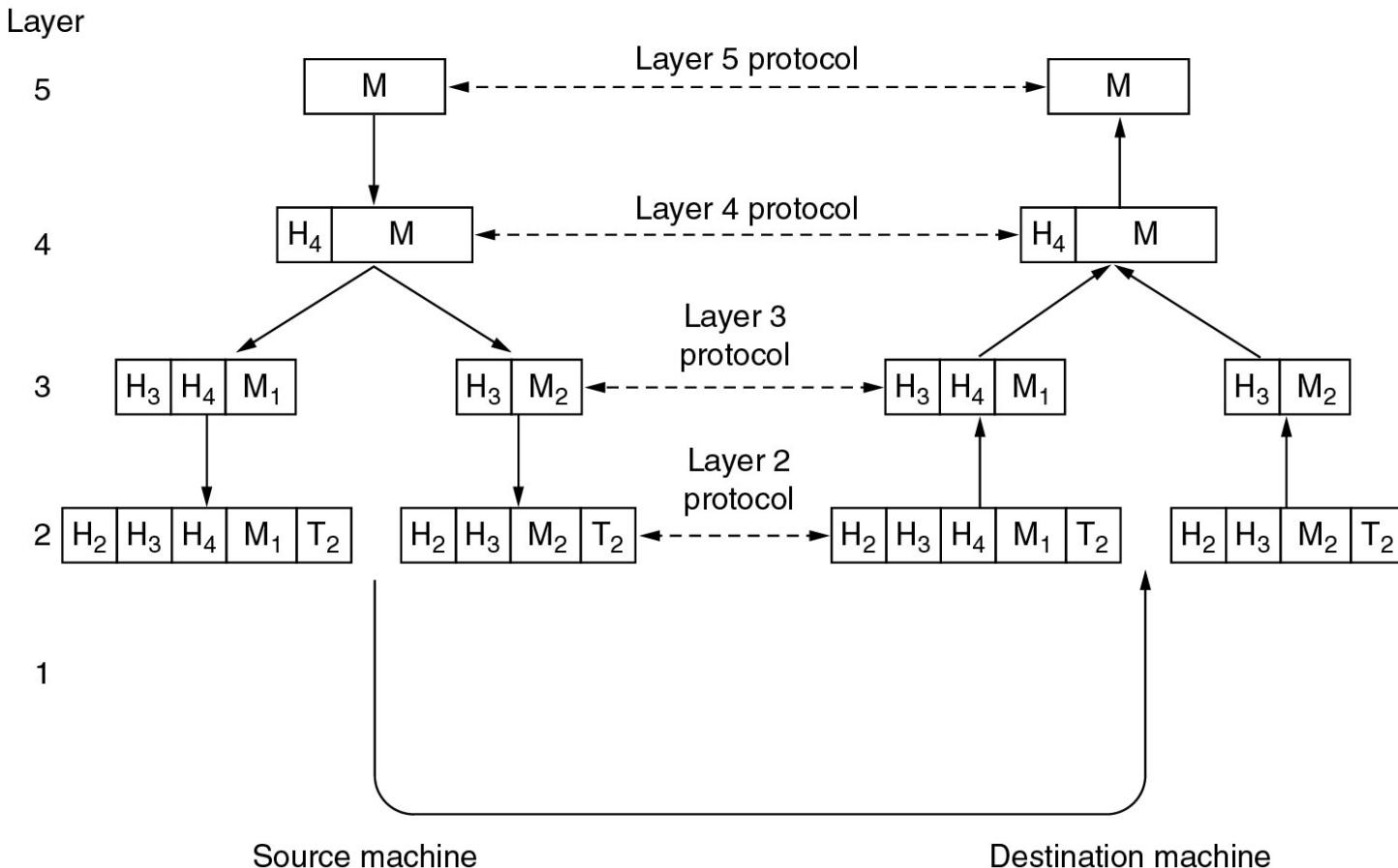
Each layer offers services to layers above it

Inter-layer exchanges are conducted according to a protocol

- Layers, protocols, and interfaces

Protocol Hierarchies (3)

- Example information flow supporting virtual communication in layer 5



Design Issues for the Layers

- Connection Oriented: connect, use, disconnect (similar to telephone service)
 - Negotiation inherent in connection setup
- Connectionless: use (similar to postal service)
- Choice of service type has a corresponding impact on the reliability and quality of the service itself

Service Primitives

- Primitives are a formal set of operations for services
- The number and type of primitives in any particular context is dependent on nature of service itself - in general more complex services require more primitives service

Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
ACCEPT	Accept an incoming connection from a peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

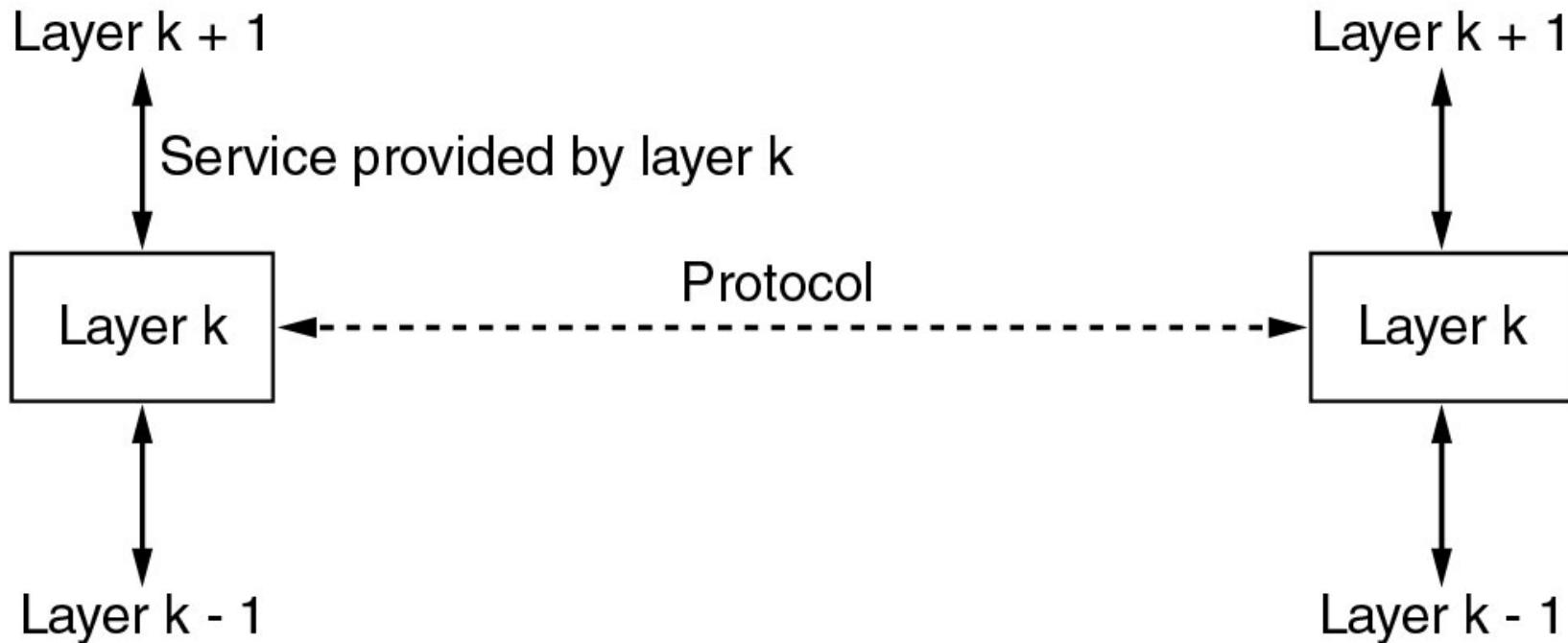
- Six service primitives for implementing a simple connection-oriented service

Relationship of Services and Protocols

- Service = set of primitives that a layer provides to a layer above it
 - Defines what operations the layer is prepared to perform on behalf of its users
 - It says nothing about how these operations are implemented
 - interfaces between layers (service provider vs service users)
- Protocol = a set of rules governing the format and meaning of packets that are exchanged by peers within a layer
 - Packets sent between peer entities

Services to Protocols Relationship

- The relationship between a service and a protocol.



Reference Models

- The OSI Reference Model
- The TCP/IP Reference Model
- A Comparison of OSI and TCP/IP
- A Critique of the OSI Model and Protocols
- A Critique of the TCP/IP Reference Model

Why do we need a network reference model?

- A reference model provides a *common baseline for the development* of many services and protocols by independent parties
- Since networks are multi-dimensional, a reference model can serve to *simplify the design process*
- It's engineering *best practice* to have an *abstract reference model*, and corresponding implementations are always required for validation purposes

OSI Reference Model

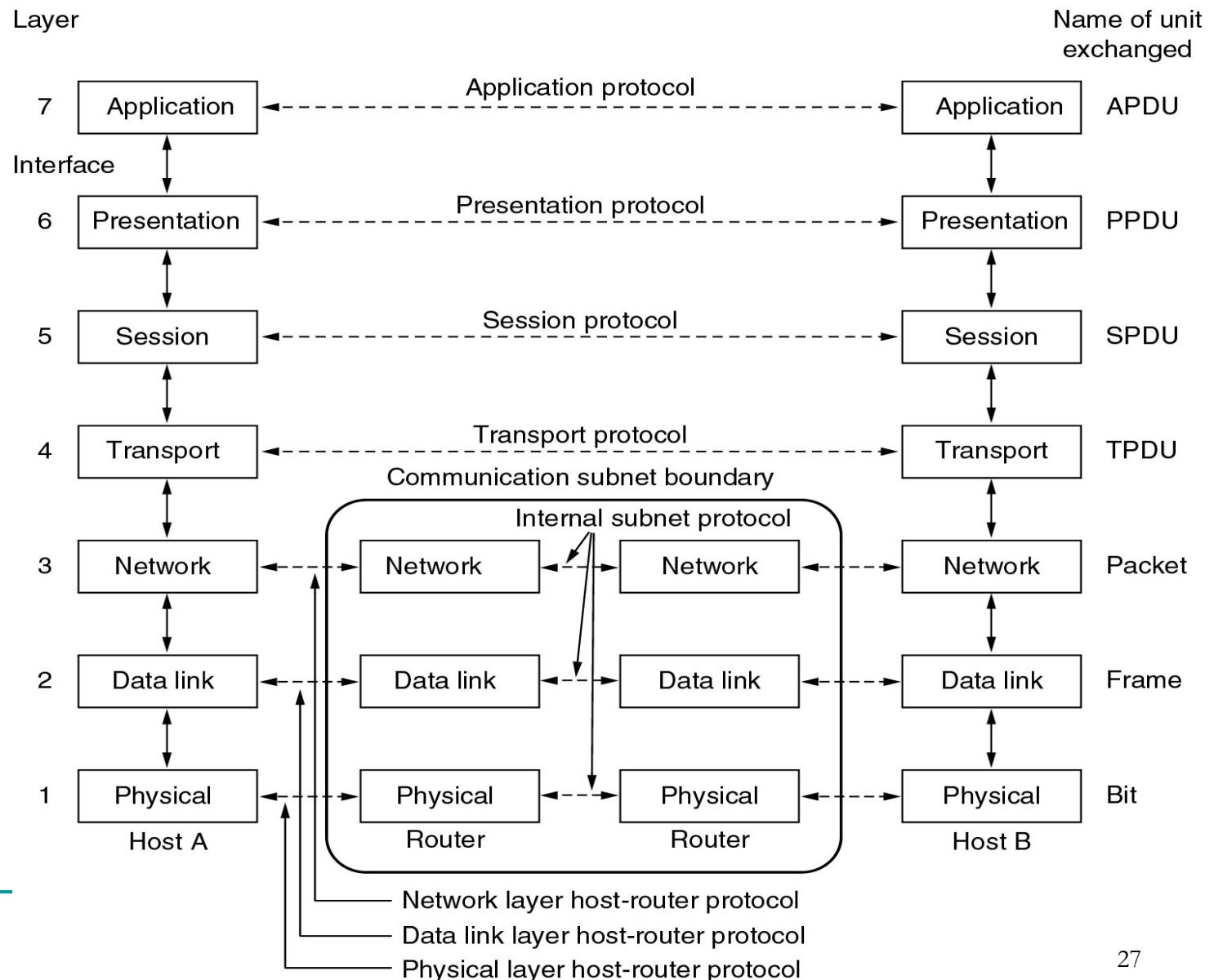
- Open Systems Interconnection (OSI)
- ISO, Day (revised 1995)
- 7 Layers
- Layer divisions based on principled decisions

OSI Layer Division Principles

1. A layer should be created where a different abstraction is needed
2. **Each layer should perform a well defined function**
3. The function of each layer should be chosen with a view toward defining internationally standardised protocols
4. The layer boundaries should be chosen to minimise the information flow across the interfaces
5. The number of layers should be large enough that distinct functions need not to be thrown together in the same layer out of necessity, and small enough that the architecture does not become unwieldy

Reference Models

The OSI reference model

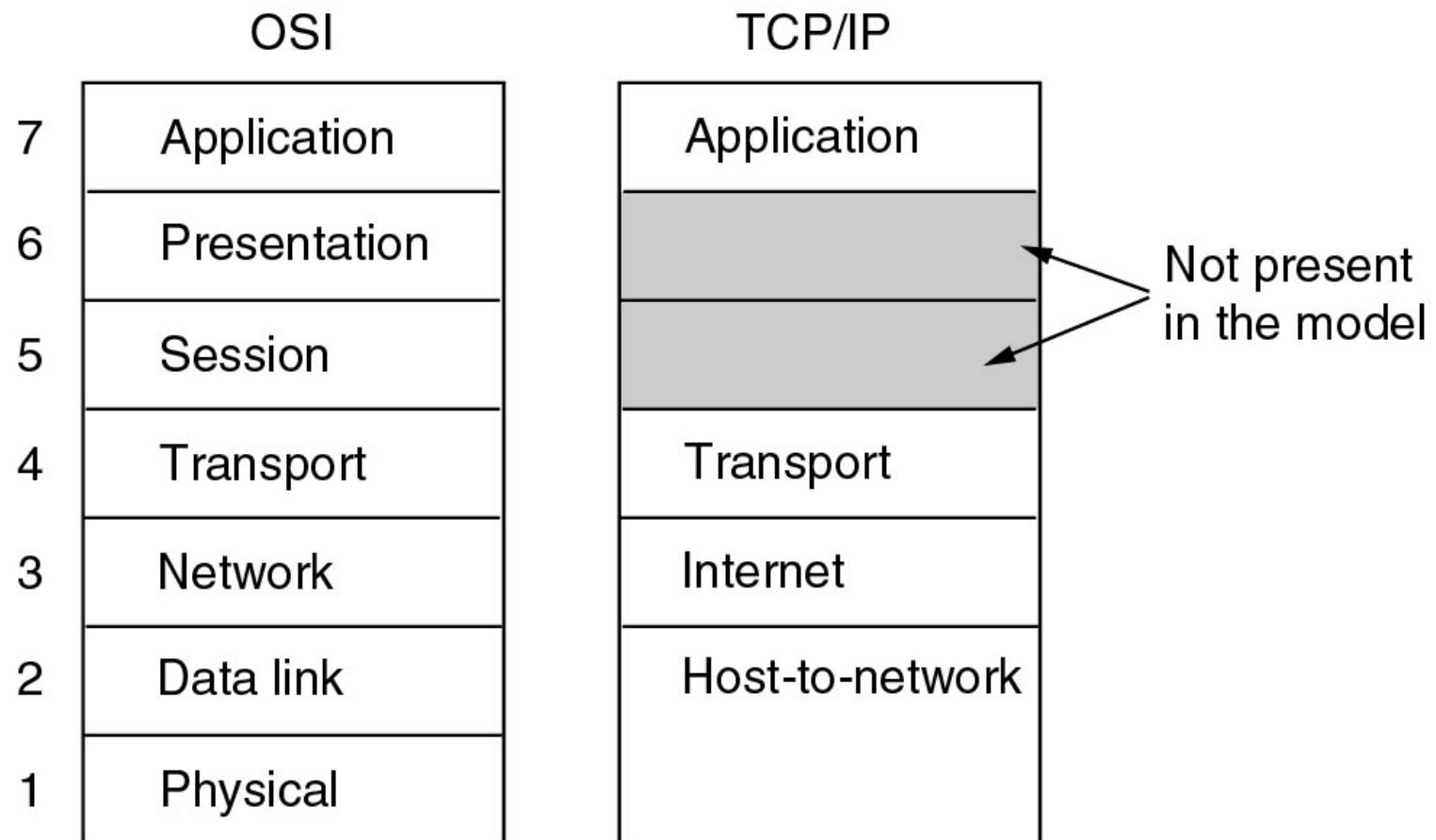


TCP/IP Reference Model

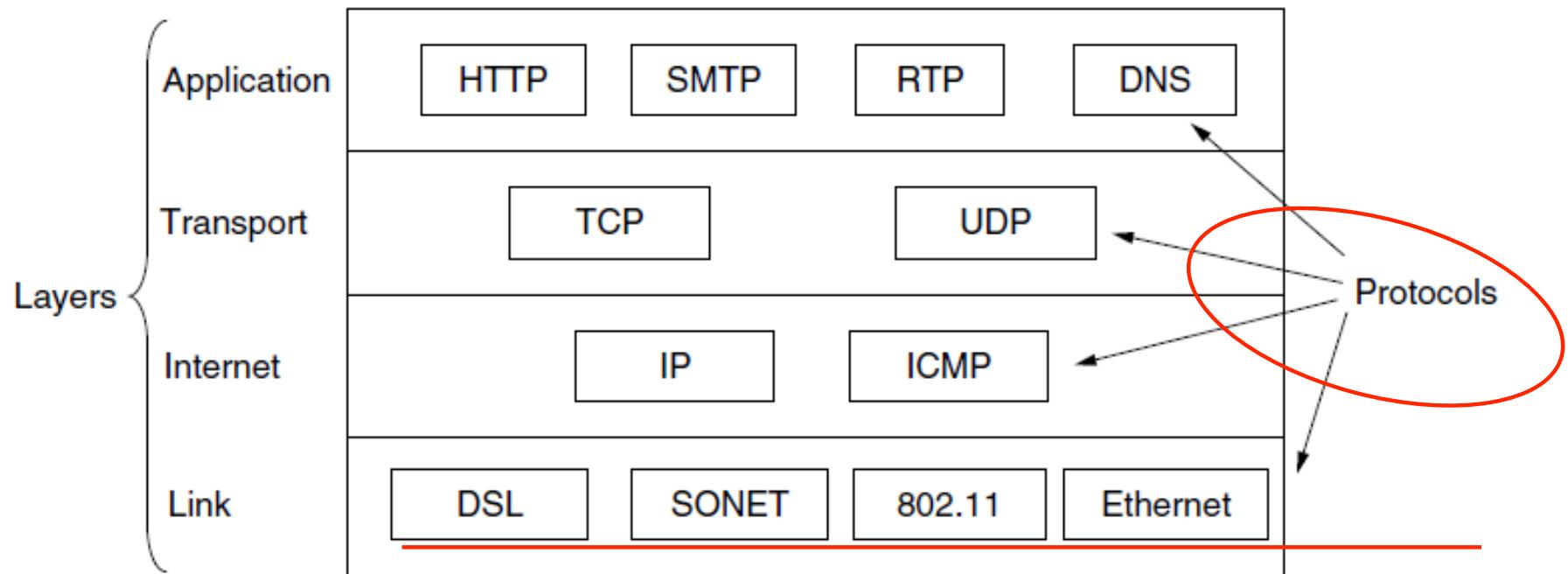
- Transmission Control Protocol/Internet Protocol
- Cerf & Kahn (1974)
- 4 layers

TCP/IP Model Illustrated

- The TCP/IP reference model.



Reference Models (3)



Comparing OSI and TCP/IP Models

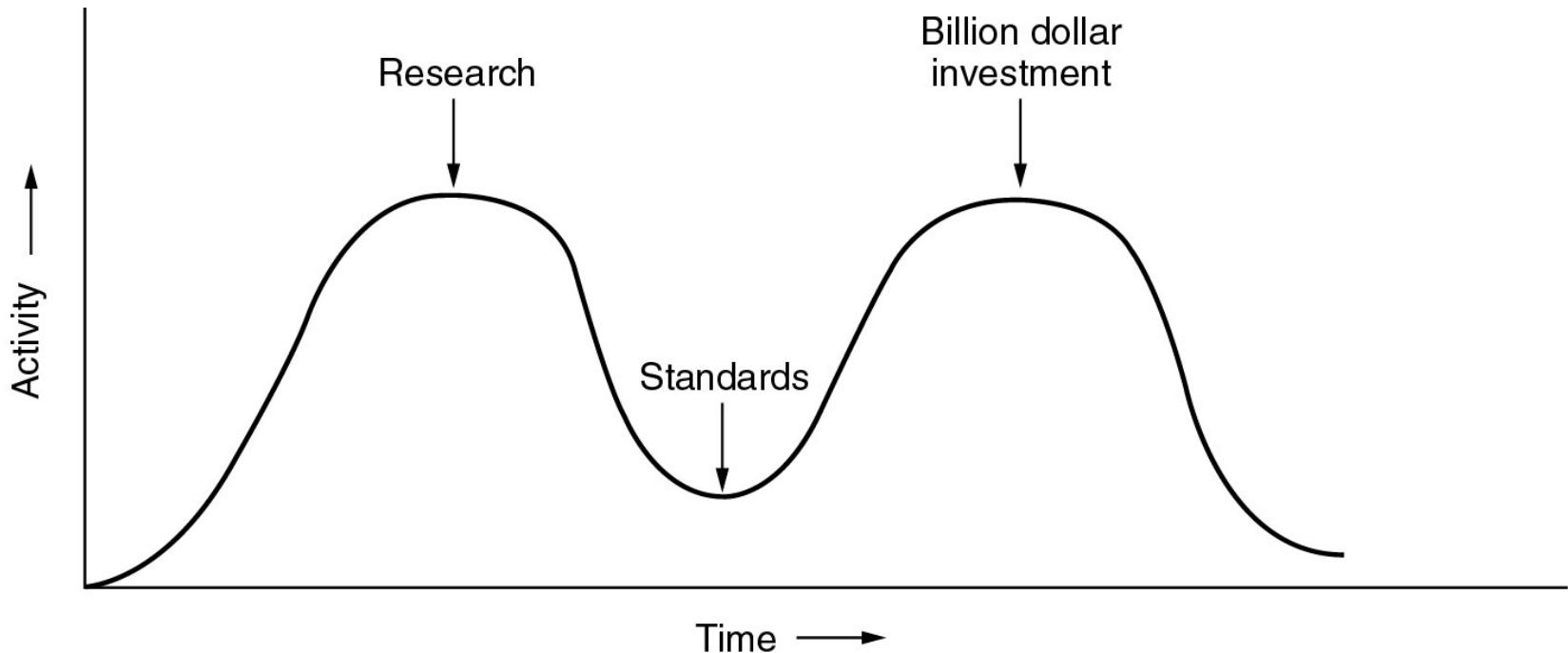
- Concepts central to the OSI model
- Services
- Interfaces
- Protocols

A Critique of the OSI Model and Protocols

- Why OSI did not take over the world?
- Bad timing
- Bad technology
- Bad implementations
- Bad politics

Bad Timing

■ The apocalypse of the two elephants



A Critique of the TCP/IP Reference Model

Problems:

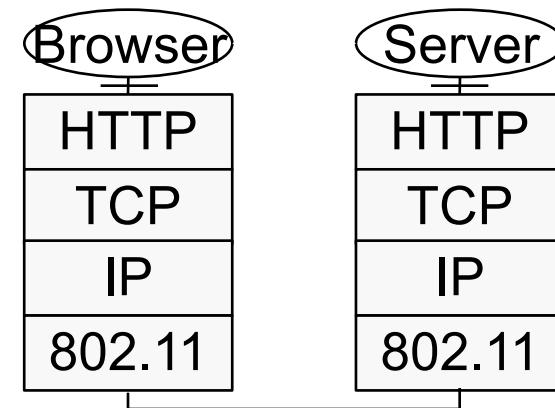
- Service, interface, and protocol not distinguished
- Not a general model
- Host-to-network “layer” not really a layer – interface between network and data link layers
- No mention of physical and data link layers
- Minor protocols deeply entrenched, hard to replace

Hybrid Model

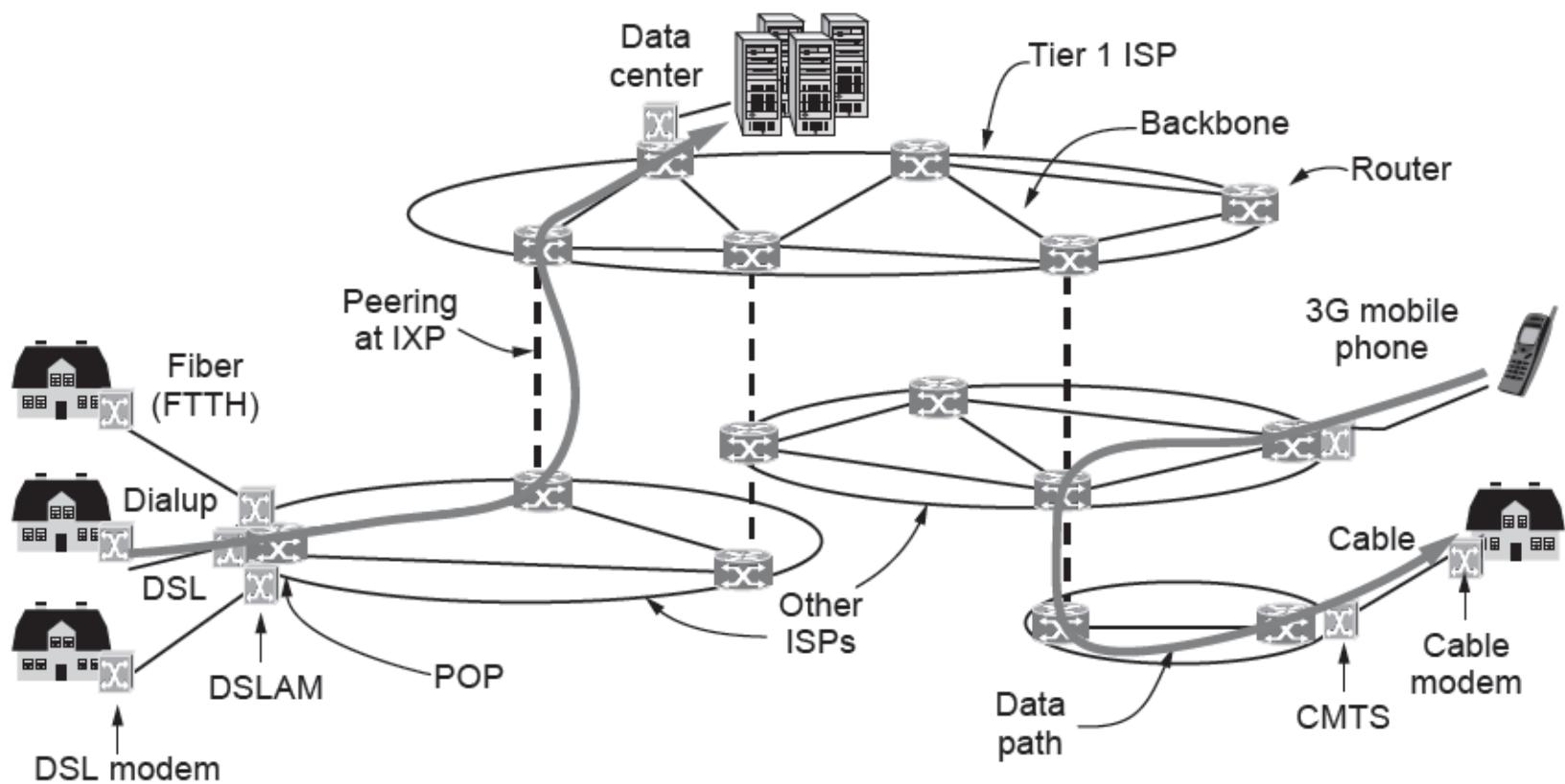
- The hybrid reference model to be used in this book. We follow this in this semester

5	Application layer
4	Transport layer
3	Network layer
2	Data link layer
1	Physical layer

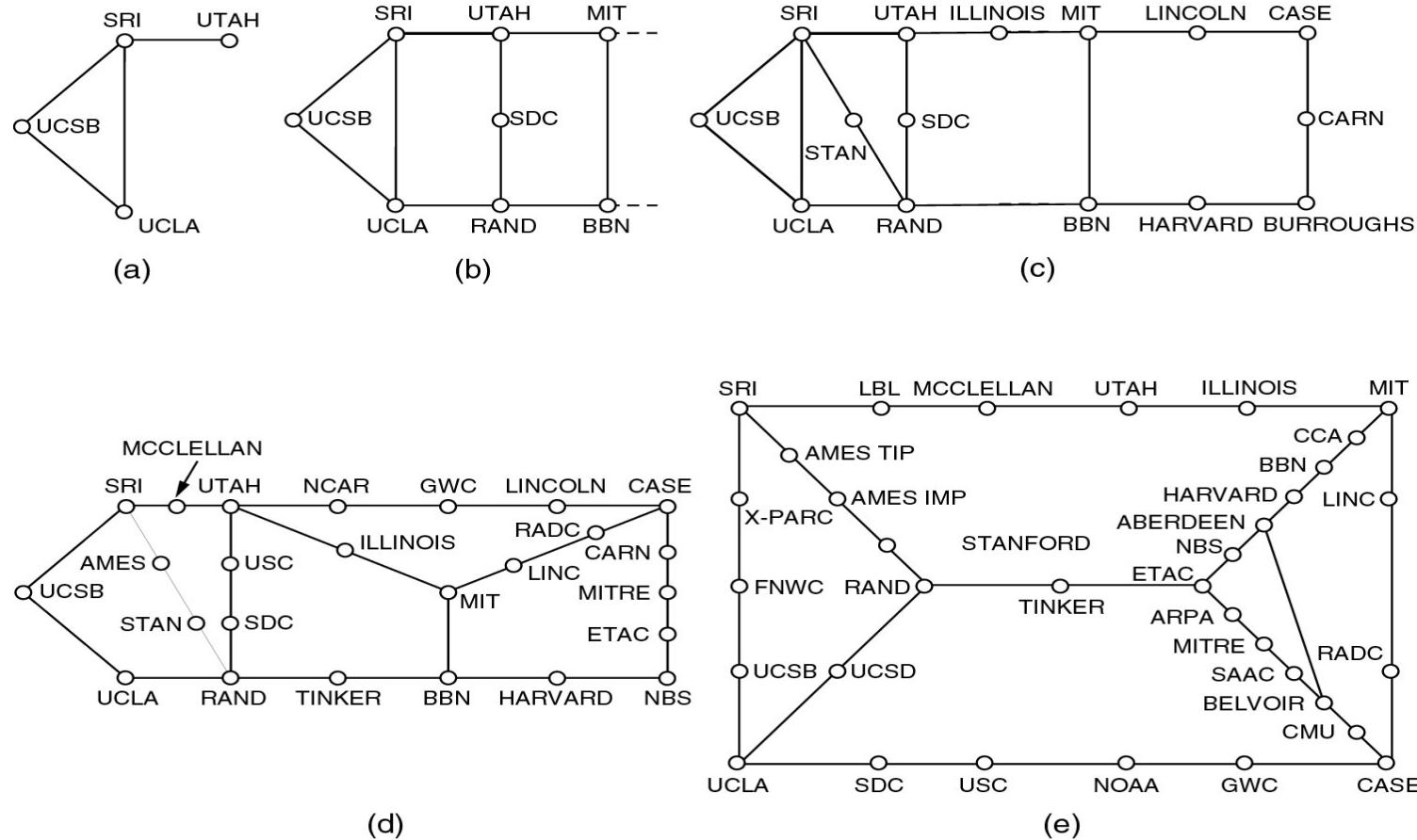
A typical network scenario



Architecture of the Internet



The ARPANET (3) Advanced Research Project Agency



- Growth of the ARPANET (a) December 1969. (b) July 1970.
- (c) March 1971. (d) April 1972. (e) September 1972.

Network Standardization

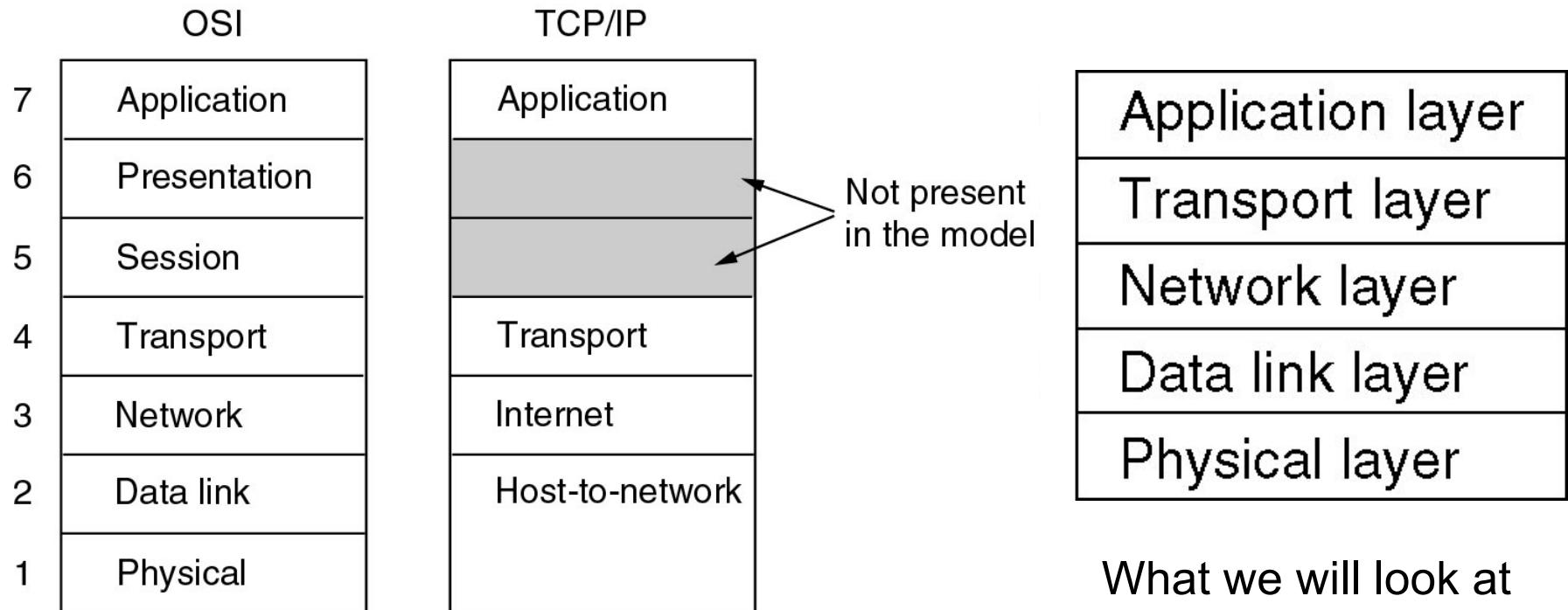
Body	Area	Examples
ITU (International Telecommunication Union)	Telecommunications	ADSL PON MPEG4
IEEE (Institute of Electrical and Electronics Engineers)	Communications	Ethernet, WiFi
IETF (Internet Engineering Task Force)	Internet	HTTP/1.1 DNS
W3C (The World Wide Web Consortium)	Web	HTML5 standard

Week 2 – Physical Layer

COMP90007
Internet Technologies

Chien Aun Chan

Review



Outline

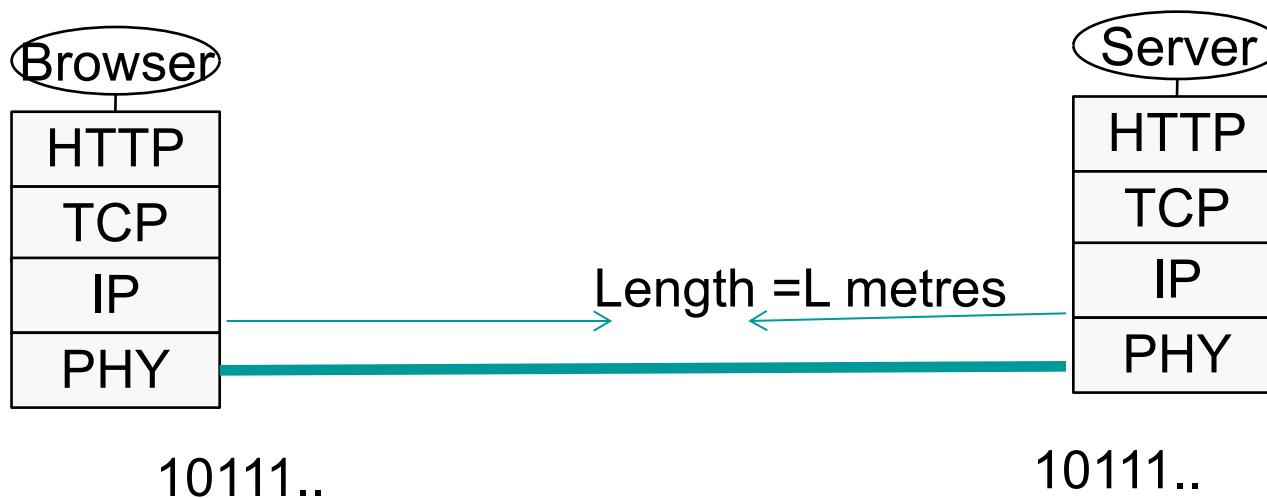
- The service
 - Link Model
 - Bandwidth and delay
- Guided and Unguided Transmission Media
 - Twisted Pair
 - Coax
 - Fibre Optics
 - Wireless Transmission
- Multiplexing
- Digital Modulation and information theory

What is the Physical Layer ?

- Recall the layer hierarchy from network reference models
 - The physical layer is the lowest Layer in OSI model
 - The physical layer's properties in TCP/IP model are in the “host-to-network” division.
- The physical layer is concerned with the mechanical, electrical and timing interfaces of the network
- Various physical media can be used to transmit data, but all of them are affected by a range of physical properties and hence have distinct differences
- How many different types of physical media can you think of?

Link Model

- Simplified Link Model: Consider the network as a connected link between computers
- We can abstract the physical channel as a link



Link Model

- *Bandwidth* is usually treated as rate of transmission in bits/second
- *Delay* (in seconds) is the time required for the first bit to travel from computer A to computer B.

Example

- We need about 1 kbit/sec to transmit voice.
- Bandwidth of single mode fibre can reach 1 Tbit/sec.
- How many voice calls can be transmitted through an Fiber Optic Cable?

$$10^{12} / 10^3 = 1 \text{ billion calls per channel}$$

 
Tbit/s kbit/s

Message Latency

- Latency is the time delay associated with sending a message over a link
- This is made of up two parts
 - **Transmission delay:**
 - $T\text{-delay} = \text{Message in bits} / \text{Rate of transmission}$
 - $= M/R$ seconds
 - **Propagation delay**
 - $P\text{-delay} = \text{length of the channel} / \text{speed of signals}$
 - $\text{Length} / \text{Speed of signal}$ ($2/3$ of speed of light for wire)
 - **Latency** = $L = M/R + P\text{-delay}$

Example -1

- A home computer is connected to an ISP server through 56 K bps modem. Assuming a frame size of 5600 bits, compute P-Delay and T-Delay for the link. Assume speed of signal = $2/3 C$ and length of the link is 5 K metres.
- $T\text{-delay} = 5600 \text{ (bits)} / 56000 \text{ (kbps)} = 100 \text{ m sec}$
- $P\text{-delay} = 5 \text{ (km)} / 200000 \text{ (km/s)} = 0.025 \text{ m sec}$
- Latency = 100.025 m sec

Example-2

- Now for the previous question, assume a countrywide optical broadband link of length 1000 kms of bandwidth 100 M bits/sec. Assuming a frame size of 5600 bits, compute P-Delay and T-Delay for the link. Assume speed of signal = C = 300000 km/sec.
- $T\text{-delay} = 5600 \text{ (bits)} / 100\ 000\ 000 \text{ (bits/s)} = 0.056 \text{ m sec}$
- $P\text{-delay} = 1000 \text{ (km)} / 300000 \text{ (km/s)} = 3.33 \text{ m sec}$
- $\text{Latency} = 3.386 \text{ m sec}$

The Bandwidth Revolution?

- Evolutionary steps in available bandwidth:
 - CPU speeds increase by a factor of ~20 per decade
 - 1981: PC 4.77Mhz vs 2001: PC 2 Ghz
- Bandwidth speeds increase by a factor of ~125 per decade (1981: Modem 56kbps vs 2001: Net 1Gbps)
- Current CPU speed now approaching physical limits - constrained by physical properties pertaining to granularity of engraving on silicon
- Current bandwidth available up to 50Tbps - vastly exceeding the rate at which we can convert electrical impulses to optical pulses

Outline

- The service
 - Link Model
 - Bandwidth and delay
- Transmission Media
 - Twisted Pair
 - Coax
 - Fibre Optics
 - Wireless Transmission
- Multiplexing
- Digital Modulation and information theory

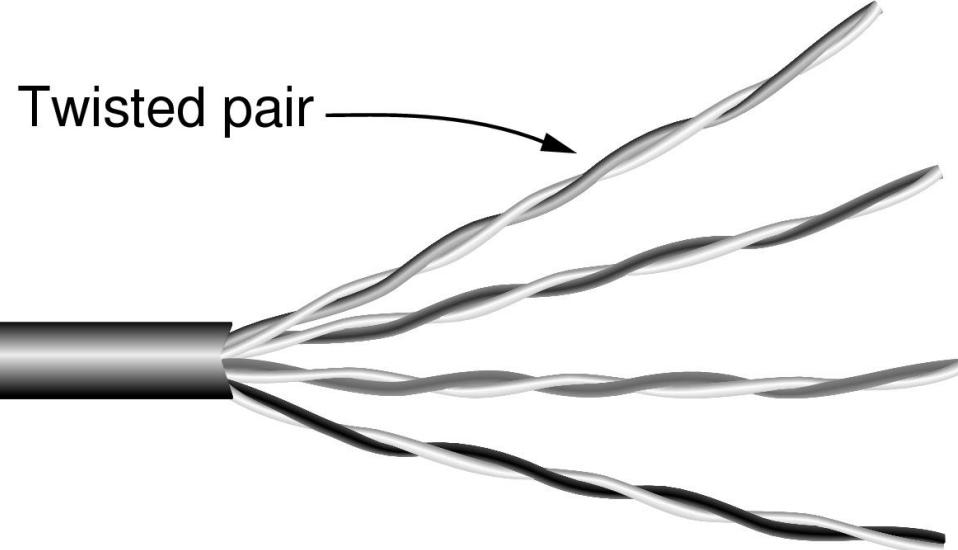
Signal Attenuation

- How far and how much data a medium can carry has a lot to do with signal attenuation:
 - “**Attenuation** is the loss or reduction in the amplitude (strength) of a signal as it passes through a medium.”

Wires – Twisted Pair

- Two insulated copper wires, twisted in helical (DNA) form.
- Twisting reduces radiance of waves from effectively parallel antennae
- Distance up to <5km, repeaters can extend this distance (large buildings often have km's of cabling)
- twisting reduces interference

Category 5 UTP
cable with four
twisted pairs



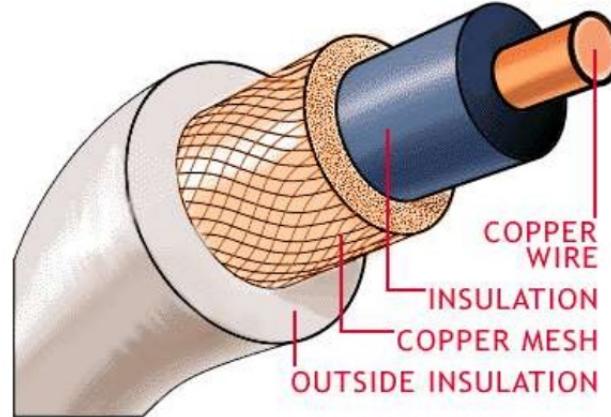
Properties and Types of Twisted Pair

- ❑ Bandwidth dependent on distance, wire quality/density
- ❑ Cat 3 - 2 wires, 4 pairs in sheath, 16Mhz
- ❑ Cat 5 - 2 wires, 4 pair in sheath, more twists = less interference, higher quality over longer distance, 100 Mhz
- ❑ Cat 6 - 250 Mhz
- ❑ Cat 7 - 600Mhz + ?

Coaxial Cable (“Co-ax”)

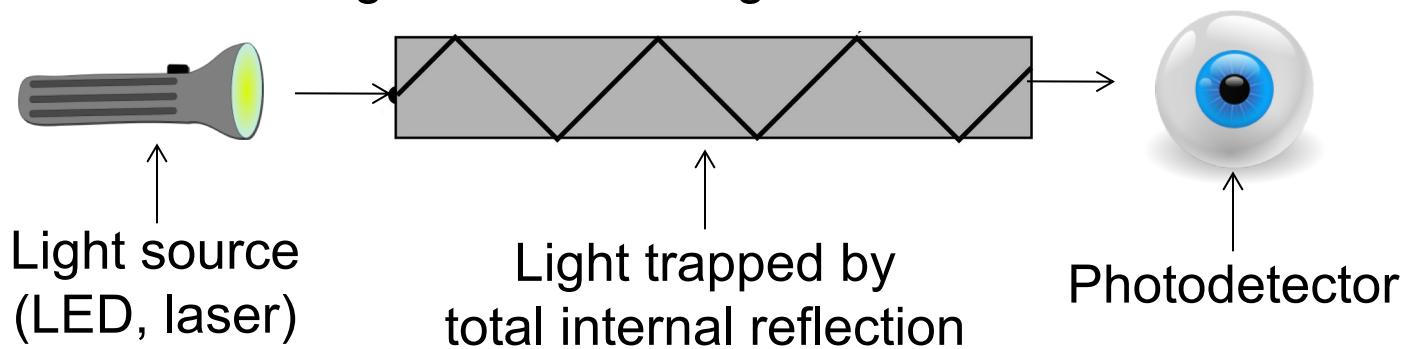
- Better shielding than twisted pair = higher speeds over greater distances
- Copper core with insulation, mesh, and sheath
- Bandwidth approaches 1Ghz
- Still widely used for cable TV/Internet

A diagram of a coaxial cable



Fiber Optics

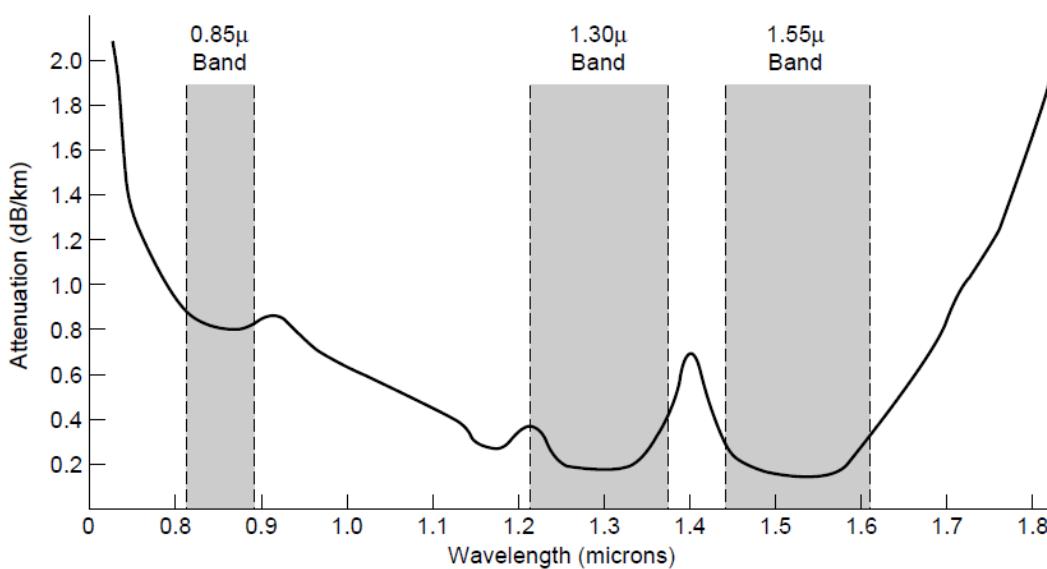
- ❑ Optical transmission has 3 components: light source, transmission medium, detector
- ❑ Semantics: light = 1, no light = 0 (basic binary system)
- ❑ Data transmission over a fibre of glass
- ❑ A detector generates electrical pulse when light hits it
- ❑ Refraction between air/silica boundary is compensated for by design - total internal reflection
- ❑ Common for high rates and long distances



Transmission of Light Through Fibre

Fiber has enormous bandwidth (THz) and tiny signal loss – hence high rates over long distances

Attenuation (loss per km) of light through glass depends on wavelength of light



Optical communications at 0.85, 1.30, 1.55 microns;

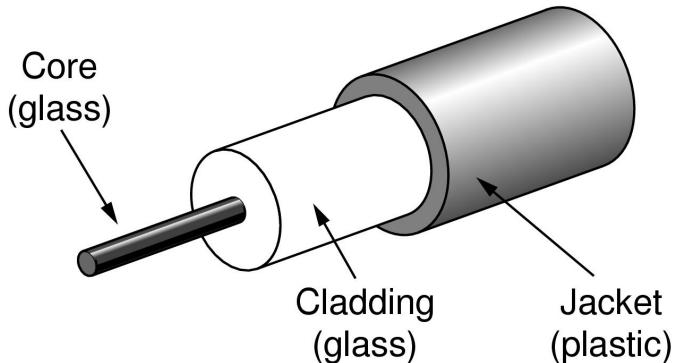
1.30 and 1.55 have low loss <5%/km)

0.85 physical property sharing between laser and electronics

Fiber Optic Cables #1

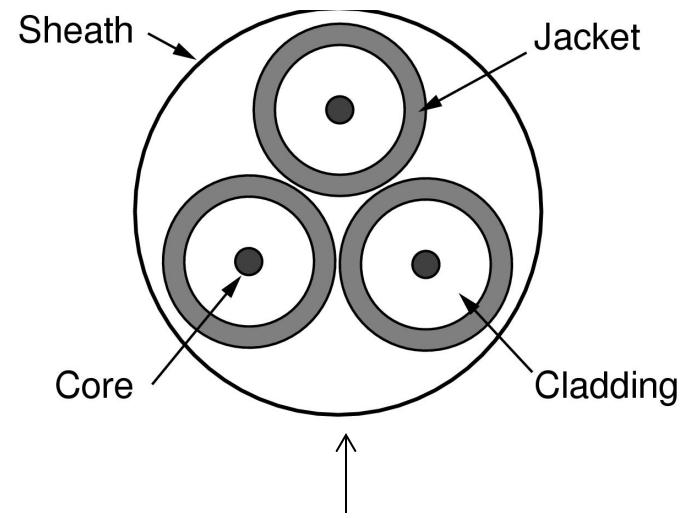
Single-mode

- ❑ Core so narrow (10um) light can't even bounce around
- ❑ Used with lasers for long distances, e.g., 100km



Multi-mode

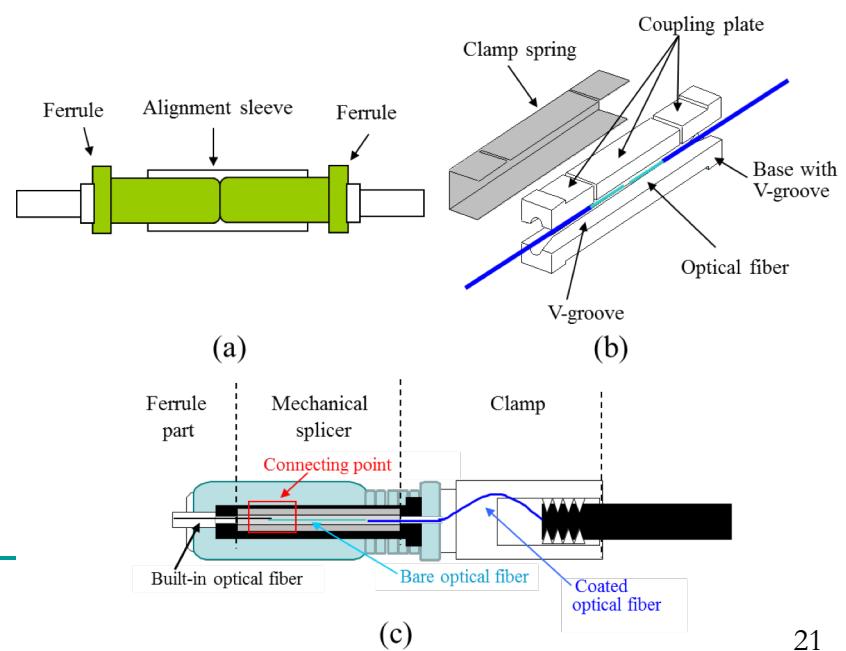
- ❑ Other main type of fiber
- ❑ Light can bounce (50um core)
- ❑ Used with LEDs for cheaper, shorter distance links



Fibers in a cable

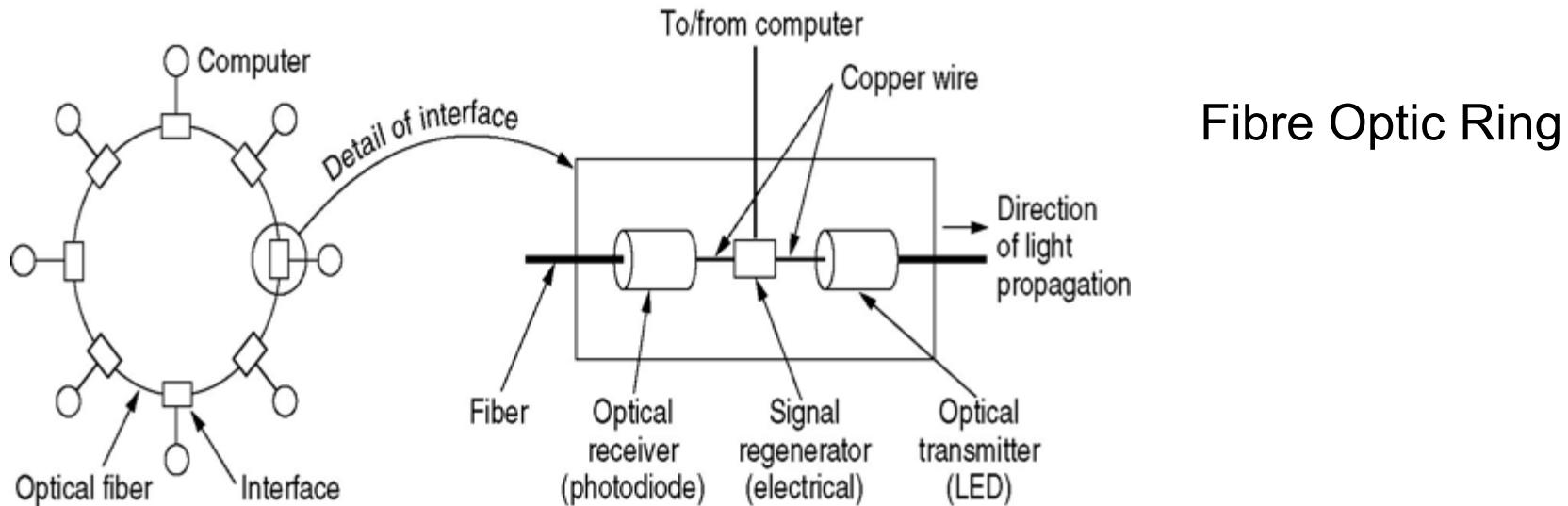
Fiber Optic Connections

- Connectors and Fiber Sockets (10-20% loss, but easy to configure)
- Mechanical Splice (10% loss, labour intensive)
- Fusion (<1% loss, but specialised)
- Signalling using LED's or semiconductor lasers



Fiber Optic Networks

- Fiber optic cable is a scalable network media - LAN, WAN, long haul
- Fibre optic cable can be considered either as a ring or as a bus network type (series of point to point connections)



Comparison: Wires and Fiber

Comparison of the properties of wires and fiber:

Property	Wires	Fiber
Distance	Short (100s of m)	Long (tens of km)
Bandwidth	Moderate	Very High
Cost	Inexpensive	More Expensive
Convenience	Easy to use	Harder to use
Security	Easy to tap	Hard to tap

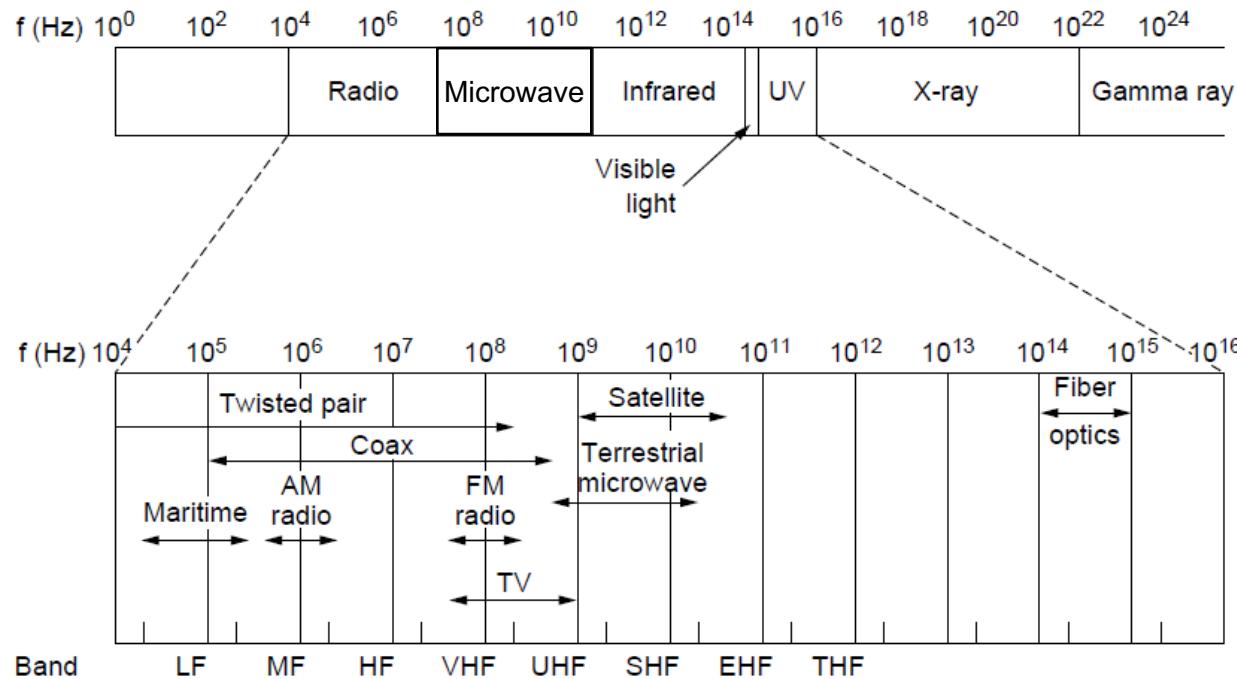
Wireless Transmission

- Mobile users requires a mobility enabled network - contrast with the wired networks
- Wireless networks can provide advantages even in fixed location environments
- There are many types of wireless data transmission networks, but they all have a common basis - radio wave propagation
- Unlike previous mediums wireless signals are broadcasted over a region
- Potential signal collisions – Need regulations

Electromagnetic Spectrum (1)

Different bands have different uses:

- Radio: wide-area broadcast;
- Infrared/Light: line-of-sight
- Microwave: LANs and 3G/4G;



Wireless vs. Wires/Fiber

Wireless:

- + Easy and inexpensive to deploy
- + Naturally supports mobility
- + Naturally supports broadcast
- Transmissions interfere and must be managed
- Signal strengths hence data rates vary greatly

Wires/Fiber:

- + Easy to engineer a fixed data rate over point-to-point links
- Can be expensive to deploy, esp. over distances
- Doesn't readily support mobility or broadcast

Communication Satellites

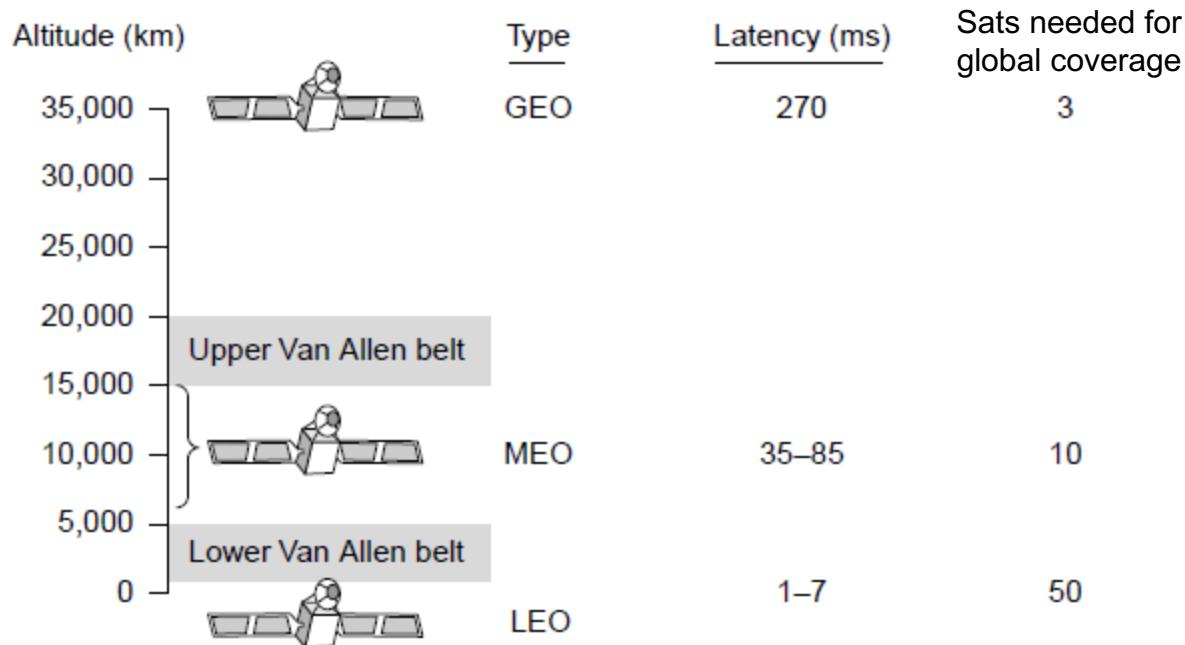
Satellites are effective for broadcast distribution and anywhere/anytime communications

- Kinds of Satellites
- Geostationary (GEO) Satellites
- Medium-Earth Orbit (MEO) Satellites
- Low-Earth Orbit (LEO) Satellites
- Satellites vs. Fiber

Kinds of Satellites

Satellites and their properties vary by altitude:

- Geostationary (GEO), Medium-Earth Orbit



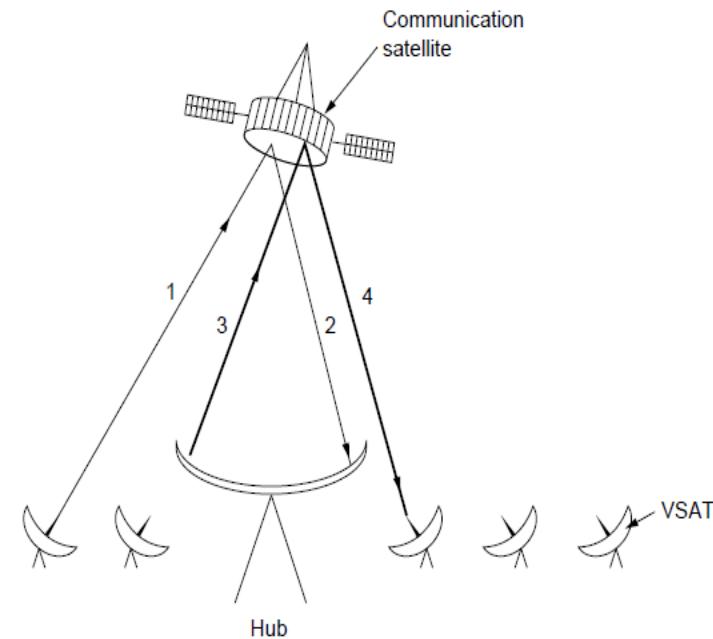
Geostationary Satellites

GEO satellites orbit 35,000 km above a fixed location

- VSAT (computers) can communicate with the help of a hub
- Different bands (L, S, C, Ku, Ka) in the GHz are in use but may be crowded or susceptible to rain.

