CSMC 412

Computer Networks

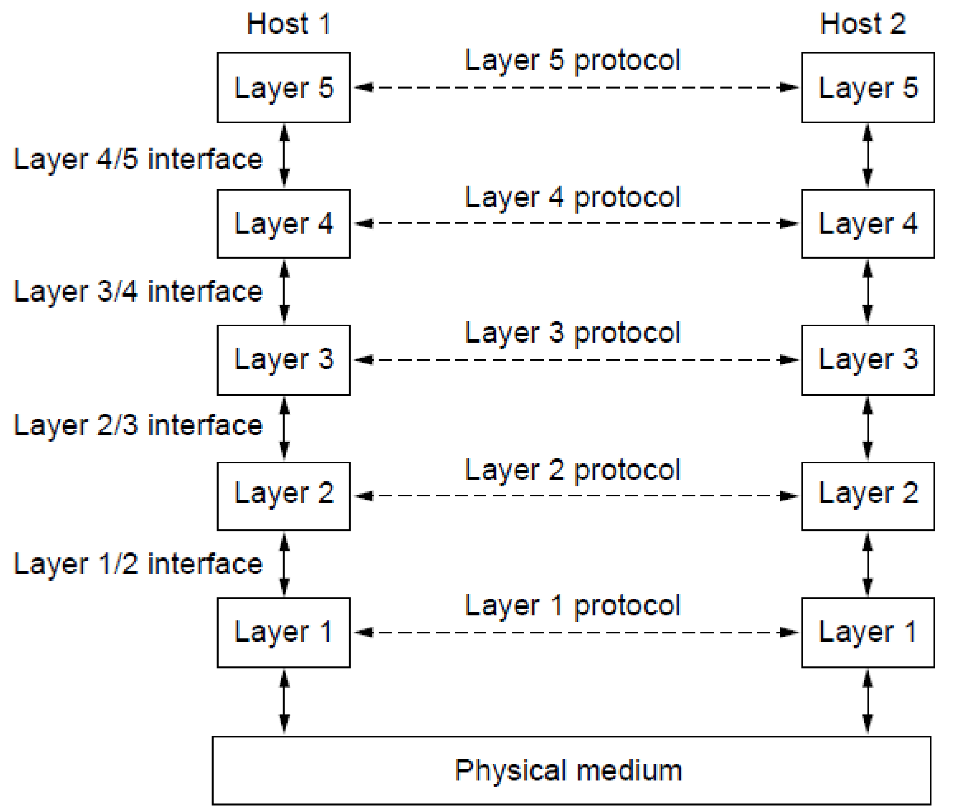
Prof. Ashok K Agrawala

Set 2

Protocol Layers (1)

Protocol layering is the main structuring method used to divide up network functionality.

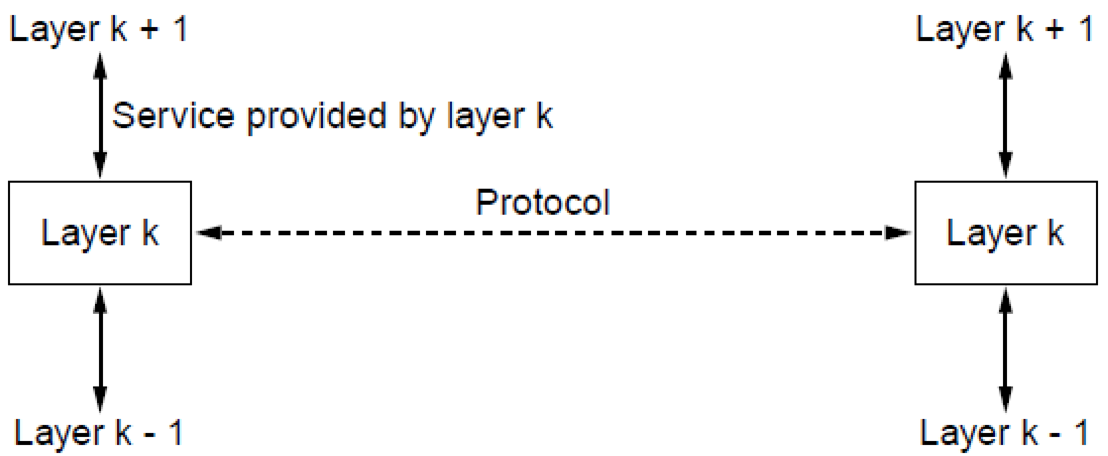
- Each protocol instance talks virtually to its <u>peer</u>
- Each layer communicates only by using the one below
- Lower layer <u>services</u> are accessed by an <u>interface</u>
- At bottom, messages are carried by the medium



Relationship of Services to Protocols

Recap:

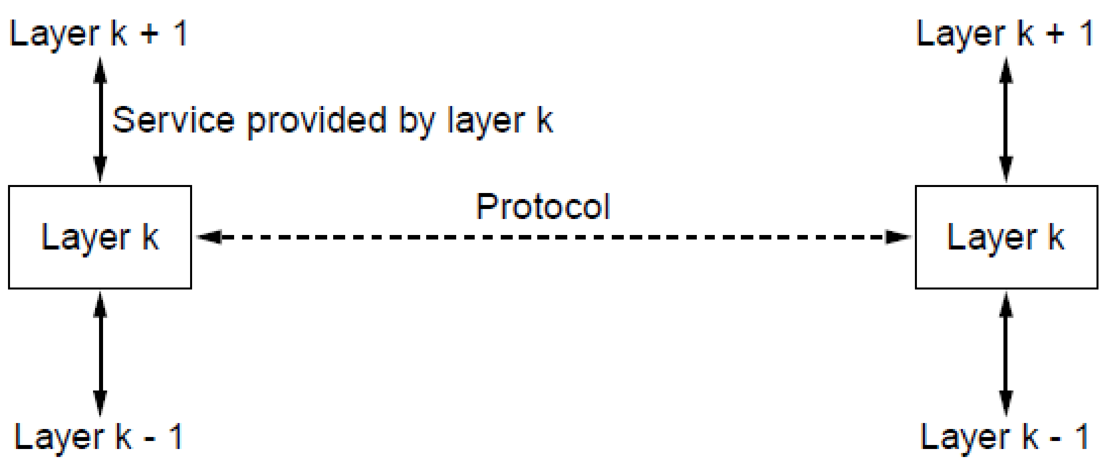
- A layer provides a service to the one above [vertical]
- A layer talks to its peer using a protocol [horizontal]



Relationship of Services to Protocols

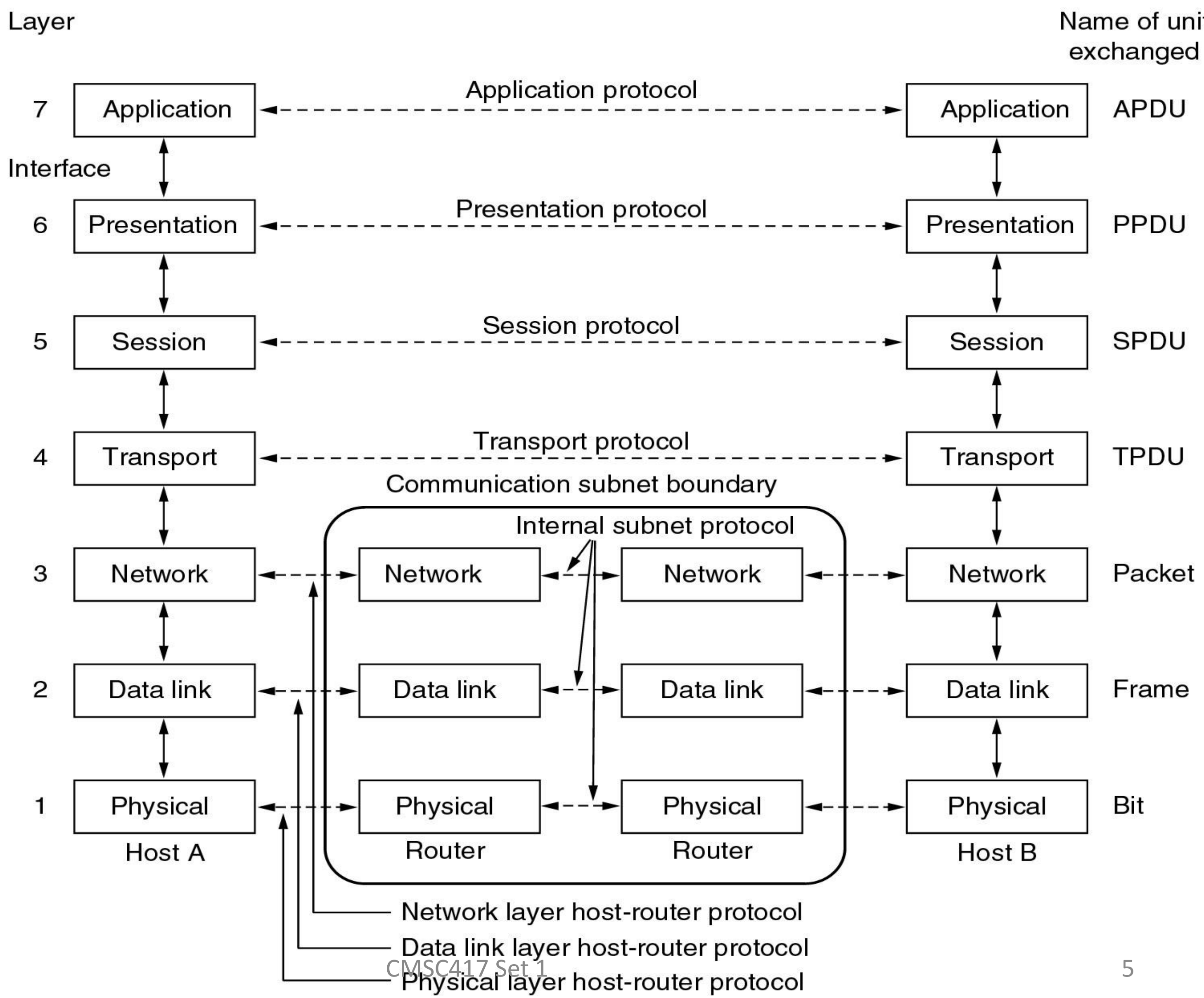
Recap:

- A layer provides a <u>service</u> to the one above [vertical]
- A layer talks to its peer using a <u>protocol</u> [horizontal]



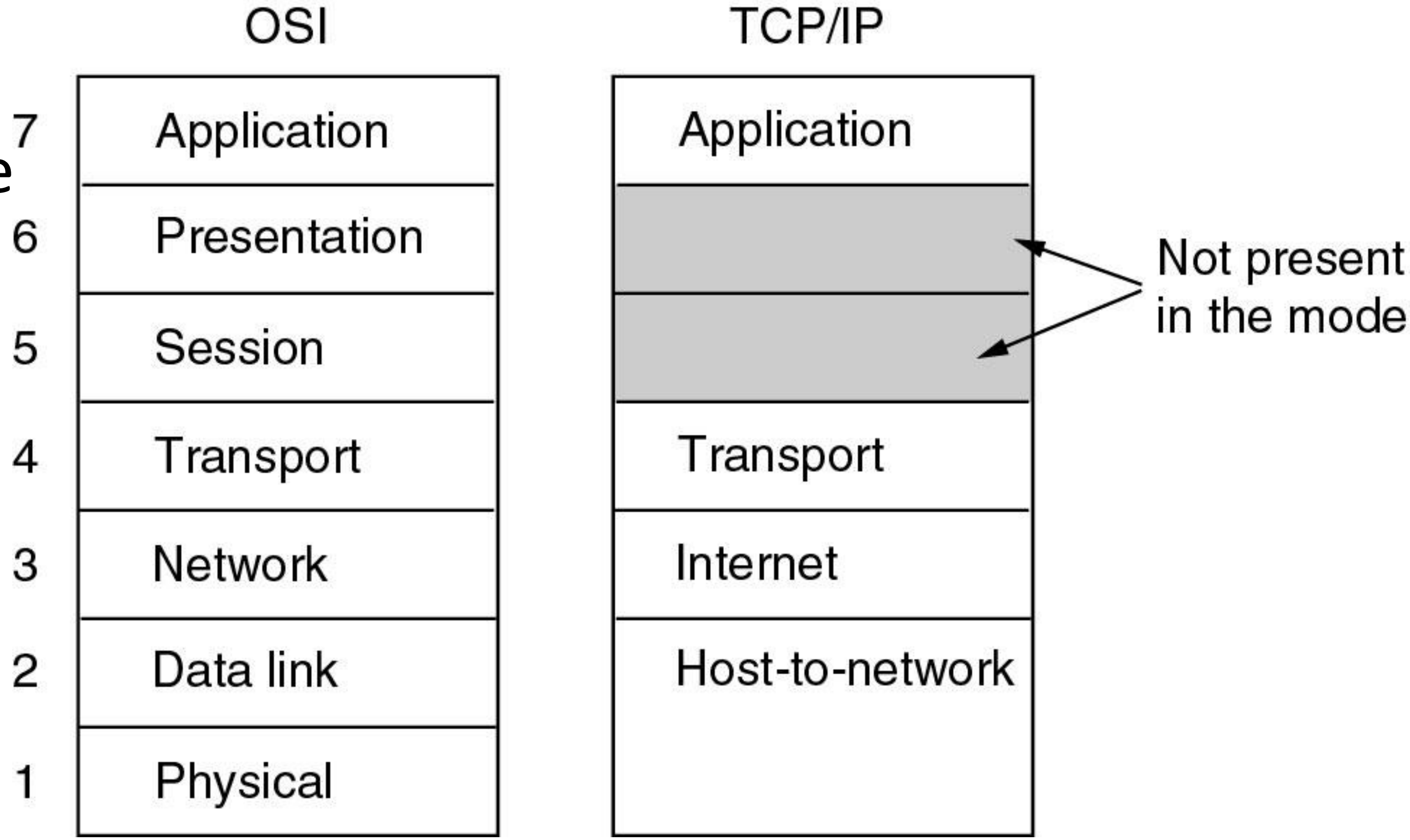
Reference Models

The OSI reference model.



Reference Models (2)

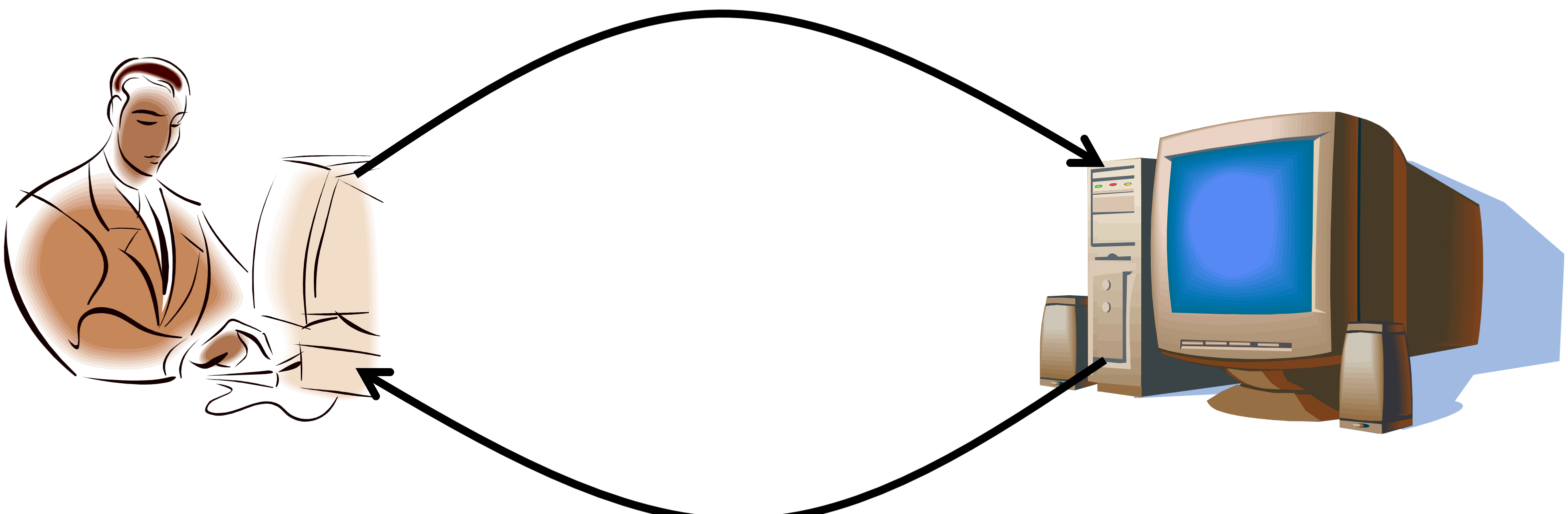
- The



Clients and Servers

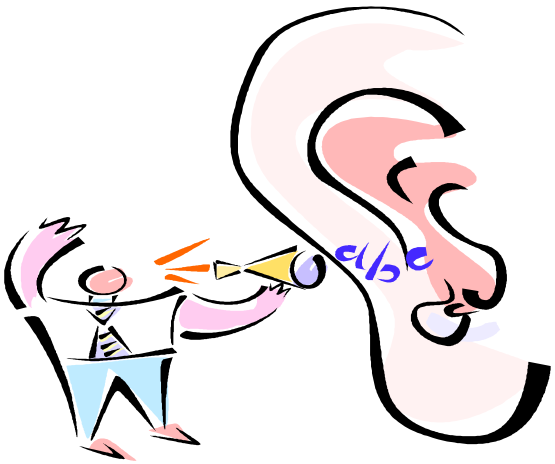
- Client program
  - Running on end host
  - Requests service
  - E.g., Web browser
- Server program
  - Running on end host
  - Provides service
  - E.g., Web server

GET /index.html



“Site under construction”

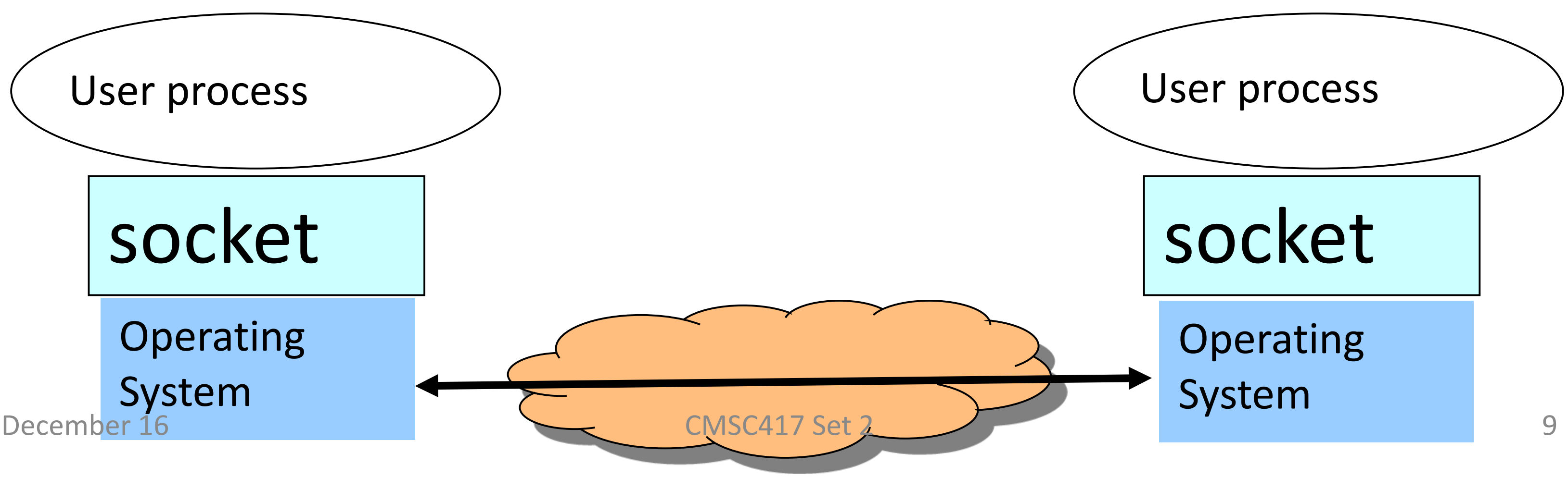
Servers Differ From Clients



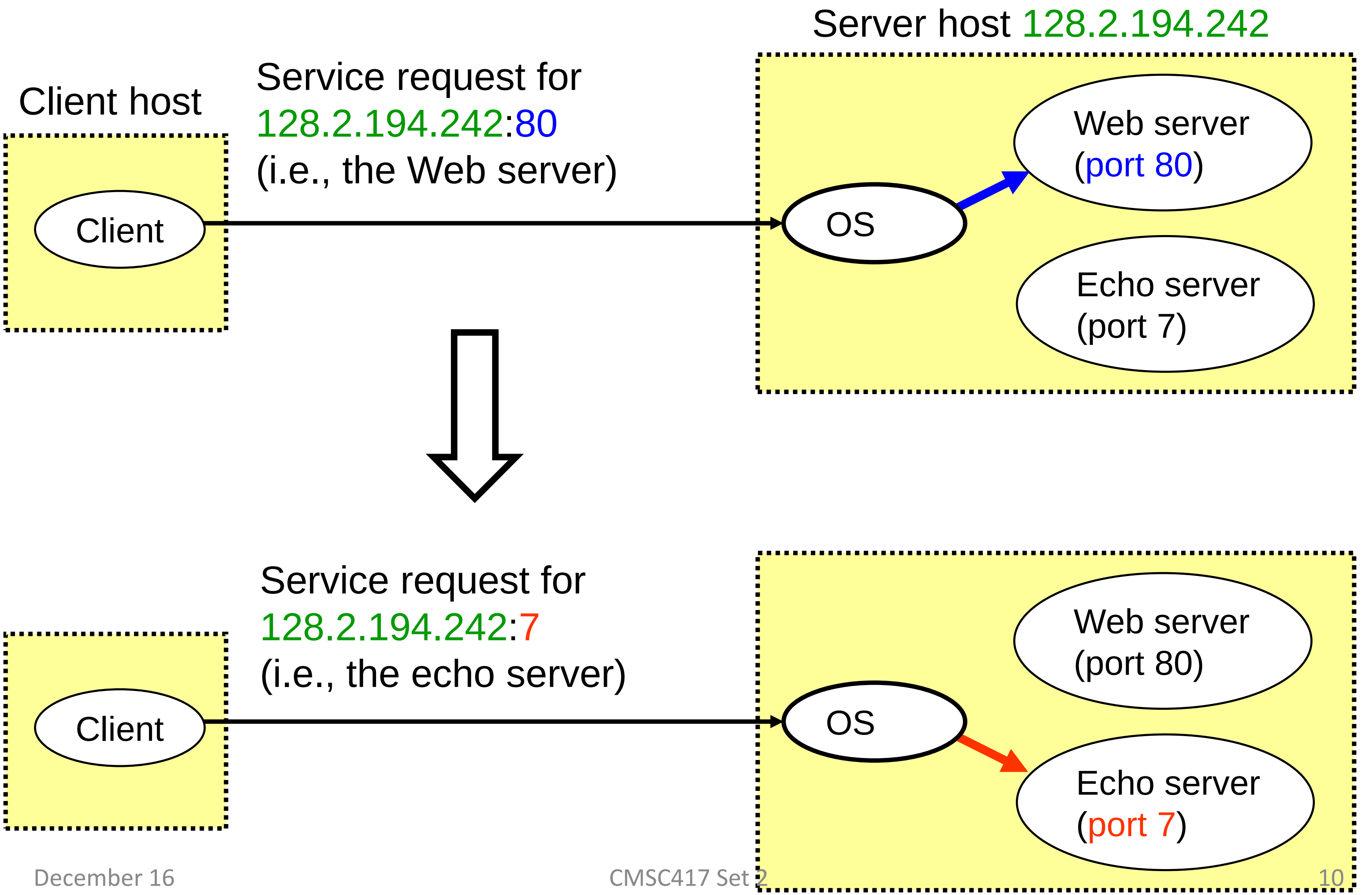
- Passive open
  - Prepare to accept connections
  - … but don’t actually establish one
  - … until hearing from a client
- Hearing from multiple clients
  - Allow a backlog of waiting clients
  - ... in case several try to start a connection at once
- Create a socket for each client
  - Upon accepting a new client
  - … create a *new* socket for the communication

Socket: End Point of Communication

- Sending message from one process to another
  - Message must traverse the underlying network
- Process sends and receives through a "socket"
  - In essence, the doorway leading in/out of the house
- Socket as an Application Programming Interface
  - Supports the creation of network applications

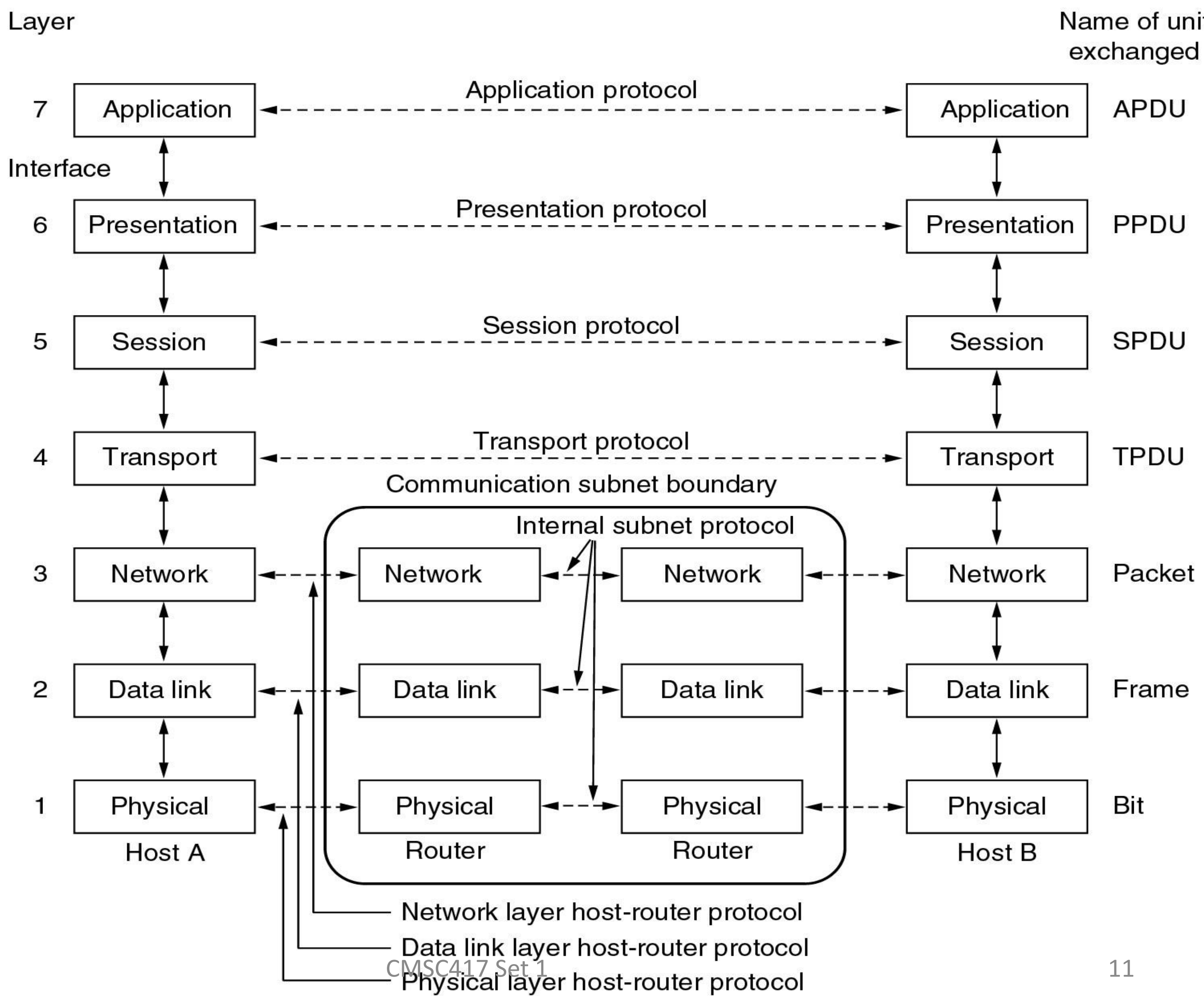


Using Ports to Identify Services



Reference Models

The OSI reference model.



