Last Time

6.1 introduction, services

6.2 error detection, correction

6.3 Multiple Access Protocols

- channel partitioning
  - TDMA, FDMA, CSMA
- random access
  - Aloha: pure, slotted
  - CSMA, CSMA/CD

6.4 LANs

- addressing, ARP
- Ethernet
- switches
- VLANS

6.5 link virtualization: MPLS

6.6 data center networking

6.7 a day in the life of a web request

CSMA (carrier sense multiple access)

CSMA: listen ("sense") before transmitting:

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

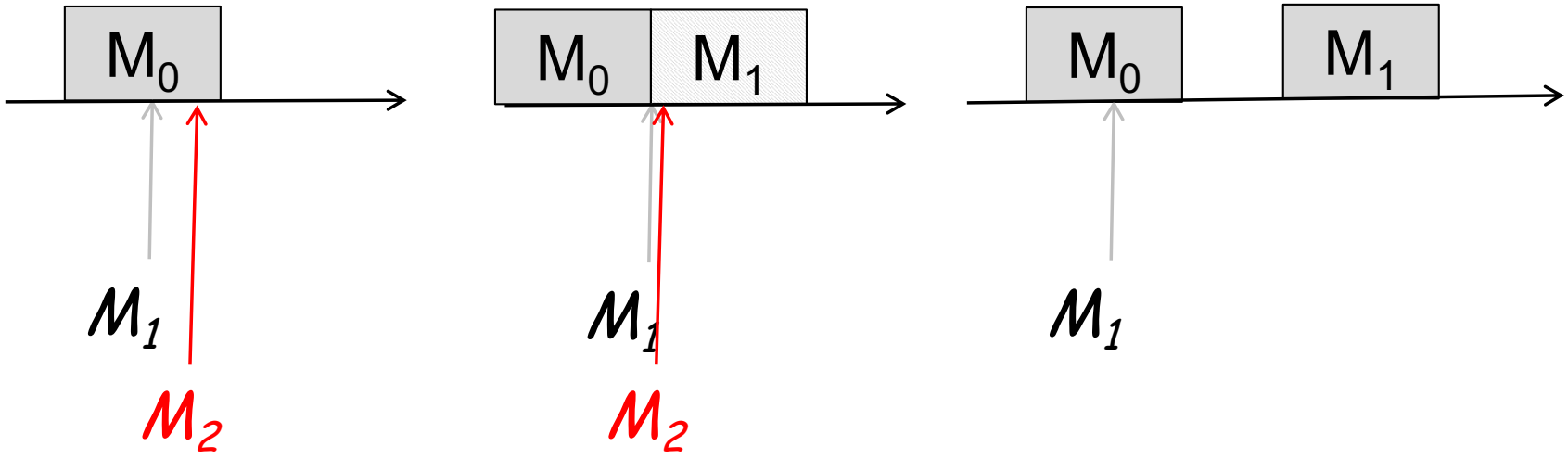
- human analogy: don't interrupt others!

CSMA (carrier sense multiple access)

CSMA: listen before transmit:

if channel sensed idle: transmit entire frame

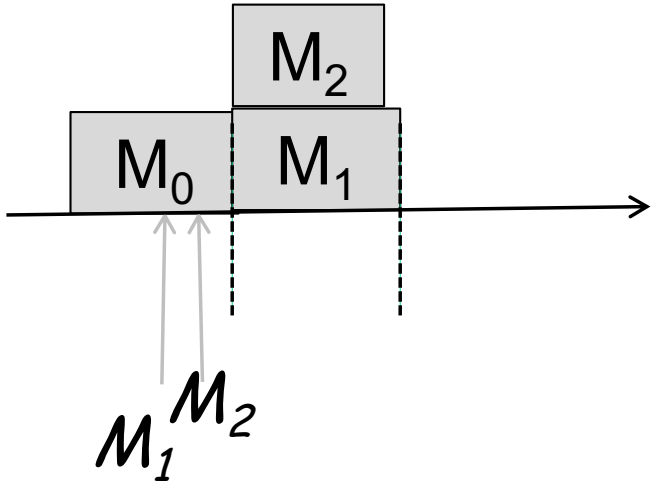
- if channel sensed busy, defer transmission
- human analogy: dont interrupt others!



