Link layer, LANs: roadmap

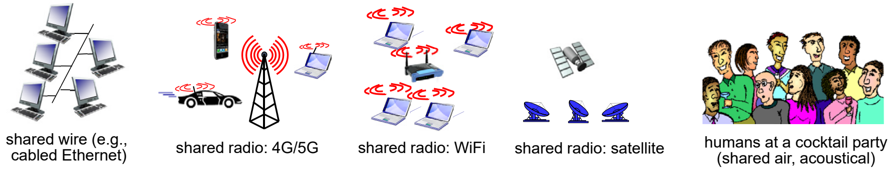
- introduction
- error detection, correction
- **multiple access protocols**
- LANs
  - addressing, ARP
  - Ethernet
  - switches
  - VLANs
- link virtualization: MPLS
- data center networking

- a day in the life of a web request

Multiple access links, protocols

two types of "links":

- point-to-point
  - point-to-point link between Ethernet switch and host
- broadcast (shared wire or medium)
  - old-fashioned Ethernet
  - 802.11 wireless LAN, mobile networks (4G/5G), satellite



Multiple access protocols

- 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 (or centralized) 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

An ideal multiple access protocol

given: multiple access channel (MAC) of capacity *R* bps

desiderate:

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 single point of failure
   - no synchronization of clocks, slots
4. simple

MAC protocols: taxonomy

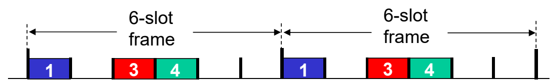
three broad classes:

- channel partitioning
  - divide channel into smaller "pieces" (time slots, frequency, …)
  - allocate piece to node/user 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
    - similar to round robin

Channel partitioning MAC protocols: TDMA

TDMA: time division multiple access

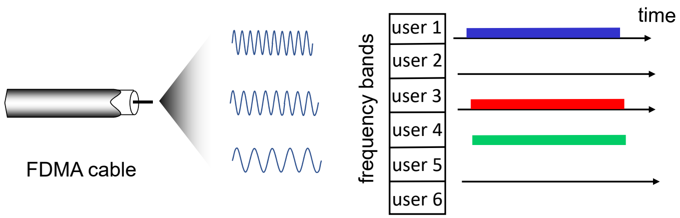
- access to channel in "rounds"
- each station gets fixed length slot (length = packet transmission time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have packets to send, slots 2,5,6 idle



Channel partitioning MAC protocols: 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 packet to send, frequency bands 2,5,6 idle



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:
  - ALOHA, slotted ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

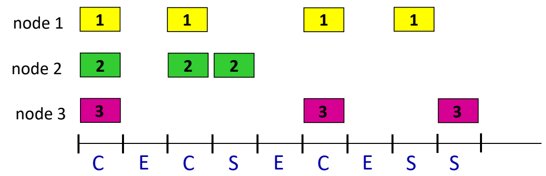
assumptions:

- time is divided into equal-sized slots
  - slot: time to transmit 1 frame
    - all frames have the same size
  - nodes are time-synchronized
  - nodes start to transmit only slot beginning
- if 2 or more nodes transmit in a slot, all nodes detect collision

operation:

- stop-and-wait
- 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

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
- 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

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

Slotted ALOHA: efficiency (cont'd)

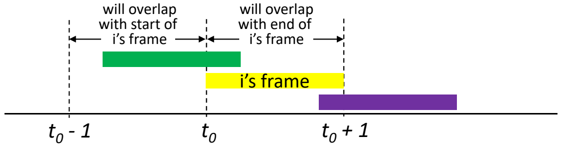
- slotted ALOHA's efficiency = probability that *any* node has a success
  - Given N and p, $f(p) = Np(1-p)^{N-1}$
- max efficiency: find p^* that maximizes $f(p)$
- for many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity:

$$\lim_{N\to\infty} f(p^*) = \lim_{N\to\infty} N \cdot \frac{1}{N} \cdot \left(1 - \frac{1}{N}\right)^{N-1} = \lim_{N\to\infty} \left(\frac{1}{1+\frac{1}{N-1}}\right)^{N-1} = \frac{1}{e}$$

- slotted ALOHA's efficiency is at best $1/e$ = 37%
  - 37% of time is useful transmissions.

