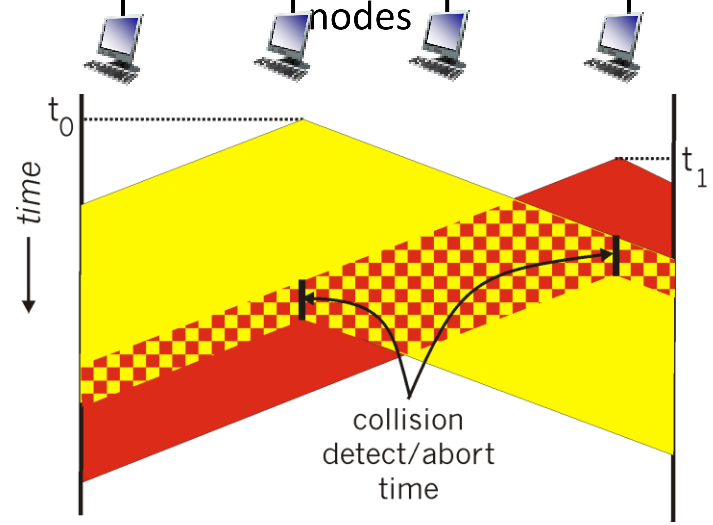
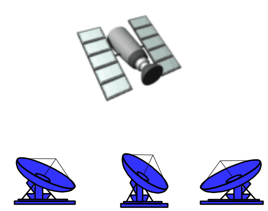
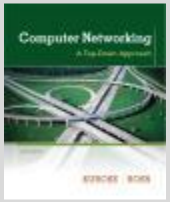
Media access control (MAC)





Computer Networking: A Top Down Approach
6th edition
Jim Kurose, Keith Ross
Addison-Wesley



Link layer, LANs: outline

5.1 Introduction, services

5.2 Error detection, correction

5.3 Multiple access protocols

5.4 LANs
- Addressing, ARP
- Ethernet
- Switches
- VLANS

5.5 Link virtualization: MPLS

5.6 Data center networking

5.7 A day in the life of a web request

MAC sublayer

- Media Access Control (MAC) sublayer
  - Who goes next on a shared medium
- Example:
  - Ethernet hosts can sense if medium in use
  - Algorithm for sending data:
    1. Is medium idle? If not, wait.
    2. Start transmitting data, listen for collision.
    3. If collision detected, transmit 32-bit jamming sequence. Stop transmitting and go to backoff procedure.

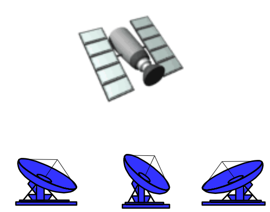
Multiple access links, protocols

Two types of links:

- Point-to-point
  - PPP for dial-up access
  - Point-to-point link between Ethernet switch, host
- *Broadcast (shared wire or medium)*
  - Old-fashioned Ethernet
  - Upstream HFC (cable modem)
  - 802.11 wireless LAN

Shared wire (e.g., cabled Ethernet)

Shared RF (e.g., 802.11 WiFi)



Shared RF (satellite)

Humans at a cocktail party (shared air, acoustical)

Multiple access protocols

- Single shared broadcast channel
- 2+ simultaneous transmissions by nodes: interference
  - *Collision* if node receives 2+ signals at the same time

Multiple access protocol

- Distributed algorithm, determines how nodes share channel
  - When can a node transmit?
- Communication about channel sharing must use channel itself!
  - No out-of-band channel for coordination

An ideal multiple access protocol

Given: Broadcast channel of rate R bps

Desiderata:

1. When one node wants to transmit, 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

MAC protocols: taxonomy

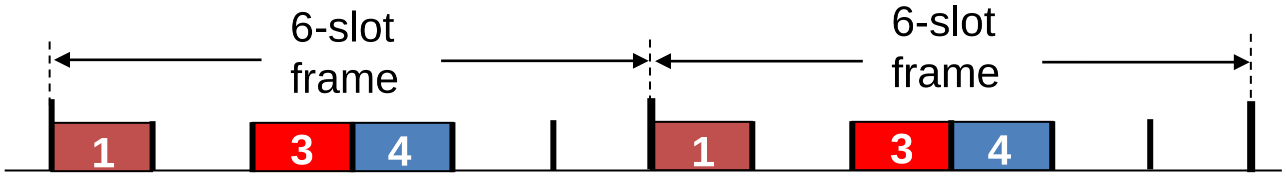
Three broad classes:

- *Channel partitioning*
  - Divide channel into smaller pieces (time slots, frequency, code)
  - 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

