Outline

- Introduction
- Applications
- Transport Layer
- Network Layer
- LANs and WLANs
- Multimedia Networking

Reference Models

- OSI model: 7-layer model; application, presentation, session, transport, network, data link, and physical.
- TCP/IP model: 4-layers; application, transport, Internet, and Host-to-Network.
- Internet Protocol Stack – a *hybrid* of TCP/IP and OSI reference models

Layers and Protocols?

- Layered network architecture … why?
  - Reduces design complexity
- The purpose of a layer?
  - Layer-N peers converse with each other using *protocols*; each layer provides functionality to the a higher layer;
- Protocols?
  - A set of rules governing the format and meaning of messages exchanges by peer entities within a layer

Protocols, Layers, and PDU's

Protocol	Layer	PDU
HTTP, FTP	Application layer	message
TCP, UDP	Transport layer	segment
ICMP, IP	Network layer	datagram
PPP, Ethernet, IEEE 802.11	Data link layer	frame

- Assuming the Internet Protocol Stack.

More on Layers and Protocols

Protocol	Layer	Imm. Lower Layer
HTTP, FTP	Application	Transport
TCP, UDP	Transport	Network
ICMP, IP	Network	Data Link
PPP, Ethernet, IEEE 802.11	Data Link	Physical

- Assuming the Internet Protocol Stack.

Internetworks

- A collection of interconnected networks is called an "internetwork" or an "internet". Internet is one example of a really big internetwork.

- Internet structure: Tier-1 ISPs, Tier-2 ISPs, Tier-3 ISPs, NAP, POP, etc.

Network Core

- Packet switching – a technique for transmission of packets that allows multiple end systems to share a "route"
  - Virtual circuit vs. datagram networks
- Circuit switching – a technique that requires end-to-end resource reservation for a "call"
  - TDMA, FDMA
- Circuit vs. Message vs. Packet Switching

Delay/Loss in Packet Switched Networks

- Queuing delay and packet loss
- Transmission delay
- Propagation delay

Outline

- Introduction
- Applications
- Transport Layer
- Network Layer
- LANs and WLANs
- Multimedia Networking
- Questions

Application Layer

- Processes communicating across networks
- What is HTTP?
  - Hypertext Transfer Protocol – the Web's application layer protocol
  - HTTP 1.0, HTTP 1.1
  - Pipelined – requests sent as soon as it is encountered
  - Persistent – multiple objects can be sent over a single TCP connection between the server and the client

HTTP Continued …

- HTTP headers
  - How is the end of an object determined in HTTP/1.1?
- HTTP methods
  - GET, HEAD, POST, PUT …
- HTTP response codes
  - 1xx, 2xx, 3xx, 4xx, 5xx

The World Wide Web

- What is a Web Proxy?
- What is a caching hierarchy?
- Caching issues –
  - Cache consistency issues
    - The "conditional HTTP GET" request
  - Cache replacement issues
  - Prefetching
- Cookies – a means to maintain state in stateless HTTP servers

DNS: Domain Name System

Internet hosts:

- IP address (32 bit) - used for addressing datagrams
- "name", e.g., ww.yahoo.com - used by humans

DNS: provides translation between host name and IP address

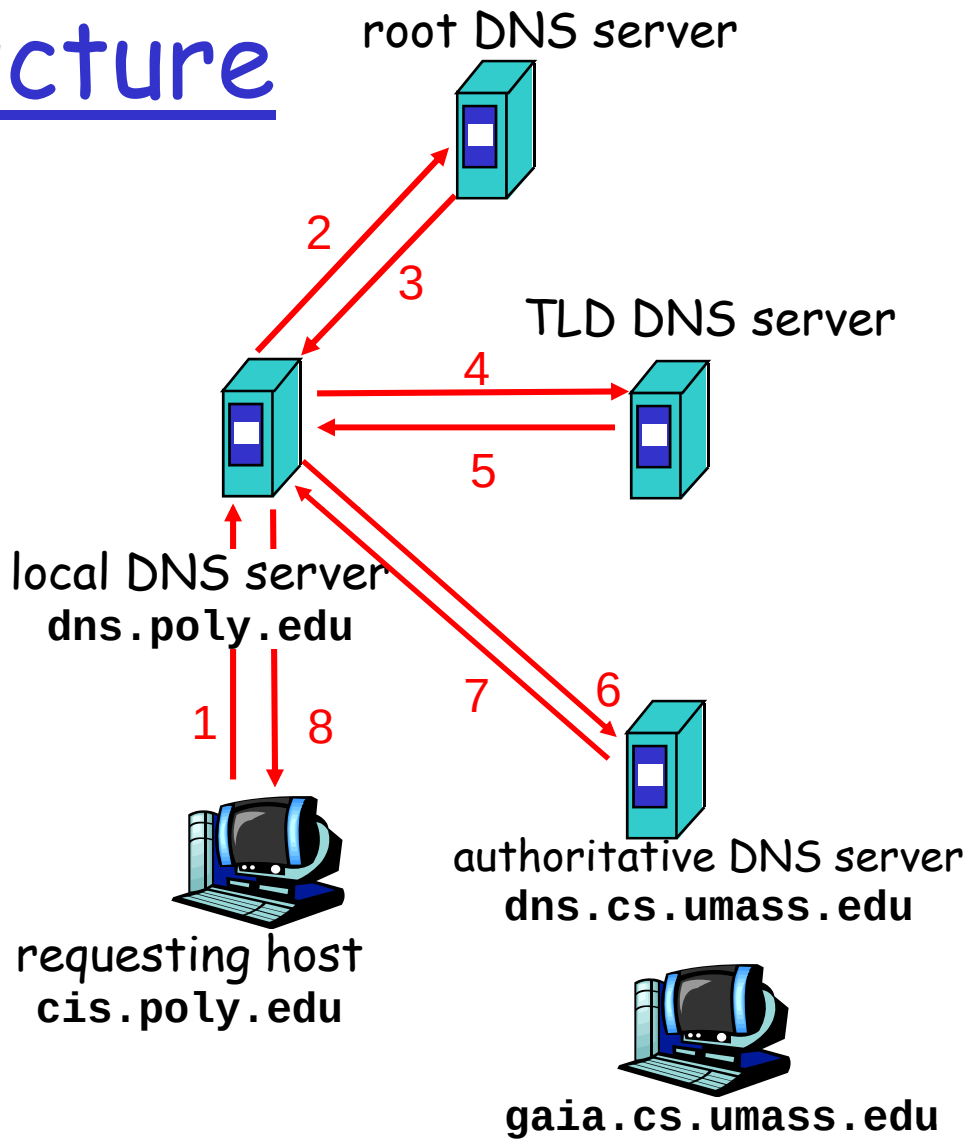
- *distributed database* implemented in hierarchy of many *name servers*
- *Distributed for scalability & reliability*

DNS Services

- Hostname to IP address translation
- Host aliasing
  - Canonical and alias names
- Mail server aliasing
- Load distribution
  - Replicated Web servers: set of IP addresses for one canonical name

DNS Infrastructure

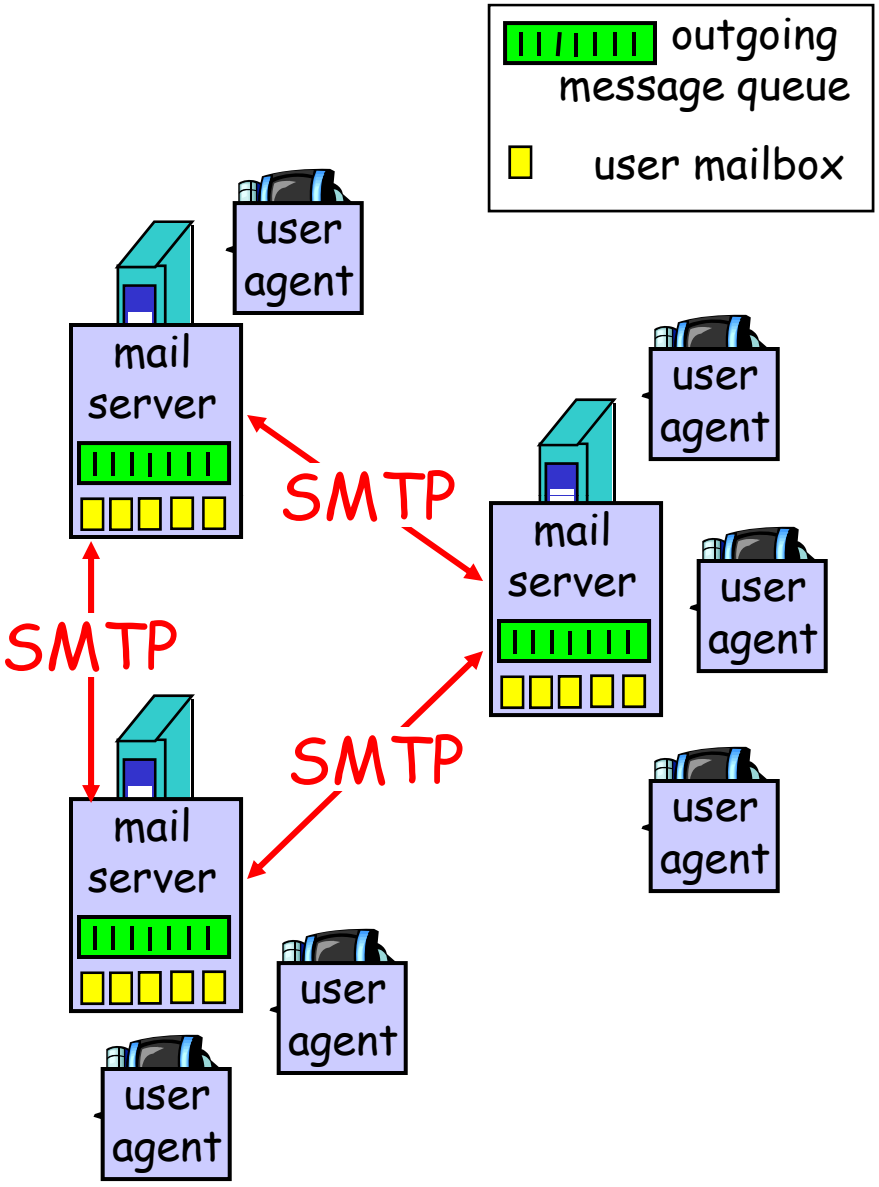
- Host at cis.poly.edu wants IP address for gaia.cs.umass.edu
- Infrastructure:
  - Client resolver
  - Local DNS server
  - Authoritative DNS Server
  - Root DNS Server
  - Top-Level Domain DNS Server