GEO satellite

VSAT
(Very Small Aperture Terminals)



Low-Earth Orbit Satellites

Systems such as Iridium use many low-latency satellites for coverage and route communications via them



Each satellite has
four neighbors

The Iridium satellites form six
necklaces around the earth.

Satellite vs. Fiber

Satellite:

- + Can rapidly set up anywhere/anytime communications (after satellites have been launched)
- + Can broadcast to large regions
- Limited bandwidth and interference to manage

Fiber:

- + Enormous bandwidth over long distances
- Installation can be more expensive/difficult

Link Terminology

Full-duplex link

- ❑ Used for transmission in both directions at once
 - ❑ e.g., use different twisted pairs for each direction

Half-duplex link

- ❑ Both directions, but not at the same time
 - ❑ e.g., senders take turns on a wireless channel

Simplex link

- ❑ Only one fixed direction at all times; not common

Outline

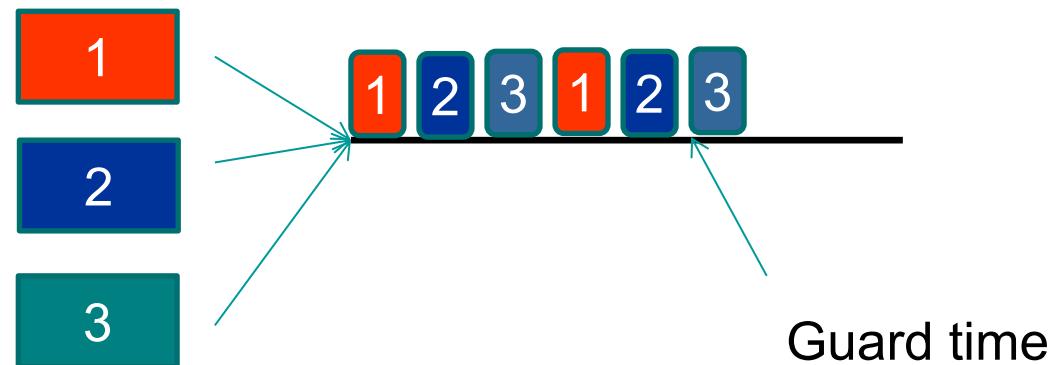
- The service
 - Link Model
 - Bandwidth and delay
- Transmission Media
 - Twisted Pair
 - Coax
 - Fibre Optics
 - Wireless Transmission
- Multiplexing
- Digital Modulation and information theory

Multiplexing

- When multiple sources want to access the medium
 - Time Division Multiplexing
 - Frequency Division Multiplexing
 - Statistical Multiplexing (for curious readers)
 - Code Division Multiple Access

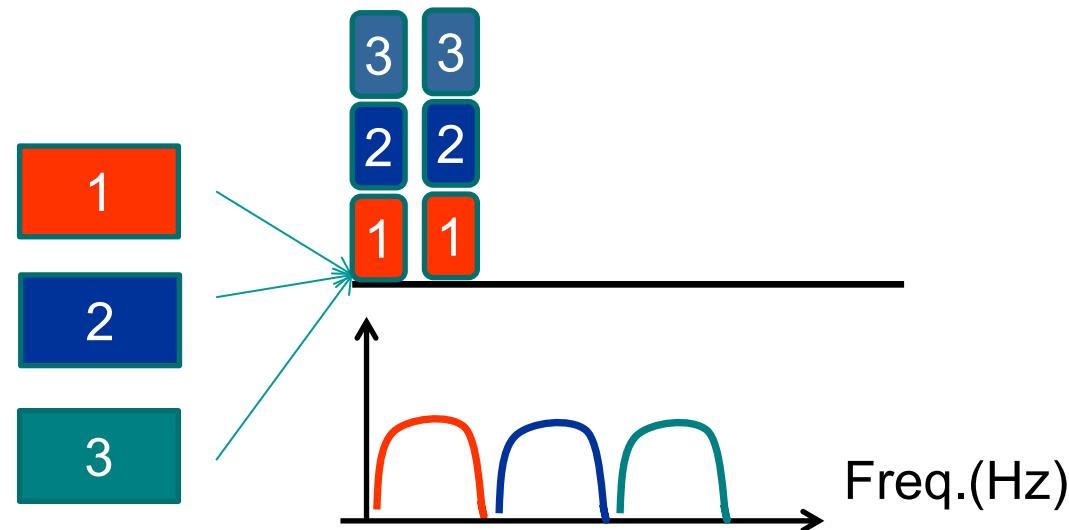
Time Division Multiplexing

- Users can send according to a fixed schedule
- Slotted access to the full speed of the media



Frequency Division Multiplexing

- Users can only use specific frequencies to send their data
- Continuous access with lower speed

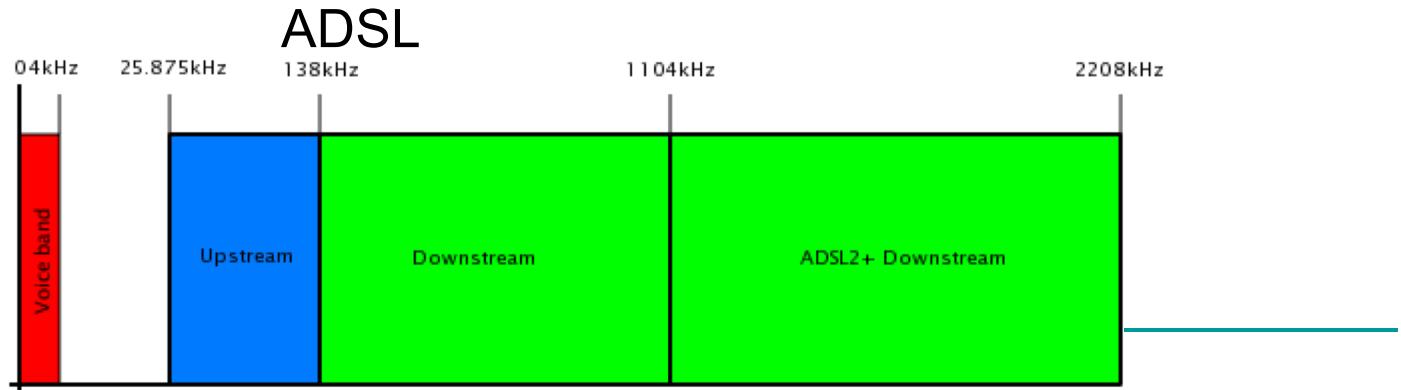


Digital Modulation

Modulation schemes send bits as signals

- ❑ Baseband Transmission
 - Signal that run from 0 up to a maximum frequency
 - E.g., Telephone system: $0 \sim 4\text{kHz}$
- ❑ Passband Transmission
 - Signals that are shifted to occupy a higher range of frequencies

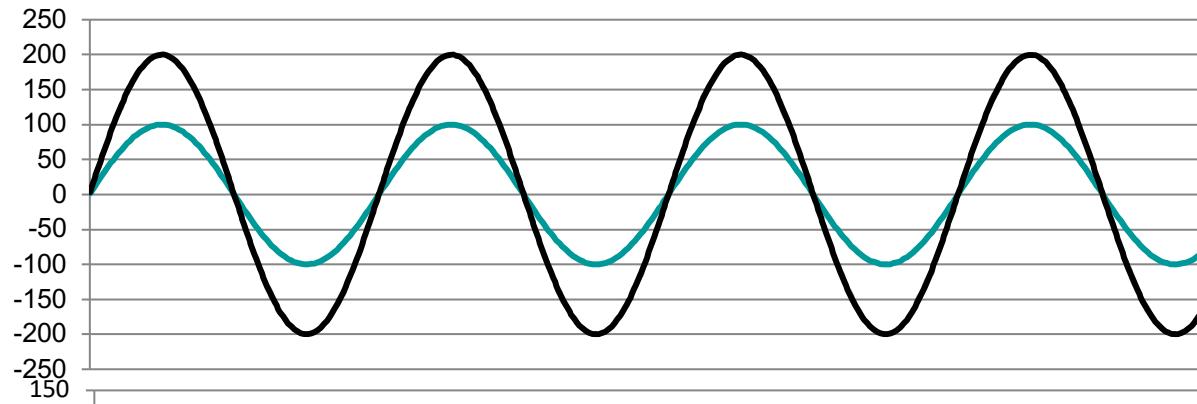
Example:



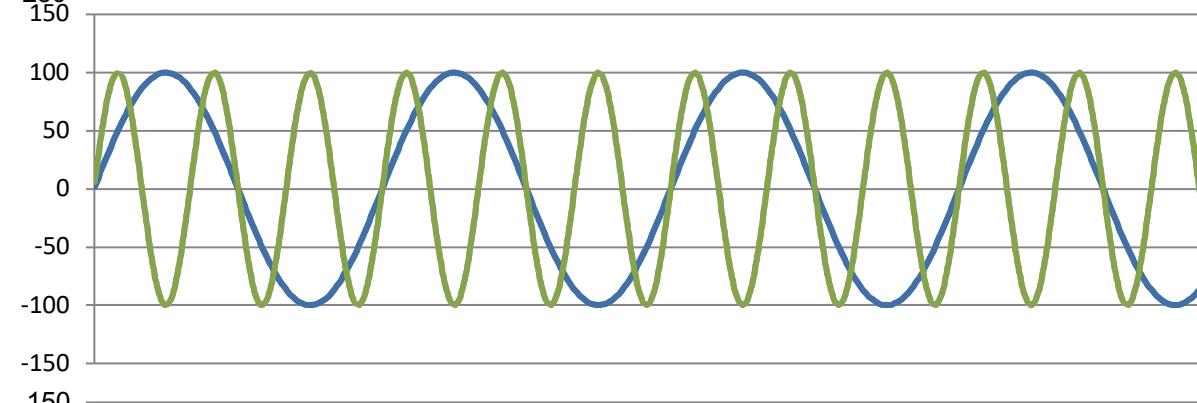
Electrical Analogue Signal to Digital

- Information on wire transmitted by variance of a physical property eg voltage, current.
 - Generating a periodic function, imagine a Sine function
- Sine function: $c \cdot \sin(ax+b)$: Three things can change the behaviour of the function:
 - C: Amplitude, A:Frequency and B:Phase

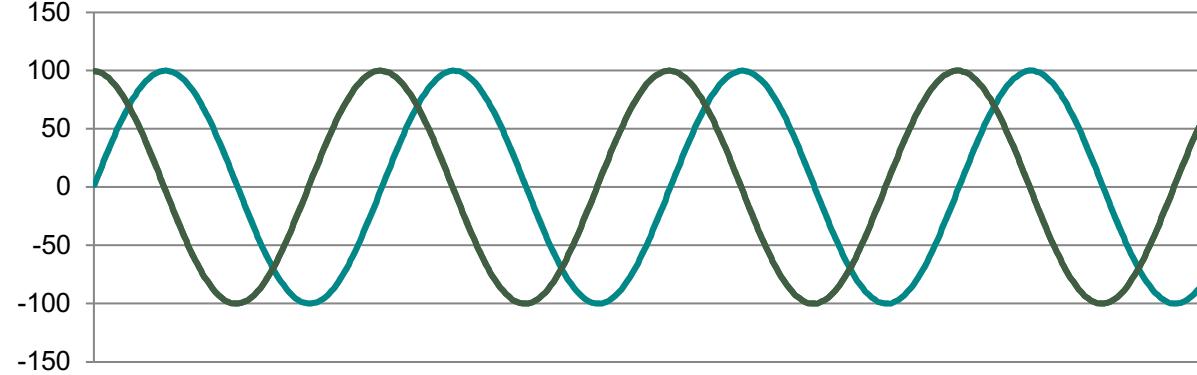
Change in Amplitude, Frequency, & Phase



Change in
Amplitude



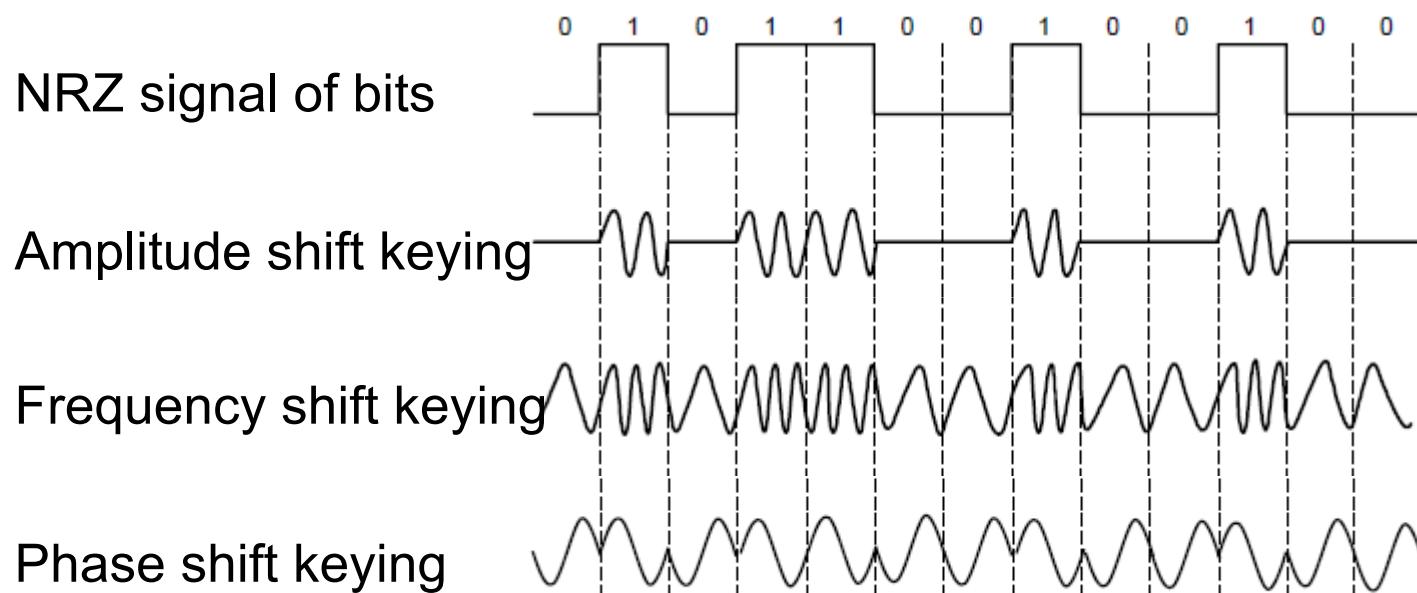
Change in
Frequency



Change in
Phase

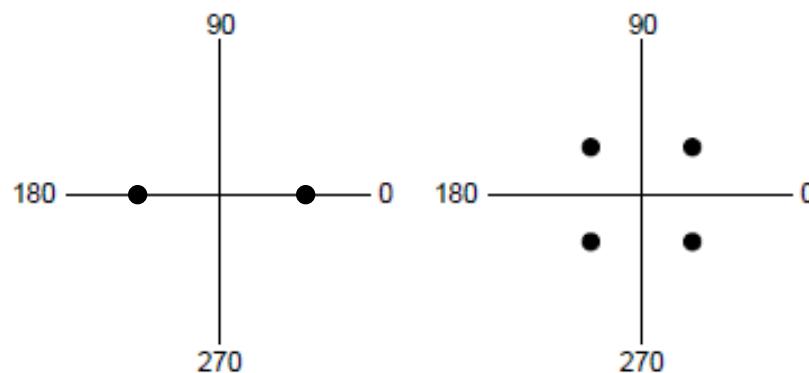
Modulation Types

Modulating the amplitude, frequency/phase of a carrier signal sends bits in a (non-zero) frequency range



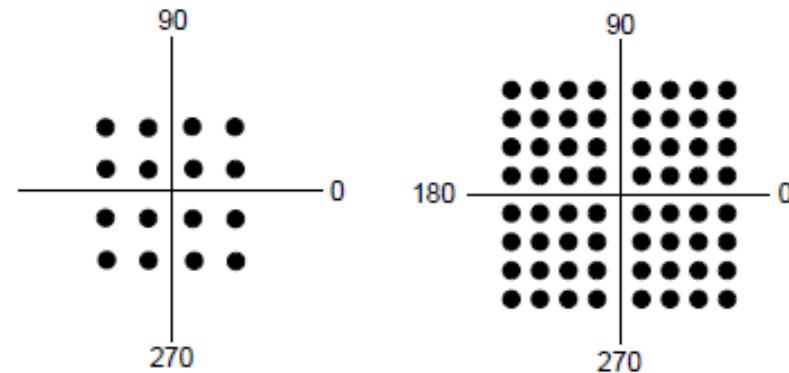
Passband Transmission

Constellation diagrams are a shorthand to capture the amplitude and phase modulations of symbols:



BPSK
2 symbols
1 bit/symbol

QPSK
4 symbols
2 bits/symbol



QAM-16
16 symbols
4 bits/symbol

QAM-64
64 symbols
6 bits/symbol

BPSK/QPSK varies only phase

QAM varies amplitude and phase

Harry Nyquist



- Early theoretical work on determining fundamental limits for the bandwidth required for communication-heralded digital revolution

A Brief Introduction to Data Communications Theory

- How to transform continuous signals into digital values?
 - Sampling the amplitude values of the signal
- Symbol Rate (Baud Rate): number of signal changes per second

Maximum Data Rate of a Channel

- Nyquist's theorem relates the data rate to the bandwidth (B) and number of signal levels (V) (channel without noise):

$$\text{Max. data rate} = 2B \log_2 V \text{ bits/sec}$$

- If a signal has bandwidth B, then the signal can be fully reconstructed by sampling with $2B$ rate.
- If signal has V levels, each symbol can be represented by $\log_2 V$ bits.

Claude Shannon



- Father of Information theory.
- 1948 monograph – "The mathematical theory of communication" defined a new area;
- 1949 monograph- "Communication Theory of Secrecy Systems" is another foundational work on modern Cryptography

Maximum Data Rate of a Channel

- Shannon's theorem relates the data rate to the bandwidth (B) and signal strength (S) relative to the noise (N):

$$\text{Max. data rate} = B \log_2(1 + S/N) \text{ bits/sec}$$

↑ ↑
How fast signal How many levels
can change can be seen

Example 1

Q: Given the signal-to-noise ratio (SNR) of 20 dB, and the bandwidth of 4kHz (telephone communications), what is the maximum data rate according to Shannon's theorem?

Ans:

$C = 4000 \log_2(1 + 100) = 4000 \log_2 (101) = 26.63$ kbit/s. Note that the value of S/N = 100 is equivalent to the SNR of 20 dB

Example 2

Q: If a binary signal is sent over a 3-kHz channel whose signal-to-noise ratio is 20 dB, what is the maximum achievable data rate?

Ans:

SNR of 20 dB = S/N = 100.

Since $\log_2(101)$ is about 6.658, the Shannon limit is about 19.975 kbps but the Nyquist limit is 6 kbps.

The bottleneck is therefore the Nyquist limit, giving a maximum channel capacity of 6 kbps

Summary

- Bandwidth & delay
- Message Latency = Propagation + Transmission delay
- Wired and Wireless Mediums, complications of wireless and attenuations of the signal
 - For different applications, what type of network is more suitable?
- Full-duplex vs Half-duplex vs Simplex
- Bit representation in the physical layer
- Multiplexing - multiple access to shared medium
- Nyquist's Theorem
- Shannon's Theorem

Week 3 – Data Link Layer

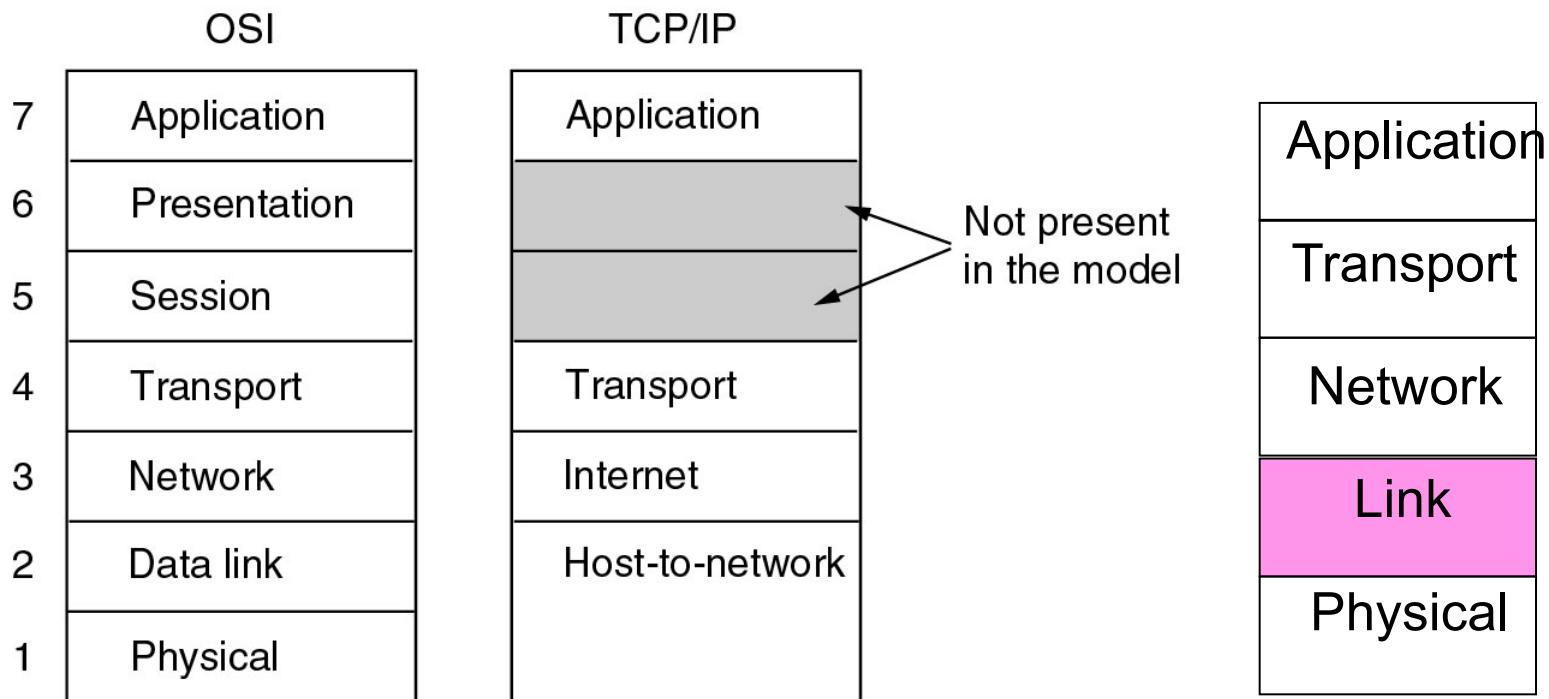
COMP90007
Internet Technologies

Chien Aun Chan
Rahul Sharma

Outline

- Data Link Layer Design Issues
 - Units
 - Services
 - Framing
- Error Detection & Correction
- Data Transmission

The Data Link Layer in OSI and TCP/IP



- Reliable, efficient communication of “frames” between two adjacent machines.
- Handles transmission errors and flow control.

Functions & Methods of the Data Layer

■ Functions of the data link layer:

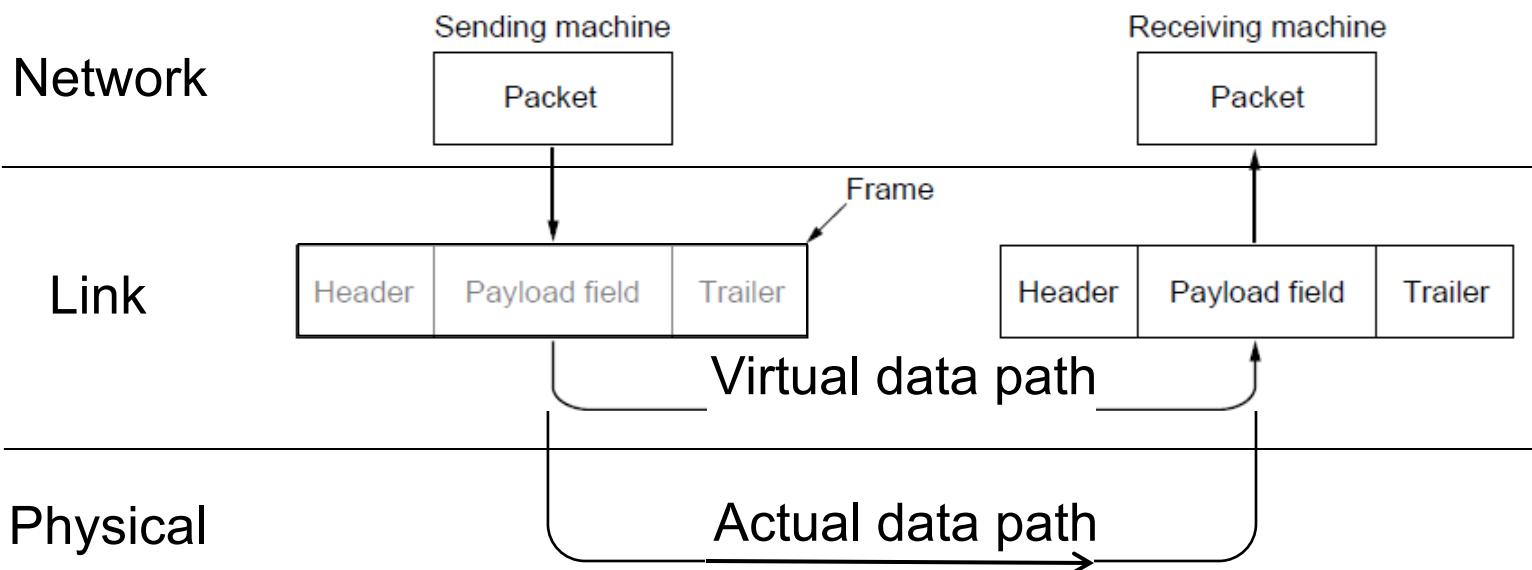
1. Provide a well-defined service interface to network layer
2. Handling transmission errors
3. Data flow regulation

■ Primary method:

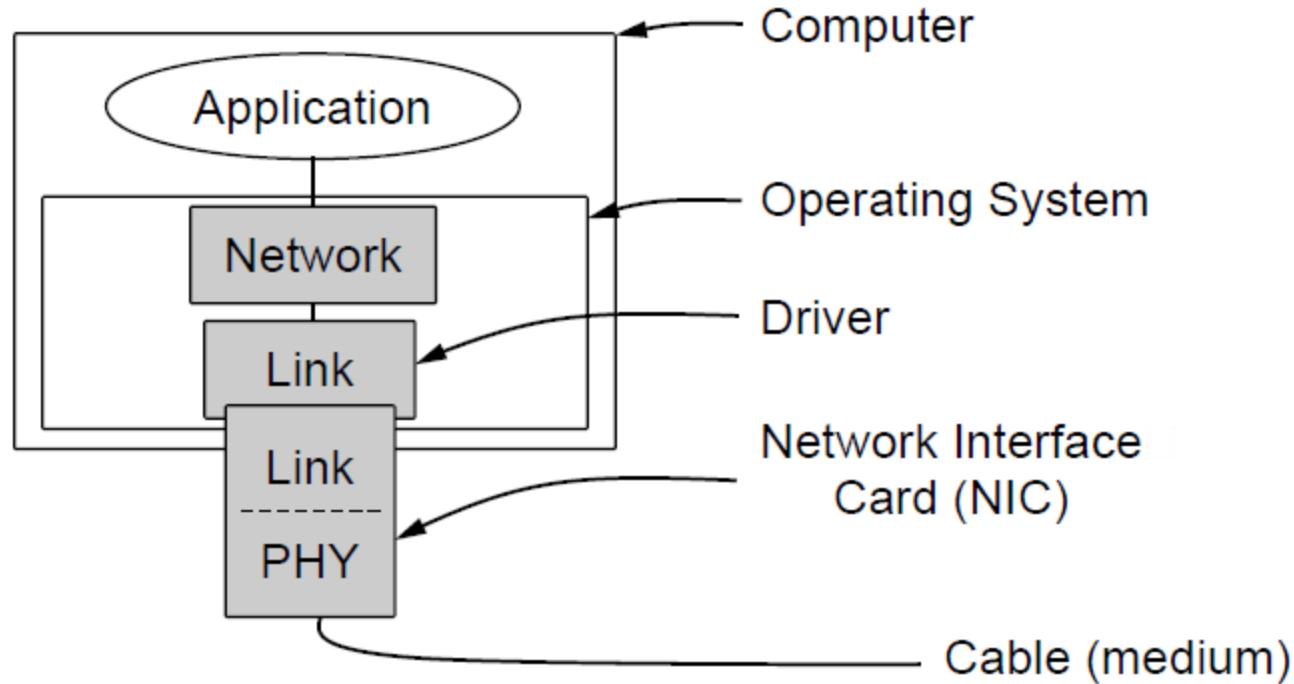
- Take packets from network layer, and encapsulate them into frames (containing a header, a payload, a trailer)

Relation Between Packets and Frames

Link layer accepts packets from the network layer, and encapsulates them into frames that it sends using the physical layer; reception is the opposite process



Typical Implementation



Outline

- Data Link Layer Design Issues
 - Units
 - Services
 - Framing
- Error Detection & Correction
- Data Transmission

Type of Services

- Connection-Oriented vs Connectionless:
Whether a connection is setup before sending a message
- Acknowledged vs Unacknowledged:
Whether the service provider give the service user an acknowledgement upon delivering the message

Services Provided to Network Layer

- Principal concern is transferring data from network layer on source host to network layer on destination host
- Services provided:
 - Unacknowledged connectionless service
 - Acknowledged connectionless service
 - Acknowledged connection-oriented service

Unacknowledged Connectionless Service

- Source host transmits independent frames to recipient host with no acknowledgement
- No logical connection establishment or release
- No lost frame recovery mechanism (or left to higher levels)
- E.g. Ethernet LANs (No logical connection is established beforehand or released afterward)
- Real-time traffic, e.g., voice

Acknowledged Connectionless Service

- Source host transmits independent frames to recipient host with acknowledgement
- No logical connection establishment or release
- Each frame individually acknowledged (retransmission if lost or errors)
- E.g. Wireless – IEEE 802.11 WiFi

Acknowledged Connection-Oriented Service

- Source host transmits independent frames to recipient host after connection establishment and with acknowledgement
- Connection established and released (communicate rate and details of message)
- Frames numbered, counted, acknowledged with logical order enforced
- For long, unreliable links such as satellite channel or long distance telephone circuit

Outline

- Data Link Layer Design Issues
 - Units
 - Services
 - Framing
- Error Detection & Correction
- Data Transmission

Framing

- Physical layer provides no guarantee a raw stream of bits is error free
- Framing is the method used by data link layer to break raw bit stream into discrete units and generate a checksum for the unit
- Checksums can be computed and embedded at the source, then computed and compared at the destination
 $\text{checksum} = f(\text{payload})$
- The primary purpose therefore, of framing, is to provide some level of reliability over the unreliable physical layer

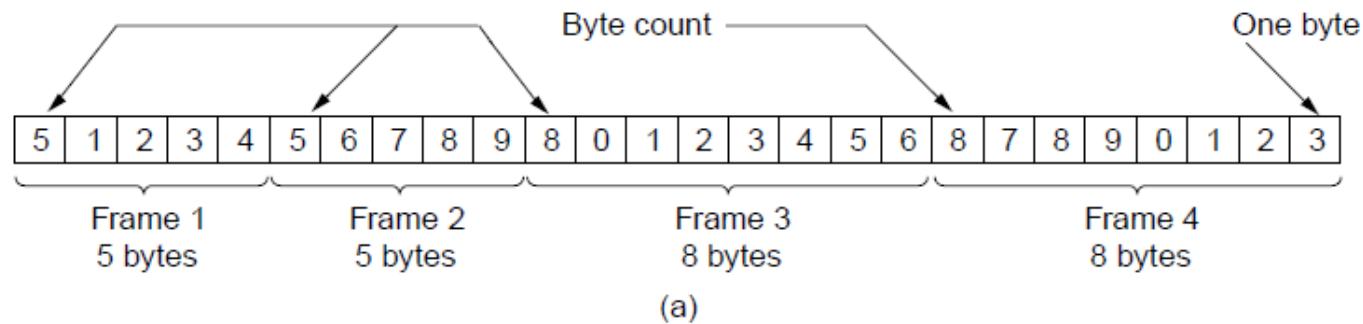
Framing Methods

- **Framing methods:**
 - Character(Byte) count
 - Flag bytes with byte stuffing
 - Start and end flags with bit stuffing
- Most data link protocols use a combination of character count and one other method

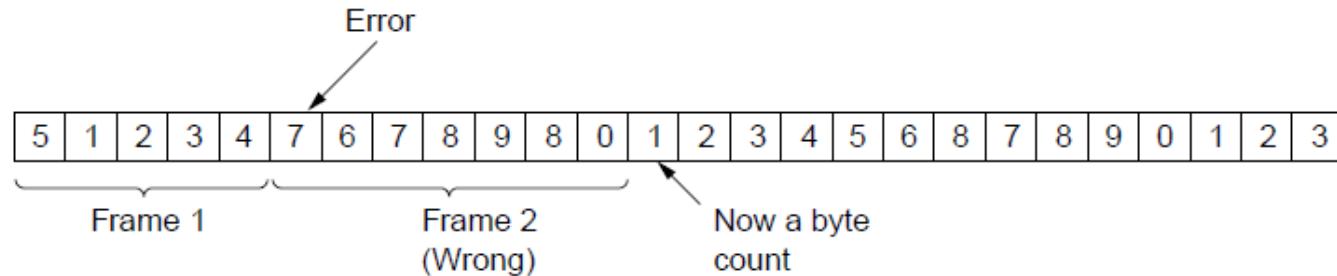
Character Counts

- Uses a field in the frame header to specify the number of characters in a frame

No error

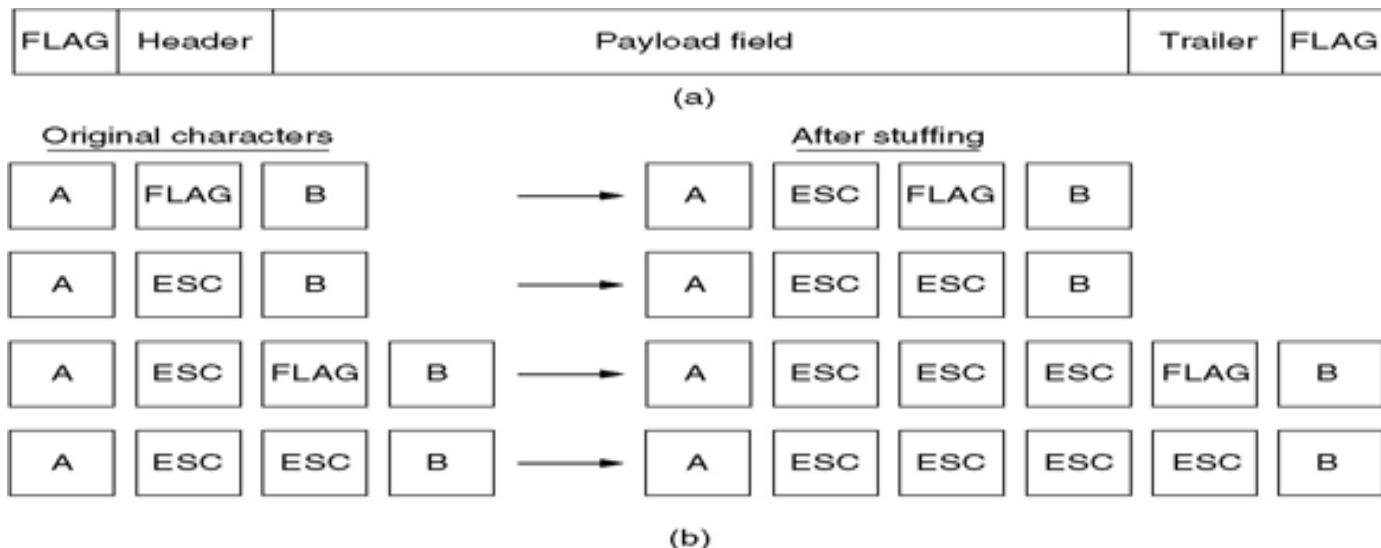


Case with
error



Flag Bytes with Byte Stuffing

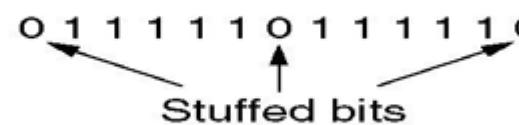
- Each frame starts and ends with a special byte -“flag byte”



Start and End flags with Bit stuffing

- Frames contain an arbitrary number of bits and allow character codes
- with an arbitrary number of bits per character
- Each frame begins and ends with a special bit pattern e.g. 01111110 – [NOT SHOWN IN THE EXAMPLE BELOW]

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 The original data

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0 Sent data


Stuffed bits

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 Destuffing at receiver

Insert 0 after five ones (11111)

Outline

- Data Link Layer Design Issues
 - Units
 - Services
 - Framing
- Error Detection & Correction
- Data Transmission

Error Control

- Ensuring that a garbled message by the physical layer is not considered as the original message by the receiver by adding check bits
- Error Control deals with
 - Detecting the error
 - Correcting the error
 - Re-transmitting lost frames
- Link layer deals with bit errors

Error Detection and Correction

- Physical media may be subject to errors
- Errors may occur randomly or in bursts
- Bursts of errors are easier to detect but harder to resolve
- Resolution needs to occur before handing data to network layer
- Key issues
 - *Fast* mechanism and *low* computational overhead
 - Detection of different kinds of error
 - *Minimum* amount of extra bits send with the data

Example



- Repeat the bits, if a copy is different than the other there is an error
 - 01101 -> 000 111 111 000 111
- How many different errors can this detect? **2**
- How many errors can this correct? **1**
- What is the minimum number of errors that can fail the algorithm? **3**
(Note: the algorithm can detect up to 2 errors and retransmission is needed but completely failed with 3 errors)
- What is the overhead? **Sending 1 bit 3 times**

Error Bounds – Hamming distance

Code turns data of n bits into codewords of $n+k$ bits

Hamming distance is the minimum bit flips to turn one valid codeword into any other valid one.

- Example with 4 codewords of 10 bits ($n=2$, $k=8$):
 - 0000000000, Hamming distance is 5
 - 0000011111,
 - 1111100000, and n=2对应4种情况00, 01, 10, 11
 - 1111111111

Bounds for a code with distance:

- $2d+1$ – can correct d errors (e.g., 2 errors above)
- $d+1$ – can detect d errors (e.g., 4 errors above)

Error Bounds

Q: Why can a code with distance $2d+1$ detect up to d errors?

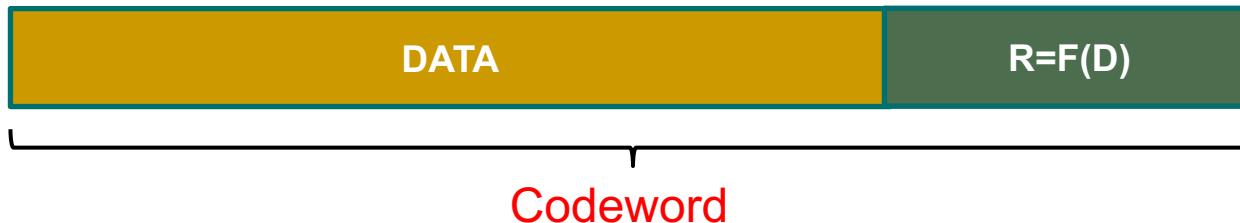
- Errors are corrected by mapping a received invalid codeword to the nearest valid codeword, i.e., the one that can be reached with the fewest bit flips
- If there are more than d bit flips, then the received codeword may be closer to another valid codeword than the codeword that was sent

Example: Sending 0000000000 with 2 flips might give 1100000000 which is closest to 0000000000, correcting the error.

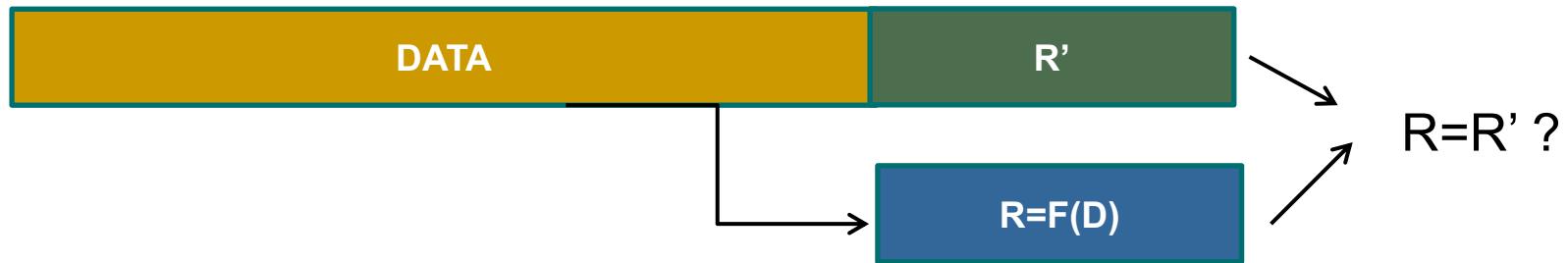
But with 3 flips 1110000000 might be received, which is closest to 1111100000, which is still an error

How it works?

- Sender calculates R check bits using a function of data bits:



- Receiver: Receive the codeword and calculates the same function on the data and match the results with received data check bits:



Error Detecting Codes

- More efficient in some transmission media –
e.g. where low error rates occur (copper wires)
- *Parity* (1 bit): XOR all the data bits and add the result as the check bit (Hamming distance=2)
- *Checksum* (16 bit): Add 16 bits of data and calculate 1's complement and add to the data as the check bits (Hamming distance=2)
- *Cyclical Redundancy Check (CRC)* – Use division by a k bits polynomial in base-2's representation (Standard 32 bit CRC: Hamming distance=4)

Error Correcting Codes

- More efficient in noisy transmission media e.g., wireless
- Challenge is that the error can be in the check bits
- Assumption on a specific number of errors occurring in transmission
- Hamming code: put parity bits at positions of power of 2

(Internet) Checksum

- There are different variation of checksum
- Because it is a word-based code as opposed to parity (a bit-based code) it is more robust but have the same hamming distance as parity
- Internet Checksum (16-bit word): Sum modulo 2^{16} and adding any overflow of high order bits back into low-order bits
- In general it may refer to any error detecting code

Example of Check Sum

Calculate checksum (5bit word) for data 00110
10001 11001 01011

1 $00110 + 10001 = 10111$

2 $11001 + 10111 = 10000 + 1 = 10001$

4 The checksum is
one's complement
of 11100 which is
00011

3 $01011 + 10001 = 11100$

Data sent: 00110 10001 11001 01011 00011

Cyclic Redundancy Check

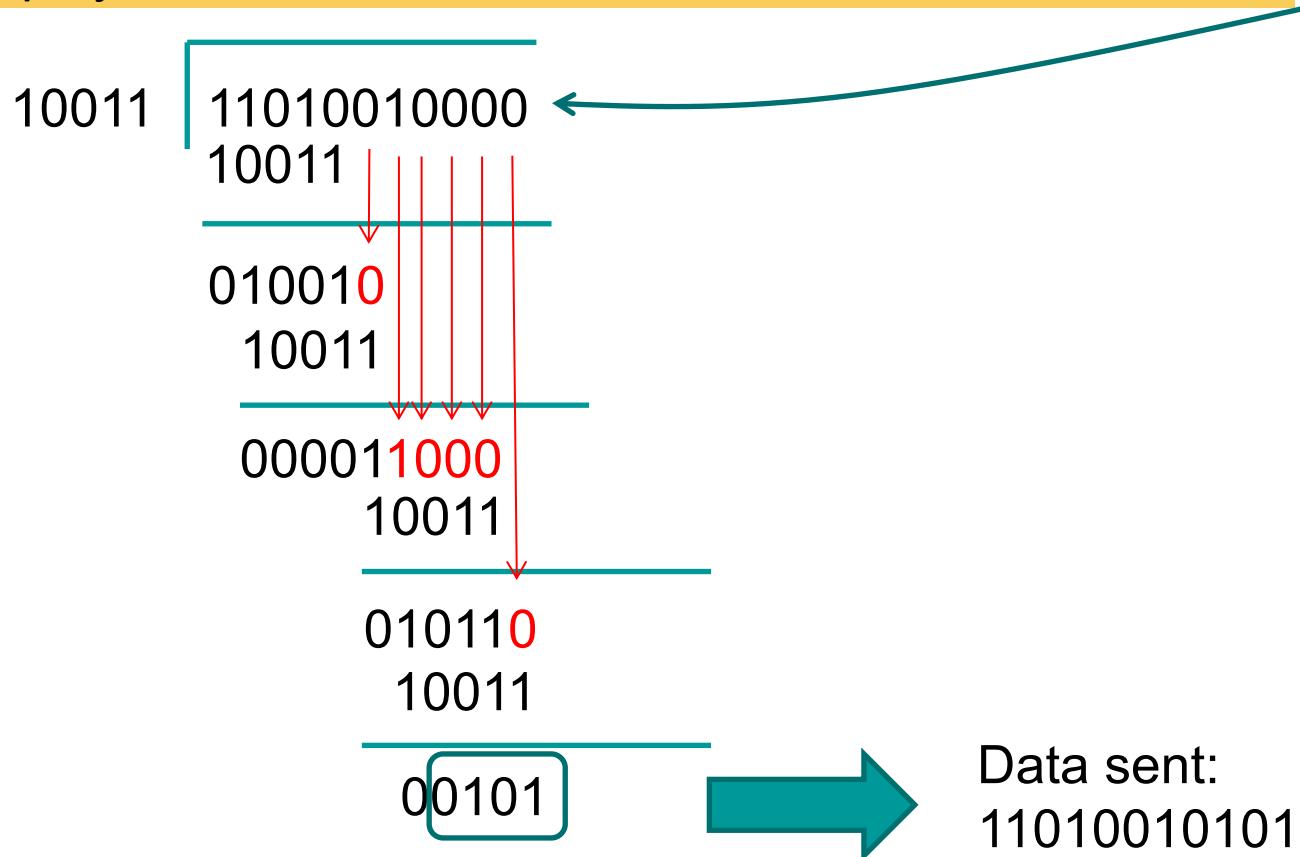
■ Based on a generator polynomial $G(x)$

- Eg. $G(x) = x^4 + x + 1$ (10011)
- Let r be the degree of $G(x)$ ($r=4$). Append r zero bits to the low-order end of the frame so it now contains $m + r$ bits and corresponds to the polynomial $x^r M(x)$.
- Divide the bit string corresponding to $G(x)$ into the bit string corresponding to $x^r M(x)$, using modulo 2 division.
- Subtract the remainder (which is always r or fewer bits) from the bit string corresponding to $x^r M(x)$ using modulo 2 subtraction. The result is the checksummed frame to be transmitted. Call its polynomial $T(x)$.

Example

Data: 1101001 and $G(x) = x^4 + x + 1$ (10011)

5 bits polynomial add 4 bits as the checksum – so add 0000



Hamming Code

- $n=2^k-k-1$ (n : number of data, k : check bits)
- Put check bits in positions p that are power of 2, starting with position 1
- Check bit in position p is parity of positions with a p term in their value
- Example: Data: 0101 -> requires 3 check bits

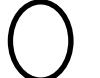
Example

Put check bits in positions p that are power of 2, starting with position 1

- Data: 0101 → requires 3 check bits

Position	P1	P2	P3	P4	P5	P6	P7
Data	?	?	0	?	1	0	1

1. Calculate the parity bits for P1, P2, P4



$$P1 + P3 + P5 + P7 = ? + 0 + 1 + 1 = 0 \text{ (even)}$$



$$P2 + P3 + P6 + P7 = ? + 0 + 0 + 1 = 1 \text{ (odd)}$$



$$P4 + P5 + P6 + P7 = ? + 1 + 0 + 1 = 0 \text{ (even)}$$

Data sent: 0100101

error

error

Example 1: At the receiver: 0100100

$$P1 + P3 + P5 + P7 = 0 + 0 + 1 + 0 = 1 \times$$

$$P2 + P3 + P6 + P7 = 1 + 0 + 0 + 0 = 1 \times$$

$$P4 + P5 + P6 + P7 = 0 + 1 + 0 + 0 = 1 \times$$

$$\text{Error bit} = P1 + P2 + P4 = P7$$

Example 2: At the receiver: 0000101

$$P1 + P3 + P5 + P7 = 0 + 0 + 1 + 1 = 0$$

$$P2 + P3 + P6 + P7 = 0 + 0 + 0 + 1 = 1 \times$$

$$P4 + P5 + P6 + P7 = 0 + 1 + 0 + 1 = 0$$

$$\text{Error bit} = P2$$

Outline

- Data Link Layer Design Issues
 - Units
 - Services
 - Framing
- Error Detection & Correction
- Data Transmission

Data Transmission

- So far we discussed how to send single messages and now we will look at a series of messages
- A service to send messages should have:
 - Reliability
 - Flow Control
- The mechanisms discussed can be implemented in other layers as well

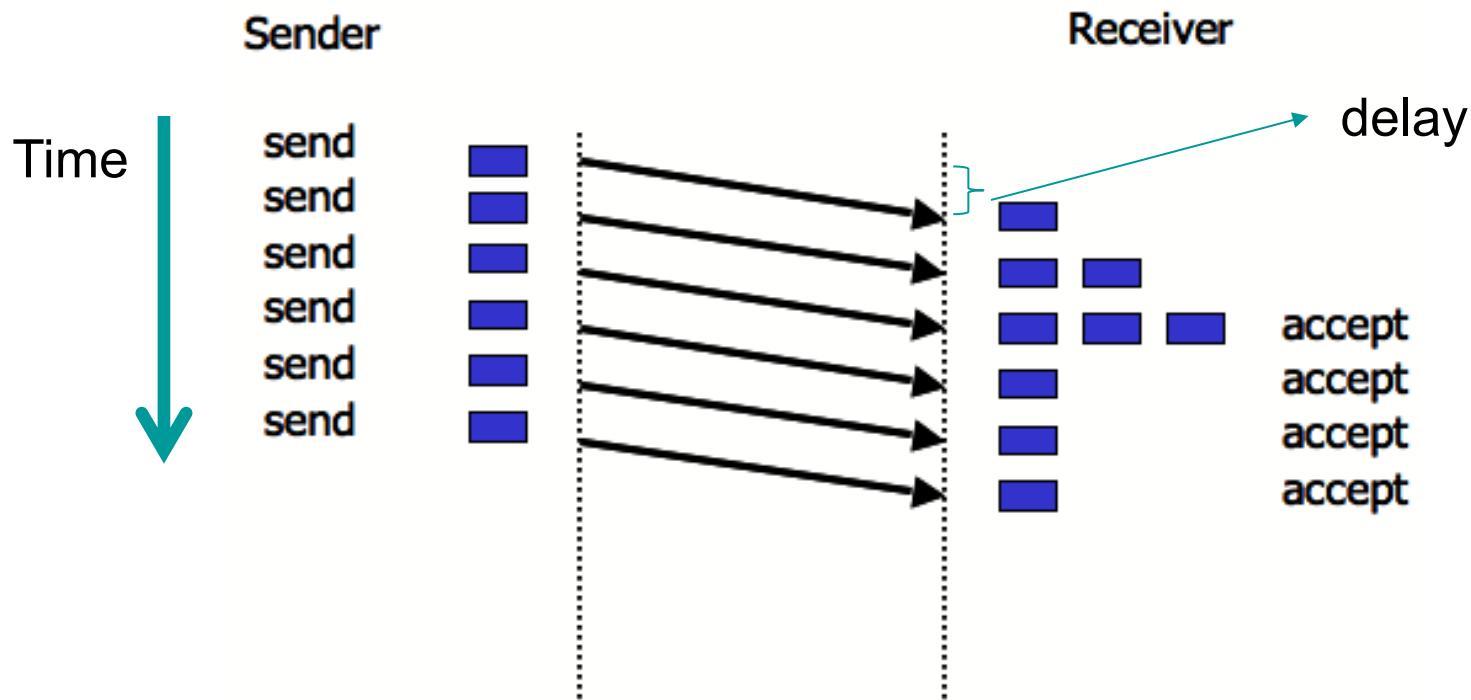
Reliability

- Each layer need to make sure the service provided to other layers is reliable
- Retransmission with error detection is a way of ensuring reliability
- Error correction is another way but has its own shortcomings

Flow Control

- The fast senders vs slow receivers problem requires a solution
- Principles to control when sender can send next frame
 - Feedback based flow control (usually used in DL layer)
 - Rate based flow control

A Very Simple Protocol

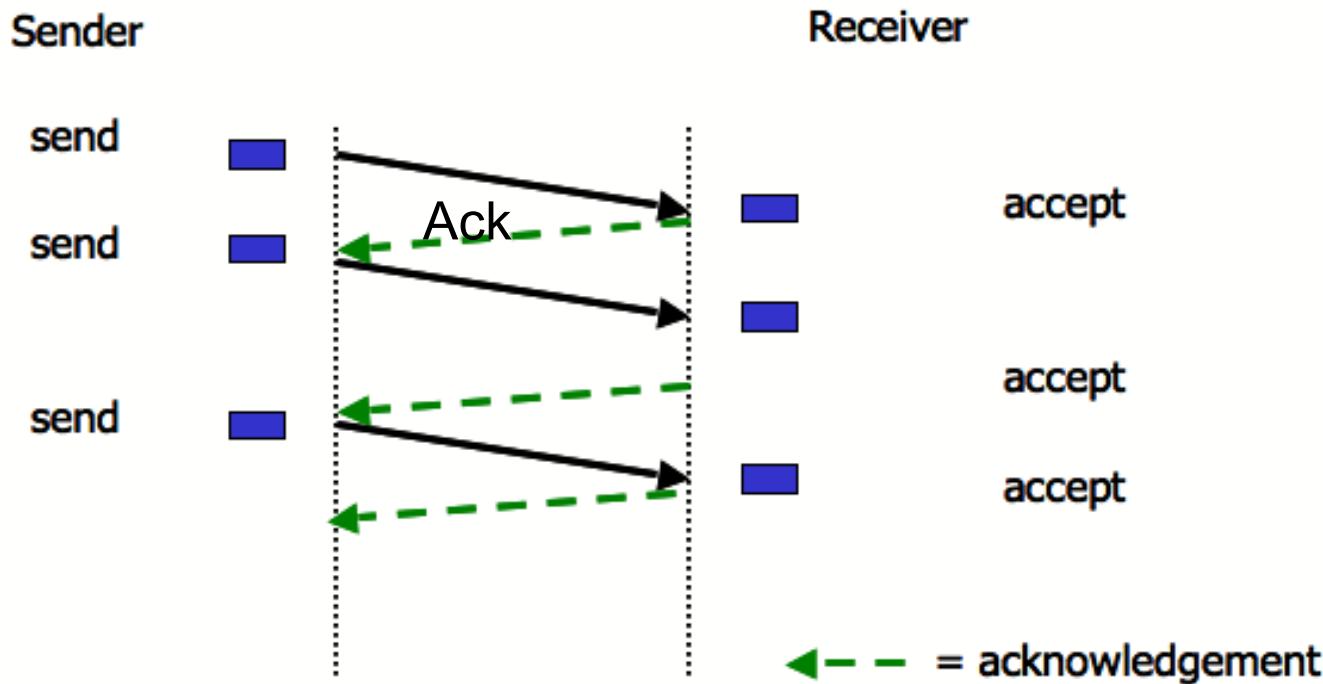


Acknowledged transmission

(think: fast sender / slow receiver)

Data transmitted in one direction

Time is relatively important, buffer space constrained

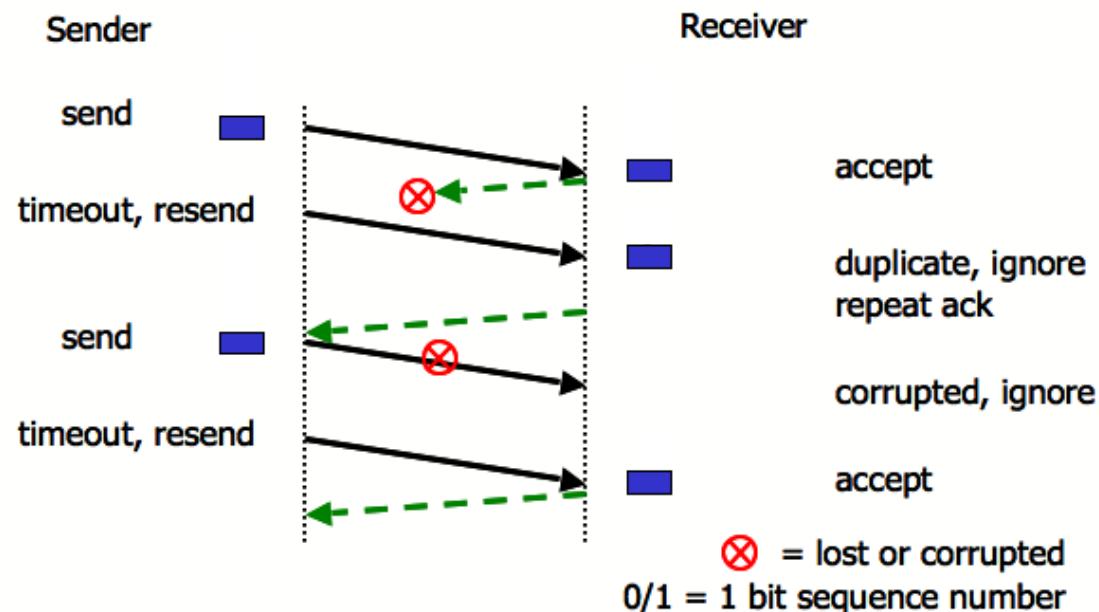


Noisy Channel Protocol

- Frames can be lost either entirely or partially
- Requires distinction between frames already sent/received and those being re-transmitted
- Requires *timeout function* to determine arrival or non-arrival of complete frames

Stop and Wait Protocol

- Concept of ARQ (Automatic Repeat reQuest)
 - Ack and Timeout
- Stop and Wait
 - One bit Ack



Link Utilization in Stop and Wait Protocols

Principle of efficiency in communication is measured by **Link Utilization (U)**.