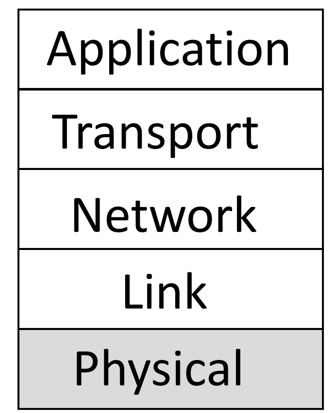
The Physical Layer

Foundation on which other layers build

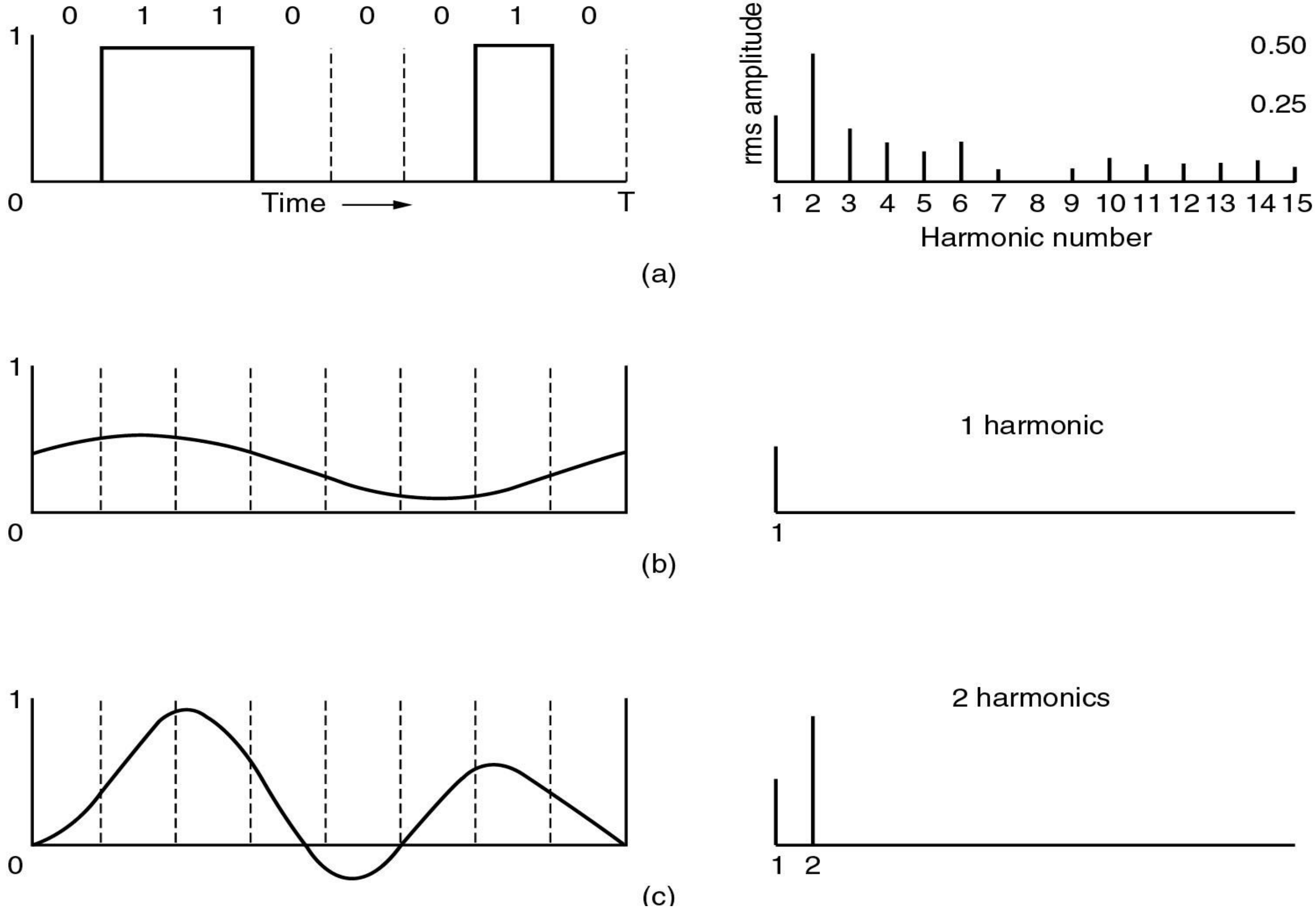
- Properties of wires, fiber, wireless limit what the network can do

Key problem is to send (digital) bits using only (analog) signals

- This is called modulation



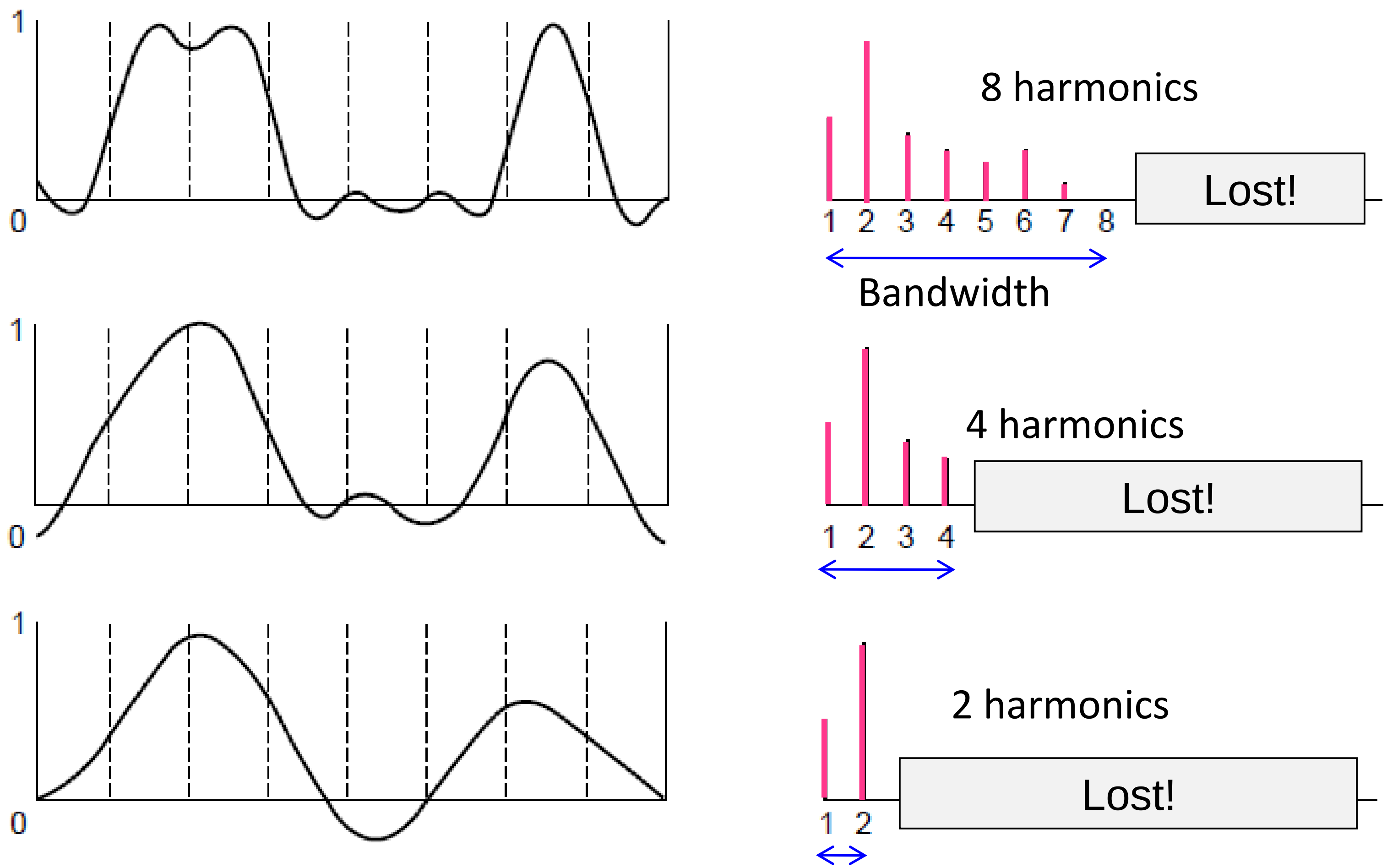
Bandwidth-Limited Signals



A binary signal and its root-mean-square Fourier amplitudes.

(b) – (c) Successive approximations to the original signal.

Bandwidth-Limited Signals

- 

Guided Transmission (Wires & Fiber)

Media have different properties, hence performance

- Reality check
  - Storage media »
- Wires:
  - Twisted pairs »
  - Coaxial cable »
  - Power lines »
- Fiber cables »

Wireless Transmission

- Electromagnetic Spectrum »
- Radio Transmission »
- Microwave Transmission »
- Light Transmission »
- Wireless vs. Wires/Fiber »

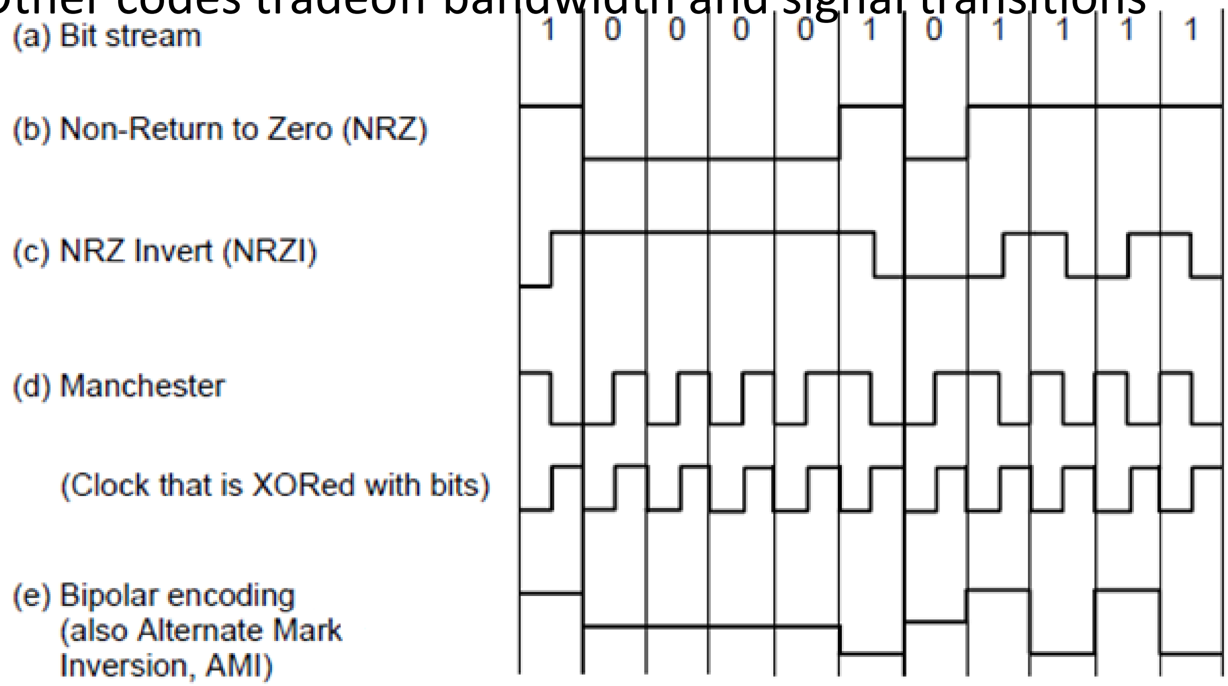
Digital Modulation and Multiplexing

Modulation schemes send bits as signals; multiplexing schemes share a channel among users.

- Baseband Transmission »
- Passband Transmission »
- Frequency Division Multiplexing »
- Time Division Multiplexing »
- Code Division Multiple Access »

Baseband Transmission

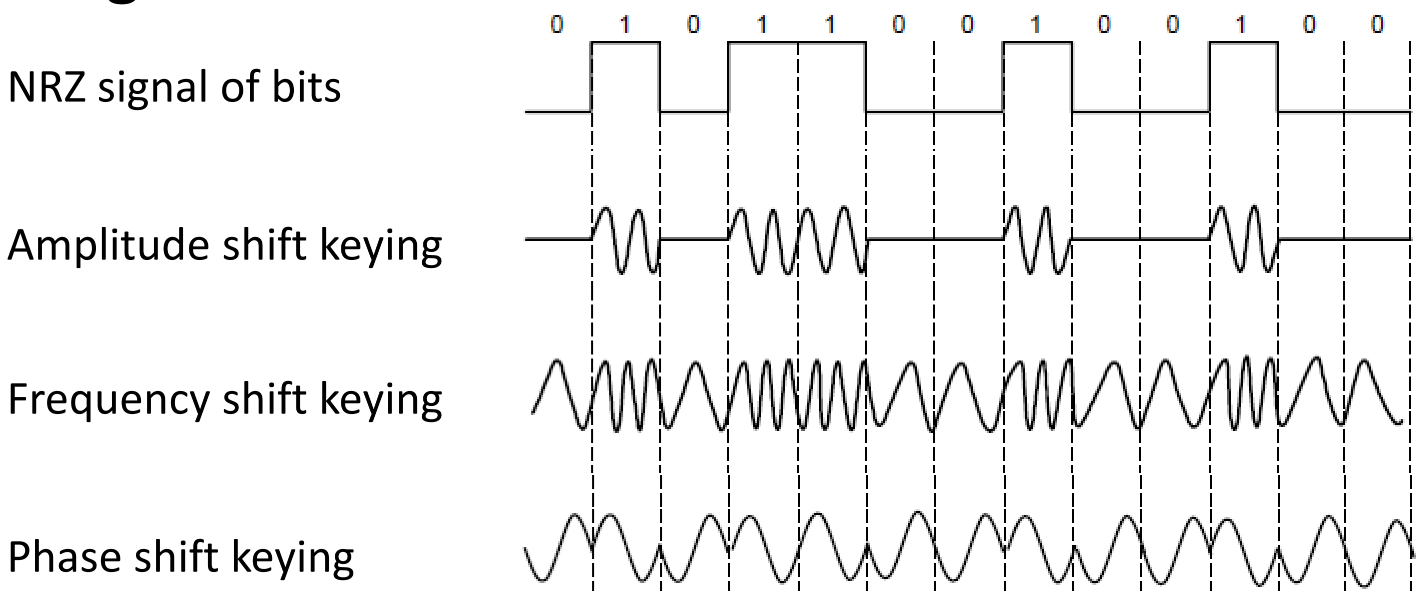
- Line codes send <u>symbols</u> that represent one or more bits
  - NRZ is the simplest, literal line code (+1V="1", -1V="0")
  - Other codes tradeoff bandwidth and signal transitions
-



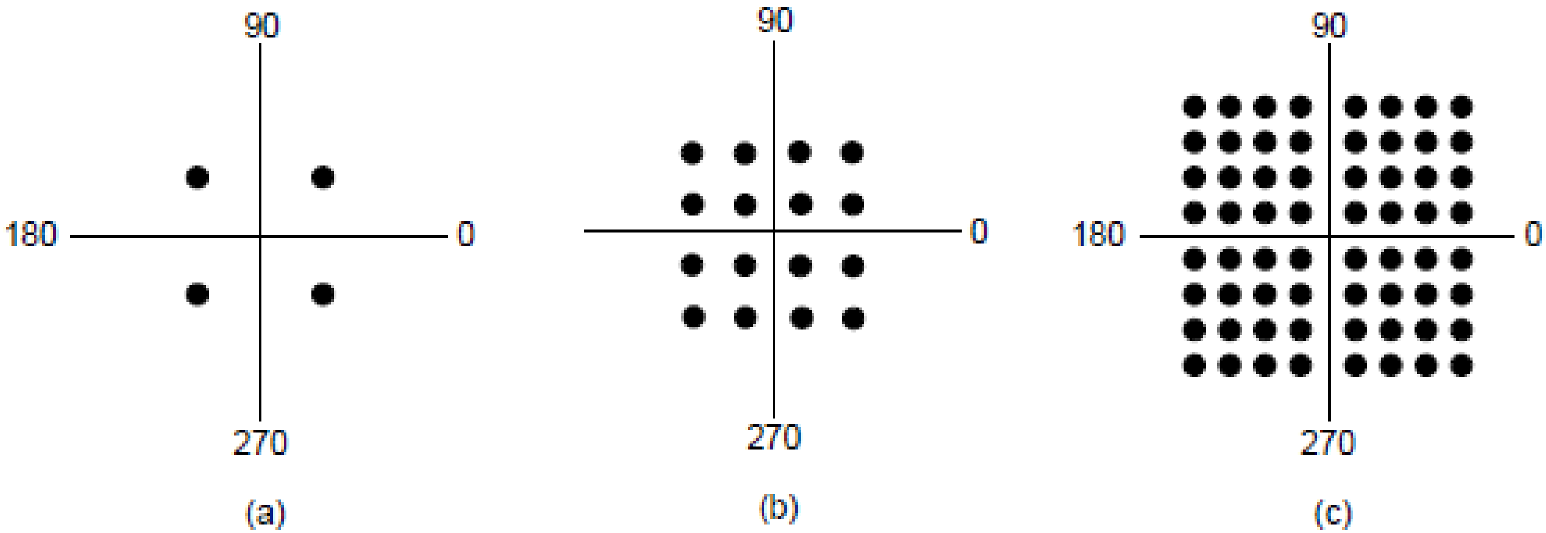
Four different line codes

Passband Transmission (1)

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



Passband Transmission (2)

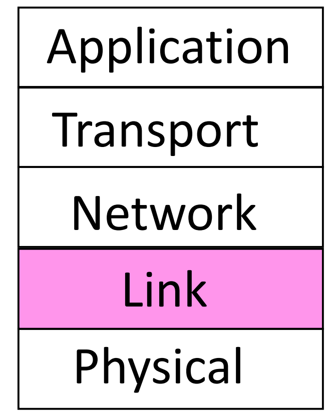


(a) QPSK. (b) QAM-16. (c) QAM-64.

The Data Link Layer

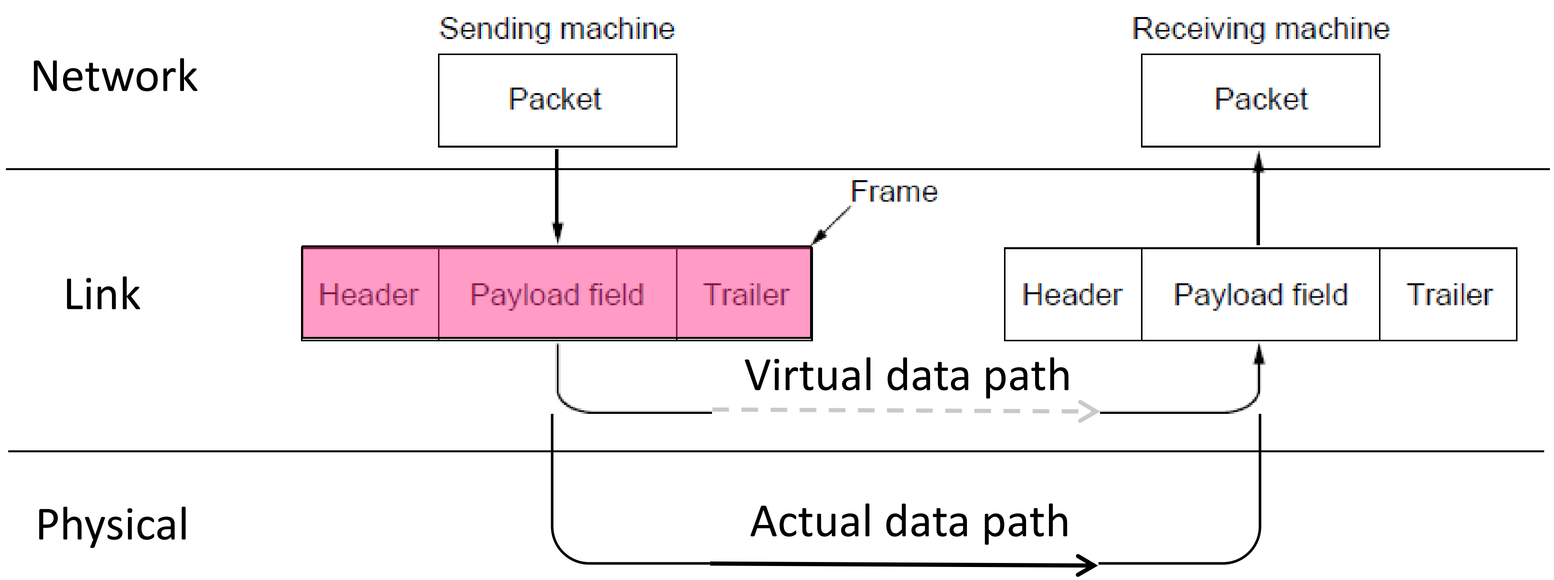
Responsible for delivering frames of information over a single link

- Handles transmission errors and regulates the flow of data



Frames

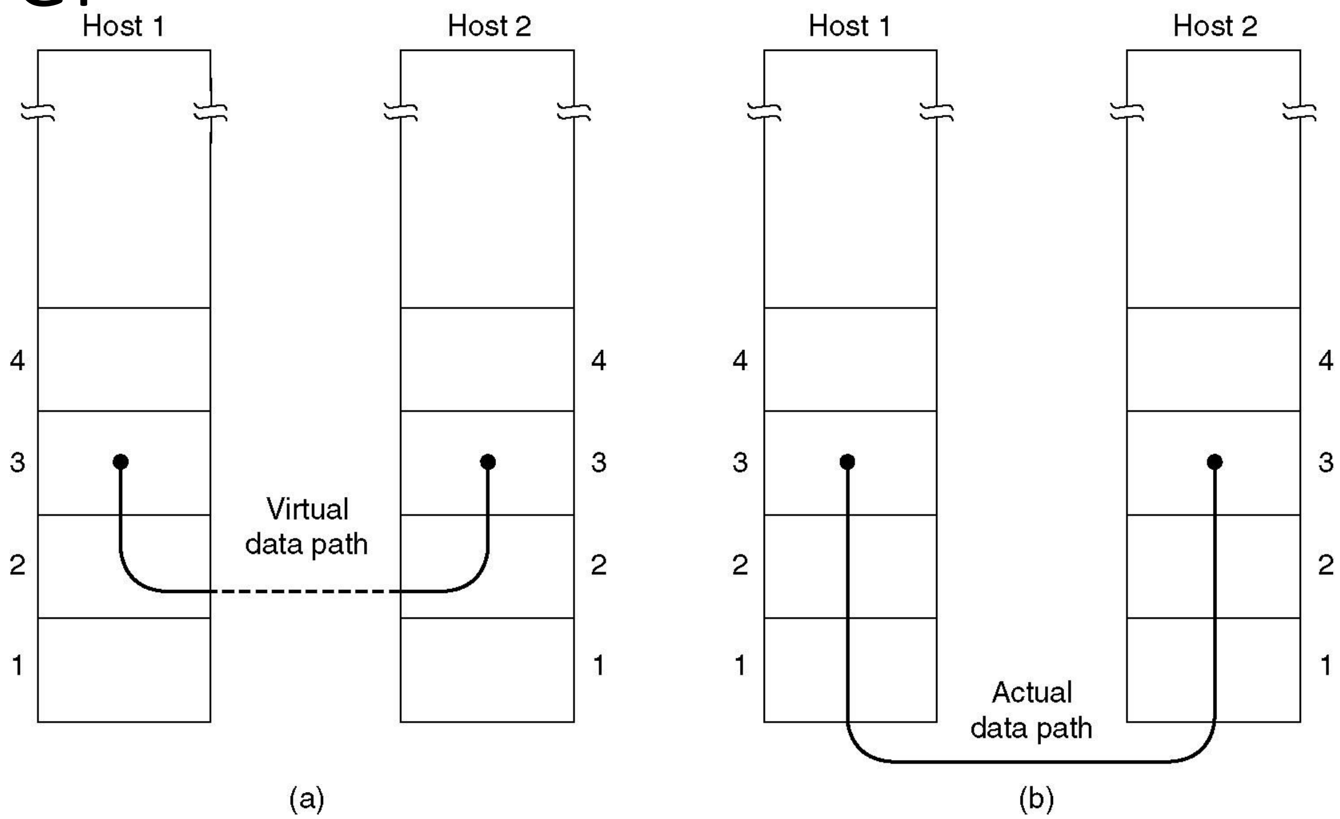
Link layer accepts packets from the network layer, and encapsulates them into frames that it sends using the physical layer; reception is the opposite process



Functions of the Data Link Layer

- Provide service interface to the network layer
- Dealing with transmission errors
- Regulating data flow
  - Slow receivers not swamped by fast senders

Services Provided to Network Layer



(a) Virtual communication.
(b) Actual communication.