Electronic Mail

Three major components:

- user agents
  - e.g., Eudora, Outlook, Pine, Netscape Messenger
- mail servers
  - Incoming, outgoing messages
- Push protocol –SMTP
- Pull protocol – HTTP, IMAP, POP3



Outline

- Introduction
- Applications
- Transport Layer
- Network Layer
- LANs and WLANs
- Multimedia Networking
- Questions

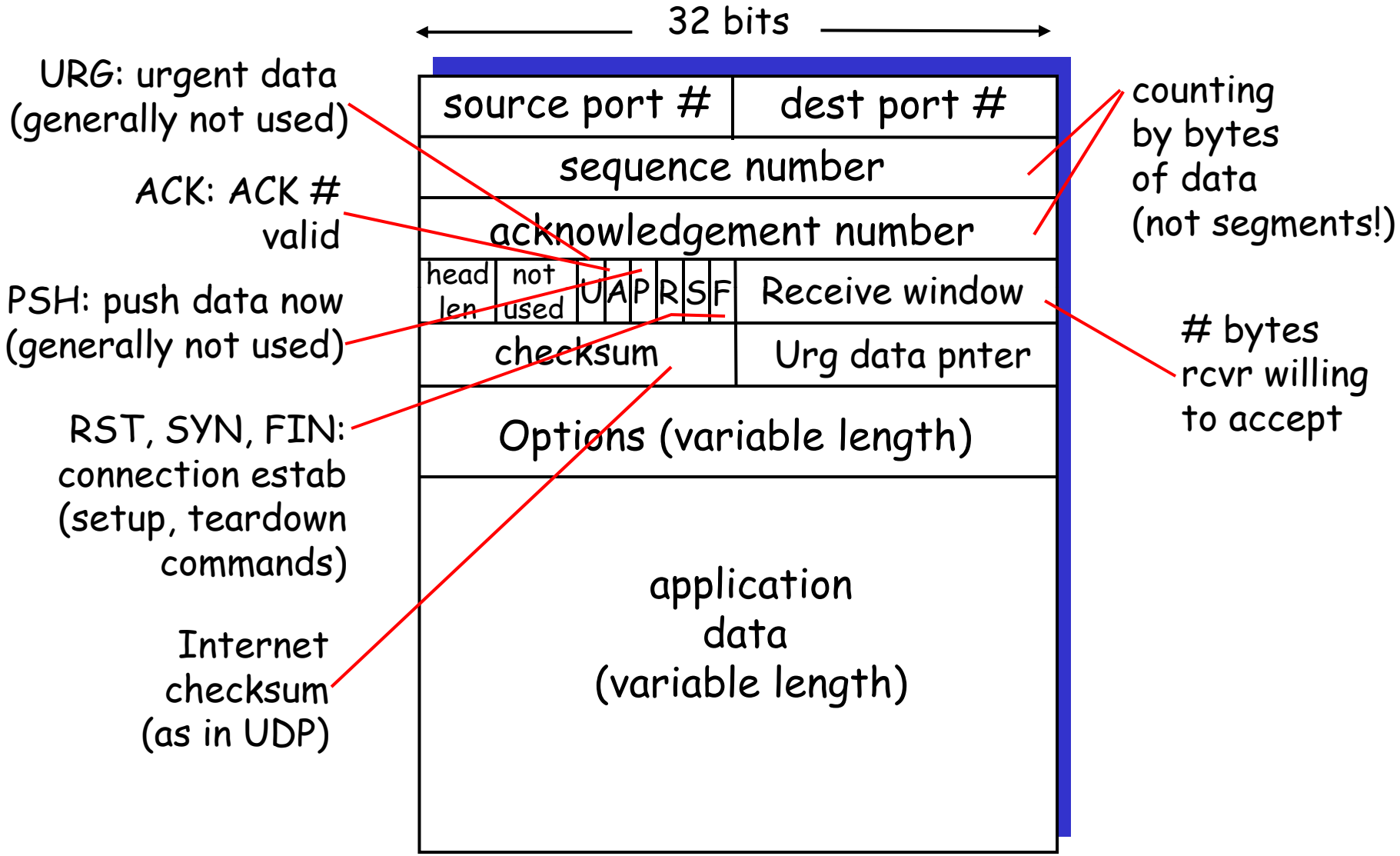
Transport Layer

- Logical end-to-end communication between processes running on different end systems
- Multiplexing/de-multiplexing
- Service models: UDP vs. TCP
  - UDP provides multiplexing/de-multiplexing
  - In addition to the above, TCP provides flow control, congestion control, and reliable data delivery
  - Some applications use UDP, while some use TCP. Why?

Reliable Delivery Concepts

- GBN, SR – also called "stop-and-wait" protocols
- ACK, NAK, SACK?
- Performance of "stop-and-wait"
  - Pipelining
- Similarity/differences between TCP and the above protocols

TCP segment structure (1/2)

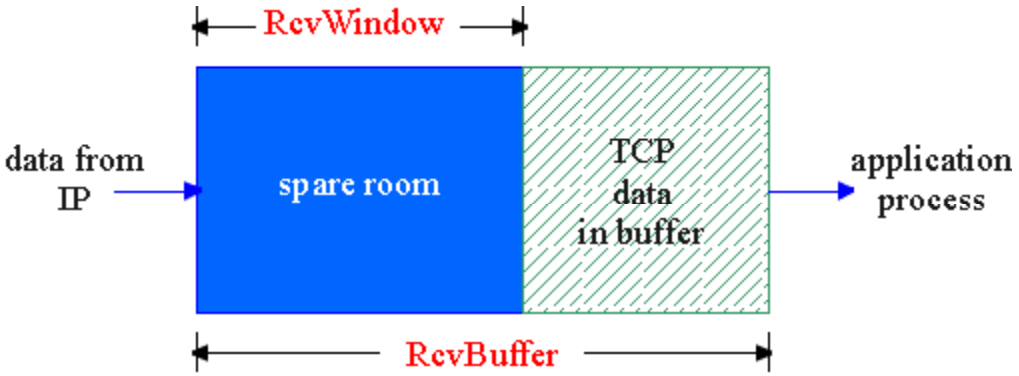


TCP Segment Structure (2/2)

- Sequence and acknowledgement numbering
- TTL
- Checksum – compulsory in TCP but not in UDP
- Handshaking procedures during TCP connection set-up and connection termination
  - SYN, FIN, RST fields

TCP Flow Control

- receive side of TCP connection has a receive buffer:



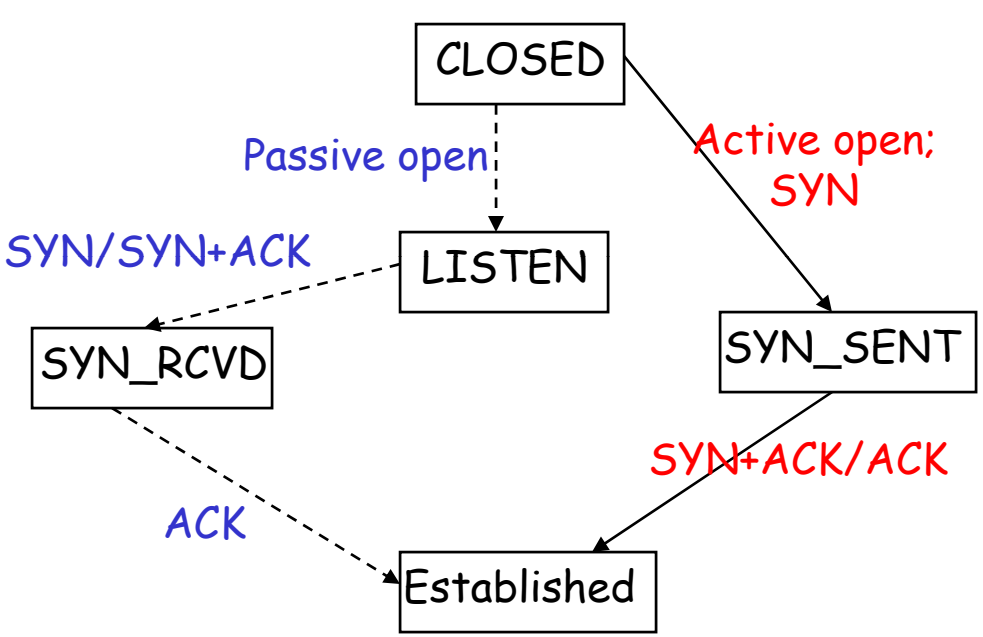
- app process may be slow at reading from buffer

flow control

sender won't overflow receiver's buffer by transmitting too much, too fast

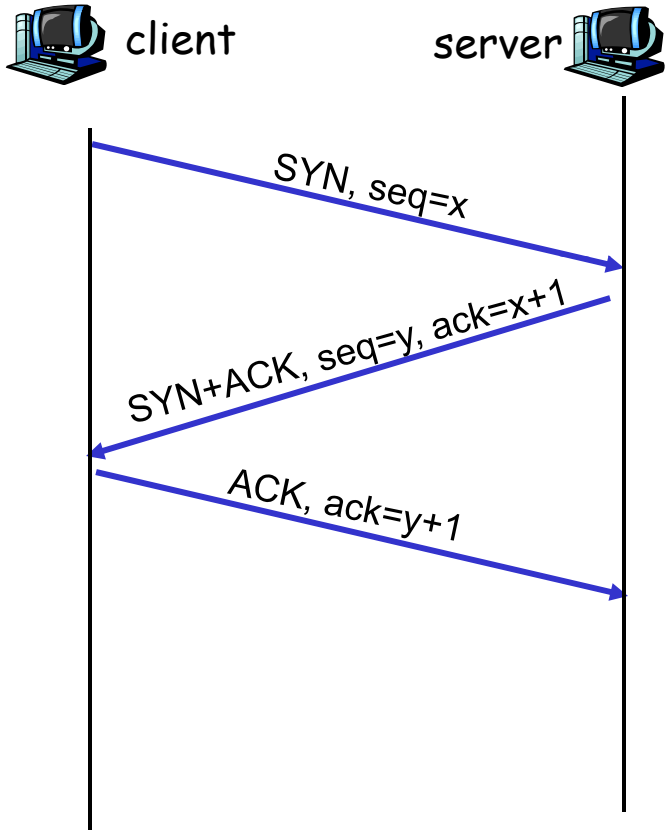
- speed-matching service: matching the send rate to the receiving app's drain rate

