Cs 232

Local Area Networks (LANs)

Internet Layering

Content (layer 7)
- web pages, videos, blogs, RSS feeds, online radio stations, …

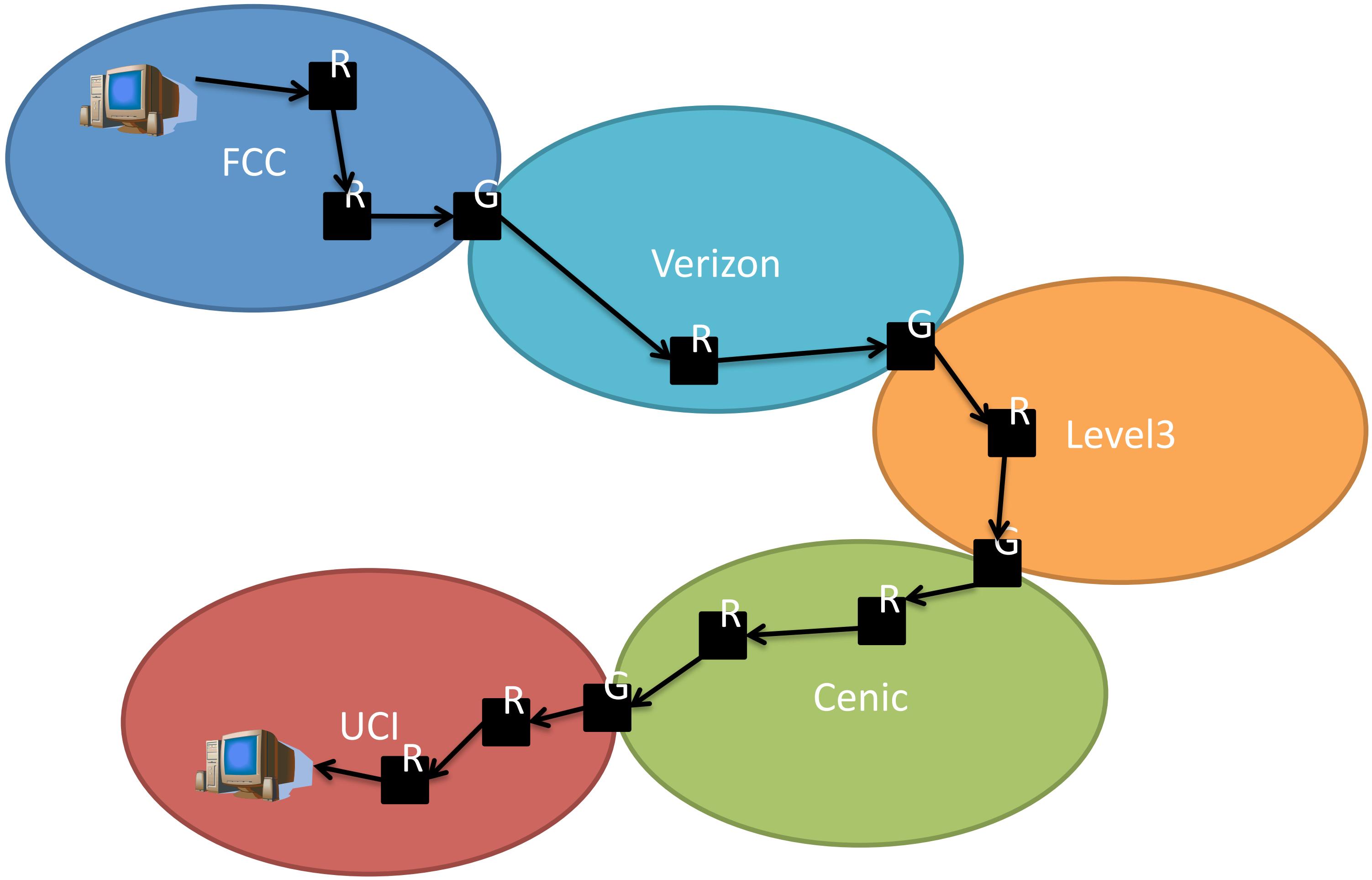
Applications (layers 5-7)
- http, email, file transfer, streaming, instant messaging, web hosting, blogging, video conferencing, remote desktop, …

Transport (layer 4)
- TCP congestion control

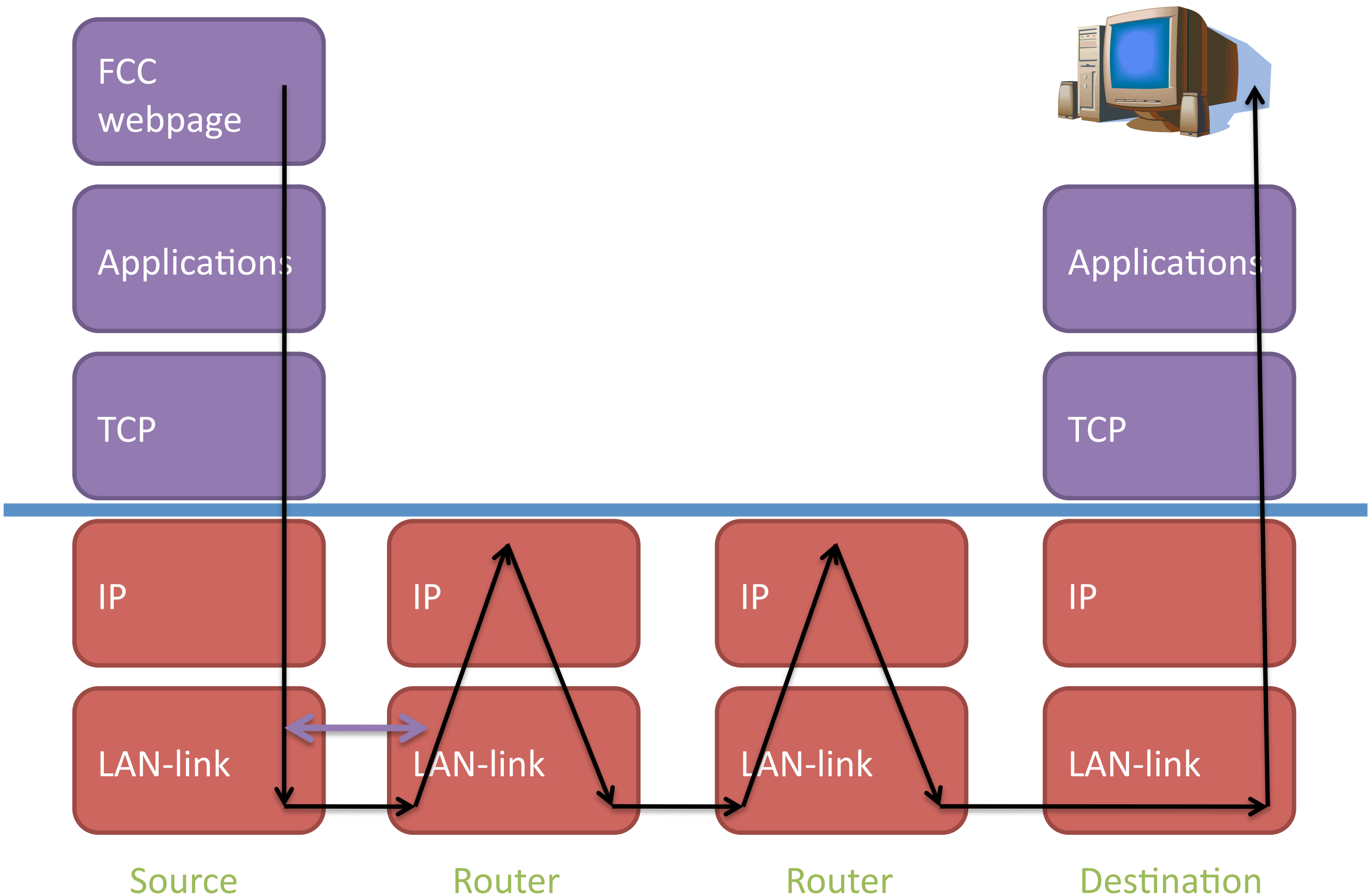
Network (layer 3)
- IP addresses, routing