Framing Methods

- Byte count »
- Flag bytes with byte stuffing »
- Flag bits with bit stuffing »
- Physical layer coding violations
  - Use non-data symbol to indicate frame

Flow Control

Prevents a fast sender from out-pacing a slow receiver

- Receiver gives feedback on the data it can accept
- Rare in the Link layer as NICs run at “wire speed”
  - Receiver can take data as fast as it can be sent

Flow control is a topic in the Link and Transport layers.

Error Detection and Correction

Error codes add structured redundancy to data so errors can be either detected, or corrected.

Error correction codes:

- Hamming codes »
- Binary convolutional codes »
- Reed-Solomon and Low-Density Parity Check codes
  - Mathematically complex, widely used in real systems

Error detection codes:

- Parity »
- Checksums »
- Cyclic redundancy codes »

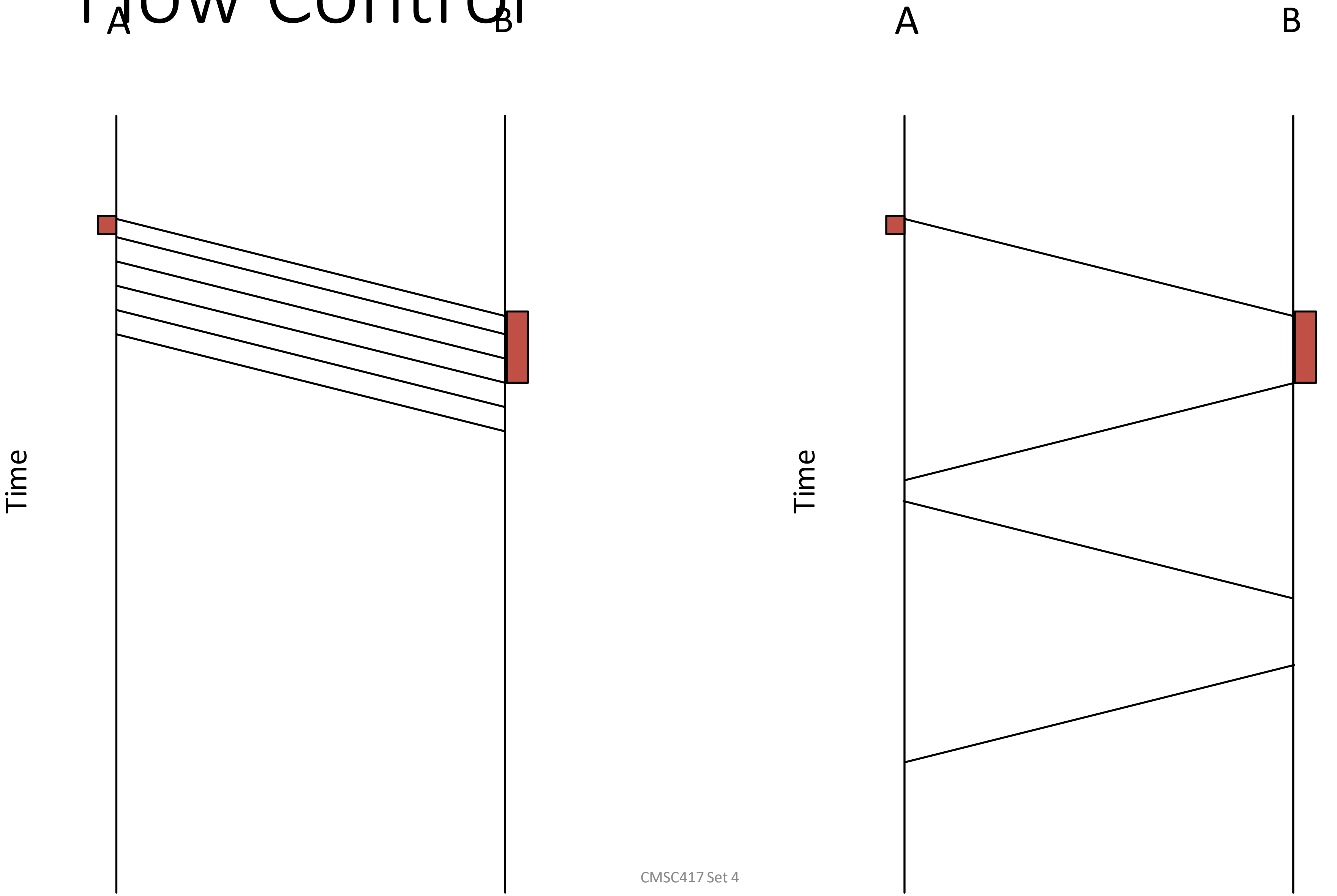
Cyclic Redundancy Check

- Have to maximize the probability of detecting the errors using a small number of additional bits.
- Based on powerful mathematical formulations – theory of finite fields
- Consider (n+1) bits as n degree polynomial
- Message M(x) represented as polynomial
- Divisor C(x) of degree k
- Send P(x) as (n+1) bits +k bits such that P(x) is exactly divisible by C(x)

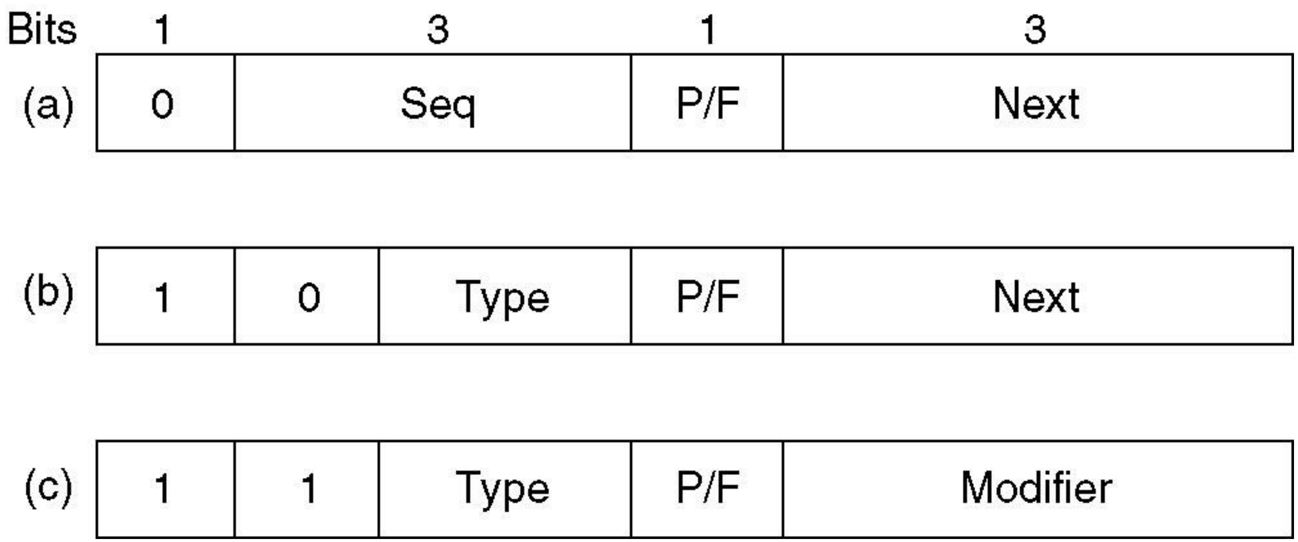
$$C(x) = x^3 + x^2 + 1$$

$$M(x) = x^7 + x^4 + x^3 + x^1$$

Flow Control



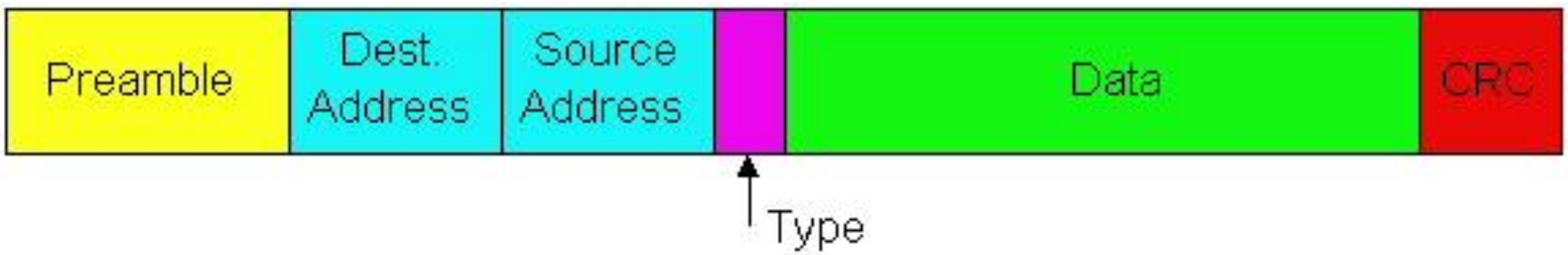
High-Level Data Link Control (2)



Control field of
(a) An information frame.
(b) A supervisory frame.
(c) An unnumbered frame.

Ethernet Frame Structure

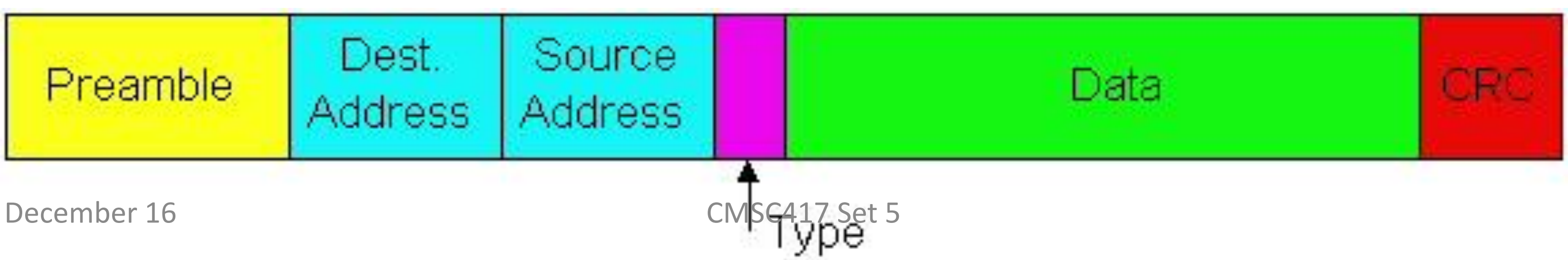
- Sending adapter encapsulates packet in frame



- Preamble: synchronization
  - Seven bytes with pattern 10101010, followed by one byte with pattern 10101011
  - Used to synchronize receiver, sender clock rates

Ethernet Frame Structure (Continued)

- Addresses: source and destination MAC addresses
  - Adaptor passes frame to network-level protocol
    - If destination address matches the adaptor
    - Or the destination address is the broadcast address
  - Otherwise, adapter discards frame
- Type: indicates the higher layer protocol
  - Usually IP
  - But also Novell IPX, AppleTalk, …
- CRC: cyclic redundancy check
  - Checked at receiver
  - If error is detected, the frame is simply dropped



Classic Ethernet (2) – MAC

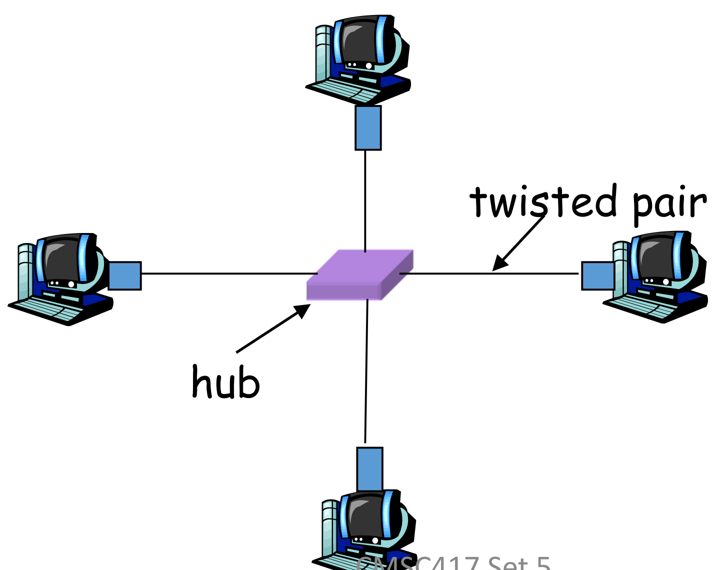
MAC protocol is 1-persistent CSMA/CD Random Delay (backoff) after collision is computed with BEB (Binary Exponential Backoff)

- Frame format is still used with modern Ethernet.

Bytes	8		6	6	2	0-1500	0-46	4
Ethernet (DIX)	Preamble		Destination address	Source address	Type	Data	Pad	Check-sum
IEEE 802.3	Preamble	SoF	Destination address	Source address	Length	Data	Pad	Check-sum

Hubs: Physical-Layer Repeaters

- Hubs are physical-layer repeaters
  - Bits coming from one link go out all other links
  - At the same rate, with no frame buffering
  - No CSMA/CD at hub: adapters detect collisions



Switch

- Link layer device
  - Stores and forwards Ethernet frames
  - Examines frame header and selectively forwards frame based on MAC dest address
  - When frame is to be forwarded on segment, uses CSMA/CD to access segment
- Transparent
  - Hosts are unaware of presence of switches
- Plug-and-play, self-learning
  - Switches do not need to be configured

Wireless Networking Technologies

IEEE 802.11 Overview

- Adopted in 1997.

Defines;

- MAC sublayer
- MAC management protocols and services
- Physical (PHY) layers
  - IR
  - FHSS
  - DSSS

Goals

- To deliver services in wired networks
- To achieve high throughput
- To achieve highly reliable data delivery
- To achieve continuous network connection.

802.11 Frames

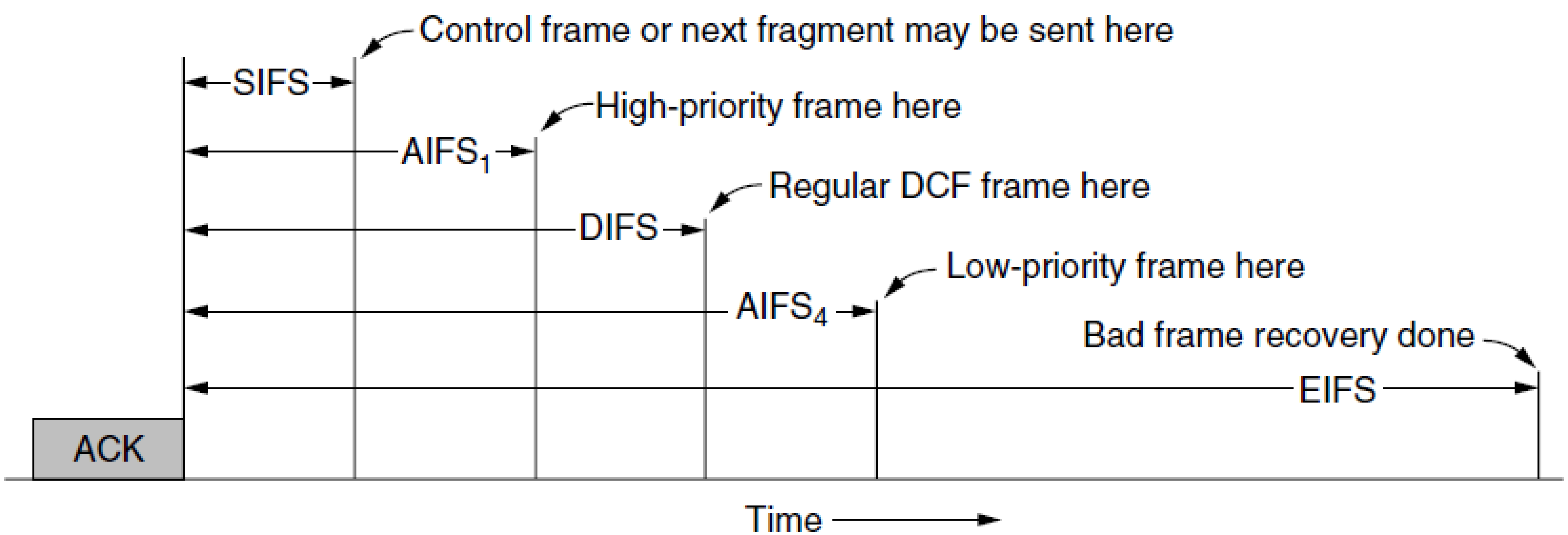
- Frames vary depending on their type (Frame control)
- Data frames have 3 addresses to pass via APs

Bytes	2	2	6	6	6	2	0–2312	4
	Frame control	Duration	Address 1 (recipient)	Address 2 (transmitter)	Address 3	Sequence	Data	Check sequence

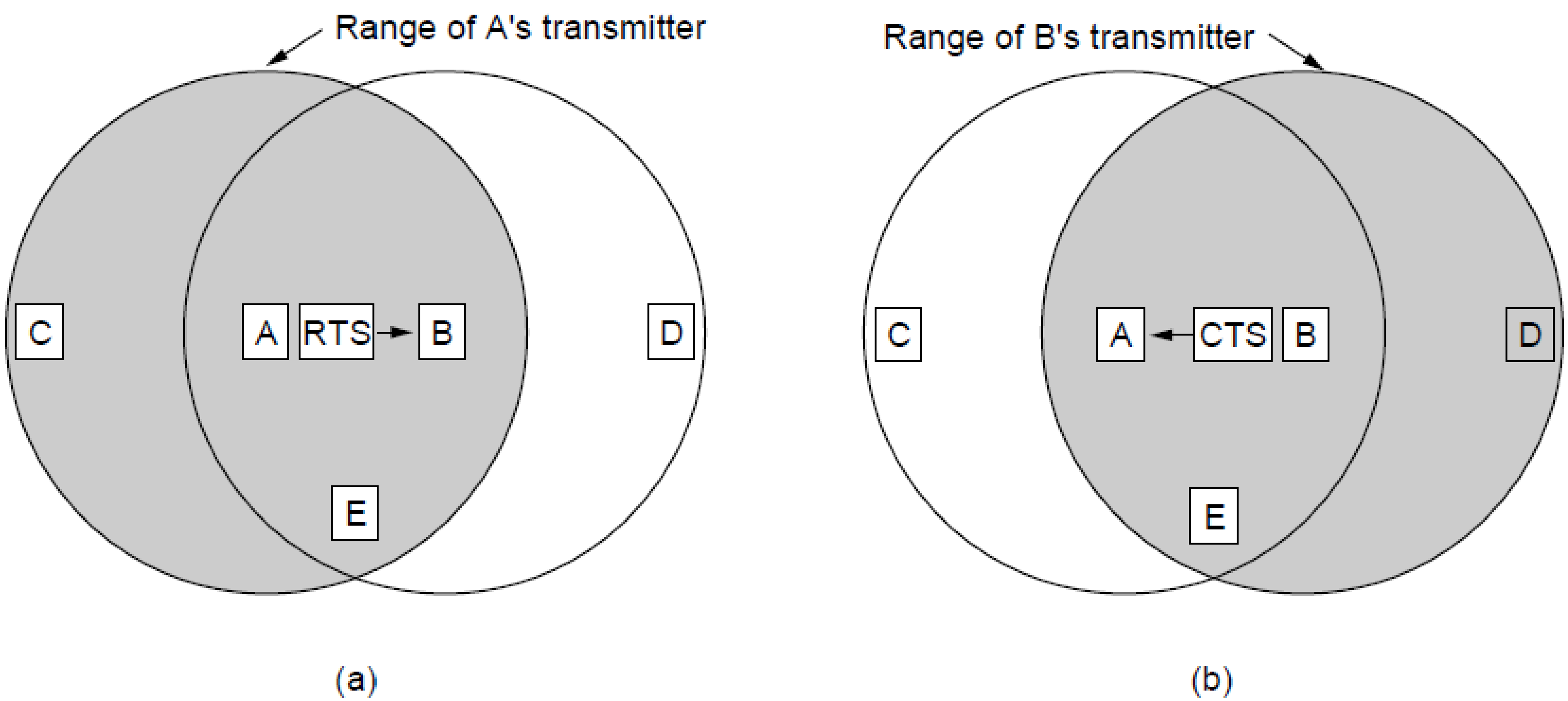
	Version = 00	Type = 10	Subtype = 0000	To DS	From DS	More frag.	Retry	Pwr. mgt.	More data	Protected	Order
Bits	2	2	4	1	1	1	1	1	1	1	1

802.11 MAC

- Different backoff slot times add quality of service
  - Short intervals give preferred access, e.g., control, VoIP
- MAC has other mechanisms too, e.g., power save



RTS/CTS

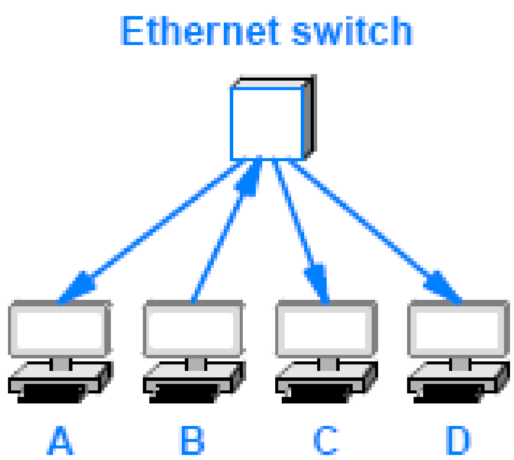


The MACA protocol. (a) *A sending an RTS to B. (b) B responding* with a CTS to *A.*