TCP Connection Establishment

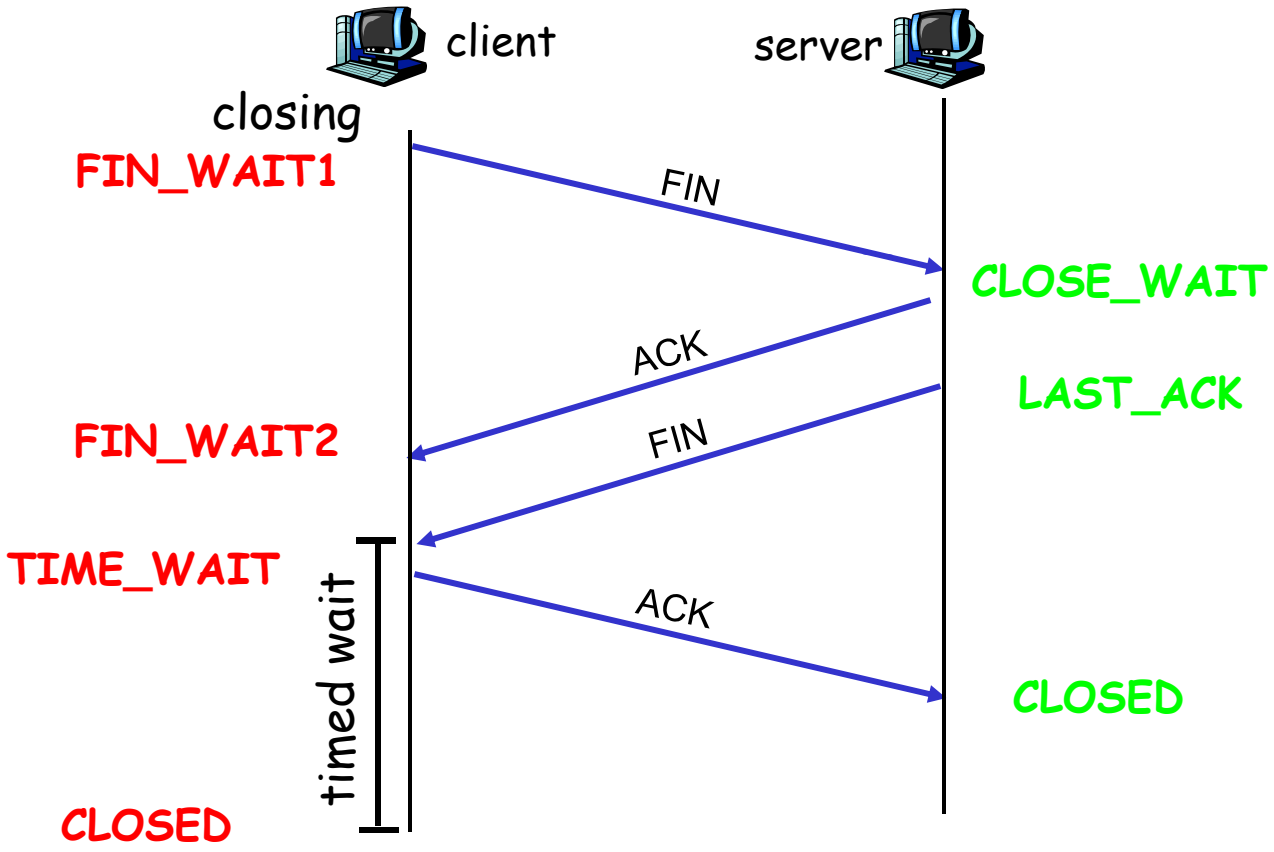


Solid line for client

Dashed line for server



TCP Connection Termination



Principles of Congestion Control

- Congestion: informally: "too many sources sending too much data too fast for *network* to handle"
- Different from flow control!
- Manifestations:
  - Packet loss (buffer overflow at routers)
  - Increased end-to-end delays (queuing in router buffers)
- Results in unfairness and poor utilization of network resources
  - Resources used by dropped packets (before they were lost)
  - Retransmissions
  - Poor resource allocation at high load

Congestion Control: Approaches

- Goal: Throttle senders as needed to ensure load on the network is "reasonable"
- End-end congestion control:
  - no explicit feedback from network
  - congestion inferred from end-system observed loss, delay
  - approach taken by TCP
- Network-assisted congestion control:
  - routers provide feedback to end systems
  - single bit indicating congestion (e.g., ECN)
  - explicit rate sender should send at

TCP Congestion Control: Overview

- end-end control (no network assistance)
- Limit the number of packets in the network to window W
- Roughly,

$$\text{rate} = \frac{W}{RTT} \text{ Bytes/sec}$$

- W is dynamic, function of perceived network congestion

TCP Congestion Controls

- Tahoe (Jacobson 1988)
  - Slow Start
  - Congestion Avoidance
  - Fast Retransmit
- Reno (Jacobson 1990)
  - Fast Recovery

TCP Tahoe

- Basic ideas
  - Gently probe network for spare capacity
  - Drastically reduce rate on congestion
  - Windowing: self-clocking
  - Other functions: round trip time estimation, error recovery

```
for every ACK {
        if (W < ssthresh) then W++      (SS)
        else          W += 1/W          (CA)

    }
    for every loss {
            ssthresh = W/2
            W  = 1
    }
```

TCP Reno: Fast Recovery

- Objective: prevent `pipe' from emptying after fast retransmit
  - each dup ACK represents a packet having left the pipe (successfully received)
  - Let's enter the "FR/FR" mode on 3 dup ACKs

ssthresh ← W/2
retransmit lost packet
W ← ssthresh + ndup (window inflation)
Wait till W is large enough; transmit new packet(s)
On non-dup ACK (1 RTT later)
    W ← ssthresh (window deflation)
    enter CA mode

TCP Reno: Summary

- Fast Recovery along with Fast Retransmit used to avoid slow start
- On 3 duplicate ACKs
  - Fast retransmit and fast recovery
- On timeout
  - Fast retransmit and slow start

TCP Reno Throughput

- Average throughout: .75 W/RTT
- Throughput in terms of loss rate:

$$\frac{1.22 \cdot MSS}{RTT\sqrt{L}}$$

- What happens if L->0?
- Does link capacity matter if we experience random loss?
- High-speed TCP?

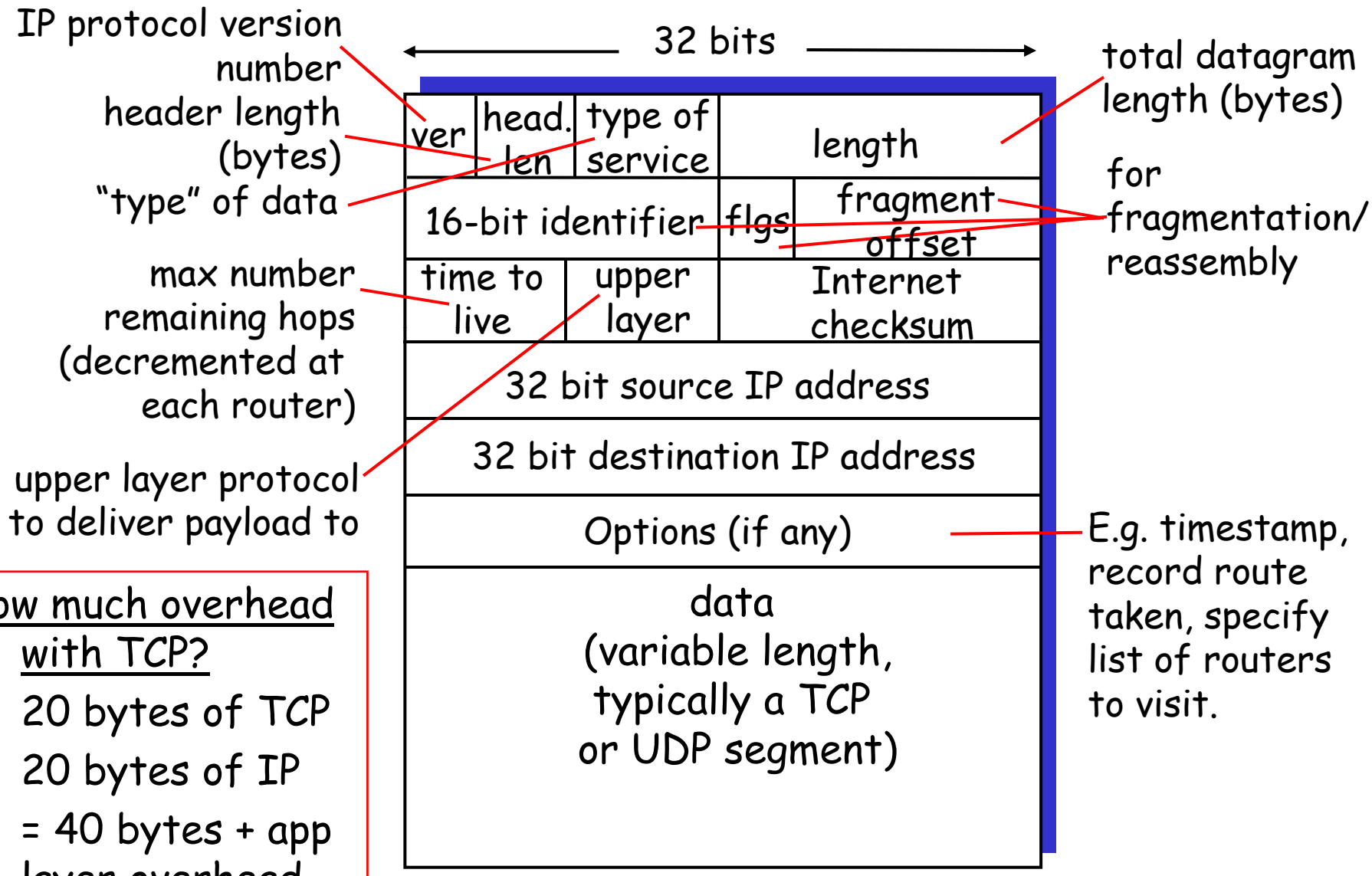
Outline

- Introduction
- Applications
- Transport Layer
- Network Layer
- LANs and WLANs
- Multimedia Networking
- Questions