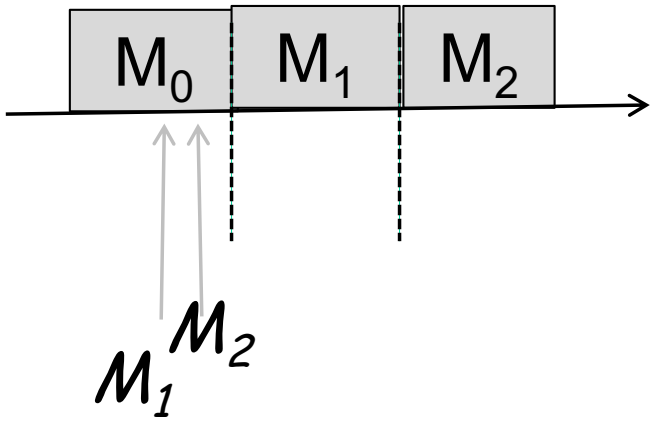
- One would hope that CS eliminate collisions. It doesn't!
  - Because of protocol (other users synchronized)
  - Because of physics (sensing is not instantaneous)

CSMA: Carrier Sensing and Persistence

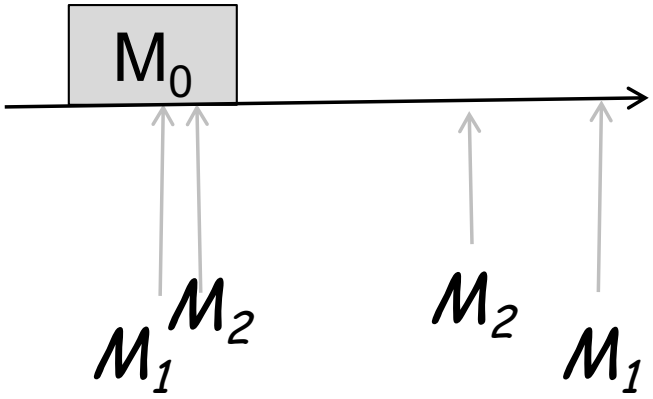
(1-) Persistent: defer transmission until channel becomes idle



P-Persistent: once the channel becomes idle, transmit with prob. p; or defer until next slot