Let **B** be the **bit-rate** of the link and **L** the **length of the frame**,

T_f = Time needed to transmit a frame of length L,

T_p = Propagation delay of the channel,

T_a = Time for transmitting an Ack,

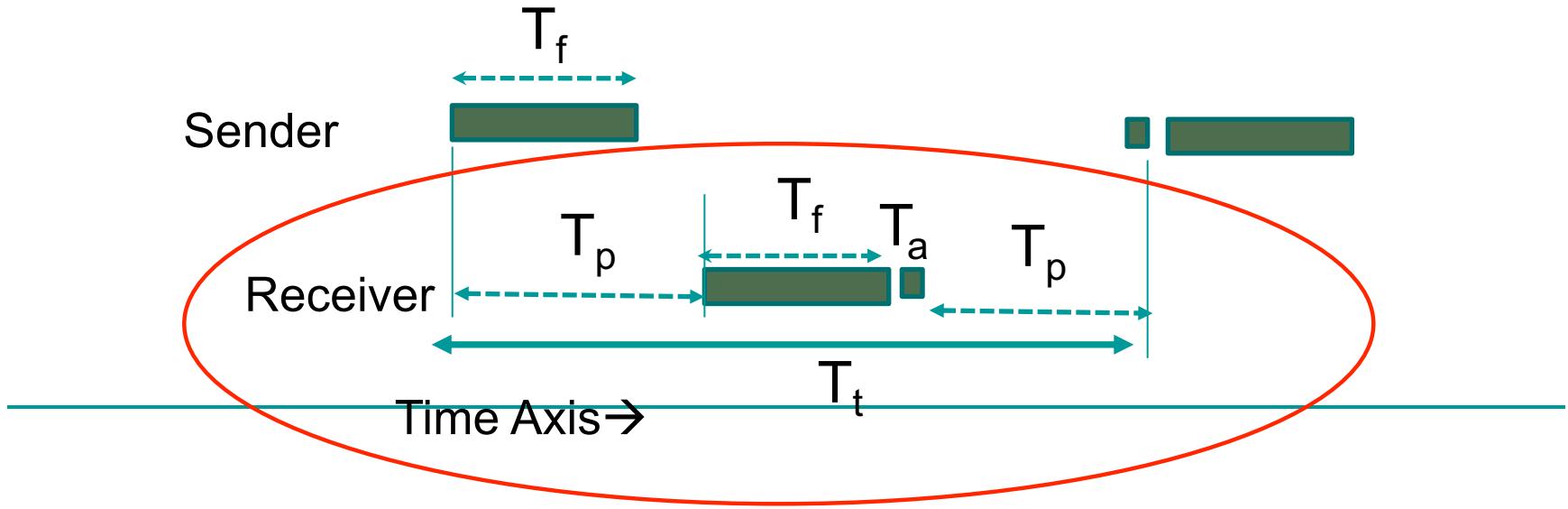
So we have $T_f = L/B$. We can assume $T_a = 0$. $T_t = T_f + 2T_p$

For example for a Link with $B=1\text{Mbps}$ and $T_p=50\text{ms}$ and frame size 10Kb :

$$U = \frac{10000}{(10000 + 0.1 \cdot 10^6)} = 1/11;$$

$$U = (\text{Time of transmitting a frame}) / (\text{Total time for the transfer}) = T_f / T_t$$

$$\text{We have } U = T_f / (T_f + 2T_p) = (L/B) / (L/B + 2T_p) = L / (L + 2T_p B).$$

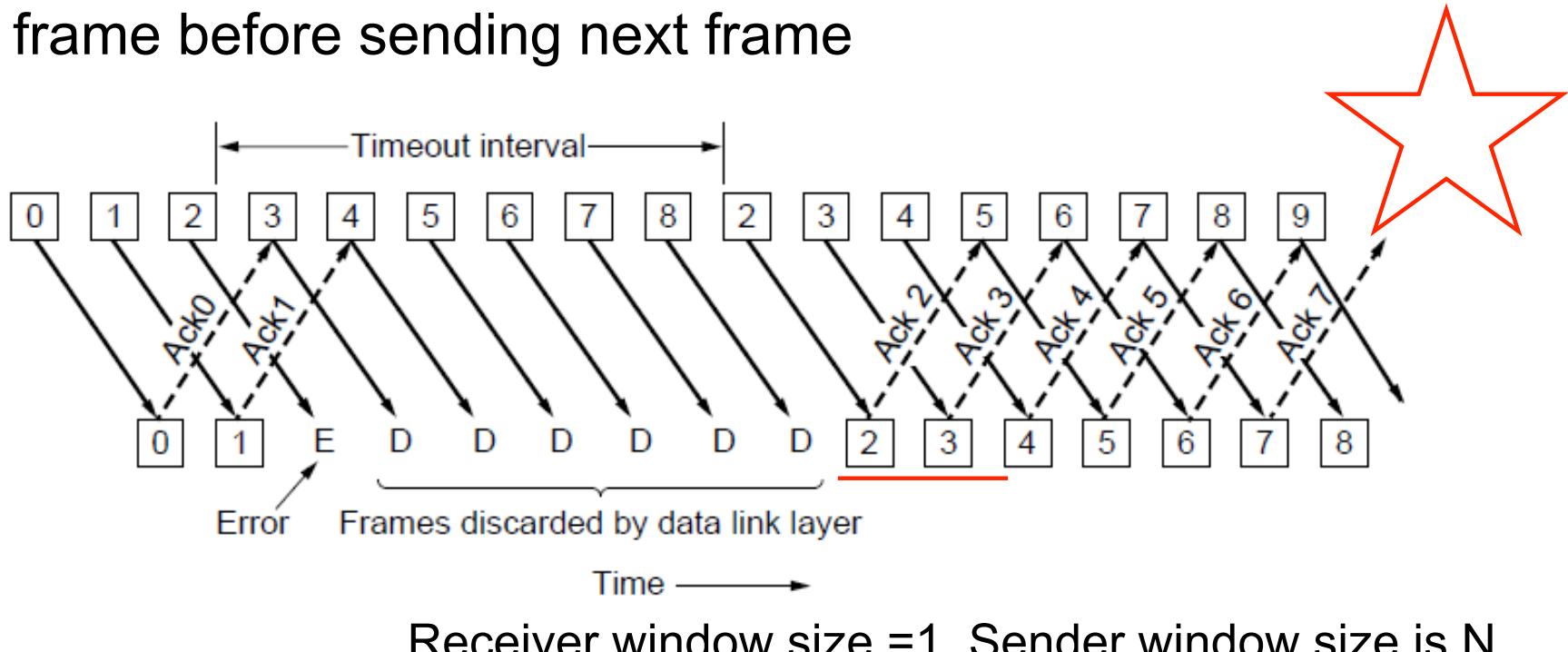


Sliding Window Protocols

- Data is commonly transmitted in both directions simultaneously
- Sender maintains a set of sequence numbers corresponding to frames it is allowed to send (within the “sending window”)
- Receiver maintains a set of sequence numbers corresponding to frames it is allowed to accept (within the “receiving window”)
- Stop and Wait can be seen as a special case with window size 1

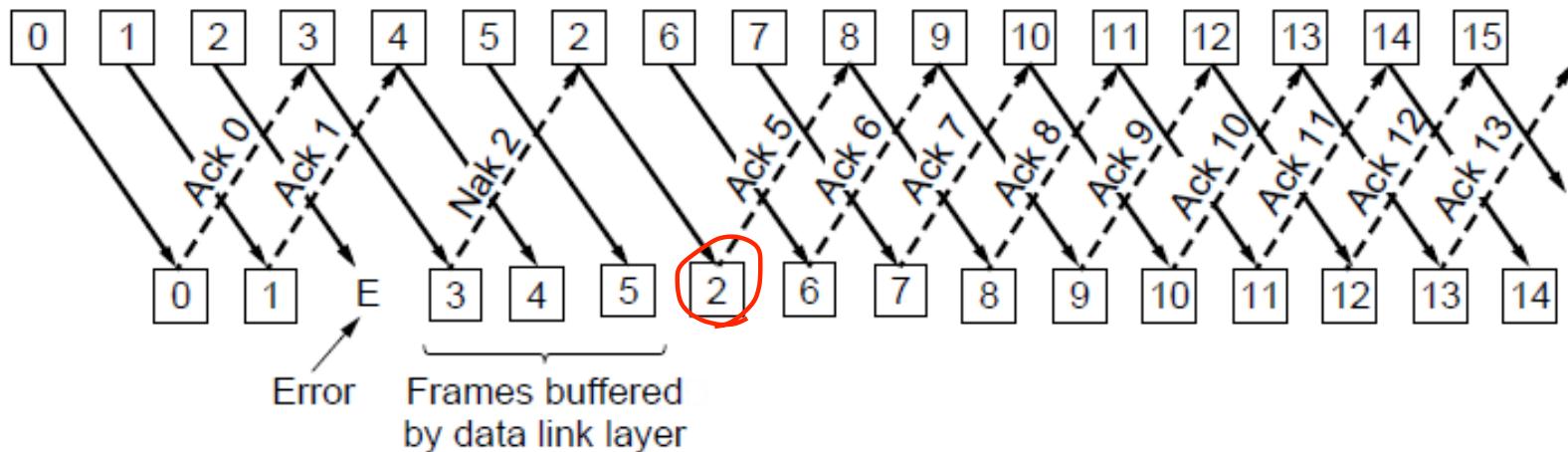
Protocol Using Go-Back-N

- Long transmission times need to be taken into account when programming timeouts e.g., low bandwidth or long distance
- Senders don't need to wait for acknowledgement for each frame before sending next frame



Selective Repeat

- Receiver accepts frames anywhere in receive window
 - Cumulative ack indicates highest in-order frame
 - NAK (negative ack) causes sender retransmission of a missing frame before a timeout resends window

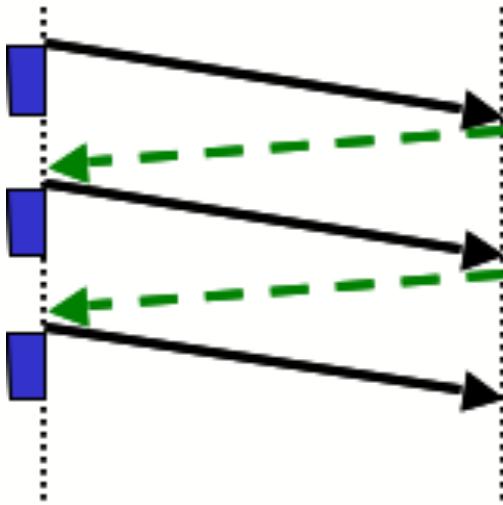


Go-Back-N vs Selective Repeat

- Go-Back-N: receiver discards all subsequent frames from error point, sending no acknowledgement, until the next frame in sequence
- Selective Repeat: receiver buffers good frames after an error point, and relies on sender to resend oldest unacknowledged frames
- Trade-off between efficient use of bandwidth and data link layer buffer space

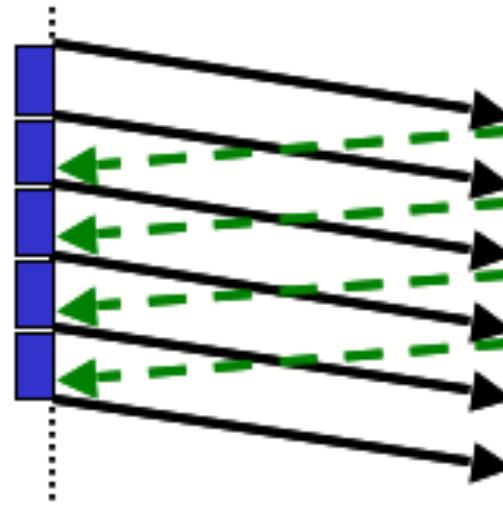
Link Efficiency Compared

Stop and Wait



50% utilisation

Sliding Window



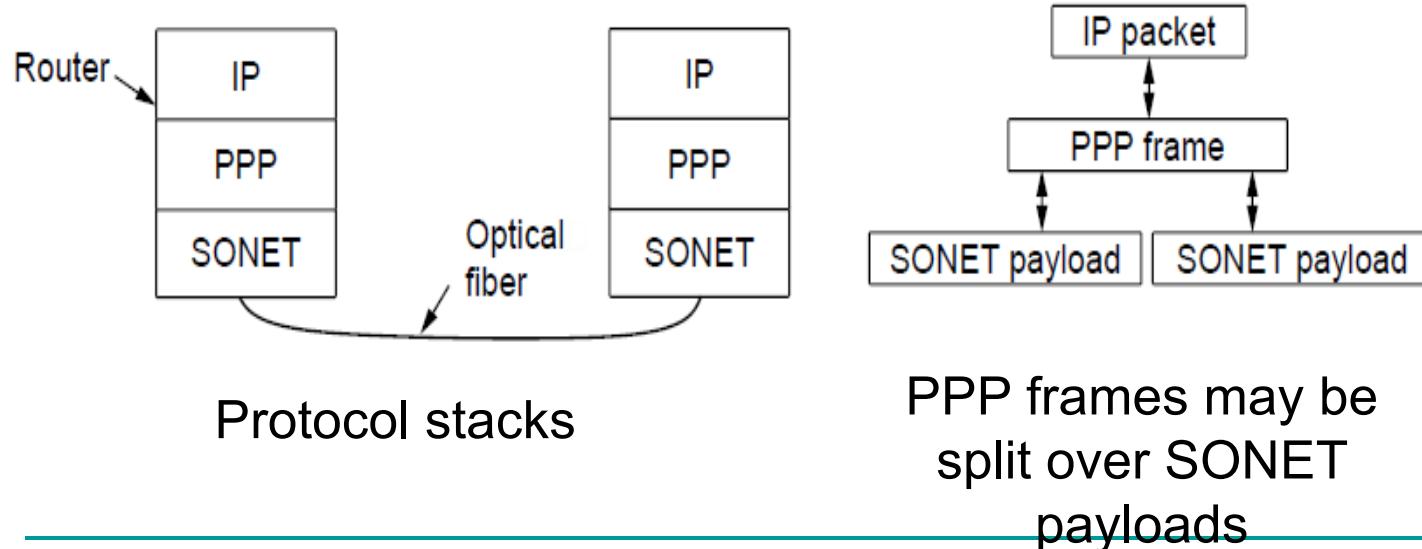
100% utilisation

Example Data Link Protocols

- Packet over SONET
- PPP (Point-to-Point Protocol)
- ADSL (Asymmetric Digital Subscriber Loop)

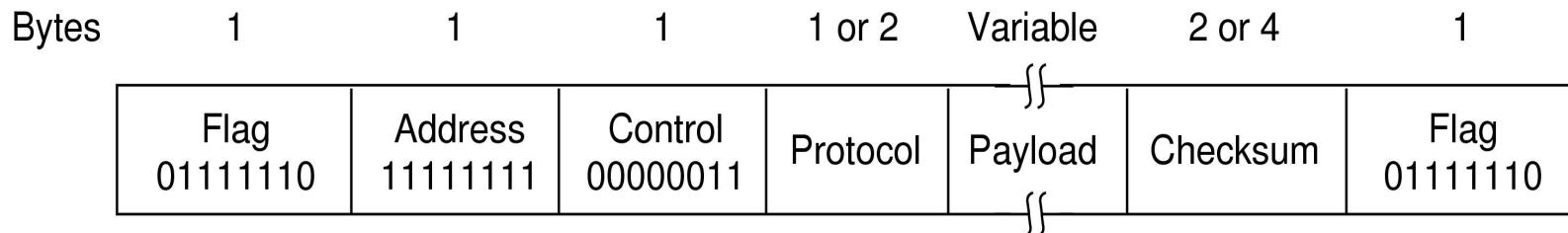
Packet over SONET

- Packet over SONET is the method used to carry IP packets over SONET optical fiber links
 - Uses PPP (Point-to-Point Protocol) for framing



PPP (1)

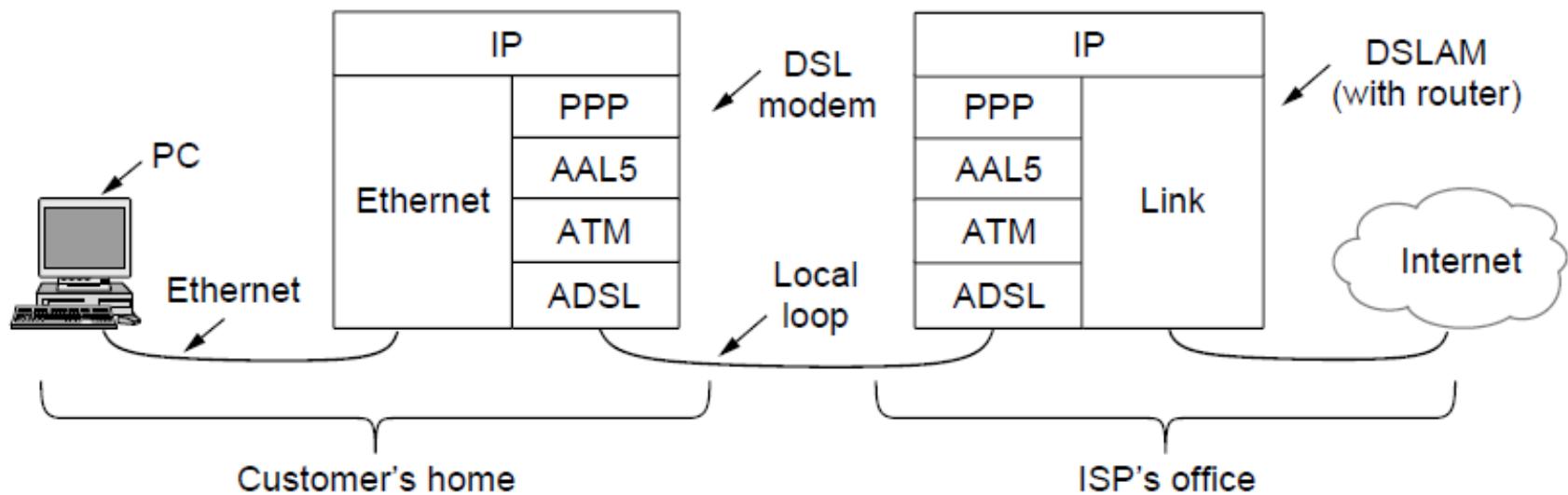
- PPP (Point-to-Point Protocol) is a general method for delivering packets across links
 - Framing uses a flag (0x7E) and byte stuffing
 - “Unnumbered mode” (connectionless unacknowledged service) is used
 - Errors are detected with a checksum



0x21 for IPv4IP packet

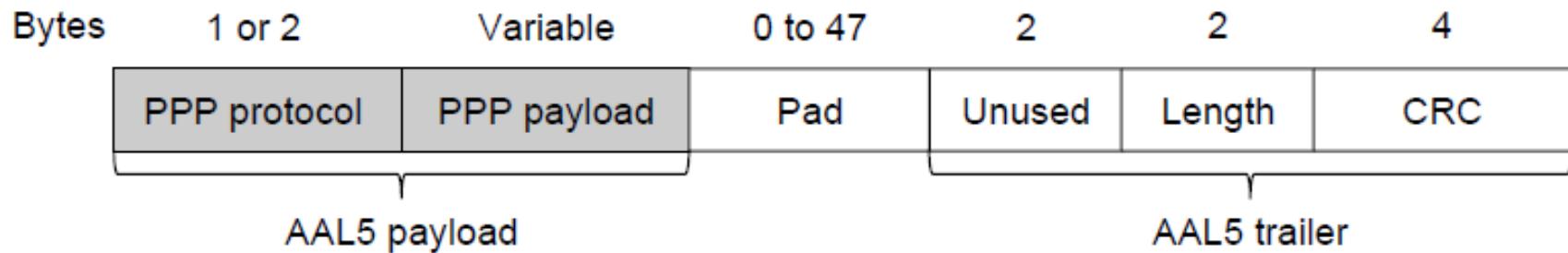
ADSL (1)

- Widely used for broadband Internet over local loops
 - ADSL runs from modem (customer) to DSLAM (ISP)
 - IP packets are sent over PPP and AAL5/ATM (over)



ADSL

- PPP data is sent in AAL5 frames over ATM cells:
 - ATM is a link layer that uses short, fixed-size cells (53 bytes); each cell has a virtual circuit identifier
 - AAL5 is a format to send packets over ATM
 - PPP frame is converted to a AAL5 frame (PPPoA)



AAL5 frame is divided into 48 byte pieces, each of which goes into one ATM cell with 5 header bytes

Summary

- Functions & methods of DL layer
- Framing
 - Character count
 - Byte stuffing and bit stuffing
- Error control
 - Detection and correction
- Reliability
- Flow Control Procedures
 - Stop and wait
 - Sliding window protocols (Go-back-N vs Selective request)

Week 4 - MAC Sub-Layer

COMP90007
Internet Technologies

Chien Aun Chan

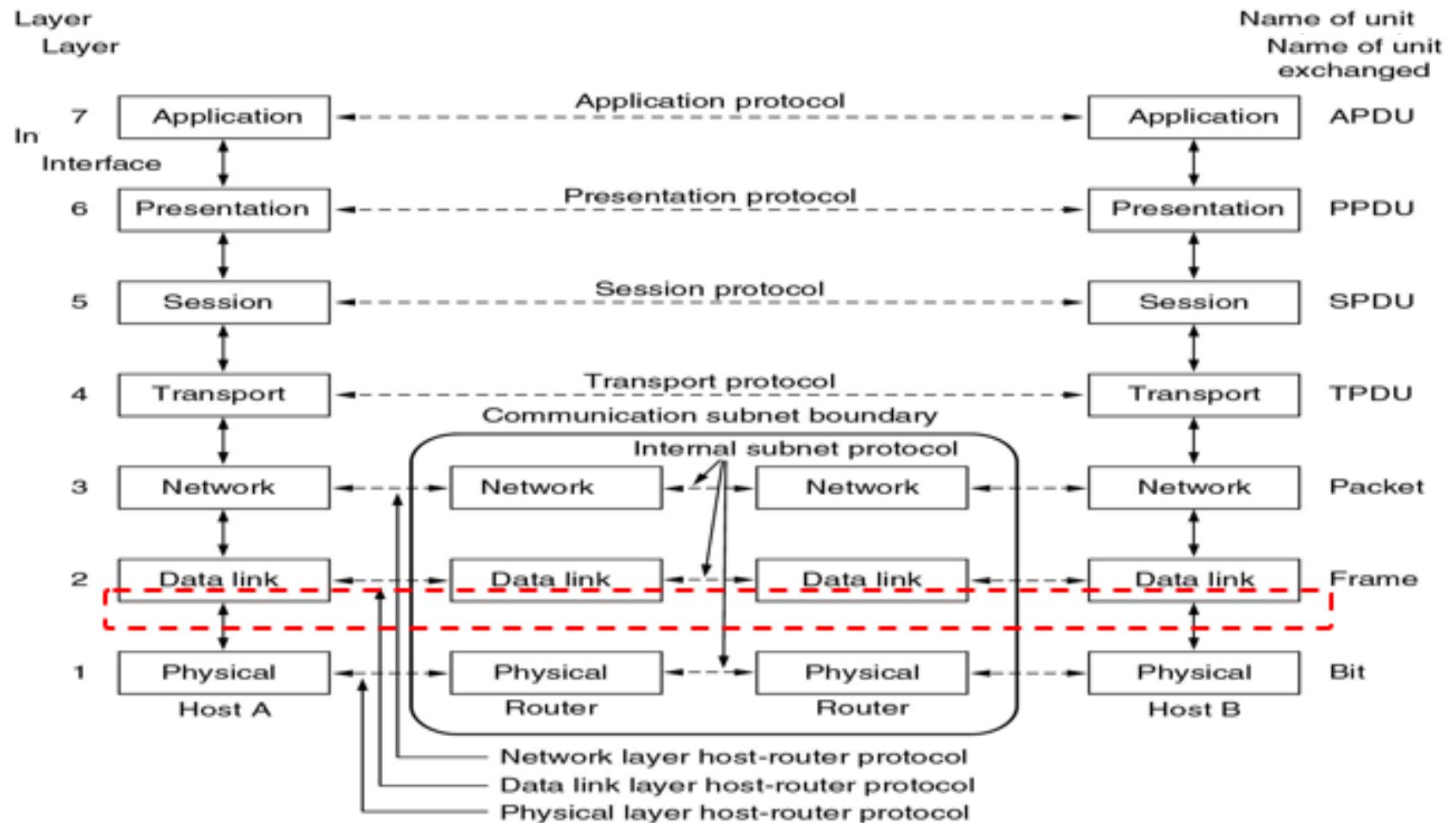
Outline

- The MAC Sub-layer
- Channel Allocation Problems
- Multiple Access Protocols

Introduction

- On **point to point networks**, there are only singular sender and receiver pairs, eliminating transmission contention
- On **broadcast networks**, determining right to transmit is a complex problem
- **Medium Access Control (MAC)** sub-layer is used to assist in resolving transmission conflicts

The MAC Sub-layer



Types of Channel Allocation Mechanisms

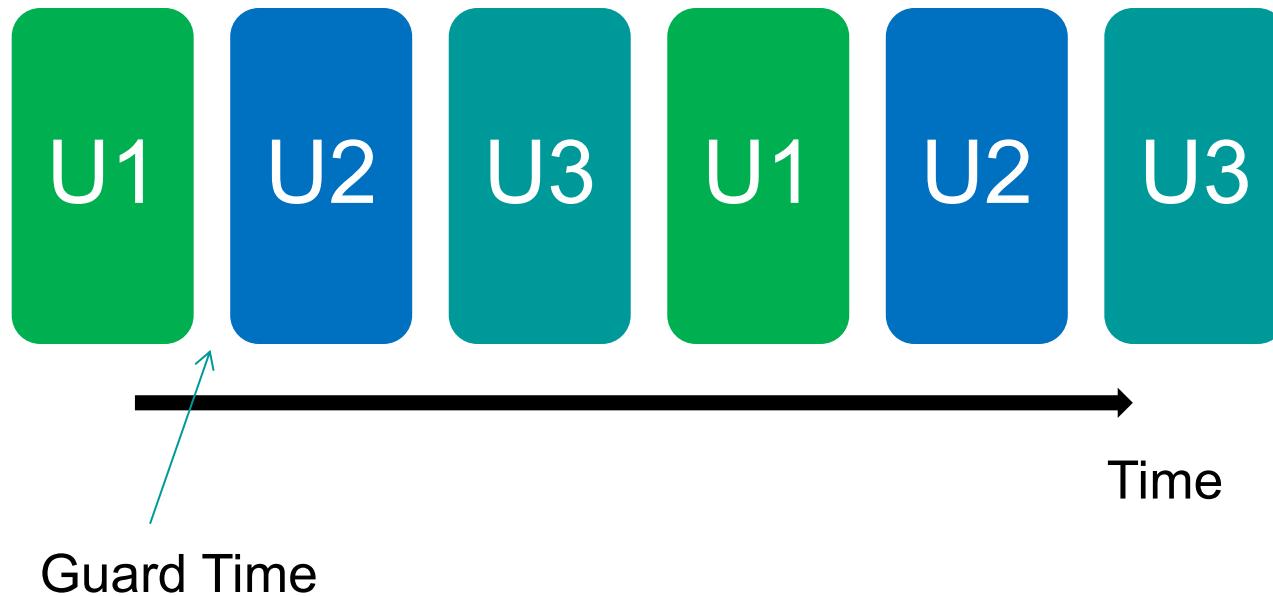
- Various methods exist for allocating a single broadcast channel amongst competing users
 - Static Channel Allocation
 - Dynamic Channel Allocation

Static Channel Allocation

- Arbitrary division of a channel into segments and each user allocated a dedicated segment for transmission
- Frequency Division Multiplexing (FDM) is typically used
- Significant inefficiencies arise when:
 - Number of senders > allocated segments
 - Number of senders is not static
 - Traffic is bursty

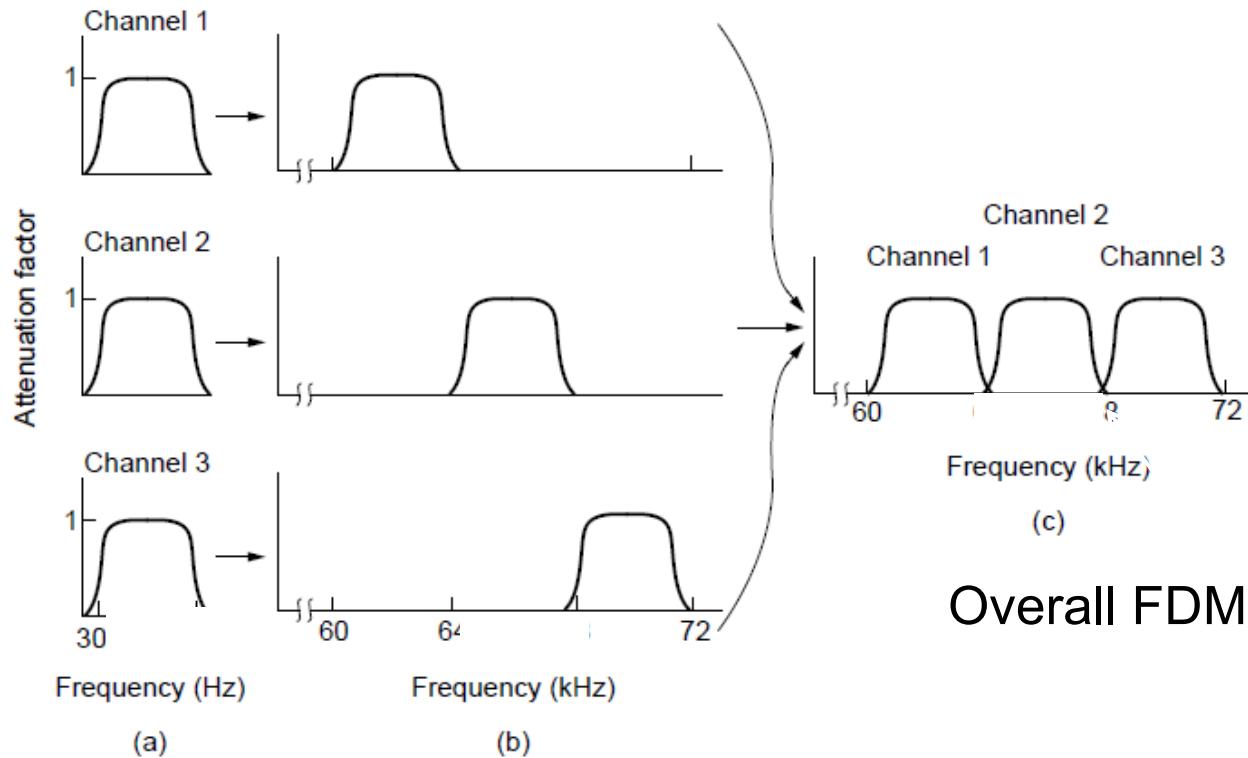
Time Division Multiplexing

Users take turns on a fixed schedule



Frequency Division Multiplexing

FDM (Frequency Division Multiplexing) shares the channel by placing users on different frequencies:



Overall FDM channel

Down falls

- Usually good for fixed number of users
- Network traffic is bursty
 - TDM and FDM try to give consistent access to the network leading to inefficiency in the use of network resources
- Where?
 - TV and Radio (FDM)
 - 2G uses TDM

Dynamic Channel Allocation

- Channel segmentation is dynamic, segment allocation is dynamic
- Assumptions for dynamic channel allocation:
 - Independent transmission stations
 - Single channel for all communication
 - Simultaneous transmission results in damaged frames
- Time
 - Transmission can begin at any time
 - Transmission can begin only within discrete intervals
- Carrier Sense
 - Detection of channel use prior to transmission
 - No detection of channel use prior to transmission

Multiple Access Protocols

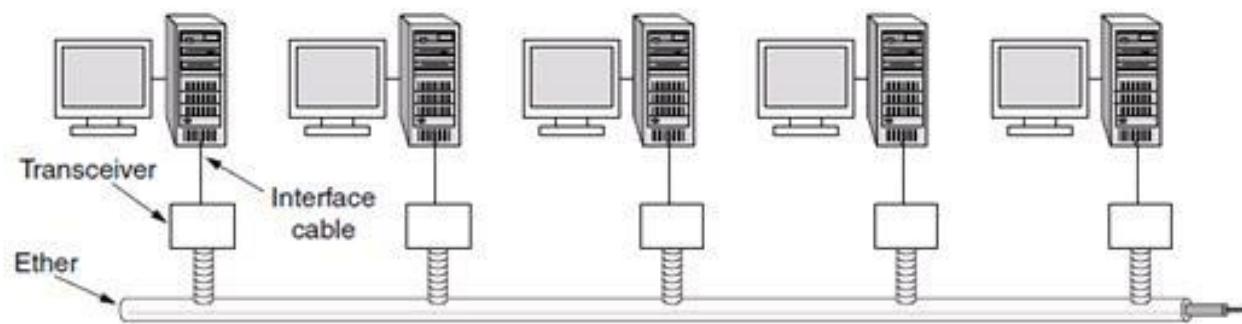
- ALOHA
- Carrier Sense Multiple Access
- Collision Free
- Limited Contention
- MACA/MACAW (for Wireless LANs)

ALOHA

- Users transmit frames whenever they have data; users retry after a random time if there are collisions (or no Ack is arrived)
- Requires no central control mechanism
- Efficient under low load but inefficient under high traffic loads
- Slotted ALOHA: Allows the users to start sending only at the beginning of defined slots. Increase efficiency of pure ALOHA by reducing possibility of collisions

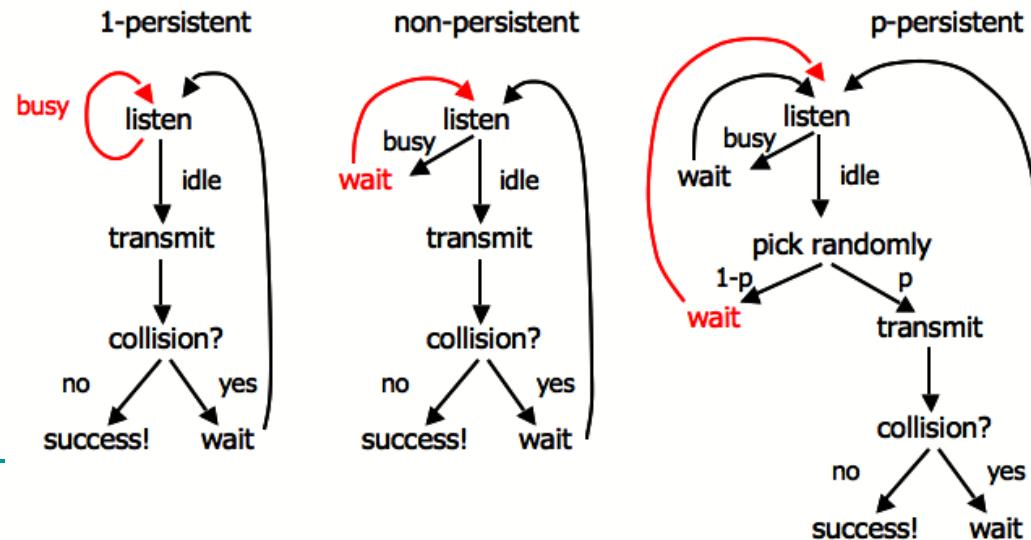
Carrier Sense Multiple Access (CSMA) Protocols

- In networks which require transmission state detection to determine transmission rights dynamically, there are specific protocols which are used
 - Persistent and Non-Persistent CSMA
 - CSMA with Collision Detection

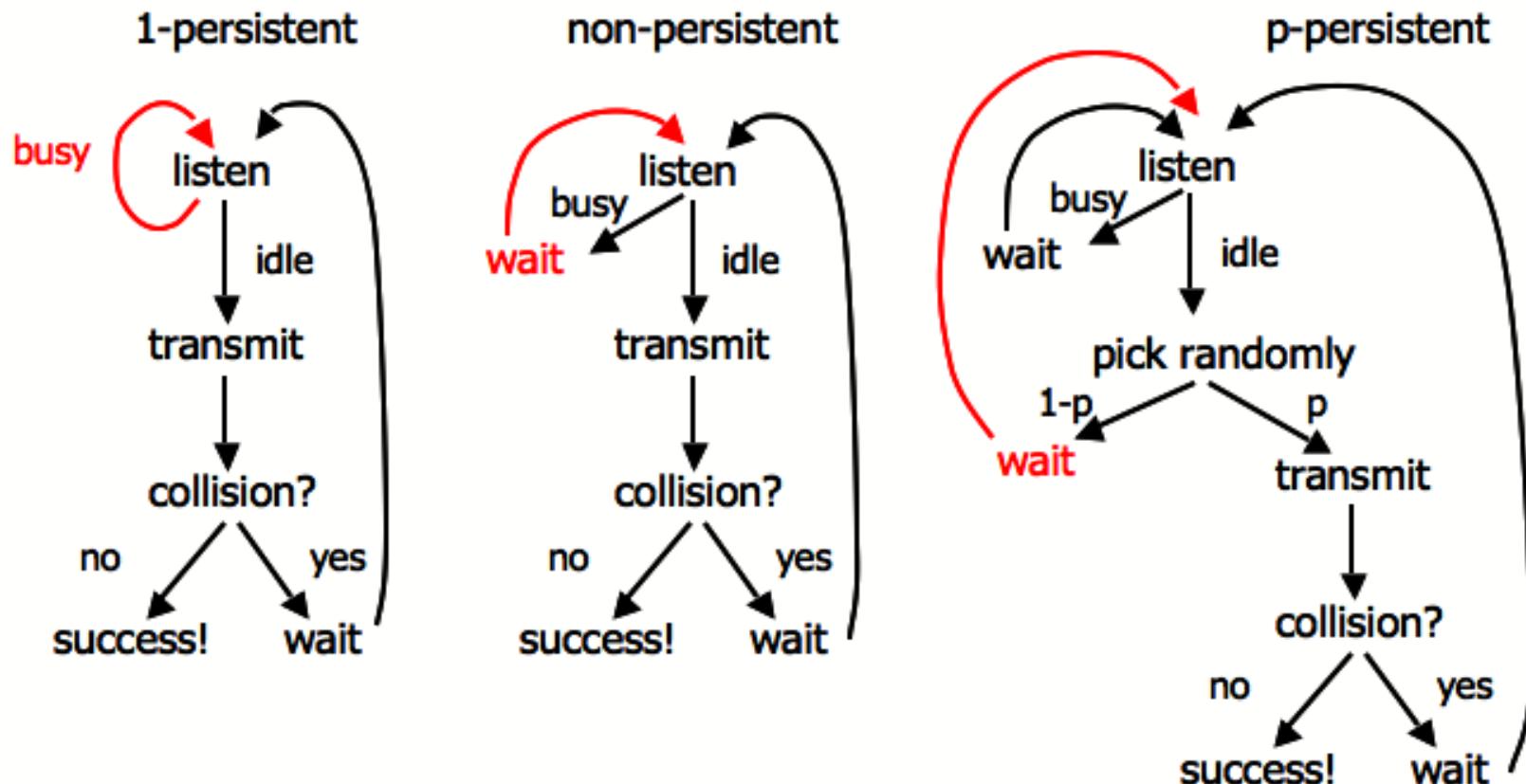


Persistent and Non-Persistent CSMA

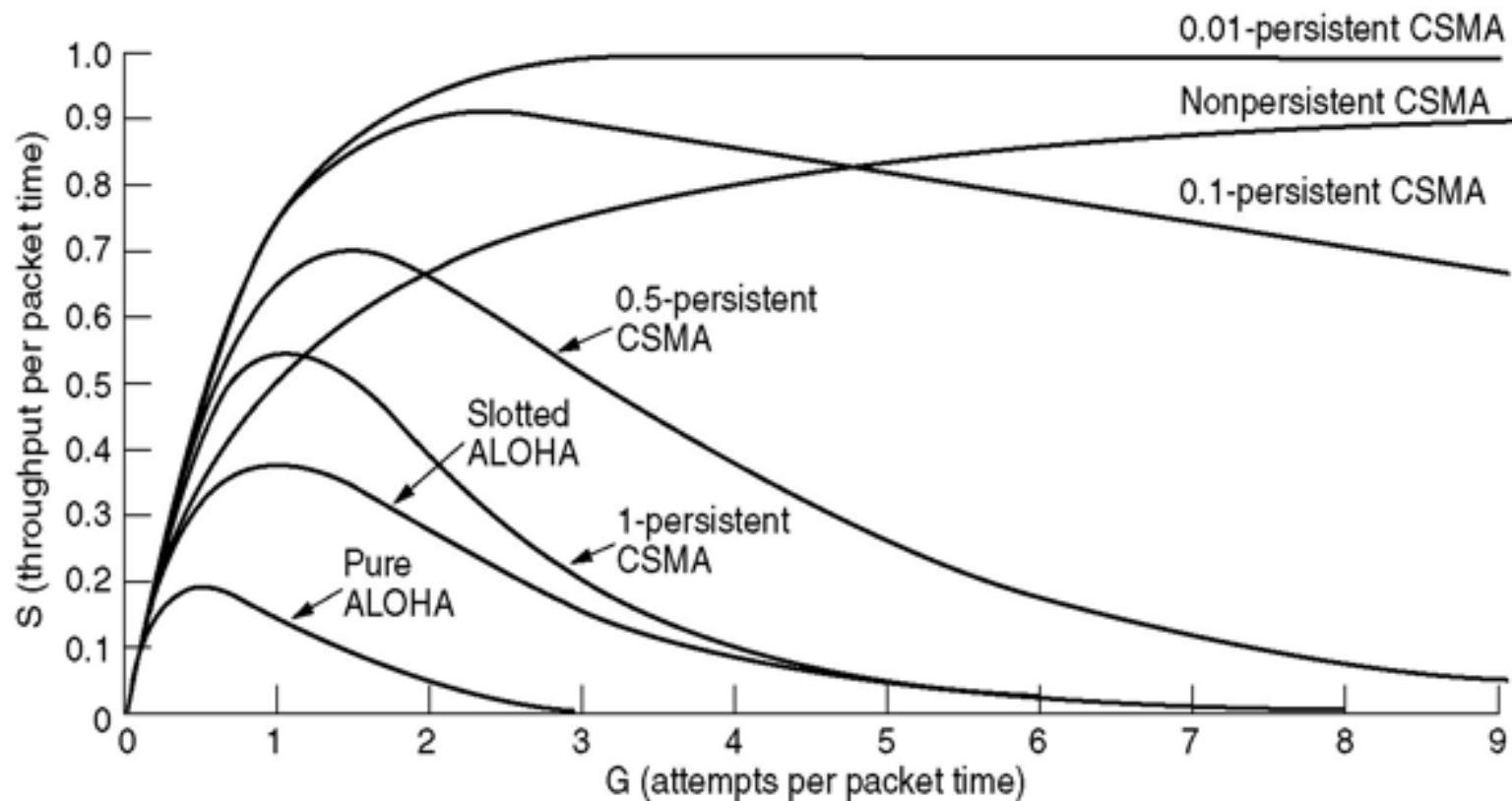
- When a sender has data to transmit, first check channel to detect other active transmission
- 1-persistent CSMA**
 - Wait until channel idle; transmit one frame and check collisions; if collision, wait for a random time and repeat
- Non-persistent CSMA**
 - If channel busy, wait random period and check again; if not, start transmitting
- p-persistent CSMA**
 - If channel idle, transmit with probability p , or wait with probability $(1-p)$ and check again



Persistent and Non-Persistent CSMA



CSMA Variants

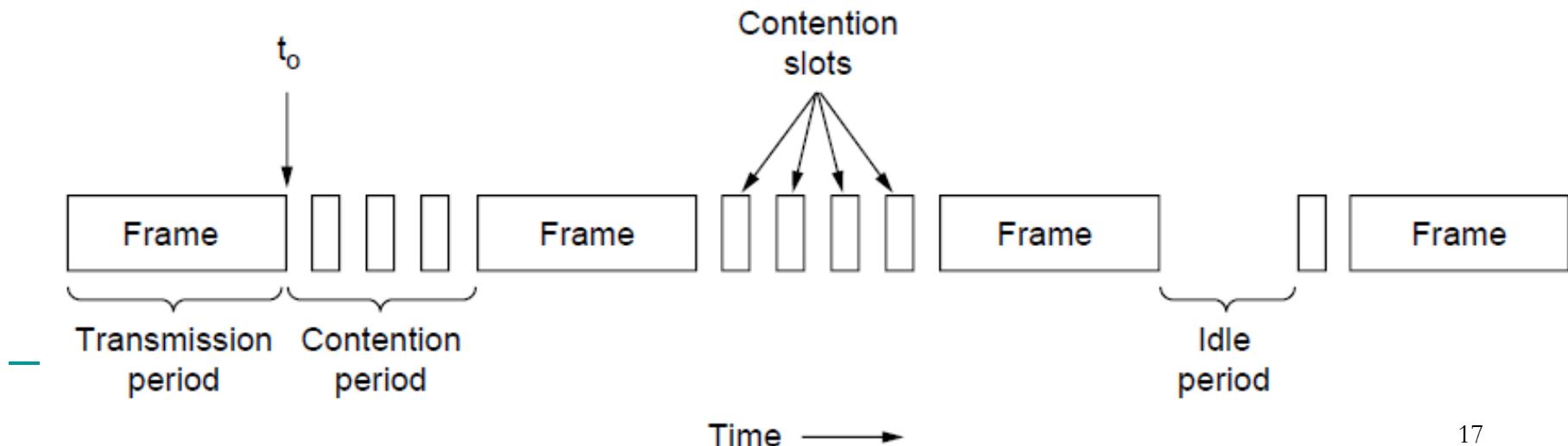


CSMA outperforms ALOHA, and being less persistent is better under high load

CSMA with Collision Detection

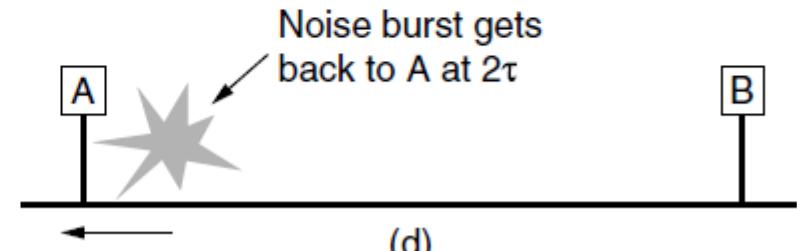
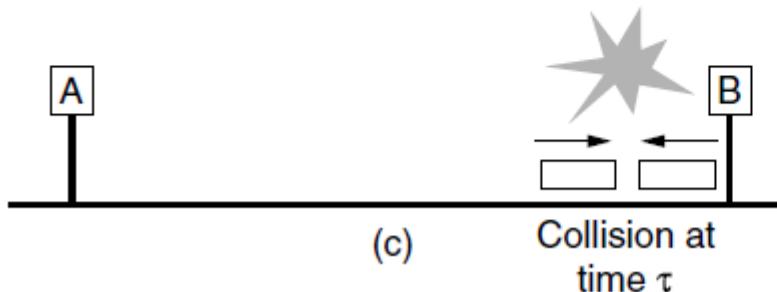
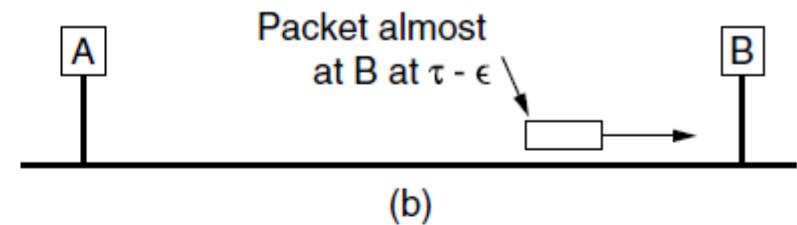
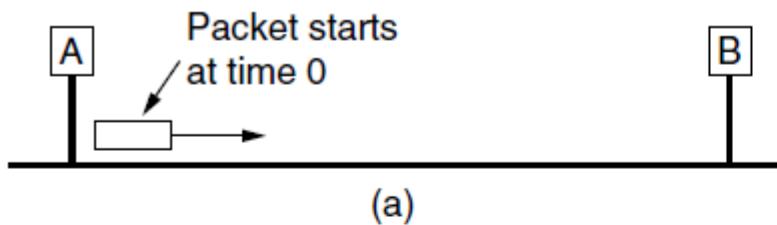
- Principle that transmission aborted when collision detected
- After collision detected, abort, wait random period, try again
- Channel must be continually monitored
- Used in half-duplex system (e.g., with Hub or repeater)

Reduced contention times improve performance



Classic Ethernet Minimum Packet Size

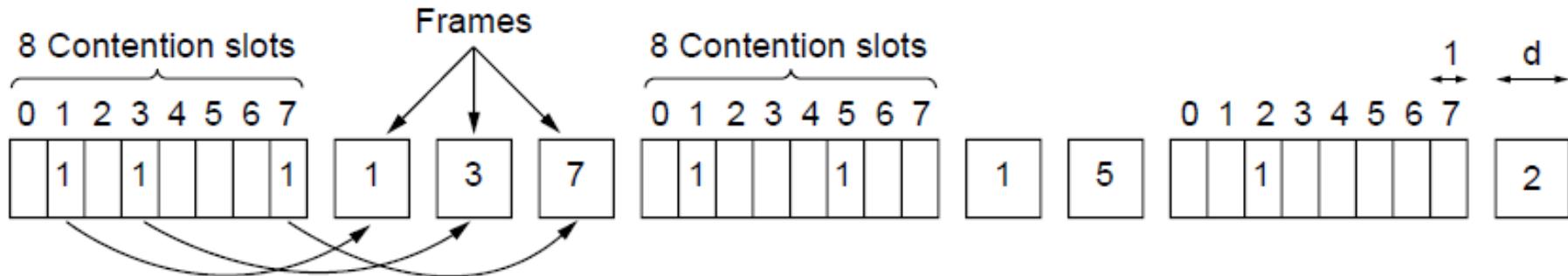
- Collisions can occur and take as long as 2τ to detect
 - τ is the time it takes to propagate over the Ethernet
 - Leads to minimum packet size for reliable detection



Collision Free Protocols

■ Bit Map Protocol

- ❑ Reservation-based protocol
- ❑ 1 bit per station overhead
- ❑ Division of transmission right, and transmission event - no collisions as this is a reservation based protocol



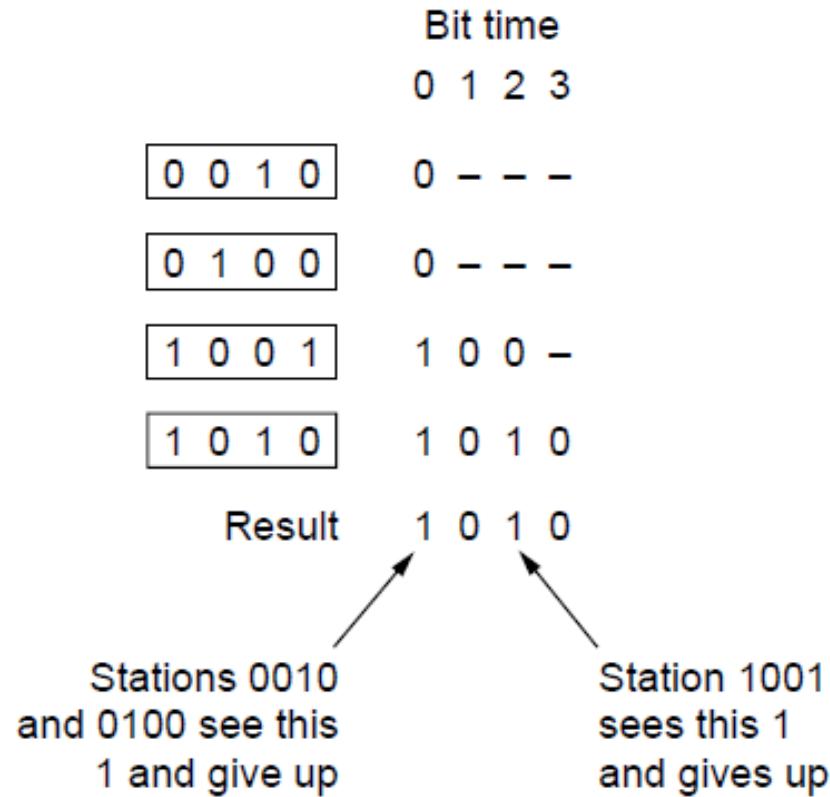
Collision Free Protocols

■ Binary Countdown Protocol

- Avoid the 1 bit per station scalability problem by using binary station addressing
- No collisions as higher-order bit positions are used to arbitrate between stations wanting to transmit
- Higher numbered stations have a higher priority

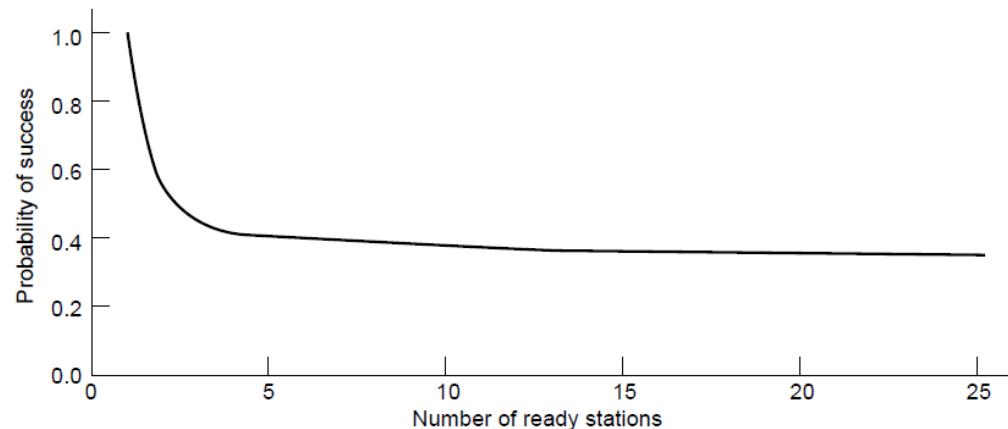
Binary Countdown Protocol

- Stations send their address in contention slot ($\log N$ bits instead of N bits)
- Channel medium ORs bits; stations give up when they send a “0” but see a “1”
- Station that sees its full address is next to send



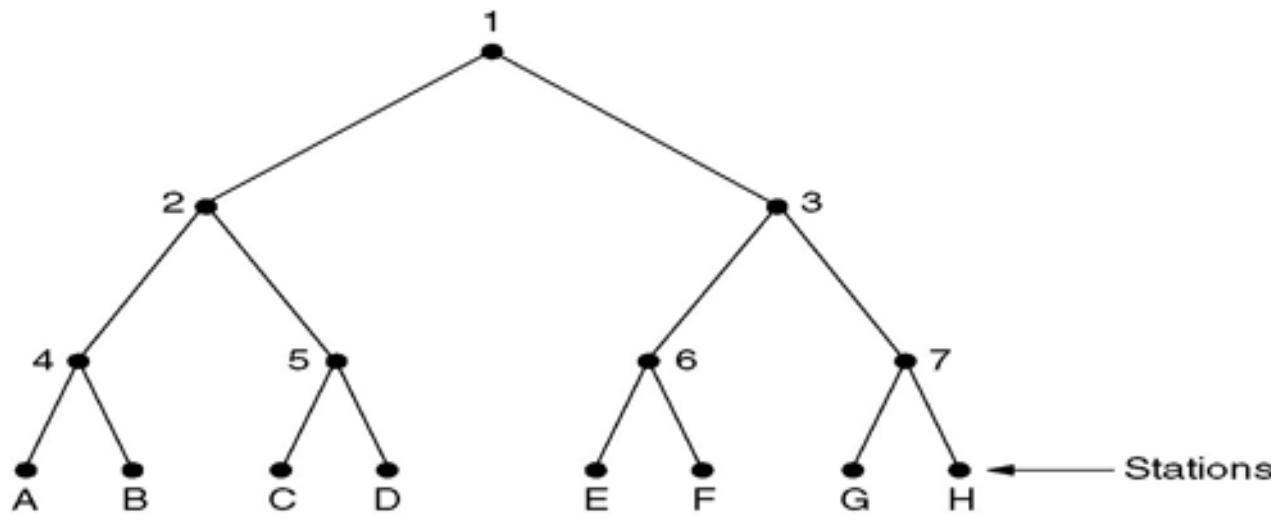
Limited Contention Protocols

- 2 strategies - **contention** and **collision free** - both become inefficient at different points
- Under **low loads**, collision free is less attractive because of a higher delay between transmissions
- Under **higher loads**, contention is less attractive because overhead associated with channel arbitration becomes greater
- **Limited Content Protocols** increase the probability of stations acquiring transmission rights by arbitrarily dividing stations and using a binary algorithm to determine rights allocation
 - Idea is to divide stations into groups within which only a very small number are likely to want to send
 - Avoids wastage due to idle periods and collisions



Adaptive Tree Walk Protocol

- All stations compete for right to transmit, if a collision occurs, binary division is used to resolve contention
- Tree divides stations into groups (nodes) to poll
 - Depth first search under nodes with poll collisions
 - Start search at lower levels if >1 station expected



Example 1: D G
Slot 1 → D, G – collision
Slot 2 → D
Slot 3 → G

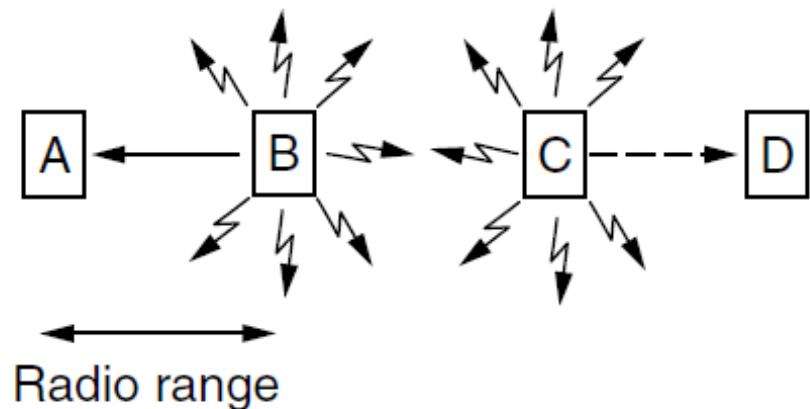
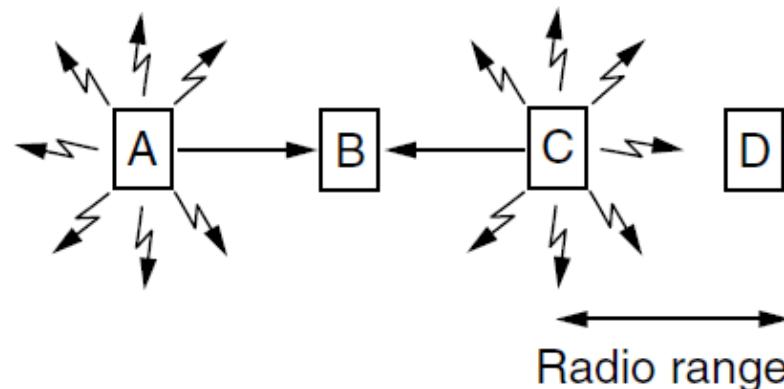
Example 2: B D G
Slot 1 → B, D, G – collision
Slot 2 → B, D - collision
Slot 3 → B
Slot 4 → D
Slot 5 → G

Wireless LAN Protocols

- Wireless Complications: when a station is in the range of two transmitters or relays, interference affects signal reception
- Leads to hidden and exposed terminal problems
- Require detection of transmissions to receiver, not just carrier sensing
- Transmission Protocols for Wireless LANs (802.11)
 - Multiple Access with Collision Avoidance for Wireless (MACAW)

Hidden and Exposed terminals

- **Hidden terminals** are senders that cannot sense each other but nonetheless collide at intended receiver
- Want to prevent; loss of efficiency
- A and C are hidden terminals when sending to B
- **Exposed terminals** are senders who can sense each other but still transmit safely (to different receivers)
 - Desirably concurrency; improves performance
 - B → A and C → D are exposed terminals



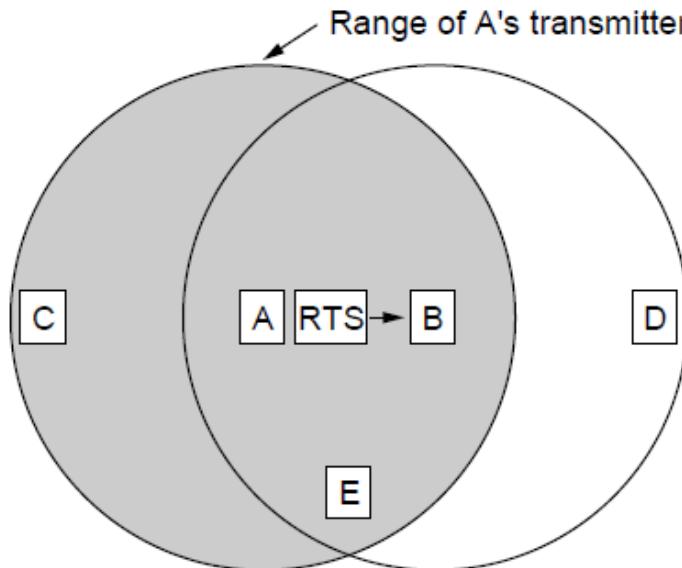
Multiple Access with Collision Avoidance (MACA)

- Sender asks receiver to transmit short control frame
- Stations near receiver hear control frame
- Sender can then transmit data to receiver

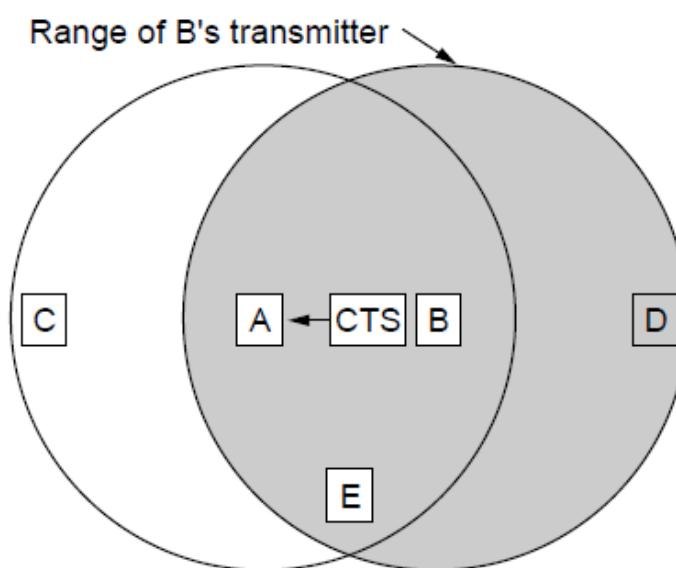
MACA

MACA protocol grants access for A to send to B:

- ❑ A sends RTS to B [left]; B replies with CTS [right]
- ❑ A can send with exposed but no hidden terminals



A sends RTS to B; C and E
hear and defer for CTS



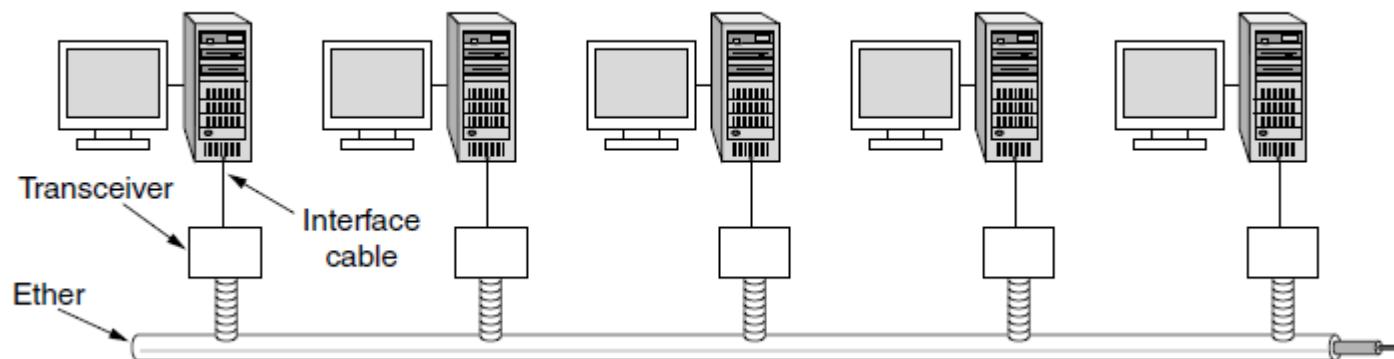
B replies with CTS; D and
E hear and defer for data

A MAC Sub-Layer Case Study: Ethernet

- Ethernet Frame Format
- MAC Addressing
- Ethernet Performance
- Switched Ethernet
- Fast Ethernet
- Gigabit Ethernet
- Ethernet in Retrospect

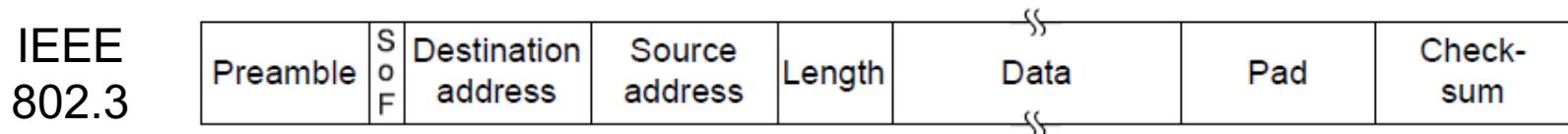
Classic Ethernet

- Each type of Ethernet has a maximum cable length per segment.
- Multiple cable lengths can be connected by repeaters - a physical device which receives, amplifies and retransmits signals in both directions.