Channel partitioning, TDMA

TDMA: Time Division Multiple Access

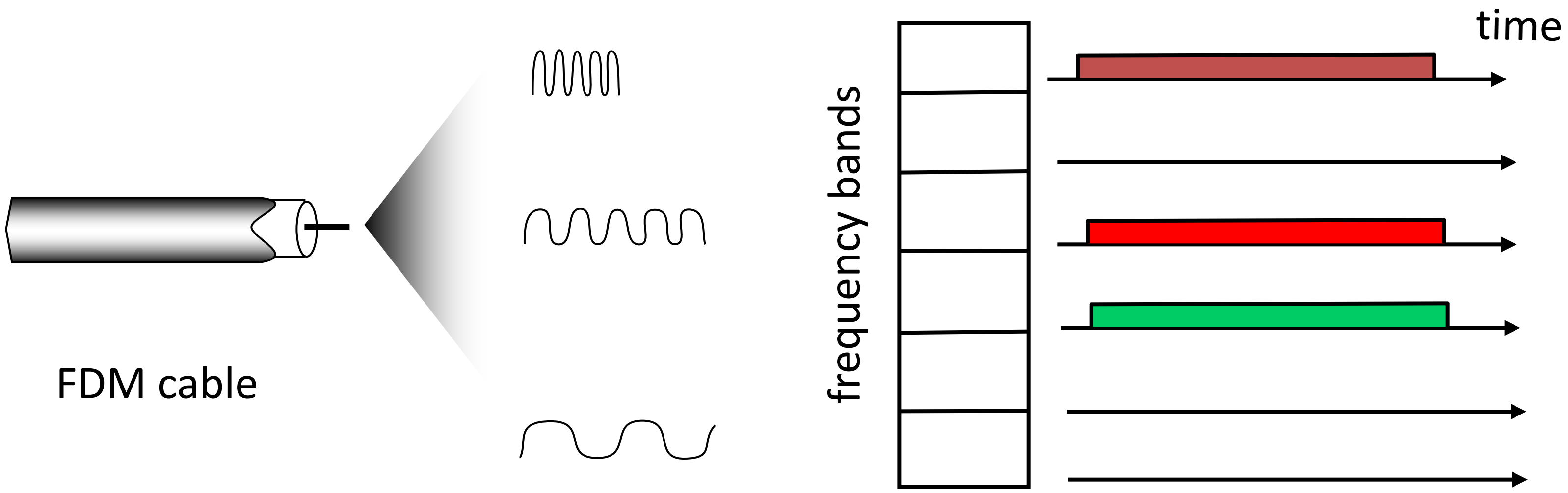
- Access to channel in rounds
- Each station gets fixed length slot
  - Length = packet transmission time in each round
- Unused slots go idle
- Example:
  - 6-station LAN: 1,3,4 have packet, slots 2,5,6 idle



Channel partitioning, FDMA

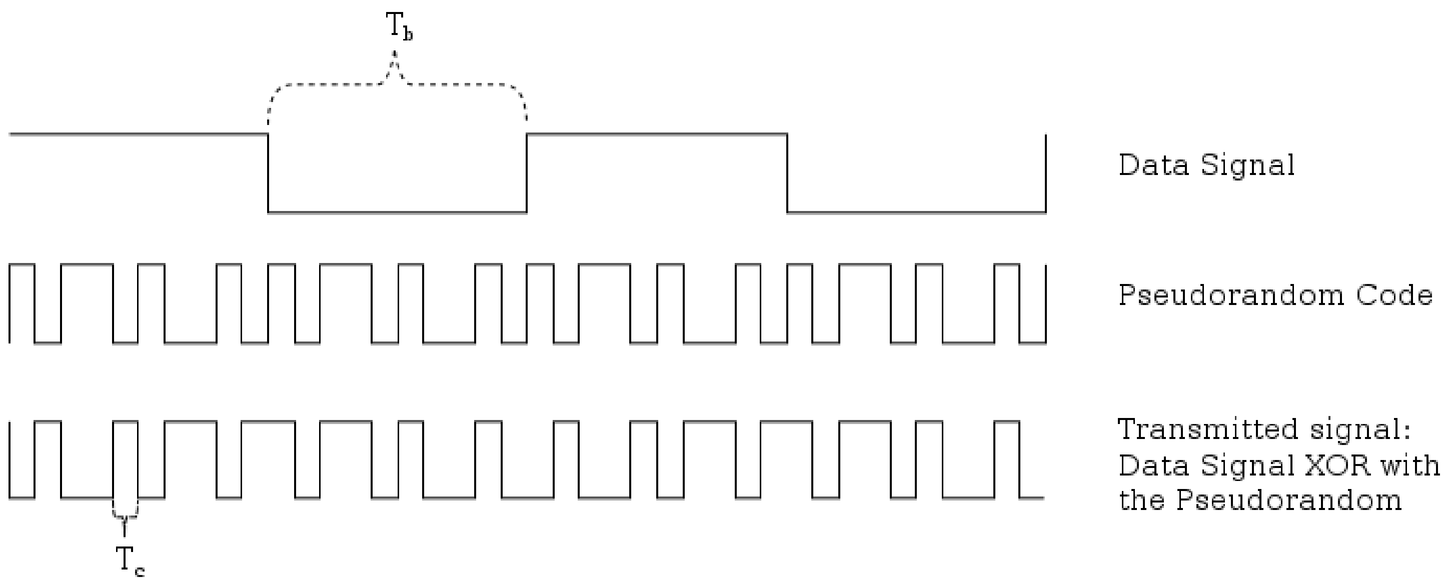
FDMA: frequency division multiple access