Network Layer

- Transport segment from sending to receiving host
- Network layer protocols in every host, router [contrast with transport/application layer]
- Main functions of network layer
  - Forwarding – moving datagrams within a router
  - Routing – determine end-to-end paths taken by packets

IP datagram format

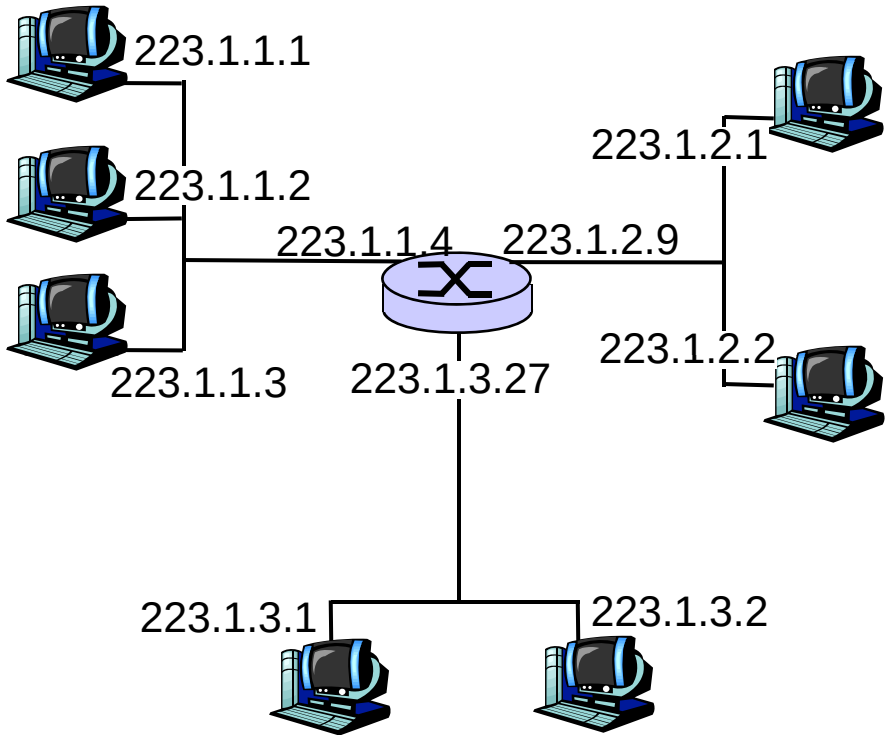


how much overhead with TCP?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

IPv4 Addressing

- IP address: 32-bit identifier for host, router *interface*
- *interface:* connection between host/router and physical link
  - router's typically have multiple interfaces
  - host may have multiple interfaces
  - IP addresses associated with each interface



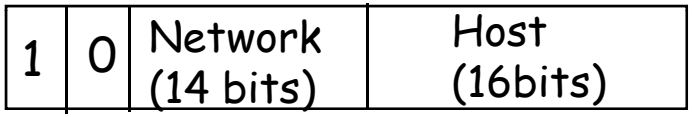
223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

Classful Addressing

- Addresses consists of:
  - Network part
  - Host part
- IP addresses divided into five classes: A, B, C, D, and E.
- Problems ??

Class A



Class B

Class C

Class D

Class E

Subnets: Motivation

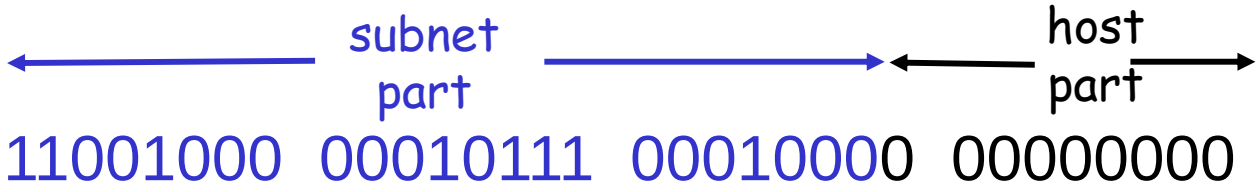
- The "classful" addressing scheme proposes that the network portion of a IP address uniquely identifies one physical network.
  - Any network with more than 255 hosts needs a class B address. Class B addresses can get exhausted before we have 4 billion hosts!

- Take bits from the host number part to create a "subnet" number.

Addressing in the Internet

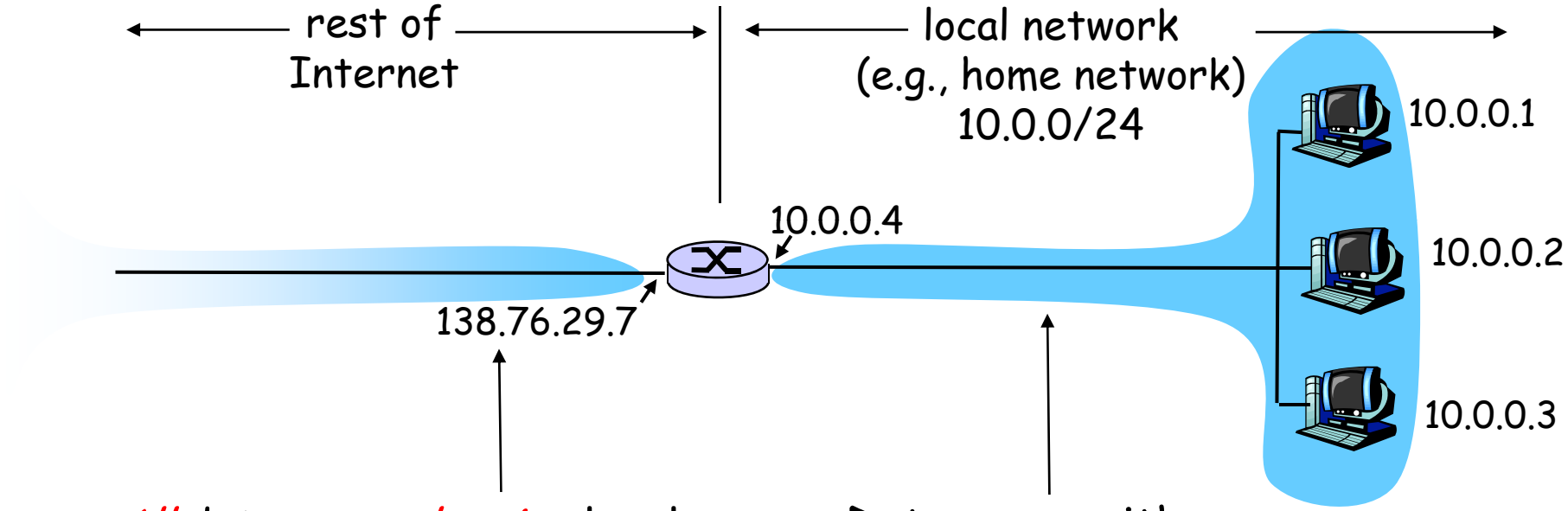
CIDR: Classless InterDomain Routing

- subnet portion of address of arbitrary length
- address format: a.b.c.d/x, where x is # bits in subnet portion of address
- Before CIDR, Internet used a class-based addressing scheme where x could be 8, 16, or 24 bits. These corrsp to classes A, B, and C resp.



200.23.16.0/23

NAT: Network Address Translation



All datagrams *leaving* local network have same single source NAT IP address: 138.76.29.7, different source port numbers

Datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

Routing Algorithm Classification

1. Global, decentralized ?

Global:

- all routers have complete topology, link cost info
- "link state" algorithms

Decentralized:

- router knows about physically-connected neighbors
- Iterative, distributed computations
- "distance vector" algorithms

2. Static, dynamic?

Static:

- routes change slowly over time

Dynamic:

- routes change more quickly
  - periodic update
  - in response to link cost changes

3. Load sensitivity?

- Many Internet routing algos are load insensitive

Why Hierarchical Routing?