Physical & Link "LAN-link" (layers 1-2)
- Ethernet, DSL, cable modems, Wi-Fi, local loop, …

Peering



LAN: router-to-router



Multiple Access

- Problem: Nodes may have to share link capacity
  - Only a limited number may be able to talk at once
    - shared bandwidth
    - half-duplex
- Solutions:
  - Multiple access protocols
    - Possibly including prioritization

Switching

- Problem:
  - Determination of route through LAN
  - Implementation of routing through LAN
- Solutions:
  - Hubs
  - Switching tables

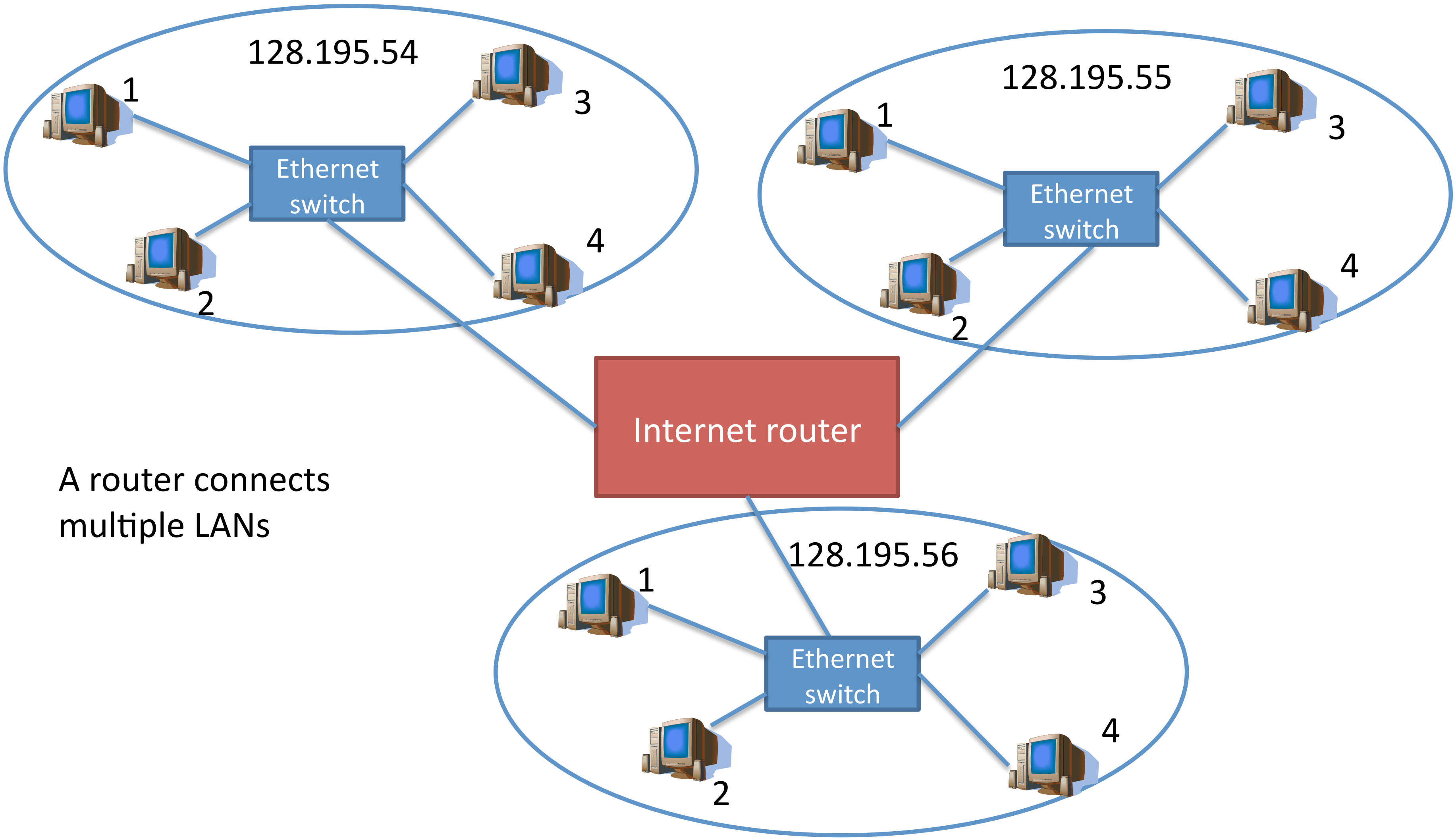
Framing

- Problem: bits need to be assembled into packets
- Solution: headers & trailers that help determine frame
  - Known bits at beginning of frame
    - “preamble”
    - “start of frame delimiter”
    - also helps with synchronization
    - requires a method to encode that same bit pattern if it’s in datagram

Error detection & correction

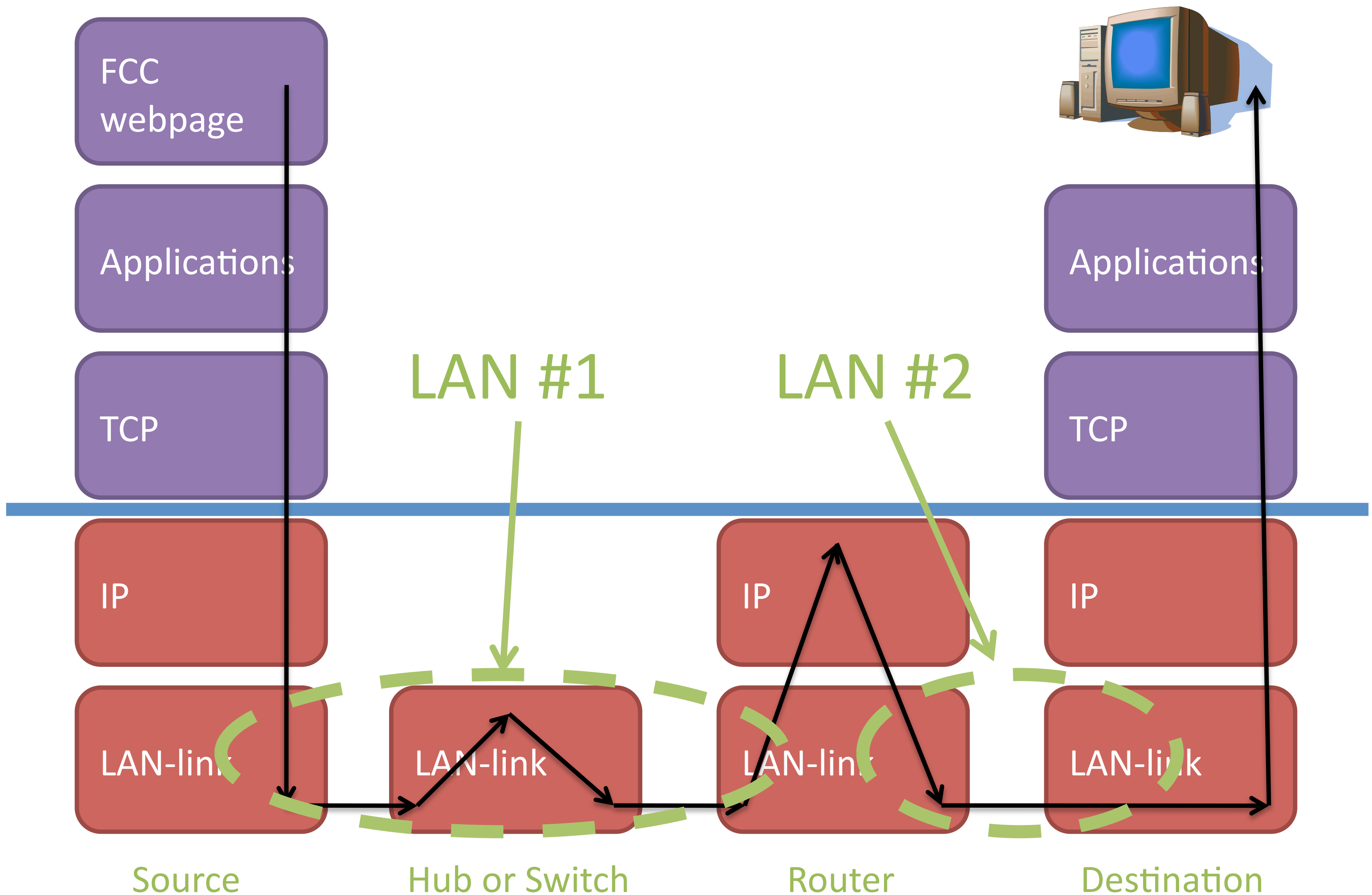
- Problem: bits may be interpreted incorrectly by physical layer
- Solutions:
  - Error detection
  - Error correction
  - Retransmission of corrupted packets from previous link layer hop