Ethernet Frame Format

- MAC protocol is 1-persistent CSMA/CD (earlier)
 - Random delay (backoff) after collision is computed with BEB (Binary Exponential Backoff, i.e., random number 0 and $2^i - 1$)
 - Frame format is still used with modern Ethernet.



Preamble (7B) – synchronisation between sender and receiver

Start of Frame (1B) – FLAG bytes

Dest. & Source addresses – to identify who send, who receive

Type & Length (2B) – specifies which process to give the frame to (0x0800 means data contains IPv4)

Pad(0~46B) – Minimum size of the message of the Ethernet – 64 Bytes

CRC (4B) – 32 bits checksum

MAC ADDRESSING

- Source and Destination Addressing can be done at a local or global levels
- The **MAC Address** provides the unique identifier for a physical interface
- MAC Address is a 48-bit number encoded in the frame
 - eg 00:02:2D:66:7C:2C

Ethernet Performance

Definition

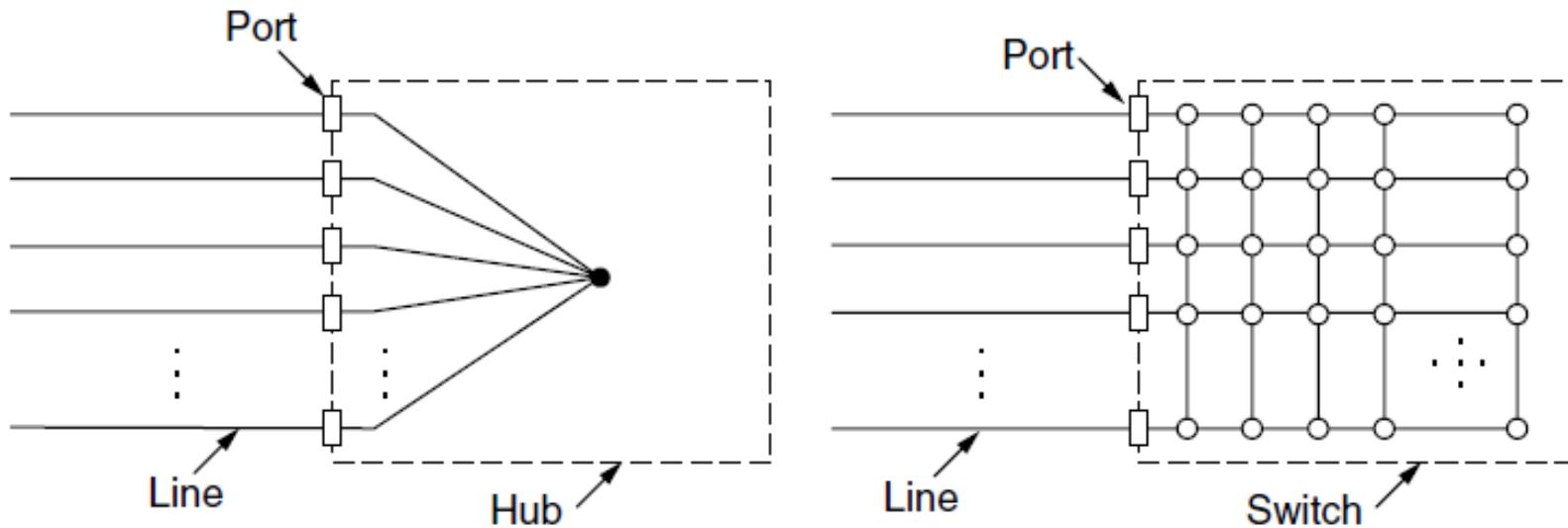
$$\text{Channel Efficiency} = \frac{1}{1 + 2BLe/cF}$$

- F : frame length
- B : bandwidth
- L : cable length
- c : speed of light
- optimal case of e contention slots per frame

- When cF is large, the channel efficiency will be high.
- Increasing network bandwidth or distance (BL) reduces the efficiency for a given frame size

Switched Ethernet

- Hubs wire all lines into a single CSMA/CD domain
- Switches isolate each port to a separate domain
 - Much greater throughput for multiple ports
 - No need for CSMA/CD with full-duplex lines



Summary

- MAC Sub-layer
 - Compare different CSMA schemes
 - Summarise collision free protocols
 - Explain for Wireless protocols
- Ethernet
 - Explain key features of Ethernet
 - Evaluate factors affecting Ethernet performance

Week 5 – Network Layer

COMP90007
Internet Technologies

Chien Aun Chan

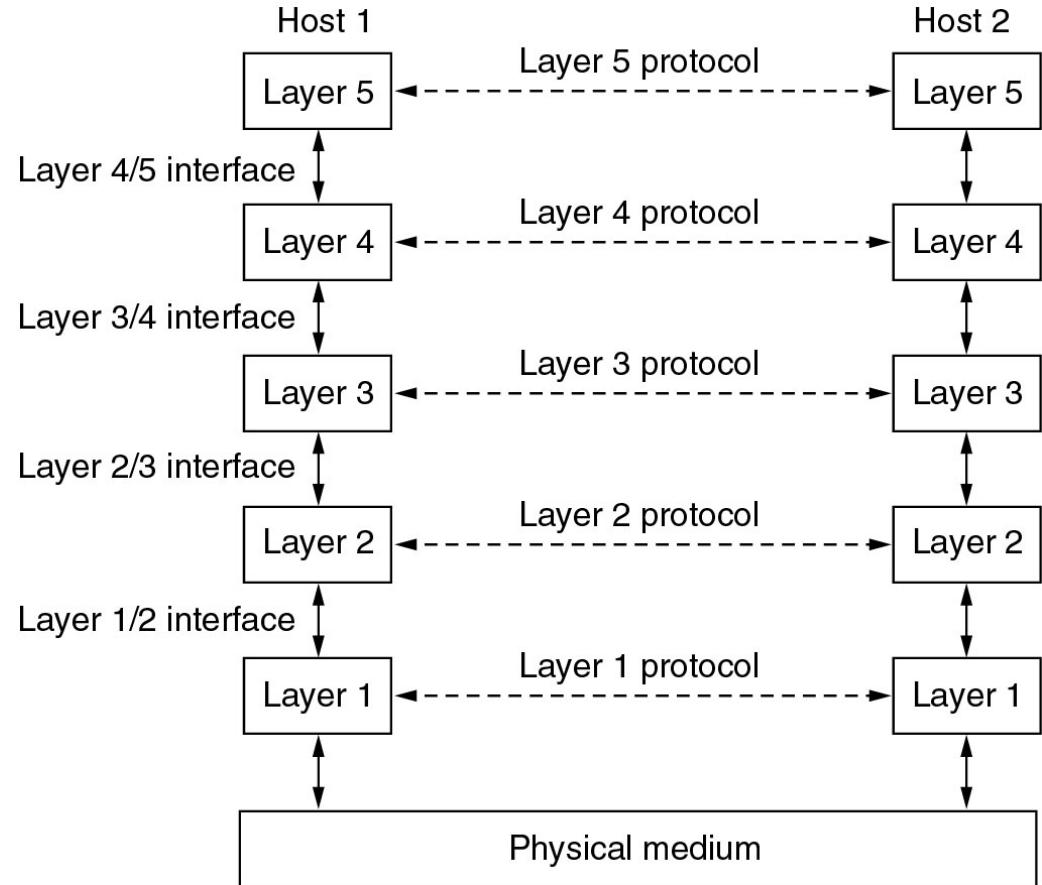
Network Software

Protocol Hierarchies

Connecting different networks
(internetworking)

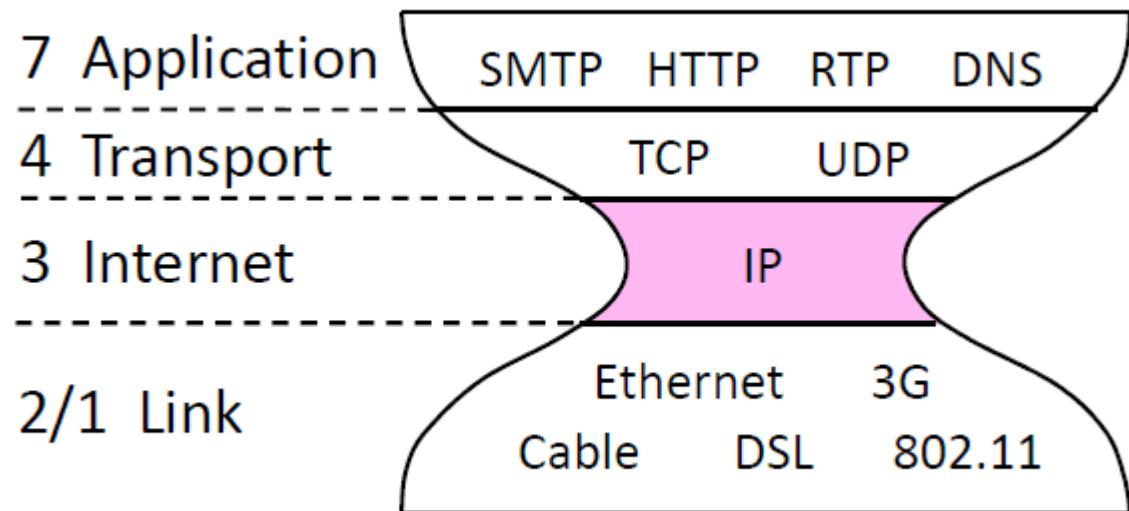
Framing, reliability and
flow control (direct conn.)

Different Cables, wireless
signalling digital to analogue



Outline

- Network Layer Design Issues
- Packet Forwarding
- Internetworking
- IPv4
- Routing Algorithms

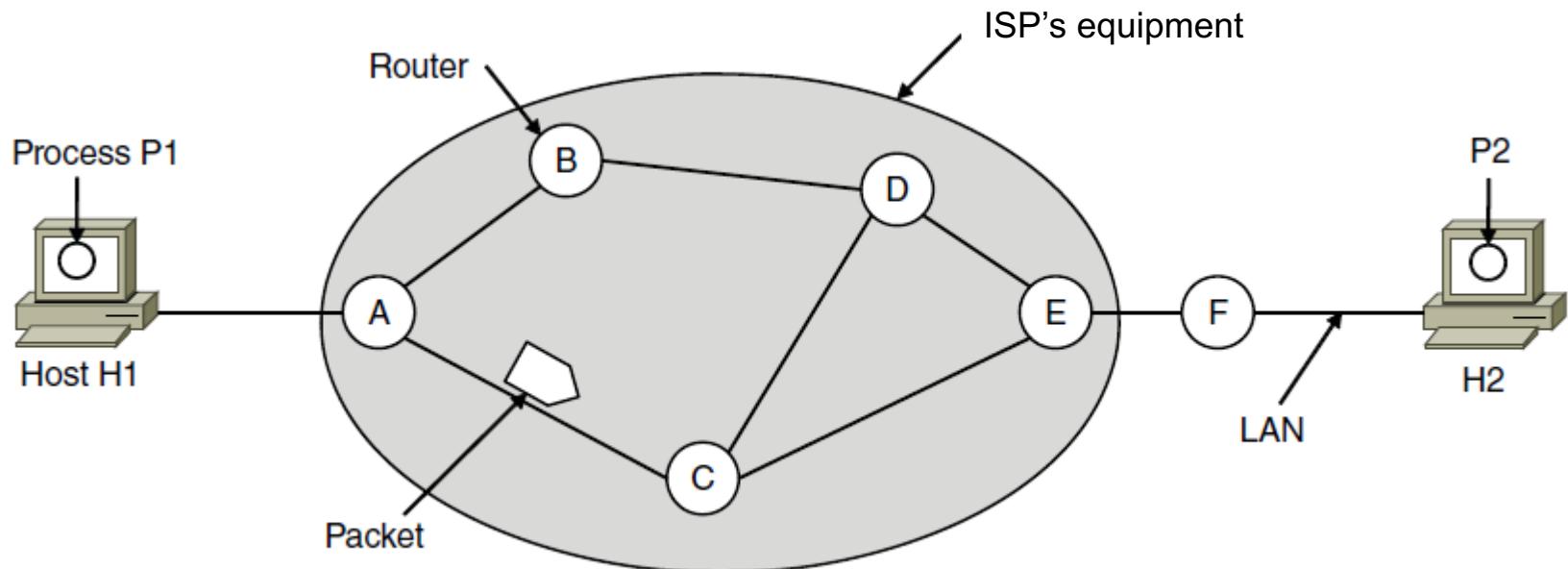


Network Layer Design Issues

- Which services are provided to the transport layer ?
- How do design issues affect performance ?
 - Store-and-forward packet switching
 - Connectionless service – datagrams
 - Connection-oriented service – virtual circuits
- Comparison of virtual-circuits and datagrams

Store and Forward Packet Switching

- Hosts generate packets and injects into the network
- Routers treat packets as messages, receiving (storing) them and then forwarding them based on how the message is addressed
- Router routes packets through the network



Services Provided to the Transport Layer

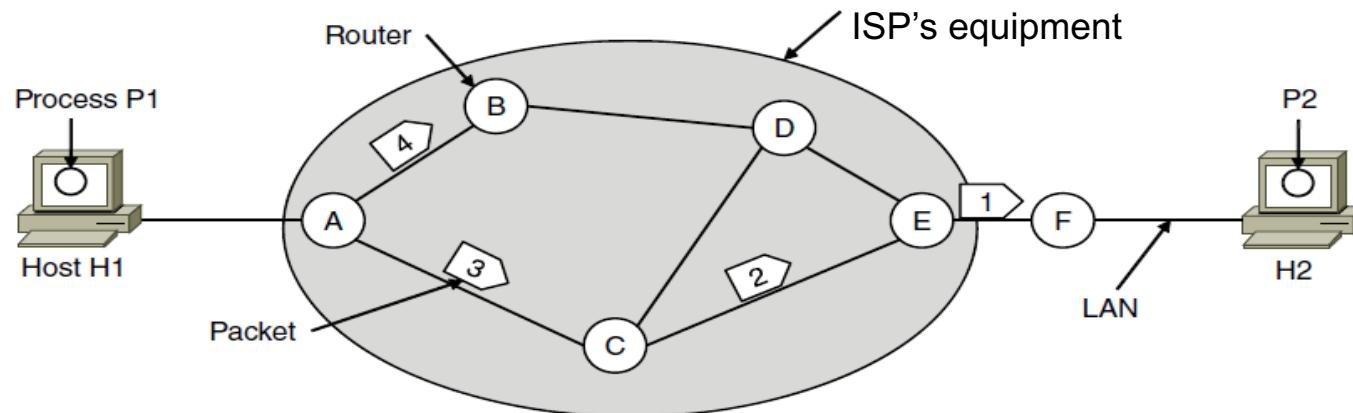
- Design goals:
 - Services should be independent of router technologies
 - Transport layer should be shielded from number, type and topology of routers
 - Network addressing should use a uniform numbering plan (network identifier)

Types of Services

1. **Connectionless:** Packets (datagrams) injected into subnet independently and packets individually routed to destination
 - ❑ Internet: move packets in a potentially unreliable subnet - QoS is not easily implemented
 - ❑ Flow and error control done by the hosts
2. **Connection-oriented:** Packets travelling between destinations all use the same route
 - ❑ Telco: guarantee reliability of subnet - QoS is important

Routing within a datagram subnet

- Post office model: packets are routed individually based on destination addresses in them
- Packets can take different paths
- E.g., P1 sends a long message to P2



A's table (initially)

A	☒
B	B
C	C
D	B
E	C
F	C

A's table (later)

A	☒
B	B
C	C
D	B
E	B
F	B

C's Table

A	A
B	A
C	☒
D	E
E	E
F	E

E's Table

A	C
B	D
C	C
D	D
E	☒
F	F

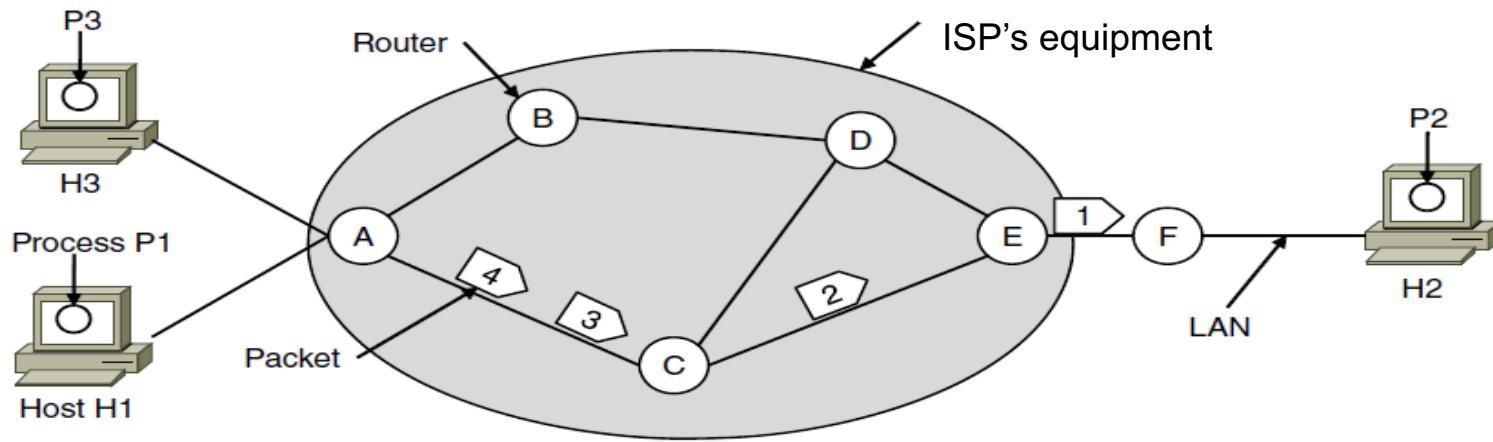
Routing table (can be fixed, can change over time)

Routing algorithm – manages the routing table

Routing within a virtual-circuit subnet

- Model is like telephone network

- Packets are routed through virtual circuits (created earlier) based on tag number (not full address but unique at a given link) in them
- Packets take the same path (to avoid having to choose a new route for every packet sent)
- E.g., Multi-protocol Label Switching Network (to provide QoS) – 20 bit label or conn. Identifier



connection identifier

A's table

H1		1
C		1

In Out

C's Table

A		1
E		1

E's Table

C		1
F		1

Differences in Virtual Circuit and Datagram Subnets

Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Compromises in VC and Datagram Subnets

- Compromises:

- Memory vs bandwidth
 - VC's require more space in router memory, and save potential overhead in full addressing of each packet
 - Setup time vs address parsing time
 - VC's require setup time and resources, but packet transmission is very fast
 - Amount of memory
 - datagram subnets require large tables of every possible destination routes, whereas VC does not. Really?
 - QoS and congestion avoidance
 - VC's can use a tighter QoS - able to reserve CPU, bandwidth and buffer in advance
 - Longevity
 - VC's can exist for a long time eg Permanent VC's
 - Vulnerability
 - VC's particularly vulnerable to hardware/software crashes - all VC's aborted and no traffic until they are rebuilt; datagram uses an alternative route

Outline

- Network Layer Design Issues
- Packet Forwarding
- Internetworking
- IPv4
- Routing Algorithms

Internetworking

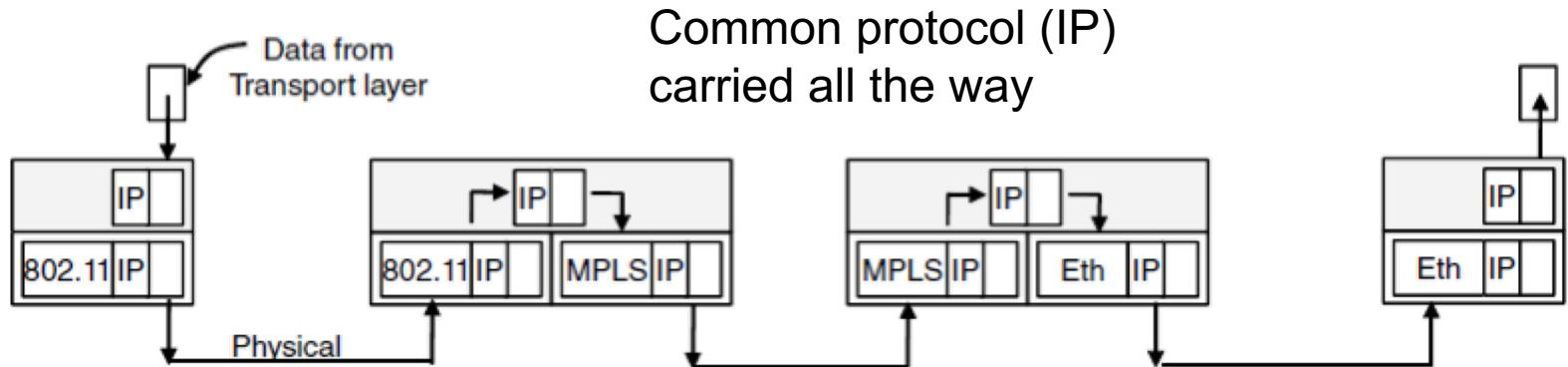
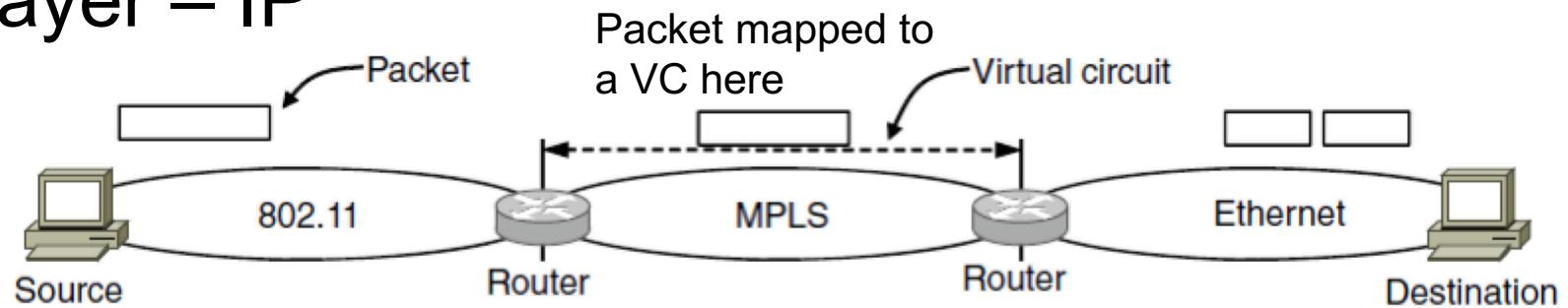
- Until now we assumed a single homogeneous network
- Internetworking joins multiple, different networks into a single larger network
- Issues when connecting networks:
 - Different network types and protocols
 - Different motivations for network choices
 - Different technologies at both hardware and software levels

Differences at the Network Layer

Item	Some Possibilities
Service offered	Connectionless versus connection oriented
Addressing	Different sizes, flat or hierarchical
Broadcasting	Present or absent (also multicast)
Packet size	Every network has its own maximum
Ordering	Ordered and unordered delivery
Quality of service	Present or absent; many different kinds
Reliability	Different levels of loss
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, packet, byte, or not at all

How Different Networks are Connected

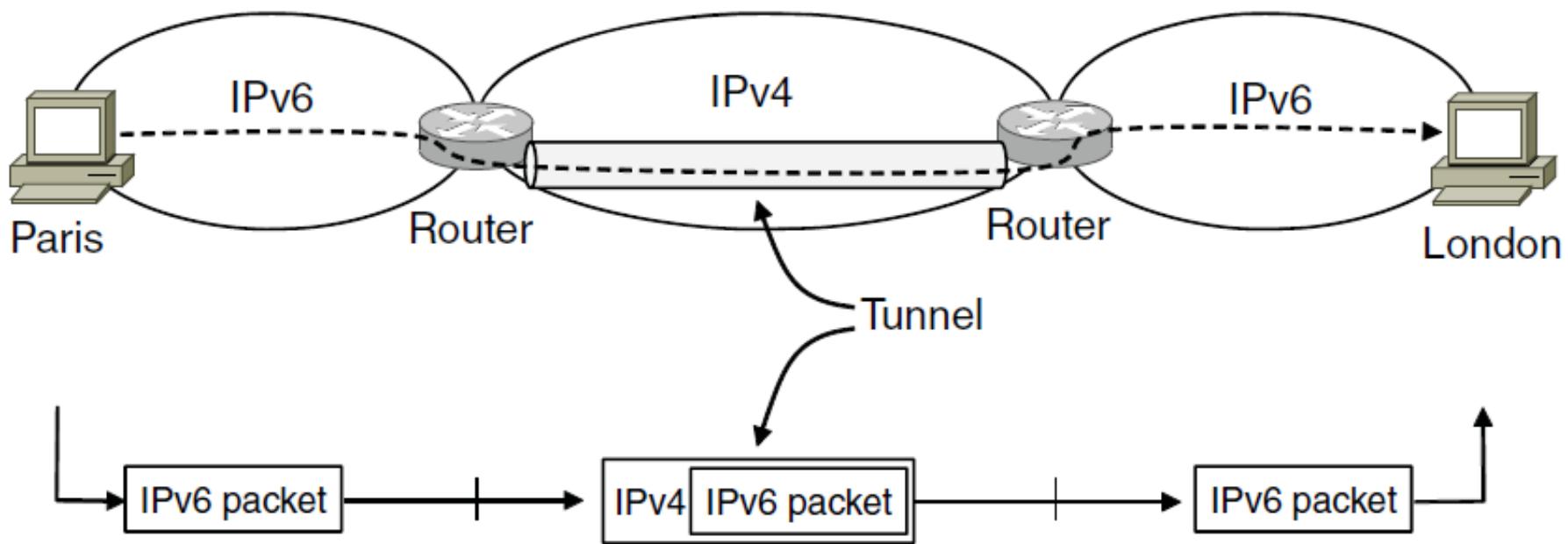
- Internetworking based on a common network layer – IP



Tunneling

- Tunneling is used when the source and destination are on the same network, but there is a different network in between.
 - Source Packets are encapsulated over the packets in the connecting network

Tunneling IPv6 packets through IPv4

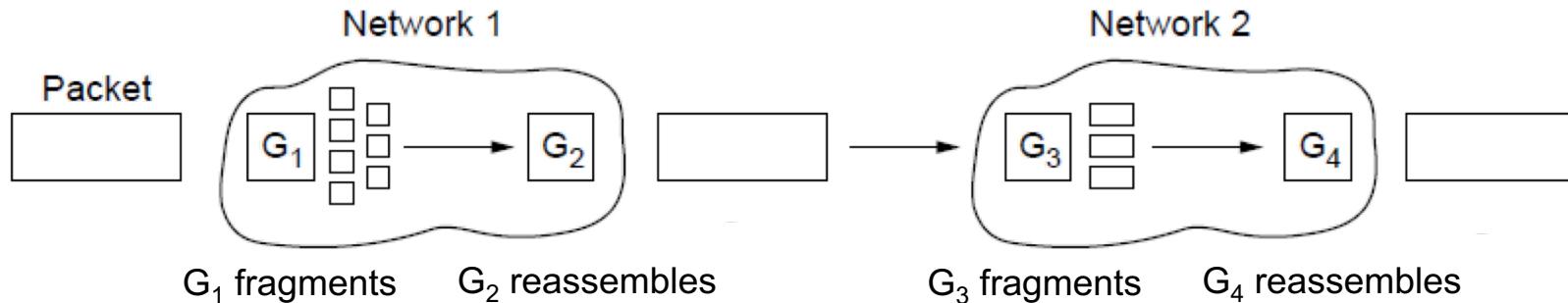


Fragmentation

- All networks have a maximum size for packets, could be motivated by:
 - Hardware
 - Operating system
 - Protocols
 - Standards compliance
 - Desire to reduce transmissions due to errors
 - Desire for efficiency in communication channel
- **Fragmentation** (division of packets into fragments) allows network gateways to meet size constraints

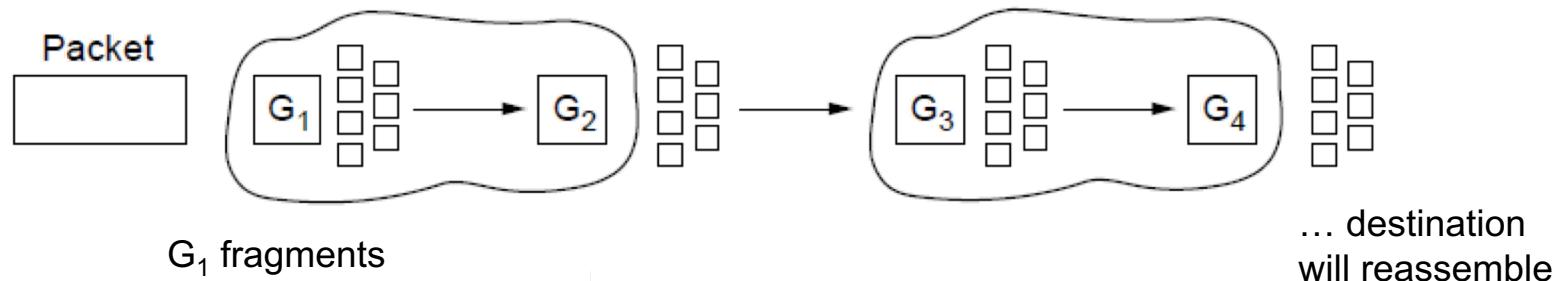
Types of Fragmentation

- Large packets need to be routed through a network whose maximum packet size is too small.
- **Fragmentation and Reassembly is a solution.**



Transparent – packets fragmented / reassembled in each network

- Route constrained, more work



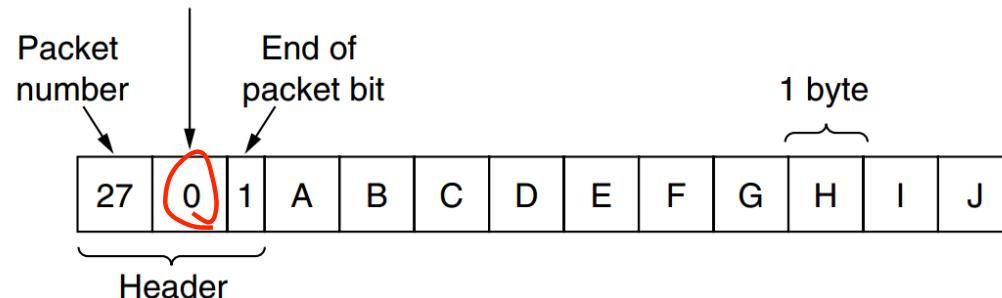
Non-transparent – fragments are reassembled at destination

- Less work (IP works this way) – packet number, byte offset, end of packet flag

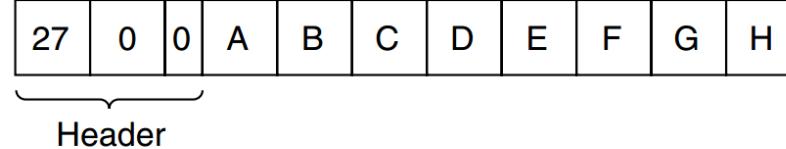
Example: IP Style Fragmentation

Original packet:
(10 data bytes)

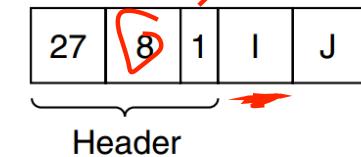
Number of the first elementary fragment in this packet



Fragmented:
(to 8 data bytes)

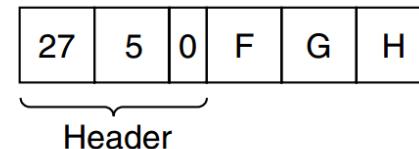
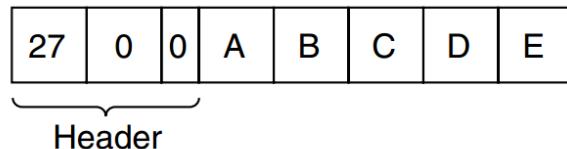


(a)

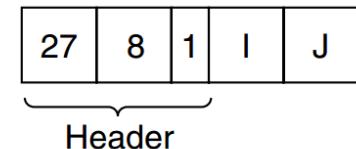


(b)

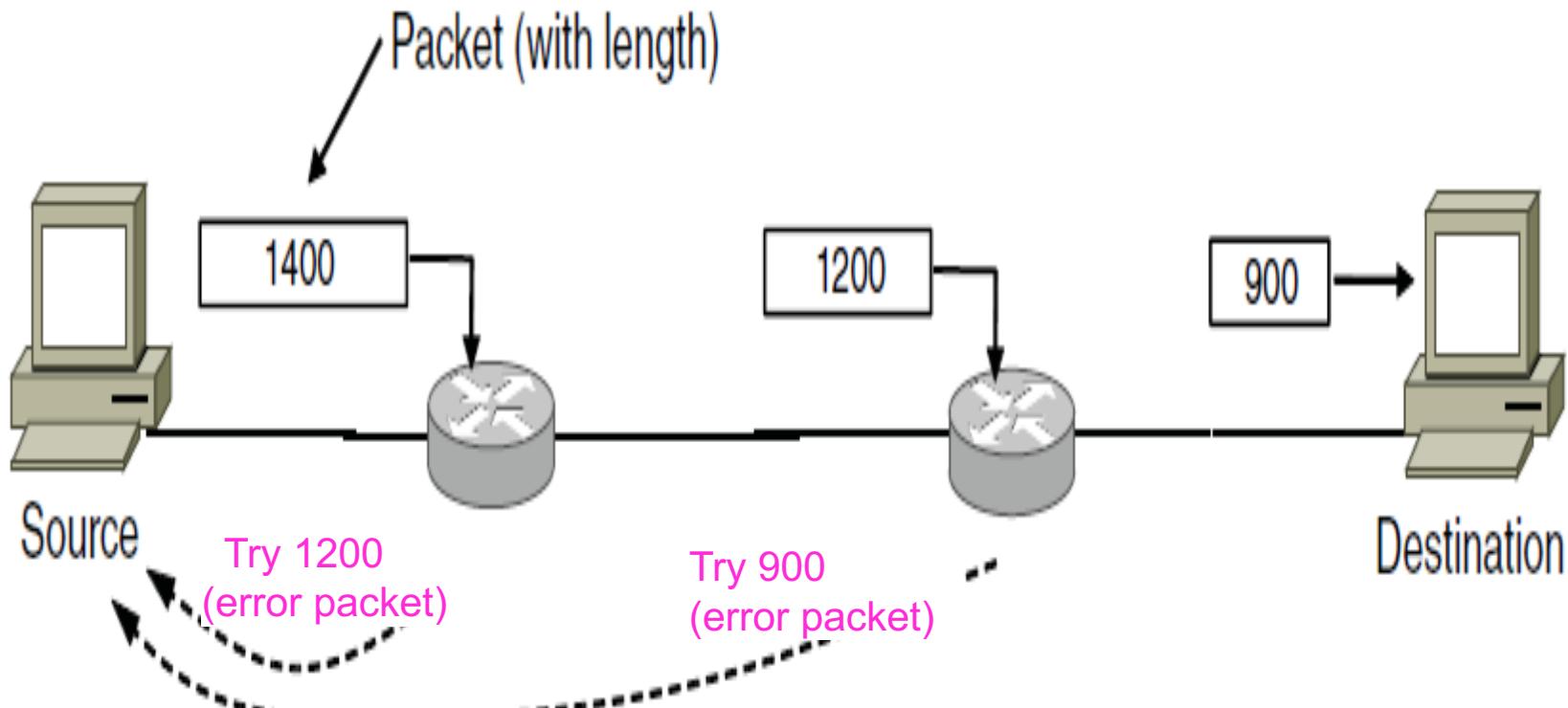
Re-fragmented:
(to 5 bytes)



(c)



Path MTU Discovery: Alternative to Fragmentation



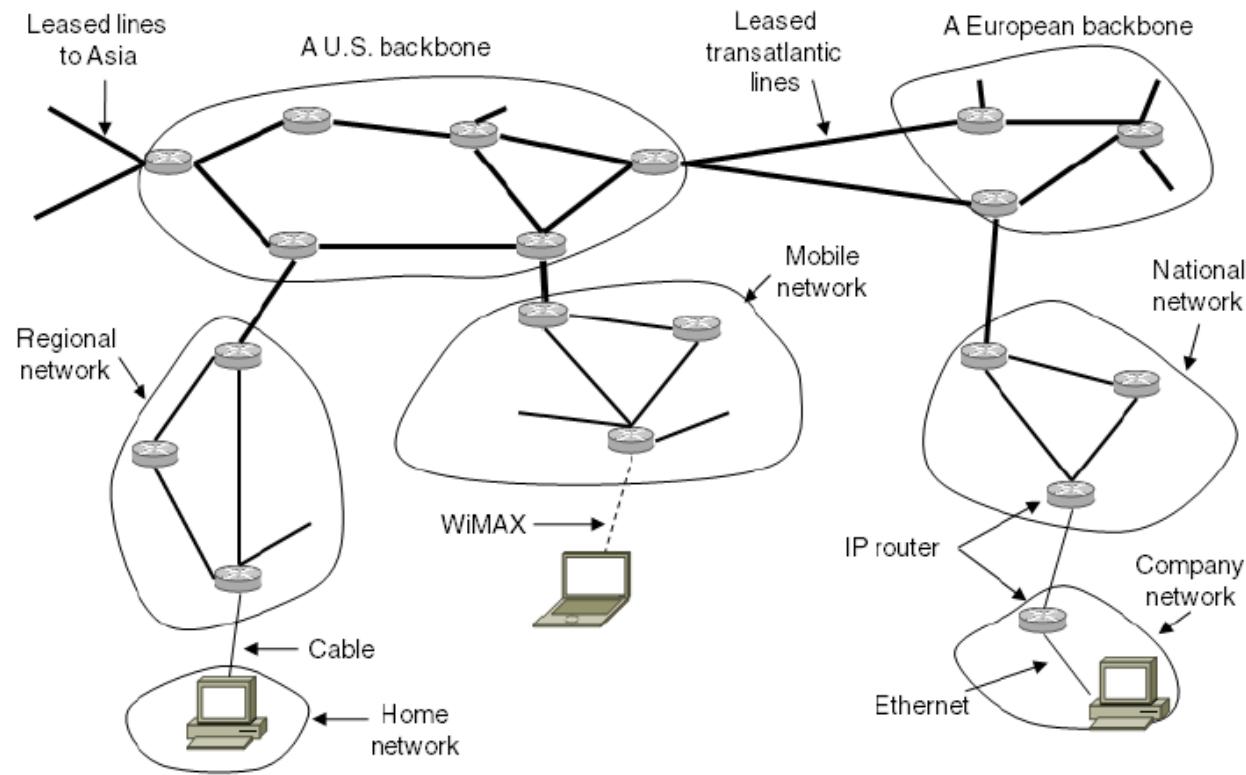
Advantage: The source now knows what length packet to send but if the routes and path MTU change, new error packets will be triggered and the source will adapt to the new path

Principles of Internet Design

- RFC 1958 “Architectural Principles of the Internet” (supplementary reading)
- Core Principles
 - Make sure it works
 - Keep it simple
 - Make clear choices
 - Exploit modularity
 - Expect heterogeneity
 - Avoid static options and parameters
 - Choose a good, but not necessarily perfect design
 - Be strict in sending and tolerant in receiving
 - Consider scalability
 - Consider performance vs costs

Network Layer in the Internet

- Internet is an interconnected collection of many networks of Autonomous systems that is held together by the IP protocol

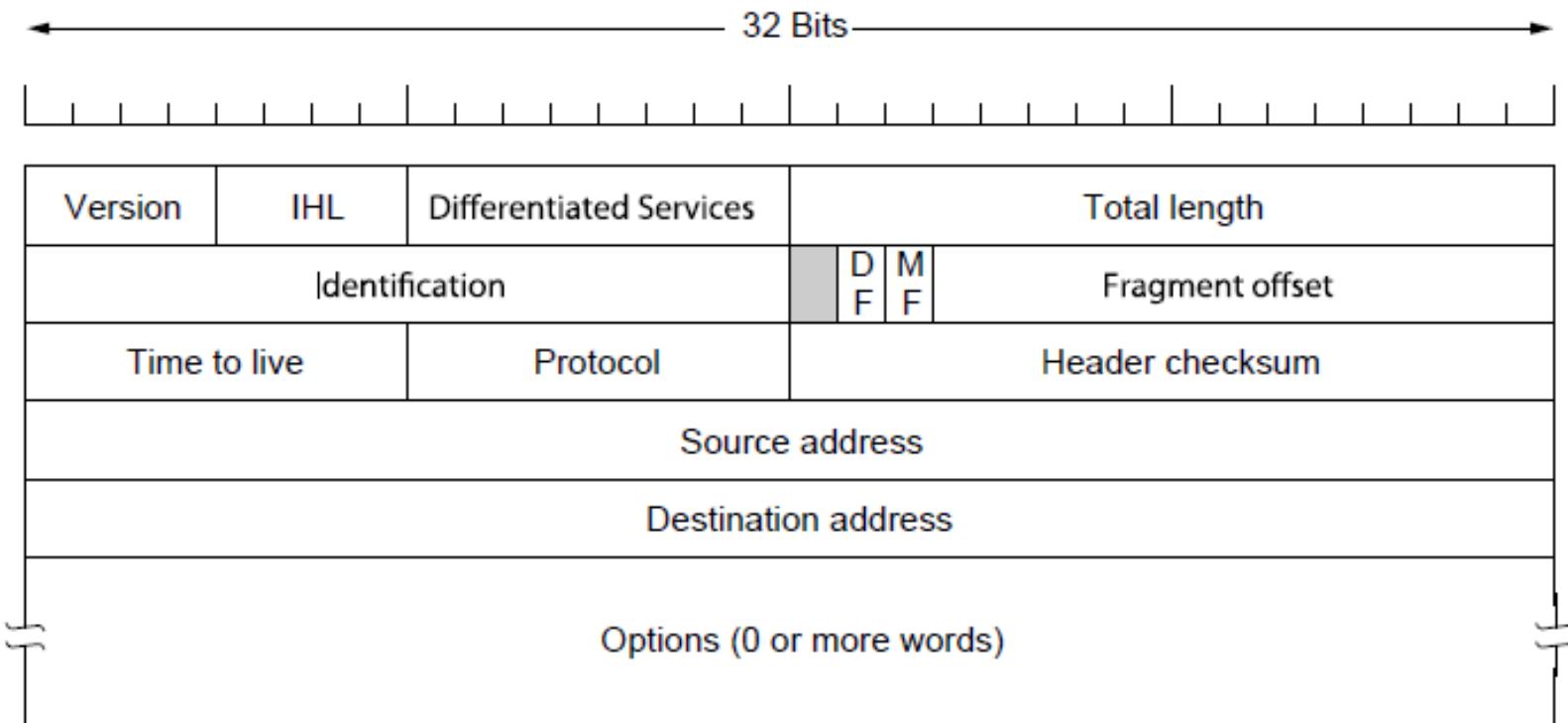


Internet Protocol (IP)

- The glue that holds the whole Internet together is the network layer protocol, IP (Internet Protocol)
- Provides a “best-effort” service to route datagrams from source host to destination host
- These hosts may be
 - On **same** network
 - On **different** networks
- Each network is called an **Autonomous System (AS)**

IPv4 Frame Structure Illustrated

- IPv4 (Internet Protocol) header is carried on all packets and has fields for the key parts of the protocol



IPv4 Frame Structure in Detail

- IPv4 Frame consists of a header and some text
- header is 20 byte fixed part + variable length optional part
- Version: IPv4 or IPv6
- IHL: Header Length – in 32bits units, min 5 and max is 15
- Type: differentiates different classes of service
- Total Length: header and payload, maximum length 65535 bytes
- Identification: allows host to determine which datagram the new fragment belongs to - all fragments of same datagram have same ID
- DF: Don't Fragment byte
 - Originally, it was intended to support hosts incapable of putting the pieces back together again.
 - Now it is used as part of the process to discover the path MTU, which is the largest packet that can travel along a path without being fragmented

IPv4 Frame Structure in Detail (continued)

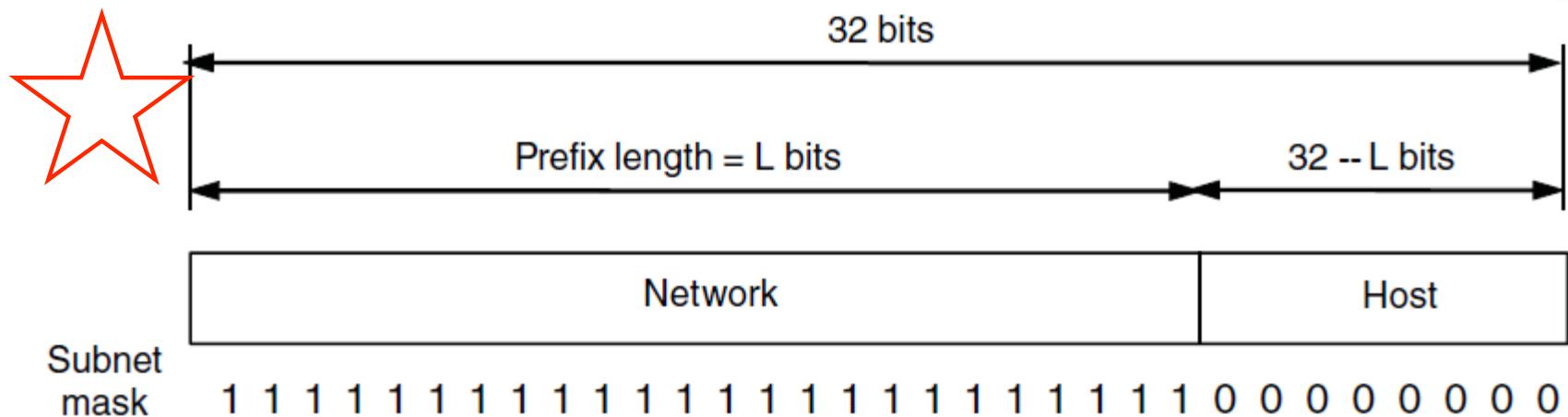
- MF: More Fragment byte - are there more or is this the last one ?
- Fragment offset: where in the datagram the current fragment belongs
- TTL: limits packet lifetimes - hops or seconds
- Protocol: TCP, UDP, others ...
- Header Checksum: verifies the header only
- Source Address: IP - host/network
- Destination Address: IP - host/network
- Options: eg security, strict vs loose source routing, record route, timestamp

IP Addresses

Example:

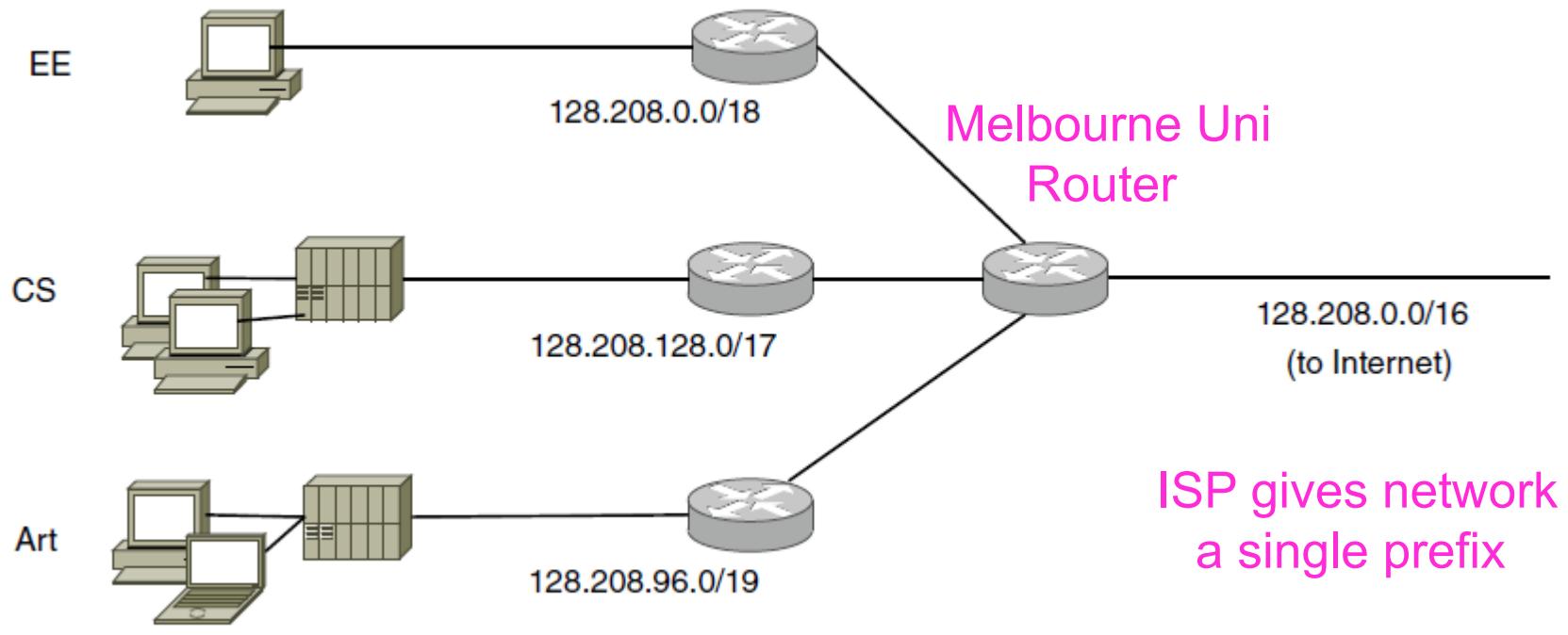
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	0	0	1	0	0	1	0

- Addresses are allocated in blocks called prefixes
 - Prefix is determined by the network portion
 - IP addresses are written in dotted decimal notation
 - Written lowest address/length, e.g., 18.0.31.0/24
- Overall IP allocation responsibility of Internet Corporation for Assigned Names and Numbers (ICANN) by delegation to IANA and Regional Internet Registries (RIR's)



Subnets

- Subnetting allows networks to be split into several parts for internal uses whilst acting like a single network for external use
 - Looks like a single prefix outside the network

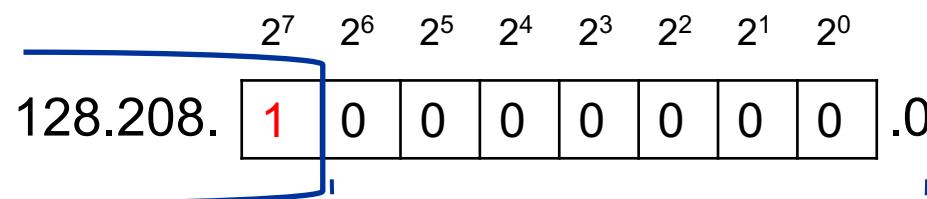
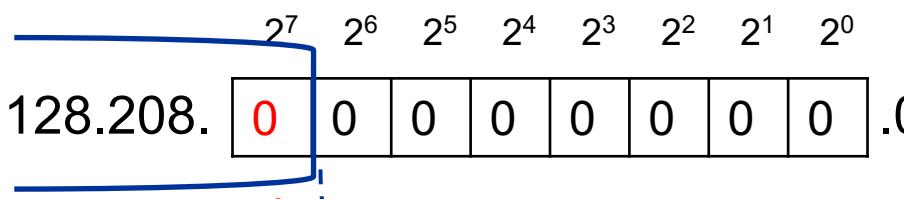


Network divides it into subnets internally

Example

128.208.0.0/16

Given network prefix

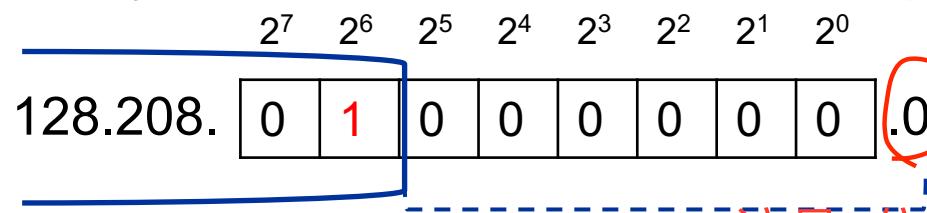
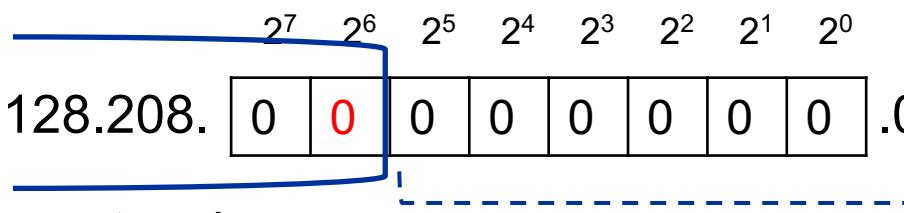


network=17bits

128.208.0.0/17

CS

Number of addresses: 2^{15}
 $2^7 * 2^8$



network一共18位

EE

128.208.0.0/18

Further split

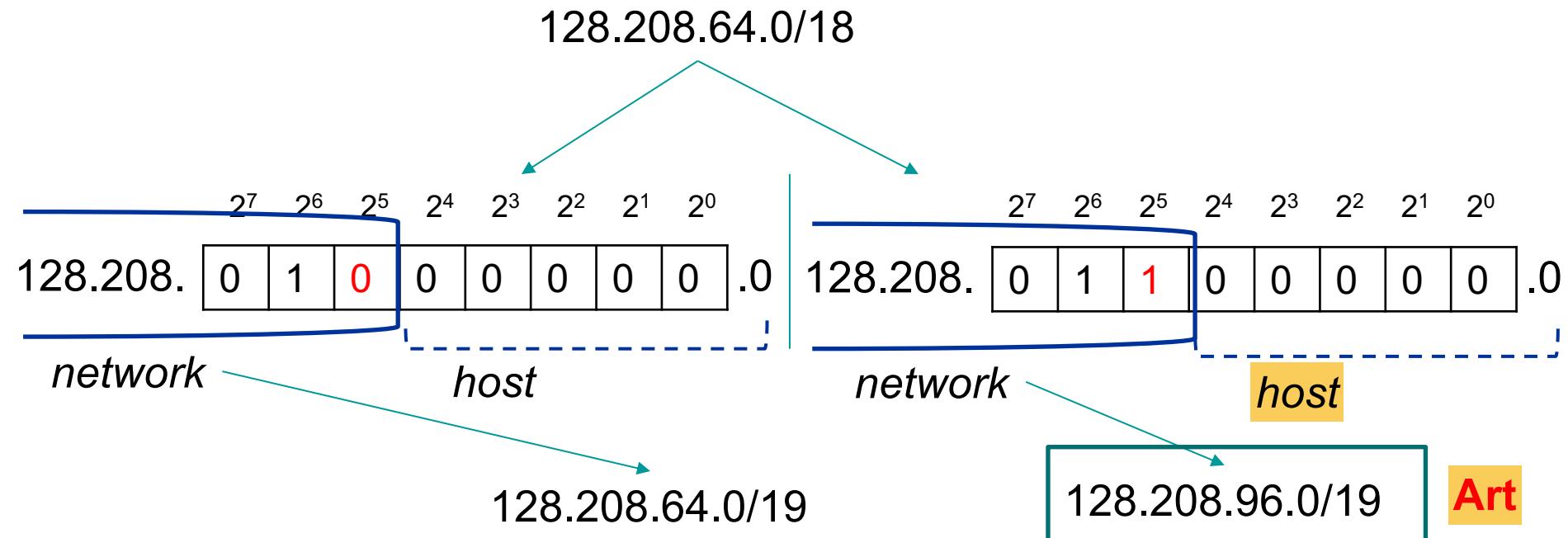
128.208.64.0/18

Number of addresses: 2^{14}

$2^6 * 2^8$

这是8位

Example (cont.)