- 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, bands 2,5,6 idle



Code Division Multiple Access, CDMA

- Unique code assigned to each user
  - All users share same frequency, but each user has own *chipping* sequence to encode data
  - Allows multiple users to coexist and transmit simultaneously with minimal interference
- *Encoded signal* = (original data) x (chipping sequence)
- *Decoding:* inner-product of encoded signal and chipping sequence



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

Slotted ALOHA

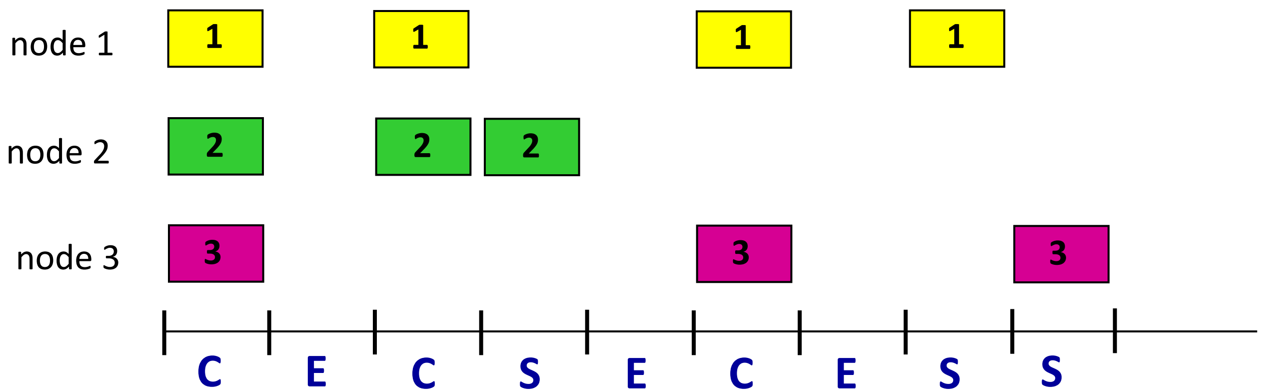
Assumptions:

- All frames same size
- Time divided into equal size slots (time to transmit 1 frame)
- Nodes start to transmit only at slot beginning
- Nodes are synchronized
- If 2+ 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:

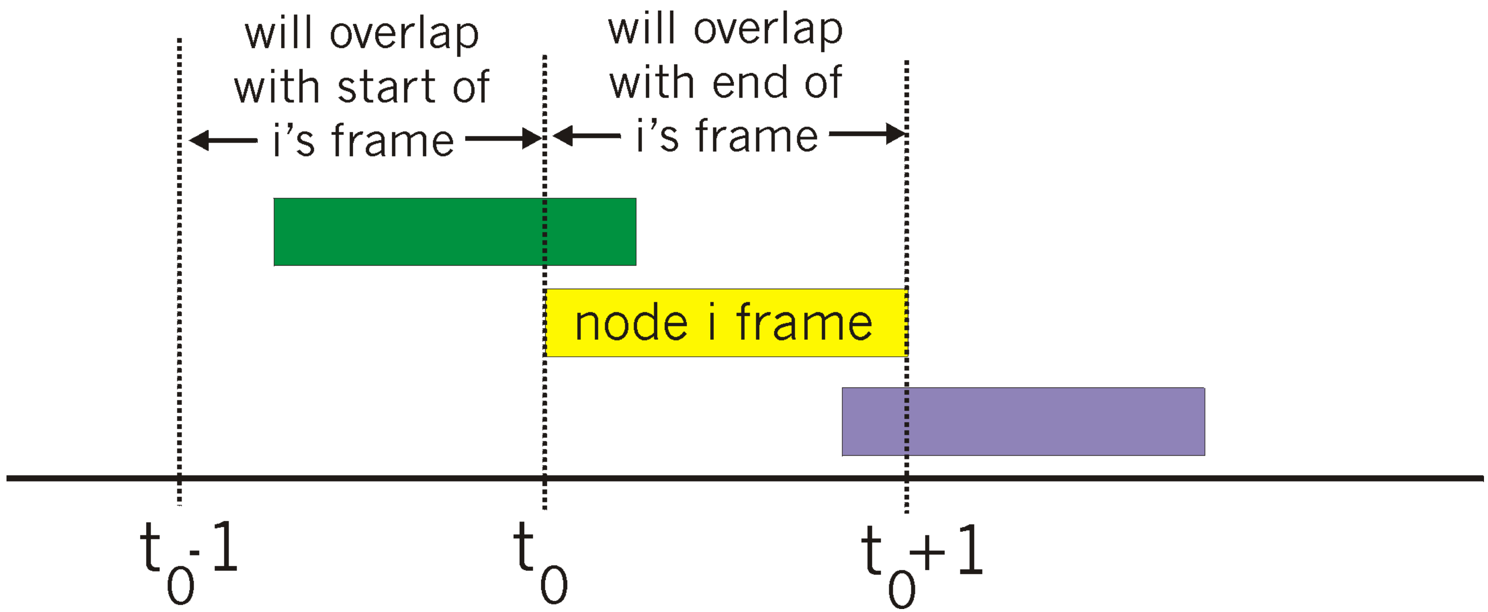
  Max efficiency = 1/e = .37

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 $[t_0-1,t_0]$ $\cdot$

P(no other node transmits in $[t_0-1,t_0]$

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

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

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

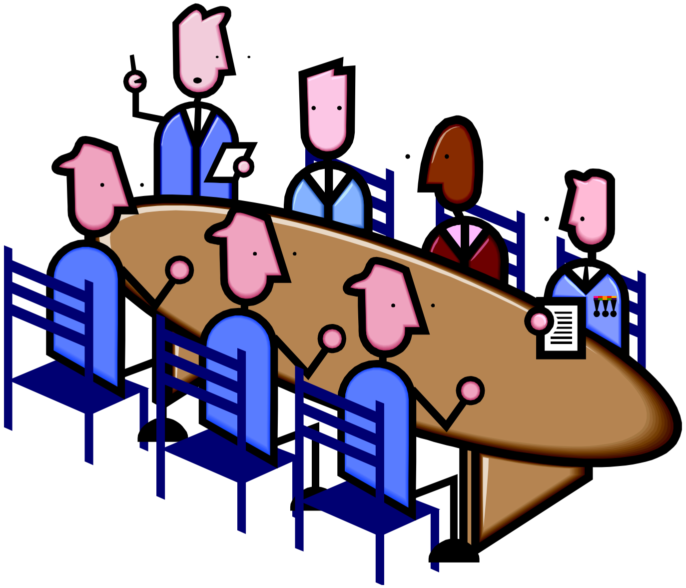
$$= 1/(2e) = .18$$

even *worse* than slotted Aloha!

Carrier Sense Multiple Access (CSMA)