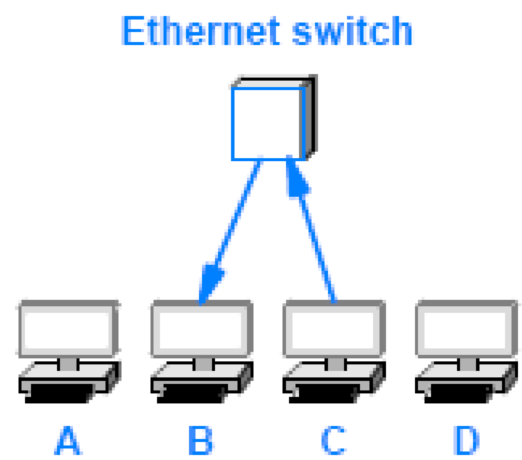
Address Resolution

- A crucial step of the forwarding process requires a translation:
  - forwarding uses IP addresses
  - a frame transmitted must contain the MAC address of the next hop
  - IP must translate the next-hop IP address to a MAC address
- The principle is:
  - IP addresses are abstractions
    - provided by protocol software
  - Network does not know how to locate a computer from its IP address
    - the next-hop address must be translated to an equivalent MAC address
- Translation from a computer's IP address to an equivalent hardware address is known as address resolution
  - And an IP address is said to be resolved to the correct MAC address
- Address resolution is local to a network

The Address Resolution Protocol (ARP)



(a)

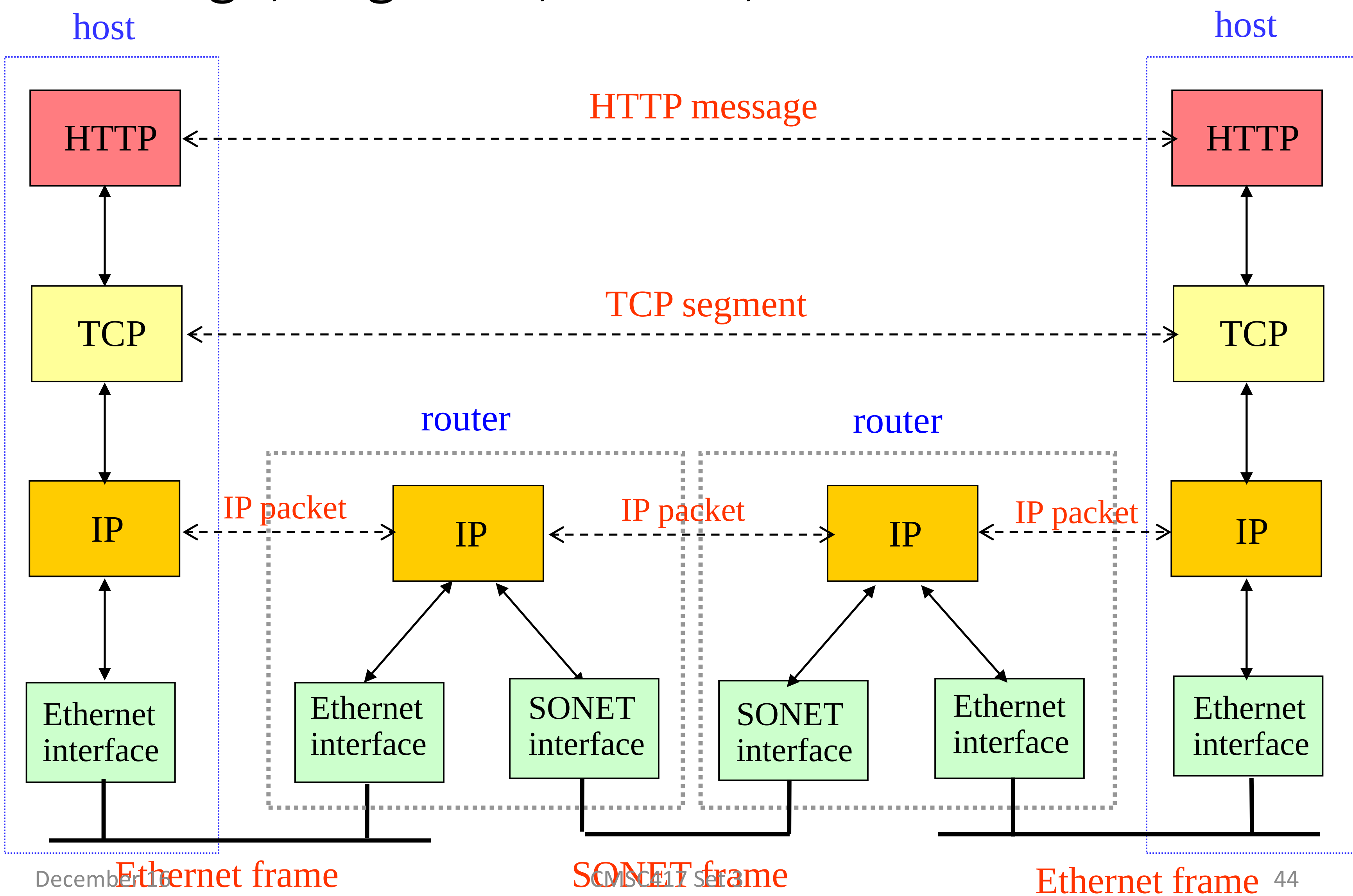


(a)

Illustration of the ARP message exchange when computer *B* resolves the address of computer *C*.

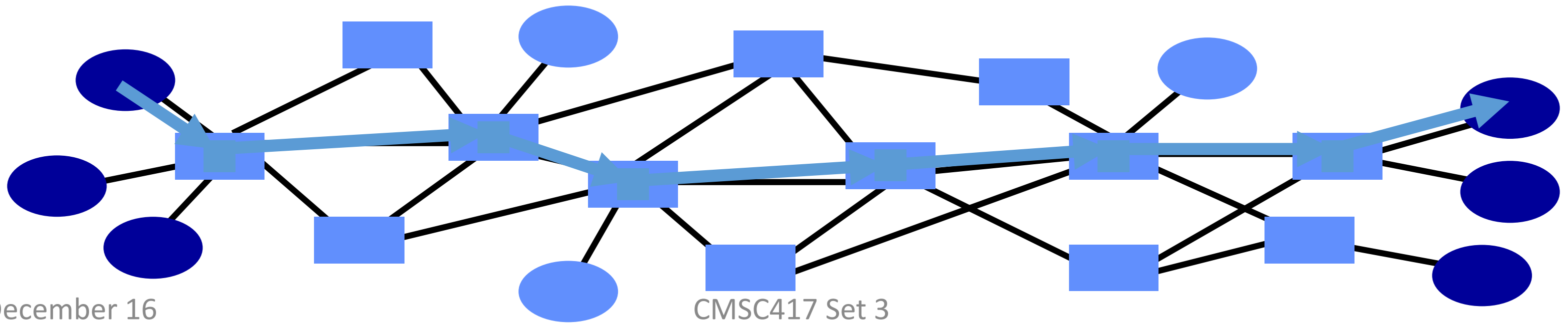
The Network Layer

Message, Segment, Packet, and Frame

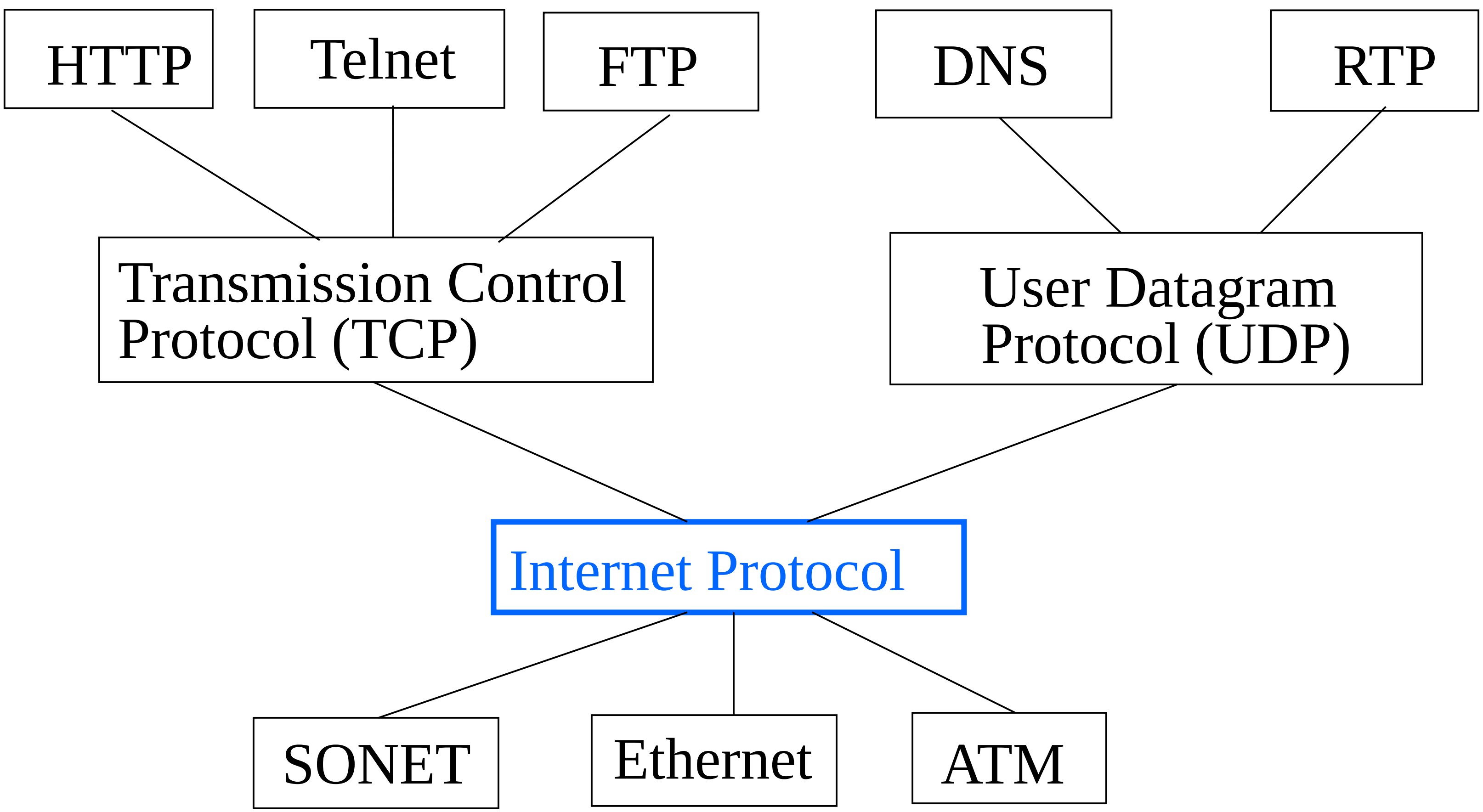


Packet Switching (e.g., Internet)

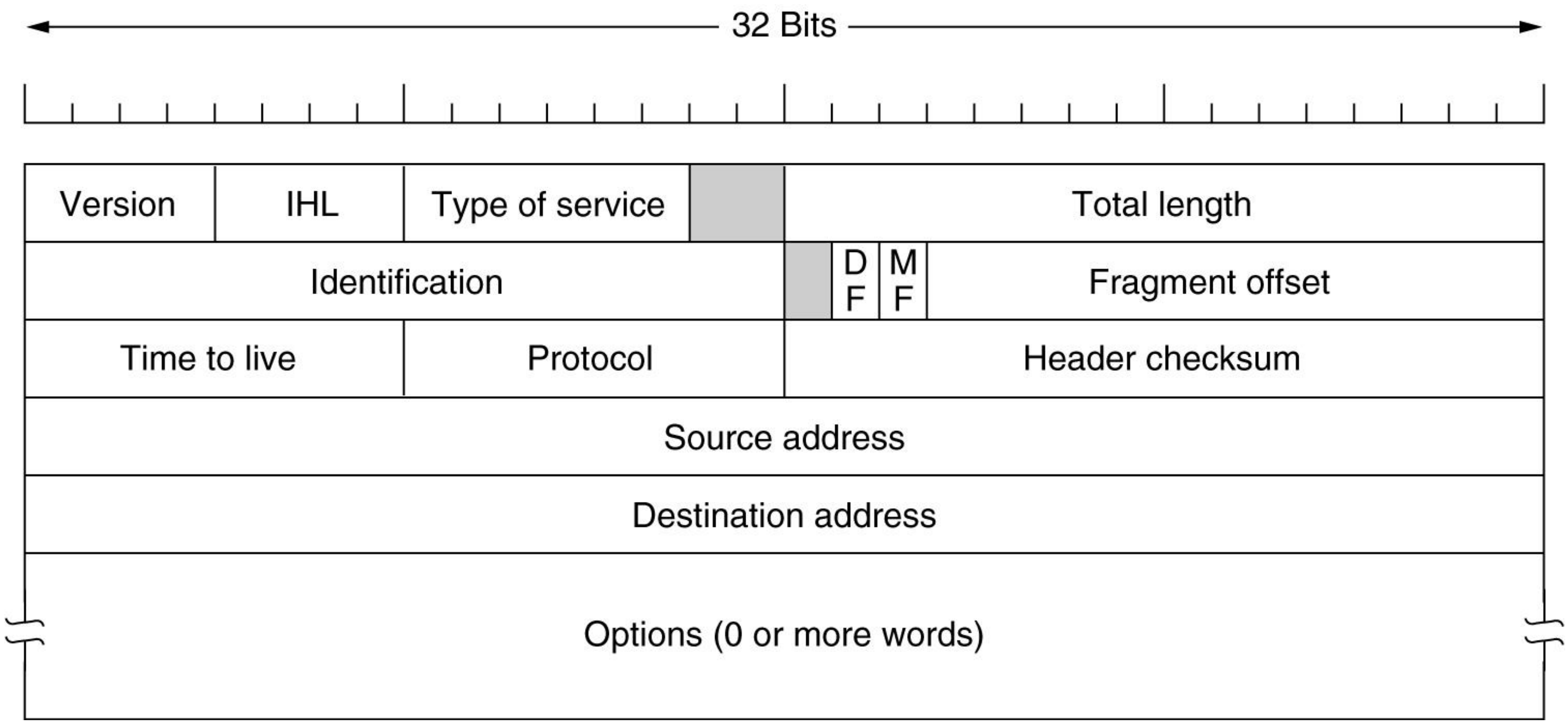
- Data traffic divided into packets
  - Each packet contains a header (with address)
- Packets travel separately through network
  - Packet forwarding based on the header
  - Network nodes may store packets temporarily
- Destination reconstructs the message



Layering in the IP Protocols



The IP Protocol

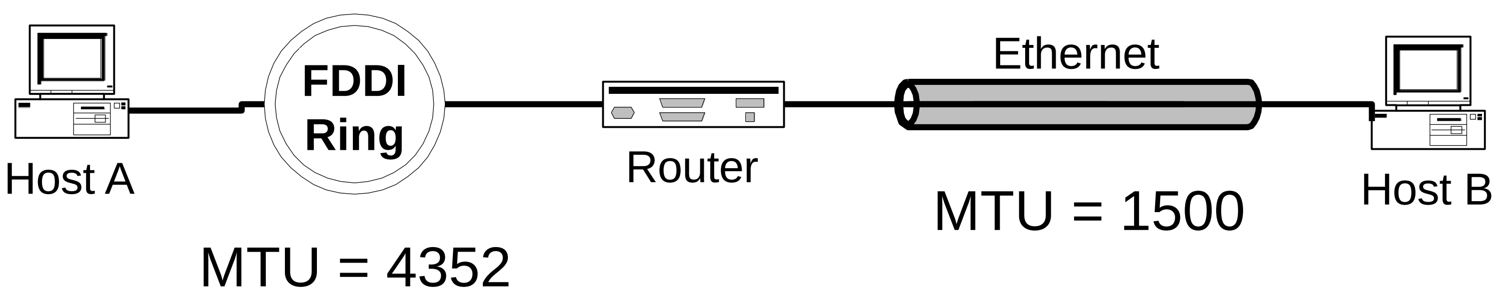


IP Datagram

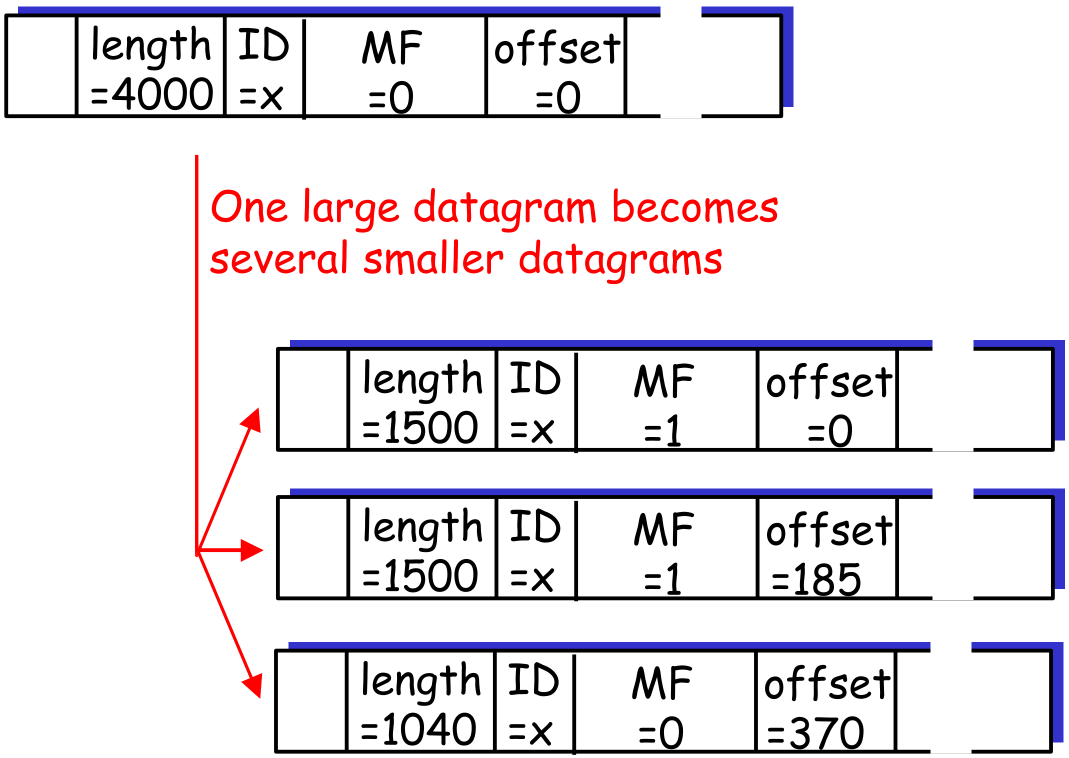
<table>
<tr><td></td><td colspan="2">Bit 0</td><td colspan="3" align="right">Bit 31</td></tr>
<tr><td rowspan="6">Header</td><td>Version (4)</td><td>Hdr Len (4)</td><td>TOS (8)</td><td colspan="2">Total Length in bytes (16)</td></tr>
<tr><td colspan="3">Identification (16 bits)</td><td>Flags (3)</td><td>Fragment Offset (13)</td></tr>
<tr><td colspan="2">Time to Live (8)</td><td>Protocol (8)</td><td colspan="2">Header Checksum (16)</td></tr>
<tr><td colspan="5">Source IP Address</td></tr>
<tr><td colspan="5">Destination IP Address</td></tr>
<tr><td colspan="5">Options (if any)</td></tr>
<tr><td>Data</td><td colspan="5">Data (variable length)</td></tr>
</table>

IP Fragmentation & Reassembly

- Maximum Transmission Unit (MTU)
  - Largest IP packet a network will accept
  - Arriving IP packet may be larger (max IP packet size = 65,535 bytes)
- Sender or router will split the packet into multiple fragments
- Destination will reassemble the packet
- IP header fields used to identify and order related fragments



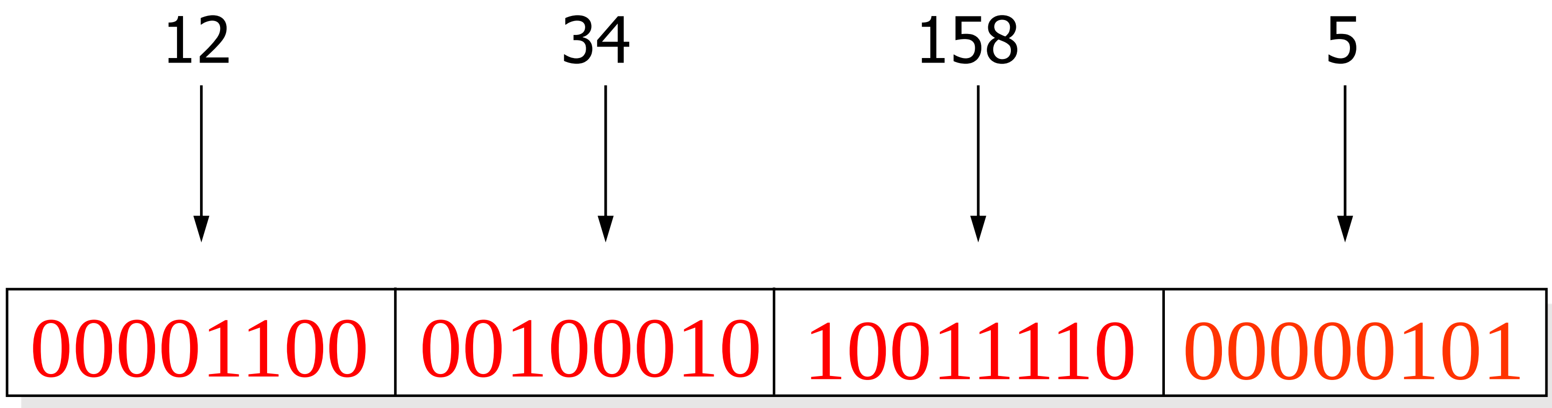
IP Fragmentation Example



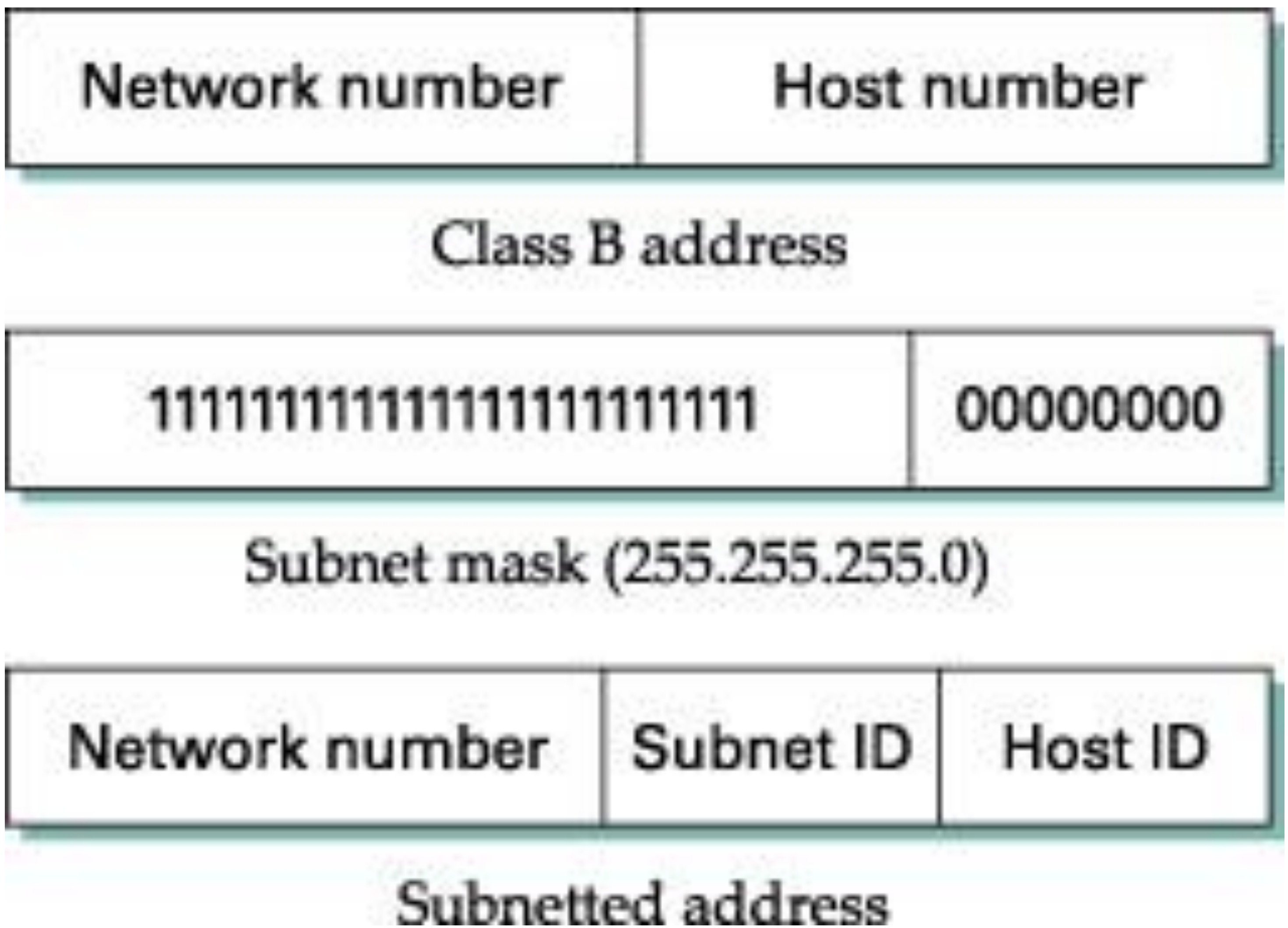
IP Addressing and Forwarding

IP Address (IPv4)

- A unique 32-bit number
- Identifies an interface (on a host, on a router, …)
- Represented in dotted-quad notation



Subnetted Address



Classful Addressing

- In the olden days, only fixed allocation sizes
  - Class A: 0*
    - Very large /8 blocks (e.g., MIT has 18.0.0.0/8)
  - Class B: 10*
    - Large /16 blocks (e.g,. Princeton has 128.112.0.0/16)
  - Class C: 110*
    - Small /24 blocks (e.g., AT&T Labs has 192.20.225.0/24)
  - Class D: 1110*
    - Multicast groups
  - Class E: 11110*
    - Reserved for future use
- This is why folks use dotted-quad notation!