(0-) Non-Persistent: defer transmission, sense again after a random time



Choice of p:
throughput vs delay

Higher Throughput

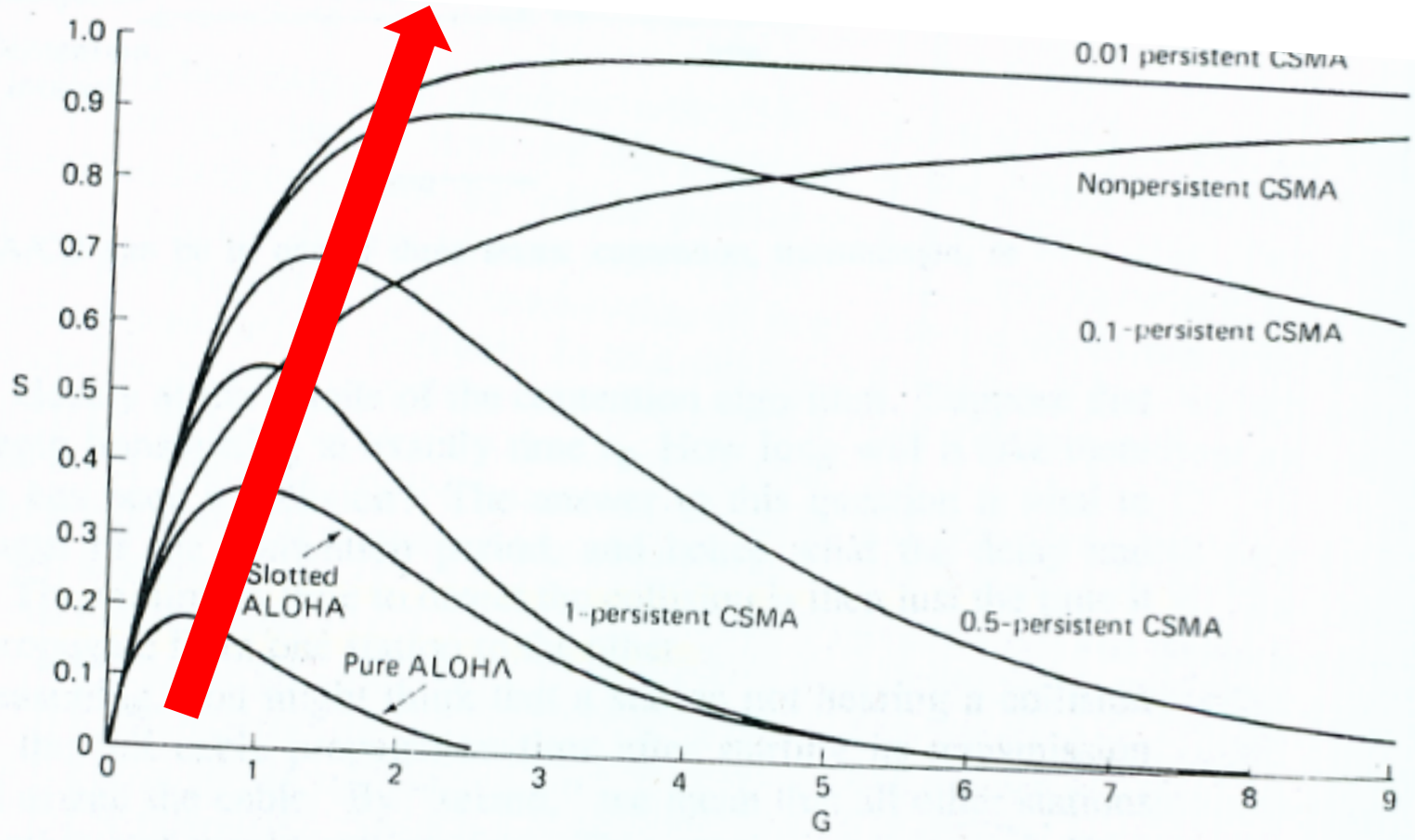
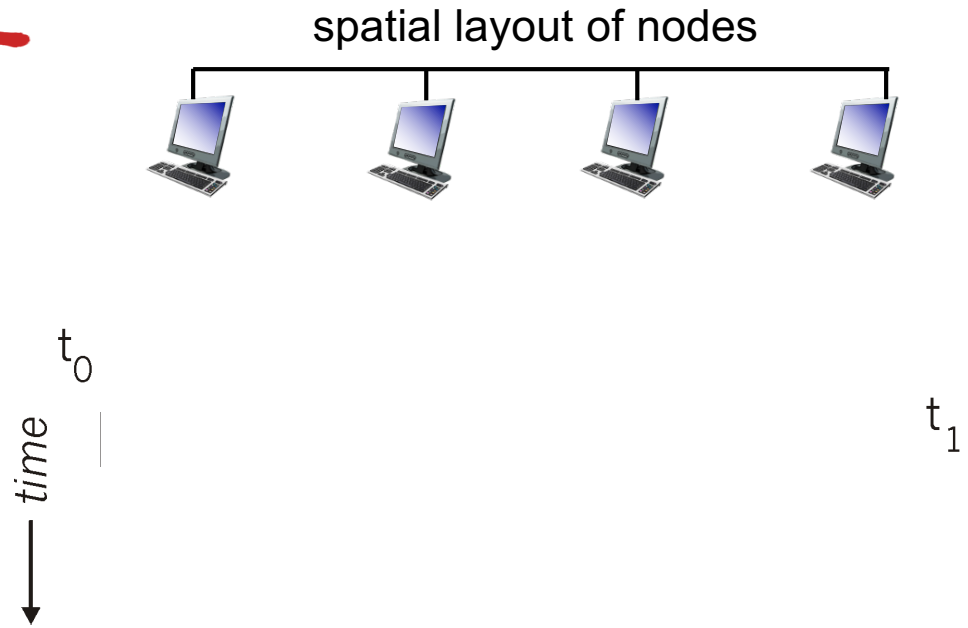


Fig. 3-5. Comparison of the channel utilization versus load for various random access protocols.