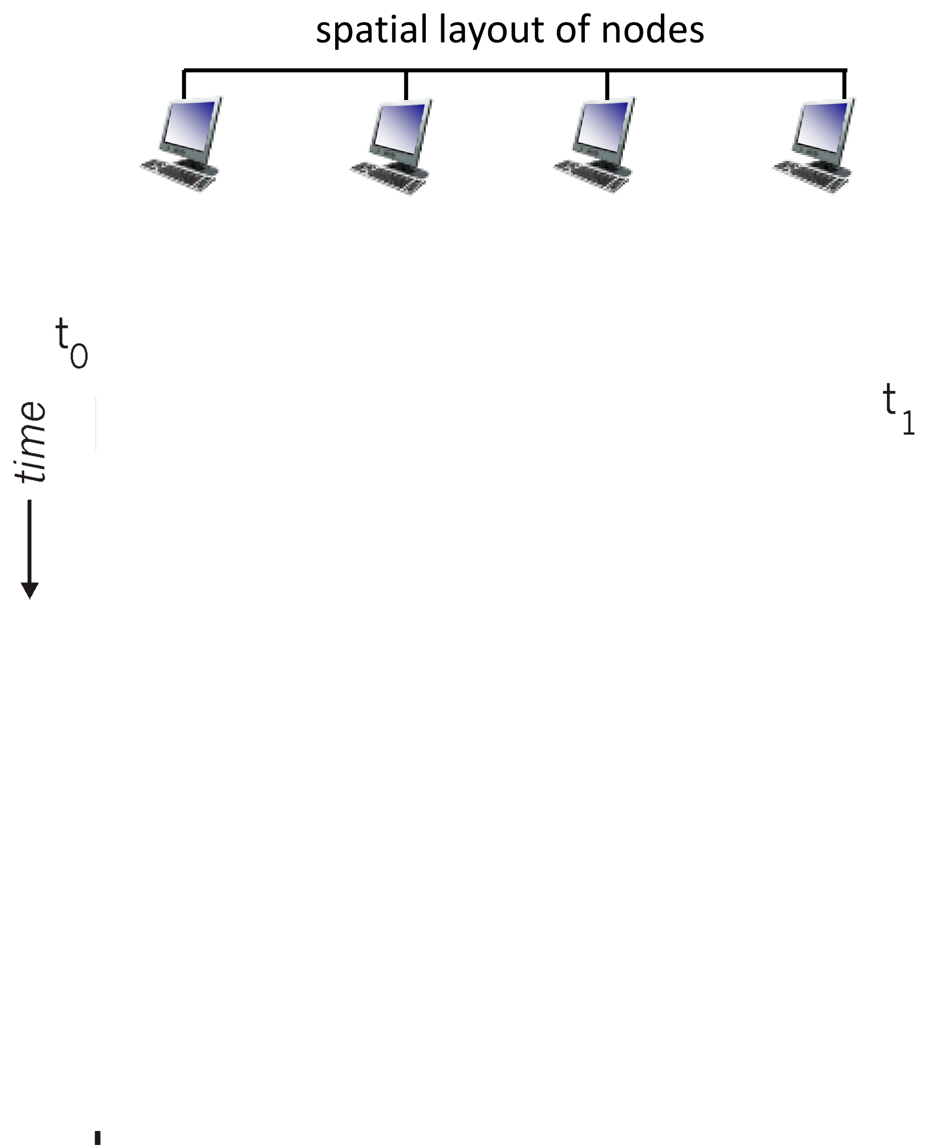
CSMA: Listen before transmit

- 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:
  - Propagation delay means two nodes may not hear each other's 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

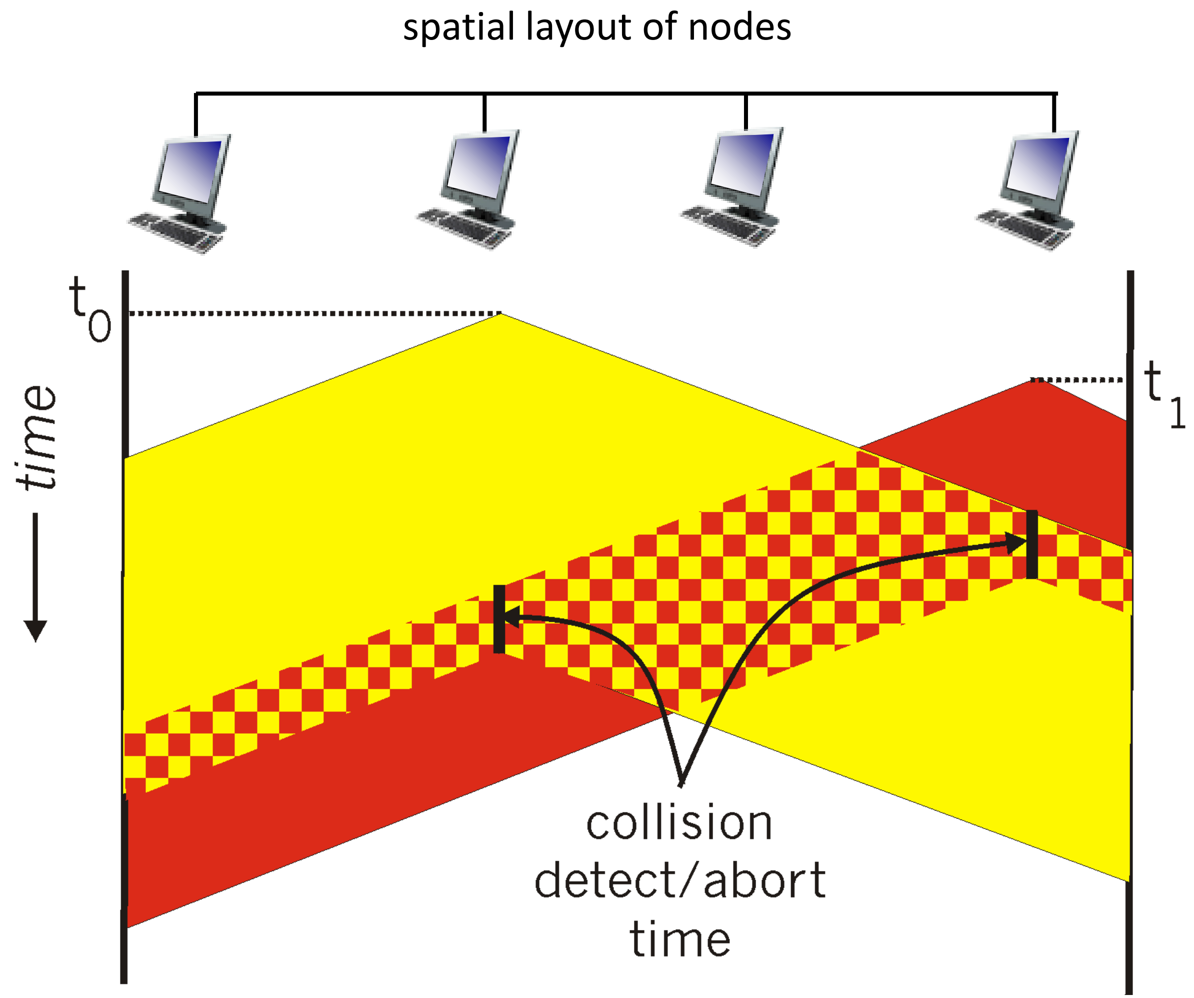
- Collisions *detected* within short time
- Colliding transmissions aborted, reducing channel wastage