Local Area Networks

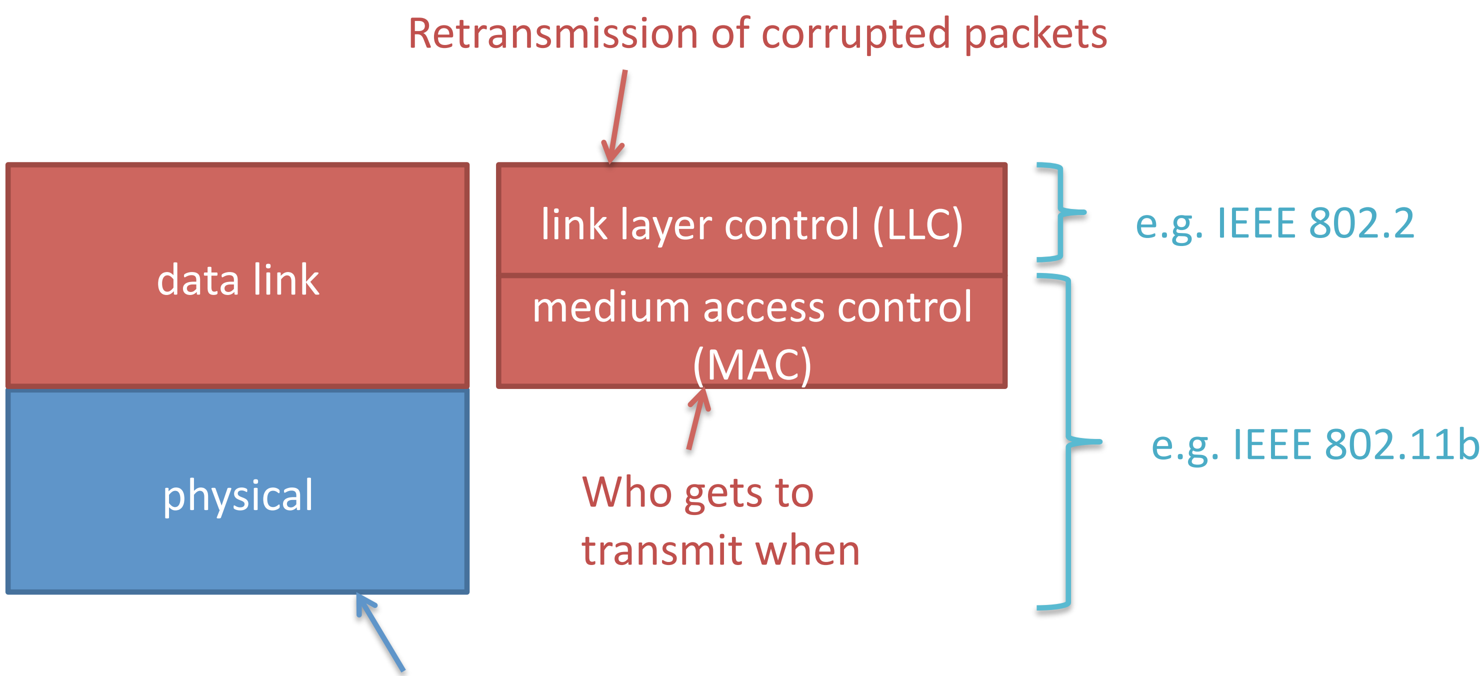


A router connects multiple LANs

Hub / Switch / Router



Layers



Everything we listed for the OSI physical layer, e.g. bits to analog signal and vice versa. Depends on transmission medium, e.g. twisted pair, coax, fiber, wireless !!

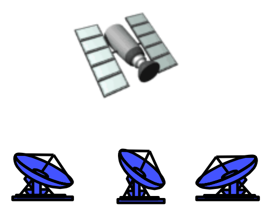
Multiple access

two types of "links":

- point-to-point
  - PPP (point-to-point protocol) for dial-up access
  - point-to-point link between Ethernet switch, host
  - Half-duplex
- *broadcast (shared wire or medium)*
  - Ethernet
  - upstream HFC
  - 802.11 wireless LAN

shared wire (e.g., cabled Ethernet)

shared RF (e.g., 802.11 WiFi)



shared RF (satellite)

Multiple access

given: broadcast channel of rate R bps

desiderata:

1. when one node wants to transmit, it can send at rate R.
2. when M nodes want to transmit, each can send at average rate R/M
3. fully decentralized:
   - no special node to coordinate transmissions
   - no synchronization of clocks, slots
4. simple

Multiple access

- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference
  - *collision* if node receives two or more signals at the same time

multiple access protocol

- distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!
  - no out-of-band channel for coordination

Multiple access strategies

- Taking turns
  – Need to know when it's your turn
- Random access
  – Need to recover from collisions

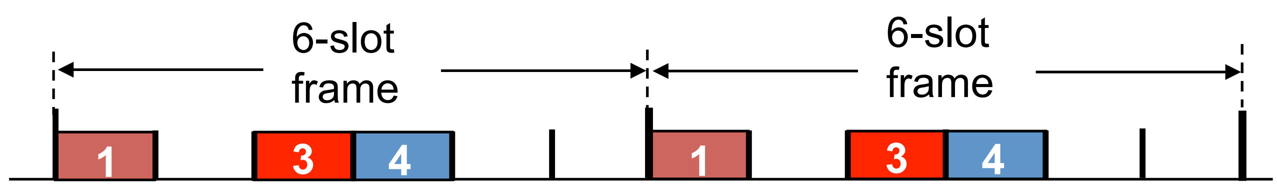
Taking turns

- Channel partitioning
  - TDMA
  - FDMA
  - CDMA
- Round robin
  - Polling
  - Tokens