Classless Inter-Domain Routing (CIDR)

Use two 32-bit numbers to represent a network.
Network number = IP address + Mask

IP Address : 12.4.0.0 IP Mask: 255.254.0.0

Address	00001100	00000100	00000000	00000000
Mask	11111111	11111110	00000000	00000000

← Network Prefix → ← for hosts →

Written as 12.4.0.0/15

Packet Forwarding

Separate Entry Classful Address

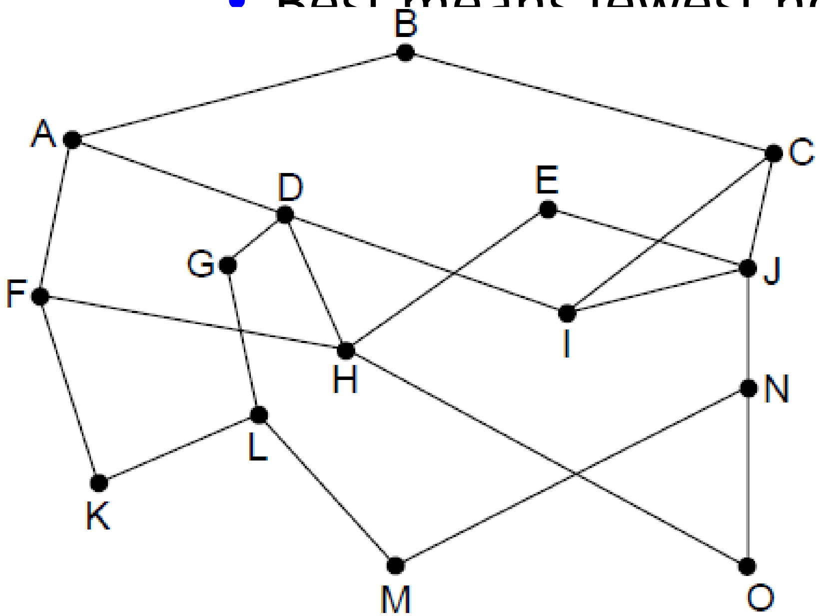
- If the router had an entry per classful prefix
  - Mixture of Class A, B, and C addresses
  - Depends on the first couple of bits of the destination
- Identify the mask automatically from the address
  - First bit of 0: class A address (/8)
  - First two bits of 10: class B address (/16)
  - First three bits of 110: class C address (/24)
- Then, look in the forwarding table for the match
  - E.g., 1.2.3.4 maps to 1.2.3.0/24
  - Then, look up the entry for 1.2.3.0/24
  - … to identify the outgoing interface

Routing

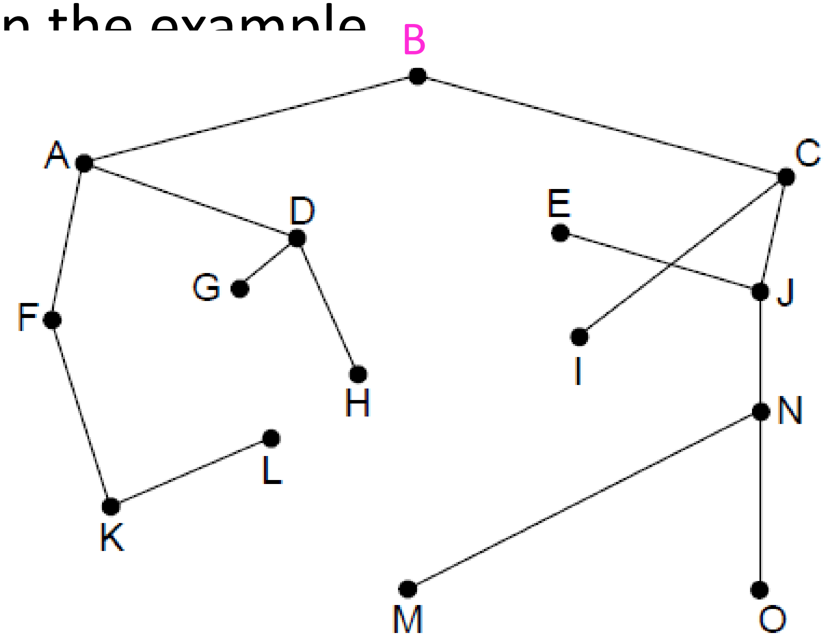
The Optimality Principle

Each portion of a best path is also a best path; the union of them to a router is a tree called the sink tree

- Best means fewest hops in the example



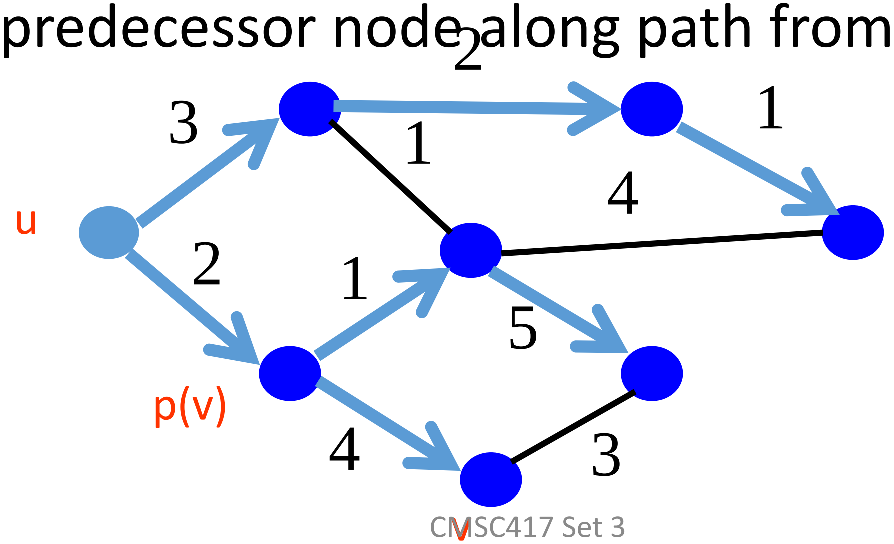
Network



Sink tree of best paths to router B

Shortest-Path Problem

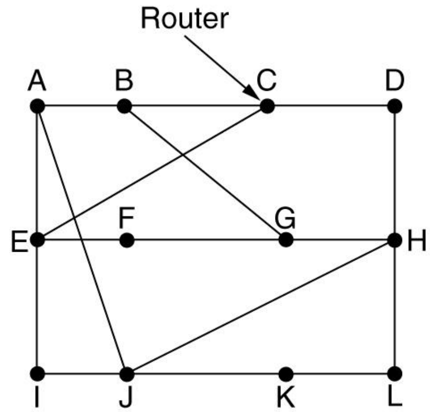
- Given: network topology with link costs
  - **c(x,y)**: link cost from node x to node y
  - Infinity if x and y are not direct neighbors
- Compute: least-cost paths to all nodes
  - From a given source u to all other nodes
  - **p(v)**: predecessor node along path from source to v



Link-State Routing

- Each router keeps track of its incident links
  - Whether the link is up or down
  - The cost on the link
- Each router broadcasts the link state
  - To give every router a complete view of the graph
- Each router runs Dijkstra's algorithm
  - To compute the shortest paths
  - … and construct the forwarding table
- Example protocols
  - Open Shortest Path First (OSPF)
  - Intermediate System – Intermediate System (IS-IS)

Distance Vector Routing



(a)

To	A	I	H	K	New estimated delay from J	Line
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 routing table for J	

Vectors received from J's four neighbors

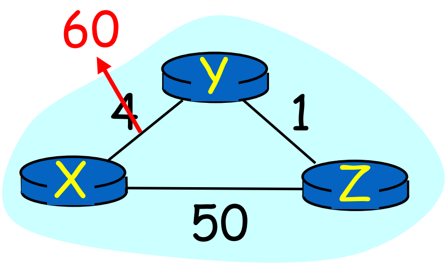
(b)

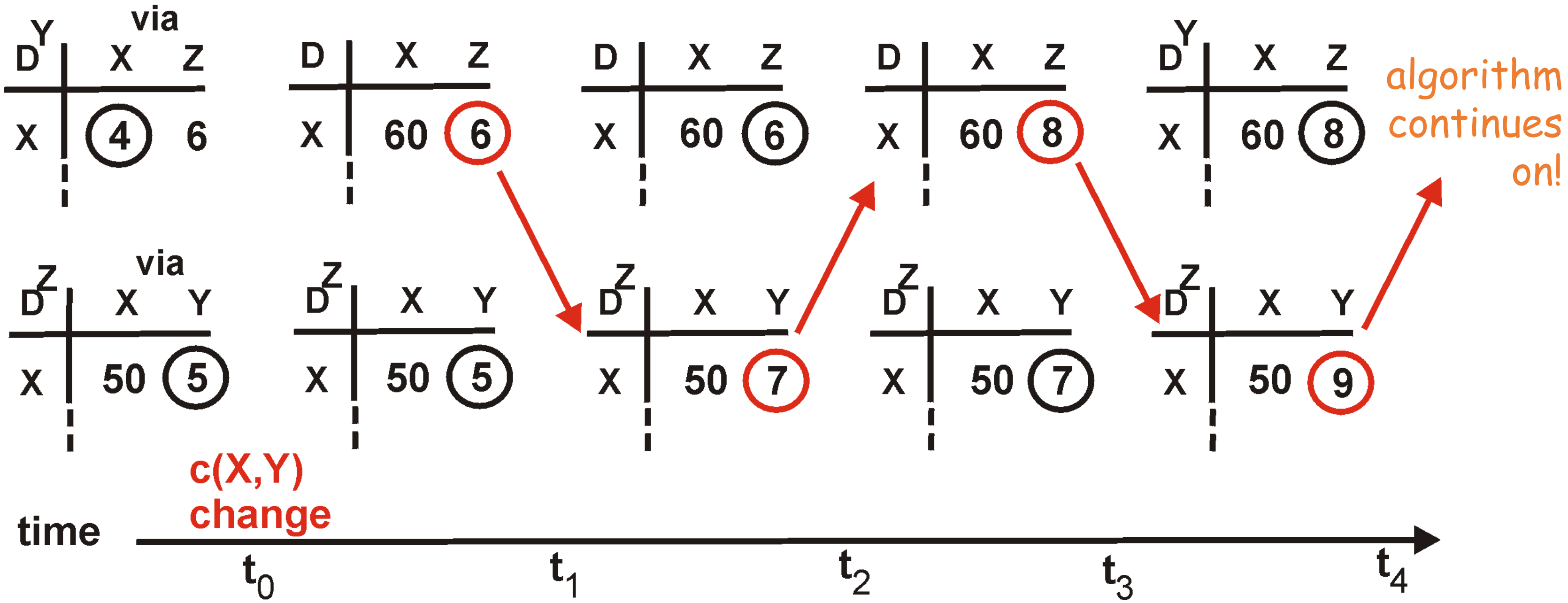
(a) A subnet. (b) Input from A, I, H, K, and the new routing table for J.

Distance Vector: Link Cost Changes

Link cost changes:

- Good news travels fast
- Bad news travels slow - “count to infinity” problem!

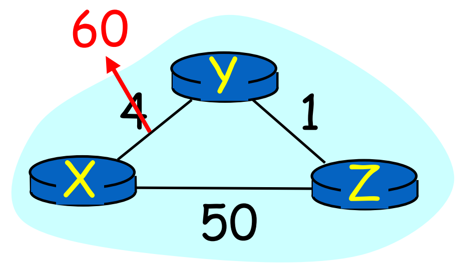


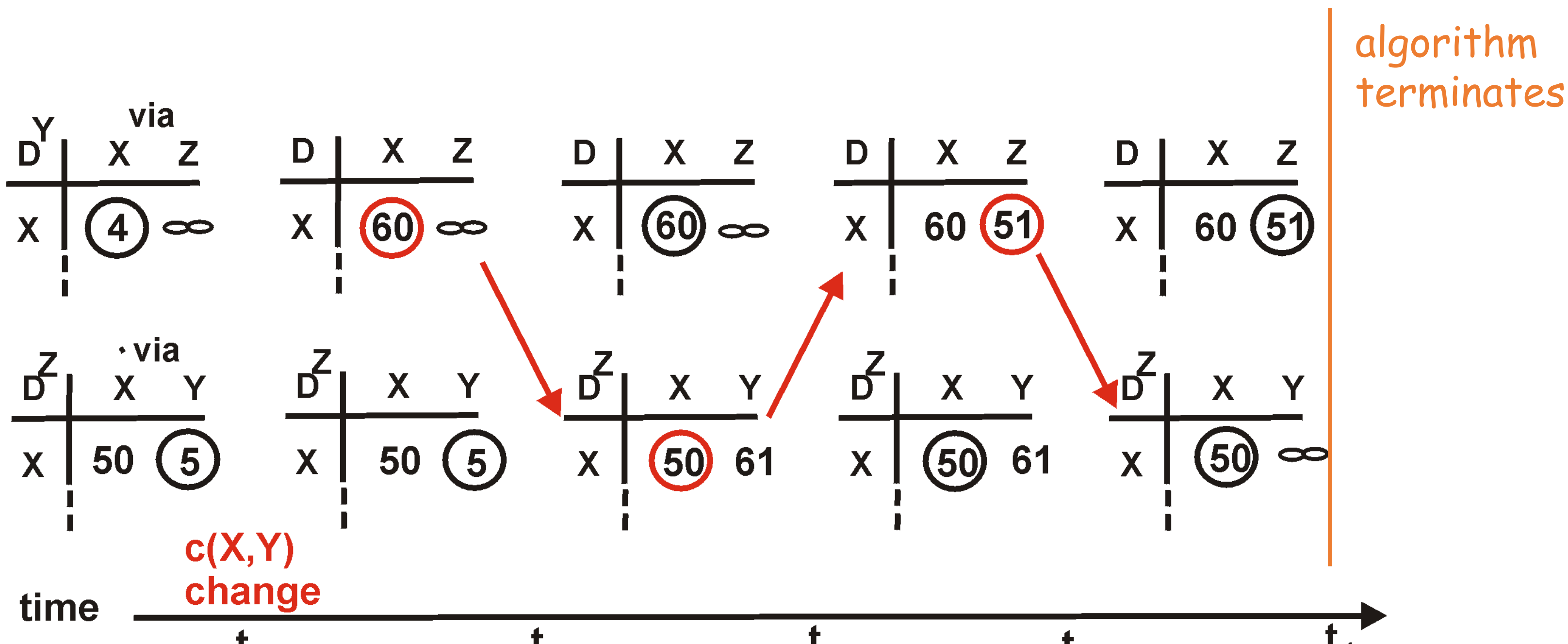


Distance Vector: Poison Reverse

If Z routes through Y to get to X :

- Z tells Y its (Z's) distance to X is infinite (so Y won't route to X via Z)
- Still, can have problems when more than 2 routers are involved



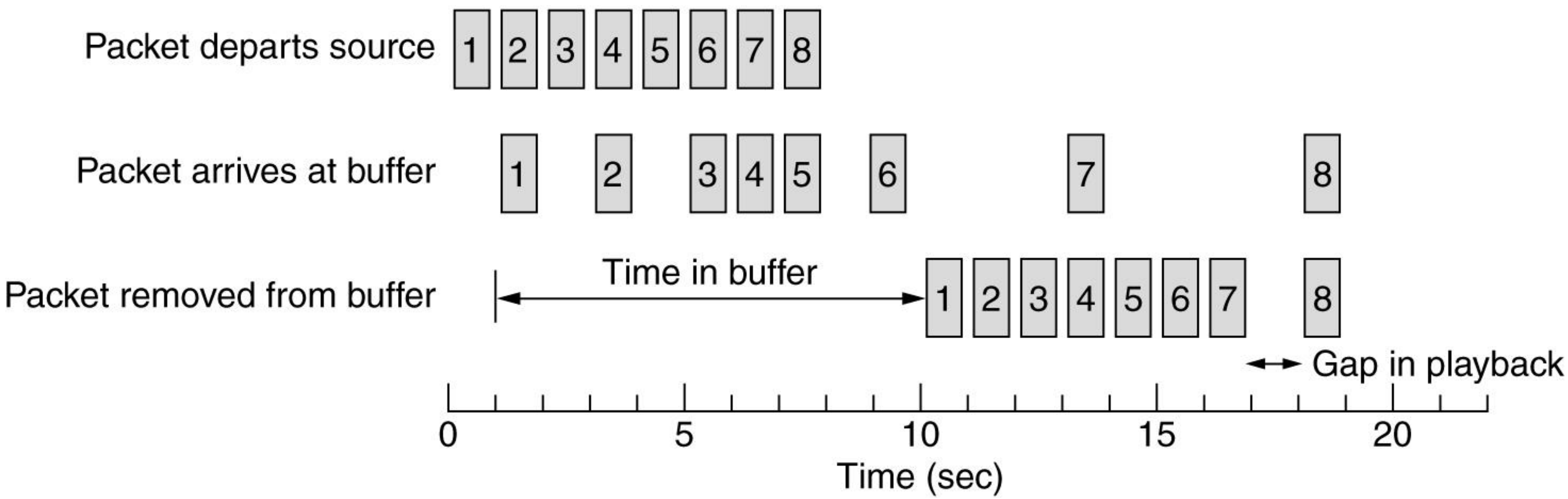


Congestion Control

- General Principles of Congestion Control
- Congestion Prevention Policies
- Congestion Control in Virtual-Circuit Subnets
- Congestion Control in Datagram Subnets
- Load Shedding
- Jitter Control

Buffering

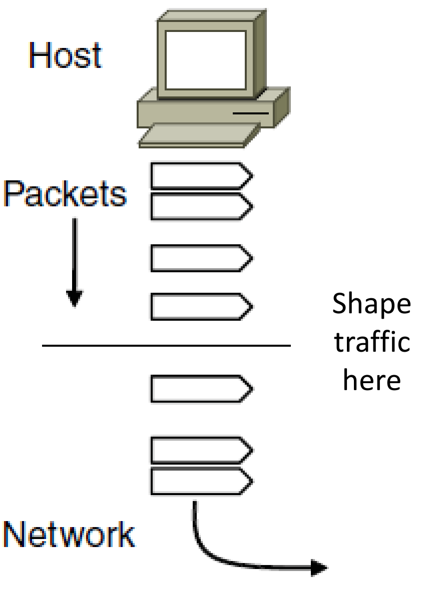
Smoothing the output stream by buffering packets.



Traffic Shaping (1)

Traffic shaping regulates the average rate and burstiness of data entering the network

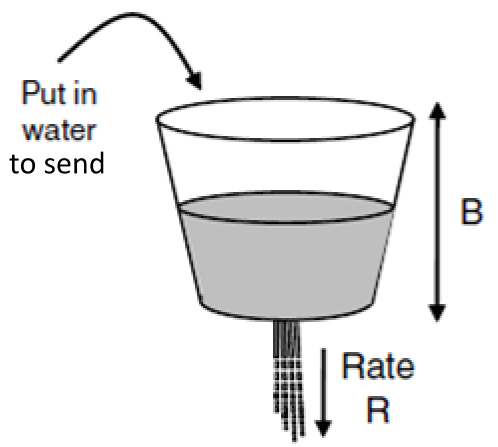
- Lets us make guarantees



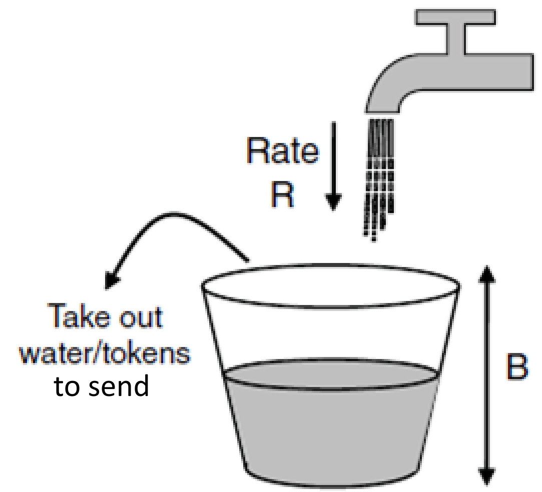
Traffic Shaping (2)

Token/Leaky bucket limits both the average rate (R) and short-term burst (B) of traffic

- For token, bucket size is B, water enters at rate R and is removed to send; opposite for leaky.