❖ 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

❖ Human analogy: the polite conversationalist

CSMA/CD (Collision Detection)



Ethernet CSMA/CD algorithm

1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel idle, starts frame transmission. If NIC senses channel busy, waits until channel idle, then transmits.
3. If NIC transmits entire frame without detecting another transmission, NIC is done with frame!
4. If NIC detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, NIC enters *binary (exponential) backoff:*
   - After *m*th collision, NIC chooses *K* at random from *{0,1,2, ..., 2^m-1}*. NIC waits K·512 bit times, returns to Step 2
   - Longer backoff interval with more collisions

CSMA/CD efficiency

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

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

- Efficiency goes to 1
  - As t_{prop} goes to 0
  - As t_{trans} goes to infinity
- Better performance 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: delay in channel access, 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

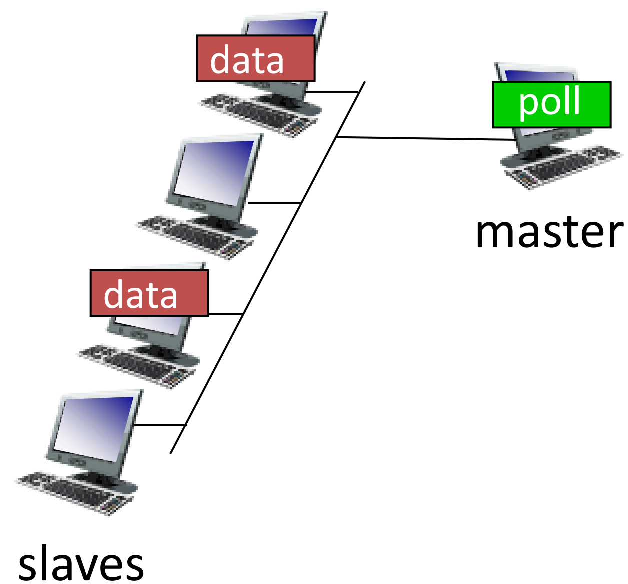
Taking turns protocols:

Look for best of both worlds!