TDMA

TDMA: time division multiple access

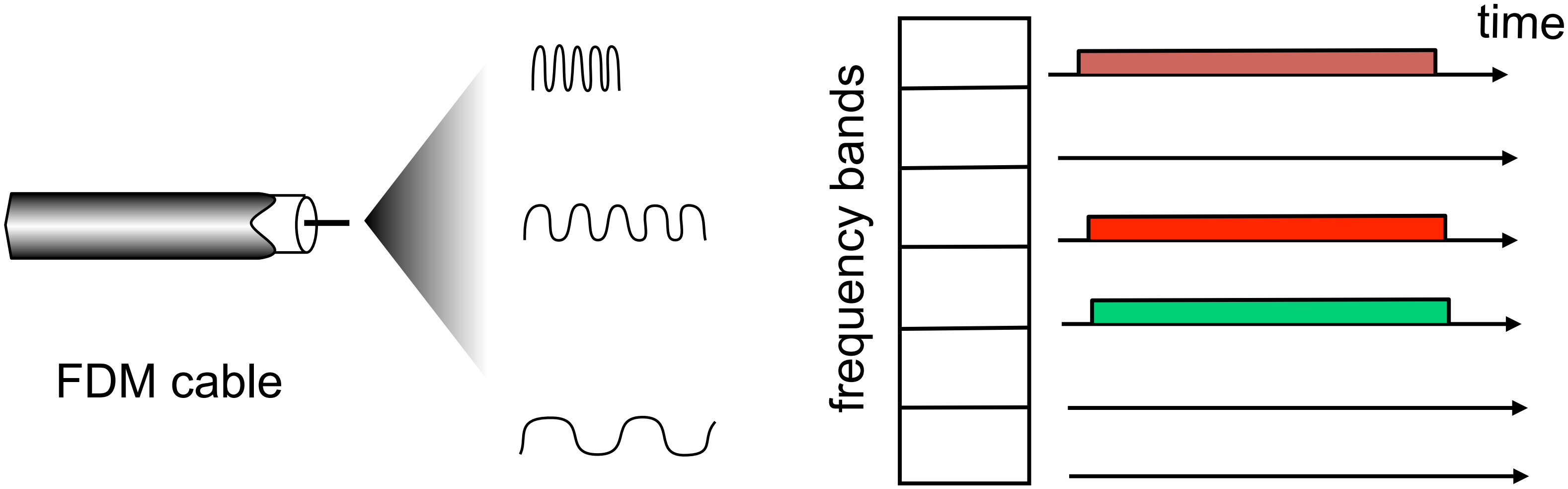
- access to channel in "rounds"
- each station gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle



FDMA

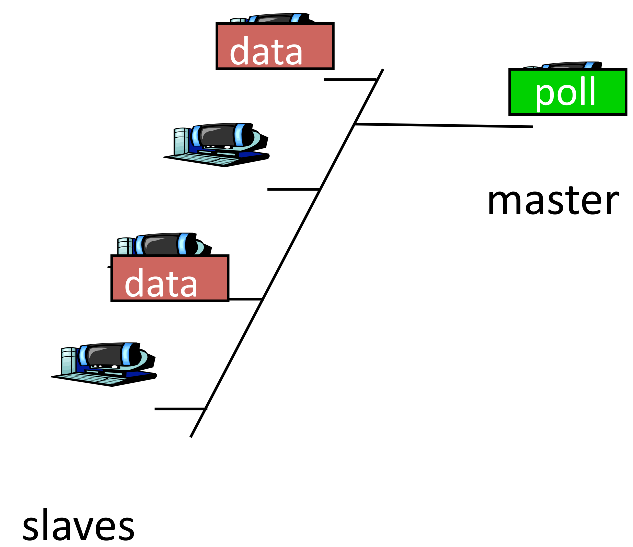
FDMA: frequency division multiple access

- channel spectrum divided into frequency bands
- each station assigned fixed frequency band
- unused transmission time in frequency bands go idle
- example: 6-station LAN, 1,3,4 have pkt, frequency bands 2,5,6 idle



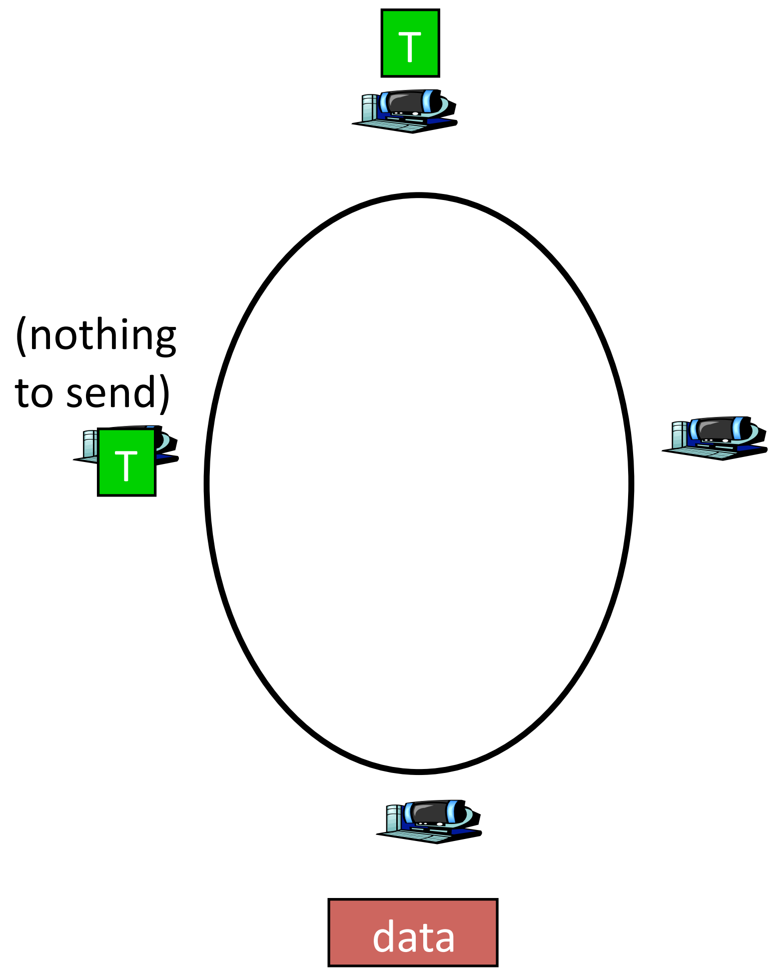
Polling

- Master & slaves
- Master polls each slave
  - Round robin
- Slave responds with transmitted packet
  - if any
  - up to a maximum
- Note:
  - Latency
  - Overhead
  - Single point of failure



Tokens

- Special packet (token) indicates turn
- TDMA with:
  - maximum time per turn
  - ability to pass up turn
- Note:
  - Latency
  - Overhead
  - Network failure



Random access

- No turns!
  - Thus collisions
- Need protocol for telling when it’s your turn
- And for dealing with collisions
- Many, many ways to do this …

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

Slotted ALOHA

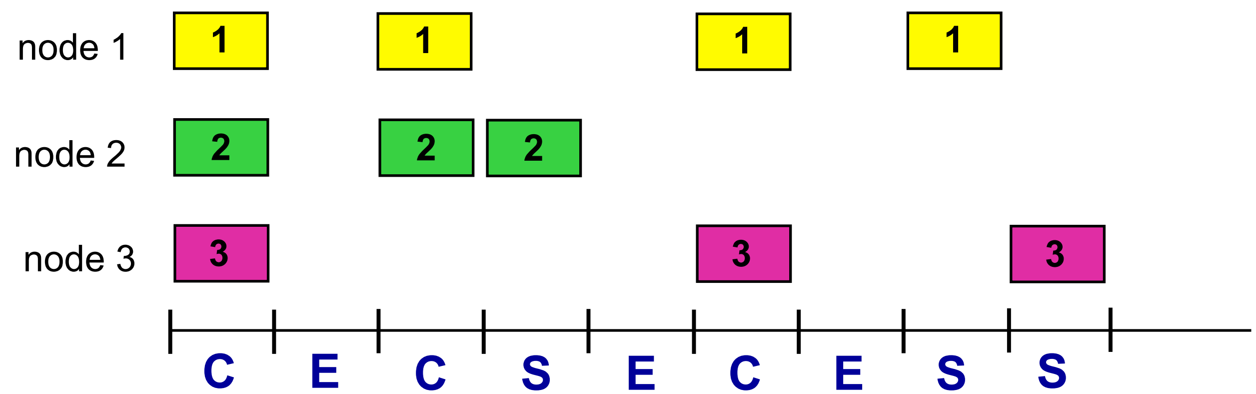
assumptions:

- all frames same size
- time divided into equal size slots (time to transmit 1 frame)
- nodes start to transmit only slot beginning
- nodes are synchronized
- if 2 or more nodes transmit in slot, all nodes detect collision

operation:

- when node obtains fresh frame, transmits in next slot
  - *if no collision:* node can send new frame in next slot
  - *if collision:* node retransmits frame in each subsequent slot with prob. p until success

Slotted ALOHA



Pros:

- single active node can continuously transmit at full rate of channel
- highly decentralized: only slots in nodes need to be in sync
- simple

Cons:

- collisions, wasting slots
- idle slots
- nodes may be able to detect collision in less than time to transmit packet
- clock synchronization

Slotted ALOHA: efficiency

efficiency: long-run fraction of successful slots (many nodes, all with many frames to send)

- *suppose: N* nodes with many frames to send, each transmits in slot with probability *p*
- prob that given node has success in a slot = $p(1-p)^{N-1}$
- prob that *any* node has a success = $Np(1-p)^{N-1}$

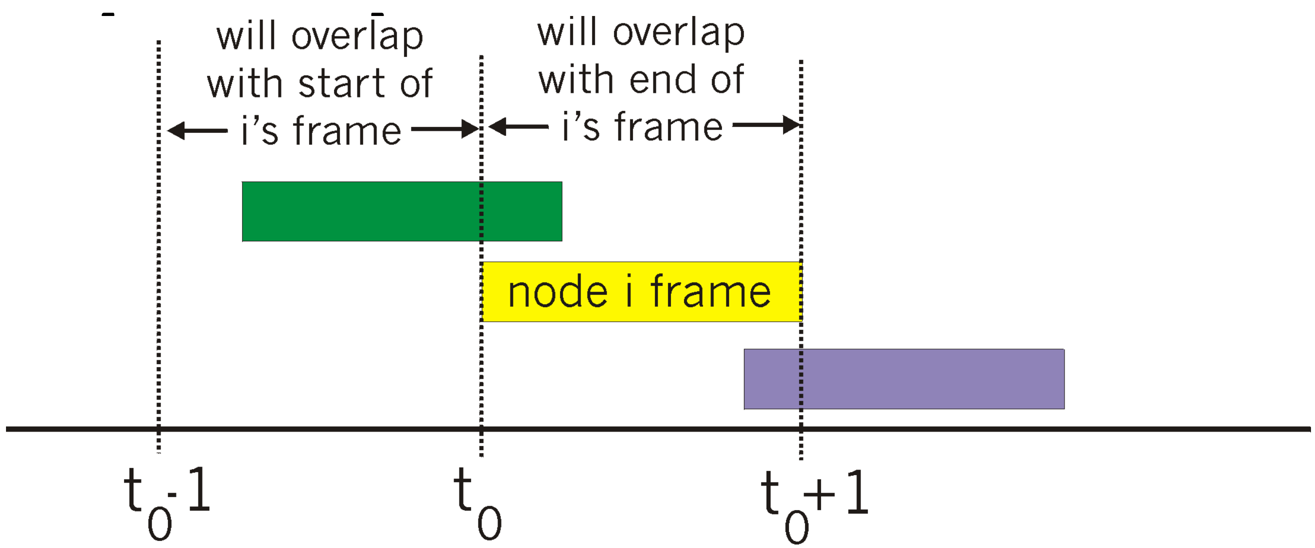
- max efficiency: find p* that maximizes $Np(1-p)^{N-1}$
- for many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives:

max efficiency = 1/e = .37

at best: channel used for useful transmissions 37% of time!

Pure (unslotted) ALOHA

- unslotted Aloha: simpler, no synchronization
- when frame first arrives
  - transmit immediately
- collision probability increases:
  - frame sent at t_0 collides with other frames sent in $[t_0-1, t_0+1]$



Pure ALOHA efficiency

P(success by given node) = P(node transmits) $\cdot$

P(no other node transmits in $[t_0\text{-}1, t_0]$ $\cdot$

P(no other node transmits in $[t_0\text{-}1, t_0]$

$= p \cdot (1-p)^{\mathbf{N-1}} \cdot (1-p)^{\mathbf{N-1}}$

$= p \cdot (1-p)^{\mathbf{2(N-1)}}$

… choosing optimum p and then letting $n \longrightarrow \infty$

= 1/(2e) = .18

even worse than slotted Aloha!

CSMA (carrier sense multiple access)

CSMA: listen before transmit:

if channel sensed idle: transmit entire frame

- if channel sensed busy, defer transmission

- collisions *can* still occur: propagation delay means two nodes may not hear each other transmission
- collision: entire packet transmission time wasted
  - distance & propagation delay play role in in determining collision probability

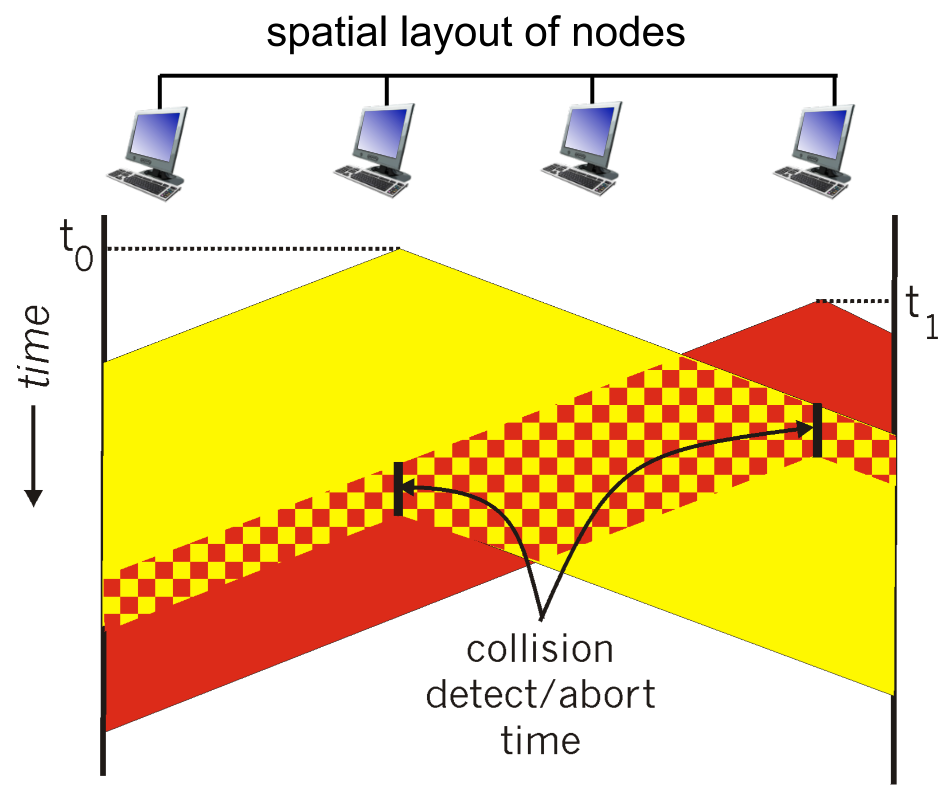
CSMA/CD (collision detection)