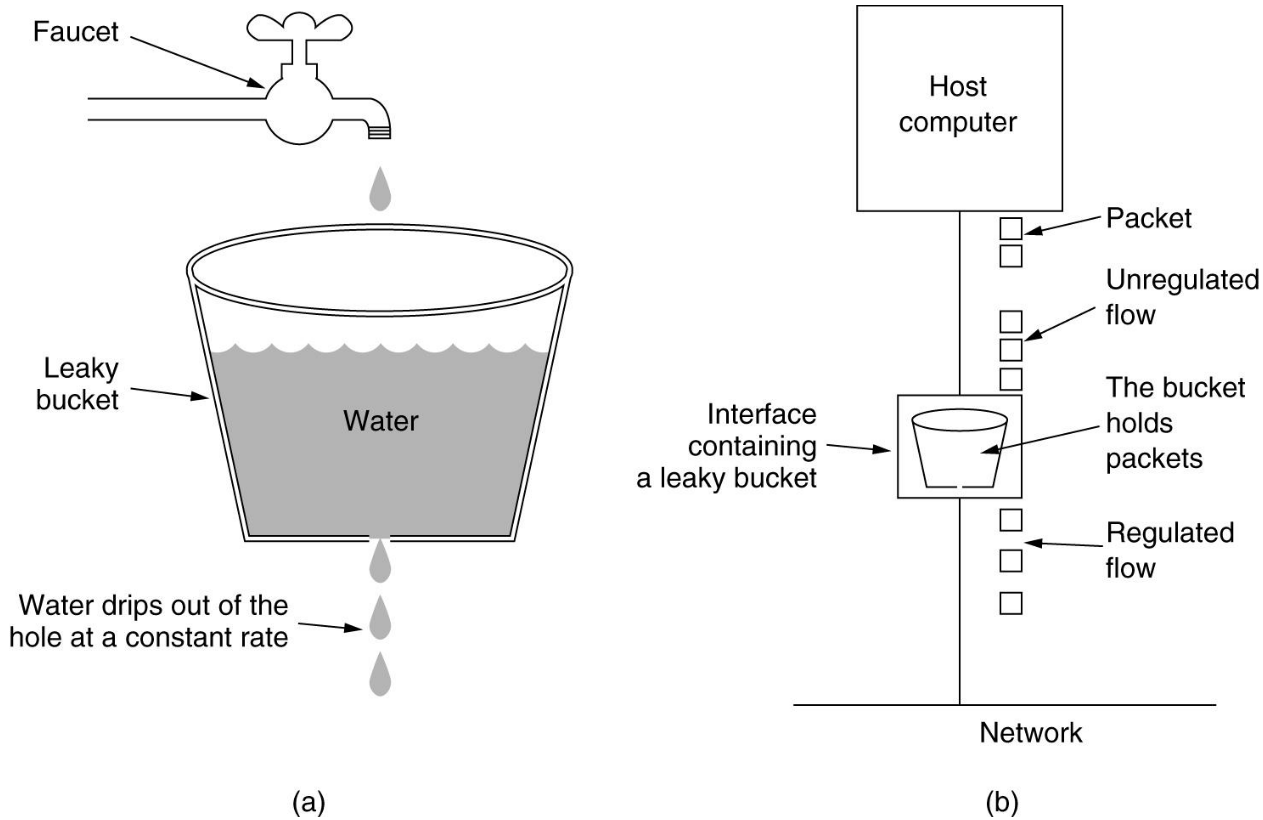


Leaky bucket
(need not full to send)



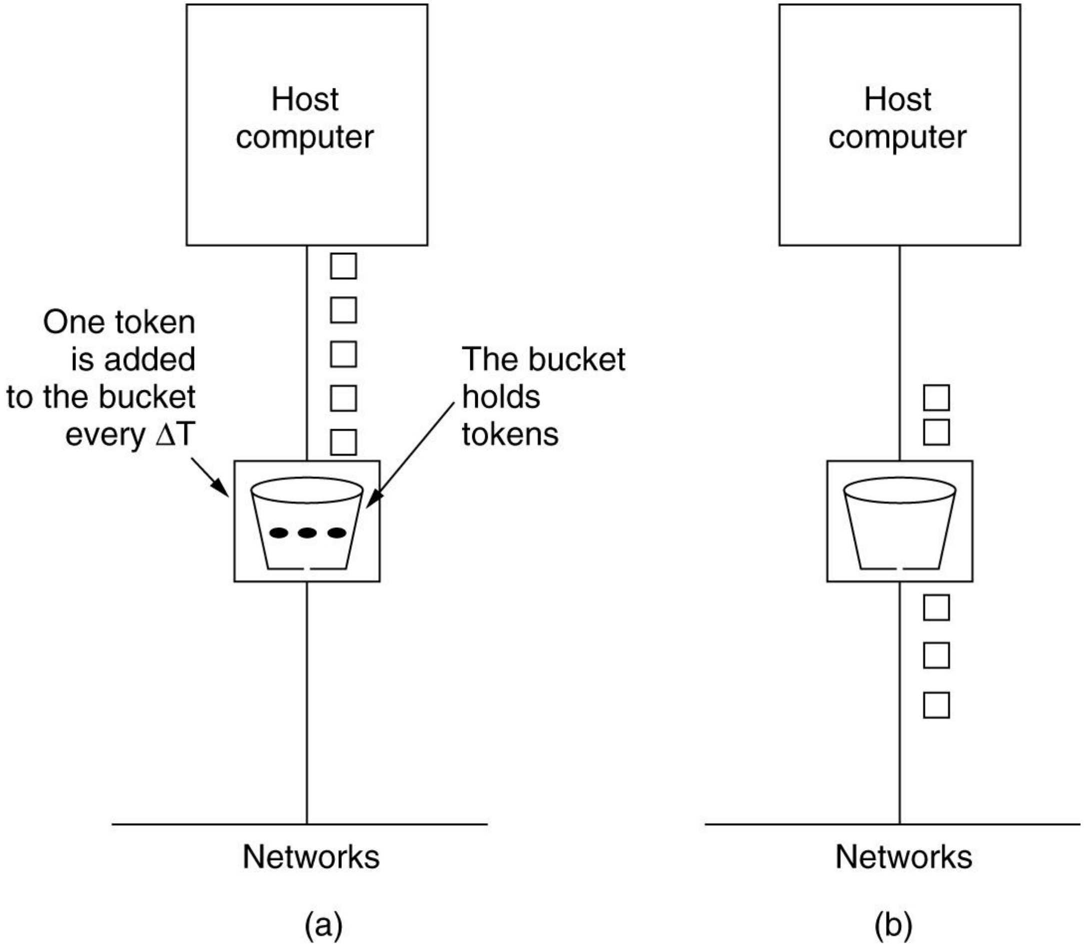
Token bucket
(need some water to send)

The Leaky Bucket Algorithm



(a) A leaky bucket with water. (b) a leaky bucket with packets.

The Token Bucket Algorithm

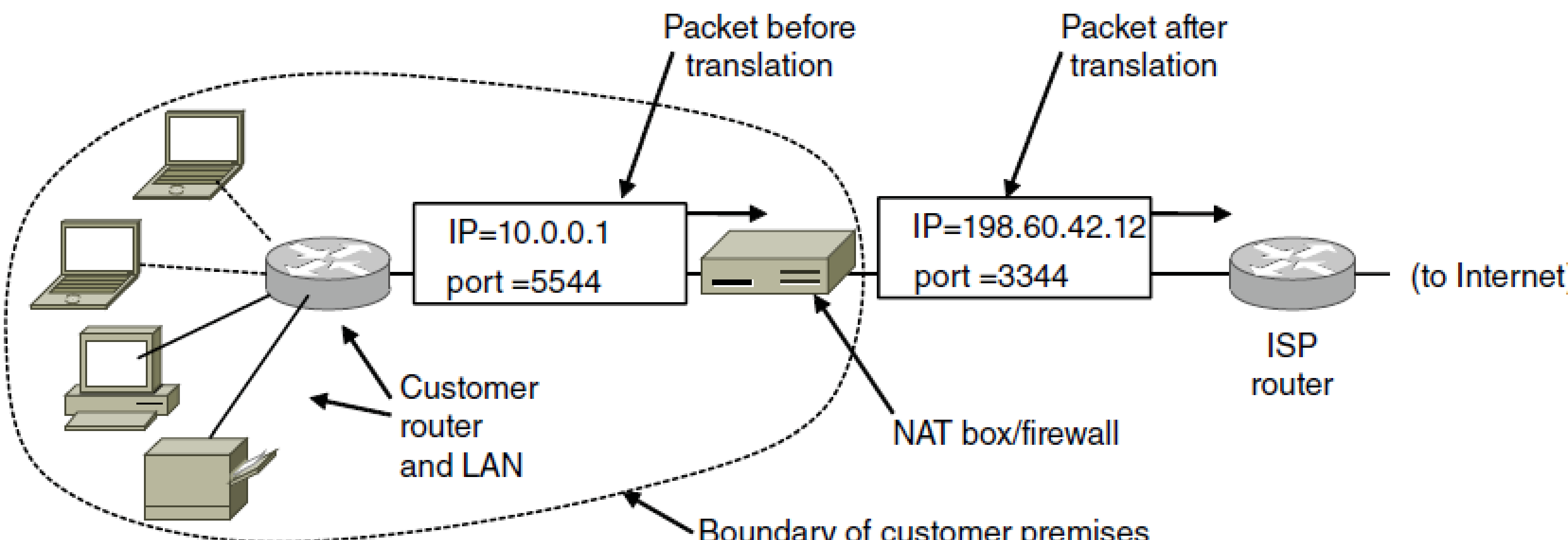


(a) Before. (b) After.

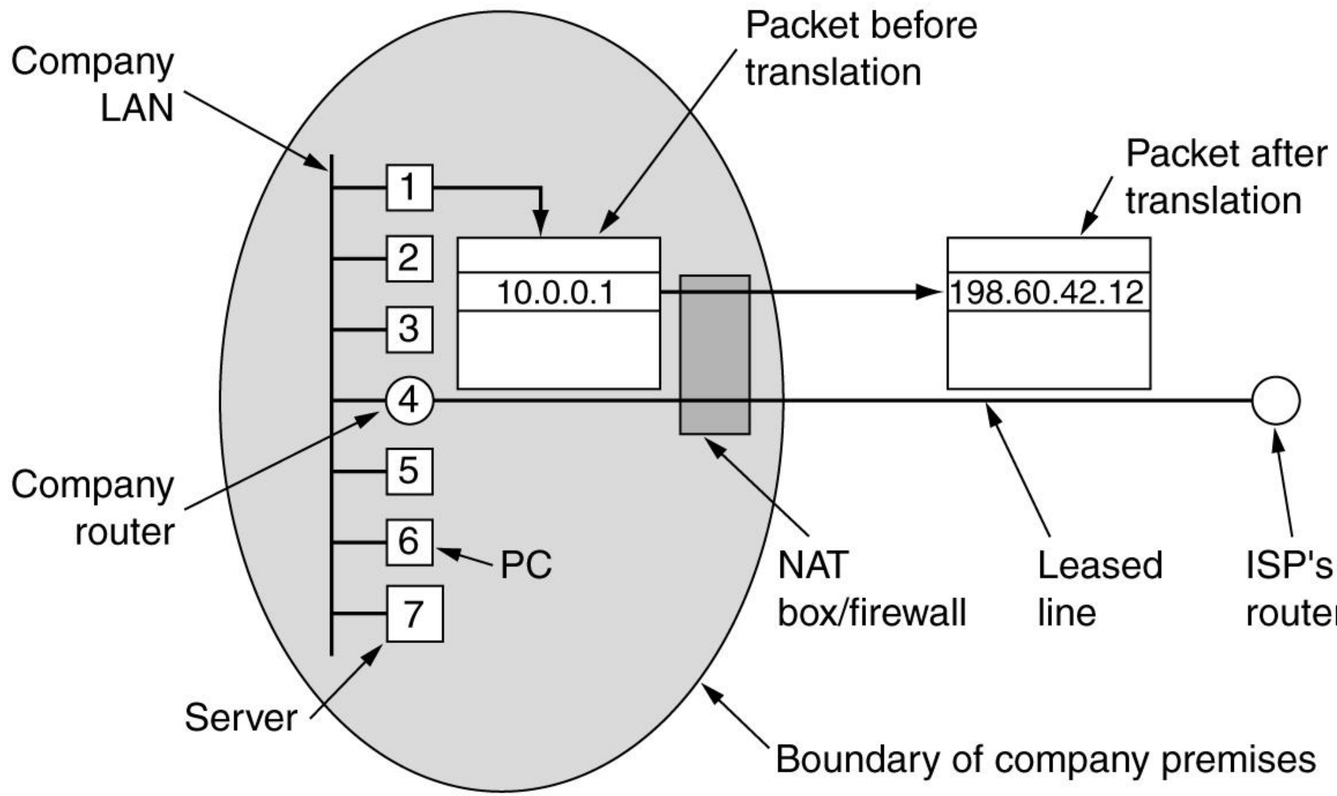
IP Addresses (6) – NAT

NAT (Network Address Translation) 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.



NAT – Network Address Translation



Internet Control Protocols (1)

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 Ethernet 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

What is IPv6

- Also known as IPng (next generation)
- A new version of the Internet Protocol
  - Primarily designed to extend address space
  - Enhancements and new features

IPv6 Packet Header

- Compare to IPv4 header

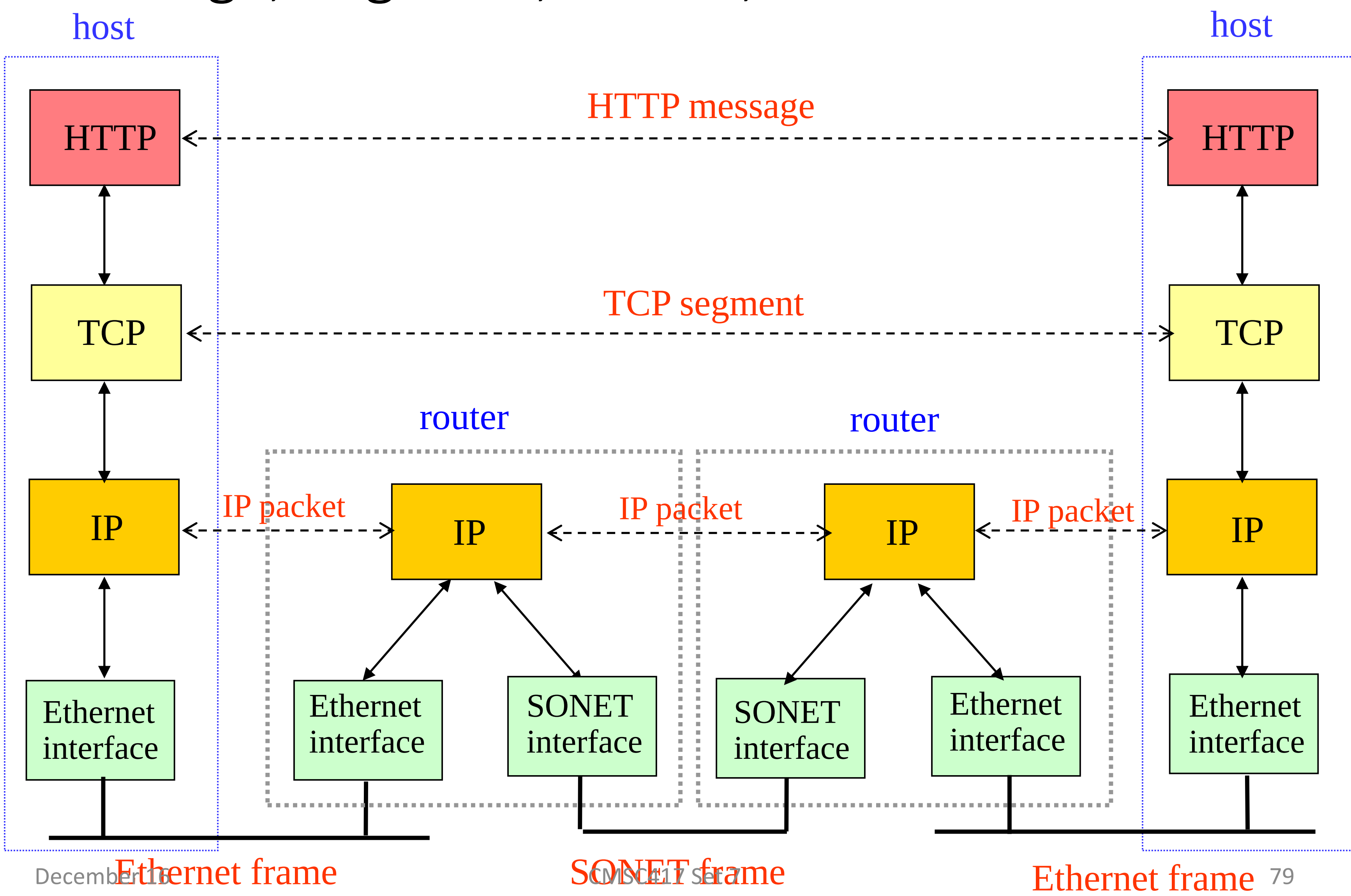
<table>
<tr><td colspan="10"></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>3</td><td>3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="4">Version=4</td><td colspan="4">IHL</td><td colspan="8">Type of Service</td><td colspan="16">Total Length</td></tr>
<tr><td colspan="16">Identifier</td><td colspan="3">Flags</td><td colspan="13">Fragment Offset</td></tr>
<tr><td colspan="8">Time to Live</td><td colspan="8">Protocol</td><td colspan="16">Header Checksum</td></tr>
<tr><td colspan="32">Source Address</td></tr>
<tr><td colspan="32">Destination Address</td></tr>
<tr><td colspan="32">Options + Padding</td></tr>
</table>

What Is BGP?

- Border Gateway Protocol BGP-4
- The de-facto interdomain routing protocol
- BGP enables policy in routing:
  - Which information gets advertised and how
- BGP is a Distance Vector like protocol
- Within an AS, Internal Gateway Protocol (IGP or I-BGP)

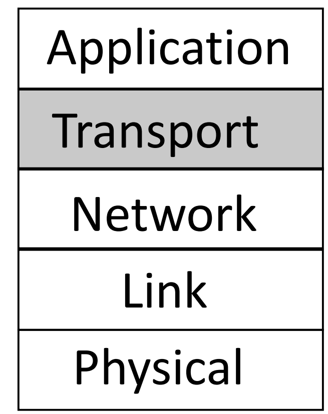
The Transport Layer

Message, Segment, Packet, and Frame



The Transport Layer

Responsible for delivering data across networks with the desired reliability or quality



Berkeley Sockets

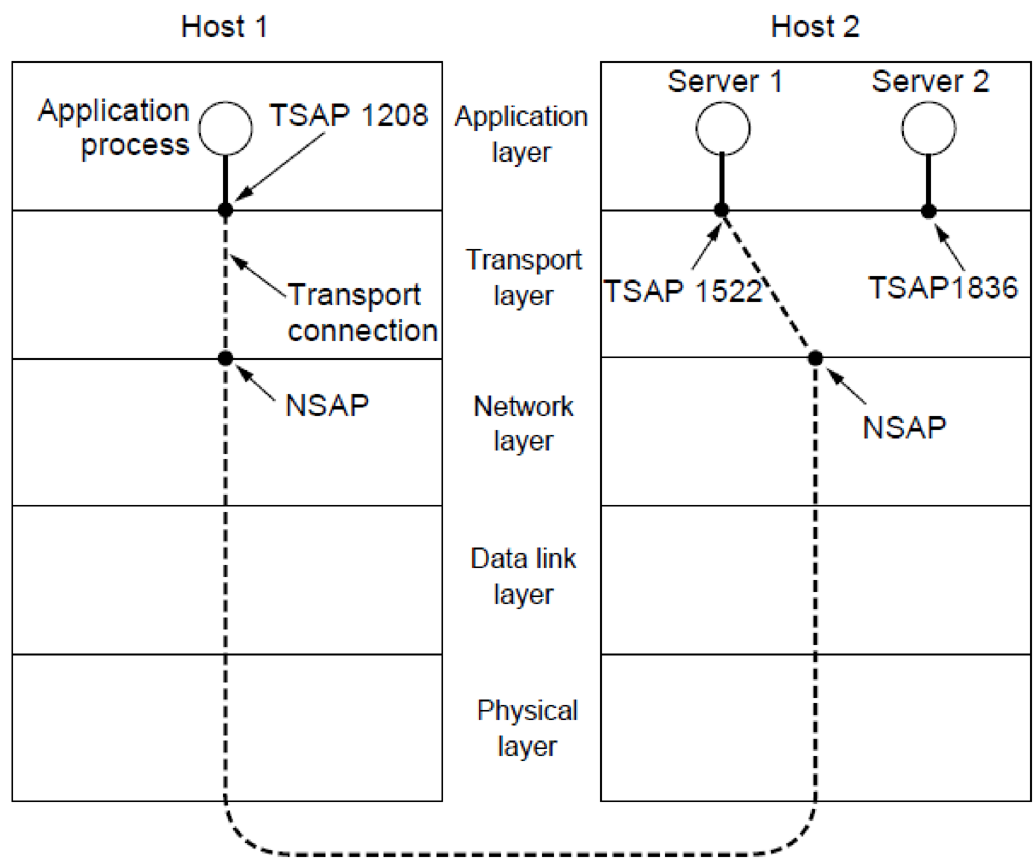
Very widely used primitives started with TCP on UNIX

- 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

Addressing

- Transport layer adds TSAPs
- Multiple clients and servers can run on a host with a single network (IP) address
- TSAPs are ports for TCP/UDP



Connection Establishment

- Connect
  - Expect confirmation that one and only one connection has been established
  - In a reasonable time
- Send Connection Request
- Expect Connection Accepted
- Segments (Packets) may be lost, corrupted, delayed, out of order, duplicated.

Sequence Numbers

- An integer number with finite number of bits
  - 8 bits => 256
  - 16 bits => 64K
- Roll Over
  - How long does it take?
  - Can there be old duplicates in the system?

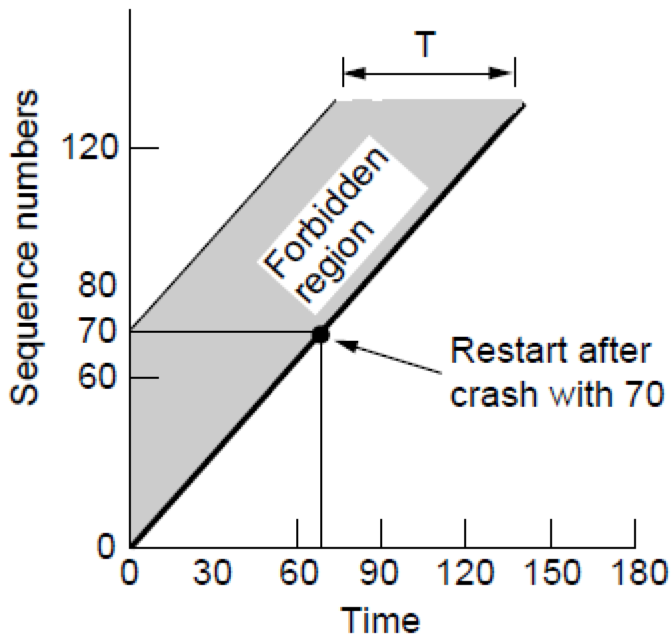
Approach:

- Don’t reuse sequence numbers within twice the MSL (Maximum Segment Lifetime) of 2T=240 secs
- Three-way handshake for establishing connection

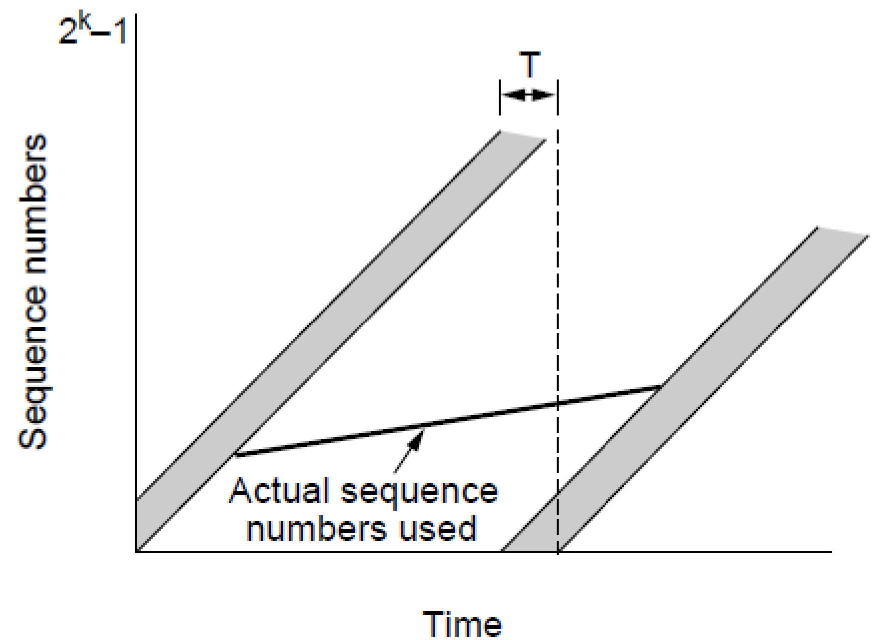
Connection Establishment (2)

Use a sequence number space large enough that it will not wrap, even when sending at full rate

- Clock (high bits) advances & keeps state over crash



Need seq. number not to wrap within T seconds

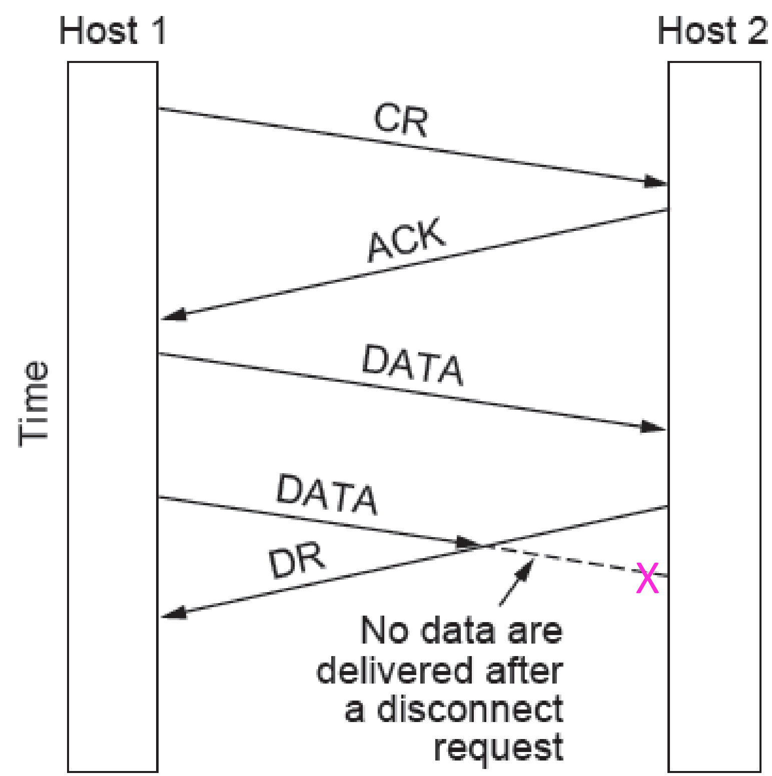


Need seq. number not to climb too slowly for too long

Connection Release (1)

Key problem is to ensure reliability while releasing

Asymmetric release (when one side breaks connection) is abrupt and may lose data



Flow Control

- Use Sliding Window
- Buffering
  - Sender buffers all TPDUs until acknowledged
  - TPDU lost by the network
    - Unreliable service
    - Receiver not having buffer
- How should buffers be managed
  - Dedicate
  - Acquire when needed
- Traffic
  - Low bandwidth, Bursty – buffer at sender – acquire at receiver
  - High bandwidth, smooth – Buffer at both ends
- Exchange Buffer information

Error Control and Flow Control (1)

Foundation for error control is a sliding window (from Link layer) with checksums and retransmissions