Taking turns MAC protocols

Polling:

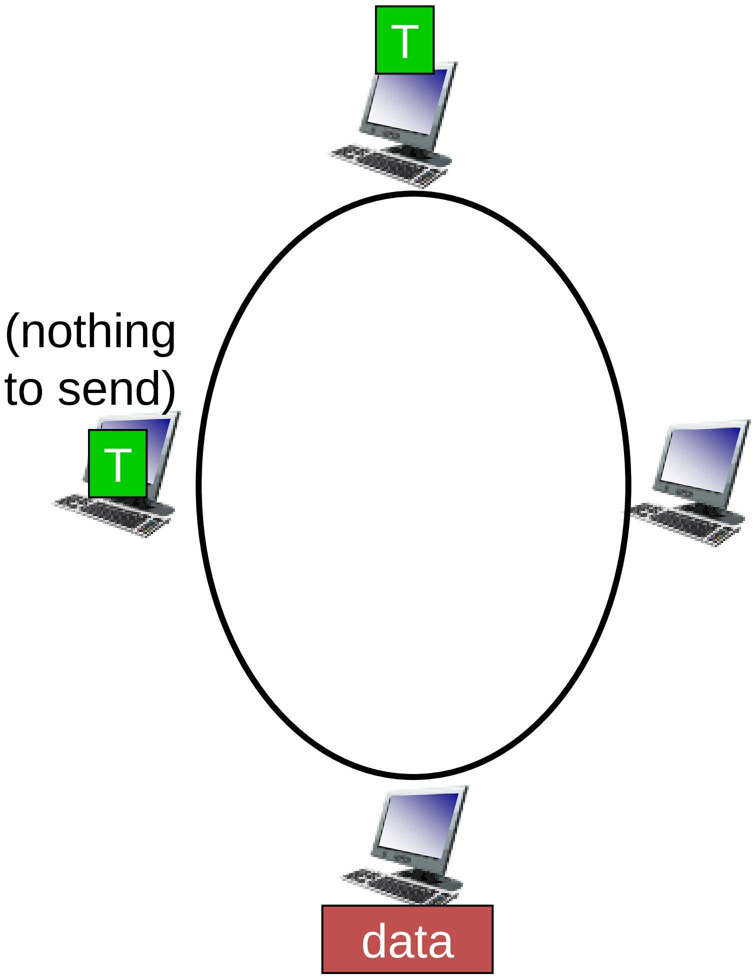
- Master node invites slave nodes to transmit in turn
- Typically used with dumb slave devices
- Concerns:
  - Polling overhead
  - Latency
  - Single point of failure (master)