Number of addresses: 2^{13}
32bits in total
netwotk takes
19bits, so 13bits
left, 2^{13}

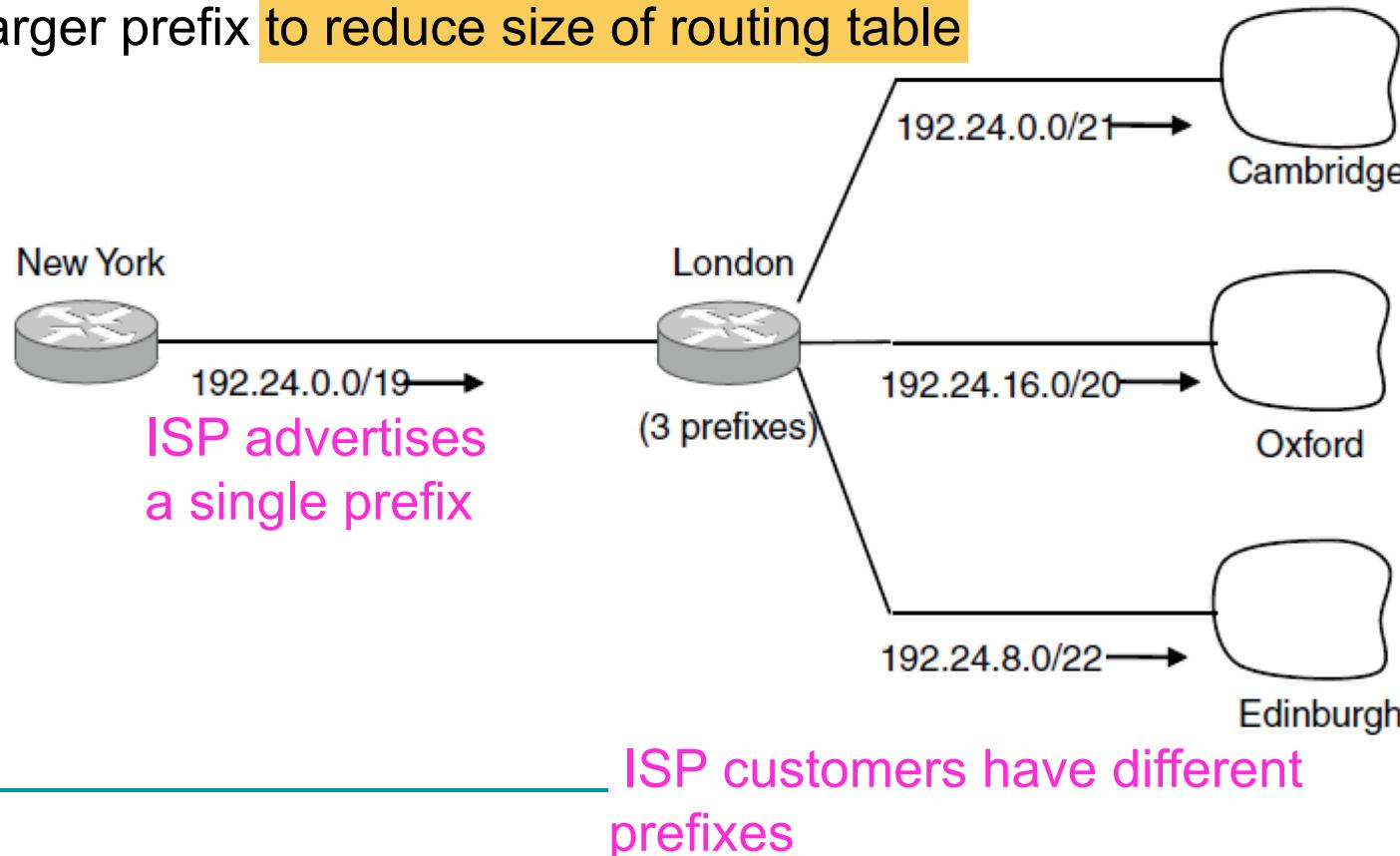
IP Addressing and Routing Tables

- Routing tables are typically based around a triplet:
 - IP Address
 - Subnet Mask
 - Outgoing Line (physical or virtual)
- Eg: A row of a routing table:

Prefix addr	Subnet Mask	Interface
203.32.8.0	255.255.255.0	Eth 0

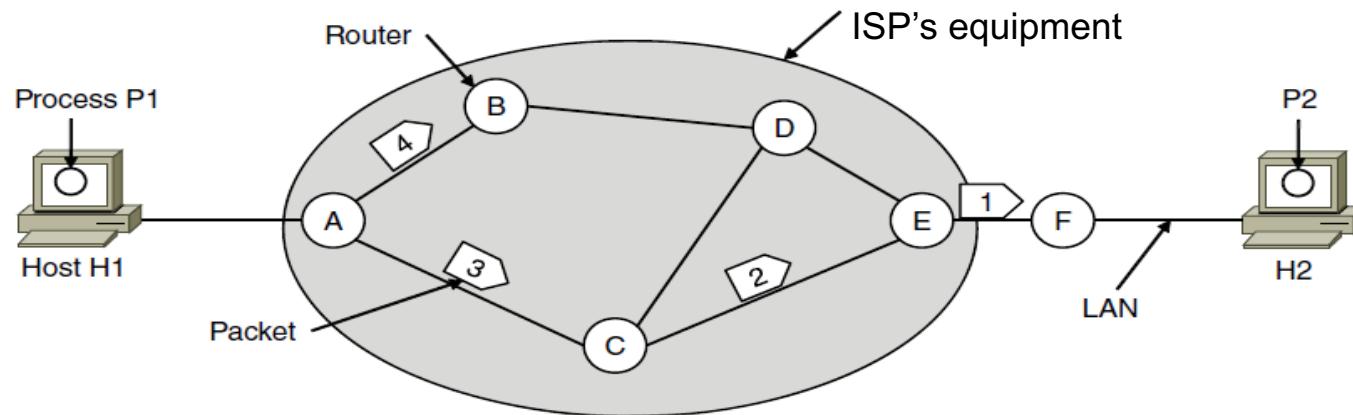
Aggregation of IP addresses

- Backbone router connecting networks around the world → 300k networks
- So we search each line for each incoming packet?
- Aggregation – Process of joining multiple IP prefixes into a single larger prefix **to reduce size of routing table**



Reminder: Routing within a datagram subnet

- Post office model: packets are routed individually based on destination addresses in them
- Packets can take different paths

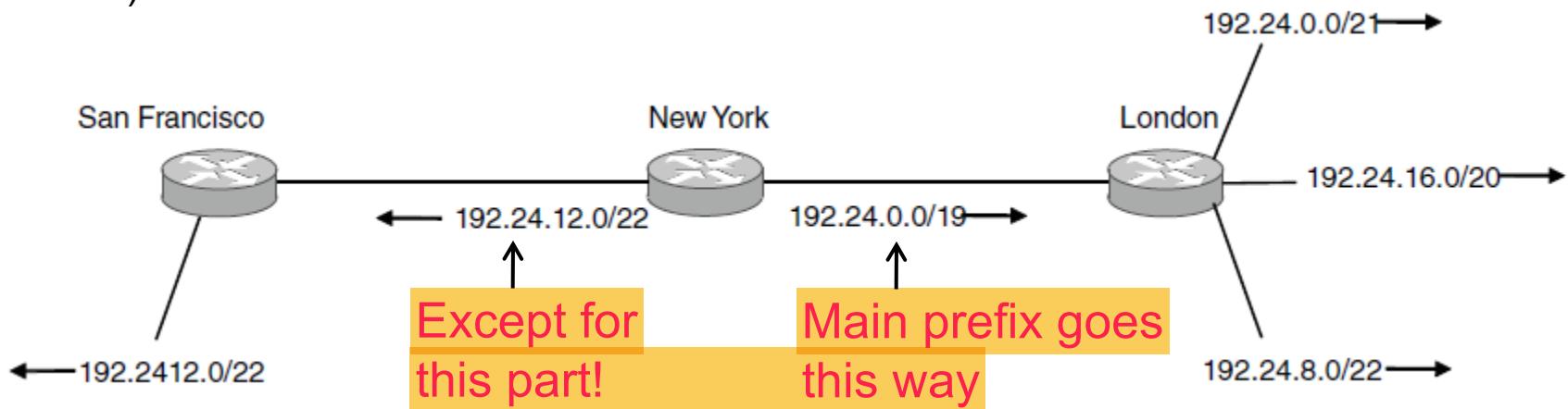


A's table (initially)	A's table (later)	C's Table	E's Table																																																
<table border="1"><tr><td>A</td><td>X</td></tr><tr><td>B</td><td>B</td></tr><tr><td>C</td><td>C</td></tr><tr><td>D</td><td>B</td></tr><tr><td>E</td><td>C</td></tr><tr><td>F</td><td>C</td></tr></table>	A	X	B	B	C	C	D	B	E	C	F	C	<table border="1"><tr><td>A</td><td>X</td></tr><tr><td>B</td><td>B</td></tr><tr><td>C</td><td>C</td></tr><tr><td>D</td><td>B</td></tr><tr><td>E</td><td>B</td></tr><tr><td>F</td><td>B</td></tr></table>	A	X	B	B	C	C	D	B	E	B	F	B	<table border="1"><tr><td>A</td><td>A</td></tr><tr><td>B</td><td>A</td></tr><tr><td>C</td><td>X</td></tr><tr><td>D</td><td>E</td></tr><tr><td>E</td><td>E</td></tr><tr><td>F</td><td>E</td></tr></table>	A	A	B	A	C	X	D	E	E	E	F	E	<table border="1"><tr><td>A</td><td>C</td></tr><tr><td>B</td><td>D</td></tr><tr><td>C</td><td>C</td></tr><tr><td>D</td><td>D</td></tr><tr><td>E</td><td>X</td></tr><tr><td>F</td><td>F</td></tr></table>	A	C	B	D	C	C	D	D	E	X	F	F
A	X																																																		
B	B																																																		
C	C																																																		
D	B																																																		
E	C																																																		
F	C																																																		
A	X																																																		
B	B																																																		
C	C																																																		
D	B																																																		
E	B																																																		
F	B																																																		
A	A																																																		
B	A																																																		
C	X																																																		
D	E																																																		
E	E																																																		
F	E																																																		
A	C																																																		
B	D																																																		
C	C																																																		
D	D																																																		
E	X																																																		
F	F																																																		

Dest. Line

Longest Matching Prefix

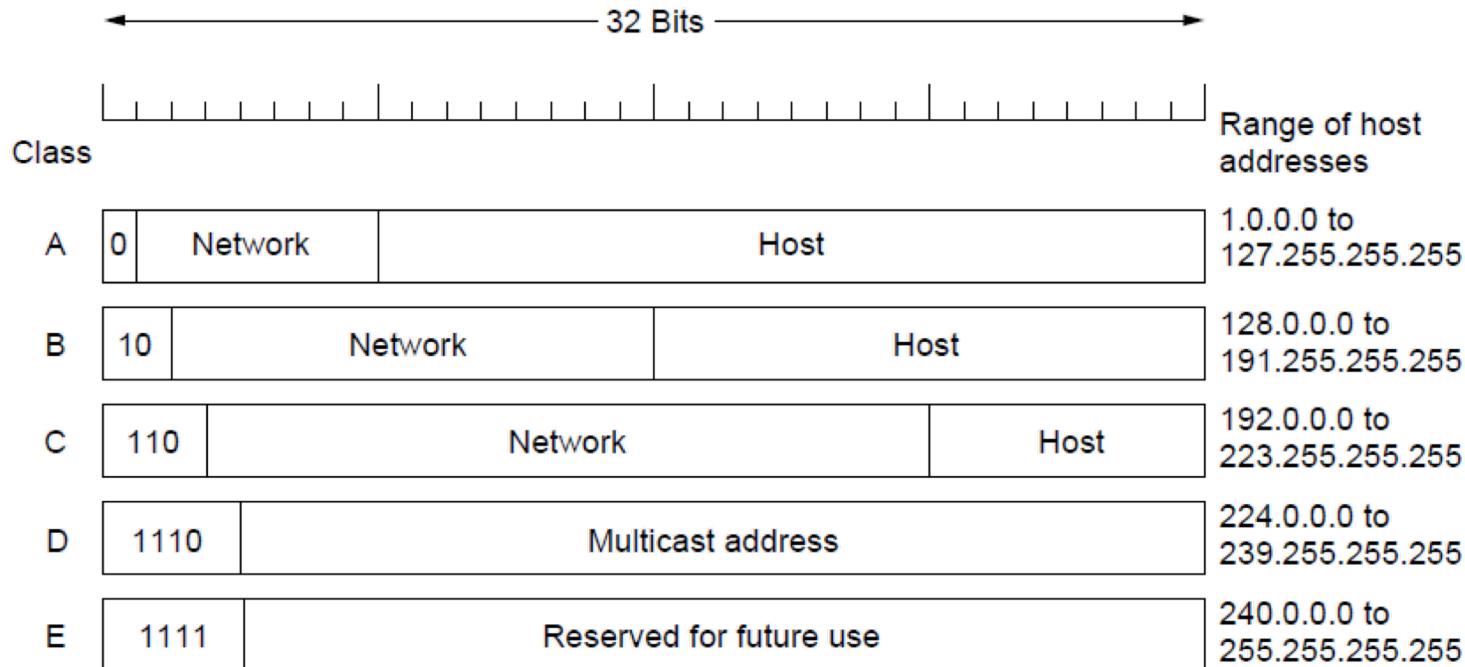
- Packets are forwarded to the entry with the longest matching prefix or smallest address block
 - Complicates forwarding but adds flexibility
- 1) Check address whether matches the longest prefix → /22
 - 2) If not the see if it matches /19



Prefix addr	Subnet Mask	Interface
192.24.12.0	255.255.252.0	Eth 0
192.24.0.0	255.255.224.0	Eth 1

Classful Addressing

- Part of history now-old addresses came in blocks of fixed size (A, B, C)
 - Carries size as part of address, but lacks flexibility
 - Called classful (vs. classless) addressing

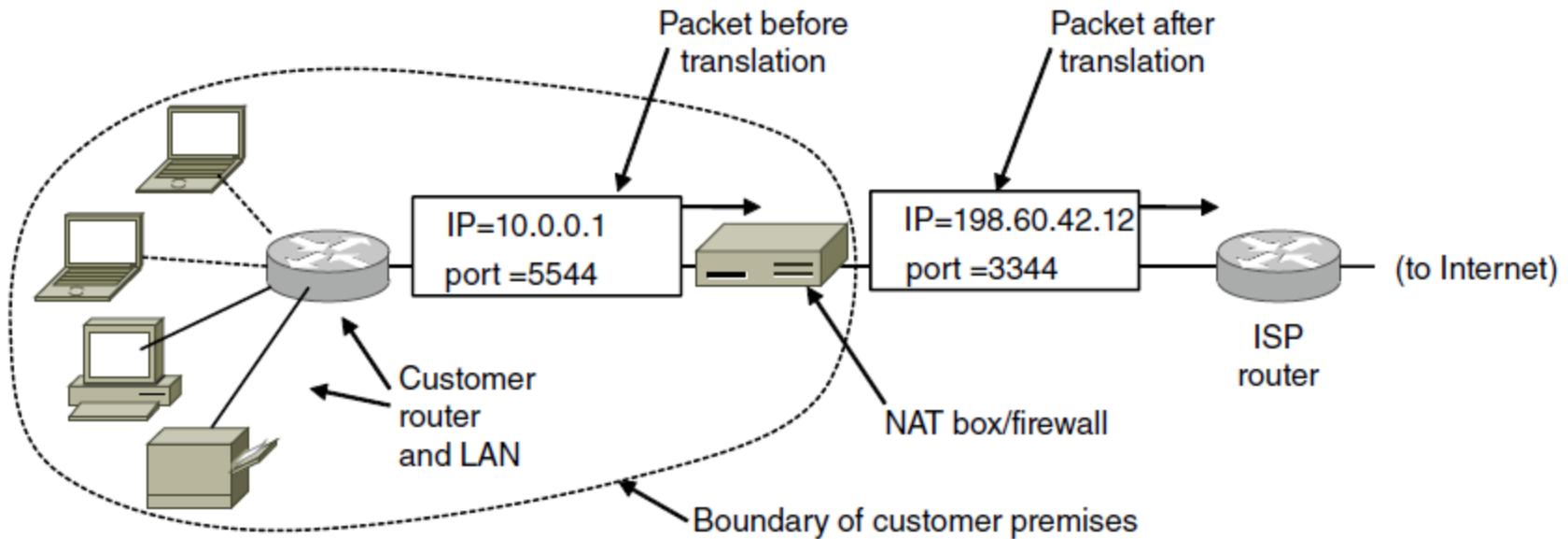


Private IP Ranges

- Range of IP addresses that CANNOT appear in the Internet
- Only for private networks
- 10.0.0.0/8 (16,777,216 hosts)
- 172.16.0.0/12 (1,048,576 hosts)
- 192.168.0.0 /16 (65,536 hosts)

Network Address Translation (NAT)

- NAT box maps one external IP address to many internal IP addresses
 - Uses TCP/UDP port to tell connections apart
 - Violates layering; very common in homes, etc.
- <https://youtu.be/QBqPzHEDzvo>



Internet Control Protocols

- IP works with the help of several control protocols:
 - ICMP is a companion to IP that returns error info
 - Required, and used in many ways, e.g., for traceroute
 - ARP finds MAC address of a local IP address
 - Glue that is needed to send any IP packets
 - Host queries an address and the owner replies
 - DHCP assigns a local IP address to a host
 - Gets host started by automatically configuring it
 - Host sends request to server, which grants a lease

ICMP

- Internet Control Message Protocol
- Used for testing and monitoring ambient conditions between hosts and routers

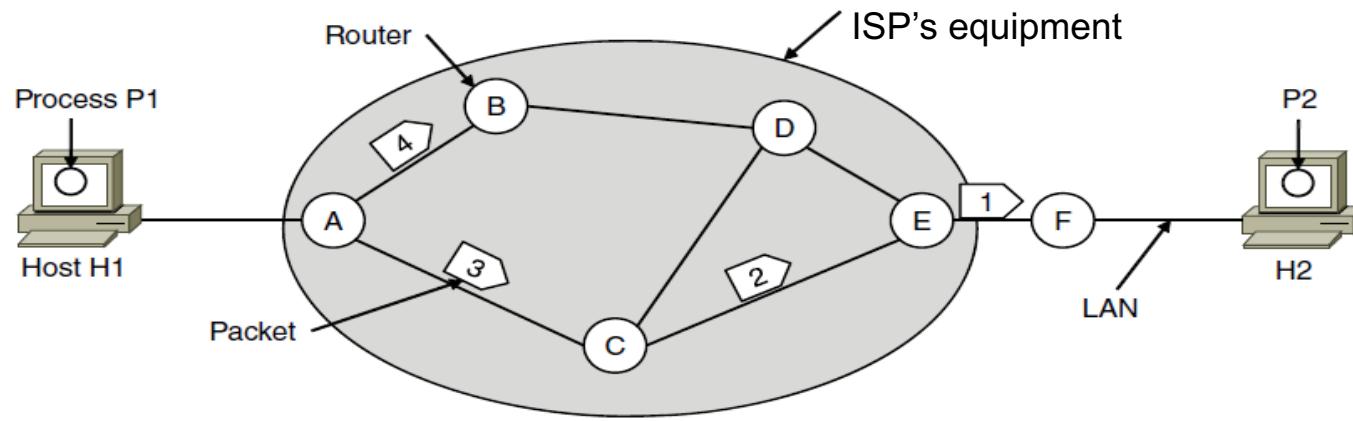
Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo and Echo reply	Check if a machine is alive
Timestamp request/reply	Same as Echo, but with timestamp
Router advertisement/solicitation	Find a nearby router

Outline

- Routing Algorithms
- Routing and Routing Control
- Quality of Service

Reminder: Routing within a datagram subnet

- Post office model: packets are routed individually based on destination addresses in them
- Packets can take different paths



A's table (initially)	A's table (later)	C's Table	E's Table																																																
<table border="1"><tr><td>A</td><td>X</td></tr><tr><td>B</td><td>B</td></tr><tr><td>C</td><td>C</td></tr><tr><td>D</td><td>B</td></tr><tr><td>E</td><td>C</td></tr><tr><td>F</td><td>C</td></tr></table>	A	X	B	B	C	C	D	B	E	C	F	C	<table border="1"><tr><td>A</td><td>X</td></tr><tr><td>B</td><td>B</td></tr><tr><td>C</td><td>C</td></tr><tr><td>D</td><td>B</td></tr><tr><td>E</td><td>B</td></tr><tr><td>F</td><td>B</td></tr></table>	A	X	B	B	C	C	D	B	E	B	F	B	<table border="1"><tr><td>A</td><td>A</td></tr><tr><td>B</td><td>A</td></tr><tr><td>C</td><td>X</td></tr><tr><td>D</td><td>E</td></tr><tr><td>E</td><td>E</td></tr><tr><td>F</td><td>E</td></tr></table>	A	A	B	A	C	X	D	E	E	E	F	E	<table border="1"><tr><td>A</td><td>C</td></tr><tr><td>B</td><td>D</td></tr><tr><td>C</td><td>C</td></tr><tr><td>D</td><td>D</td></tr><tr><td>E</td><td>X</td></tr><tr><td>F</td><td>F</td></tr></table>	A	C	B	D	C	C	D	D	E	X	F	F
A	X																																																		
B	B																																																		
C	C																																																		
D	B																																																		
E	C																																																		
F	C																																																		
A	X																																																		
B	B																																																		
C	C																																																		
D	B																																																		
E	B																																																		
F	B																																																		
A	A																																																		
B	A																																																		
C	X																																																		
D	E																																																		
E	E																																																		
F	E																																																		
A	C																																																		
B	D																																																		
C	C																																																		
D	D																																																		
E	X																																																		
F	F																																																		

Routing Algorithms

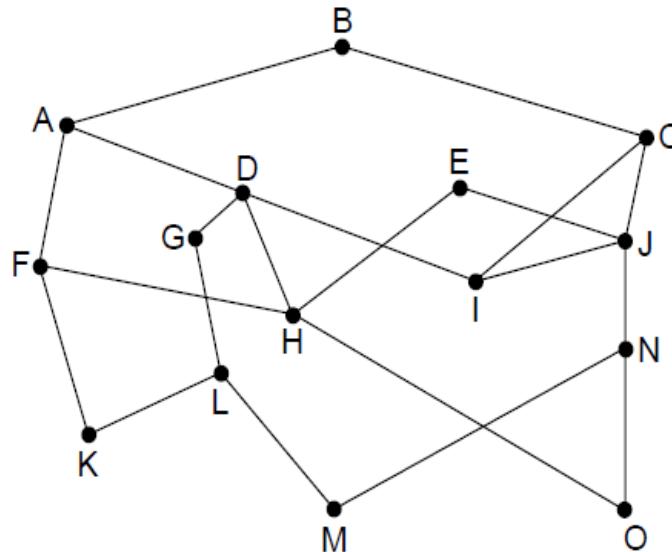
- Consider the network as a graph of nodes and links:
 - Decide what to optimize (e.g., fairness vs efficiency)
 - Update routes for changes in topology (e.g., failures)
 - Routing is the process of discovering network paths
- The routing algorithm is responsible for deciding on which output line an incoming packet should be transmitted
- Non-Adaptive Algorithms
 - Static decision making process (e.g., static routing)
- Adaptive Algorithms
 - Dynamic decision making process (e.g., dynamic routing)
 - Changes in network topology, traffic, etc.

Optimality Principle

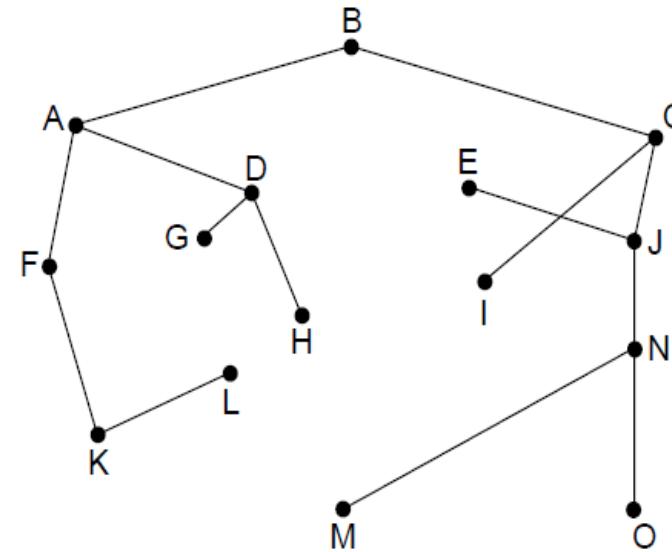
- “If router B is on the optimal path from router A to router C, then the optimal path for B to C also falls along the same route”.

Sink Tree

- The set of optimal routes from all sources to a given destination form a tree rooted at the destination - “sink tree”
- The goal of a routing algorithm is to discover and utilise the sink trees for all routers



Network



Sink tree of best paths to router B

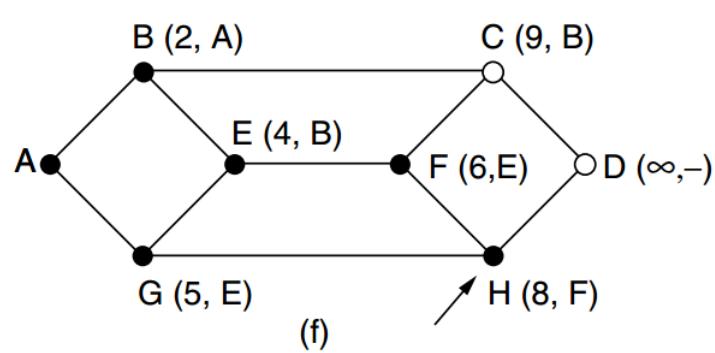
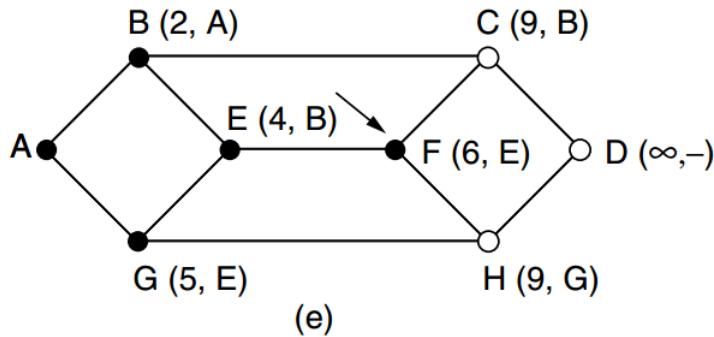
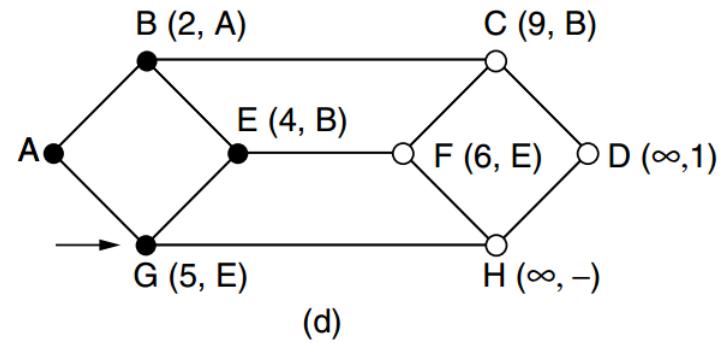
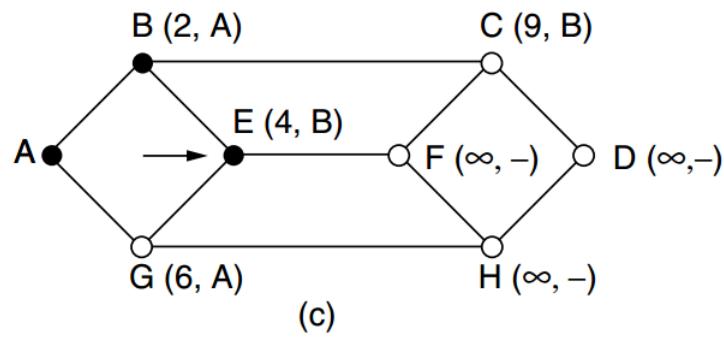
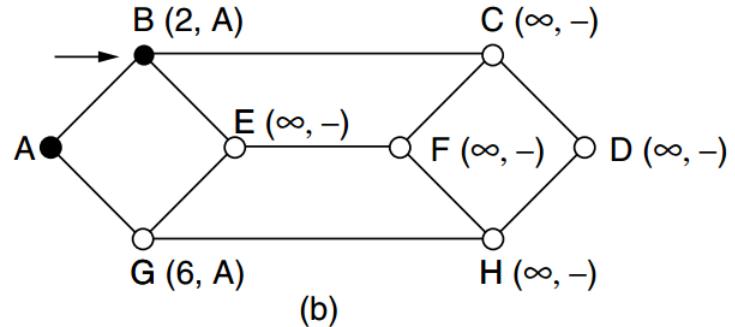
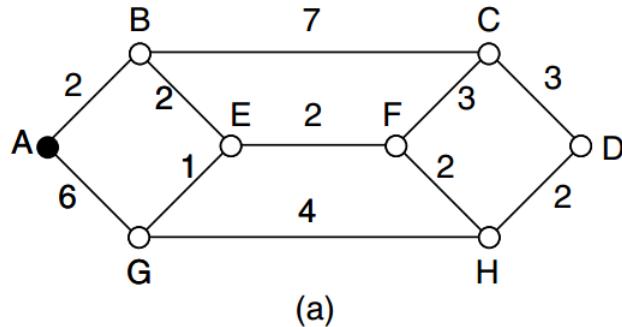
Shortest Path Routing

- A non-adaptive algorithm
- Shortest path can be determined by building a graph with each node representing a router, and each arc representing a communication link
- To choose a path between 2 routers, the algorithm finds the shortest path between them on the graph

Dijkstra's algorithm for computing a sink tree

- Dijkstra's algorithm computes a sink tree on the graph:
 - Each link is assigned a non-negative weight/distance
 - Shortest path is the one with lowest total weight
 - Using weights of 1 gives paths with fewest hops
- Algorithm:
 - Start with sink, set distance at other nodes to infinity
 - Relax distance to other nodes
 - Pick the lowest distance node, add it to sink tree
 - Repeat until all nodes are in the sink tree

Shortest Path Algorithm (2)



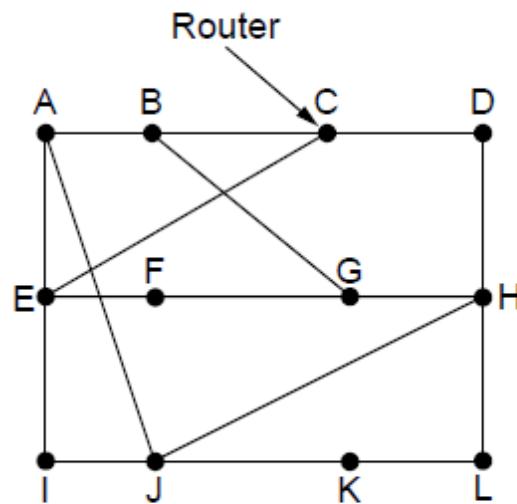
Flooding

- A non-adaptive algorithm
- Every incoming packet is sent out on every outgoing line except the one on which it arrived
- Generates a large number of duplicate packets - inefficient
- Selective flooding (where routers send packets only on links which are in approximately the right direction) is an improved variation

Distance Vector Routing

- A **dynamic** algorithm
- Each router maintains a **table** which includes the **best known distance** to each destination (a metric) and which line to use to get there.
- Tables are exchanged with **neighbouring routers**
- “**Global information shared locally**”
- Algorithm:
 - Each node knows distance of links to its neighbors
 - Each node advertises vector of lowest known distances to all neighbors
 - Each node uses received vectors to update its own
 - Repeat periodically

Distance Vector Routing (2)



Network

New estimated delay from J

↓ Line

To	A	I	H	K	
A	0	24	20	21	8 A
B	12	36	31	28	20 A
C	25	18	19	36	28 I
D	40	27	8	24	20 H
E	14	7	30	22	17 I
F	23	20	19	40	30 I
G	18	31	6	31	18 H
H	17	20	0	19	12 H
I	21	0	14	22	10 I
J	9	11	7	10	0 -
K	24	22	22	0	6 K
L	29	33	9	9	15 K

JA delay is 8 JI delay is 10 JH delay is 12 JK delay is 6

New vector for J

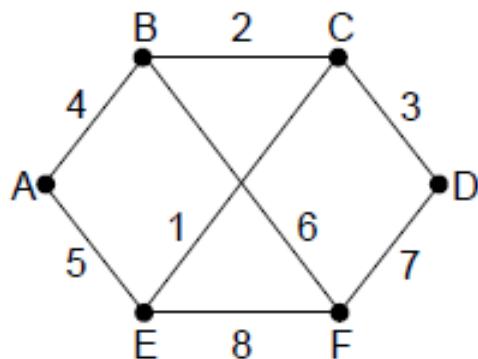
Vectors received at J from Neighbors A, I, H and K

Link State Routing

- A dynamic algorithm
 - An alternative to distance vector
 - DV - primary problem that caused its demise was that the algorithm often took too long to converge after the network topology changed
 - Widely used in the Internet (OSPF, ISIS)
 - More computation but simpler dynamics
- Each router has to do 5 steps:
 1. Discover neighbours and learn network addresses
 2. Measure delay or cost to each neighbour
 3. Construct packet resulting from previous steps
 4. Send this packet to all other routers
 5. Compute the shortest path to every other router
- “**Local information shared globally**” using flooding

Building link state packets

- LSP (Link State Packet) for a node lists neighbors and weights of links to reach them



Network

Link	State	Packets
A	C	E
Seq.	Seq.	Seq.
Age	Age	Age
B 4	B 2	A 5
E 5	C 2	B 6
	D 3	C 1
	F 6	D 7
	E 1	F 8

LSP for each node

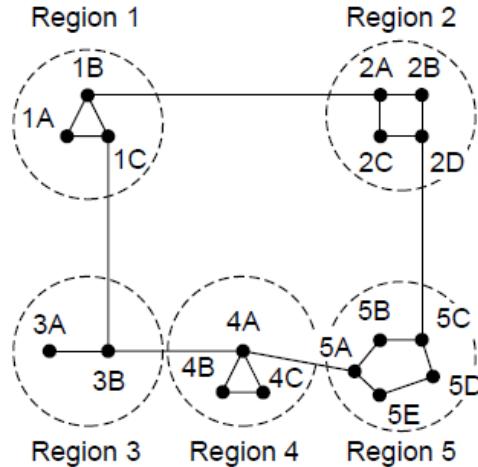
- The hard part is determining when to build LSP.
- Periodically, that is, at regular intervals
- Build them when some significant event occurs, such as a line or neighbour going down or coming back up again or changing its properties appreciably

Hierarchical Routing

- As networks grow in size, routing tables expand but this impacts CPU and memory requirements
- Dividing all routers into regions allows efficiencies
 - Each router knows everything about other routers in its region but nothing about routers in other regions
 - Routers which connect to two regions act as exchange points for routing decisions

Hierarchical routing, cont

- Hierarchical routing reduces the work of route computation but may result in slightly longer paths than flat routing



Full table for 1A

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

Hierarchical table for 1A

Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

Broadcast Routing

- Broadcast routing allows hosts to send messages to many or all other hosts
 - Single distinct packet (inefficient, source needs all destination addresses)
 - Flooding
 - Multi-destination routing (efficient but source needs to know all the destinations)
 - a router receives a single packet which encapsulates the list of destinations, and then constructs a specific packet for each one (acts as a relay)
- Reverse path forwarding
- When a broadcast packet arrives at a router, the router checks to see if the packet arrived on the line normally used for sending packets to the source of the broadcast. If so there is a high probability that the route used to transmit the received packet is the best route. The router then forwards the packet onto all other lines.
- If the broadcast packet arrived on a link other than the preferred one for reaching the source, the packet is discarded as a likely duplicate

Multicast Routing

- A routing algorithm used to send a message to a well-defined group within the whole network
- Each router computes a spanning tree covering all other routers – the first router to receive the packet prunes the spanning tree to eliminate all lines which do not lead to members of the group

Internetwork Routing

- Need to route:
- Within a network using an interior gateway protocol (eg OSPF)
- Between networks using an exterior gateway protocol (eg BGP)

Routing and Routing Control

- Congestion Control Algorithms
 - Handling congestion is the responsibility of the Network and Transport layers working together
 - We look at the Network portion here
- Quality of Service (QoS)

Summary

- Type of services provided by Network Layer
- Internetworking
 - Tunneling
 - Fragmentation
 - Path MTU discovery
- Internet Protocol (IP)
 - Explain principles of Internet design
 - Analyse structure of IP addresses, subnetting, IP add. aggregation, NAT
 - Explain roles of different Internet Control Protocols
- Routing
 - Dijkstra's algorithm
 - Distance Vector routing
 - Link State routing

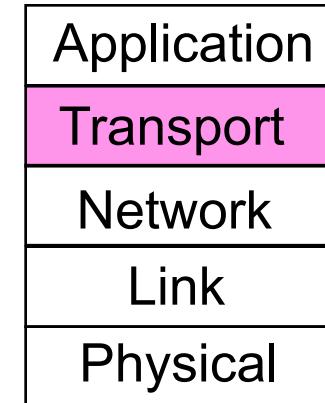
Transport Layer

COMP90007
Internet Technologies

Chien Aun Chan

Outline

- Transport Layer Services
- Transport Layer Primitives
- Elements of Transport Protocols
- Sockets



The Transport Service

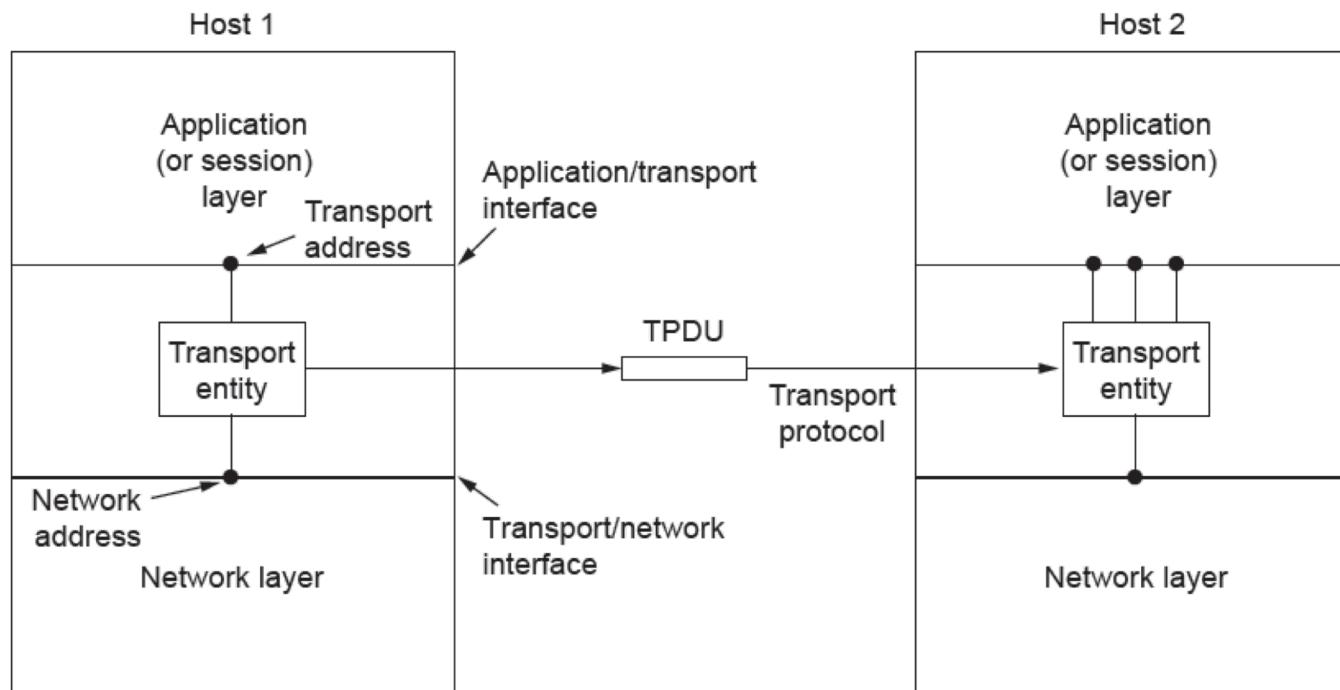
- Primary function
 - provide efficient, reliable & cost-effective data transmission service to the processes in the application layer, independent of physical or data networks
- To Achieve this
 - It uses service provided by the network layer
- *The Transport entity*: the software or hardware within the transport layer that does the work.

Transport Layer Services

- Transport Layer **Services** provide interfaces between the Application Layer and the Network Layer
- Transport **Entities** (the hardware or software which actually does the work) can exist in multiple locations:
 - Common
 - OS kernel
 - System library (library package bound into network applications)
 - Not so common
 - User process
 - NIC

Services

- Transport layer adds *reliability* to the network layer
 - Offers **connectionless** (e.g., UDP) and **connection-oriented** (e.g., TCP) services to applications
- Relationship between network, transport and application layers:



Transport Layer and Network Layer Services Compared

- If **transport** and **network** layers are so similar, why are there two layers?
- Transport layer code runs entirely on **hosts**
- Network layer code runs almost entirely on **routers**
- Transport layer can fix reliability problems caused by the Network layer (e.g., delayed, lost or duplicated packets)
- Users have no real control over the network layer – Transport layer: improve QoS

Role of the Transport Layer

- The Transport Layer occupies a key position in the layer hierarchy because it clearly delineates
 - providers of (reliable) data transmission services
 - at the network, data link, and physical layers
 - users of reliable data transmission services
 - at the application layer
- In particular, connection-oriented transport services provide a reliable service on top of an unreliable network

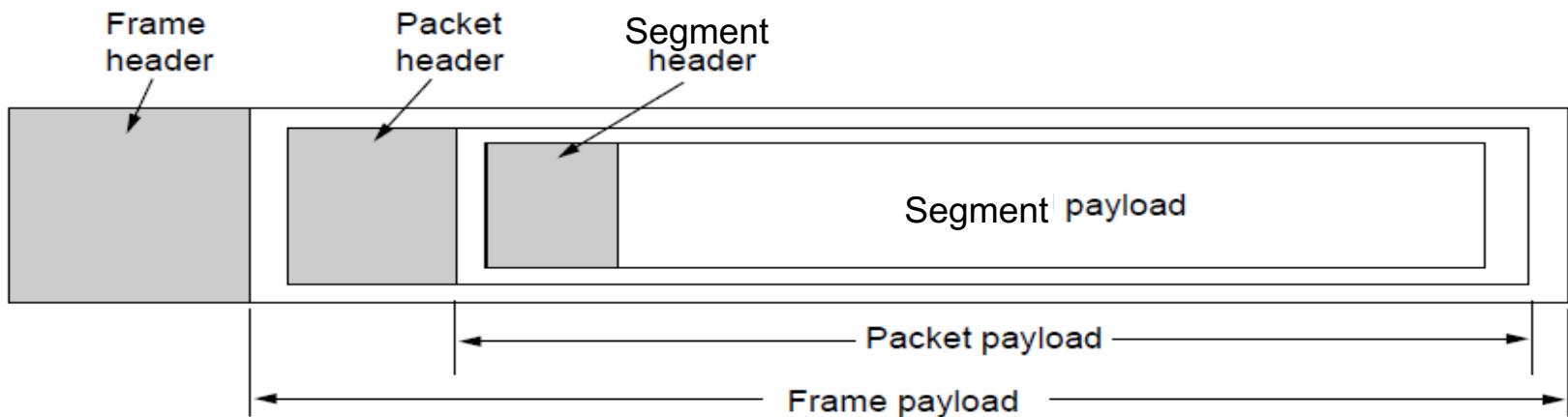
Features of Transport Layer

- Mechanism for improved QoS for users
 - Reliability at application level through interface with network layer
- Abstraction and primitives provide a **simpler API** for application developers independent of network layer

Primitive	Meaning
LISTEN	Block waiting for an incoming connection
CONNECT	Establish a connection with a waiting peer
RECEIVE	Block waiting for an incoming message
SEND	Send a message to the peer
DISCONNECT	Terminate a connection

Transport Layer Encapsulation

- Abstract representation of messages sent to and from transport entities
 - Transport Protocol Data Unit (TPDU)
 - segment
- Encapsulation of **TPDUs** (transport layer units) in **packets** (network layer units) in frames (data layer units)



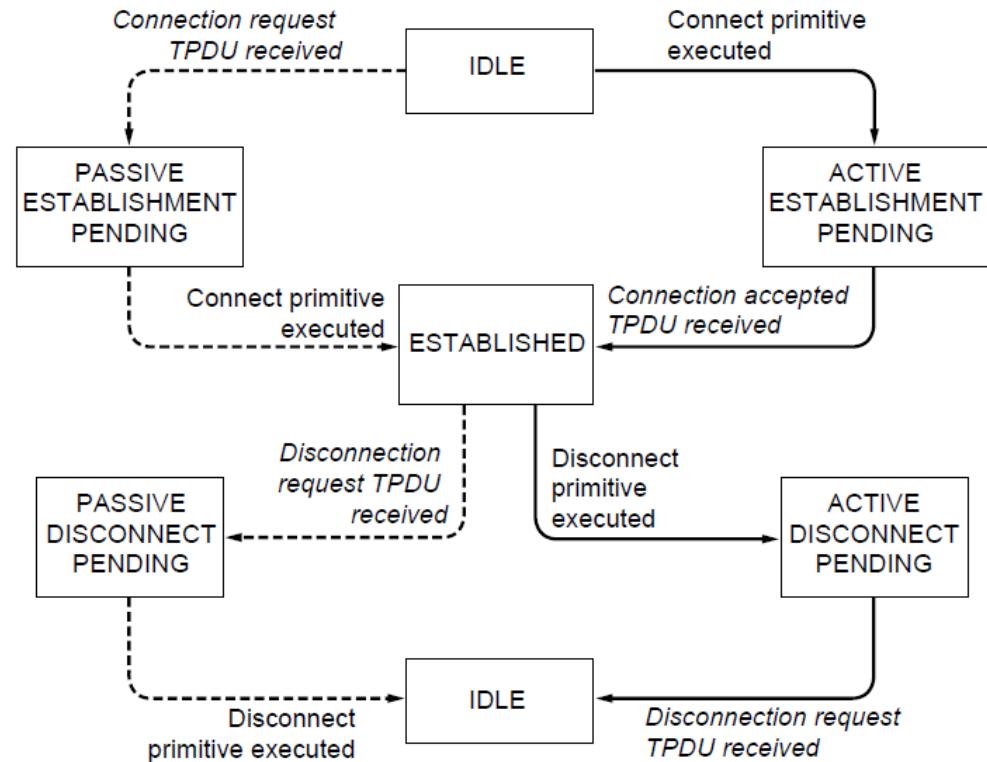
Transport Service Primitives

- Primitives that applications might call to transport data for a simple connection-oriented service:
 - Server executes **LISTEN**
 - Client executes **CONNECT**
 - Sends CONNECTION REQUEST TPDU to Server
 - Receives CONNECTION ACCEPTED TPDU to Client
 - Data exchanged using **SEND** and **RECEIVE**
 - Either party executes **DISCONNECT**

Primitive	Segment sent	Meaning
LISTEN	(none)	Block until some process tries to connect
CONNECT	CONNECTION REQ.	Actively attempt to establish a connection
SEND	DATA	Send information
RECEIVE	(none)	Block until a DATA packet arrives
DISCONNECT	DISCONNECTION REQ.	This side wants to release the connection

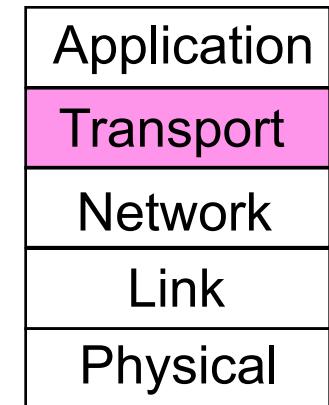
Simple Connections Illustrated

- Solid lines (right) show client state sequence
- Dashed lines (left) show server state sequence
- Transitions in italics are due to segment arrivals



Elements of Transport Protocols

- ❑ Connection establishment
- ❑ Connection release
- ❑ Addressing



Connection Establishment in the Real World

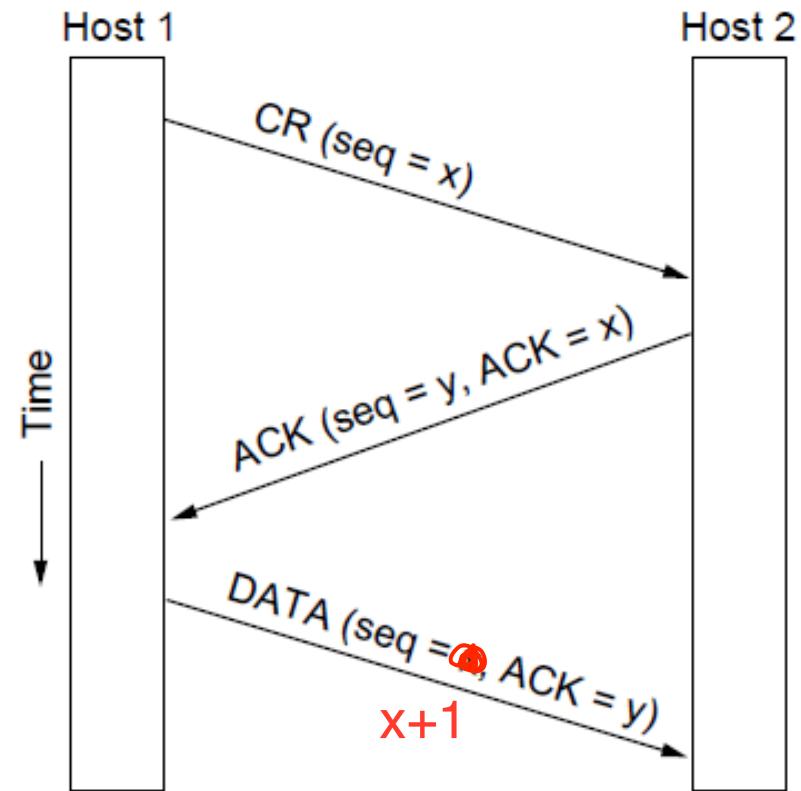
- When networks can **lose, store and duplicate** packets, connection establishment can be complicated
 - congested networks may delay acknowledgements
 - incurring repeated multiple transmissions
 - any of which may not arrive at all or out of sequence – **delayed duplicates**
 - Applications degenerate with such congestion (eg. Imagine duplication of bank withdrawals)

Reliable Connection Establishment

- Key challenge is to ensure reliability even though packets may be lost, corrupted, delayed, and duplicated
 - Don't treat an old or duplicate packet as new
 - (Use ARQ and checksums for loss/corruption)
- Approach:
 - Don't reuse **Maximum Segment Lifetime** sequence numbers within twice the MSL (2min)
 - Three-way handshake for establishing connection
 - Use a sequence number space large enough that it will not wrap, even when sending at full rate

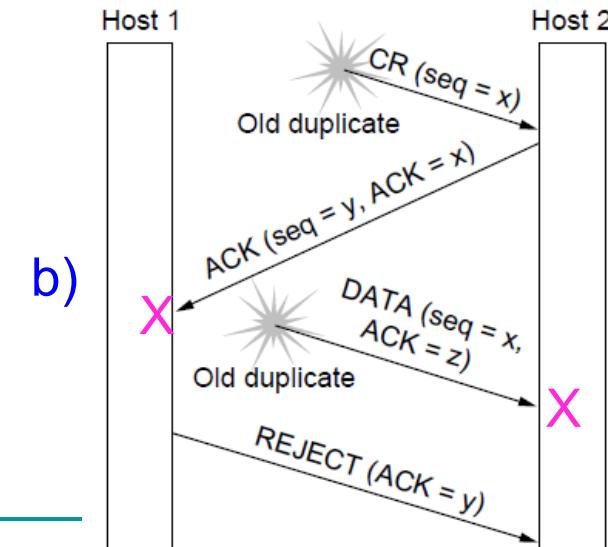
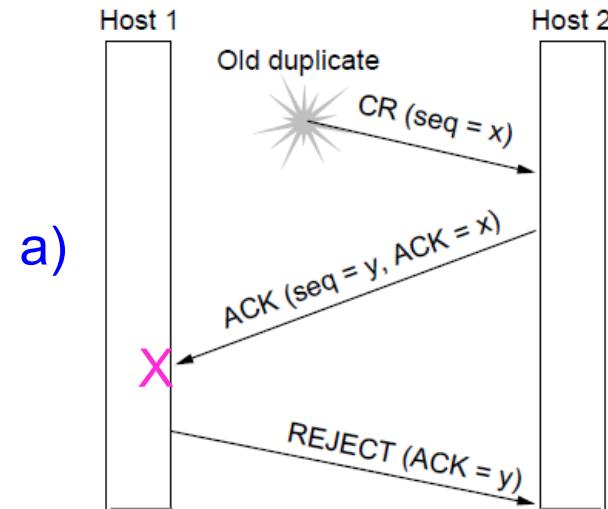
Three Way Handshake

- Three-way handshake used for initial packet
 - Since no state from previous connection
 - Both hosts contribute fresh seq. numbers
 - CR = Connect Request



Three Way Handshake (Working)

- Three-way handshake protects against odd cases:
 - Duplicate CR. Spurious ACK does not connect
 - Duplicate CR and DATA. Same plus DATA will be rejected (wrong ACK).
- Within a connection, a timestamp is used to extend the 32-bit sequence number so that it will not wrap within the maximum packet lifetime, even for gigabit-per-second connections

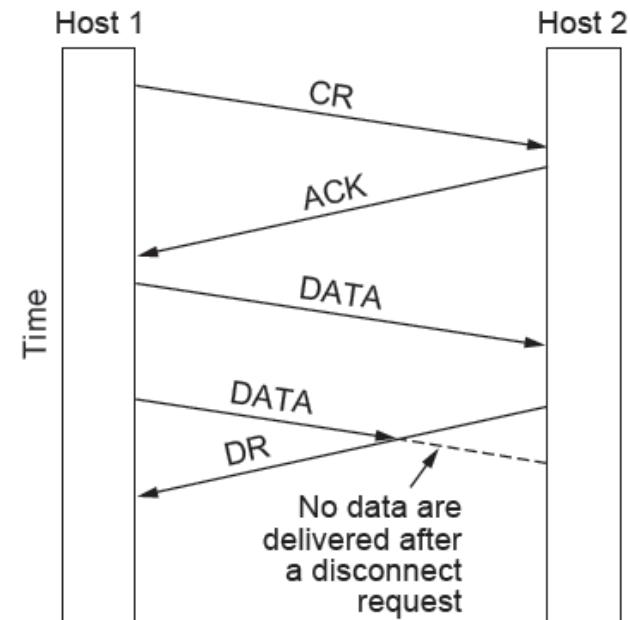


Connection Release

- **Asymmetric Disconnection**
 - Either party can issue a DISCONNECT, which results in DISCONNECT TPDU and transmission ends in both directions
- **Symmetric Disconnection**
 - Both parties issue DISCONNECT, closing only *one direction at a time* -allows flexibility to remain in receive mode

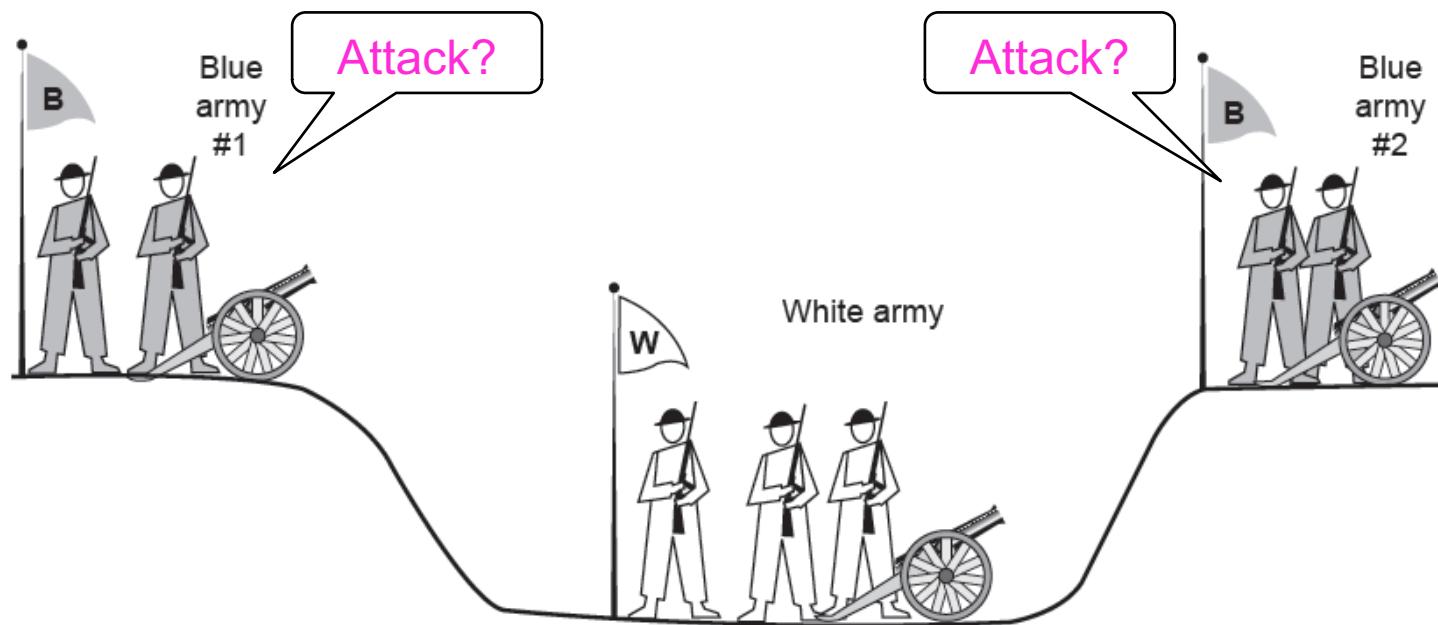
Connection Release (Cont.)

- Asymmetric vs Symmetric connection release types
- **Asymmetric** release may result in data loss hence symmetric release is more attractive
- **Symmetric** release works well where each process has a set amount of data to transmit and knows when it has been sent
- What happens in other cases?