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



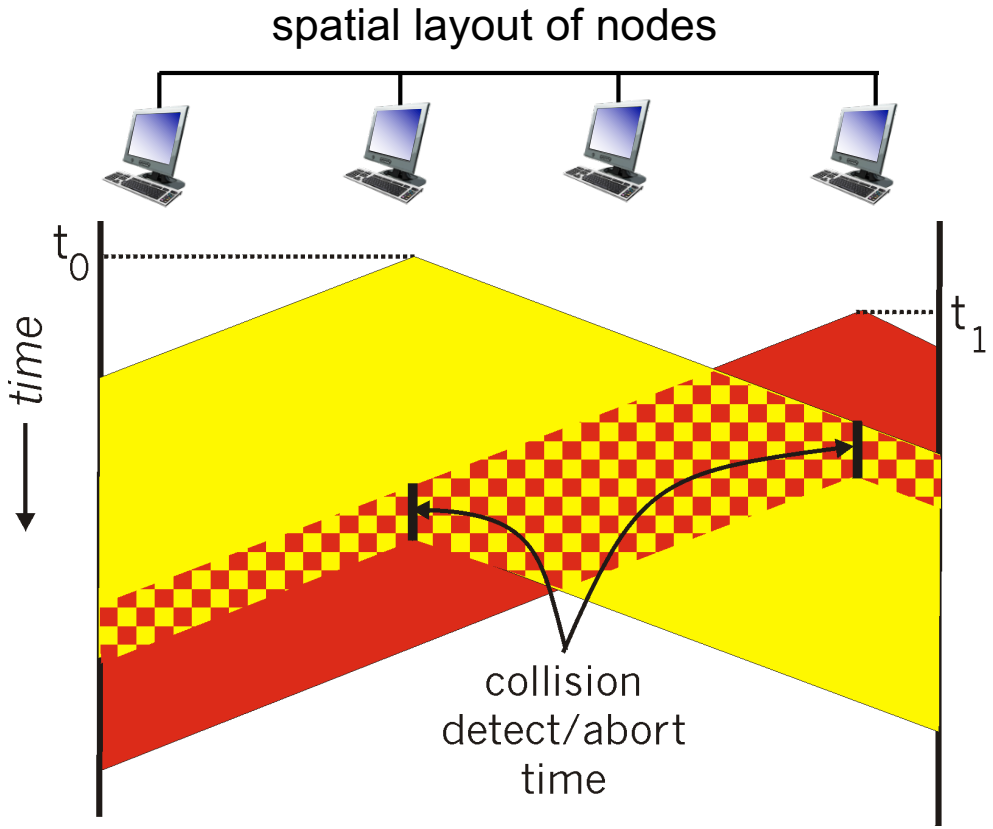
CSMA/CD (with collision detection)

- CSMA/CD: carrier sensing
  - If channel is busy, defer as in CSMA

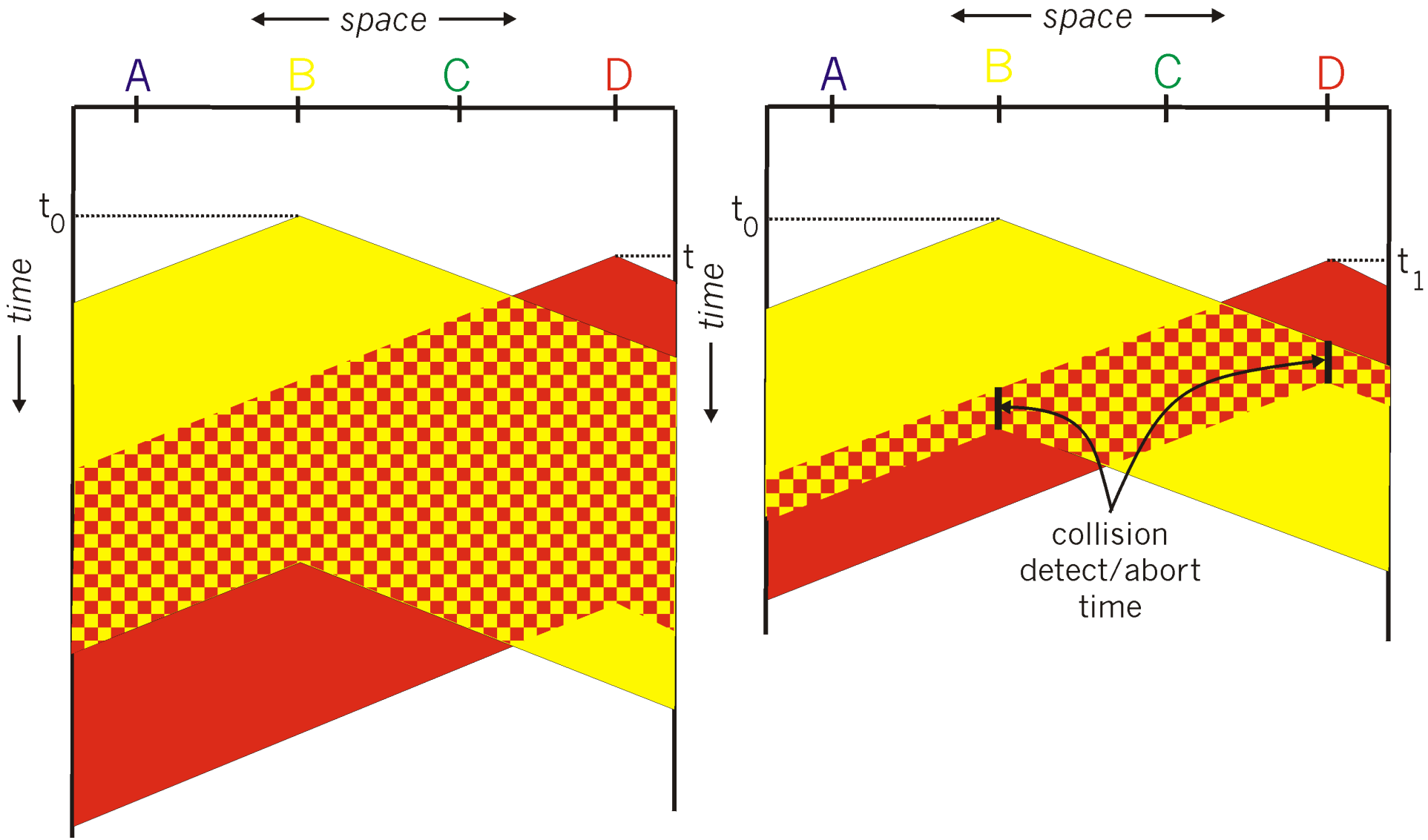
- CSMA/CD collision detection:
  - collisions *detected* within short time
  - as soon as you detect collision → do not complete, abort→reduce channel waste
  - how easy is collision detection?
    - easy in wired LANs: measure signal strengths, compare transmitted, received signals
    - difficult in wireless LANs: received signal strength overwhelmed by local transmission strength