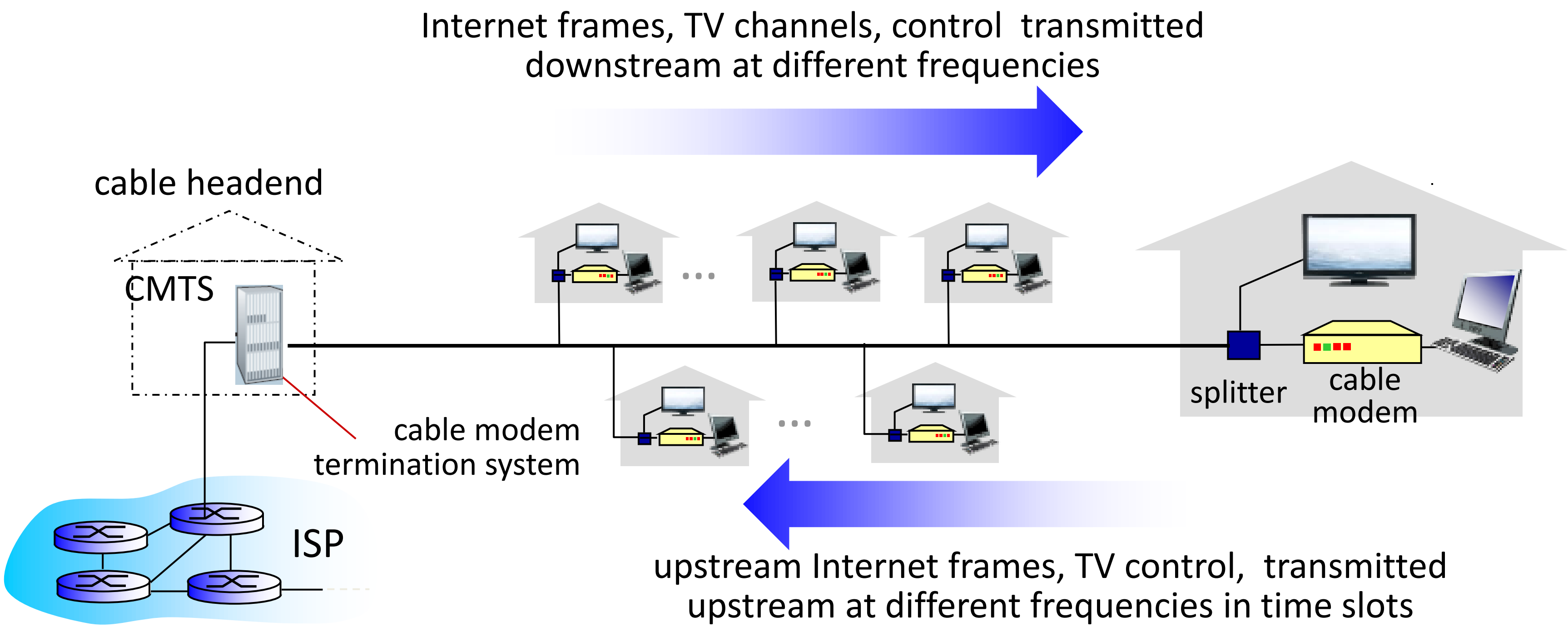
Taking turns MAC protocols

Token passing:

- Control token passed from one node to next sequentially.
- Token message
- Concerns:
  - Token overhead
  - Latency
  - Single point of failure (token)

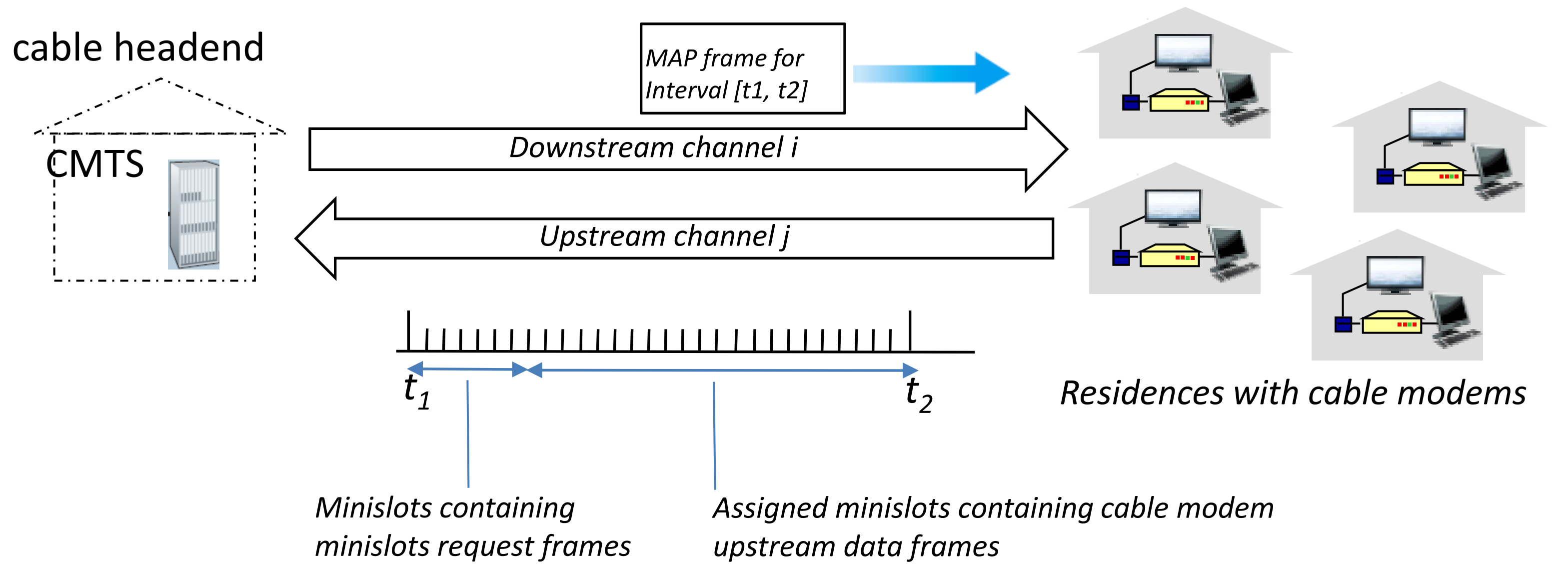


Cable access network



- Multiple 40Mbps downstream (broadcast) channels
  - Single CMTS transmits into channels
- Multiple 30 Mbps upstream channels
  - Multiple access: all users contend for certain upstream channel time slots (others assigned)

Cable access network



DOCSIS: data over cable service interface specification

- FDM over upstream, downstream frequency channels
- TDM upstream: some slots assigned, some have contention
  - Downstream MAP frame: assigns upstream slots
  - Request for upstream slots (and data) transmitted random access (binary backoff) in selected slots

Summary of MAC protocols

- *Channel partitioning*
  - By time, frequency or code
  - Time Division, Frequency Division, Code Division
- *Random access*
  - 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