- scale: with 200 million destinations:
  - can't store all dest's in routing tables!
  - routing table exchange would swamp links!
- administrative autonomy
  - internet = network of networks
  - each network admin may want to control routing in its own network

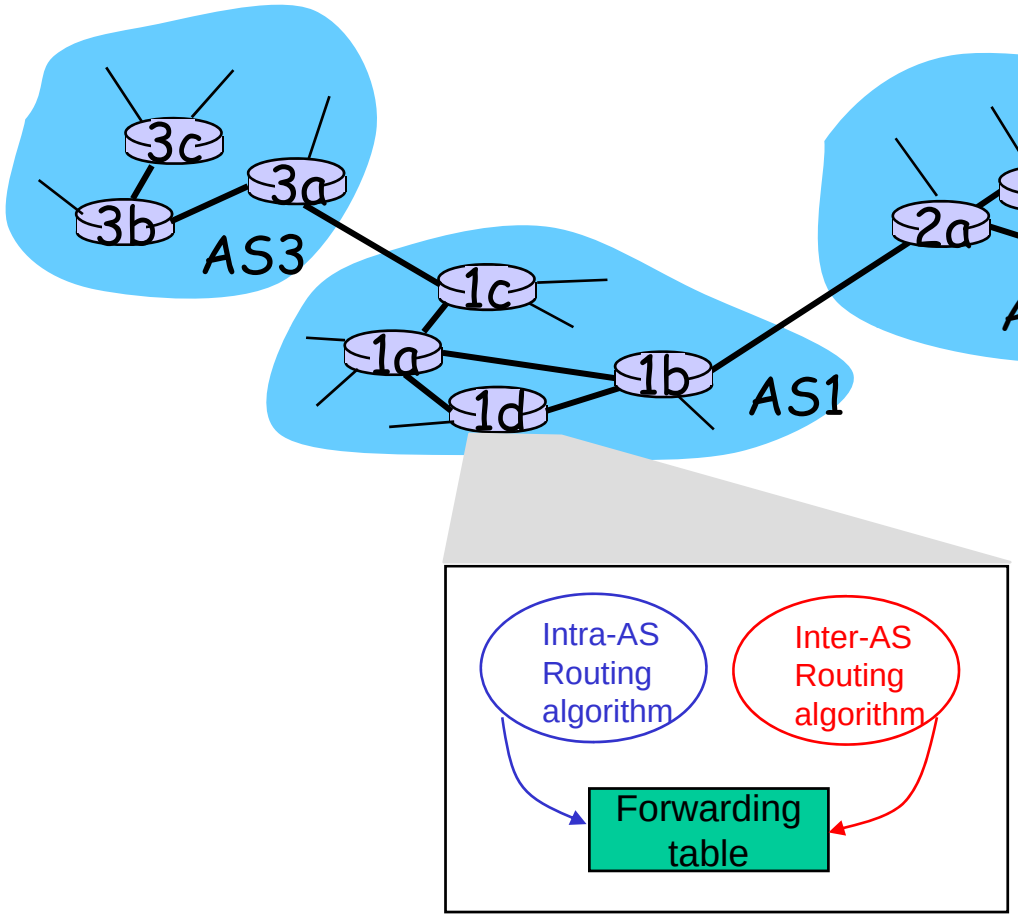
Hierarchical Routing

- aggregate routers into regions, "autonomous systems" (AS)
- routers in same AS run same routing protocol
  - "intra-AS" routing protocol
  - routers in different AS can run different intra-AS routing protocol

Gateway router

- Direct link to router in another AS
- Establishes a "peering" relationship
- Peers run an "inter-AS routing" protocol

Interconnected ASes



- Forwarding table is configured by both intra- and inter-AS routing algorithm
  - Intra-AS sets entries for internal dests
  - Inter-AS & Intra-As sets entries for external dests

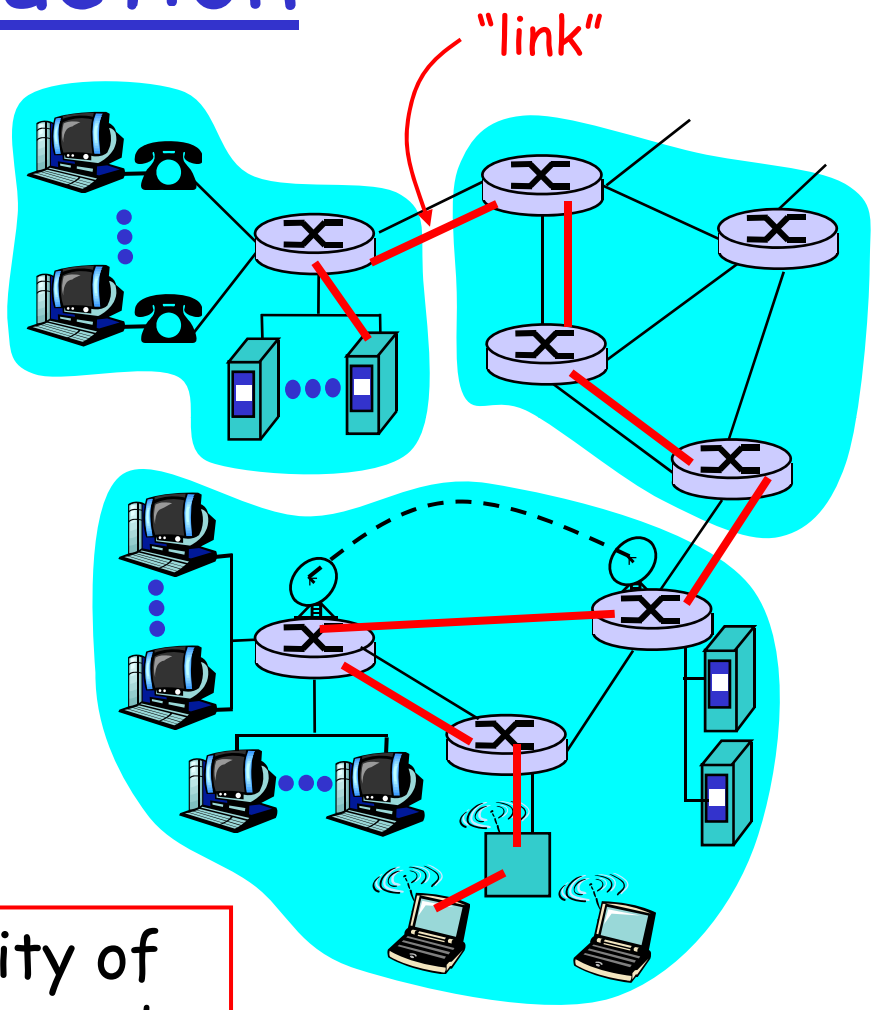
Outline

- Introduction
- Applications
- Transport Layer
- Network Layer
- LANs and WLANs
- Multimedia Networking
- Questions

Link Layer: Introduction

Some terminology:

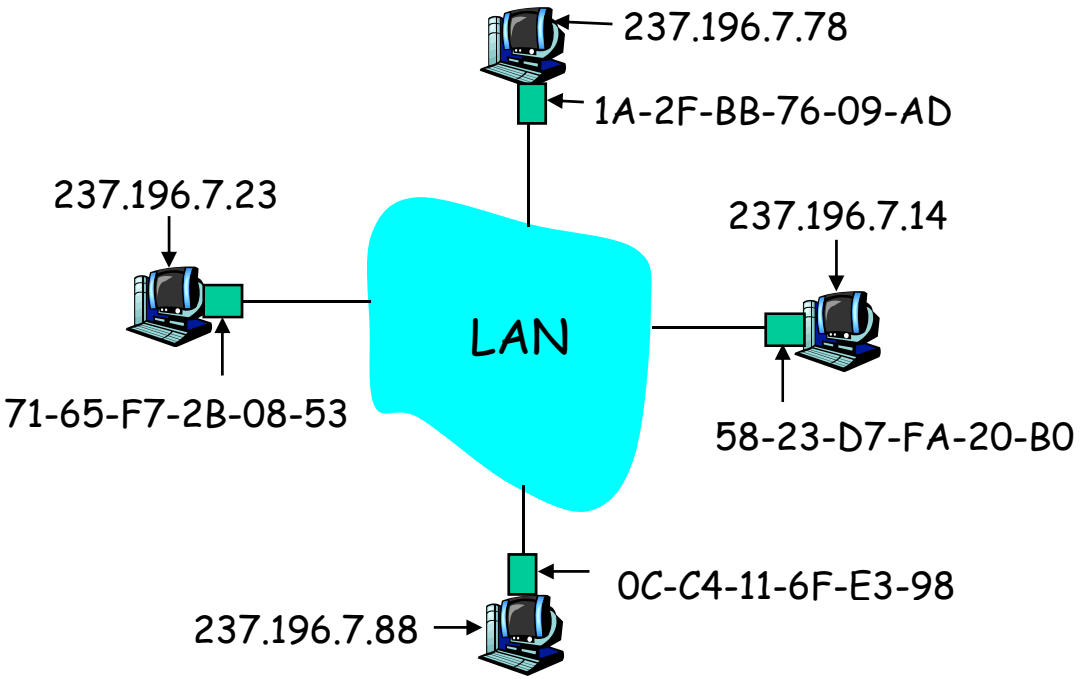
- hosts and routers are **nodes**
- communication channels that connect adjacent nodes along communication path are **links**
  - wired links
  - wireless links
  - LANs
- layer-2 packet is a **frame**, encapsulates datagram



data-link layer has responsibility of transferring datagram from one node to adjacent node over a link

ARP: Address Resolution Protocol

Question: how to determine MAC address of B knowing B's IP address?