CSMA/CD: carrier sensing, deferral as in CSMA

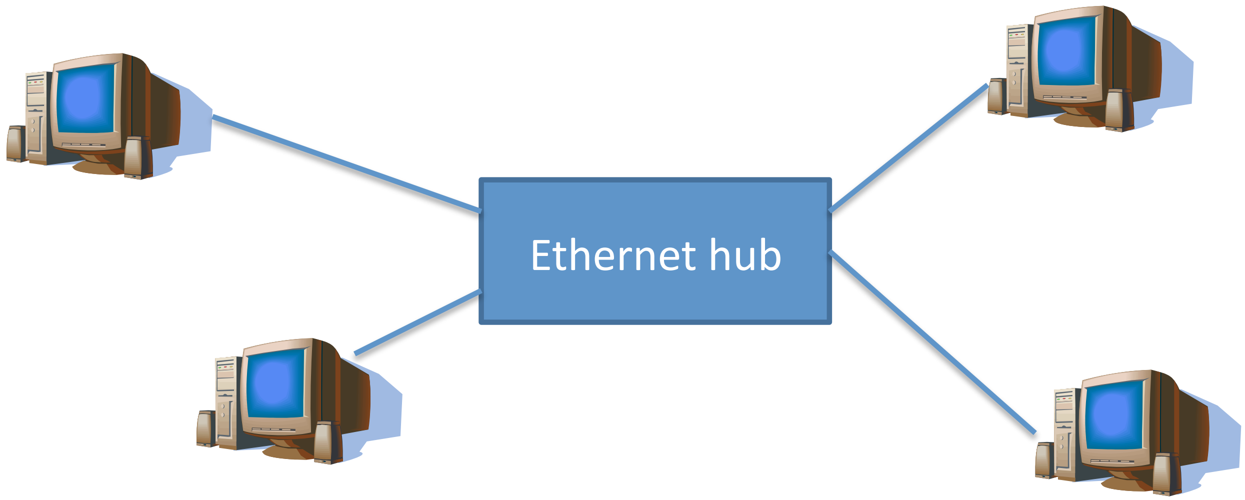
- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage

- collision detection:
  - easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - difficult in wireless LANs: received signal strength overwhelmed by local transmission strength

CSMA/CD (collision detection)



Shared Ethernet



- Wired version of ALOHA
- "Star" topology with Ethernet hub at the center.
- Hub receives all packets and retransmits them onto all lines except where they came from.

Shared Ethernet MAC

- Computers are smarter than in ALOHA:
  - Carrier Sensing (CS): start to transmit only if you don't hear anyone else talking
  - Collision Detection (CD): stop transmitting if you hear a collision.
- Hub operation:
  - Copies bits from one link to all other links, at same rate
  - No buffering
  - Hub doesn't do CS or CD

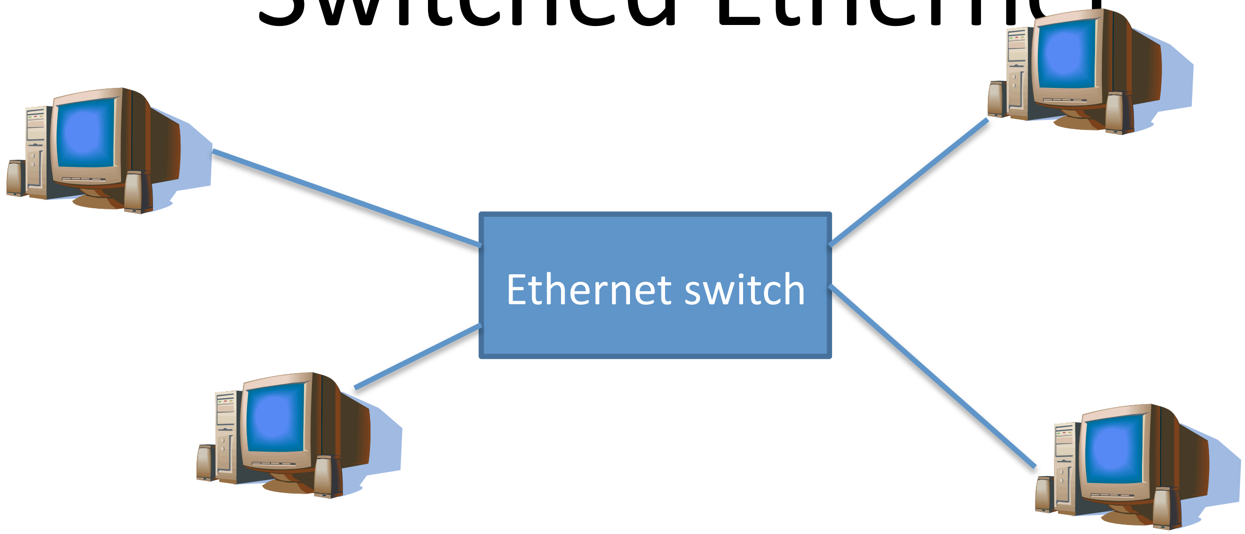
Shared Ethernet MAC

- Protocol:
  - (1) If you have a packet to transmit, wait until the line is idle (CS).
  - (2) While transmitting, continue to listen.
  - (3) If collision, stop transmitting packet, transmit “jam” signal, wait random delay, and go back to step (1).

Shared Ethernet Retransmission

- Retransmission delay:
  - How long should the random delay be?
  - Dynamic: be more patient if the network is overloaded
  - Implementation: Increase the random delay (on average) the more times the packet collides.
- Define "mini-slot" = 2 T_{prop}, where T_{prop} = max prop delay between adapters
  - Upon 1st collision, randomly choose among {0,1} mini-slot delay
  - Upon 2nd collision, randomly choose among {0,1,2,3} mini-slot delay
  - …
  - Upon 10th-15th collision, randomly choose among {0,1,…,1023} mini-slot delay
  - Upon 16th collision, give up
- Called "truncated binary Exponential backoff"