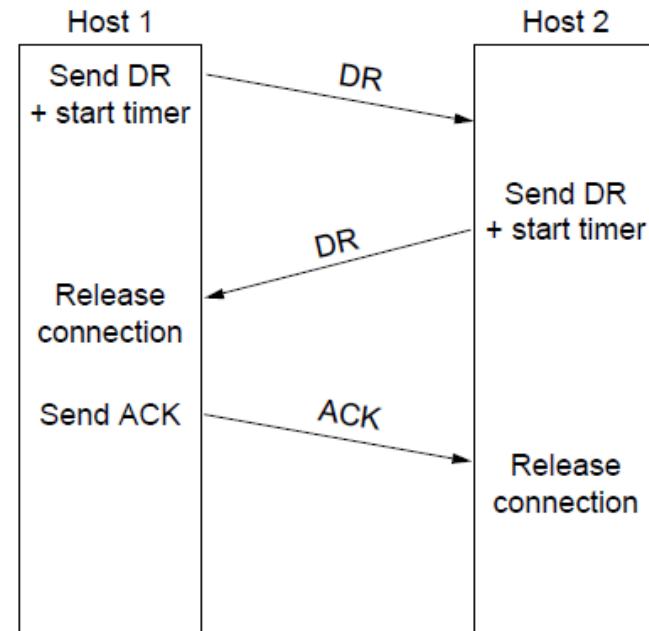
Resolving the Connection Release Problem

- How do we decide the importance of the last message? Is it essential or not?
- No protocol exists which can resolve this ambiguity- Two-army problem shows pitfall of agreement



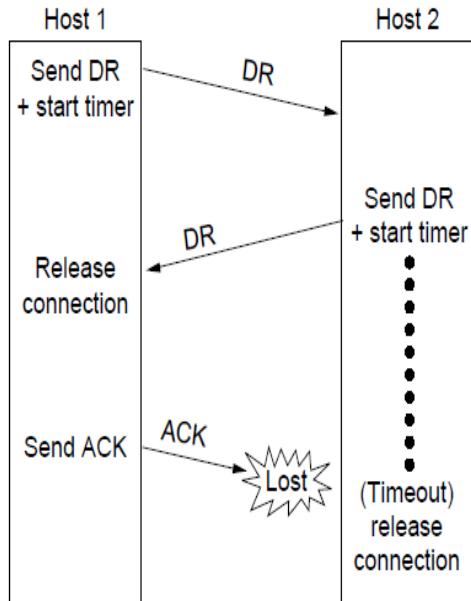
Strategies to Allow Connection Release

- 3 way handshake
- Finite retry
- *Timeouts*
- Normal release sequence,
initiated by transport user on
Host 1
 - DR=Disconnect Request
 - Both DRs are ACKed by the other side

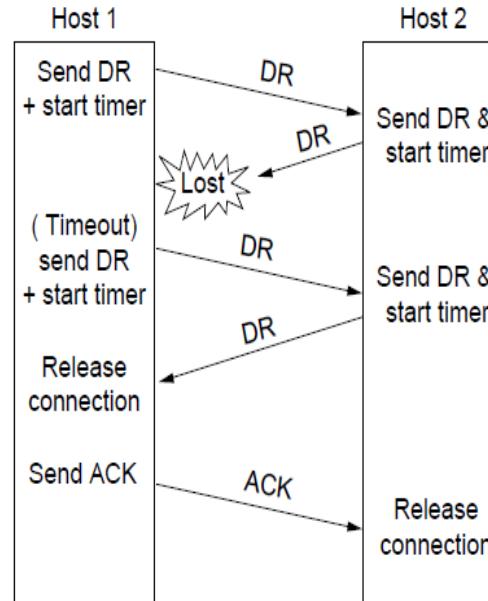


Connection Release (Error Cases)

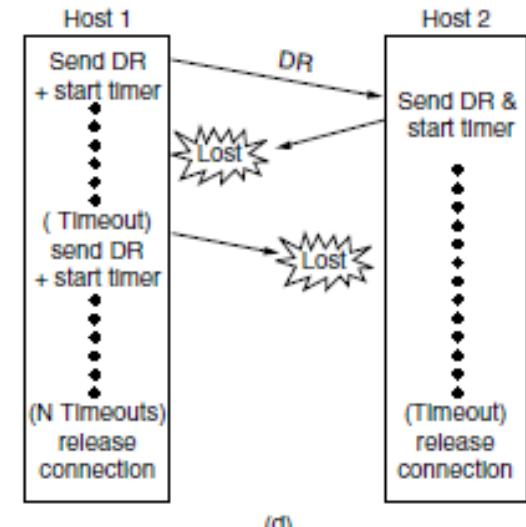
- Error cases are handled with timer and retransmission



Final ACK lost,
Host 2 times
out



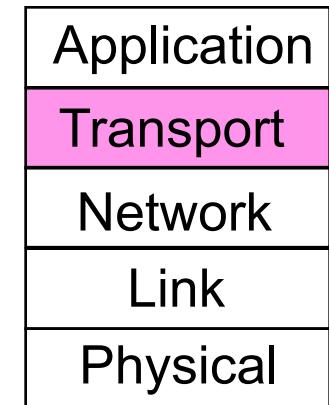
Lost DR
causes
retransmission



Extreme: Many lost
DRs cause both
hosts to timeout

Elements of Transport Protocols

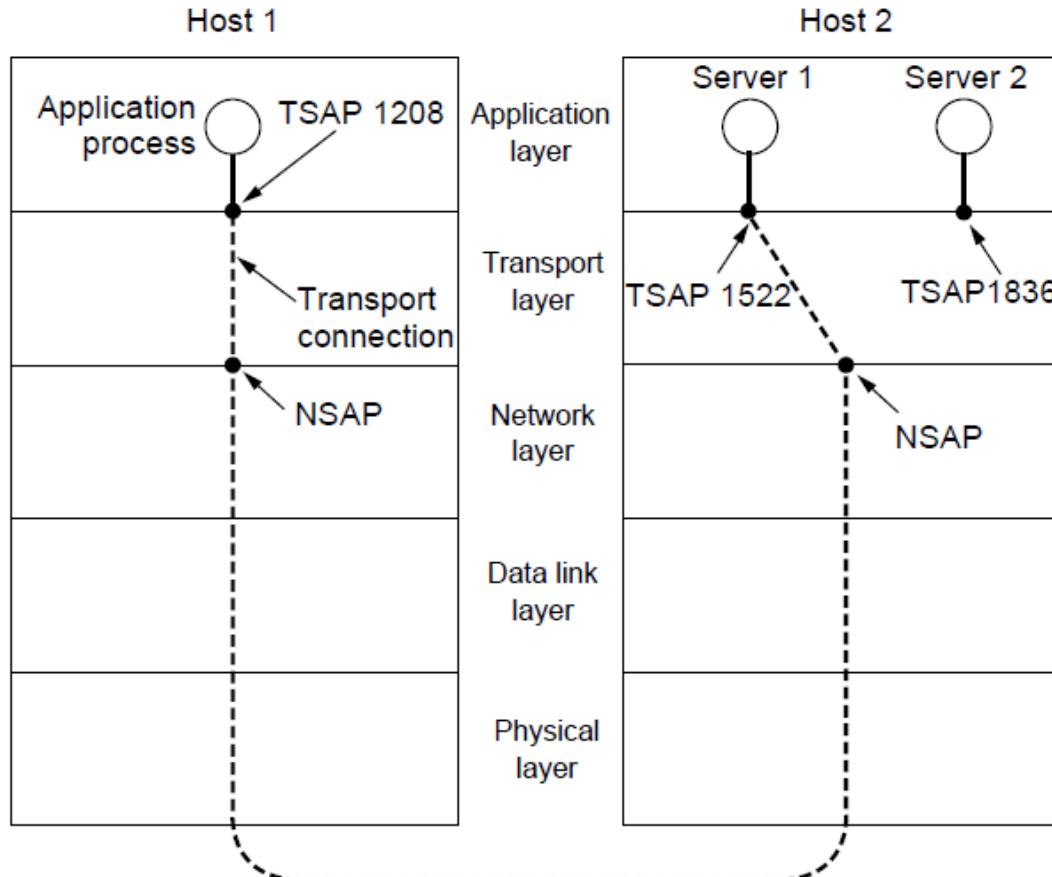
- ❑ Connection establishment
- ❑ Connection release
- ❑ Addressing



Addressing

- Specification of remote process to connect to is required at application and transport layers
- Addressing in transport layer is typically done using Transport Service Access Points (TSAPs)
 - on the internet, a TSAP is commonly referred to as a port (eg port 80)
- Addressing in the network layer is typically done using Network Service Access Points (NSAPs)
 - on the internet, the concept of an NSAP is commonly interpreted as simply an IP address

TSAPs, NSAPs and Transport Layer Connections Illustrated



Types of TSAP Allocation

1. Static

- ❑ Well known services have standard allocated TSAPs/ports, which are embedded in OS
- ❑ cf. Unix /etc/services, www.iana.org

2. Directory Assistance – Port-mapper

- ❑ A new service must register itself with the portmapper, giving both its service name and TSAP

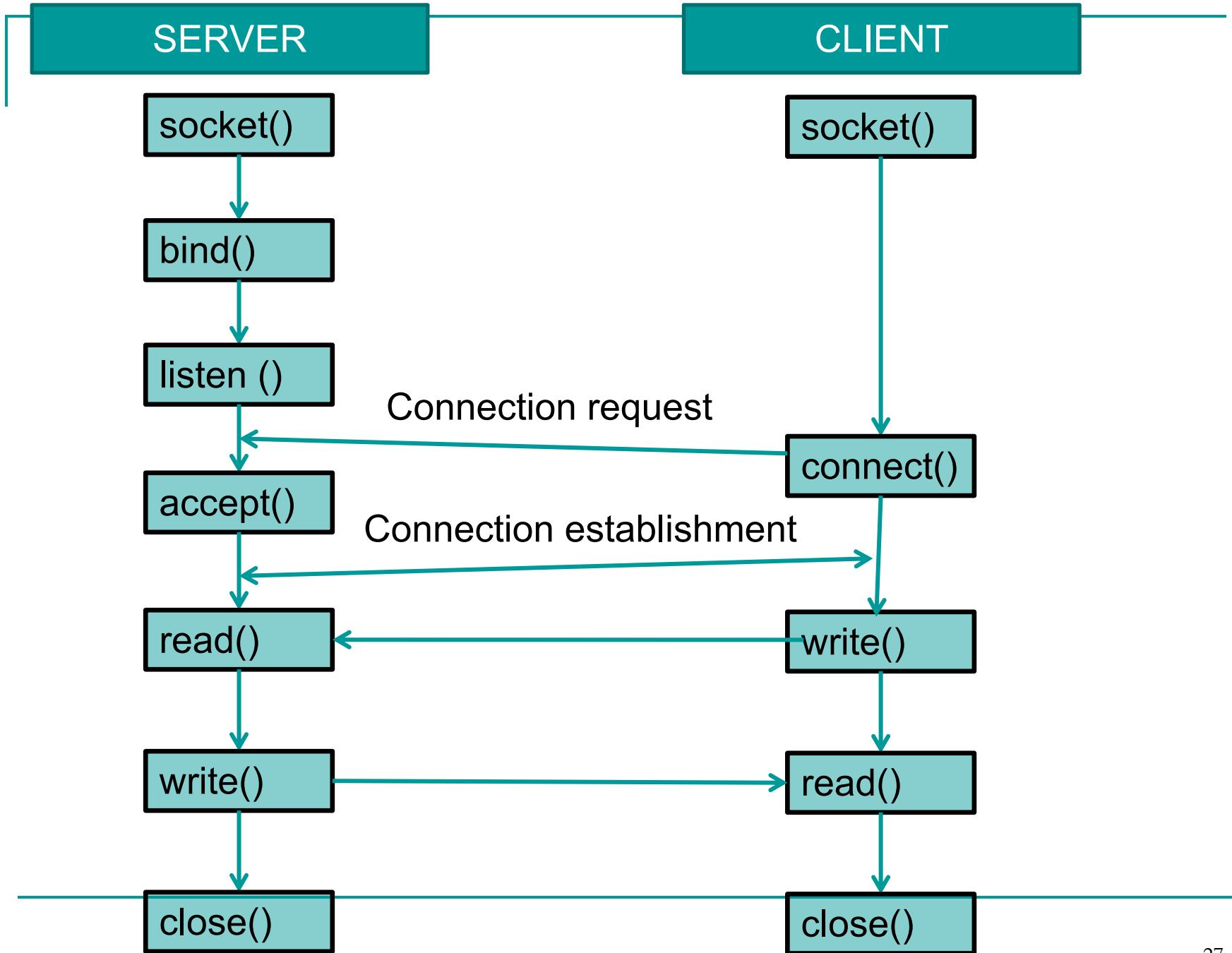
3. Mediated

- ❑ A process server intercepts inbound connections and spawns requested server and attaches inbound connection
- ❑ cf. Unix /etc/(x)inetd

Sockets

- Sockets widely used for interconnections
 - “Berkeley” sockets are predominant in internet applications
 - Notion of “sockets” as transport endpoints
 - Like simple set plus SOCKET, BIND, and ACCEPT

Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Associate a local address with a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Passively establish an incoming connection
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection



Socket Example – Internet File Server

Client code

```
if (argc != 3) fatal("Usage: client server-name file-name");
h = gethostbyname(argv[1]);                                Get server's IP
if (!h) fatal("gethostbyname failed");
s = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);           Make a socket
if (s <0) fatal("socket");
memset(&channel, 0, sizeof(channel));
channel.sin_family= AF_INET;
memcpy(&channel.sin_addr.s_addr, h->h_addr, h->h_length);
channel.sin_port= htons(SERVER_PORT);

c = connect(s, (struct sockaddr *) &channel, sizeof(channel));
if (c < 0) fatal("connect failed");                      Try to connect
```

socket() - parameters of the call specify the addressing format to be used, the type of service desired, and protocol

Socket Example – Internet File Server, Cont.

Client code (cont.)

...

```
write(s, argv[2], strlen(argv[2])+1);
```

Write data (equivalent to send)

```
while (1) {
    bytes = read(s, buf, BUF_SIZE);
    if (bytes <= 0) exit(0);
    write(1, buf, bytes);
}
```

Loop reading (equivalent to receive) until no more data; exit implicitly calls close

Socket Example – Internet File Server (3)

Server code

```
...  
  
memset(&channel, 0, sizeof(channel));  
channel.sin_family = AF_INET;  
channel.sin_addr.s_addr = htonl(INADDR_ANY);  
channel.sin_port = htons(SERVER_PORT);  
  
s = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);           Make a socket  
if (s < 0) fatal("socket failed");  
setsockopt(s, SOL_SOCKET, SO_REUSEADDR, (char *) &on, sizeof(on));  
  
b = bind(s, (struct sockaddr *) &channel, sizeof(channel));      Assign address  
if (b < 0) fatal("bind failed");  
  
l = listen(s, QUEUE_SIZE);                                Prepare for  
if (l < 0) fatal("listen failed");                         incoming  
connections  
  
...
```

socket() - parameters of the call specify the addressing format to be used, the type of service desired, and protocol

Socket Example – Internet File Server Cont.

Server code

```
...  
while (1) {  
    sa = accept(s, 0, 0);  
    if (sa < 0) fatal("accept failed");  
  
    read(sa, buf, BUF_SIZE);  
    /* Get and return the file. */  
    fd = open(buf, O_RDONLY);  
    if (fd < 0) fatal("open failed");  
  
    while (1) {  
        bytes = read(fd, buf, BUF_SIZE);  
        if (bytes <= 0) break;  
        write(sa, buf, bytes);  
    }  
    close(fd);  
    close(sa);  
}
```

Block waiting for the next connection

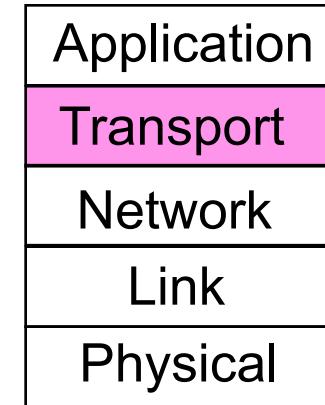
Read (receive) request and treat as file name

Write (send) all file data

Done, so close this connection

Outline

- Internet Transport Protocols
- UDP (connectionless)
- TCP (connection-oriented)

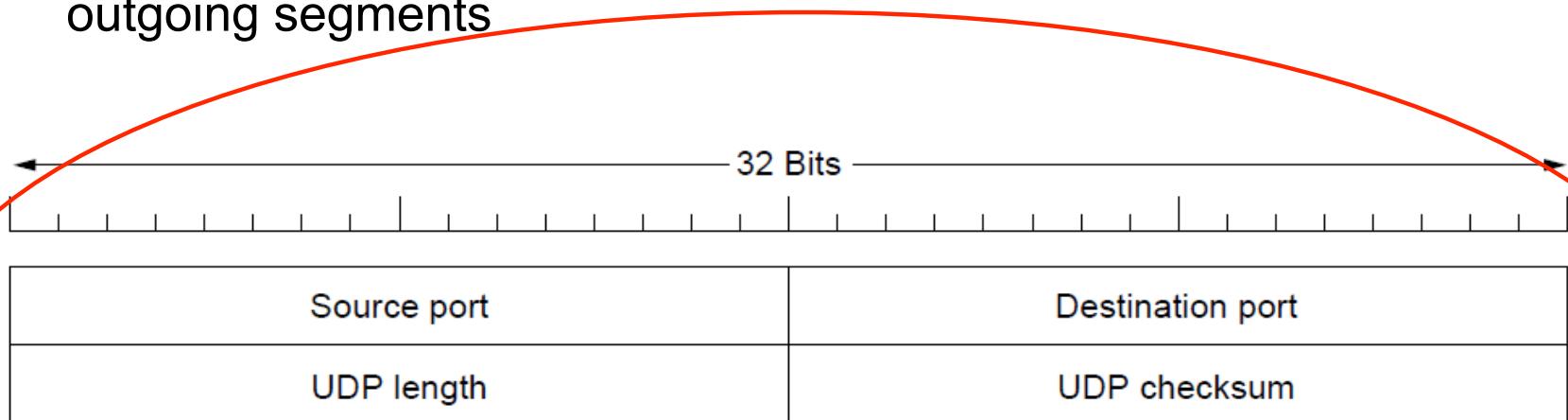


User Datagram Protocol (UDP)

- Defined in RFC 768
- Provides a protocol whereby applications can transmit encapsulated IP datagrams without a connection establishment
- UDP transmits in segments consisting of an 8-byte header followed by the payload
- UDP headers contain source and destination ports, payload is handed to the process which is attached to the particular port at destination (using BIND primitive or similar)

User Datagram Protocol (UDP)

- Main **advantage** of using UDP over raw IP is the ability to specify ports for source and destination pairs
- Both source and destination ports are required - destination allows initial routing for incoming segments, source allows reply routing for outgoing segments



Structure of UDP header: It has ports (TSAPs), length and checksum

Strengths and Weaknesses of UDP

- **Strengths:** provides an IP interface with multiplexing/demultiplexing capabilities and consequently, transmission efficiencies
- **Weaknesses:** UDP does not include support for flow control, error control or retransmission of bad segments
- **Conclusion:** where applications require a precise level of control over packet flow/error/timing, UDP is a good choice

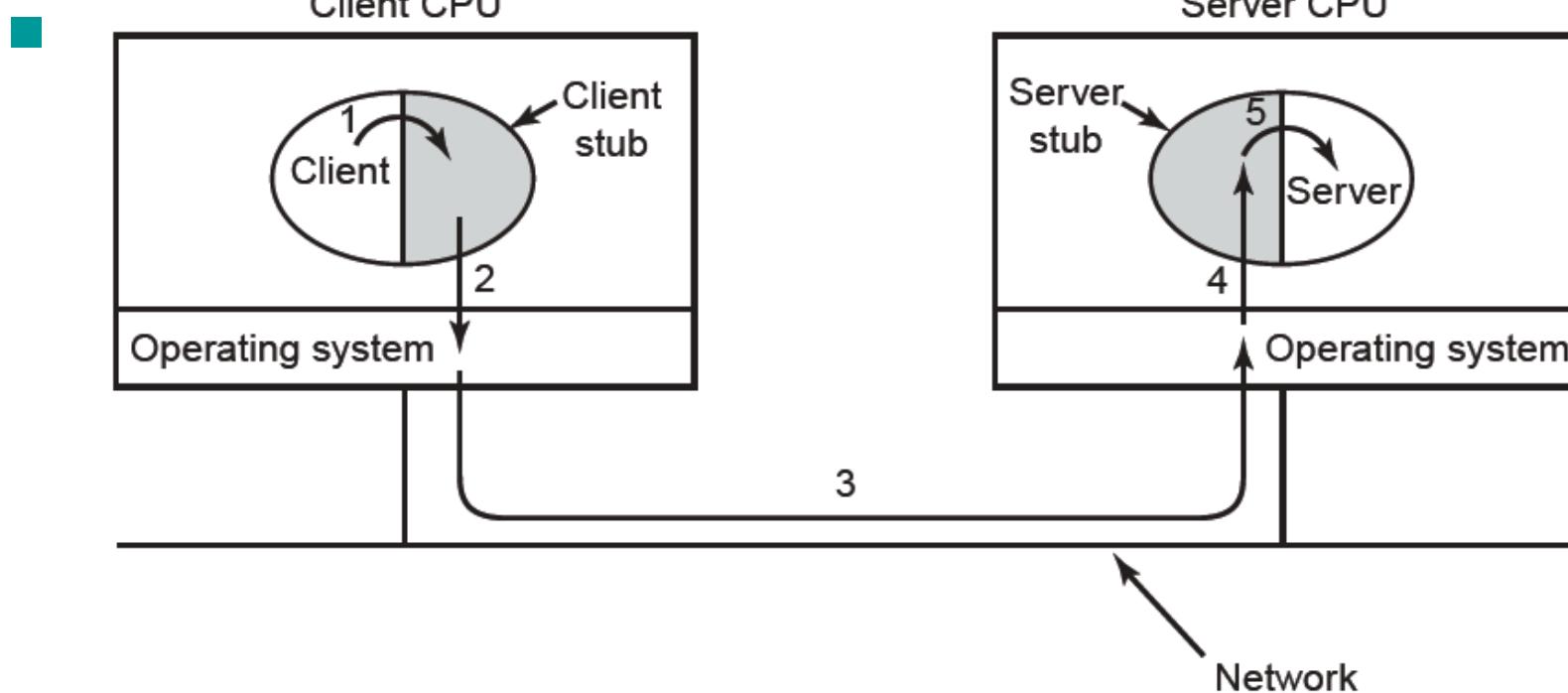
Using UDP: Remote Procedure Call (RPC)

- Sending a message and getting a reply back is analogous to making **a function call** in programming languages
- Birrell and Nelson (1984) modified this approach to allow local programs to call procedures on remote hosts using UDP as the transport protocol
 - **Remote Procedure Call (RPC)**

Using UDP: Remote Procedure Call (RPC)

- To call a remote procedure, the client is bound to a small library (the **client stub**) that represents the server procedure in the client's address space.
- Similarly the server is bound with a procedure called the **server stub**. These stubs hide the fact that the procedure itself is not local.
- UDP with retransmissions is a low-latency transport

RPC Illustrated



Transmission Control Protocol (TCP)

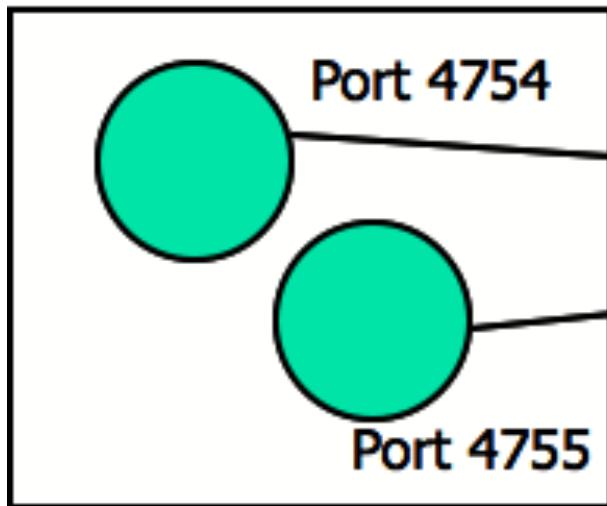
- RFC 793, 1122, 1323
- Provides a protocol by which applications can transmit IP datagrams within a **connection-oriented** framework, thus increasing reliability
- TCP transport entity manages TCP streams and interfaces to the IP layer - can exist in numerous locations (kernel, library, user process)
- TCP entity accepts user data streams, and segments them into pieces < 64KB (often 1460B in order so that the IP and TCP headers can fit into a single Ethernet frame), and sends each piece as a separate IP datagram
- Recipient TCP entities reconstruct the original byte streams from the encapsulation

The TCP Service Model

- Sender and receiver both create “sockets”, consisting of the IP address of the host and a port number
- For TCP Service to be activated, connections must be explicitly established between a socket at a sending host (src-host, src-port) and a socket at a receiving host (dest-host, dest-port)
- Special one-way server sockets may be used for multiple connections simultaneously

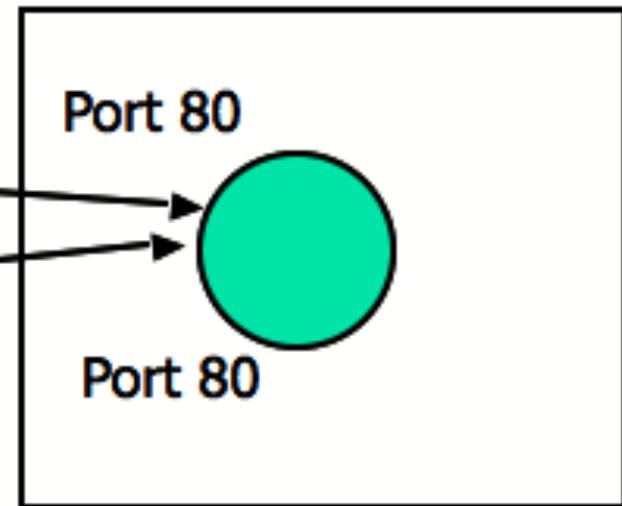
Example

Host 128.42.11.3



Web browser

Host 62.118.44.12



Web server

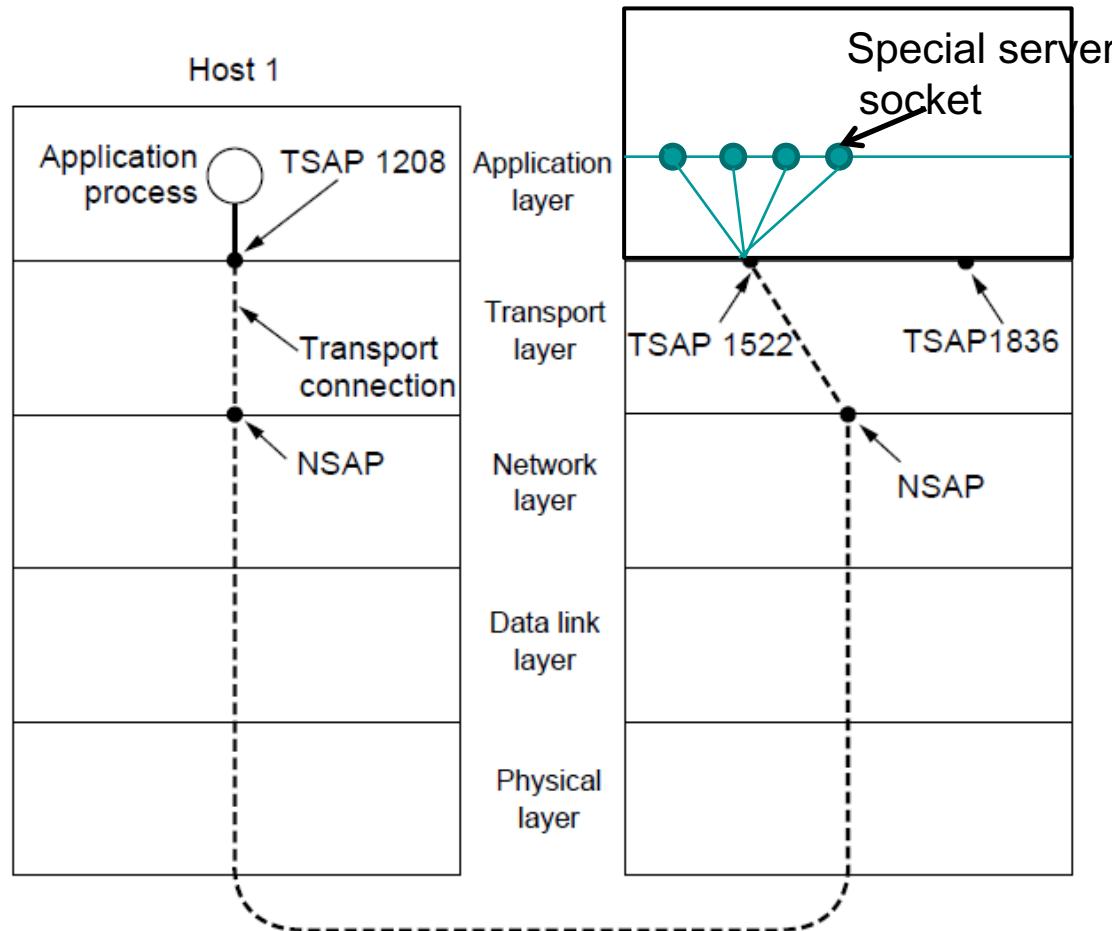
Port Allocations

- Recall TSAPs
- Port numbers can range from 0-65535
- Port numbers are regulated by IANA
(<http://www.iana.org/assignments/port-numbers>)
- Ports are classified into 3 segments:
 - Well Known Ports (0-1023)
 - Registered Ports (1024-49151)
 - Dynamic Ports (49152-65535)

Port	Protocol	Use
20, 21	FTP	File transfer
22	SSH	Remote login, replacement for Telnet
25	SMTP	Email
80	HTTP	World Wide Web
110	POP-3	Remote email access
143	IMAP	Remote email access
443	HTTPS	Secure Web (HTTP over SSL/TLS)
543	RTSP	Media player control
631	IPP	Printer sharing

Socket Library - Multiplexing

- Socket library provides a multiplexing tool on top of TSAPs to allow servers to service multiple clients
- It **simulate** the server using a different port to connect back to the client



Features of TCP Connections

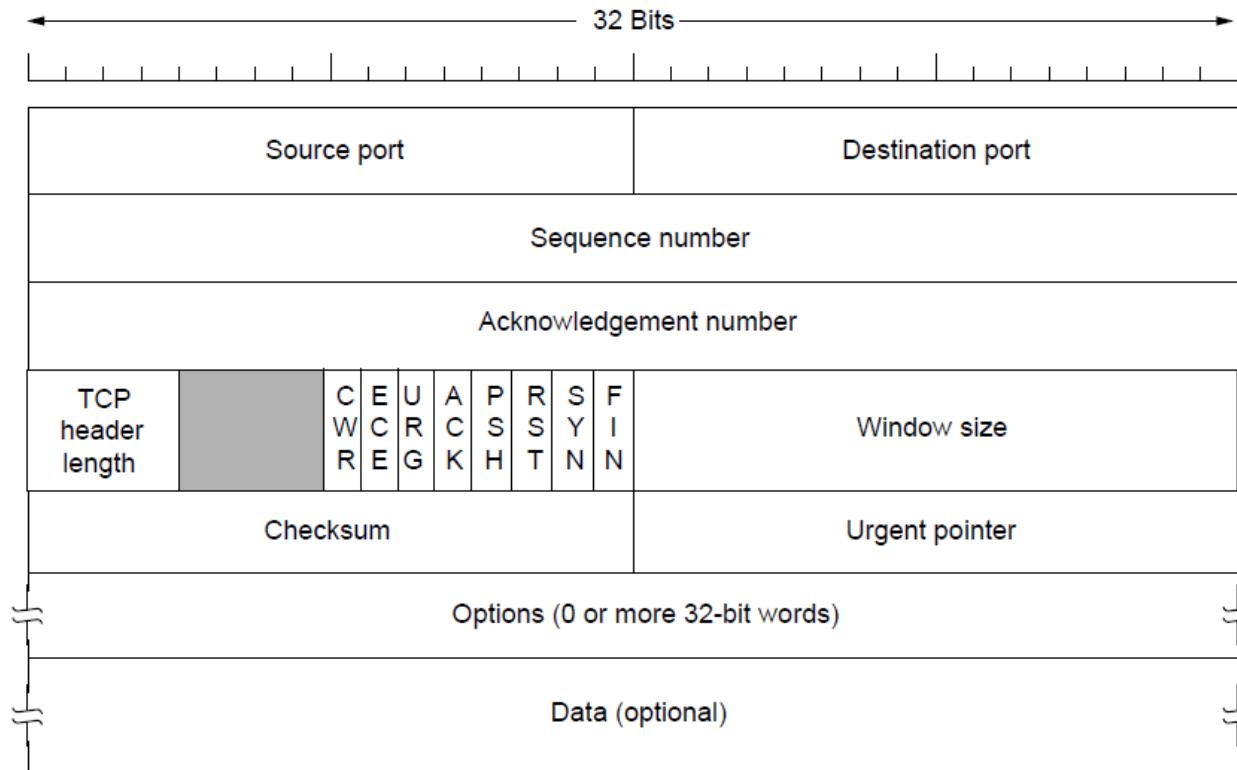
- TCP connections are:
 - **Full duplex** - data in both directions simultaneously
 - **Point to point** - exact pairs of senders and receivers
 - **Byte streams**, not message streams - message boundaries are not preserved
 - **Buffer capable** - TCP entity can choose to buffer prior to sending or not depending on the context
 - PUSH flag - indicates a transmission not to be delayed
 - URGENT flag - indicates that transmission should be sent immediately (priority above in process data)

TCP Protocol

- Data exchanged between TCP entities **in segments** - each segment has a **fixed 20 byte header plus zero or more data bytes**
- TCP entities decide how large segments should be within 2 constraints, namely:
 - 65,515 byte IP payload
 - Maximum Transfer Unit (MTU) - generally 1500 bytes
- **Sliding window protocol** - sender transmits and starts a timer, receiver sends back an acknowledgement which is **the next sequence number expected** - if sender's timer expires before acknowledgement, then the sender transmits the original segment again

The TCP Segment Header

- TCP header includes addressing (ports), sliding window (seq. / ack. number), flow control (window), error control (checksum) and more



The TCP Segment Header

- *Source port* and *Destination port* fields identify the local end points of the connection
- *Sequence number* and *Acknowledgement* number fields perform their usual functions (cumulative acknowledgement)
- *TCP header length* tells how many 32-bit words are contained in the TCP header
- *Window size* field tells how many bytes may be sent starting at the byte acknowledged
- *Checksum* is also provided for extra reliability. It checksums the header, the data
- *Options* field provides a way to add extra facilities not covered by the regular header
- *URG* is set to 1 if the *Urgent pointer* is in use. The *Urgent pointer* is used to indicate a byte offset from the current sequence number at which urgent data are to be found

The TCP Segment Header

- *CWR* and *ECE* are used to signal congestion when *ECN* (Explicit Congestion Notification) is used
- *ECE* is set to signal an ECN-Echo to a TCP sender to tell it to slow down when the TCP receiver gets a congestion indication from the network
- *CWR* is set to signal Congestion Window Reduced from the TCP sender to the TCP receiver so that it knows the sender has slowed down and can stop sending the ECN-Echo
- The *ACK* bit is set to 1 to indicate that the Acknowledgement number is valid. This is the case for nearly all packets; 0=ignore ACK number field
- *PSH* bit indicates PUSHed data. The receiver is hereby kindly requested to deliver the data to the application upon arrival and not buffer it until a full buffer has been received

The TCP Segment Header

- The *RST* bit is used to abruptly reset a connection that has become confused due to a host crash or some other reason. It is also used to reject an invalid segment or refuse an attempt to open a connection
- The *SYN* bit is used to establish connections. The connection request has *SYN* = 1 and *ACK* = 0 to indicate that the piggyback acknowledgement field is not in use. The connection reply does bear an acknowledgement, however, so it has *SYN* = 1 and *ACK* = 1. In essence, the *SYN* bit is used to denote both CONNECTION REQUEST and CONNECTION ACCEPTED, with the *ACK* bit used to distinguish between those two possibilities
- The *FIN* bit is used to release a connection. It specifies that the sender has no more data to transmit. However, after closing a connection, the closing process may continue to receive data indefinitely

TCP Connection Establishment and Release

- Connections established **using three-way handshake**
- Two simultaneous connection attempts results in only one connection (uniquely identified by end points)
- Connections released asynchronously (symmetric release)
- Timers used for lost connection releases (three army problem)

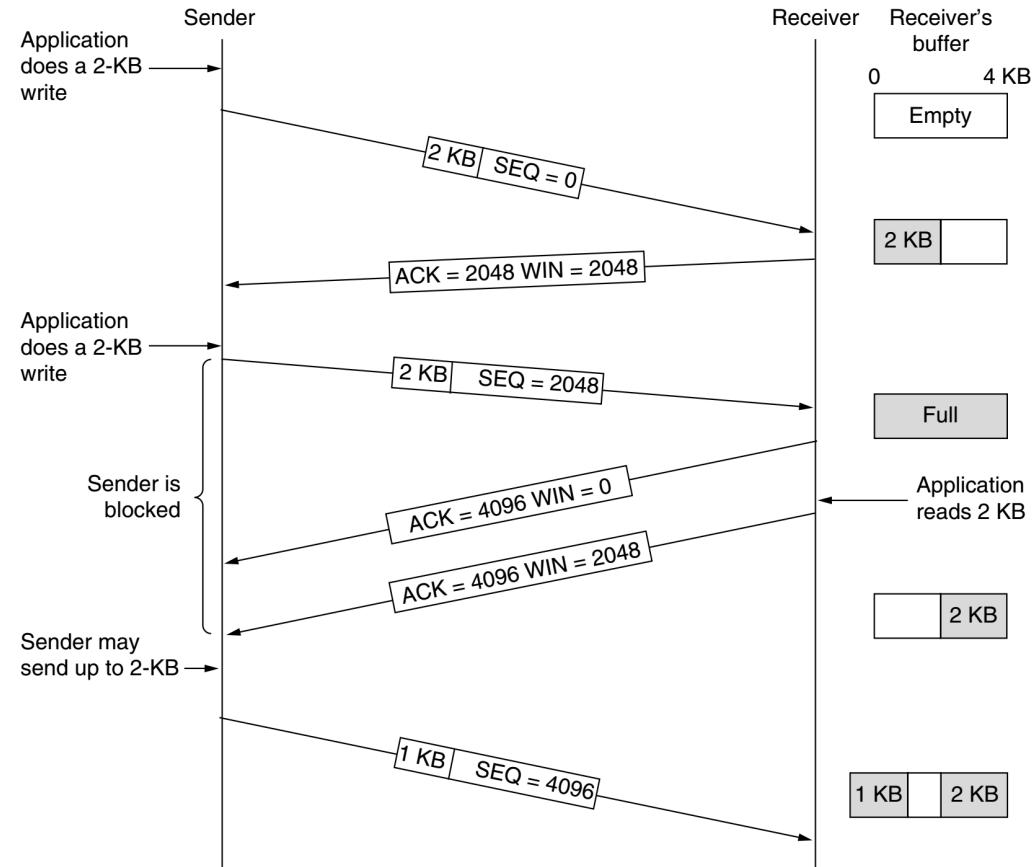
Modelling TCP Connection Management - States

- The TCP connection finite state machine has more states than our simple example from earlier.

State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCVD	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT 1	The application has said it is finished
FIN WAIT 2	The other side has agreed to release
TIME WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAST ACK	Wait for all packets to die off

TCP Transmission Policy

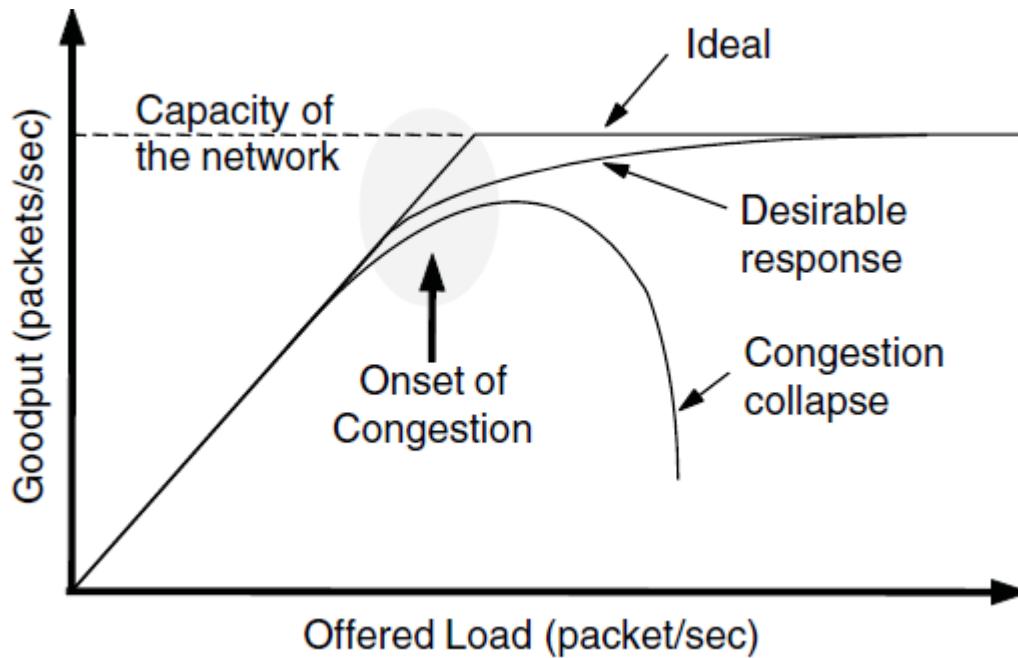
- TCP acknowledges bytes, not packets
- Receiver advertises window based on available buffer space



Congestion Control

Congestion Control

- Congestion results when too much traffic is offered; performance degrades due to loss/retransmissions
 - Goodput (=useful packets) trails offered load



Congestion Control vs Flow Control

- **Flow control** is an issue for point to point traffic, primarily concerned with preventing sender transmitting data faster than receiver can receive it
- **Congestion control** is an issue affecting the ability of the subnet to actually carry the available traffic, in a global context

Load Shedding

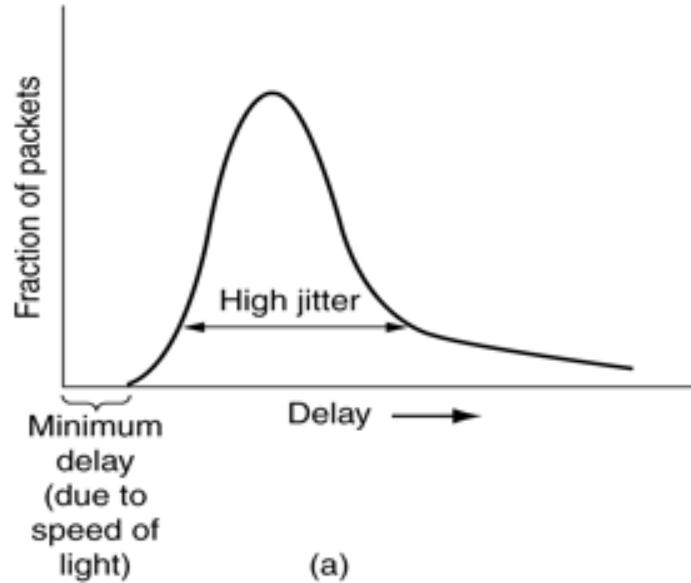
- When congestion control mechanisms fail, load shedding is the only remaining possibility - drop packets
- In order to ameliorate impact, applications can mark certain packets as priority to avoid discard policy (some applications have more stringent requirements than others)

Quality of Service

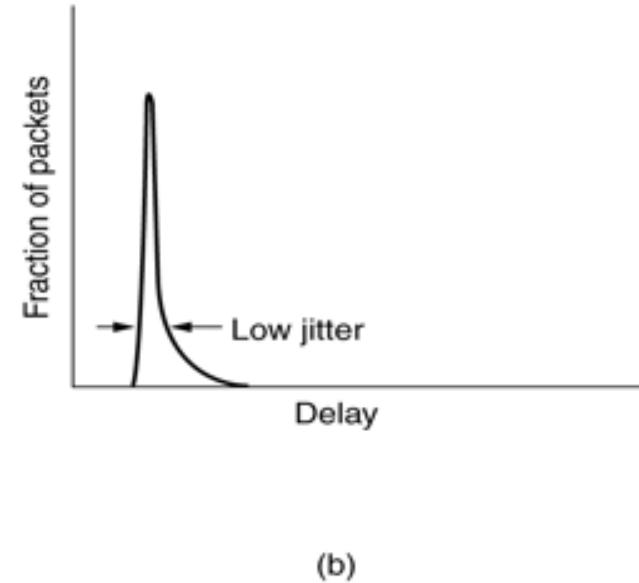
- Expected network performance is an important criterion for a wide range of network applications
- Some engineering techniques are available to guarantee Quality of Service (QoS)
- 4 parameters: reliability, delay, jitter, bandwidth

Jitter Control

- Jitter is the variation in packet arrival times
 - a) high jitter
 - b) low jitter



(a)



(b)

Mechanisms for Jitter Control

- Jitter can be contained by determining the expected transit time of a packet
- Packets can be “shuffled” at each hop in order to minimise jitter - slower packets sent first, faster packets wait in a queue
- For certain applications jitter control is extremely important (eg Voice Over IP), as it directly affects the quality perceived by the application user

QOS Requirements

- Different applications care about different properties
 - We want all applications to get what they need

Application	Bandwidth	Delay	Jitter	Loss
Email	Low	Low	Low	Medium
File sharing	High	Low	Low	Medium
Web access	Medium	Medium	Low	Medium
Remote login	Low	Medium	Medium	Medium
Audio on demand	Low	Low	High	Low
Video on demand	High	Low	High	Low
Telephony	Low	High	High	Low
Videoconferencing	High	High	High	Low

“High” means a demanding requirement, e.g., low delay

Techniques for Good QoS #1

- Over-provisioning
 - more than adequate buffer, router CPU, and bandwidth (expensive and not scalable ... yet)
- Buffering
 - buffer received flows before delivery - increases delay, but smoothes out jitter, no effect in reliability or bandwidth
- Traffic Shaping
 - regulate the average rate of transmission and burstiness of transmission
 - “Buckets”
 - leaky bucket: finite internal queue (in a buffer), regulates outbound flow as well as inbound flow
 - token bucket: finite internal queue (in buffer), variable to maximum outbound flow

Techniques for Good QoS #2

- **Resource Reservation**
 - reserve bandwidth, buffer space, CPU in advance
- **Admission Control**
 - routers can decide based on traffic patterns whether to accept new flows, or reject/reroute them
- **Proportional Routing**
 - different traffic types for same destination split across multiple routes
- **Packet Scheduling**
 - fair queuing, weighted fair queuing

TCP and Congestion Control

- When networks are overloaded, congestion occurs, potentially affecting all layers
- Although lower layers (data and network) attempt to ameliorate congestion, in reality TCP impacts congestion most significantly because TCP offers methods to transparently reduce the data rate, and hence reduce congestion itself

Congestion Control: Design

- Two different problems exist
 - network capacity and receiver capacity
 - these should be dealt with separately, but compatibly
- The sender maintains two windows
 - Window described by the receiver
 - Congestion window
- Each regulates the number of bytes the sender can transmit – the maximum transmission rate is the minimum of the two windows

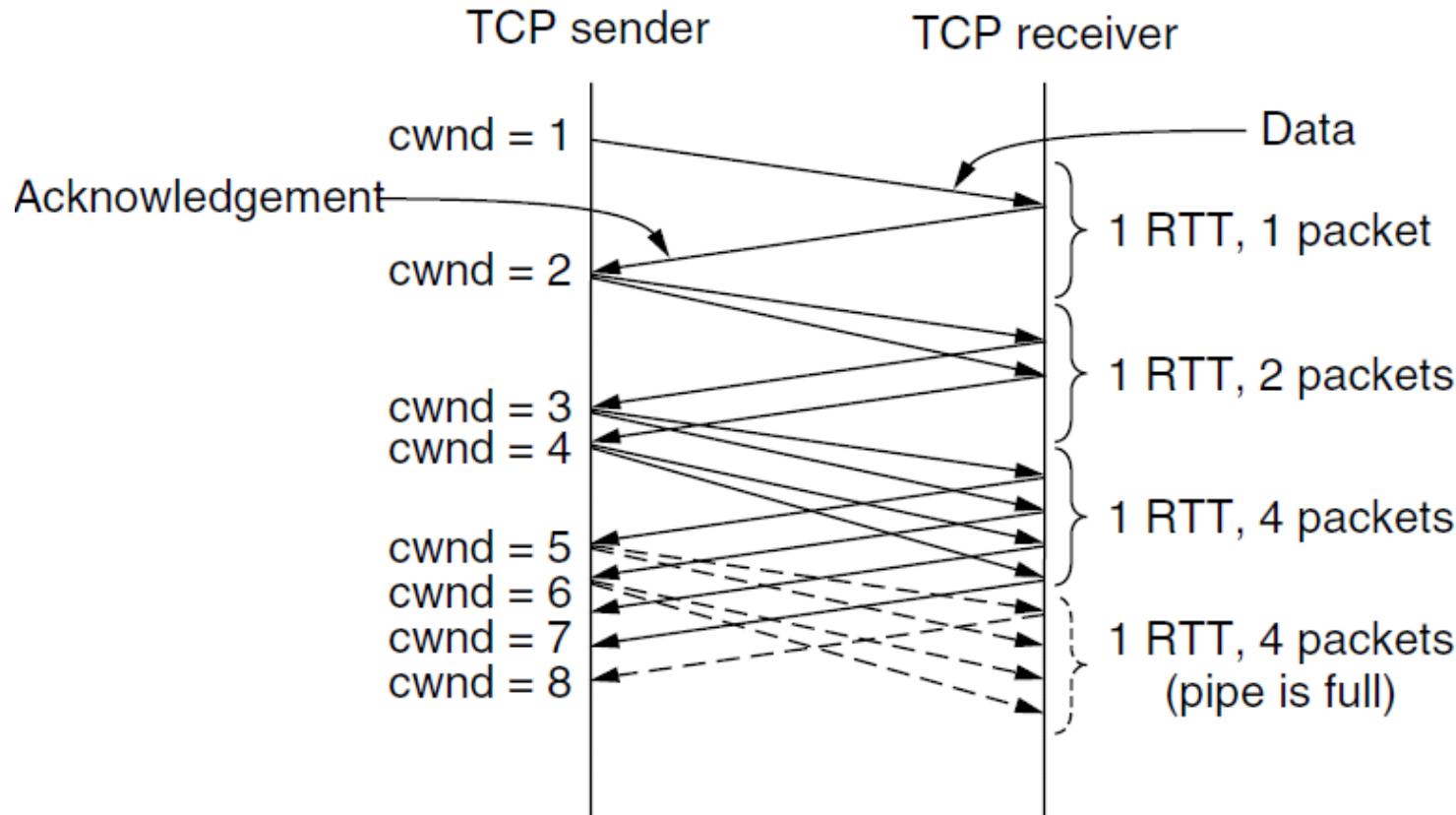
The TCP Approach to Congestion Control

- TCP adopts a defensive stance - open loop solution
 - At connection establishment, a suitable window size is chosen by the receiver based on its buffer size
 - If the sender is constrained to this size, then congestion problems will not occur due to buffer overflow at the receiver itself, but may still occur due to congestion within the network

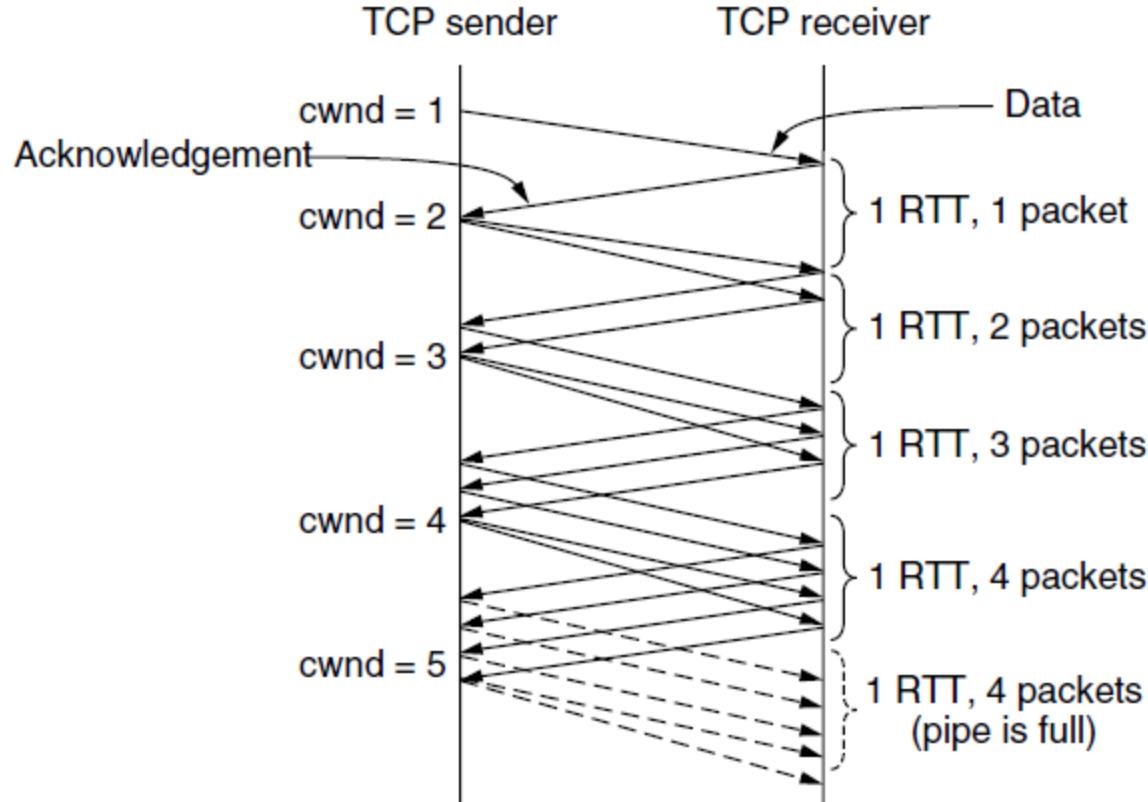
Incremental Congestion Control: Slow Start

- On connection establishment, the sender initializes the congestion window to the size of the maximum segment in use on the connection, and transmits one segment
- If this segment is acknowledged before the timer expires, the sender adds another segment's worth of bytes to the congestion window, making it two maximum size segments, and transmits two segments
- As each new segment is acknowledged, the congestion window is increased by one maximum segment size
- In effect, each set of acknowledgements doubles the congestion window - which grows until either a timeout occurs or the receiver's specified window is reached

Slow Start



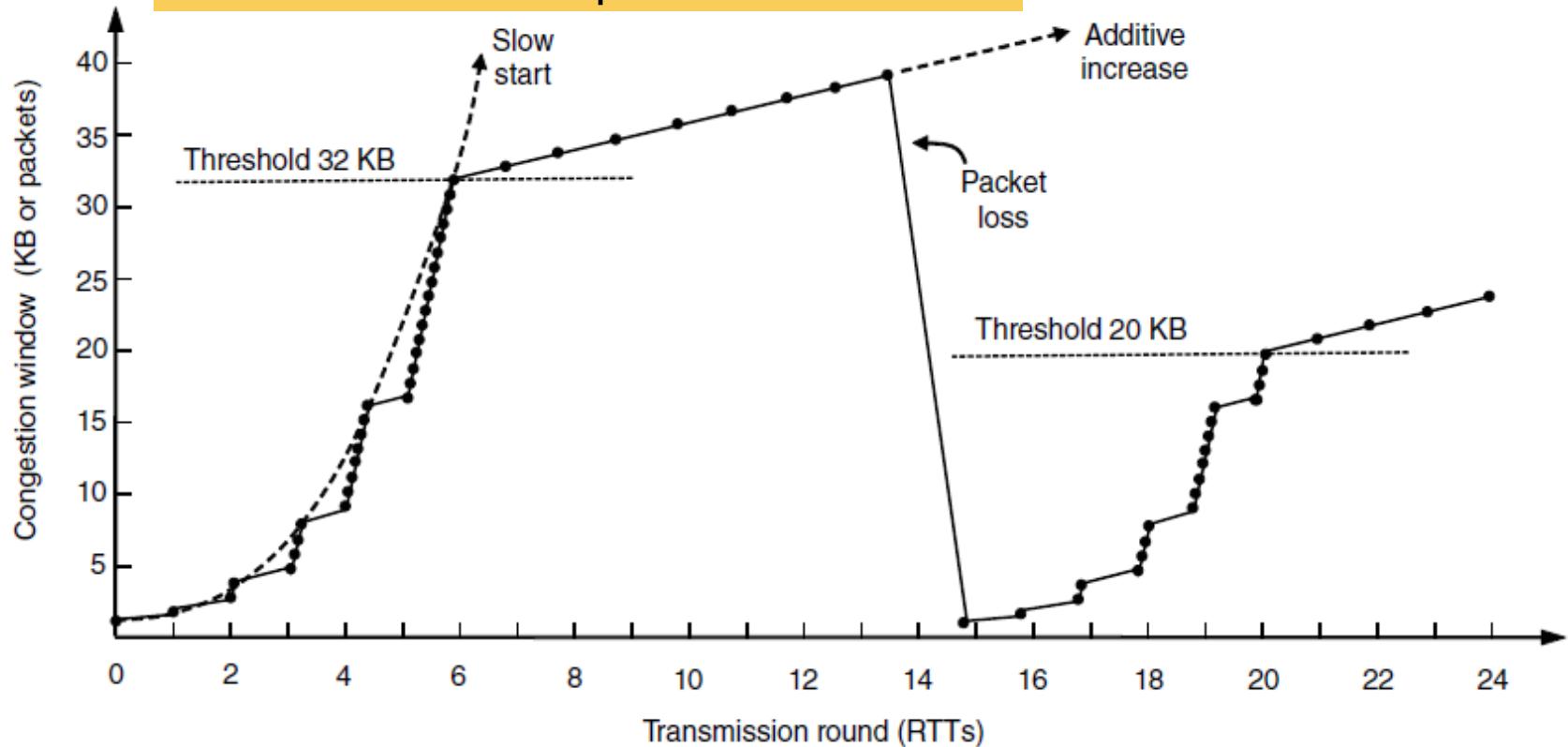
Additive increase



Internet Congestion Control Illustrated

Slow start followed by additive increase (TCP Tahoe)

Threshold is half of previous loss cwnd



General Network Policies Affecting Congestion

Layer	Policies
Transport	<ul style="list-style-type: none">• Retransmission policy• Out-of-order caching policy• Acknowledgement policy• Flow control policy• Timeout determination
Network	<ul style="list-style-type: none">• Virtual circuits versus datagram inside the subnet• Packet queueing and service policy• Packet discard policy• Routing algorithm• Packet lifetime management
Data link	<ul style="list-style-type: none">• Retransmission policy• Out-of-order caching policy• Acknowledgement policy• Flow control policy

Application Layer : DNS and SMTP

COMP90007

Internet Technologies

Chien Aun Chan

Outline

- Domain Name System
 - ❑ Division of Name Spaces
 - ❑ Services
 - ❑ Domain Name Properties
 - ❑ Resolving Domain Names
 - ❑ Name Servers

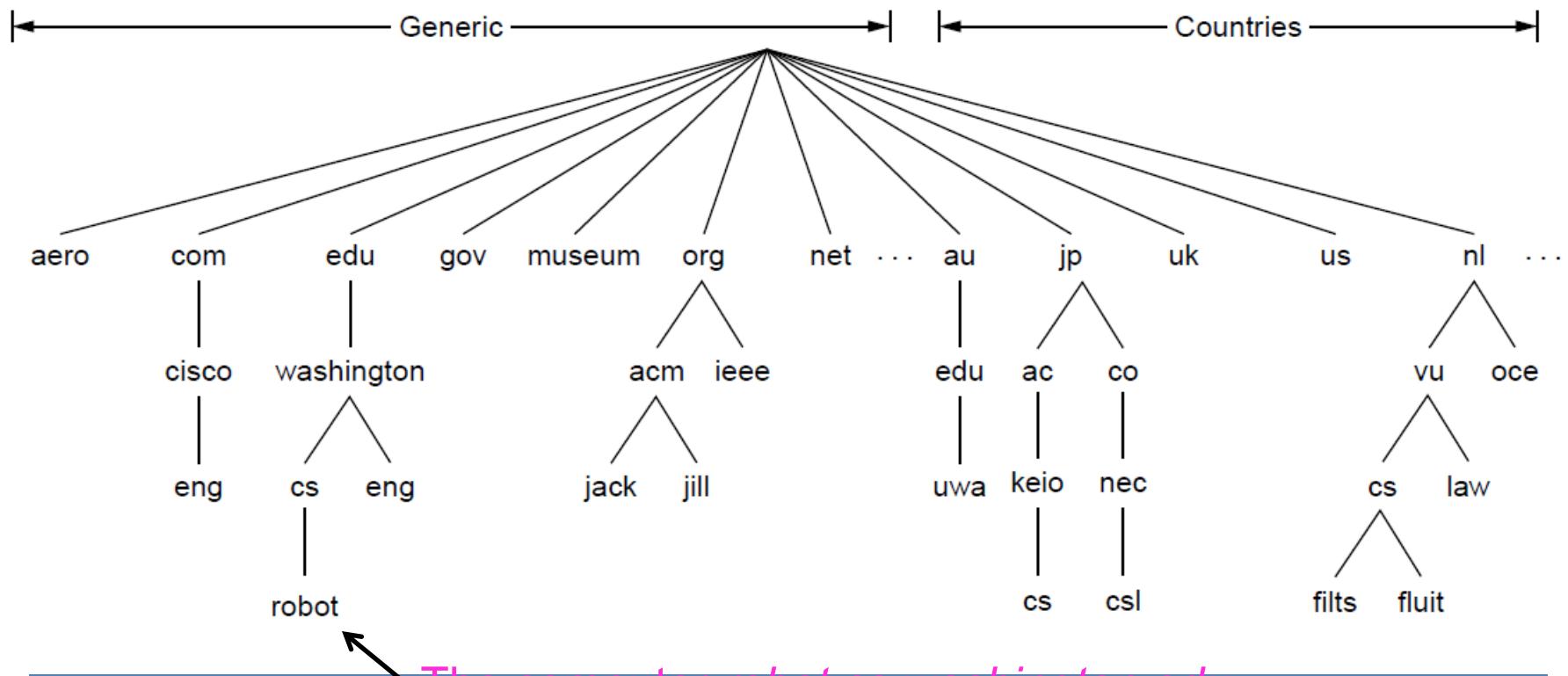
Application
Transport
Network
Link
Physical

DNS (Domain Name System)

- Problem?
 - IP address (32 bit), e.g., 121.7.106.83 – used for addressing datagrams
 - “name”, e.g., www.yahoo.com – used by humans
- Q: how do you map between IP address and name, and vice versa?
- Domain Name System:
 - *distributed database* implemented in a hierarchy of many *name servers*
 - *application-layer protocol* that allows a host to query the database in order to *resolve* names (address/name translation)
 - used by other application-layer protocols (http, ftp, smtp)

Conceptual Divisions of DNS Namespace

- A hierarchical naming convention; the top of the hierarchy is managed by ICANN (The Internet Corporation for Assigned Names and Numbers).



Name Space

- Internet is divided into over 250 top-level domains (TLD).
- Generic top-level domains are given next.

Domain	Intended use	Start date	Restricted?
com	Commercial	1985	No
edu	Educational institutions	1985	Yes
gov	Government	1985	Yes
int	International organizations	1988	Yes
mil	Military	1985	Yes
net	Network providers	1985	No
org	Non-profit organizations	1985	No
aero	Air transport	2001	Yes
biz	Businesses	2001	No
coop	Cooperatives	2001	Yes
info	Informational	2002	No
museum	Museums	2002	Yes
name	People	2002	No
pro	Professionals	2002	Yes
cat	Catalan	2005	Yes
jobs	Employment	2005	Yes
mobi	Mobile devices	2005	Yes
tel	Contact details	2005	Yes
travel	Travel industry	2005	Yes
xxx	Sex industry	2010	No

Why not centralize DNS?

- Single point of failure
- Traffic volume
- Distant centralized database
- Maintenance

Doesn't scale!