- Each IP node (Host, Router) on LAN has ARP table
- ARP Table: IP/MAC address mappings for some LAN nodes

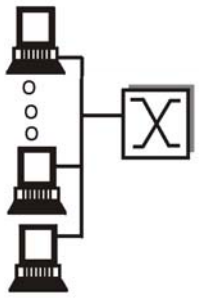
  < IP address; MAC address; TTL>

  - TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

Multiple Access Links and Protocols

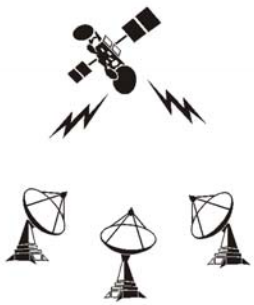
Two types of "links":

- point-to-point
  - PPP for dial-up access
  - point-to-point link between Ethernet switch and host
- broadcast (shared wire or medium)
  - traditional Ethernet
  - upstream HFC
  - 802.11 wireless LAN



shared wire
(e.g. Ethernet)

shared wireless
(e.g. Wavelan)



satellite

cocktail party

Taxonomy of Multiple Access Control Protocols

Three broad classes:

- Channel Partitioning
  - divide channel into smaller "pieces" (TDM, FDM, Code Division Multiple Access)
  - allocate piece to node for exclusive use
- Random Access
  - channel not divided, allow collisions
  - "recover" from collisions
- "Taking turns"
  - Nodes take turns, but nodes with more to send can take longer turns

Random Access Protocols

- When node has packet to send
  - transmit at full channel data rate R.
  - no *a priori* coordination among nodes
- two or more transmitting nodes ➜ "collision",
- random access MAC protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)
- Examples of random access MAC protocols:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

CSMA/CD (Collision Detection)

CSMA/CD: carrier sensing, deferral as in CSMA

- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage

- collision detection:
  - easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - difficult in wireless LANs: receiver shut off while transmitting

Ethernet CSMA/CD algorithm

1. Adaptor receives datagram from net layer & creates frame
2. If adapter senses channel idle, it starts to transmit frame. If it senses channel busy, waits until channel idle and then transmits
3. If adapter transmits entire frame without detecting another transmission, the adapter is done with frame !
4. If adapter detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, adapter enters **exponential backoff**: after the mth collision, adapter chooses a K at random from $\{0,1,2,\ldots,2^m-1\}$. Adapter waits $K \cdot 512$ bit times and returns to Step 2

Ethernet's CSMA/CD (more)

Jam Signal: make sure all other transmitters are aware of collision; 48 bits

Bit time: .1 microsec for 10 Mbps Ethernet ;
for K=1023, wait time is about 50 msec

See/interact with Java applet on AWL Web site: highly recommended !

Exponential Backoff:

- *Goal*: adapt retransmission attempts to estimated current load
  - heavy load: random wait will be longer
- first collision: choose K from {0,1}; delay is K· 512 bit transmission times
- after second collision: choose K from {0,1,2,3}…
- after ten collisions, choose K from {0,1,2,3,4,…,1023}

CSMA/CD efficiency

- t_{prop} = max prop between 2 nodes in LAN
- t_{trans} = time to transmit max-size frame

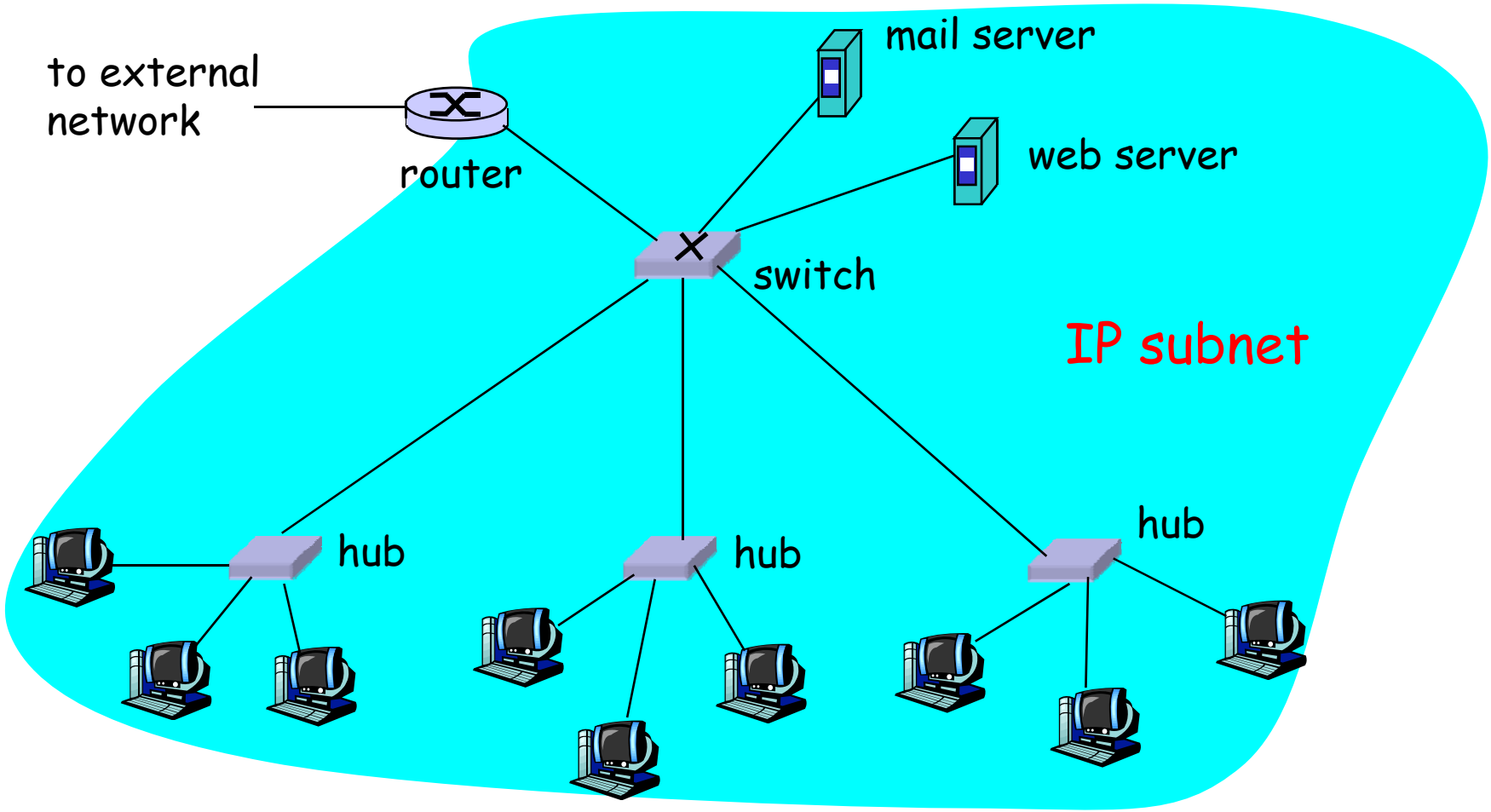
$$\text{efficiency} = \frac{1}{1 + 5t_{prop} / t_{trans}}$$

- Efficiency goes to 1 as t_{prop} goes to 0
- Goes to 1 as t_{trans} goes to infinity
- Much better than ALOHA, but still decentralized, simple, and cheap

Switches, Routers, and Hubs

- Hubs are physical layer repeaters;
  - no CSMA/CD at hub
- Switch is a link layer store-and-forward device
  - CSMA/CD at switch
  - Maintains switch tables, implement filtering, learning algos
- Routers are network-layer store-and-forward devices
  - maintain routing tables, implement routing algorithms

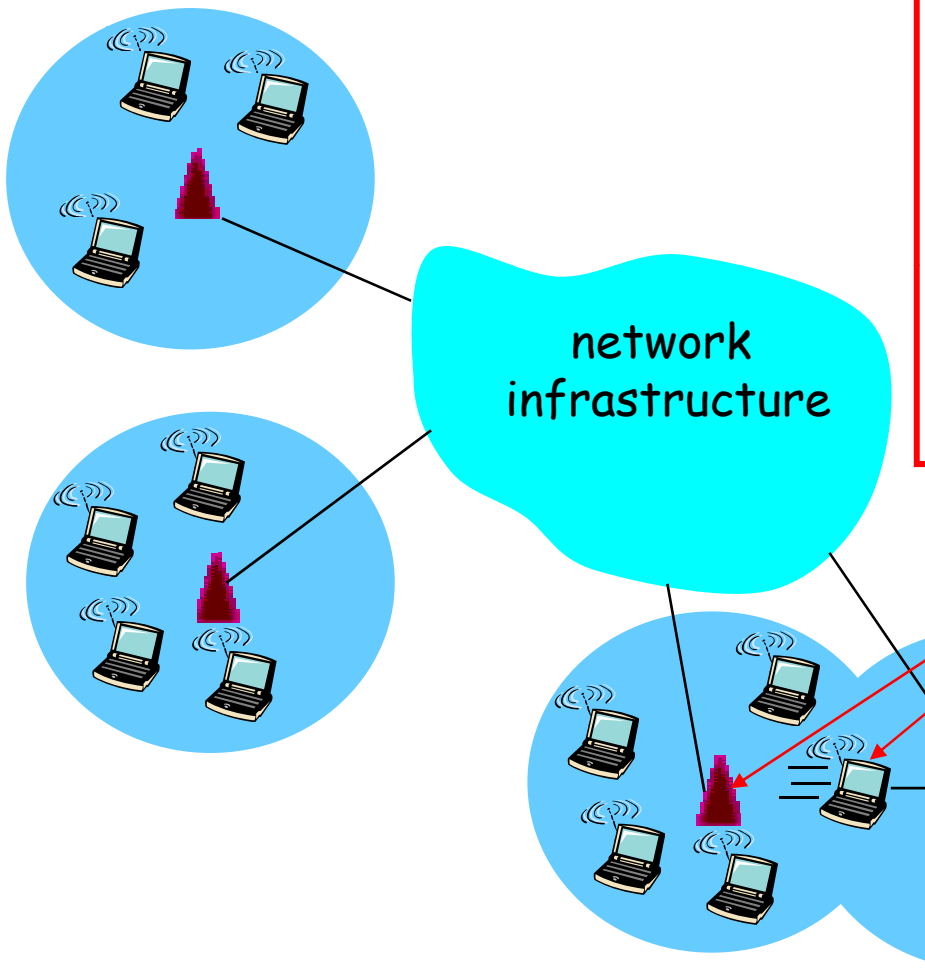
A network with switch, router, and hubs



Wireless Networking Technologies

- Mobile devices – laptop, PDA, cellular phone, wearable computer, …
- Operating modes
  - Infrastructure mode (Access Point)
  - Ad hoc mode
- Access technology
  - Bluetooth (1 Mbps, up to 3 meters)
  - IEEE 802.11 (up to 55 Mbps, 20 – 100 meters)