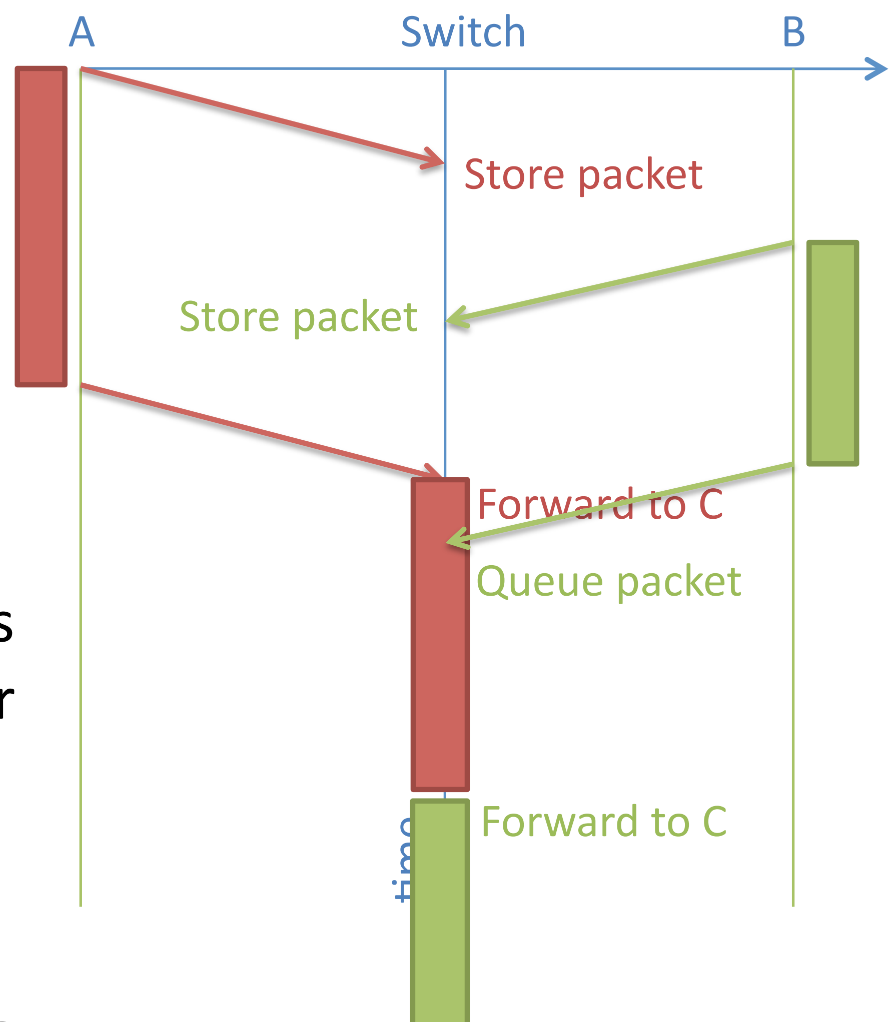
Switched Ethernet



- Same as Shared Ethernet, but switch is smarter than hub.
- Switch can receive packets from multiple lines at the same time. (No collisions!)
- Switch determines where the packet should go, and only sends it on that line.
- Switch can also transmit packets on multiple (different) lines at the same time.
- Switch queues packets as needed.
- Switch drops (throws away) packets when its queue fills up.

Switch Timing

- If *A* and *B* both transmit to *C*, what happens?
- Normally switch starts forwarding a packet only after completely receiving it.
  - Cut-through switch starts forwarding a packet after completely receiving its header.



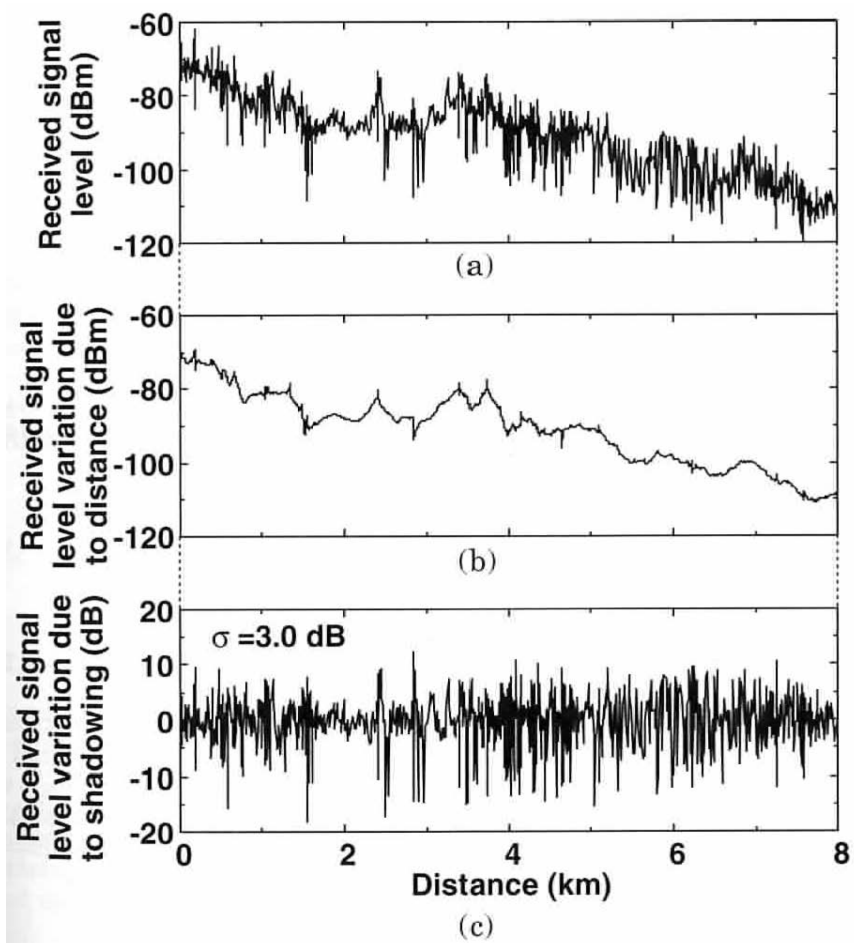
Wireless Challenges

- Physical layer
  - variability
  - power allocation
  - rate allocation
- MAC layer
  - topology
  - collisions
  - mobility

Wireless channel characteristics

- Attenuation
  - signal gets weaker with distance
- Multipath
  - signal bounces off stuff and creates multiple copies
  - that arrive at different times
- Noise
  - background noise
- Interference
  - users transmitting on the same channel

Wireless channel variability



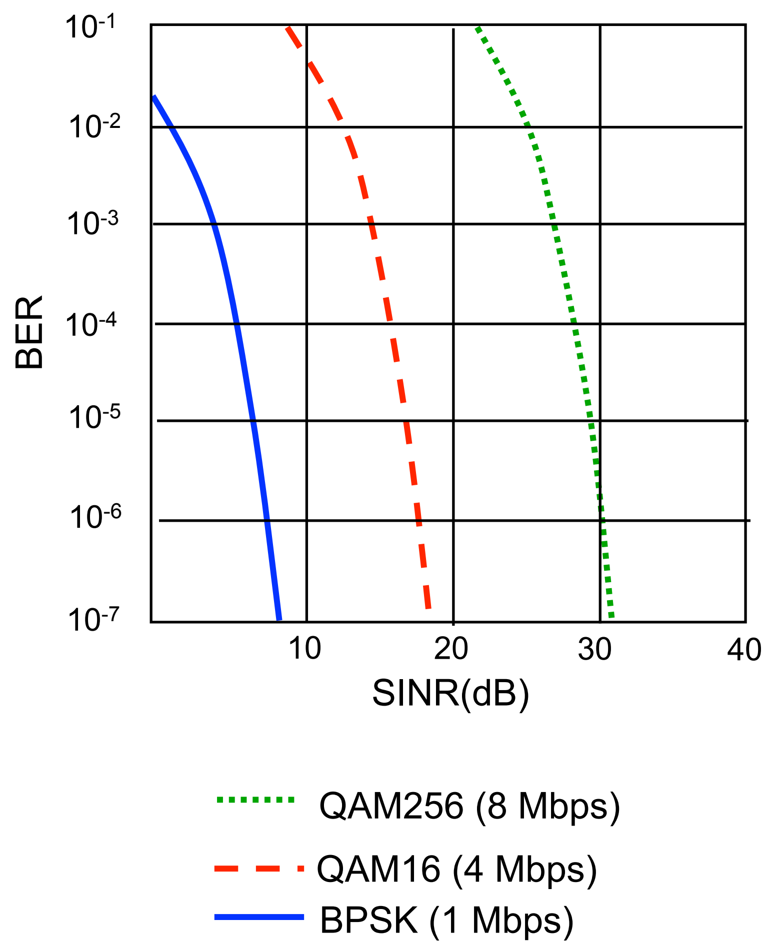
Sampei fig. 2.2

Radio Resource Allocation

- Power allocation
  - frequently modify uplink and downlink power
  - to combat attenuation, multipath, noise, interference
- Rate allocation
  - sometimes modify uplink and/or downlink rate
  - when power allocation isn't enough

Modulation, power & rate allocation

- "SINR" = signal-to-interference-plus-noise-ratio
- "BER" = bit error rate
- Higher rates usually require higher SINR for same BER
  - Higher SINR comes from higher transmission power.



