Computer Networks COL 334/672

Link Layer

Tarun Mangla

Slides adapted from KR

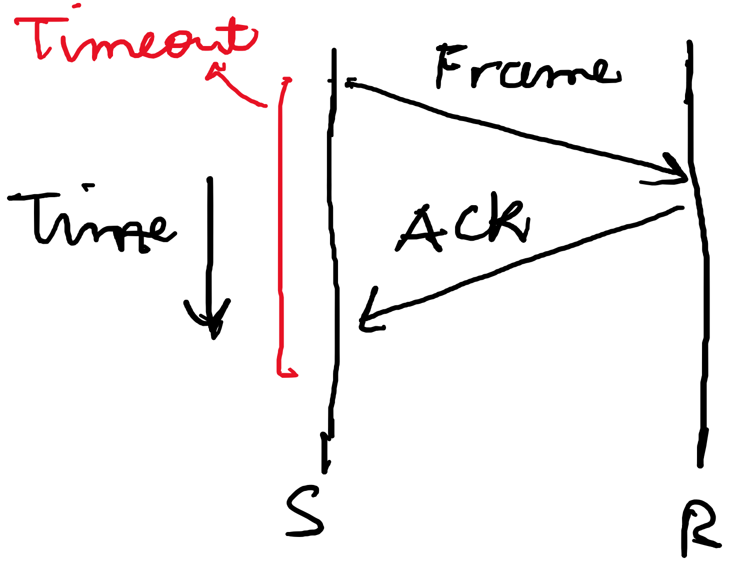
Sem 1, 2024-25

Link Layer: Services

- Framing
- Error detection
- **Reliability**
- Link access

ARQ Protocol: Stop and Wait

- Transmit one frame, wait for an acknowledgement
  - If no ack and timer expires, resend



Stop and Wait

- Transmit one frame, wait for an acknowledgement
  - If no ack and timer expires, resend
- How to handle duplicate frames?
  - Sequence numbers for duplicate frames

Stop and Wait

Bandwidth delay product ↳ Bandwidth × delay / Frame size

- Transmit one frame, wait for an acknowledgement
  - If no ack and timer expires, resend
- How to handle duplicate frames?
  - Sequence numbers for duplicate frames
- Any limitation?
  - Under-utilization of link
    - Example, 4 Mbps link, RTT – 10ms, Frame size – 1 KB. What is the link utilization? 1KB/20ms = 5Kbps
  - How to achieve higher link utilization?
    - Allow sending more than one unacknowledged packets
  - How many packets to get maximum utilization?
    - Bandwidth delay product

Bandwidth

delay / RTT

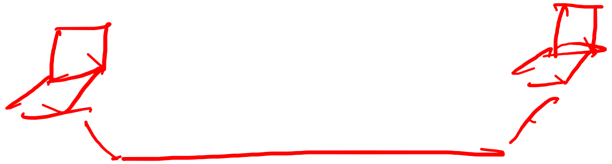
High BDP network

Link Layer: Services

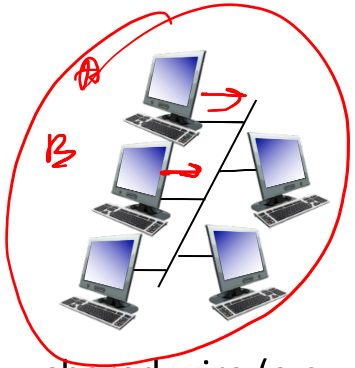
- Framing
- Error detection
- Reliability
- **Link access**

Multiple access links, protocols

two types of "links":