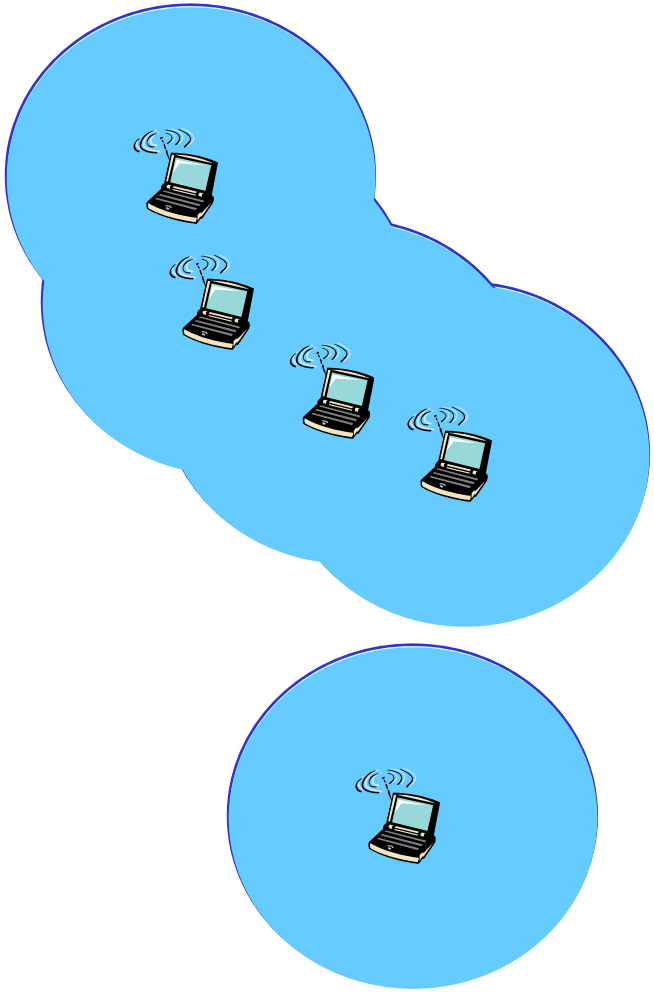
Infrastructure Mode



infrastructure mode

- base station connects mobiles into wired network
- handoff: mobile changes base station providing connection into wired network

Ad hoc Mode

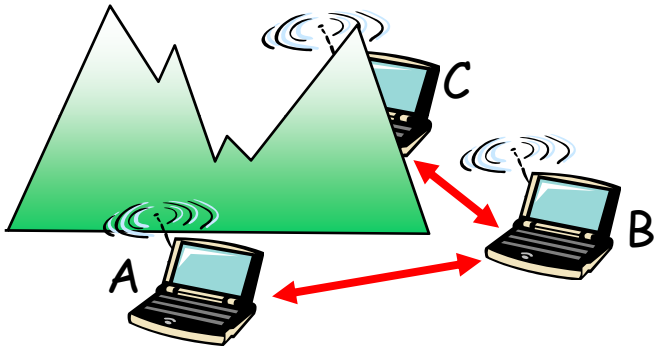


Ad hoc mode

- no base stations
- nodes can only transmit to other nodes within link coverage
- nodes organize themselves into a network: route among themselves

Wireless Network Characteristics

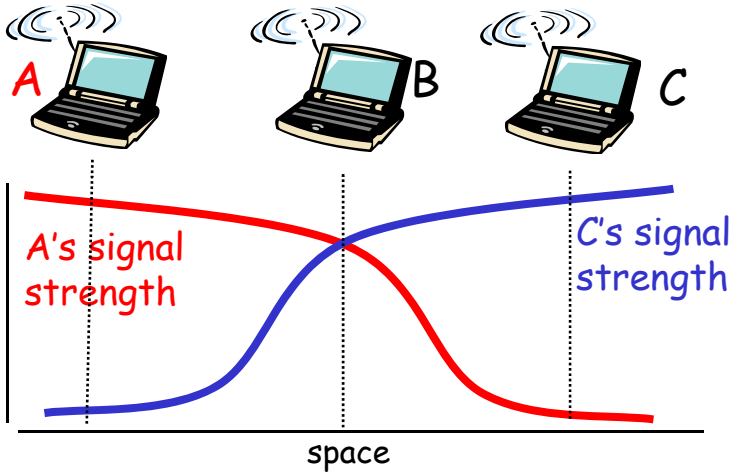
Multiple wireless senders and receivers create additional problems (beyond multiple access):



Hidden terminal problem

- B, A hear each other
- B, C hear each other
- A, C can not hear each other

means A, C unaware of their interference at B



Signal fading:

- B, A hear each other
- B, C hear each other
- A, C can not hear each other interfering at B

IEEE 802.11 MAC Protocol: CSMA/CA

802.11 sender

1 if sense channel idle for **DIFS** then

transmit entire frame (no CD)

2 if sense channel busy then

start random backoff time

timer counts down while channel idle

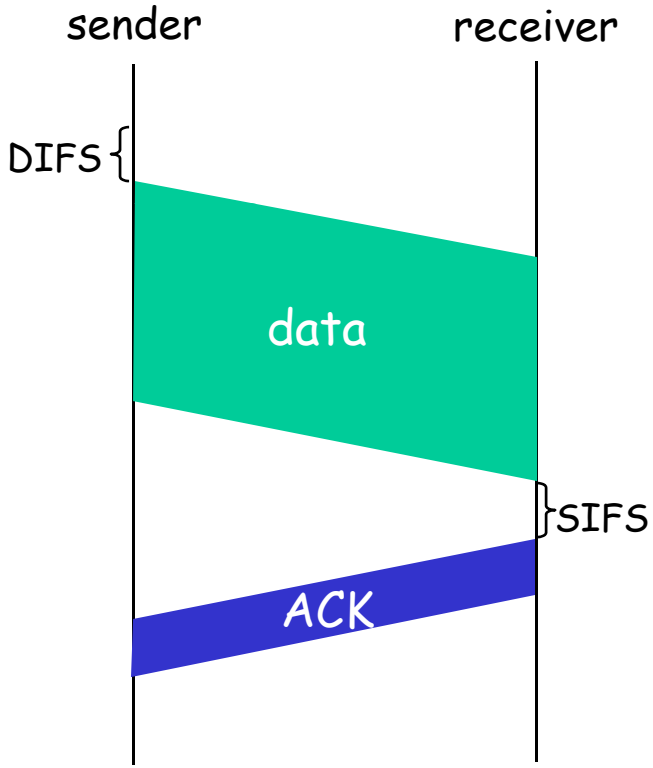
transmit when timer expires

3 if no ACK, increase random backoff interval, repeat 2

802.11 receiver

- if frame received OK

return ACK after **SIFS** (ACK needed due to hidden terminal problem)



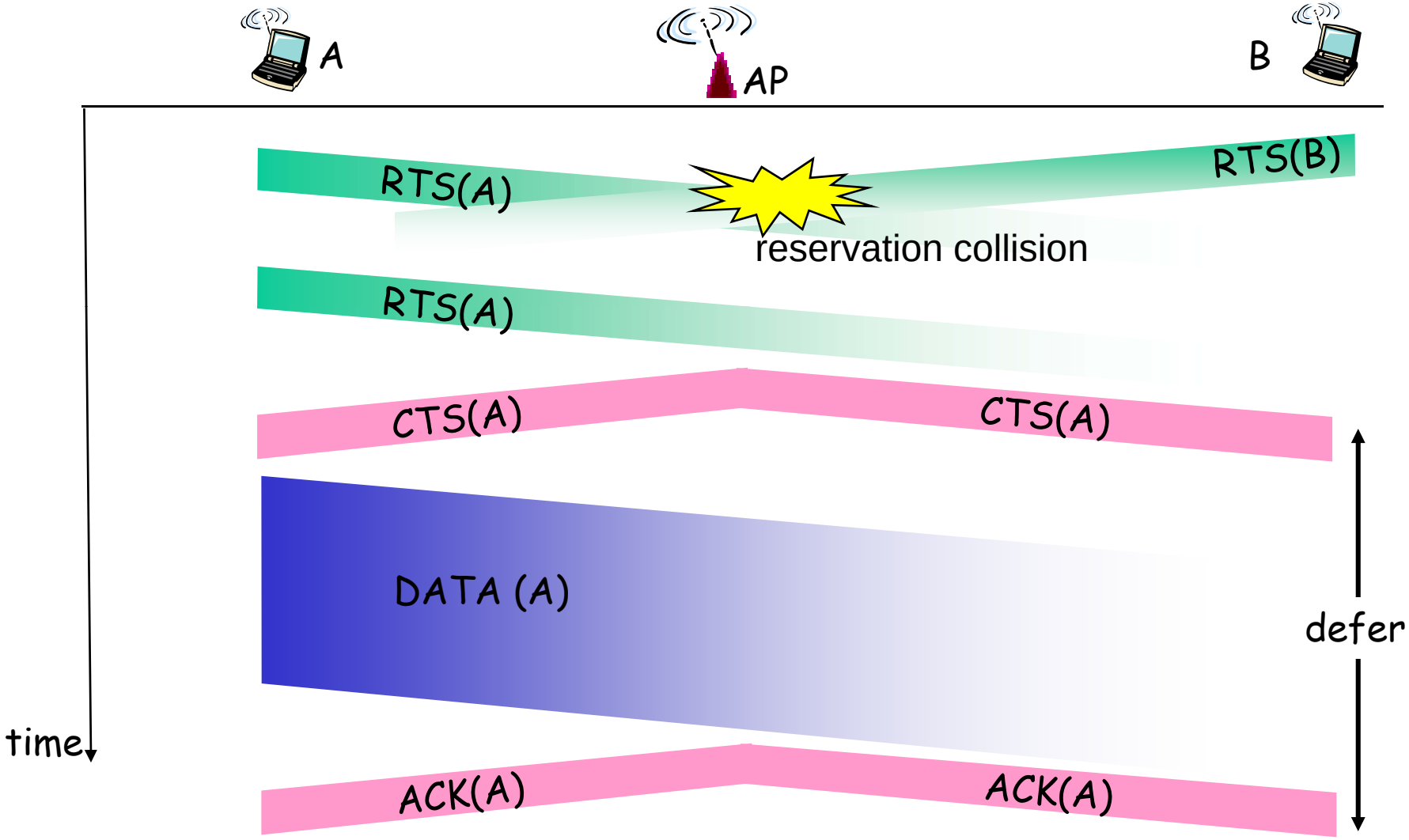
Avoiding collisions (more)

idea: allow sender to "reserve" channel rather than random access of data frames: avoid collisions of long data frames

- sender first transmits *small* request-to-send (RTS) packets to base station using CSMA
  - RTSs may still collide with each other (but they're short)
- BS broadcasts clear-to-send CTS in response to RTS
- RTS heard by all nodes
  - sender transmits data frame
  - other stations defer transmissions

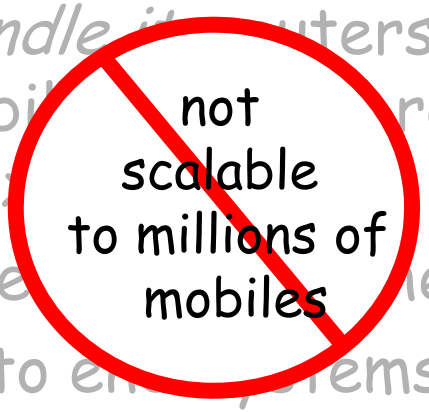
Avoid data frame collisions completely using small reservation packets!