Kurose fig. 6.3

Bluetooth MAC

- Service:
  - Short range
  - Relatively dumb devices
  - Low data rate
  - No infrastructure
- Architecture:
  - hybrid TDMA/CDMA to define channels
  - Master / slave
  - Master polls slaves for transmissions

Wi-Fi

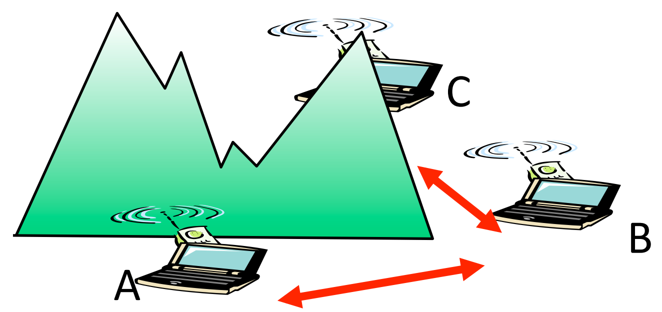
- Service:
  - Short range
  - Relatively smart devices
  - High data rate
  - Infrastructure

Wi-Fi topology

- Infrastructure mode: "base station" or "wireless access point" replaces wired hub or switch
- "link" given by who can hear whom
  - really given by signal strength, i.e. SINR
- 1st problem: a device may be able to hear more than one base station or wireless access point

Wi-Fi topology

- 2nd problem: devices can hear each other
- 3rd problem: devices within “hearing distance” of a base station or access point can’t necessarily hear each other (“hidden terminal problem”)



Kurose fig. 6.4

Wi-Fi access point selection

- 1st problem: a device may be able to hear more than one wireless access point
  - which access point are you transmitting to?
- Solution part 1: try to assign different frequencies to nearby access points
  - requires coordination
- Solution part 2: association
  - access point advertises availability (SSID + MAC)
  - device (or user) selects base station

Wi-Fi CS

- 2^{nd} problem: devices can hear each other
  - collisions happen when & where?
- A collision happens
  - if the access point or a device receives more than one transmission at the same time (overlapping packets) [also depends on SINRs]
- Partial solution: devices use Carrier Sensing (CS) [as in Shared Ethernet]
  - doesn’t prevent all collisions

Wi-Fi CD (not)

- 3rd problem: devices within "hearing distance" of a wireless access point can't necessarily hear each other
  - do you even know whether a collision has occurred?
- Shared Ethernet uses Collision Detection (CD) plus jam signals
- Wi-Fi doesn't use CD or jam signal
- Wi-Fi uses Collision Avoidance (CA) instead

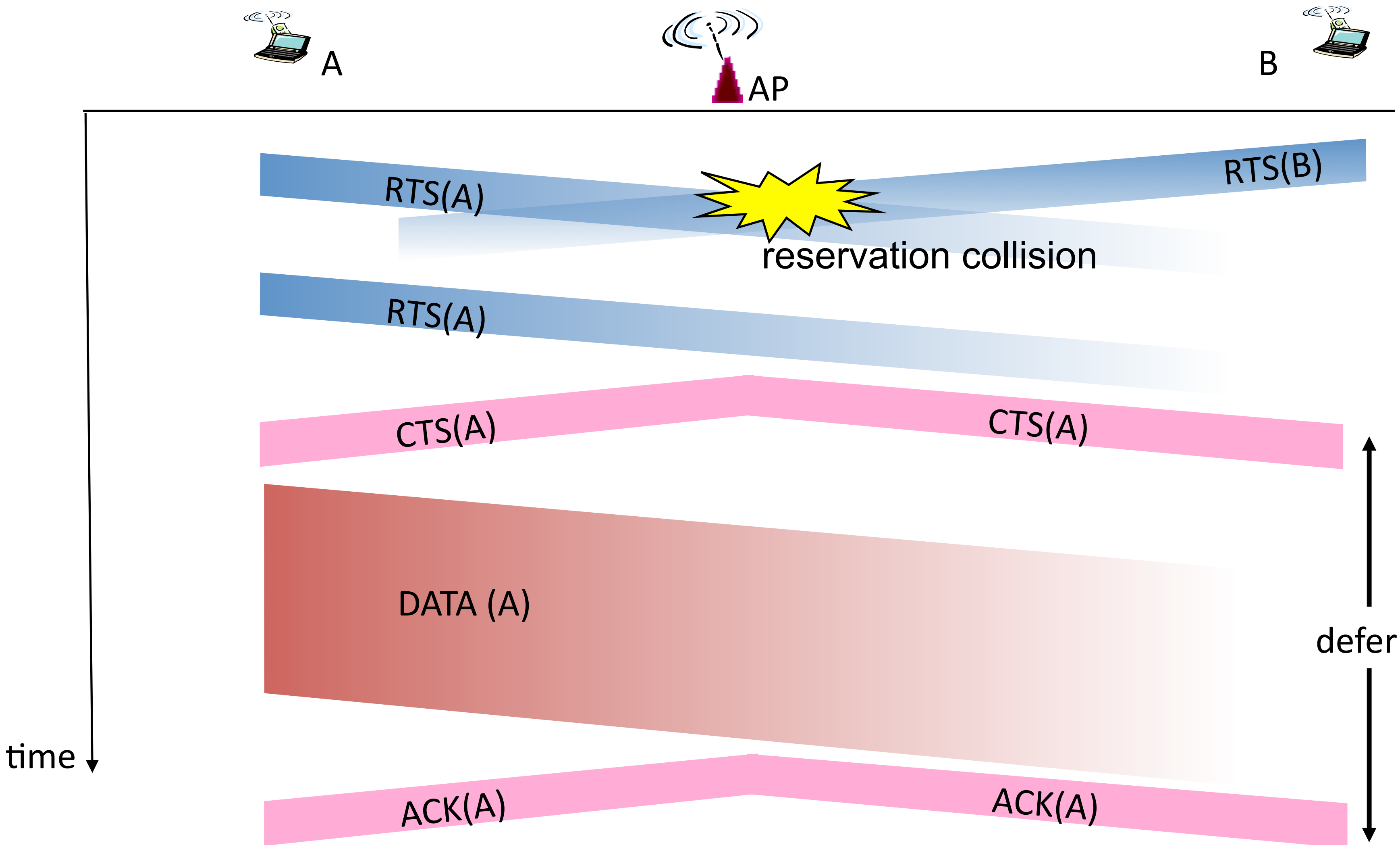
Wi-Fi CA

- Collision Avoidance method 1:
  - Use backoff in conjunction with CS (not with CD)
  - Shared Ethernet: upon CD, use backoff (truncated binary Exponential) to determine when to try again
  - Wi-Fi: upon CS indicating busy, use backoff (truncated binary Exponential) to determine when to try again
- Doesn’t prevent all collisions

Wi-Fi CA

- Collision Avoidance method 2:
  - Indicate desire to transmit a packet
    - “Request-to-transmit” (RTS)
    - RTS is a small packet
    - Use CS to determine when to send RTS
  - If RTS doesn’t collide, access point gives permission
    - “Clear-to-transmit” (CTS)
  - Device that reserved transmits user packet

Wi-Fi CA



Kurose fig. 6.12