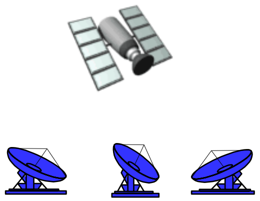
- point-to-point
  - point-to-point link between Ethernet switch, host
  - PPP for dial-up access
- broadcast (shared wire or medium)
  - old-school Ethernet
  - upstream HFC in cable-based access network
  - 802.11 wireless LAN, 4G/4G. satellite



shared wire (e.g., cabled Ethernet)

shared radio: 4G/5G

shared radio: WiFi



shared radio: satellite

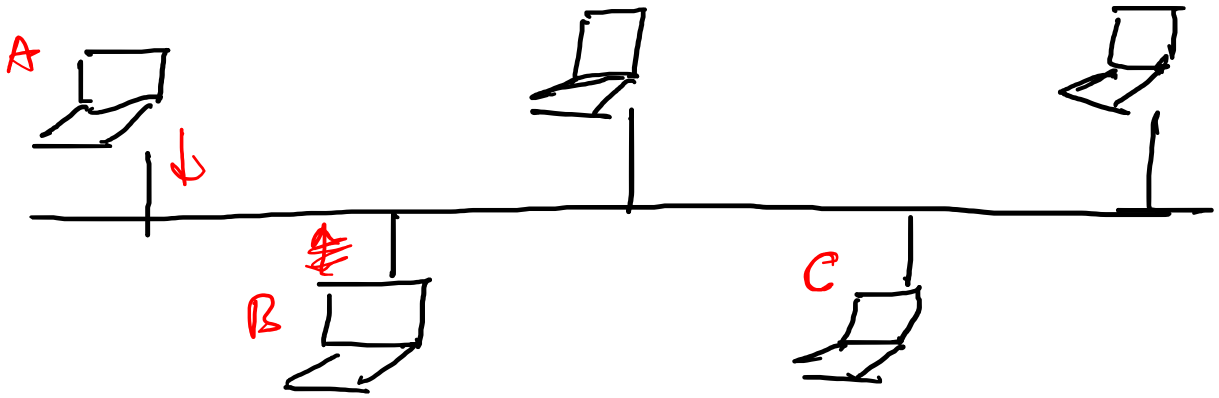
Multiple access protocols

- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference

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 protocol



collision if node receives two or more signals at the same time

Energy of signal > th_1

$th_2 > th_1$

" " " > th_2

↳ collision

FDMA

Frequence Division Multiple Access

TDMA: Time Division Multiple Access

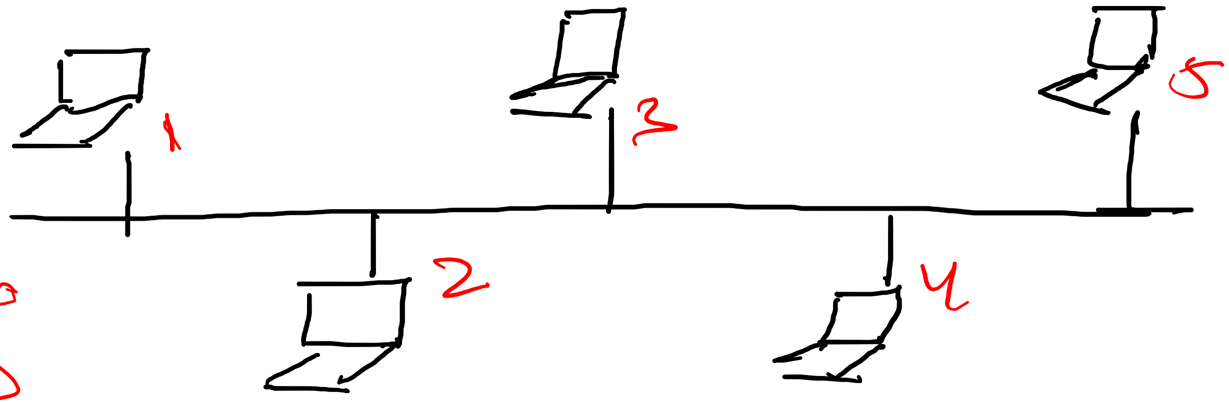
multiple access protocol

- 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

Polling : Taking Turns Protocol

5 slots

Multiple access protocol



SPF

Controlled / host

collision if node receives two or more signals at the same time

③ Simple

④ Decentralized

Ideals of MAC

① Link to be shared equally amongst nodes (fairness)