Pure ALOHA

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



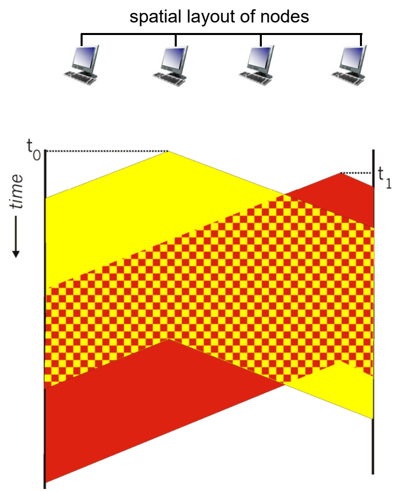
- pure Aloha efficiency: 18% !

CSMA (carrier sense multiple access)

simple CSMA: listen before transmit:
- if channel sensed idle: transmit entire frame
- if channel sensed busy: defer transmission
- human analogy: don't interrupt others!

CSMA: collisions

- collisions *can* still occur if two nodes send frames around the same time
  - propagation delay means two nodes may not hear each other's just-started transmission
- collision: entire packet transmission time wasted
  - distance & propagation delay play role in in determining collision probability

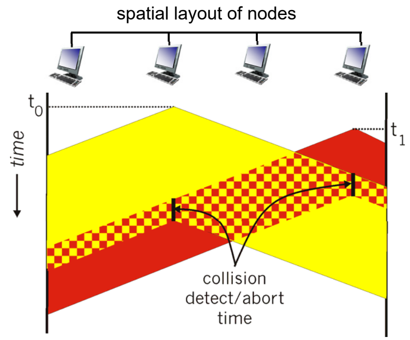


CSMA/CD

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
- human analogy: the polite conversationalist

CSMA/CD

- CSMA/CS reduces the amount of time wasted in collisions
  - transmission aborted on collision detection



Ethernet CSMA/CD algorithm

1. NIC receives datagram from network layer, creates frame
2. NIC senses channel:
   if idle: start frame transmission.
   if busy: wait until channel idle, then transmit
3. If NIC transmits entire frame without collision, NIC is done with frame !
4. If NIC detects another transmission while sending: abort, send jam signal
5. After aborting, NIC enters *binary (exponential) backoff:*
   - after m-th collision, NIC chooses K at random from $\{0,1,2, \ldots, 2^m-1\}$. NIC waits K slot times, returns to Step 2. (1 slot = minimum frame size = 64 bytes for 10/100 Mbps or 512 bytes for 1Gbps)
   - more collisions: longer backoff interval

CSMA/CD efficiency

- $t_{prop} \triangleq$ maximum propagation delay between 2 nodes in LAN
- $t_{trans} \triangleq$ time to transmit max-size frame
- efficiency is (approximately)

$$\rho \doteqdot \frac{1}{1 + 5 \cdot \frac{t_{prop}}{t_{trans}}}$$

  - efficiency goes to 1 as t_{prop}/t_{trans} goes to 0
- CSMA/CD
  - better efficiency than ALOHA
  - simple, cheap, decentralized

"Taking turns" MAC protocols

channel partitioning MAC protocols:
- share channel *efficiently* and *fairly* at high load
- inefficient at low load: 1/N bandwidth allocated even if only 1 active node!

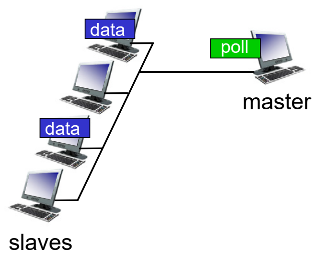
random access MAC protocols
- efficient at low load: single node can fully utilize channel
- high load: collision overhead

"taking turns" protocols
- look for best of both worlds!

“Taking turns” MAC protocols

polling:

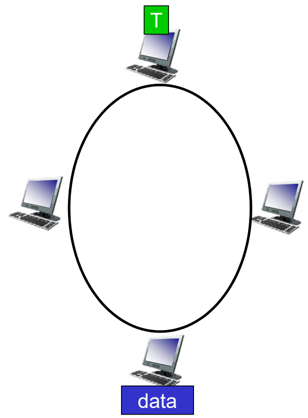
- master node “invites” other nodes to transmit in turn
- typically used with “dumb” devices
- concerns:
  - polling overhead
  - latency
  - single point of failure (master)



“Taking turns” MAC protocols

token passing:

- control *token* passed from one node to next sequentially.
- token message
- concerns:
  - token overhead
  - latency
  - single point of failure (token)



Summary of MAC protocols

- channel partitioning, by time, frequency or code
  - Time Division, Frequency Division
- random access (dynamic),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - carrier sensing: easy in some technologies (wired), hard in others (wireless)
  - CSMA/CD used in Ethernet
  - CSMA/CA used in 802.11
- taking turns
  - polling from central site, token passing
  - Bluetooth, FDDI, token ring