Outline

■ Domain Name System

- ❑ Division of Name Spaces
- ❑ Services
- ❑ Domain Name Properties
- ❑ Name Servers
- ❑ Resolving Domain Names

Application
Transport
Network
Link
Physical

DNS Services

- hostname to IP address translation
- host aliasing – alias names for canonical names
 - e.g., canonical `relay1.westcoast.enterprise.com` aliased to `www.enterprise.com`
- mail server aliasing
 - e.g., `Bob@relay1.westcoast.hotmail.com` aliased to `Bob@hotmail.com`
- load distribution
 - busy sites are replicated over multiple servers
 - a set of IP addresses is associated with one canonical name
 - DNS server rotates the order of the addresses to distribute the load

Domain Name Characteristics

- Domain names:
 - Are case insensitive
 - Can have up to 63 characters per constituent
 - Can have up to 255 chars per path
 - Can be internationalised (since 1999)
- Naming conventions usually follow either organisational or physical boundaries eg.
 - au.ibm.com / uk.ibm.com (for email)
 - ibm.com.au / ibm.co.uk (for web)
- Absolute domain names ends in a “.”
- Relative domain names partially specify the location and can be used only within the context of an absolute domain name

Outline

■ Domain Name System

- ❑ Division of Name Spaces
- ❑ Services
- ❑ Domain Name Properties
- ❑ Name Servers
- ❑ Resolving Domain Names

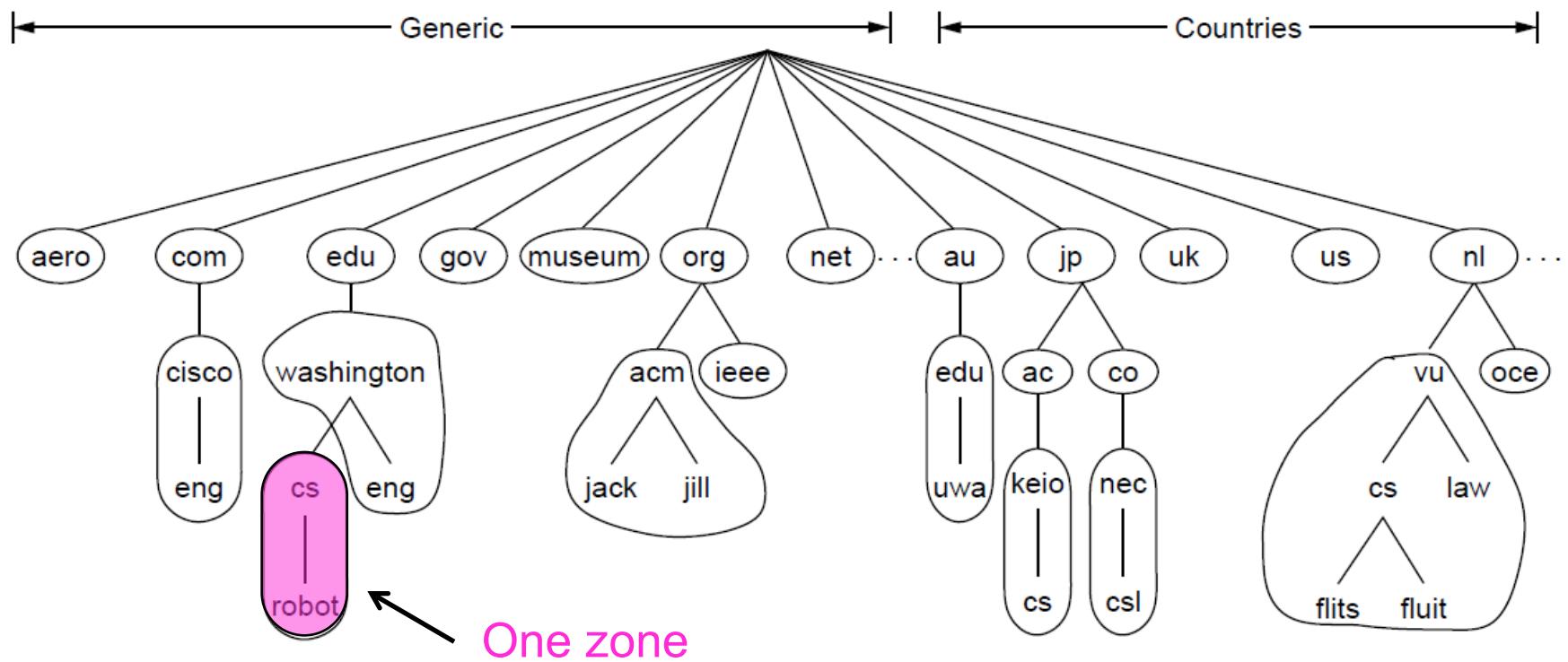
Application
Transport
Network
Link
Physical

Zone Name Servers

- DNS namespace divided into non-overlapping zones
- Each zone contains a part of the DNS tree and also name servers authoritative for that zone -
 - usually 2 name servers for a zone (called the primary and secondary name servers),
 - sometimes secondary is actually outside the zone (for reliability)
- Name servers are arranged in a hierarchical manner extending from a set of root servers

Name Servers

- The DNS name space is divided into nonoverlapping zones; each circled contains some part of the tree.



Root Name Servers

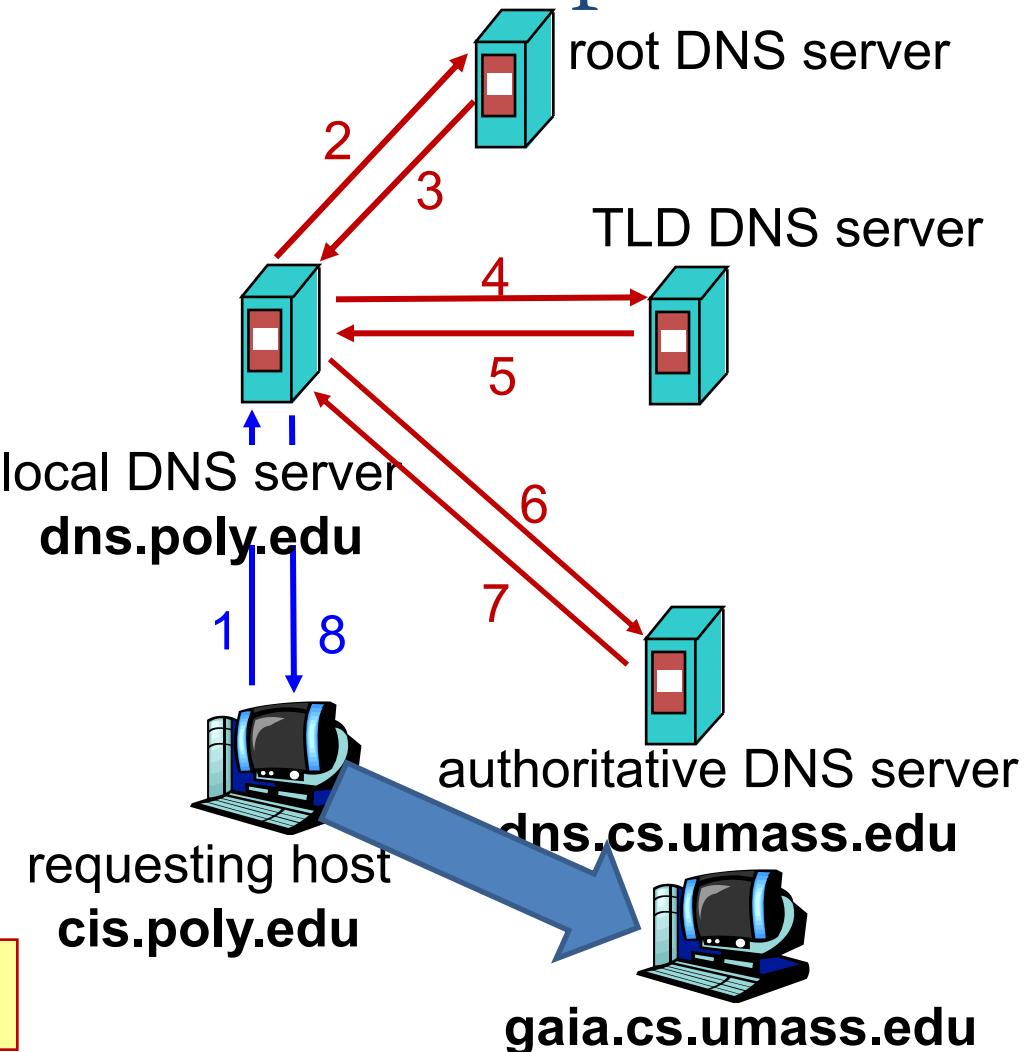
- The root servers form the authoritative cluster for enquiry in the event of locally-unresolvable name queries
- There are 13 root servers globally
 - In some cases, a root server is a cluster of servers that are in the anycast IP space

DNS In Action

- Finding the IP address for a given hostname is called name resolution and is done with the DNS protocol.
- Resolution:
 - Computer requests local name server to resolve
 - Local name server asks the root name server
 - Root returns the name server for a lower zone
 - Continue down zones until name server can answer
- DNS protocol:
 - Runs on UDP port 53, retransmits lost messages
 - Caches name server answers for better performance

DNS Name Resolution: Example

- host at cis.poly.edu wants IP address for gaia.cs.umass.edu
- iterated query:
 - contacted server replies with name of server to contact
 - “I don’t know this name, but ask this server”
- recursive query:
 - server obtains mapping on client’s behalf



Lots of network traffic!

DNS: Caching and Updating Records

- Once (any) name server learns a mapping, it *caches* the mapping
 - IP addresses of TLD servers typically cached in local name servers
 - root name servers not often visited
 - Cache entries timeout (disappear) after some time

DNS Software

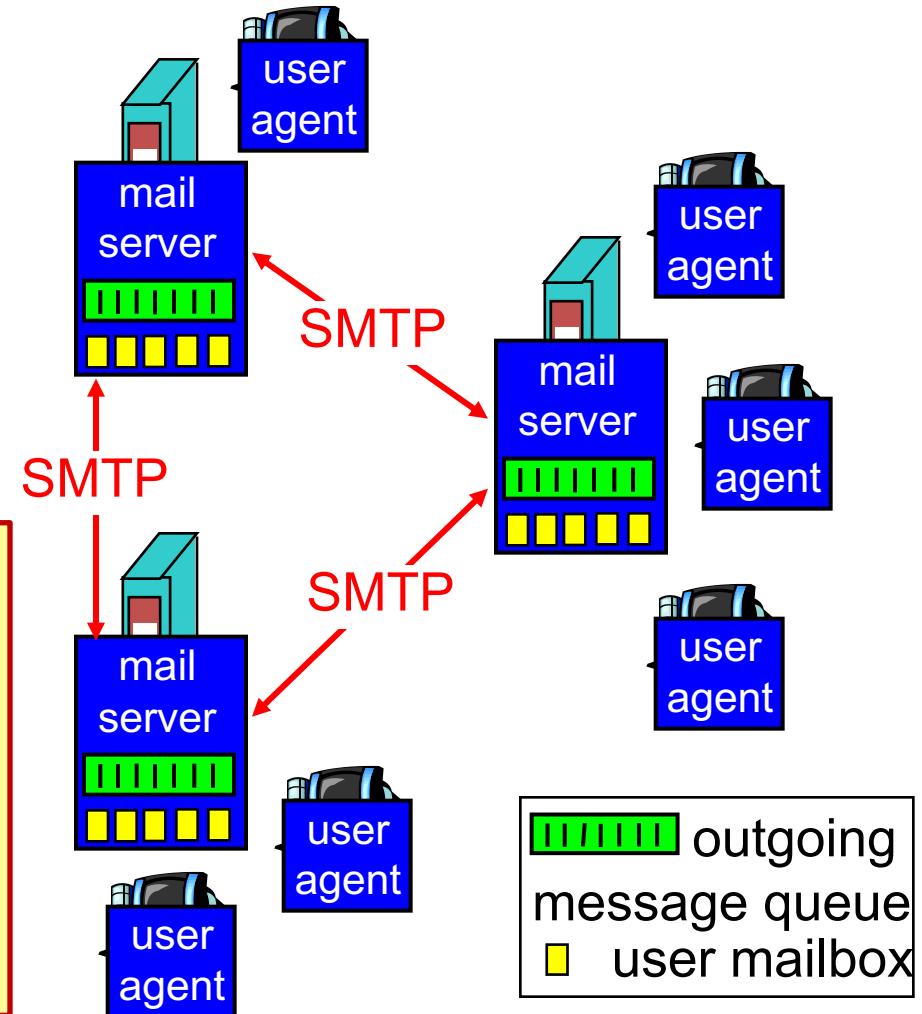
- DNS Server Software
 - BIND
 - djbdns
 - Microsoft Domain Name Server
- DNS Query Tools
 - nslookup
 - dig

Electronic Mail - Overview

- ❖ Three major components:
 - User agents
 - Mail servers
 - Simple Mail Transfer Protocol: SMTP

SMTP is used to **send** messages from the sender's

- **mail server** to the receiver's **mail server**
- **user agent** to the sender's **mail server**



The User Agent

- Basic functions: compose, report, display, dispose
- Envelope and contents: encapsulation of transport related information
 - Envelope - destination address, priority, and security level, all of which are distinct from the message itself
 - Mail servers use the envelope for routing
- Header and body: header - user agent control info; body for human recipient
 - contains control information for the user agents
- User must provide message, destination, optional other parameters
- Addressing scheme **user@dns-address**

Message Formats

- Message =
 - RFC821 envelope
 - Header fields (line of ASCII text with fieldname:value syntax)
- Blank line delimiter
- Message body

RFC 822: Message Header

- RFC 822 headers related to message header

- Date:
- Reply-To:
- Message-Id:
- In-Reply-To:
- References
- Keywords:
- Subject:

- RFC 822 allows users to invent new headers for private use but they must start with X-



Header	Meaning
To:	Email address(es) of primary recipient(s)
Cc:	Email address(es) of secondary recipient(s)
Bcc:	Email address(es) for blind carbon copies
From:	Person or people who created the message
Sender:	Email address of the actual sender
Received:	Line added by each transfer agent along the route
Return-Path:	Can be used to identify a path back to the sender

Multipurpose Internet Mail Extentsions (MIME) #1

- In the early days of email, messages were in English and used only ASCII – RFC822 was enough for these simple constraints.
- In time the inadequacy of RFC822 became apparent
 - Languages with accents (French, Spanish)
 - Non-Latin alphabets (eg Cyrillic)
 - Non-alphabetic language (eg Chinese, Japanese)
 - Messages with content other than text (audio, images)
- As a result, MIME (RFC 1341) was written (later updated in RFCs 2045-2049)

Header	Meaning
MIME-Version:	Identifies the MIME version
Content-Description:	Human-readable string telling what is in the message
Content-Id:	Unique identifier
Content-Transfer-Encoding:	How the body is wrapped for transmission
Content-Type:	Type and format of the content

Multipurpose Internet Mail Extentsions (MIME) #2

- MIME retains RFC822 format but adds structural elements to the message body and defines encoding rules for non-ASCII messages - thus leverage existing infrastructure for RFC822 services, and leaving MIME functionality changes to the user agent
- MIME has 5 additional message headers:
 - MIME-Version: identifies the MIME version
 - Content-Description: human readable describing contents
 - Content-Id: unique identifier
 - Content-Transfer-Encoding: how body is wrapped for transmission
 - Content-Type: type and format of content (e.g., text/plain, html, video, etc..)

MIME Types and Subtypes

Type	Example subtypes	Description
text	plain, html, xml, css	Text in various formats
image	gif, jpeg, tiff	Pictures
audio	basic, mpeg, mp4	Sounds
video	mpeg, mp4, quicktime	Movies
model	vrml	3D model
application	octet-stream, pdf, javascript, zip	Data produced by applications
message	http, rfc822	Encapsulated message
multipart	mixed, alternative, parallel, digest	Combination of multiple types

Message Transfer

- Transfer
 - SMTP
- Delivery
 - POP3 (Post Office Protocol 3)
 - IMAP (Internet Message Access Protocol)

SMTP

- Simple Message Transfer Protocol
- Simple ASCII protocol, operating on TCP port
25
- RFC 821: Simple Mail Transfer Protocol
- RFC 2821: Extended Simple Mail Transfer Protocol
- RFC 2821 delineated by HELO vs EHLO,
new features of 2821

SMTP Steps

- Basic steps SMTP(Simple Mail Transfer Protocol):

- Emloys readable text commands
 - User agent submits to MTA (mail transfer agent) on port 587 (Preferred RFC 4409)
 - One MTA to the next MTA on port 25
 - Other protocols for final delivery (IMAP, POP3)

Internet Message Access Protocol (IMAP)
Post Office Protocol 3 (POP3)

Message Transfer

application-layer handshake

```
C: telnet servername 25 ← establishing connection
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM:<alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO:<bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with " ." on a line by itself
C: From: alice@crepes.fr
C: To: bob@hamburger.edu
C: ... message ...
C: .
S: 250 Message accepted for delivery
C: QUIT ← closing connection
S: 221 hamburger.edu Closing connection
```

end of message

Email Software

- UA's
 - Outlook (+Express), Eudora, Pine, Mutt, elm, Evolution, Thunderbird
- Agent Middleware (User -> Server)
 - Pop3d, qpopper, fetchmail
- MTA's
 - sendmail, Qmail, PostFix, Exchange, MDaemon, IMail

Summary – Core Internet Applications

- DNS UDP
 - Describe operation of DNS lookup
- Email TCP
 - Explain which functions should be performed in Message Transfer Agent or User Agent

Multimedia

COMP90007
Internet Technologies

Chien Aun Chan

Outline: Application Layer

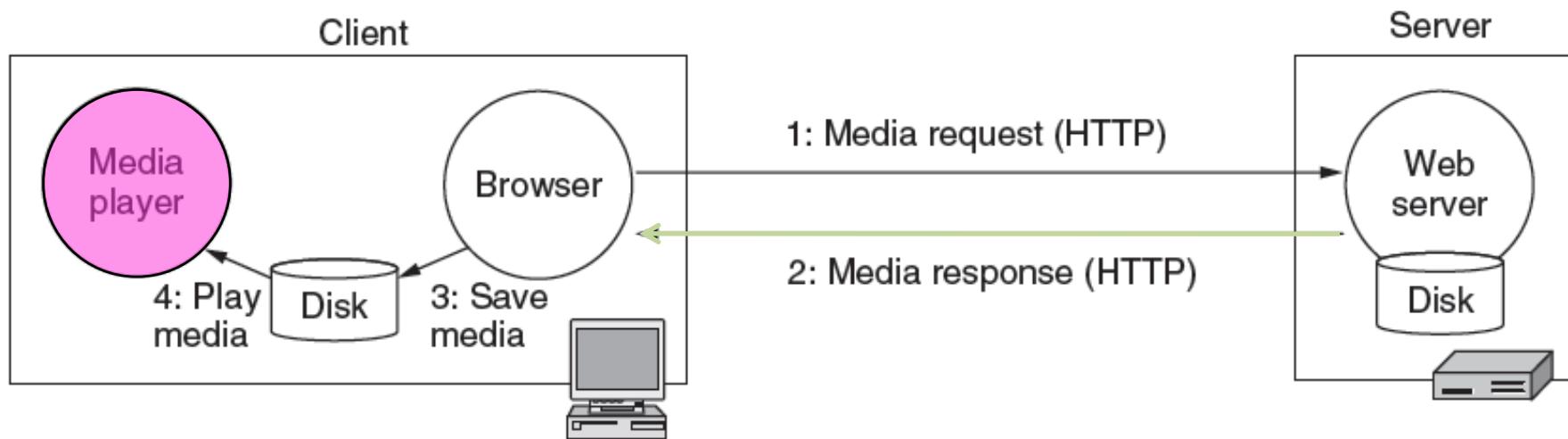
- Multimedia Networks
 - Audio
 - Video
- Audio and video have become key types of traffic, e.g., voice over IP, and video streaming.

Application
Transport
Network
Link
Physical

Characteristics of Multimedia Networks

- Higher bandwidth requirements
- Higher QoS requirement (i.e., delay sensitive)
- New infrastructure models
 - Need separate multimedia servers from web servers
- New service providers
 - Streaming multimedia service providers are often separated and highly specialised, compared to traditional web hosts

A Basic Model for Multimedia on the Web



Problems with the Basic Model

- The entire media file must be transmitted over the network before playback starts, causing delay in user experience (e.g., to transmit a 5Mb file over a 56Kbps link takes about 5 minutes)
- Basic model assumes mainly point-to-point media distribution rather than a point-to-multipoint (broadcast) distribution model
- Does not scale
- Basic model assumes the browser/plugin/helper integration and traditional service types

Streaming Media Protocols

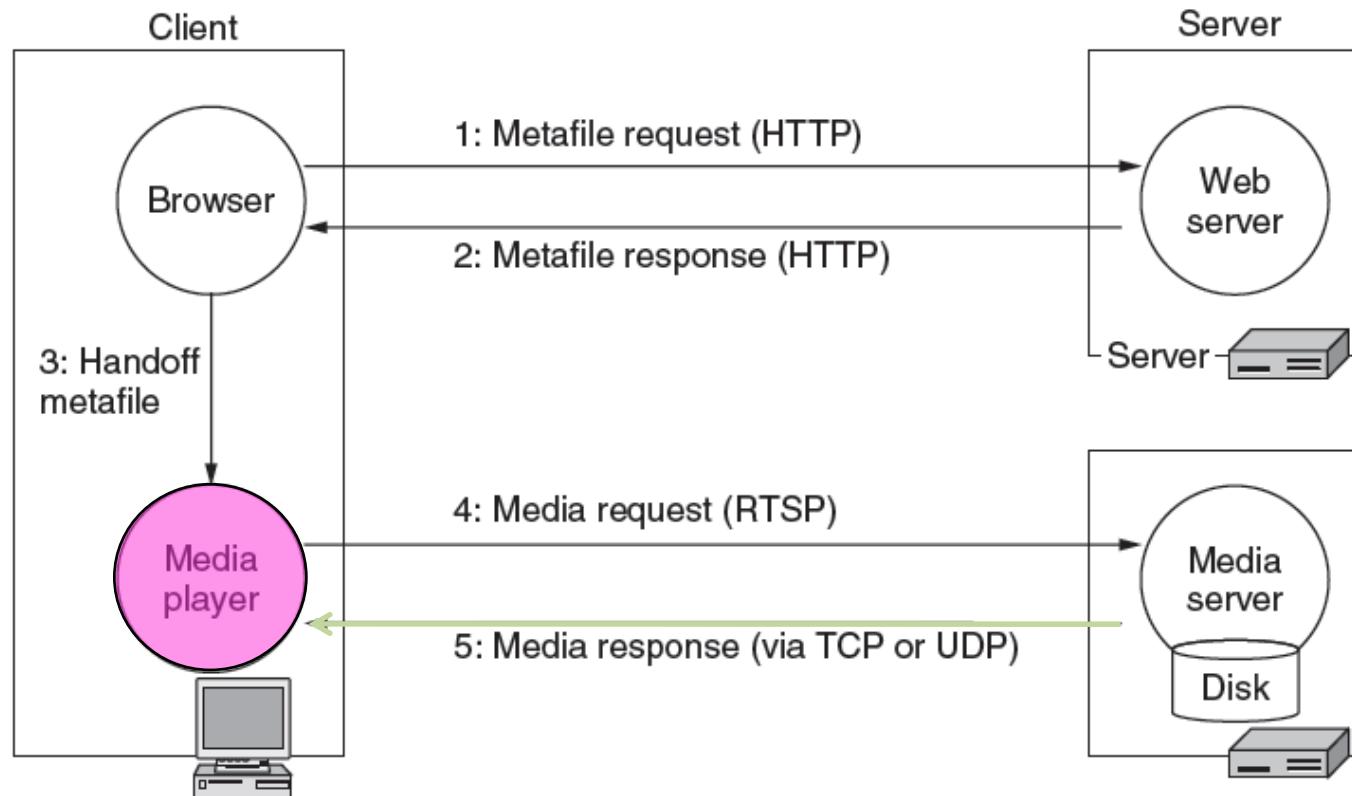
- Transport Protocols
 - TCP
- Open Protocols
 - HTTP
 - RTP - Real-time Transport Protocol (RFC 1889)
 - RTSP - Real Time Streaming Protocol (RFC 2326)
 - MPEG-4 (ISO)
- Closed Protocols
 - Real Networks' RealAudio
 - Microsoft's Windows Media
 - Apple's QuickTime

The Role of Multimedia Playback Software

- 4 main tasks of the multimedia playback software
 - Manage the user interface (e.g., volume, playback, next, etc..)
 - Handle transmission errors in conjunction with transport protocols
 - Using RTP/UDP errors will likely occur, playback software must manage them gracefully
 - Decompress the multimedia files (codecs)
 - Eliminate jitter
 - Small buffer, quick playback but susceptible to jitter/delay
 - Large buffer, delay at start of playback while buffer fills, but less susceptible to delay/jitter

Manage the user interface

Effective streaming starts the playout during transport



Handling Errors: Streaming Stored Media

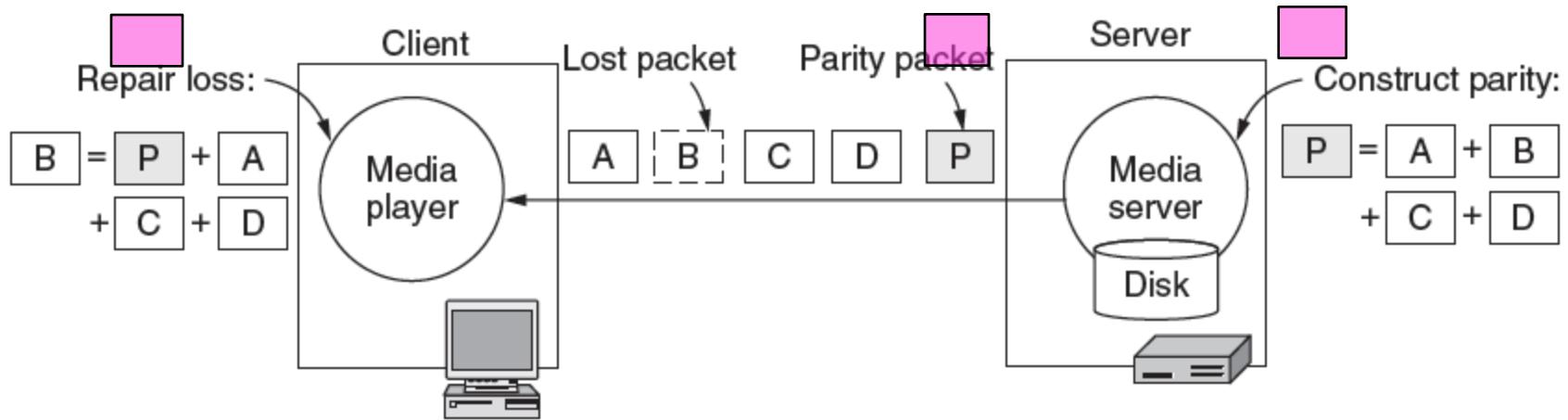
Strategy	Advantage	Disadvantage
Use reliable transport (TCP)	Repairs all errors	Increases jitter significantly
Add FEC (e.g., parity)	Repairs most errors	Increases overhead, decoding complexity and jitter
Interleave media	Masks most errors	Slightly increases decoding complexity and jitter

Forward Error Correction (FEC) is simply the error-correcting coding. For every four data packets a fifth. The parity packet, P, contains redundant bits that are the parity or exclusive-OR sums of the bits in each of the four data packets.

Streaming Stored Media

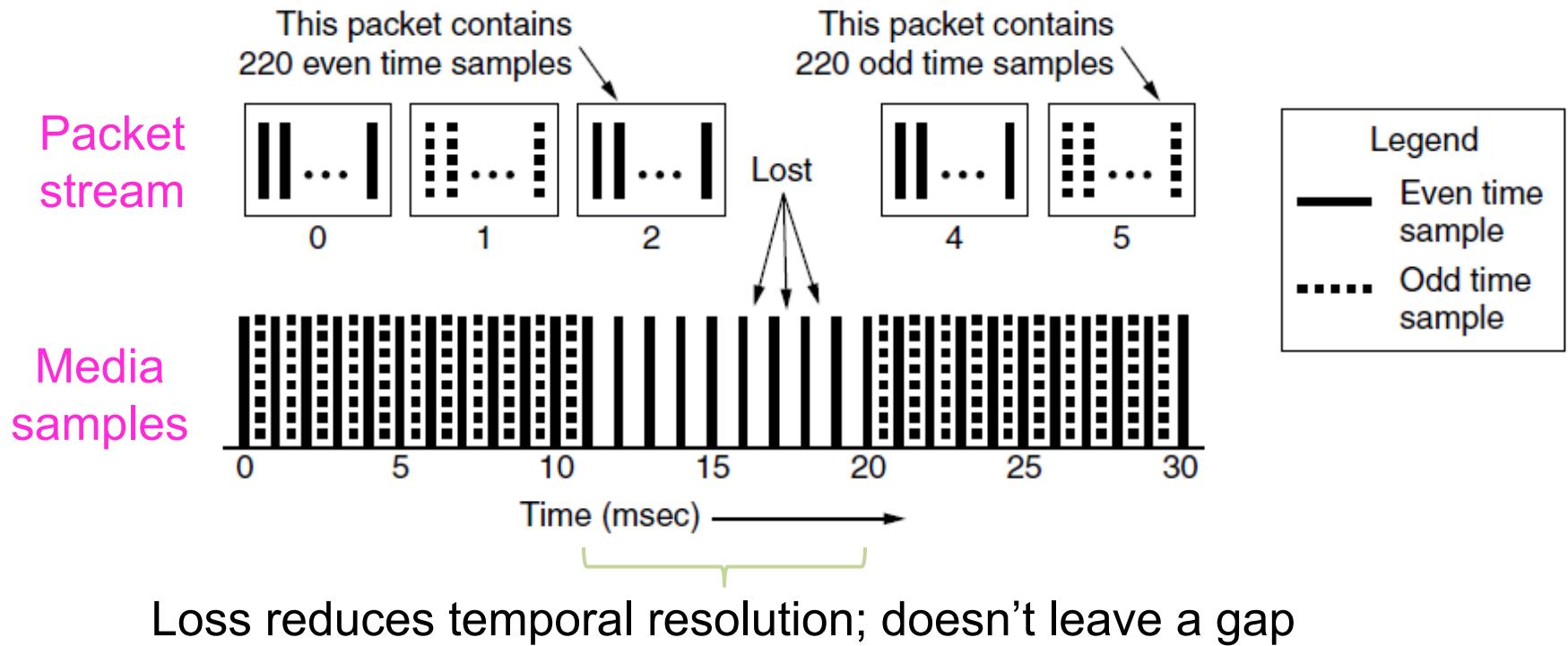
Use of **Parity packets** can repair one lost packet in a group of N

- Cons: Decoding is delayed for N packets



Streaming Stored Media

Interleaving spreads nearby media samples over different transmissions to reduce the impact of loss

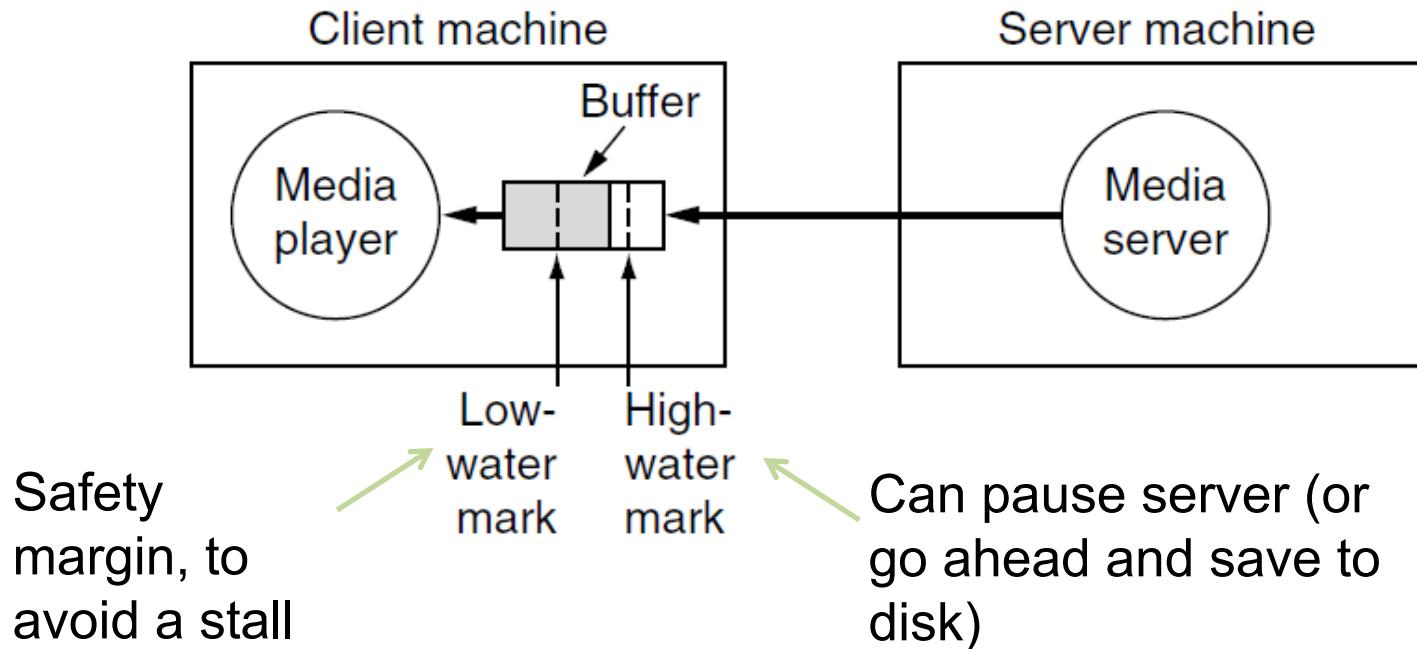


Jitter Management

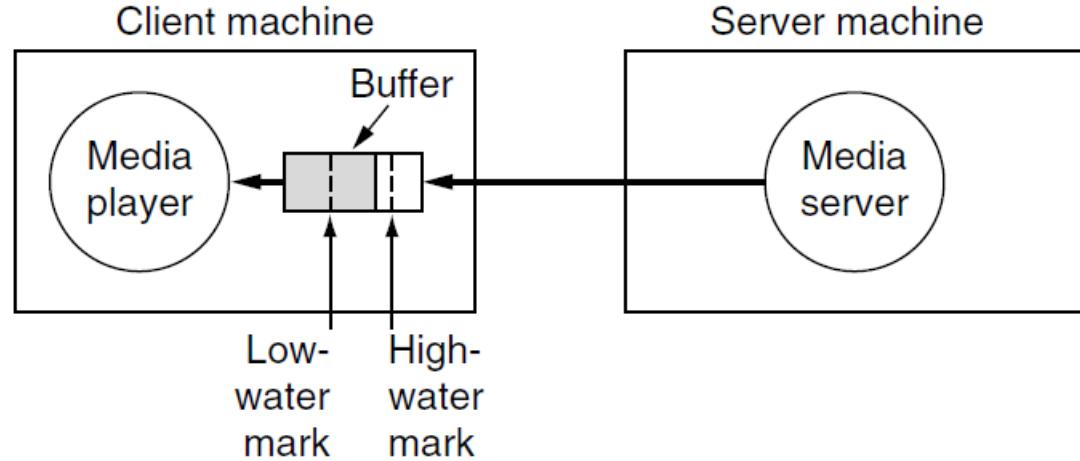
- Similarly to router buffering, multimedia software buffers streamed media sources prior to transmission
- Buffering is a defensive mechanism to reduce jitter (variance in average packet arrival times)
- Ideally the stream buffer will continue to be filled at the same rate the stream is played back to the user

Jitter Management

Jitters happens because of variable bandwidth and loss/retransmissions



Example



An video streaming server takes 10 ms to communicate with a client media player. If the media player has a 2 MB buffer, what can you say about the position of the low and high-water mark? Assume the media being streamed has a bitrate of 1 Mbps and the server sends data at a fix rate of 2 Mbps.

Ans: It takes 10 ms to send a pause command to the server and another 10 ms for the data already in the network to drain after the server stops sending.

HWM: Bandwidth delay product= $20 \text{ ms} \times 2 \text{ Mbps} = 5,000 \text{ bytes}$ will arrive, so the high-water mark should be **at least** 5,000 bytes below the top to avoid buffer overflow. **To be absolutely safe, set the mark to be 20,000 bytes from the top.**

LWM: $20 \text{ ms} \times 1 \text{ Mbps} = 2,500 \text{ bytes}$ for the client to request data from the server and start receiving it. 2,500 bytes **will be played** during this time. Hence, the low-water mark should be **at least** 2,500 bytes and probably 15,000 bytes to be safe.

Real Time Streaming (RTSP)

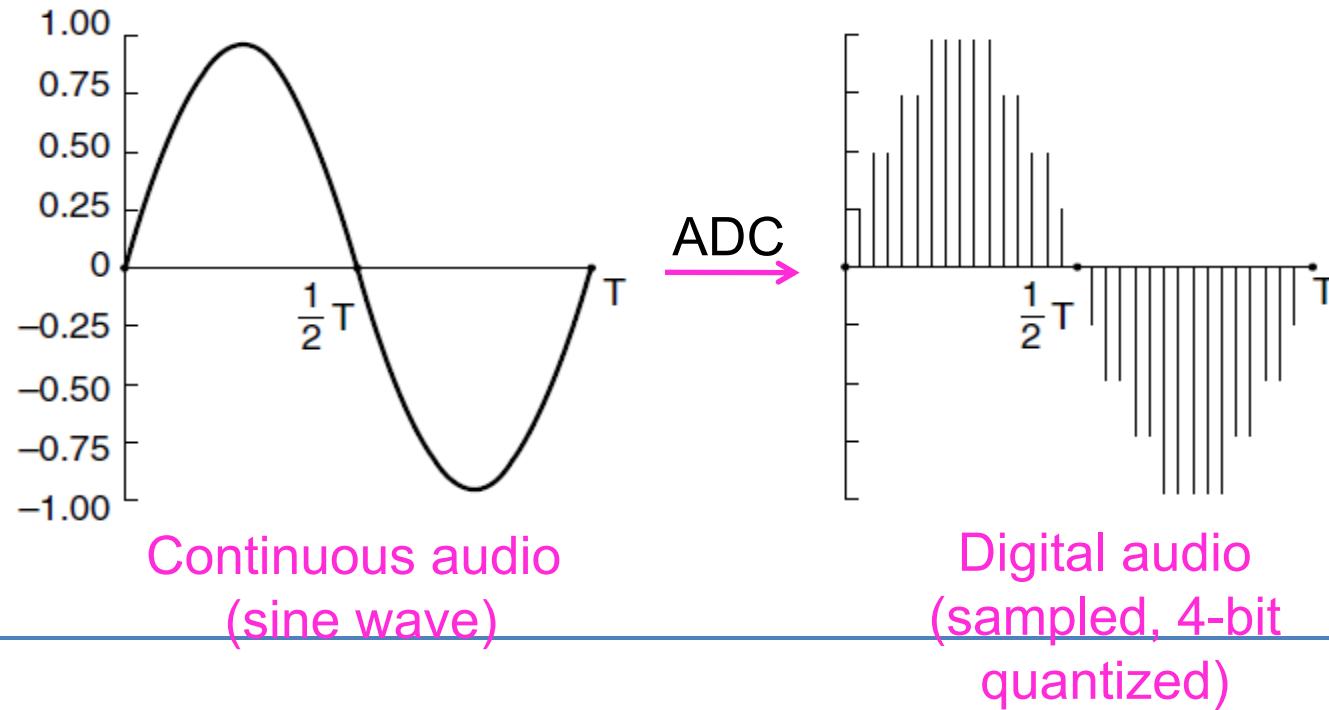
- RTSP has mechanisms to start and stop the flow of streaming media: RTSP commands are sent from player to server to adjust streaming
- Defined in RFC 2326;
- For Data stream, RTP over UDP or RTP over HTTP over TCP
- RTSP commands are sent from player to server to adjust streaming

Command	Server action
DESCRIBE	List media parameters
SETUP	Establish a logical channel between the player and the server
PLAY	Start sending data to the client
RECORD	Start accepting data from the client
PAUSE	Temporarily stop sending data
TEARDOWN	Release the logical channel

Digital Audio: Basics

ADC (Analog-to-Digital Converter) produces digital audio from a microphone

- Telephone: 8000 8-bit samples/second (64 Kbps); computer audio is usually better quality



Audio Compression

- Audio CD can represent frequencies up to 22.05kHz, hence Nyquist (sample) rate is 44.1 kHz, Stereo channels: 44100 samples/sec, 16 bits/sample = $2 \times 44100 \times 16 = 1,411,200$ bits/s
双声道
- The key property of perceptual coding is that some sounds can mask other sounds and at any point those sounds are identified and encoded for transmission
- Frequency masking: Some sounds mask/hide others so there is no point encoding them.
- Temporal masking: Human ears can miss soft sounds immediately after loud sounds, takes time for the ear to adjust, no need

Digital Video

- Video is digitized as pixels (sampled, quantized)
 - TV quality: 640x480 pixels, 24-bit color, 30 times/sec
~ 200Mbs uncompressed
- Video is sent compressed due to its large bandwidth
 - Lossy compression exploits human perception
 - E.g., JPEG for still images, MPEG, H.264 for video
 - Large compression ratios (often 50X for video)

Compression with JPEG

- JPEG lossy compression sequence for one image
- JPEG often provides compression ratios of 20:1
- JPEG compression is symmetric, decoding takes as long as encoding

MPEG Standard

- MPEG - Motion Picture Experts Group
- MPEG can compress both audio and video together (using synchronised streams)
- The evolution of MPEG
 - MPEG-1: VCR quality at 1.2 Mbps (40:1)
 - MPEG-2: Broadcast quality at 4-6Mbps (200:1)
 - MPEG-4: DVD quality at 10Mbps (1200:1)

Video over the Web

- Embedded with web content
 - Streaming servers required to deliver content over ordinary connection
- Integrated with other consumer services eg cable television (video on demand)
 - Content stored on central video servers
 - Delivered via a shared network to consumers

Outline

- **Voice over IP (VOIP)**
 - VOIP Benefits
 - VOIP Technologies
 - Protocols

The Emergence of VOIP

- Voice services becoming applications on top of data networks are being driven by a number of factors:
 - Data has overtaken voice as the primary traffic on many networks originally built for voice
 - PSTN infrastructure is not flexible enough for the rapid deployment of new features
 - PSTN technologies are largely incompatible with the convergence of data/voice/video
 - The architecture built primarily for voice is not flexible enough to carry data
 - Network providers are increasingly looking to leverage investment in network infrastructure by bringing new services to data networks
- Where suitable data networks are already in existence, the evolution of audio encoding technologies has allowed voice to be transmitted over data links - hence VOIP

Benefits of VOIP

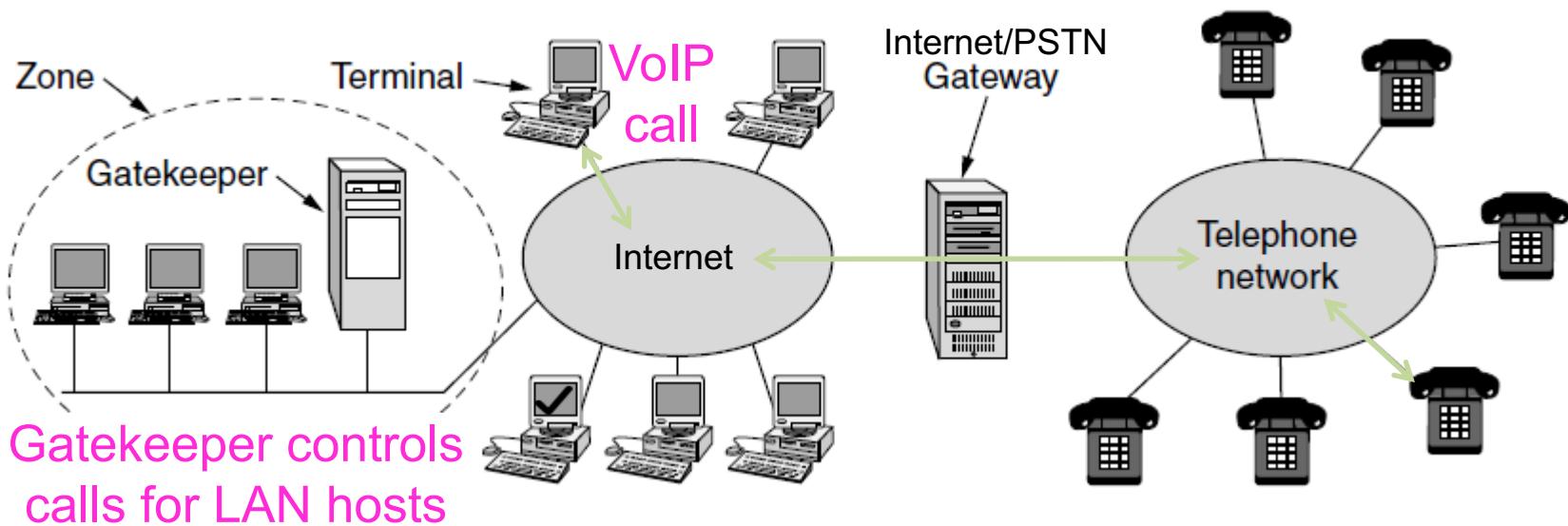
- Financial savings
- Consolidated infrastructure
- Flexible infrastructure
- Standards based voice and data

VOIP Technologies

- Within the VOIP domain, there are 3 distinct models for service provision
 - infrastructural - PSTN/PABX integration
 - virtual - media gateways, virtual directories
 - value-added - voice mail
- Alternative VOIP Technologies
 - H.323
 - SGCP (Simple Gateway Control Protocol) and MGCP
 - SIP

H.323 architecture for Internet telephony

supports calls between Internet computers and PSTN phones.

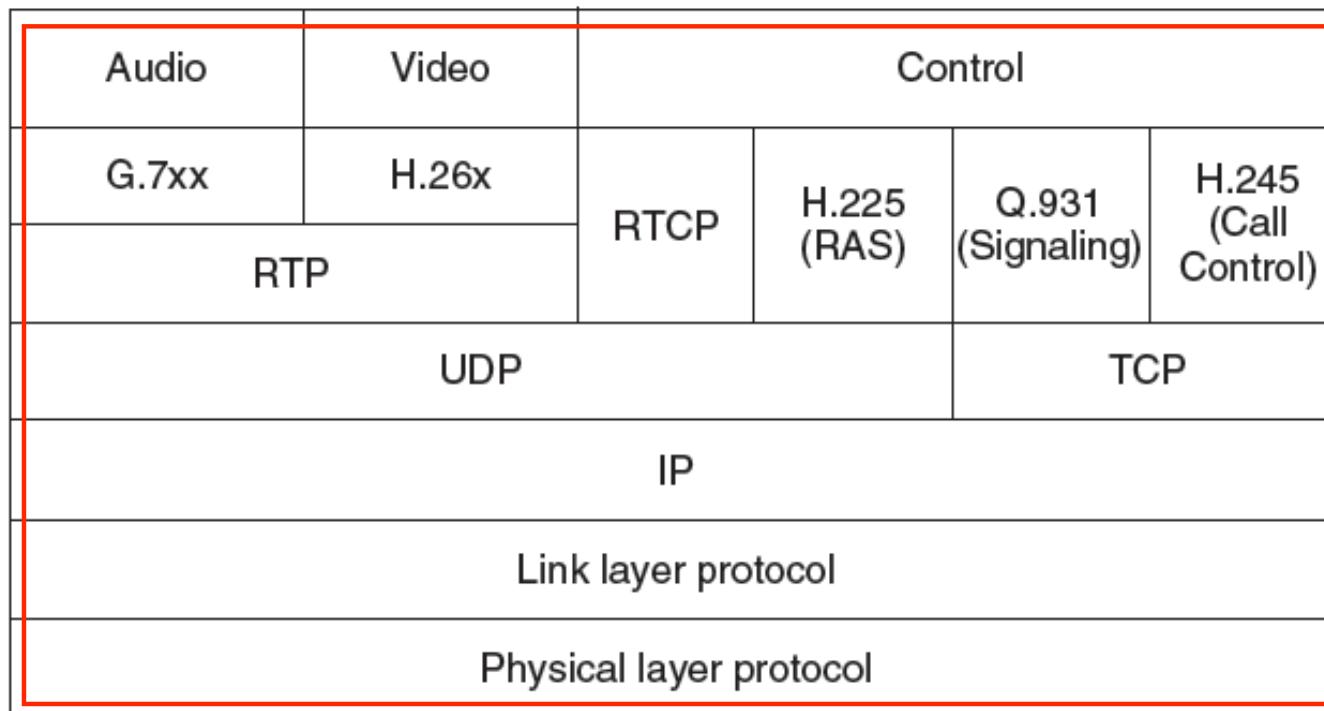


H.323

- H.323 is an international standard that specifies how multimedia traffic is carried over packet networks
- H.323 integrates existing standards to provide a layered protocol stack
- H.323 was originally developed to enable multimedia applications to run over **unreliable data networks** - includes support for audio (including VOIP), video and data sharing services within a single protocol stack

The H.323 Protocol Stack

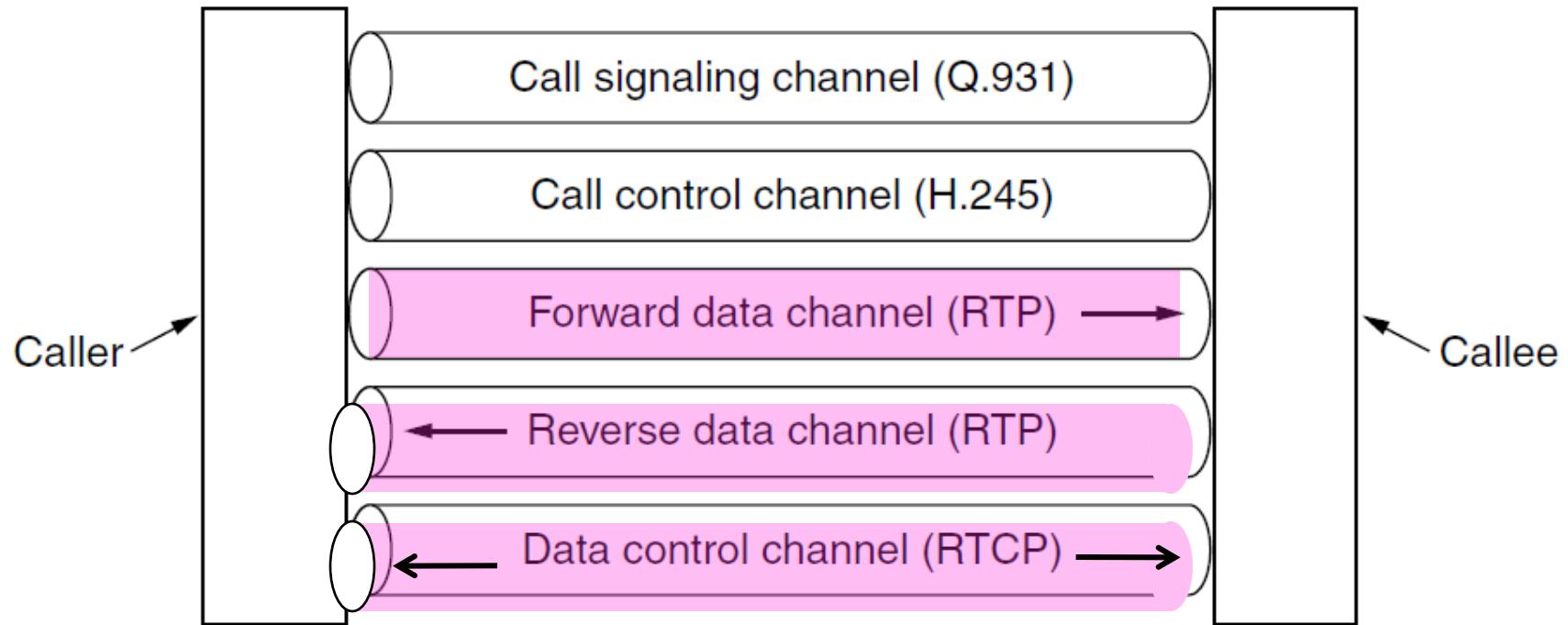
- Call is digital audio/video over RTP/UDP/IP
- Call setup is handled by other protocols (Q.931 etc.)



The H.323 Protocol Stack

- G.7xx e.g. G.711 encodes a voice channel by sampling 8000 times/sec with 8 bit sample rate, giving uncompressed 64kbps speech
 - Other G.7xx compression protocols exist that have different quality/bandwidth tradeoffs
- H.245 protocol handles negotiation to decide which compression algorithm to use, and the bit-rate.
- Q.931 protocol handles connection establishment and release, provides dial-tones, ringing sounds and other telephony functions.
- H.225 (RAS = Registration/Admission/Status) protocol is needed to talk to the gatekeeper, if the client is behind one. e.g. LAN.
- RTP is responsible for data flow once negotiation has been completed.
- RTCP (RTP Control Protocol) manages congestion control, and if the call contains video as well as audio, RTMP handles the synchronisation

Logical channels in H.323 VOIP



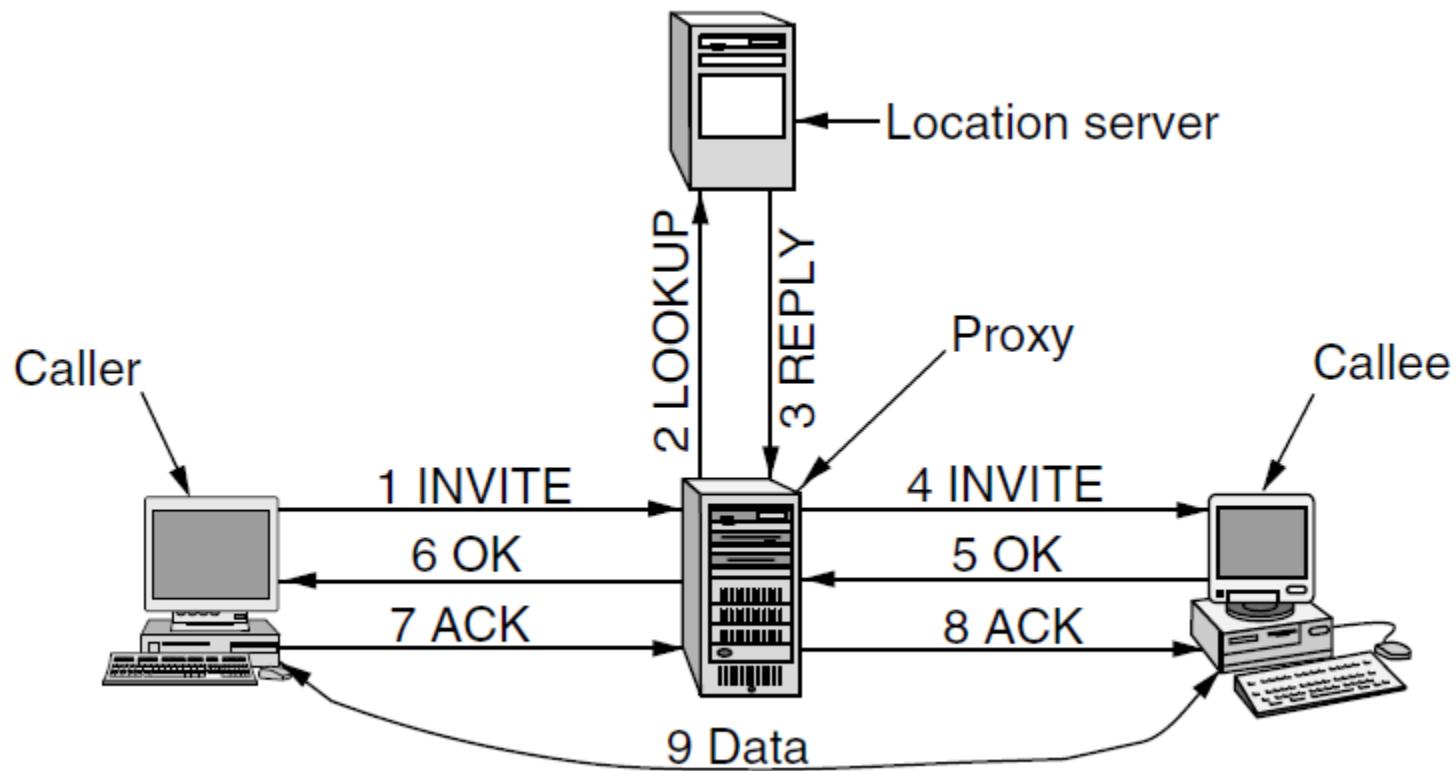
Session Initiation Protocol (SIP)

- SIP codified in RFC 3261
- Modelled on HTTP, simple ASCII based protocol, method + parameters, similar headers to MIME
- SIP Architecture
 - A single network module rather than a complete protocol suite
- SIP only handles the setup, management and termination of sessions - requires other protocols such as RTP/RTCP for data transport. SIP is an application layer protocol and can run over UDP or TCP
- SIP Functionality
 - Two party, multiparty and multicast
 - Callee location, callee capabilities, call setup, call termination
- SIP Addressing: addressing based on a URL type schema
 - `sip:badenh@estragon.cs.mu.oz.au`
 - SIP URLs can contain IPv4 or IPv6 addresses or actual telephone numbers

SIP Methods

Method	Description
INVITE	Request initiation of a session
ACK	Confirm that a session has been initiated
BYE	Request termination of a session
OPTIONS	Query a host about its capabilities
CANCEL	Cancel a pending request
REGISTER	Inform a redirection server about the user's current location

SIP Methods



H.323 and SIP Compared

Item	H.323	SIP
Designed by	ITU	IETF
Compatibility with PSTN	Yes	Largely
Compatibility with Internet	Yes, over time	Yes
Architecture	Monolithic	Modular
Completeness	Full protocol stack	SIP just handles setup
Parameter negotiation	Yes	Yes
Call signaling	Q.931 over TCP	SIP over TCP or UDP
Message format	Binary	ASCII
Media transport	RTP/RTCP	RTP/RTCP
Multiparty calls	Yes	Yes
Multimedia conferences	Yes	No
Addressing	URL or phone number	URL
Call termination	Explicit or TCP release	Explicit or timeout
Instant messaging	No	Yes
Encryption	Yes	Yes
Size of standards	1400 pages	250 pages
Implementation	Large and complex	Moderate, but issues
Status	Widespread, esp. video	Alternative, esp. voice



Evolving Voice over Data Solutions

- First voice-data integration technologies were designed to eliminate long distance phone toll charges by providing tie lines between PABXs over a WAN infrastructure
- Later support for analogue telephony devices was introduced to allow off-premises extensions to PABX
- As data networks expanded, enterprise wide call handling began to migrate towards the data network - shorter call forwarding distances between PABX and WAN gateways
- With increased voice traffic, connection admission control became a more significant issue
- With larger networks, dial plans and directory services became essential
- Centralised call control through the use of H.323 gatekeeper functions allowed voice and data transmissions to be regulated at a single point

Future of Network-Centric Telephony Applications

- As integration increases, new solutions are emerging – typically allowing for packetised voice technologies to replace PABX with end to end solutions
- Generally 2 categories of technologies and application architectures
 - **UnPABX:** Server based call routing, with direct connections to data network, trunk telephony network and analogue telephony networks
 - **LAN-PABX:** Telephony on the desktop - rather than actual telephone handsets, the telephony function is in software on a workstation

Incentives for Packet Based Telephony

- Un-PABX systems typically cost less than systems they replace
- LAN-PABX systems bring much greater flexibility to end users
- Both leverage existing infrastructure - either the convergence of communications infrastructure or wired/wireless connections to the workstation
- Fully integrated applications, which allow manipulation of complex software services via telephony interfaces will likely drive the convergence of voice and data even further

Summary - Multimedia

- Multimedia networks
 - General operations of audio and video streaming
 - Describe techniques for jitter management
- VoIP
 - Contrast H.323 and SIP
 - Explain steps in SIP call establishment

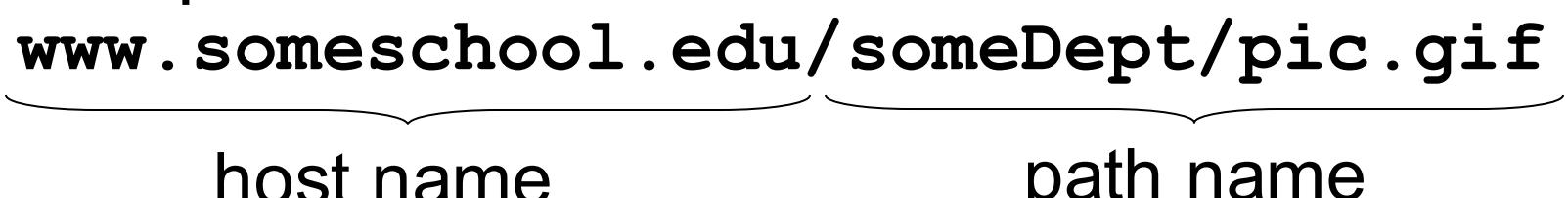
HTTP Protocol

COMP90007
Internet Technologies

Chien Aun Chan

- World Wide Web
 - HTTP
 - Web markup languages
 - Web scripting languages
 - Client and Server software

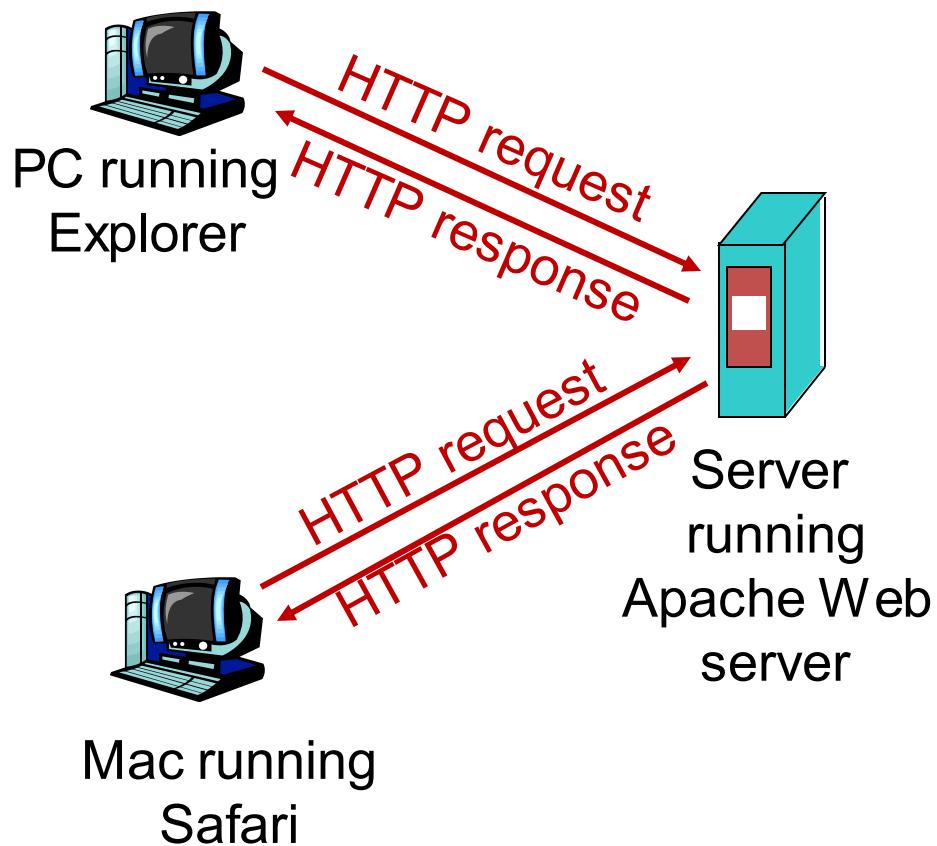
Web and HTTP – Review

- A web page consists of objects
- An object can be HTML file, JPEG image, Java applet, audio file, ...
- A web page consists of a base HTML file which includes several referenced objects
- Each object is addressable by a URL (uniform resource locator)
- Example URL:
A URL is shown as "www . someschool . edu / someDept / pic . gif". Two curly braces are placed under the URL. The first brace covers "www . someschool . edu", labeled "host name" below it. The second brace covers "/ someDept / pic . gif", labeled "path name" below it.
www . someschool . edu / someDept / pic . gif

HTTP Overview (I)

HTTP: HyperText Transfer Protocol

- Web is an application layer protocol
- client/server model
 - **client:** browser that requests, receives and displays Web objects
 - **server:** Web server sends objects in response to requests



HTTP Connections

- Non-persistent HTTP

- at most one object sent over a TCP connection

- Persistent HTTP

- multiple objects can be sent over a single TCP connection between client and server

Non-persistent HTTP (I)

suppose user enters URL:

www. someSchool. edu /someDepartment /home . index

1a. HTTP client initiates TCP connection to HTTP server (process) at www. someSchool. edu on port 80

contains text and references to 10 images

1b. HTTP server at host www. someSchool. edu waiting for TCP connection at port 80. Accepts connection, notifying client

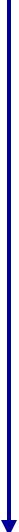
2. HTTP client sends a HTTP **request message** (containing URL) into TCP connection socket. Message indicates that client wants object **someDepartment / home . index**

3. HTTP server receives request message, forms **response message** containing requested object, and sends message into its socket

time

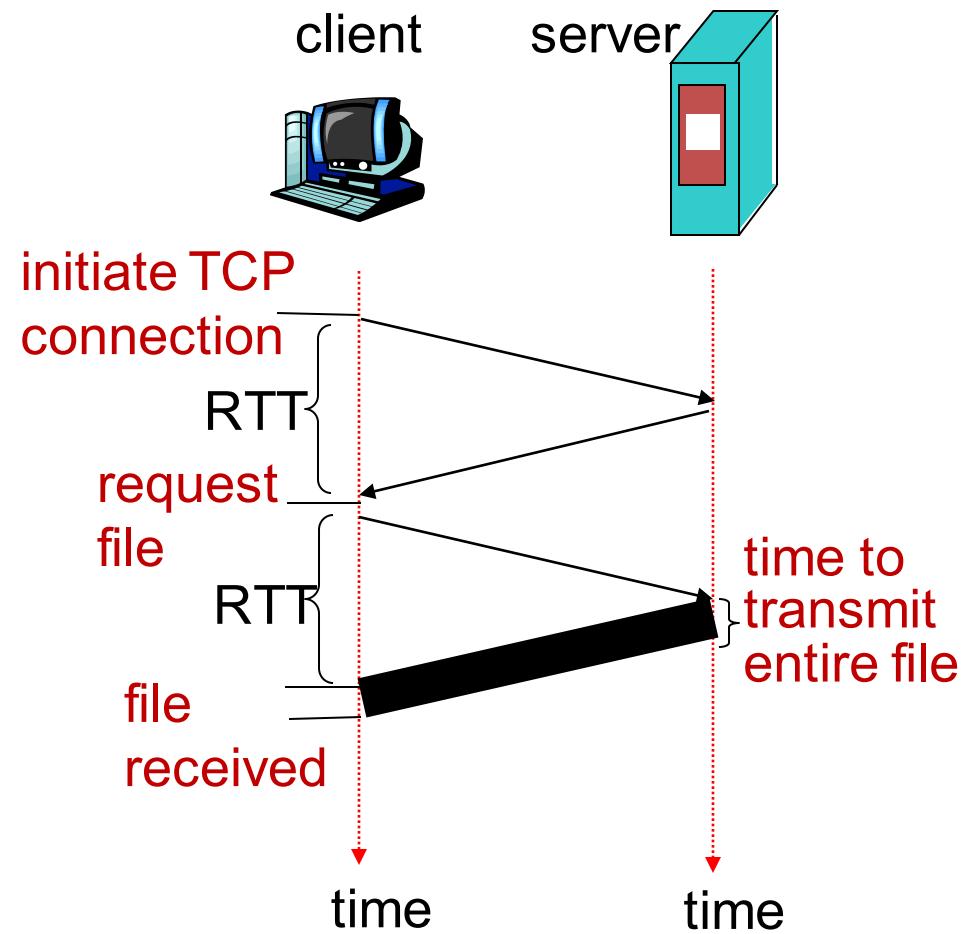
Non-persistent HTTP (II)

time

- 
4. HTTP client receives response message containing HTML file
 5. HTTP server closes TCP connection.
 6. Parses HTML file, and finds 10 referenced jpeg objects
 7. Steps 1-6 repeated for each of the 10 jpeg objects

Non-Persistent HTTP: Response Time

- Round Trip Time (RTT) – time for a small packet to travel from client to server and back
- Response time
 - one RTT to initiate TCP connection
 - one RTT for HTTP request and first few bytes of HTTP response to return
 - file transmission time
- Total response time =
$$2 \text{ RTT} + \text{file transmission time}$$



Non-Persistent HTTP – Issues

- Requires new connection per requested object
- OS overhead for each TCP connection
- Delivery delay of 2 RTTs per requested object

Persistent HTTP

- Server leaves connection open after sending response
- Subsequent HTTP messages between same client/server sent over open connection
- *Pipelining* – client sends request as soon as it encounters a referenced object
 - → as little as one RTT for all the referenced objects
- Server closes a connection if it hasn't been used for some time

Default HTTP: persistent connection with pipelining

HTTP Request Message: Example

❖ ASCII (human-readable format)

request line
(GET,
POST,
HEAD
commands)

header
lines

indicates
end of
header
lines

```
GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
\r\n
```

carriage return character
line-feed character

Persistent HTTP

Method field: GET, POST (for filling out a form), PUT (for uploading objects), DELETE, HEAD (like get, but asks only for the header, without the requested object – used for debugging).
Keep-Alive: measured in seconds.