Flow control manages buffering at sender/receiver

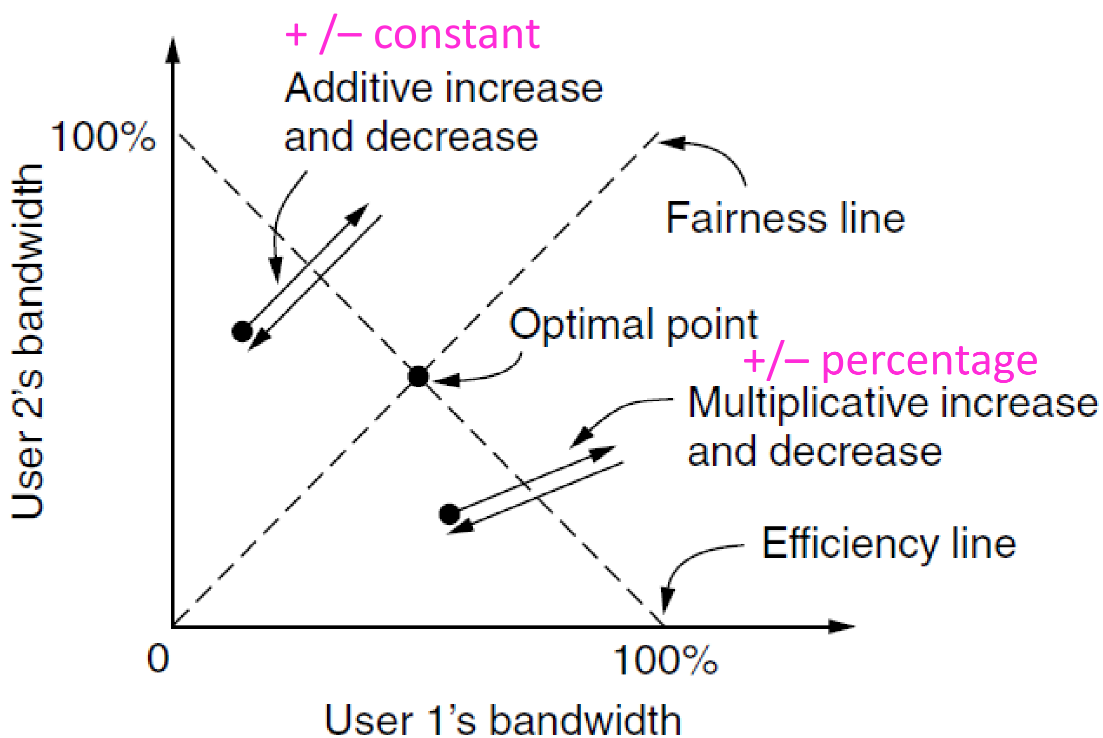
- Issue is that data goes to/from the network and applications at different times
- Window tells sender available buffering at receiver
- Makes a variable-size sliding window

Sliding Window Protocols

- Sliding Window concept »
- One-bit Sliding Window »
- Go-Back-N »
- Selective Repeat »

Regulating the Sending Rate (3)

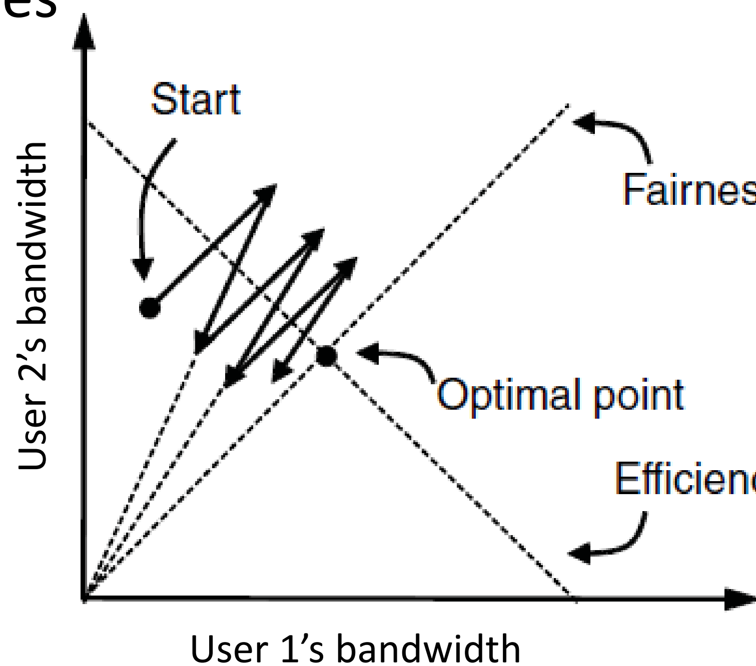
If two flows increase/decrease their bandwidth in the same way when the network signals free/busy they will not converge to a fair allocation



Regulating the Sending Rate (4)

The AIMD (Additive Increase Multiplicative Decrease) control law does converge to a fair and efficient point!

- TCP uses



Internet Protocols – UDP

- Introduction to UDP »
- Remote Procedure Call »
- Real-Time Transport »

Introduction to UDP (1)

UDP (User Datagram Protocol) is a shim over IP

- Header has ports (TSAPs), length and checksum.

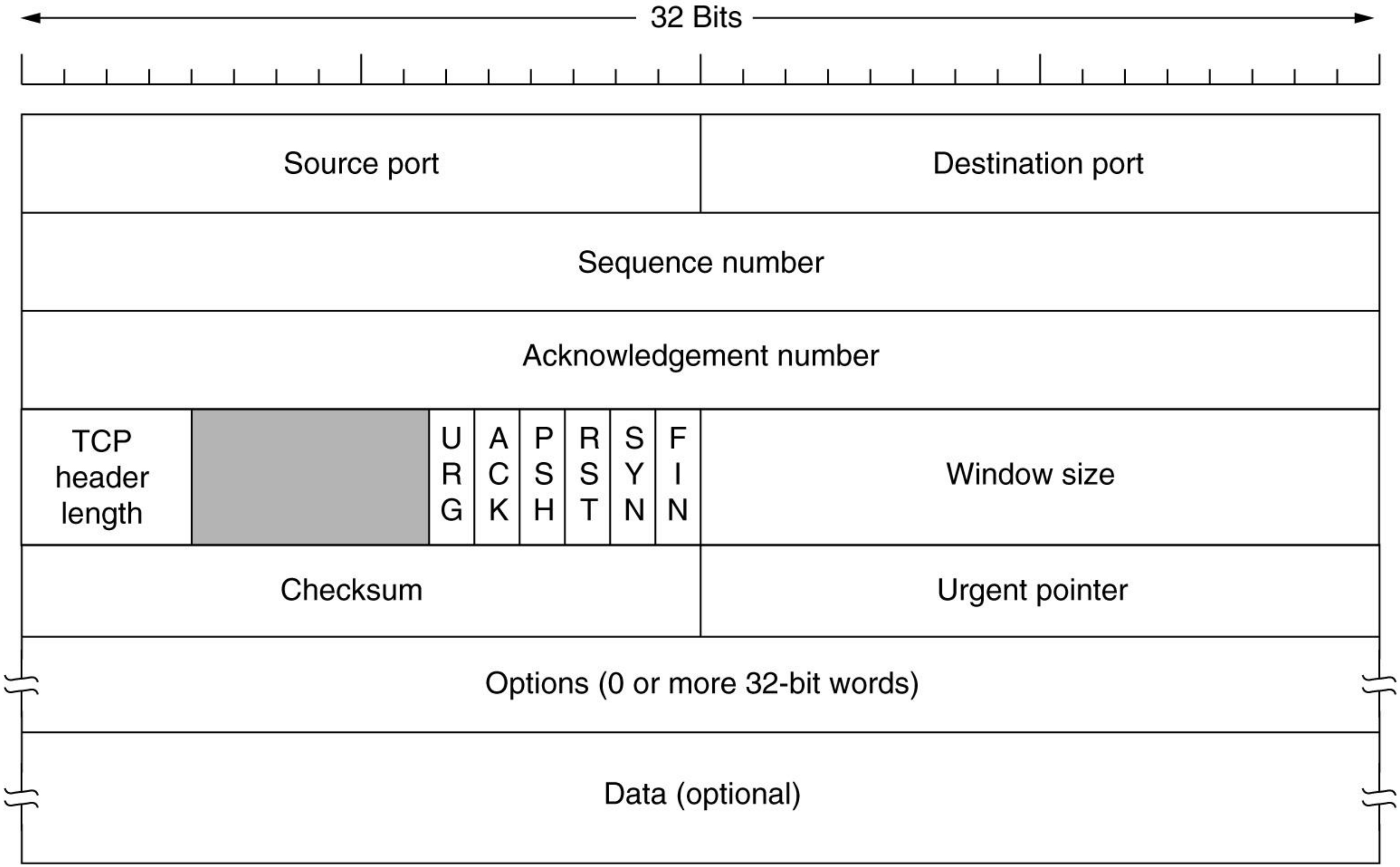
32 Bits

Source port	Destination port
UDP length	UDP checksum

Internet Protocols – TCP

- The TCP service model »
- The TCP segment header »
- TCP connection establishment »
- TCP connection state modeling »
- TCP sliding window »
- TCP timer management »
- TCP congestion control »

The TCP Segment Header



TCP Header.

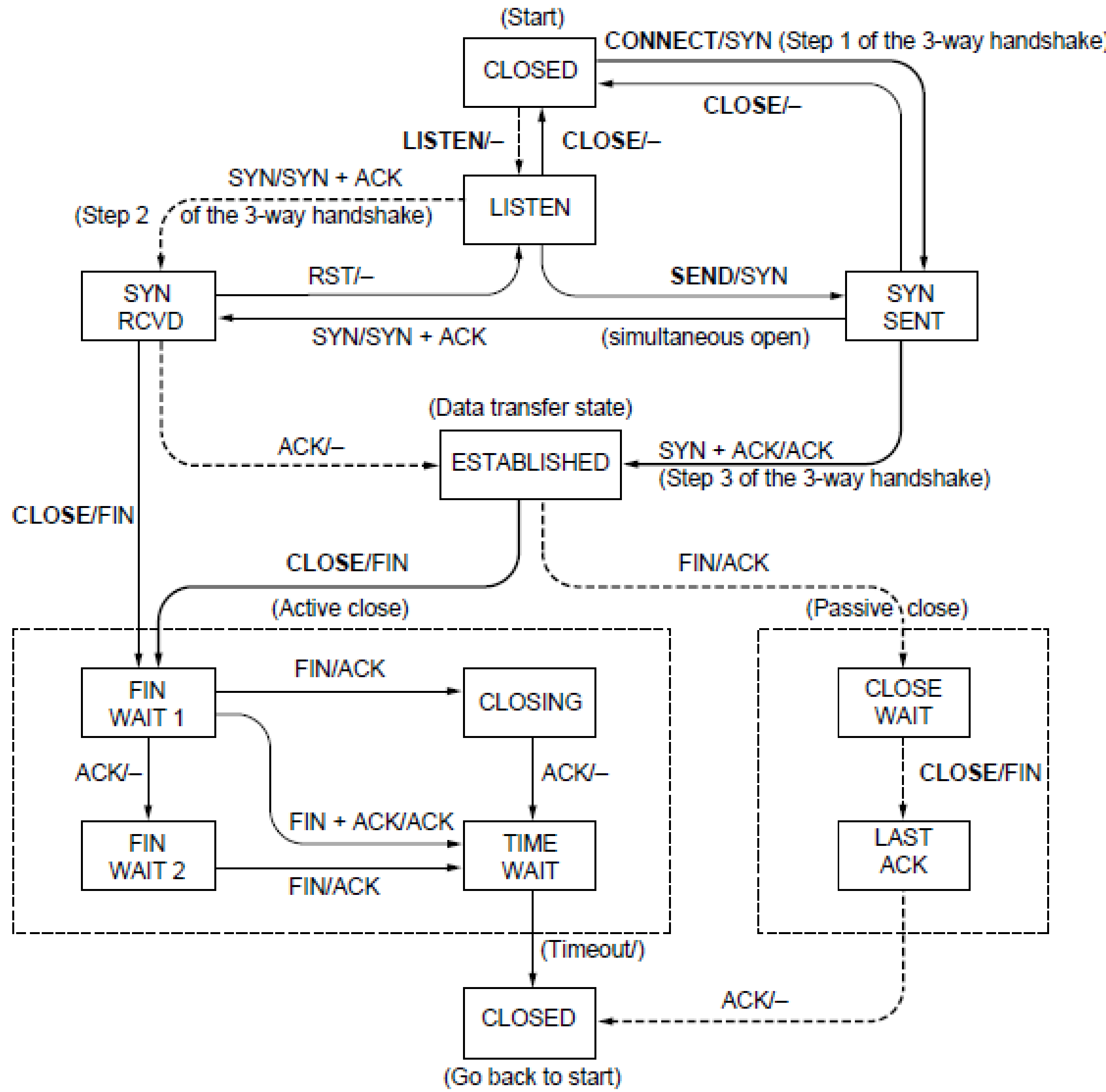
TCP Connection State Modeling (2)

Solid line is the normal path for a client.

Dashed line is the normal path for a server.

Light lines are unusual events.

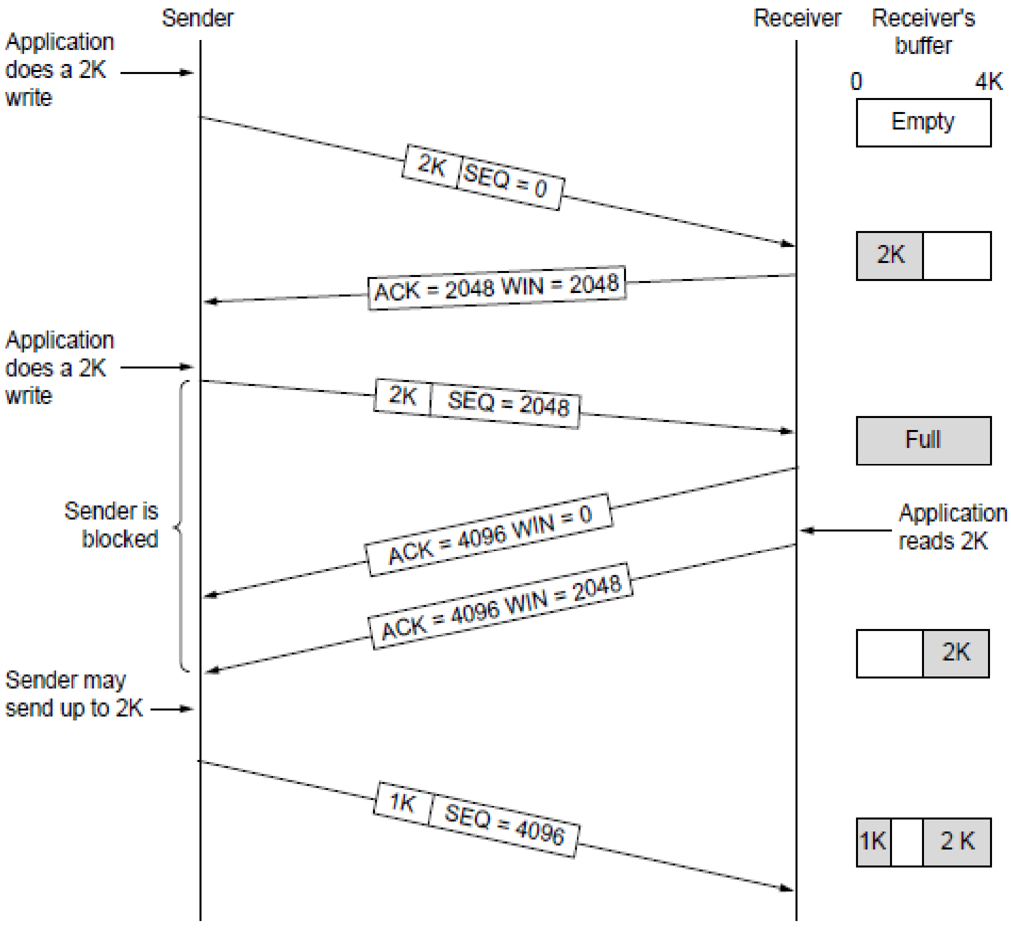
Transitions are labeled by the cause and action, separated by a slash.



TCP Sliding Window (1)

TCP adds flow control to the sliding window as before

- ACK + WIN is the sender's limit



Handling Silly Window Problem

- Delay ack and window updates for 500 ms.
- Nagle's Algorithm
  - When data comes in one byte at a time
    - Send the first byte and buffer the rest till the outstanding byte is acknowledged
    - Then send all the buffered characters in one TCP segment
  - Mouse movements have to be sent – Burst does not work well.
- Clark's Solution
  - Wait until decent amount of space available then advertise
    - Max segment size or half buffer
  - Sender not send tiny segments

Timer Management

- TCP uses multiple timers
  - Most important is the *retransmission timer*
  - What value to set it at??
- Round Trip time
  - Highly variable
  - Varies over time
  - Have to track it
  - Estimate it
  - M is a new measurement
  - α = 7/8
  - Use βRTT for retransmission timer
  - Initial values of β were 2 – make is proportional to standard deviation of M

$$RTT = \alpha RTT + (1 - \alpha) M$$

Persistence Timer

- Receiver sends a window of 0
- Later sends a window size but that packet is lost
  - Both wait
- Persistence Timer
  - When it goes off – sender sends a probe request to receiver to get a window size
  - If still zero – continue to wait and reset persistence timer
- Keepalive Timer
  - When a connection is idle for a long time – check if the other side is still there

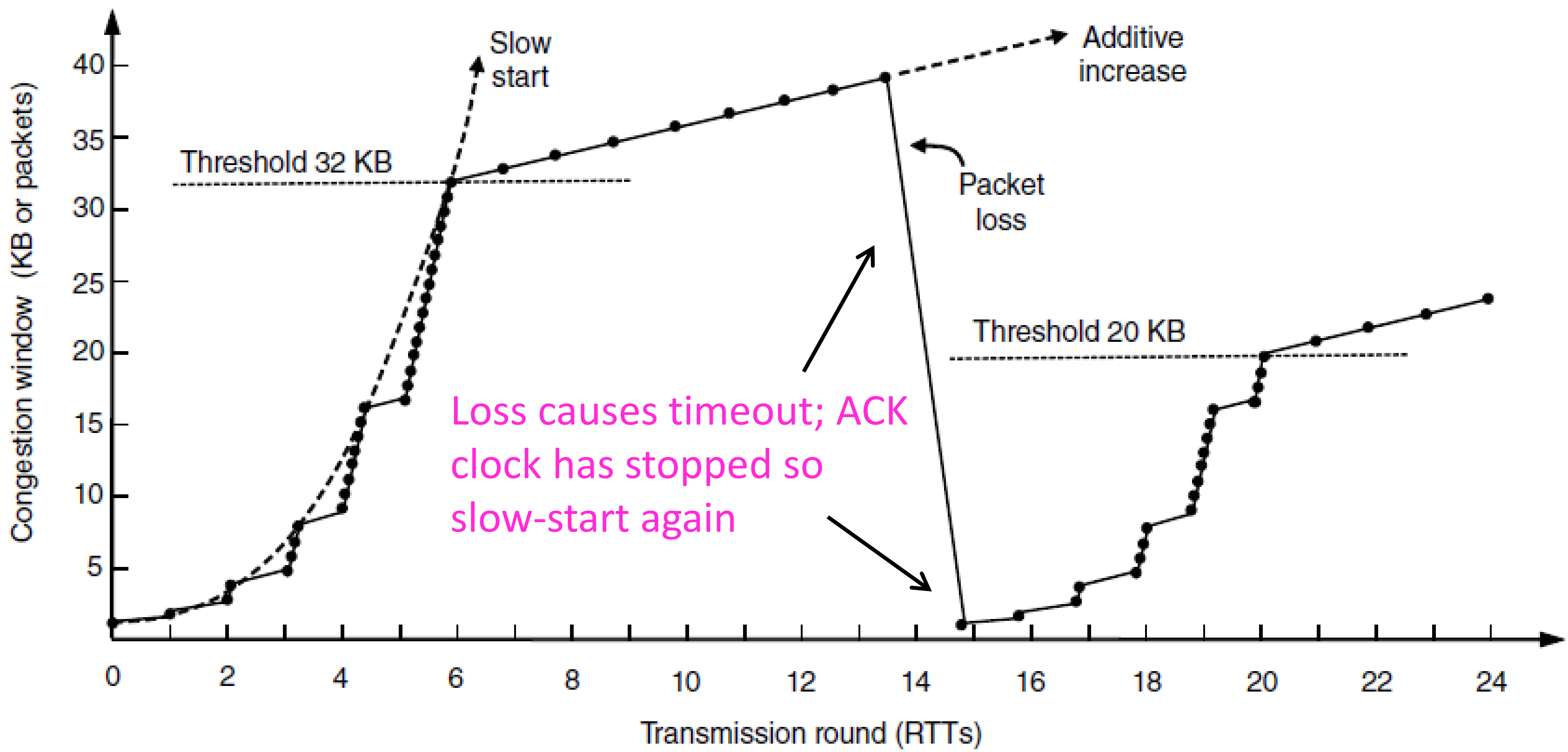
TCP Congestion Control

- Congestion – a function of total number of packets in the network, and where they are
- First step – detection
  - Is packet loss an indication of congestion??
  - All TCP algorithms assume timeouts are caused by congestion
- Initial steps
  - When connection is established – use suitable window size
    - Loss will not occur due to buffers at receiver
- Two issues
  - Network Capacity
  - Receiver Capacity

TCP Congestion Control (5)

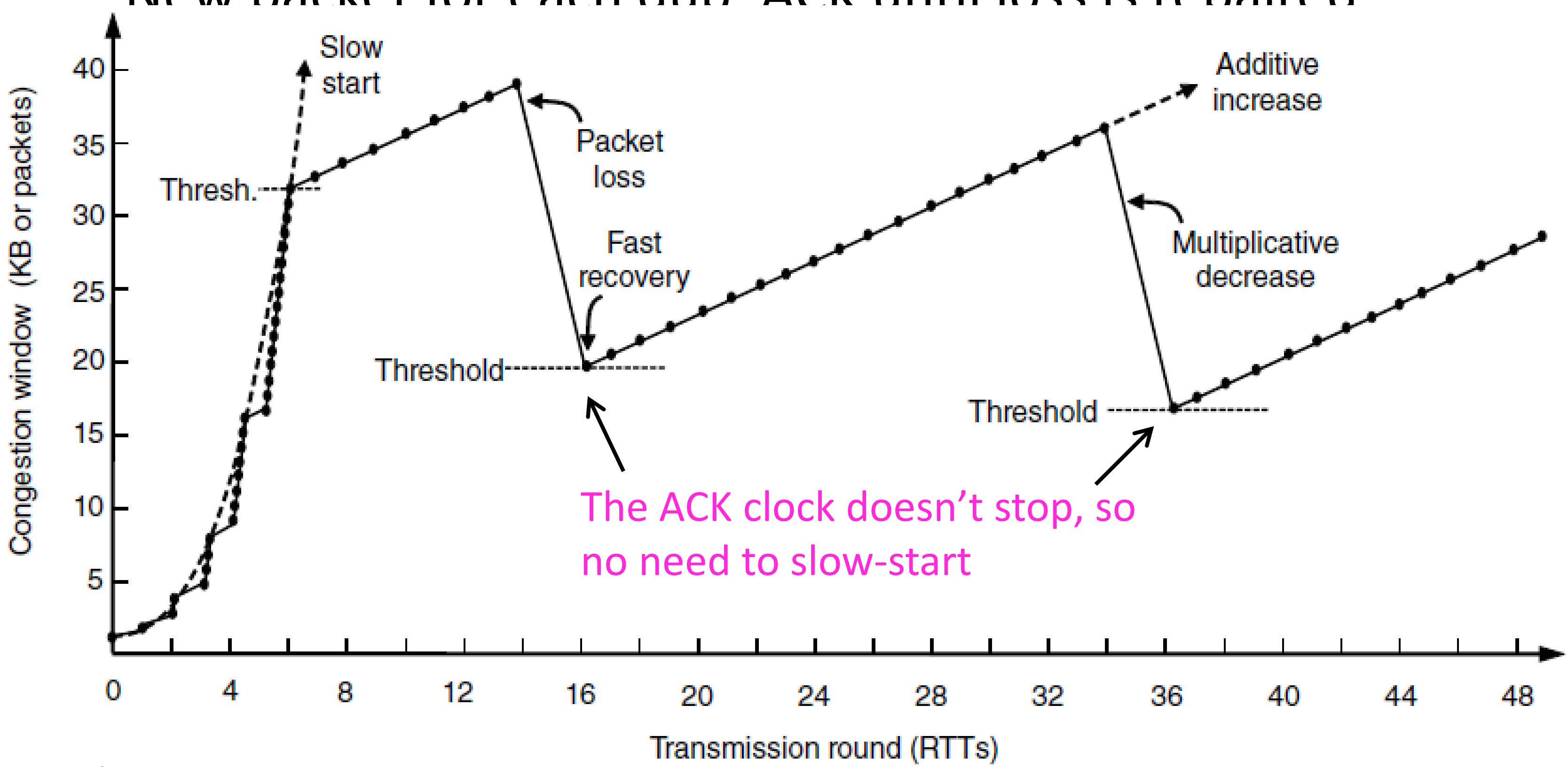
Slow start followed by additive increase (TCP Tahoe)