② If a node has data to send, it should get R bps (utilization should be high)

- **channel partitioning**
  - divide channel into smaller "pieces" (time slots, frequency, code)
  - allocate piece to node for exclusive use
  - Limitation: unused slots go idle
- **random access**
  - channel not divided, allow collisions
  - "recover" from collisions

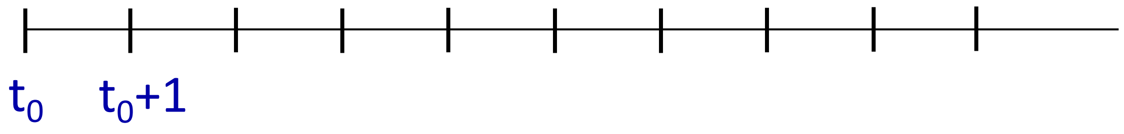
Random access protocols

- 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 protocol specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)
- examples of random access MAC protocols:
  - ALOHA, slotted ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Ethernet

WiFi

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

ACK

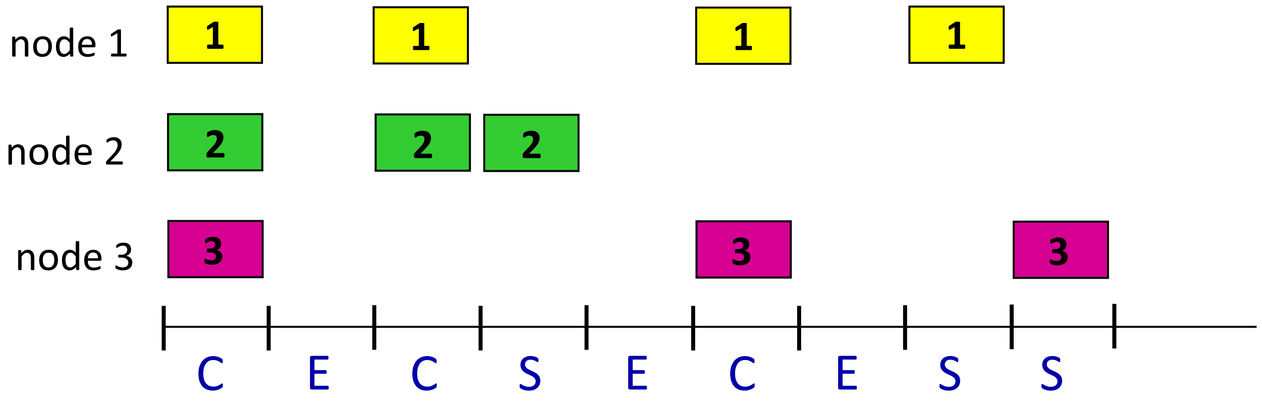
- if 2 or more nodes transmit in slot, all nodes detect collision

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 probability *p* until success

randomization – *why*?

Slotted ALOHA



C: collision
S: success
E: empty

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

$N \underline{\underline{p}} (1-p)^{N-1} \rightarrow 1/N$

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:

$\lim_{N\to\infty} \left(1-\frac{1}{N}\right)^{N-1} = e^{\lim_{N\to\infty} -\left(\frac{N-1}{N}\right)} = 1/e$

max efficiency = 1/e = .37

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

CSMA (carrier sense multiple access)

simple CSMA: listen before transmit:

- if channel sensed idle: transmit entire frame
- if channel sensed busy: defer transmission

CSMA (carrier sense multiple access)

simple CSMA: listen before transmit:

- if channel sensed idle: transmit entire frame
- if channel sensed busy: defer transmission

CSMA/CA

CSMA/CD: CSMA with *collision detection*

- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage
- collision detection easy in wired, difficult with wireless