HTTP Response Message: Example

200 OK – request succeeded, requested object later in this msg

301 Moved Permanently – requested object moved, new location specified later in this msg
(Location:)

400 Bad Request – request msg not understood by server

404 Not Found – requested document not found on this server

505 HTTP Version Not Supported

status line:

(protocol status-code status-phrase)

HTTP/1.1 200 OK\r\n

Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n

Server: Apache/2.0.52 (CentOS) \r\n

Last-Modified: Tue, 30 Oct 2007 17:00:02 GMT\r\n

Content-Length: 2652\r\n

Keep-Alive: timeout=10, max=100\r\n

Connection: Keep-Alive\r\n

Content-Type: text/html; charset=ISO-8859-1\r\n

\r\n

data data data data data ...

data, e.g.,
requested
HTML file

Connection: keep-alive or close.

Keep-Alive: max (max number of requests that will be accepted), connection will be closed if next request is not received within timeout (10 secs) time.

Last-modified: required for caching.

Content-length: number of bytes in the object (excluding header)

Content-type: HTML text (indicated in the header, not by means of file extension).

HTTP Request Methods

Method	Description
GET	Request to read a Web page
HEAD	Request to read a Web page's header
PUT	Request to store a Web page <small>(write a new page / resource)</small>
POST	Append to a named resource (e.g., a Web page)
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Reserved for future use
OPTIONS	Query certain options

POST is used to append/update a particular resource

TRACE method is for debugging. It instructs the server to send back the request

OPTIONS method provides a way for the client to query the server about its properties or those of a specific file

HTTP Error Codes

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

Cookies

- The Web is basically stateless
- Cookies to place small amount (<4Kb) of info on users computer and re-use deterministically (RFC 2109)
- Cookies have 5 fields - domain, path, content, expiry, security
- Questionable mechanism for tracking users (invisibly perhaps) and learning about user behaviour eg, undesirable content etc.

User-server Interaction: Cookies Example (I)

Susan always accesses the Internet from her (*cookie-enabled*) home PC. She visits a specific (*cookie-enabled*) e-commerce site for the first time

- When the initial HTTP request arrives at the site, the site creates:
 - unique ID
 - entry in backend database for ID
- The e-commerce site then responds to Susan's browser, including in the HTTP response
 - Set-cookie: 1234 — **ID**

User-server Interaction: Cookies Example (II)

- Susan's browser appends a line to a cookie file that it manages
 - www.e-commerce-site.com 1234
- Next time Susan request a page from that site, a cookie header line will be added to her request
 - Cookie: 1234
- The server will then perform a cookie-specific action

Advantages of Cookies

- Authorization
- Shopping carts
- Recommendations
- User session state (e.g., in Web e-mail)

Only FYI: Sessions vs Cookies

- Both introduce state into HTTP

Sessions

- Sessions information regarding visitor's interaction stored at the server side
- Just a Session ID stored at client side
- When user closes the website, the session ends
- Clusters of servers – treat as new user
- Sessions information size can be large
- E.g., count unique users to the web site, etc..

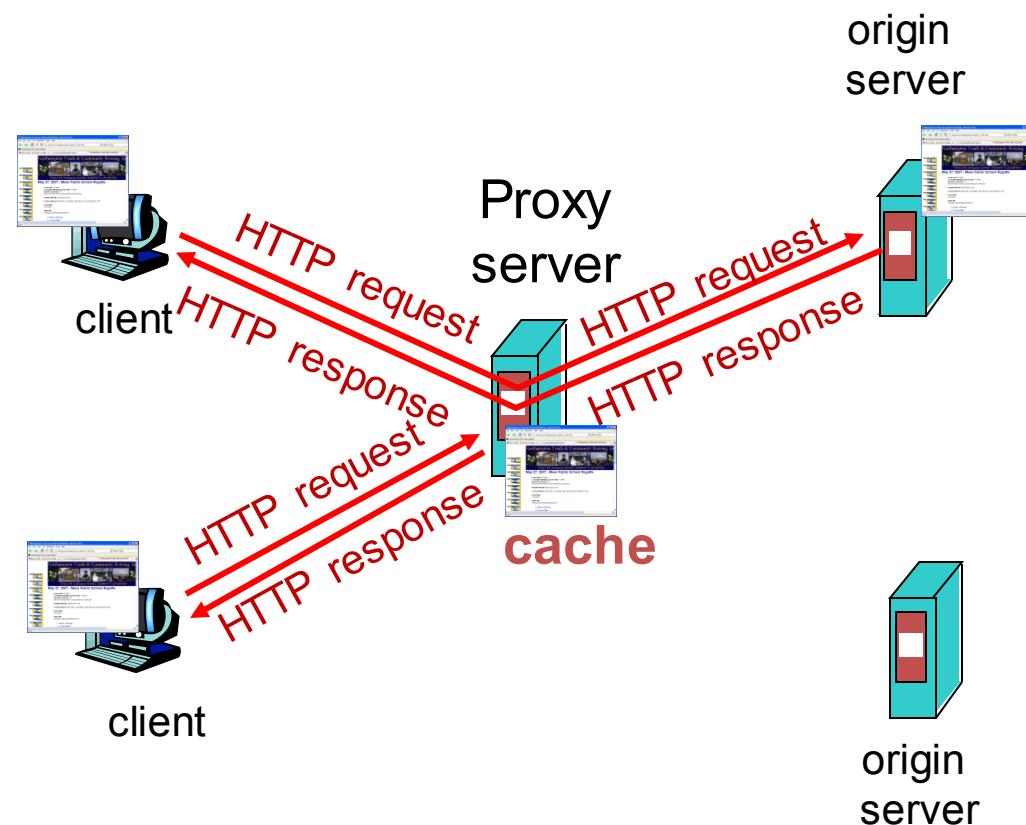
Cookies

- Cookies are transferred between server and client
- Cookie information stored at both client and server
- Use cookie ID
- Maintain client information until deleted
- Clusters of servers – same users by sending cookie info.
- Cookies information size limited
- For authentication, shopping carts, etc..

Web Caches (Proxy Server)

Goal: satisfy client request without involving origin server

- ❖ User sets browser to access Web via cache
→ browser sends all HTTP requests to cache
 - **if object in cache**, cache returns object
 - **else** cache requests object from origin server, then returns object to client



More about Web Caching

- Cache acts as both client and server
- Typically cache is installed by ISP (university, company, residential ISP)

Why Web caching?

- Reduce response time for client request
- Reduce traffic on an institution's access link

Summary

- World Wide Web
 - Persistent vs non-persistent connections
 - Describe the role of cookies
 - Web caches

Security

COMP90007

Internet Technologies

Chien Aun Chan

Outline: Network Security

- Network Security
- Cryptography
 - Basic Constituents and Relations
 - Ciphers
 - Algorithms

Application
Transport
Network
Link
Physical

Security threats

Adversary	Goal
Student	To have fun snooping on people's email
Cracker	To test out someone's security system; steal data
Sales rep	To claim to represent all of Europe, not just Andorra
Businessman	To discover a competitor's strategic marketing plan
Ex-employee	To get revenge for being fired
Accountant	To embezzle money from a company
Stockbroker	To deny a promise made to a customer by email
Con man	To steal credit card numbers for sale
Spy	To learn an enemy's military or industrial secrets
Terrorist	To steal germ warfare secrets

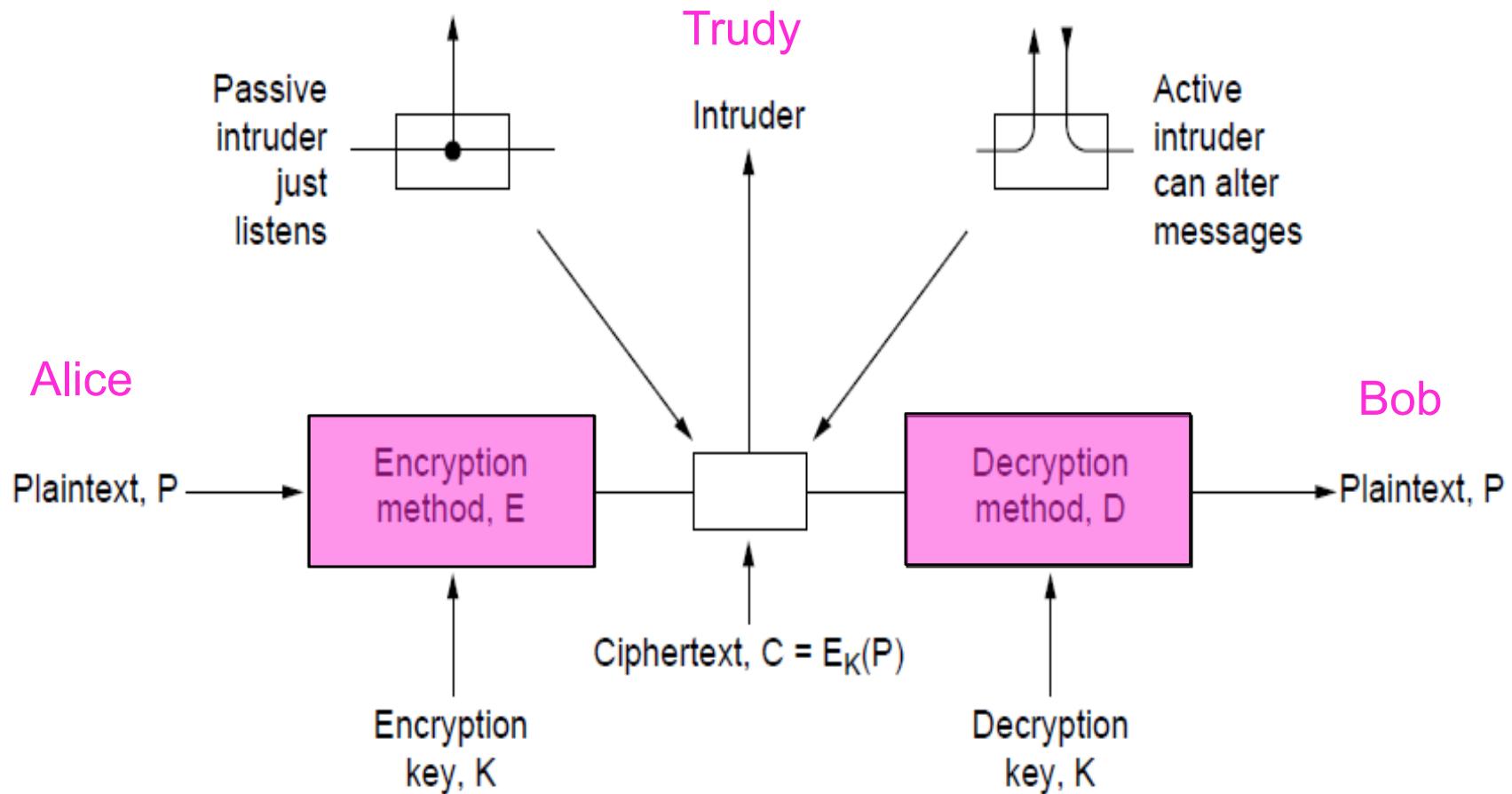
What is Security?

- Network security is a factor of 4 related concepts
 - Secrecy (Keeping information hidden from unauthorised users)
 - Authentication (Ensuring the user you are talking to has valid access to a given resource)
 - Non-repudiation (Prove a message was sent by a given user and it's contents are valid)
 - Integrity control (Ensure that a message has not been tampered with in transit)
- All of these are equally valid for traditional systems, but have different implications in a networked environment
- Aspects of security can be found at all layers of a protocol stack
- Apart from the physical layer, almost all security implementations are based on common cryptographic principles

Cryptography

- Cryptographic Constituents and Relations
- Symmetric Key Algorithms
- Assymmetric Key Algorithms
- Digital Signatures
- Public Key Management

Encryption Model



Cryptography Concepts

- Three foundational concepts
 - Plaintext
 - Keys
 - Ciphertext
- **Plaintext** messages to be encrypted can be transformed (encrypted/decrypted) by a function that is parameterized by a **key**, the output of the transformation process is **ciphertext**
- **Kerckhoff's principle:** Cryptographic Algorithms and related functions (E , D) are public; only the keys (K) are secret

Relation of Cryptographic Constituents

- $C = E_K(P)$
- $P = D_K(C)$
- $D_K(E_K(P)) = P$
- Where: C = ciphertext, P = plaintext, E = encryption, D = decryption, K = key
- In fact what we require is
- $D_{K1}(E_{K2}(P)) = P \text{ if and only if } k1=k2.$

Keys Plays an Important Role

- A key is a short string that allows the selection of one of many potential encryptions
- The key can be changed as often as required
- How many possible keys are available when using numerical strings of length:
 - 2 digits?
 - 3 digits?
 - 6 digits?
- The size of the overall key space is determined by the number of bits in the key string
- The longer the key, the more effort is required to break a given encryption

Fundamental Machinery: XOR

- An XOR is an “exclusive or” function.
- A XOR B means A or B, but not both
- XOR is commonly used in cryptography as a comparison mechanism in multiphase encryption and decryption

A	B	A XOR B
F	F	F
F	T	T
T	F	T
T	T	F

Truth values	Binary Equivalents
T	1
F	0

Types of Ciphers

■ Substitution cipher

- Each letter or group of letters is replaced systematically by other letters or groups of letters (breakable with knowledge of the replacement system)

■ Transposition cipher

- All letters are re-ordered without disguising them (breakable with knowledge of re-ordering system)

■ One-time pad

- Uses a random bit string as the key, convert the plaintext into a bit string, then XOR the two strings bit by bit (unbreakable because any other plaintext has the same likelihood as the original message)

Substitution cipher

- Substitution ciphers replace each group of letters in the message with another group of letters based on a key with an intention to disguise the message.

plaintext: a b c d e f g h i j k l m n o p q r s t u v w x y z
ciphertext: Q W E R T Y U I O P A S D F G H J K L Z X C V B N M

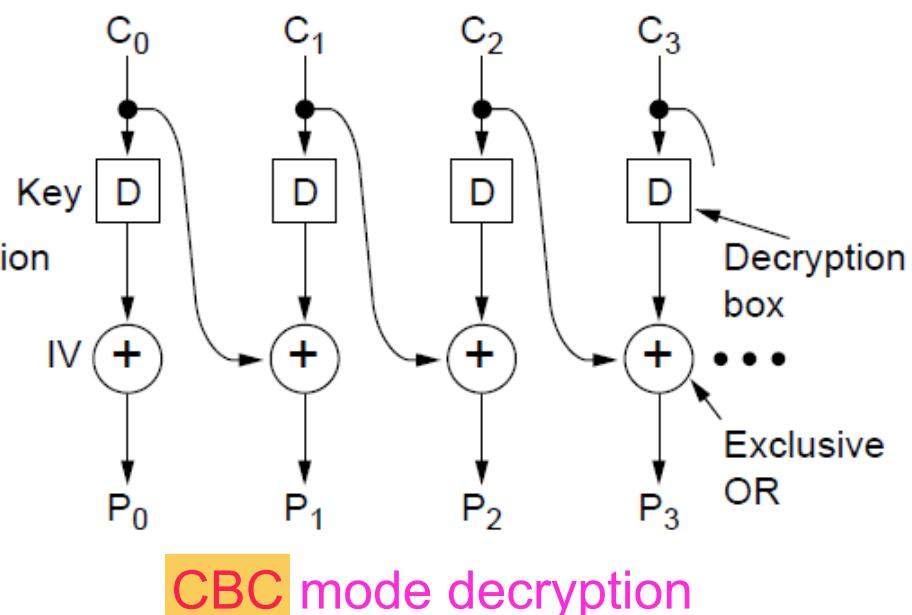
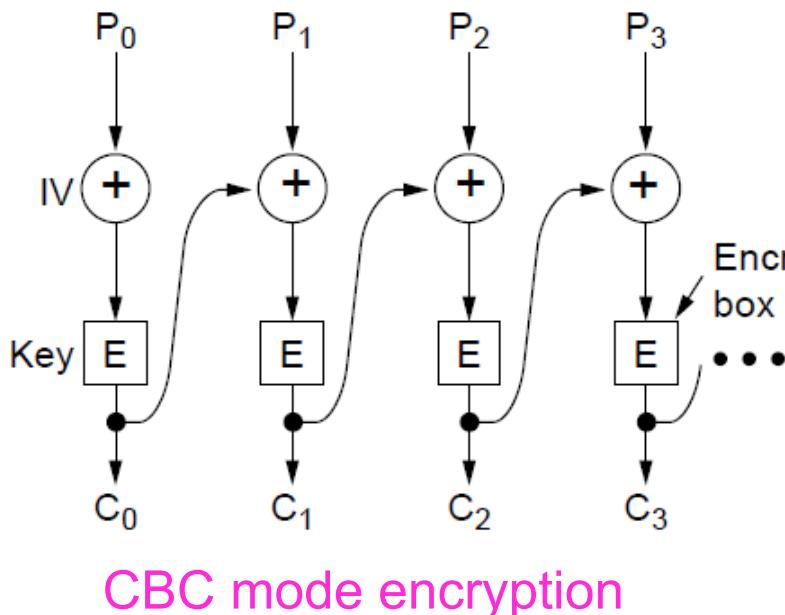
Symmetric Key Algorithms

- A symmetric key algorithm uses the same key for both encryption and decryption
- Symmetric key algorithms can use permutation, substitution and a combination of both to encrypt and decrypt
- 2 Symmetric Key Algorithms
 - Data Encryption Standard (DES)
 - Uses 64 bit blocks and 56 bit keys
 - 2^{56} key space
 - Triple DES has a 3×2^{56} key space
 - Advanced Encryption Standard (AES)
 - Uses 128 bit blocks and 128 bit keys (others available)
 - 2^{128} key space
- Electronic Code Book
 - Plain text → block cipher encryption → Ciphertext

Key ↑

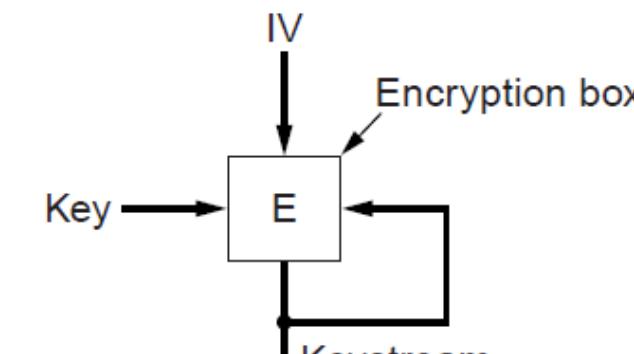
Cipher Modes: Block Chaining

- In block chaining mode, each plaintext block is XOR'ed with the previous ciphertext block before being encrypted
- (a) encryption, (b) decryption



Cipher Modes: Stream Ciphers

- In stream cipher mode, recursive sequential block encryption is used as a one-time pad, and XOR'ed with plaintext to generate ciphertext
- (a) encryption, (b) decryption



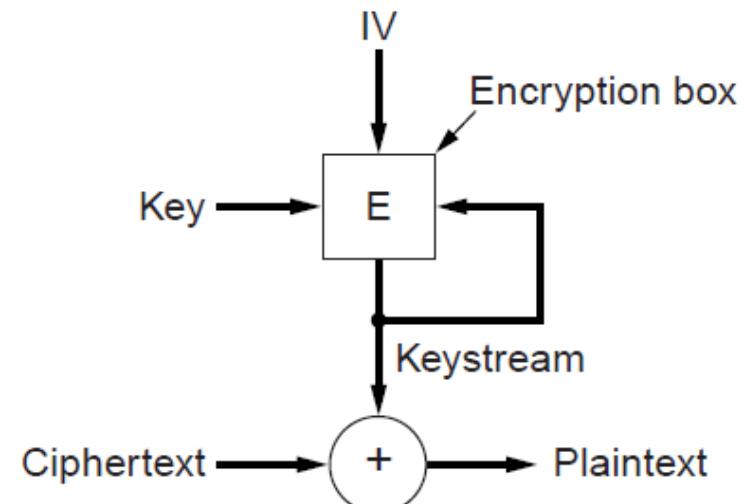
Note:

$$C_1 = A \text{ XOR } K$$

$$C_2 = B \text{ XOR } K$$

Encryption

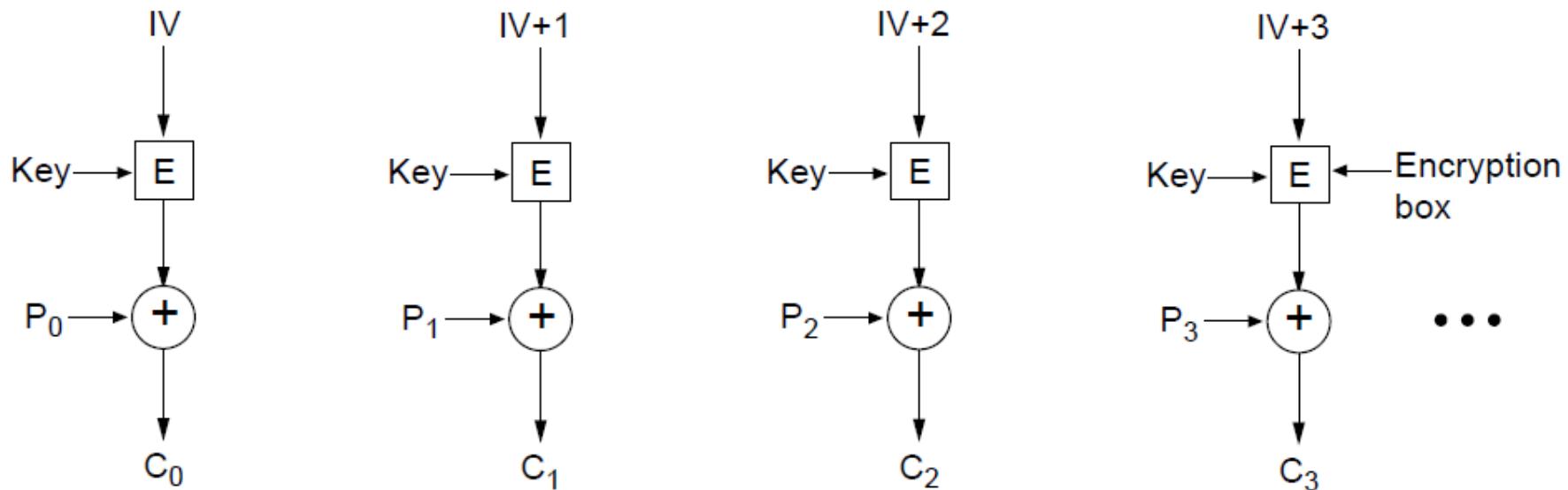
$$C_1 \text{ XOR } C_2 = A \text{ XOR } B$$



Decryption

Cipher Modes: Counter Mode

- In counter mode, plaintext is not directly encrypted, but an initialisation parameter plus an arbitrary constant is encrypted, and the resulting ciphertext is XOR'ed with plaintext to generate new ciphertext



Encryption above; repeat the operation to decrypt

Other Symmetric Key Algorithms

Cipher	Author	Key length	Comments
Blowfish	Bruce Schneier	1–448 bits	Old and slow
DES	IBM	56 bits	Too weak to use now
IDEA	Massey and Xuejia	128 bits	Good, but patented
RC4	Ronald Rivest	1–2048 bits	Caution: some keys are weak
RC5	Ronald Rivest	128–256 bits	Good, but patented
Rijndael	Daemen and Rijmen	128–256 bits	Best choice
Serpent	Anderson, Biham, Knudsen	128–256 bits	Very strong
Triple DES	IBM	168 bits	Second best choice
Twofish	Bruce Schneier	128–256 bits	Very strong; widely used

Public Key Algorithms

- Diffe & Hellman (1976) proposed a radically different model to symmetric key algorithms - asymmetric key algorithms, where the key used to encrypt and the key used to decrypt are different, and not derivable from each other
- Diffe-Hellman 2 key system
 - Key 1: public key, usable by anyone to encrypt messages to the owner of the key
 - Key 2: private key, required to decrypt the message (and held only by the owner of the key)

RSA, An Asymmetric Key Algorithm

- RSA - Rivest, Shamir, Adleman (1978)
- Very robust algorithm, but requires keys of length 1024 bits
- Key generation:
 - Choose two large primes, p and q
 - Compute $n = p \times q$ and $z = (p - 1) \times (q - 1)$.
 - Choose d to be relatively prime to z
 - Find e to satisfy the *congruence relation* of $e \times d \equiv 1 \pmod{z}$
 - Which means satisfy $(d \times e) \bmod (z) = 1$
 - Public key is (e, n) , and private key is (d, n)
- Encryption (of k bit message, for numbers up to n):
 - Cipher = Plain^e \pmod{n}
- Decryption:
 - Plain = Cipher^d \pmod{n}

RSA Security

- RSA's security is based on the difficulty involved in factoring large numbers - approx 10^{25} years to factor a 500 digit number using a brute force approach
- RSA is too slow for encrypting/decrypting large volumes of data, but is widely used for **secure key distribution**

RSA Example

- Let $p=3, q=11 \rightarrow n=33, z=20 \rightarrow d=7, e=3$

Plaintext (P)		Ciphertext (C)		After decryption	
Symbolic	Numeric	$\underline{P^3}$	$\underline{P^3 \pmod{33}}$	$\underline{C^7}$	$\underline{C^7 \pmod{33}}$
S	19	6859	28	13492928512	19
U	21	9261	21	1801088541	21
Z	26	17576	20	1280000000	26
A	01	1	1	1	01
N	14	2744	5	78125	14
N	14	2744	5	78125	14
E	05	125	26	8031810176	05

Sender's computation

Receiver's computation

Encryption: $C = P^3 \pmod{33}$

Decryption: $P = C^7 \pmod{33}$

Using Cryptography: Digital Signatures

- Cryptographic approaches can be used to ensure **authenticity** and allow for **non-repudiation**
- Requirements
 - Receiver can verify the claimed identity of the sender
 - Sender cannot repudiate contents of the message
 - Receiver cannot have derived the message themselves
- Three approaches
 - Using symmetric keys via an intermediary to ensure non-repudiation
 - Using public keys as individuals
 - Using message digests

Message Digests

- Basic concept of a message digest is to use a one-way hash function to take an arbitrary length of plaintext and compute a fixed-length bit string
- A message digest (MD) has four important properties:
 - 1 Given P, it is easy to compute $MD(P)$
 - 2 Given $MD(P)$ it is effectively impossible to find P
 - 3 Given P, no one can find P' such that $MD(P') = MD(P)$
 - 4 A change in even a single bit of input produces a very different output
- Given 3), the hashing function should be at least 128 bits long
- Given 4), the hashing function should scramble the bits very thoroughly
- Computing a message digest from plaintext is much faster than encrypting plaintext - so digests can be used to speedup the derivation of a digital signature

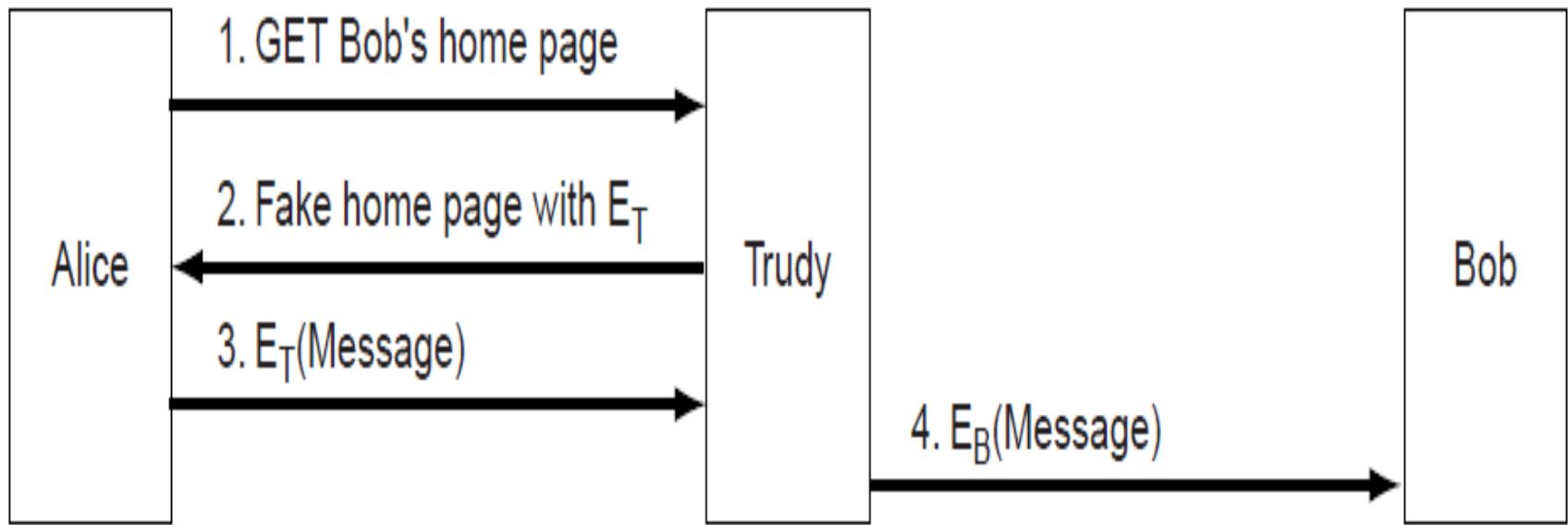
Message Digest Algorithms

- MD5
- Secure Hash Algorithm SHA-1
- Comparative output
 - File contents:\this is a test"
 - MD5: e19c1283c925b3206685522acfe3e6
 - SHA-1:
6476df3aac780622368173fe6e768a2edc3932c8

Public Key Management

- There is a need for specific PK infrastructure primarily to avoid compromising the security of PK's during the distribution process. We need a trusted way to distribute public keys.
- **Certification Authority (CA)**
 - A trusted intermediary who uses non-electronic identification to identify users prior to certifying keys and certificates
- **X.509**
 - An international standard for certificate expression
- **PKI (Public Key Infrastructure)**
 - Hierarchically structured certificate authorities allow for the establishment of a chain of trust or certification path

Management of Keys: Man in the middle



Trudy replaces E_B with E_T and acts as a “man in the middle”

Outline

- Authentication
- Firewalls
- IPSec
- Virtual Private Network
- Wireless Security

Communication Network Security

- Cryptography provides the foundation on which network and application security operates

Authentication Protocols

- Authentication is a primary tenet of network security
- However, authentication itself needs to be secure also
- There are a number of common methods for secure authentication, but all subscribe to a single principle: minimise the use of permanent/private keys in establishment of secure connections (the less packets are exchanged using permanent/private keys, the less exposure to potential attackers)
- Four methods in common use:
 - Shared keys
 - Key distribution
 - Kerberos
 - Public keys

Authentication Protocols

■ Four methods in common use:

❑ Shared keys

- one party sends a random number to the other party, who transforms it and sends the result back - essentially a challenge and response protocol

❑ Key distribution

- a trusted intermediary is used to facilitate the authentication
- Users each share a key with a central **key distribution centre**, and authenticate to the KDC directly
- The KDC then acts as a relay between the two parties

❑ Kerberos

- a multi-component system is required
 - ❑ Authentication Server
 - ❑ Ticket Granting Server
 - ❑ Recipient
- Authentication is managed centrally, and then party to party communication is facilitated by single use cryptographic tickets

❑ Public keys

- Public Key Infrastructure with directory
- Users enquire and get public keys of recipients from PKI for encryption containing sender identity

IPSec

- IPSec represents one view of how to embed security in the protocol stack - at the network level
- In the IPSec model, encryption is compulsory, but for graceful failover, a null encryption algorithm can be used between points which are not cryptographically inclined
- The major IPSec framework features are secrecy, data integrity, and replay attack protection
- The IPSec framework allows multiple algorithms and multiple levels of granularity
- IPSec is connection-oriented, with connections being called SA's (security associations)
 - SA includes attributes: cryptographic algo., cipher mode, traffic encryption key, etc..

IPSec Implementation

- IPSec has two main implementation components
 - New headers being added to packets in transit
 - ISAKMP key management
- IPSec has 2 modes
 - Transport mode - uses header insertion
 - Authentication Header (AH) - provides no data encryption but provides integrity checking using Hashed Message Authentication Code (HMAC) – for signature computation (hash over packet + shared key)
 - Tunnel mode - uses packet encapsulation
 - Encapsulating Security Payload (ESP) - provides an encryption layer as well as HMAC based integrity checking
 - Useful when a bundle of TCP connections is aggregated and handled as one encrypted stream – prevent intruder to see how many packets have been sent between two parties.

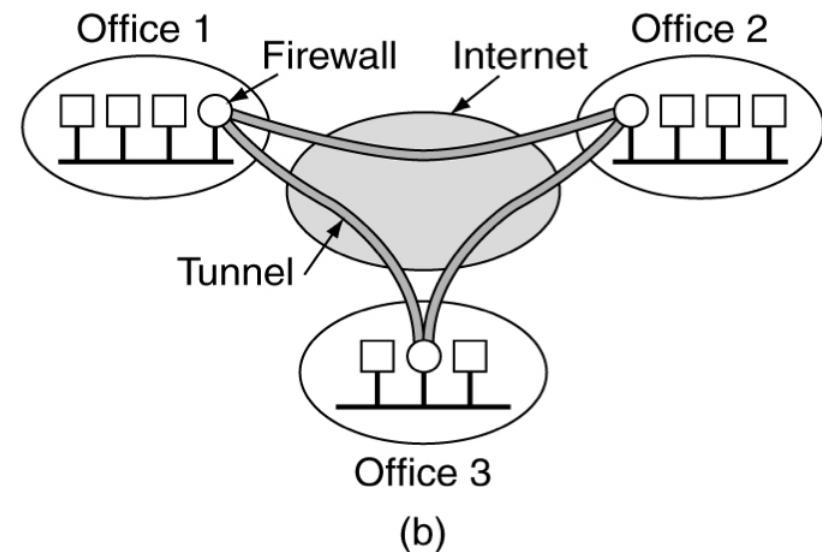
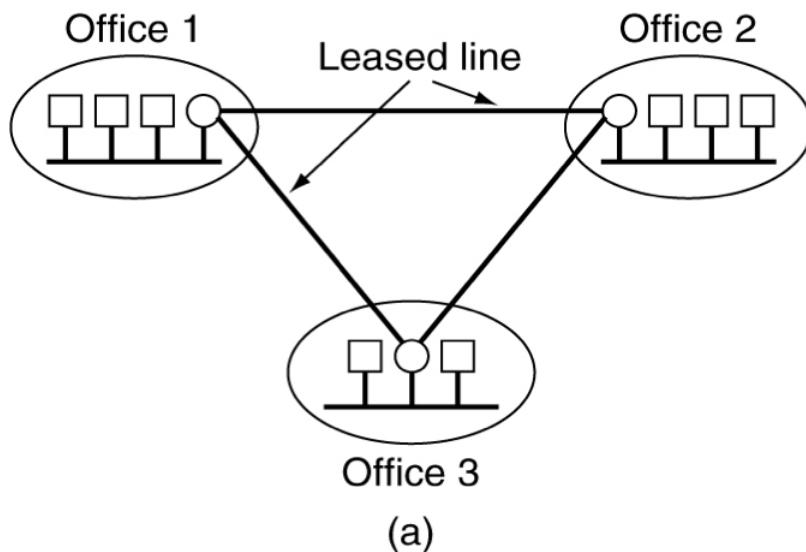
Virtual Private Networks

- Unlike a physical network based on leased lines between locations for which secure transit is required, a Virtual Private Network (VPN) is a virtual layer on top of an IP network which provides a secure end-to-end connection over public infrastructure
- A common VPN implementation model is to use a firewall at each end of a connection, use the firewalls to setup a SA to create an IPSec tunnel between the two end points, and then selectively route traffic for the specific destination via the encrypted connection
- Using such models, the security infrastructure is transparent to end users as the firewalls are responsible for the maintenance of the VPN

An SA is a simplex connection between two endpoints and has a security identifier associated with it

VPNs Illustrated

- (a) a leased line network
- (b) a virtual private network



IPSec VPNs

- Components of an IPSec VPN
 - Content encryption algorithm eg DES, 3DEs, AES
 - Collision-free hashing algorithm eg MD5, SHA-1
 - Diffie-Hellman key exchange protocol
 - Public Key Infrastructure for Certification
- IPSec VPN Architecture
 - **Authentication Header (AH)** added to an IP Packet to provide data origin authentication and data integrity checking
 - An **Encapsulating Security Payload (ESP)** for IP provides encryption, data origin authentication and data integrity checking
- **Internet Security Association and Key Management Protocol (ISAKMP) and Internet Key Exchange (IKE)** allow VPN devices to
 - securely negotiate and manage IPSec security associations (SAs)
 - to generate, exchange and manage the keys that are used by the cryptographic algorithms employed by IPSec

Firewalls

- While IPSec ensures security in transit, a **firewall** ensures security at the network perimeter
- Firewalls are positioned at the network boundary, and provide a controlled series of route between the internal and external networks
- Three characteristics of firewalls
- All **inbound and outbound** traffic must transit the firewall
- Only **authorised traffic** must pass through the firewall
- Firewalls should be **immune to penetration** themselves

Firewall Scope

- Single choke point for range of functions
 - Deny access to unauthorised users either inbound or outbound
- Monitoring location for security related events
- Platform for non-security functions
 - NAT
 - Usage logging and auditing
- Platform for extended security functions
 - IPSEC
 - VPNs
 - Anti-Virus

Firewall Constraints

- No protection against threats originating via bypass networks
 - Direct dialin systems
 - Direct dialout systems
- No protection against internal threats
 - Network segmentation and access control may alleviate internal risks
- No protection against application payload threats
 - Viruses, Trojans etc spread as application payloads eg email attachments

Wireless Security Context

- Wired networks are relatively easy to secure because they require physical access to intercept traffic
- Wireless networks are more difficult to secure because of omnidirectional signal propagation
- Additionally by default most wireless network equipment operates in an insecure and promiscuous manner
- 802.11 has a native secure protocol, **Wired Equivalency Protocol (WEP)**, which is a 40-bit encryption based on RC4 algorithm

Wireless Security Issues

- WEP users from two inherent insecurities
 - 40 bit encryption is breakable with low-moderate computational resources
 - RC4 re-uses keys, so capturing a small volume of encrypted traffic will guarantee key identification
 - An inherent weakness in WEP is that it uses only a 24 bit range to populate initialisation values for RC4, so after 2^{24} packets, the sequence cycles being to repeat
 - A number of attacks against RC4 have demonstrated that the key can be derived from the encrypted stream given sufficient sample data
- Given these constraints, how can wireless networks be secured ?

Securing Wireless

- **MAC Address Filtering**
 - Only allow specified MAC interfaces to establish connections
- **Non-Broadcast SSID**
 - Prevent discovery of Community String by eavesdroppers
- **Additional encryption (128bit WEP)**
 - Increased security through longer key lengths
- **WPA (WiFi Protected Access 2)**
- **Multilayered security**
 - All of the above plus a VPN over wireless and regimented user authentication

Summary - Security

- **Cryptography**
 - Contrast the strengths and weaknesses of different cipher modes
 - Explain the use of message digests
- **Authentication**
 - Describe the basic operation of authentication using public key cryptography
- **Network Security**
 - Describe the key functions that need to be provided by a firewall
- **Summarise some of the challenges for wireless network security**

Week 12 – Review

COMP90007
Internet Technologies

Chien Aun Chan

Final Exam Details

- Length: 3 hours in length, with 15 minutes reading time
- The exam will **NOT** be an open book exam
- I will be in attendance at the exam during reading time to answer any questions that may arise

Final Exam Structure

- Exam consists of 20 multiple choice questions (MCQs) and 10 short answer questions. You should attempt to answer all questions.
- MCQs → 1 mark each
- Short answer questions → 4 marks each
 - Questions should be able to be answered in a maximum of 2 short paragraphs (usually less).

Course Review

- “The objective for this subject is for students to develop an understanding of foundational network technologies and applications, and be able to demonstrate proficiency in internetworking.”
- Major Goals
 - Develop an understanding of network technologies and applications
 - Be able to demonstrate proficiency in internetworking and its management
 - Be able to undertake problem identification, formulation and solution

Scope of Final Exam

- Questions can be originated from any topics covered in this subject:
 - Lecture slides
 - Workshops (i.e., labs and tutorial questions)
 - Assignments (Assignment 1, Assignment 2, and Network Analysis Assignment)
 - Relevant sections in the textbook

Introduction - summary

- Networks can provide connection-oriented or connectionless services
- The Internet is the most widely known network, it is a network of networks
- Standardisation of both hardware and software is important to ensure interoperability between different implementations

Computer Networks

- Different applications have different network requirements
- How many networks do you know?
- Can you tell the main characteristics of these networks?
- What kind of applications can run well/cannot run well in these networks?

Protocol Hierarchies

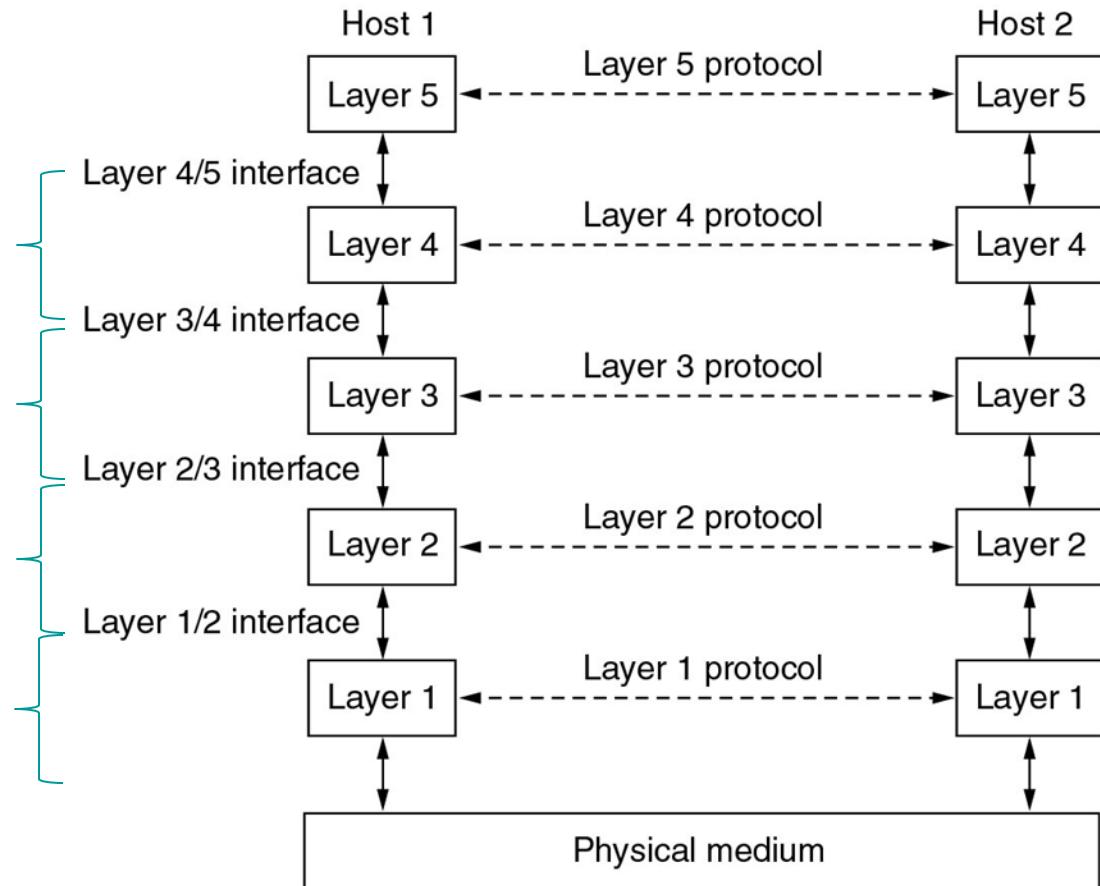
■ Protocol, layers, and services

Multiplexing/demultiplexing applications

Connecting different networks (internetworking)

Framing, reliability and flow control (direct conn.)

Different cables, wireless signalling, digital to analogue



Physical Layer -summary

- Physical layer is the basis of all network technologies - natural limits on different types of channels and thus affect their bandwidth
- Message Latency (delay)
 - Putting the message on the line (Transmission Delay)
 - Travel time of a bit from one end to the other end of channel (Propagation delay)
- Transmission media can be guided (twisted pair, coaxial cable, fibre optics) or unguided (radio, infrared, lasers, microwaves)
- The PSTN network is a key element in many networks as it provides the local loop between the consumer and the network service provider

Physical Layer

- Data Communications Theory
 - Nyquist theorem
 - Shannon theorem
 - Characterise different guided media
 - Twisted pair vs coaxial cable vs fibre optics
- Recognise different modulation types
- Wireless Data Transmission
 - Explain differences between wireless transmission types
 - Choose the appropriate type of satellite for an application

Data Link Layer - summary

- The data link layer converts a raw bit stream into a series of frames
- Various framing methods are used
- Protocols at the data link layer can provide error detection and control, as well as flow control - the sliding window mechanism is used to integrate these two concepts
- Sliding window protocols can be categorised based on the function which determines the size of the sender's and receiver's windows
- Many networks use a bit oriented protocol at the data link level – all use flag bytes to delimit frames and bit stuffing to prevent flag bytes occurring in the data

Data Link Layer

- Data Link Layer
 - Contrast types of services provided
 - Apply different framing methods
 - Character count, bit and byte stuffing
 - Explain methods for error detection / correction
- Data Link Layer Protocols
 - Characterise different protocols
 - Stop-&-wait, sliding window: go-back-N, selective repeat
 - Calculate efficiency of stop-and-wait

MAC Sublayer - summary

- In networks which have a single communication channel available, the design of the allocation mechanism for this channel is important, and many methods have been developed to achieve this
- The simplest allocation schemes are FDM and TDM - both best when traffic is continuous and number of nodes is small
- In larger systems, protocols derived from ALOHA are better choices
- Carrier sensing can be used by nodes to determine the state of the channel in advance of transmission
- Some protocols reduce or eliminate contention - binary countdown, tree walk
- Wireless LANs have unique problems and solutions for transmission management - MACA, MACAW
- Ethernet (dominant form of local area networking) is available in speeds from 10Mbps to 1Gbps using mostly twisted pair media
- Wireless LANs are becoming common, mostly using 802.11

MAC Sublayer

- MAC Sub-layer
 - Compare different CSMA schemes
 - Understand collision free protocols
 - Understand why wireless is different to wired protocols
 - Explain key features of Ethernet
 - Evaluate factors affecting Ethernet performance

Network Layer - summary

- The network layer provides services to the transport layer in either virtual circuit or datagram modes
- Main purpose of network layer is to route packets from source to destination
- Many routing algorithms are used in modern networks - static vs dynamic algorithms,
- Other routing variants - broadcast, multicast, hierarchical routing
- Congestion can occur in the network layer, necessitating mechanisms to resolve congestion including retransmission policies, flow control, load shedding, choke packets
- The Internet has a variety of protocols available at the network layer - IP as the primary data transport protocol, but also ICMP, ARP, RARP

Network Layer

- Internetworking – issues and solutions
- Routing
 - Differences between VC and Datagram subnetworks
 - Understand Distance Vector and Link State Routing protocols
 - Dijkstra's algorithm
- Internet Protocol
 - Explain principles of Internet design
 - Analyse structure of IP addresses – IP addresses allocation
 - Understand IPv4 frame structure
 - Explain roles of different Internet Control Protocols
- Quality of Service
 - Summarise effects on congestion of policies at different layers
 - Characterise QoS requirements of different applications

Transport Layer - summary

- The transport layer provides reliable, end to end connection oriented byte streams
- Primitives allow the establishment, use and release of connections
- Berkeley sockets are a common method for utilising the transport layer interface
- Transport protocols must also perform connection management over unreliable networks - mechanisms for handling lost or duplicate packets are important
- Three way handshakes are a common connection establishment mechanism
- The Internet has 2 main transport protocols - TCP and UDP

Transport Layer – summary (cont.)

- UDP is a connectionless protocol that mainly wraps IP packets with additional features of multiplexing using ports in addition to an IP address
- TCP is a reliable byte stream protocol, which allows segments to be disassembled and reassembled during transit
- Network performance is dominated by the need to reduce protocol and TPDU overhead - increasing problems at higher speeds – protocols should be designed to minimise the impact of increased traffic

Transport Layer

- Transport Layer Services
 - Discuss challenges in providing reliable services over unreliable network layer
 - Explain solutions for connection establishment and release
- Internet Transport Protocols
 - Characterise differences between TCP and UDP
 - Explain TCP congestion control
- Slow start procedure (to quickly increase the speed) and additive increase
 - Be able to explain step by step from the start

Core Applications - summary

- Foundational to the Internet architecture is the hierarchical naming scheme DNS
- DNS is implemented as a distributed database system which can be queried by clients to determine mappings between IP addresses, host names, and various other types of records
- Email is a dominant Internet application consisting of two components - the user agent and the mail transfer agent
- Email sent by SMTP, and alternatively received by POP3 or IMAP

Core Applications - summary (cont.)

- Email has been extended to allow the transmission of non-text objects through the use of MIME
- The WWW is actually an application that uses the Internet as a foundation
- Browsers are used to navigate the WWW, with the assistance of browser plugins and helper applications for rich content handling Information sources on the WWW can include both static and dynamic document types
- Web standards are continuously evolving

Core Applications

■ DNS

- Describe operation of DNS lookup

■ HTTP & WWW

- Types of connections
- Describe the role of cookies
- Describe the role of Web Cache

Emerging Applications - summary

- Multimedia on the web has become a feature of the Internet in the last 10 years
- Multimedia networks are characterised by higher bandwidth requirements and the need for Quality of Service in order to be accepted by end users
- Online audio requires compression in order to achieve efficient transmission
- Data networks are increasingly being used to carry voice traffic and this new traffic brings different perspectives to network design
- The dominant VOIP technologies are H.323 and SIP, which provide a number of core functions, with specialisation in different areas, both derive from different traditions of standards development
- Online video also requires compression in order to achieve efficient transmission

Emerging Applications

- Multimedia networks
 - Describe techniques for jitter management
 - How to manage media buffer
 - Handling errors in streaming stored media
 - Compression is necessary
- VoIP
 - Benefits of VOIP
 - Contrast H.323 and SIP
 - Explain steps in SIP call establishment

Security - summary

- Network security is an important issue to consider given our increased dependence on network transmission of data
- Cryptography is a method that can be used to protect confidential data
- Ciphers are used to convert plaintext into cipher text - many different types are available
- Cryptographic algorithms can be divided into symmetric and asymmetric types
- Public key algorithms are architecturally distinguished by the use of a widely distributed key for encryption, and a private key for decryption
- Digital signatures can be constructed to ensure the reliability and non-repudiation of data - message digests, digital certificates

Security - summary (cont.)

- Cryptographic tools can be used to secure network traffic - IPSec is the dominant model and is widely used to constructed encrypted streams between sources and destinations
- Firewalls are an important component of network security architectures, serving not only as a border between public and private networks but also as the termination points for virtual private networks
- Network security is a pervasive issue - existing at all layers of a network architecture

Security

- Cryptography
 - Contrast the strengths and weaknesses of different cipher modes
 - Understand the key issues of each cipher mode
 - Block chaining, stream ciphers, counter mode,
 - Explain the use of message digests
- Authentication
 - Understand basic concepts of authentication protocols
- Network Security
 - Key elements of IPSec VPNs
 - Summarise some of the challenges for wireless network security

Review

- “Generic skills” to take away from this course:
 - be able to analyse the relationship between different components of computer networks;
 - be able to conceptually and practically differentiate the various layers in internetwork architectures;
 - be able to conduct research into emerging networking technologies;
 - be able to apply network security and network management concepts in today’s networked environments

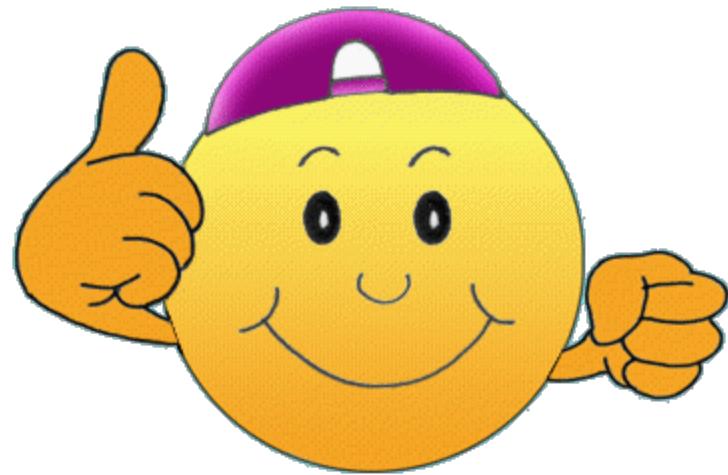
Exam Preparation: Practice Exam

- Practice Exam is available on LMS.
- Questions are indicative of the types of questions on this year's exam
- Suggest students take the practice exam as an exam

Exam Revision

- Identify the key concepts from each week's readings, lectures, tutorials and assignments
- Study and understand any of the lectures, tutorials, assignments and relevant sections from the textbook - examinable content may originate in any of them
- Look for recurring themes
 - Consider how key concepts from different topics could be related
 - E.g., jitter management for streamed media
- Try some questions from the end of chapter question sets, but don't spend all your time on them

GO FOR IT !



GOOD LUCK !

Summary

- Computer network
- Simple client-server model
- Differentiating factors of networks
 - Transmission technology types
 - Scale
- Protocols, layers, services, & interfaces
- Protocol hierarchies
- Design issues of layers
 - E.g., connection-oriented, connectionless
 - Impact on reliability and quality of the service
- OSI vs TCP/IP