Collision Avoidance: RTS-CTS exchange



Mobile IP: Overview

- *Let routing handle it:* routers advertise permanent address of mobile-end-system's residence via usual routing table exchange.
  - routing table indicates where each mobile located
  - no changes to end-systems
- *let end-systems handle it:*
  - *indirect routing:* communication from correspondent to mobile goes through home agent, then forwarded to remote
  - *direct routing:* correspondent gets foreign address of mobile, sends directly to mobile



Outline

- Introduction
- Applications
- Transport Layer
- Network Layer
- LANs and WLANs
- Multimedia Networking
- Questions

MM Networking Applications

Classes of MM applications:

1) Streaming stored audio and video
2) Streaming live audio and video
3) Real-time interactive audio and video

Jitter is the variability of packet delays within the same packet stream

Fundamental characteristics:

- Typically **delay sensitive**
  - end-to-end delay
  - delay jitter
- But **loss tolerant**: infrequent losses cause minor glitches
- *Antithesis* of data

Multimedia Over "Best Effort" Internet

- TCP/UDP/IP: *no* guarantees on delay, loss

? ? ? ? ? ? ? ? ? ? ?

But you said multimedia apps requires QoS and level of performance to be effective!

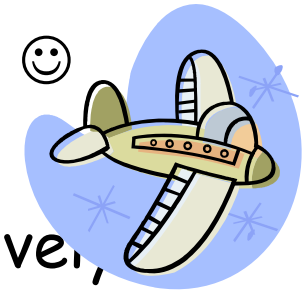
Today's multimedia applications implement functionality at the app. layer to mitigate (as best possible) effects of delay, loss

How to provide better support for Multimedia? (1/4)

- Integrated services philosophy: architecture for providing QOS guarantees in IP networks for individual flows
- Fundamental changes in Internet so that apps can reserve end-to-end bandwidth
- Components of this architecture are
  - Admission control
  - Reservation protocol
  - Routing protocol
  - Classifier and route selection
  - Packet scheduler

Concerns with Intserv (2/4)

- Scalability: signaling, maintaining per-flow router state difficult with large number of flows
- Flexible Service Models: Intserv has only two classes. Desire "qualitative" service classes
  - E.g., Courier, xPress, and normal mail
  - E.g., First, business, and cattle class ☺

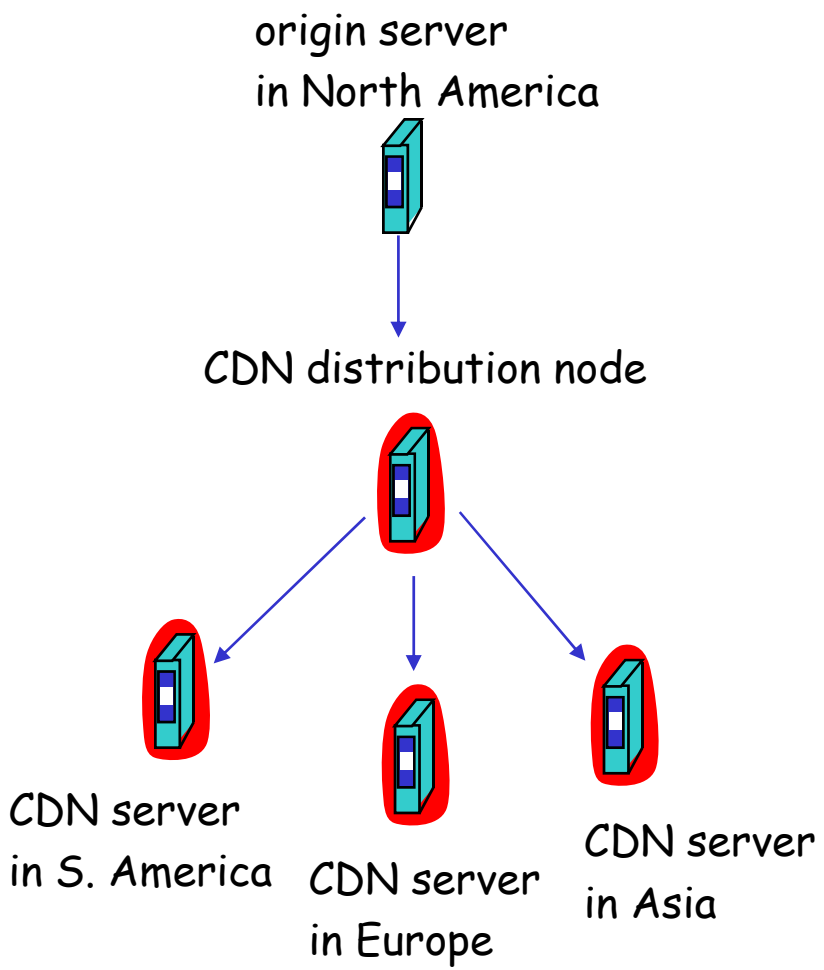


Diffserv approach:

- simple functions in network core, relative, complex functions at edge routers (or hosts)
- Don't define define service classes, provide functional components to build service classes

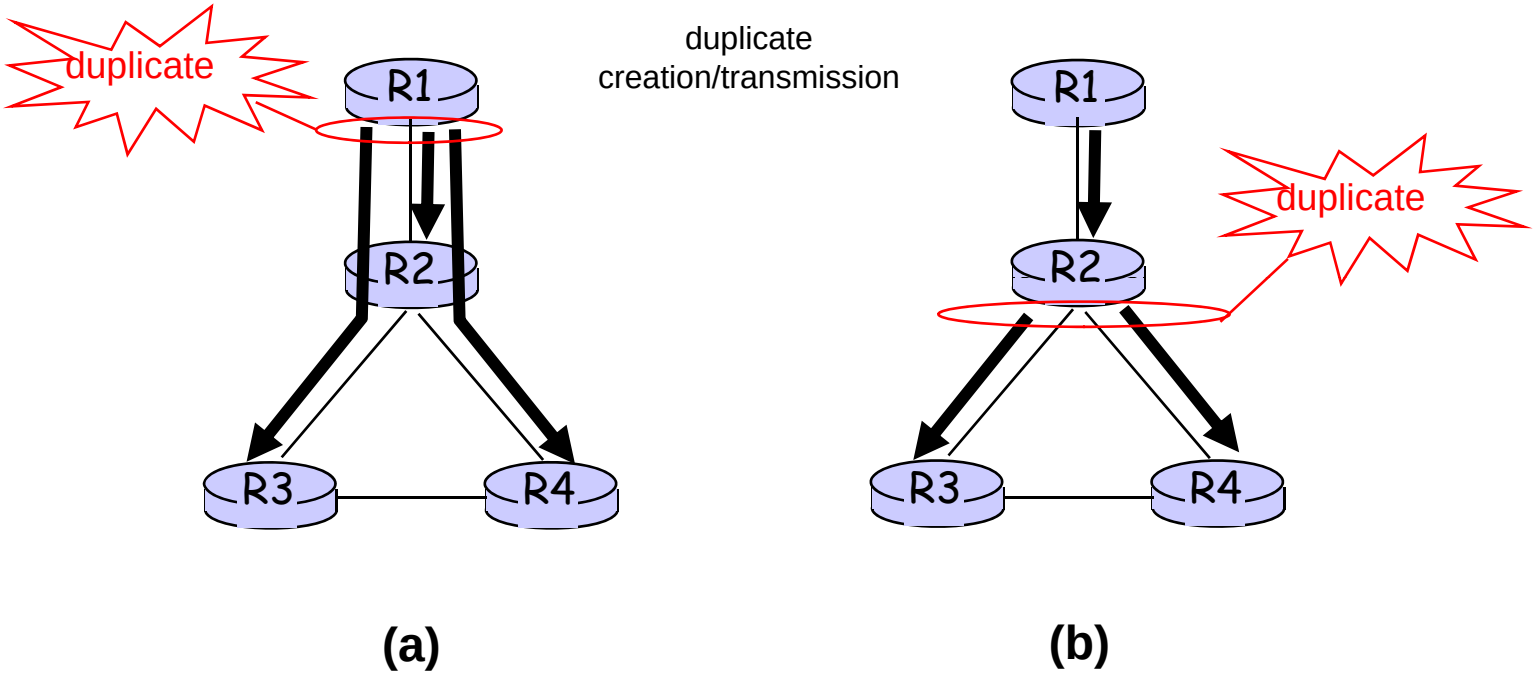
How to provide better support for Multimedia? (3/4)

- Challenging to stream large files (e.g., video) from single origin server in real time
- Solution: replicate content at hundreds of servers throughout Internet
  - content downloaded to CDN servers ahead of time
  - placing content "close" to user avoids impairments (loss, delay) of sending content over long paths
  - CDN server typically in edge/access network



How to provide better support for Multimedia? (4/4)

Multicast/Broadcast



Source-duplication versus in-network duplication.
(a) source duplication, (b) in-network duplication