- Threshold is half of previous loss cwnd



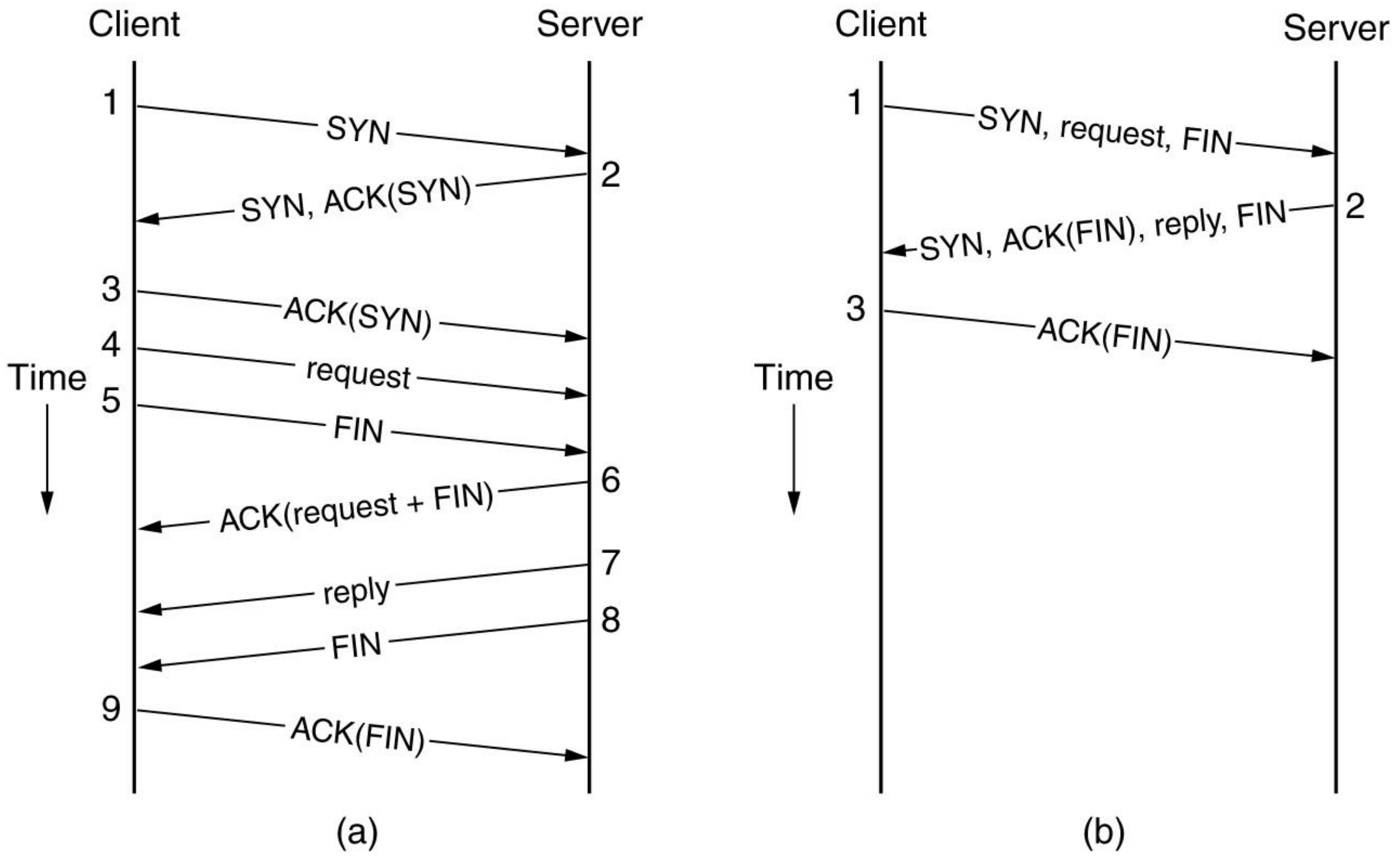
TCP Congestion Control (6)

- With fast recovery, we get the classic sawtooth (TCP Reno)
  - Retransmit lost packet after 3 duplicate ACKs
  - New packet for each dup. ACK until loss is repaired



The ACK clock doesn't stop, so no need to slow-start

Transactional TCP



(a) RPC using normal TPC.
(b) RPC using T/TCP.

Domain Name System (DNS)

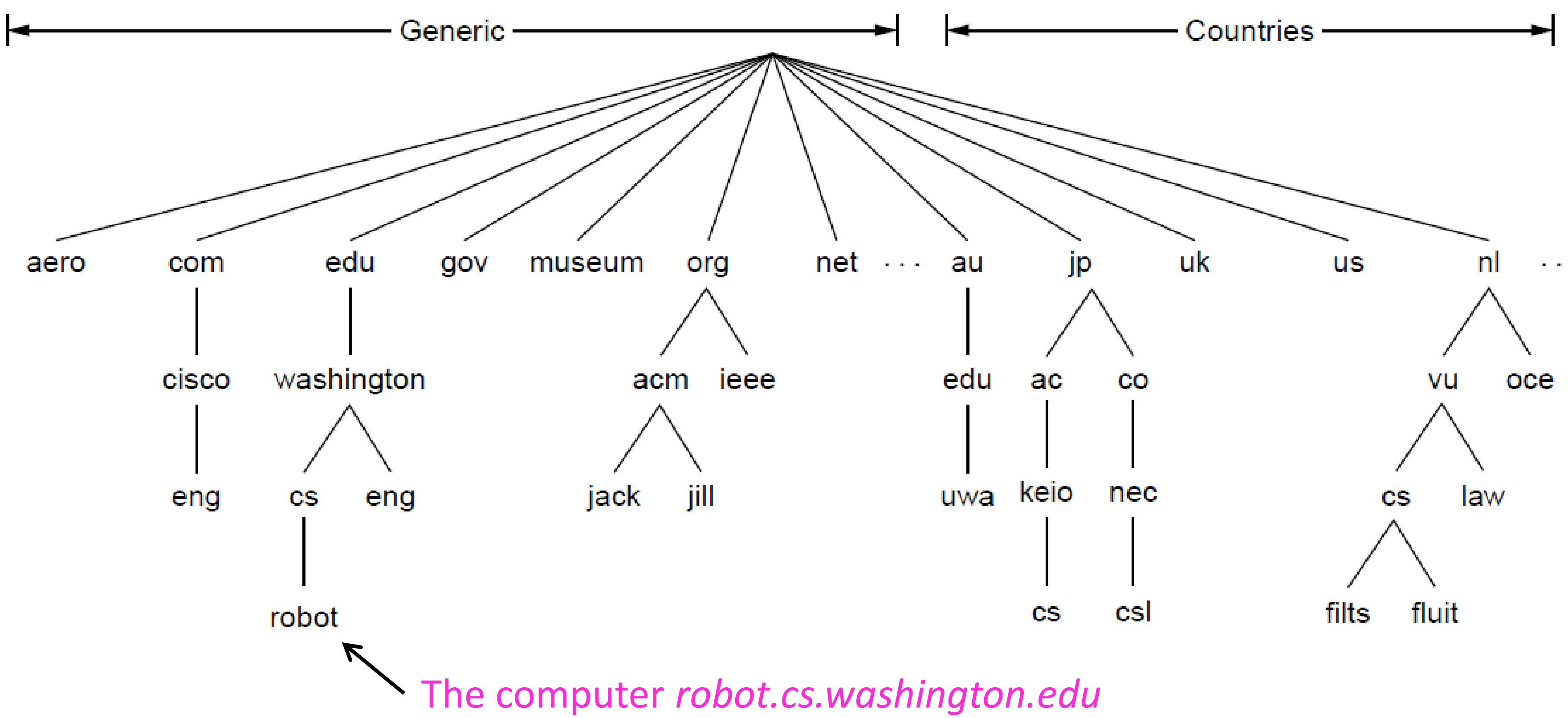
Host Names vs. IP addresses

- Host names
  - Mnemonic name appreciated by humans
  - Variable length, alpha-numeric characters
  - Provide little (if any) information about location
  - Examples: www.cnn.com and ftp.eurocom.fr
- IP addresses
  - Numerical address appreciated by routers
  - Fixed length, binary number
  - Hierarchical, related to host location
  - Examples: 64.236.16.20 and 193.30.227.161

The DNS Name Space (1)

DNS namespace is hierarchical from the root down

- Different parts delegated to different organizations



Resource Record

Domain Name Time-to-live Class Type Value

- Domain Name – Domain to which this record applies
- Time-To-Live – Indication of how stable the record is – 86400 – seconds/day
- Class – For Internet information it is always IN
- Type –
- Value -

DNS Resource Records

DNS: distributed db storing resource records (RR)

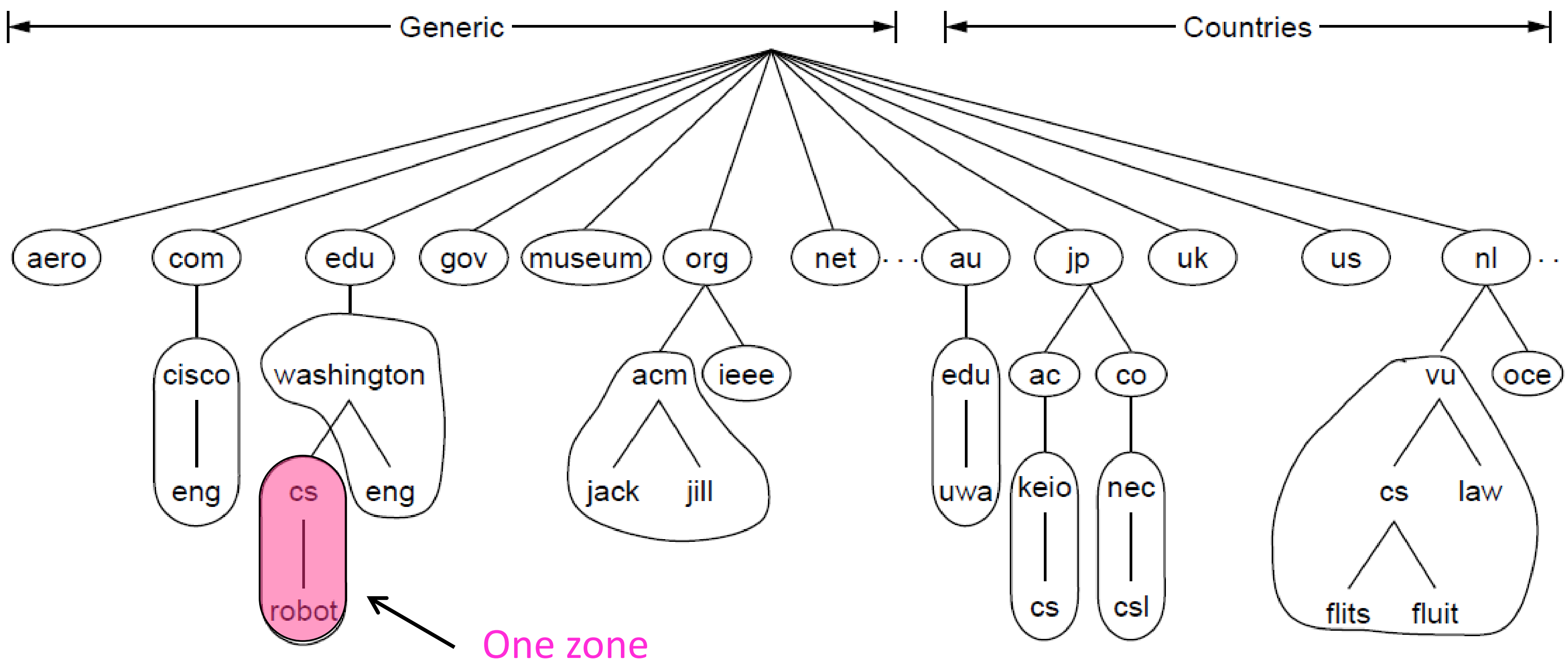
RR format: (name, value, type, ttl)

- Type=A
  - name is hostname
  - value is IP address
- Type=NS
  - **`name`** is domain (e.g. foo.com)
  - **`value`** is hostname of authoritative name server for this domain
- Type=CNAME
  - name is alias name for some "canonical" (the real) name

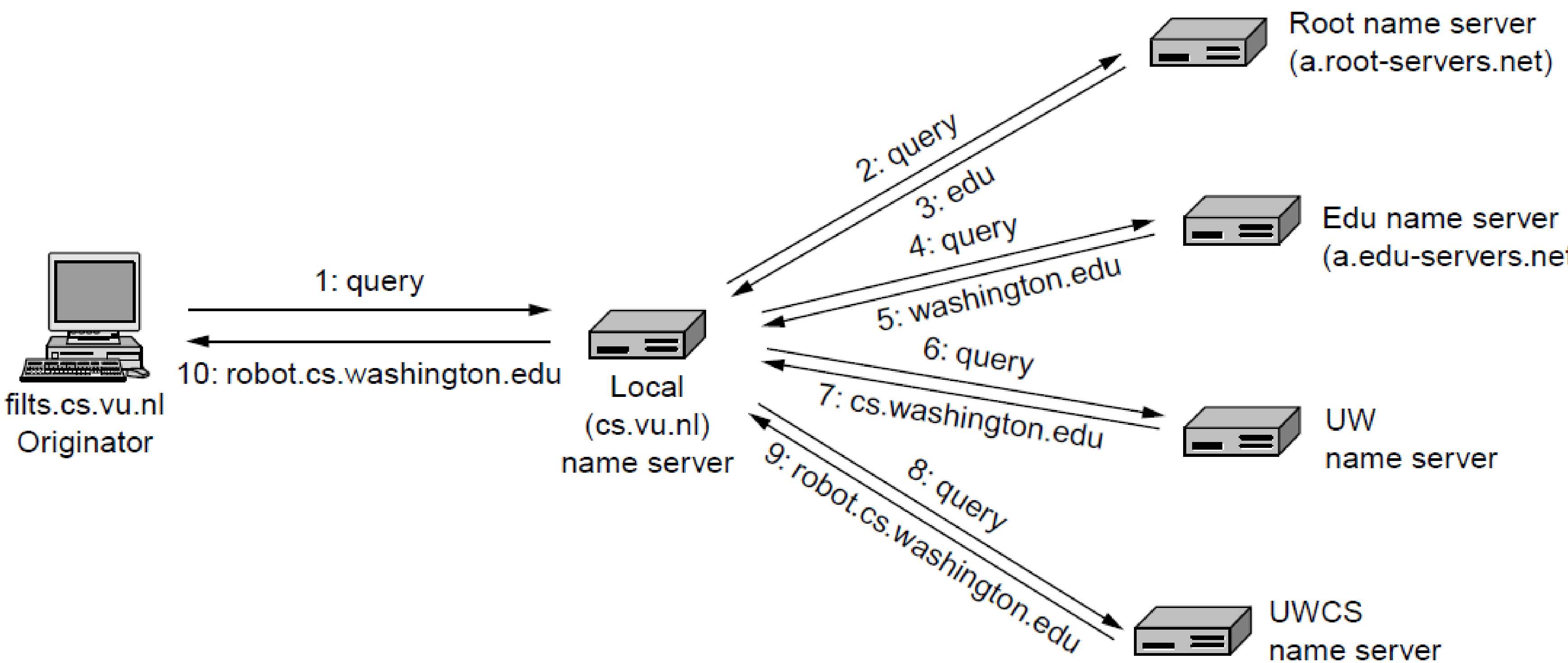
    www.ibm.com is really servereast.backup2.ibm.com
  - value is canonical name
- Type=MX
  - value is name of mailserver associated with name

Name Servers (1)

Name servers contain data for portions of the name space called zones (circled).



Name Servers (3)



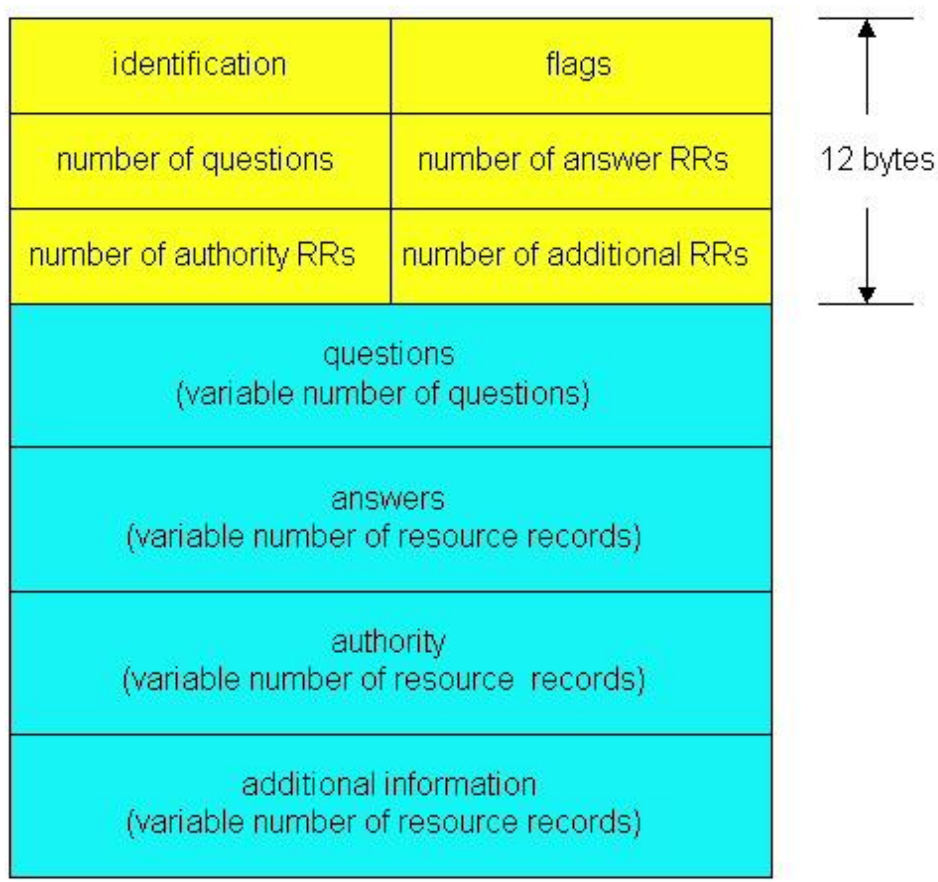
Example of a computer looking up the IP for a name

DNS Protocol

DNS protocol : *query* and *reply* messages, both with same *message format*

Message header

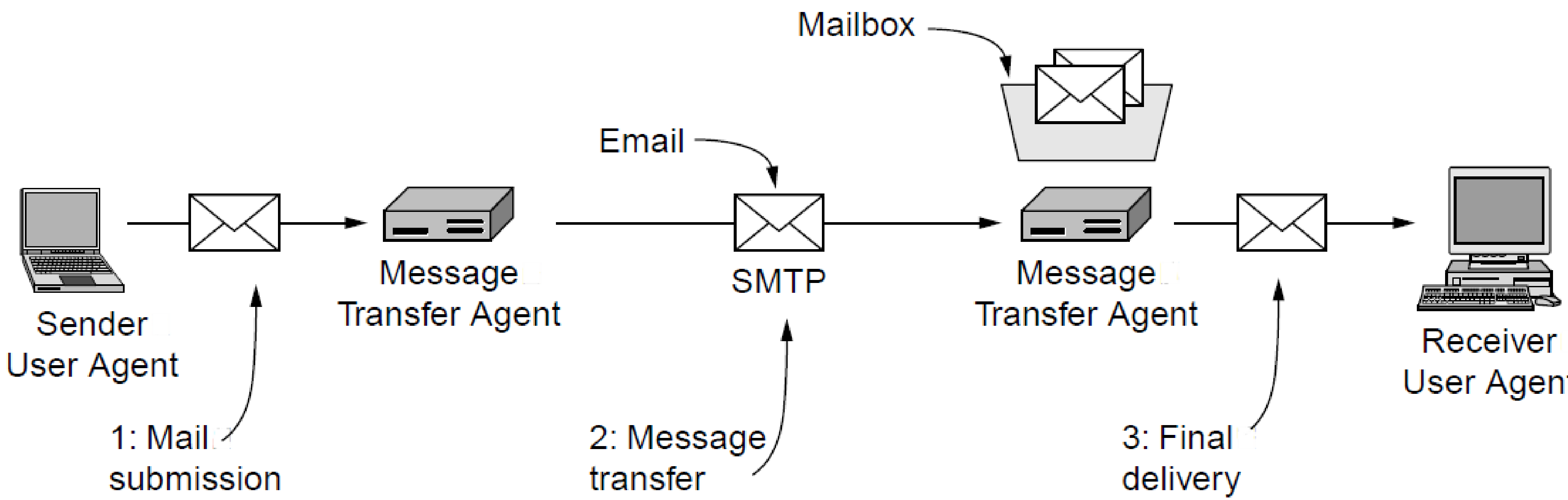
- Identification: 16 bit # for query, reply to query uses same #
- Flags:
  - Query or reply
  - Recursion desired
  - Recursion available
  - Reply is authoritative



Electronic Mail

Architecture and Services (1)

- The key components and steps (numbered) to send email



Architecture of the email system

MIME – Multipurpose Internet Mail Extensions

Problems with international languages:

- Languages with accents (French, German).
- Languages in non-Latin alphabets (Hebrew, Russian).
- Languages without alphabets (Chinese, Japanese).
- Messages not containing text at all (audio or images).

Example With Received Header

Return-Path: <casado@cs.stanford.edu>
Received: from ribavirin.CS.Princeton.EDU (ribavirin.CS.Princeton.EDU [128.112.136.44])
by newark.CS.Princeton.EDU (8.12.11/8.12.11) with SMTP id k04M5R7Y023164
for <jrex@newark.CS.Princeton.EDU>; Wed, 4 Jan 2006 17:05:37 -0500 (EST)
Received: from bluebox.CS.Princeton.EDU ([128.112.136.38])
by ribavirin.CS.Princeton.EDU (SMSSMTP 4.1.0.19) with SMTP id M2006010417053607946
for <jrex@newark.CS.Princeton.EDU>; Wed, 04 Jan 2006 17:05:36 -0500
Received: from smtp-roam.Stanford.EDU (smtp-roam.Stanford.EDU [171.64.10.152])
by bluebox.CS.Princeton.EDU (8.12.11/8.12.11) with ESMTP id k04M5XNQ005204
for <jrex@cs.princeton.edu>; Wed, 4 Jan 2006 17:05:35 -0500 (EST)
Received: from [192.168.1.101] (adsl-69-107-78-147.dsl.pltn13.pacbell.net [69.107.78.147])
(authenticated bits=0)
by smtp-roam.Stanford.EDU (8.12.11/8.12.11) with ESMTP id k04M5W92018875
(version=TLSv1/SSLv3 cipher=DHE-RSA-AES256-SHA bits=256 verify=NOT);
Wed, 4 Jan 2006 14:05:32 -0800
Message-ID: <43BC46AF.3030306@cs.stanford.edu>
Date: Wed, 04 Jan 2006 14:05:35 -0800
From: Martin Casado <casado@cs.stanford.edu>
User-Agent: Mozilla Thunderbird 1.0 (Windows/20041206)
MIME-Version: 1.0
To: jrex@CS.Princeton.EDU
CC: Martin Casado <casado@cs.stanford.edu>
Subject: Using VNS in Class
Content-Type: text/plain; charset=ISO-8859-1; format=flowed
Content-Transfer-Encoding: 7bit

Post Office Protocol (POP)

- POP goals
  - Support users with intermittent network connectivity
  - Allow them to retrieve e-mail messages when connected
  - … and view/manipulate messages when disconnected
- Typical user-agent interaction with a POP server
  - Connect to the server
  - Retrieve all e-mail messages
  - Store messages on the user's PCs as new messages
  - Delete the messages from the server
  - Disconnect from the server
- User agent still uses SMTP to send messages

IMAP

A comparison of POP3 and IMAP.

Feature	POP3	IMAP
Where is protocol defined?	RFC 1939	RFC 2060
Which TCP port is used?	110	143
Where is e-mail stored?	User's PC	Server
Where is e-mail read?	Off-line	On-line
Connect time required?	Little	Much
Use of server resources?	Minimal	Extensive
Multiple mailboxes?	No	Yes
Who backs up mailboxes?	User	ISP
Good for mobile users?	No	Yes
User control over downloading?	Little	Great
Partial message downloads?	No	Yes
Are disk quotas a problem?	No	Could be in time
Simple to implement?	Yes	No
Widespread support?	Yes	Growing