- human analogy: the polite conversationalist

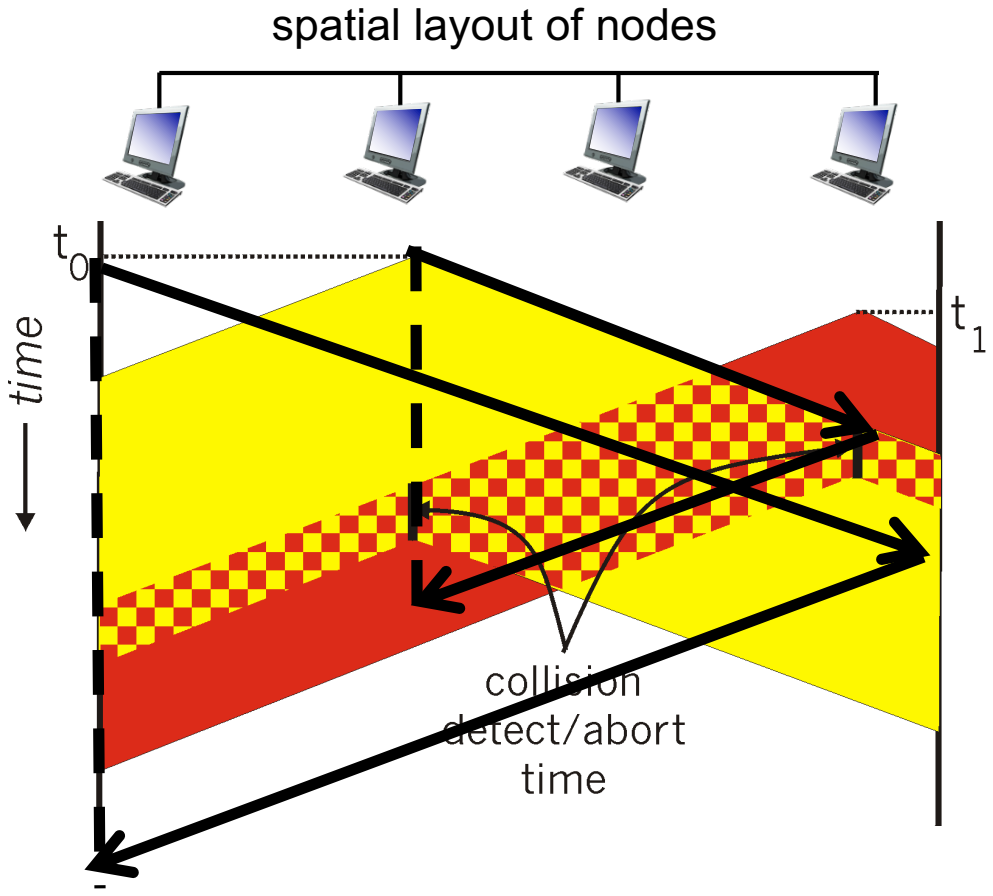
CSMA/CD (collision detection)



CSMA collisions without/with CD



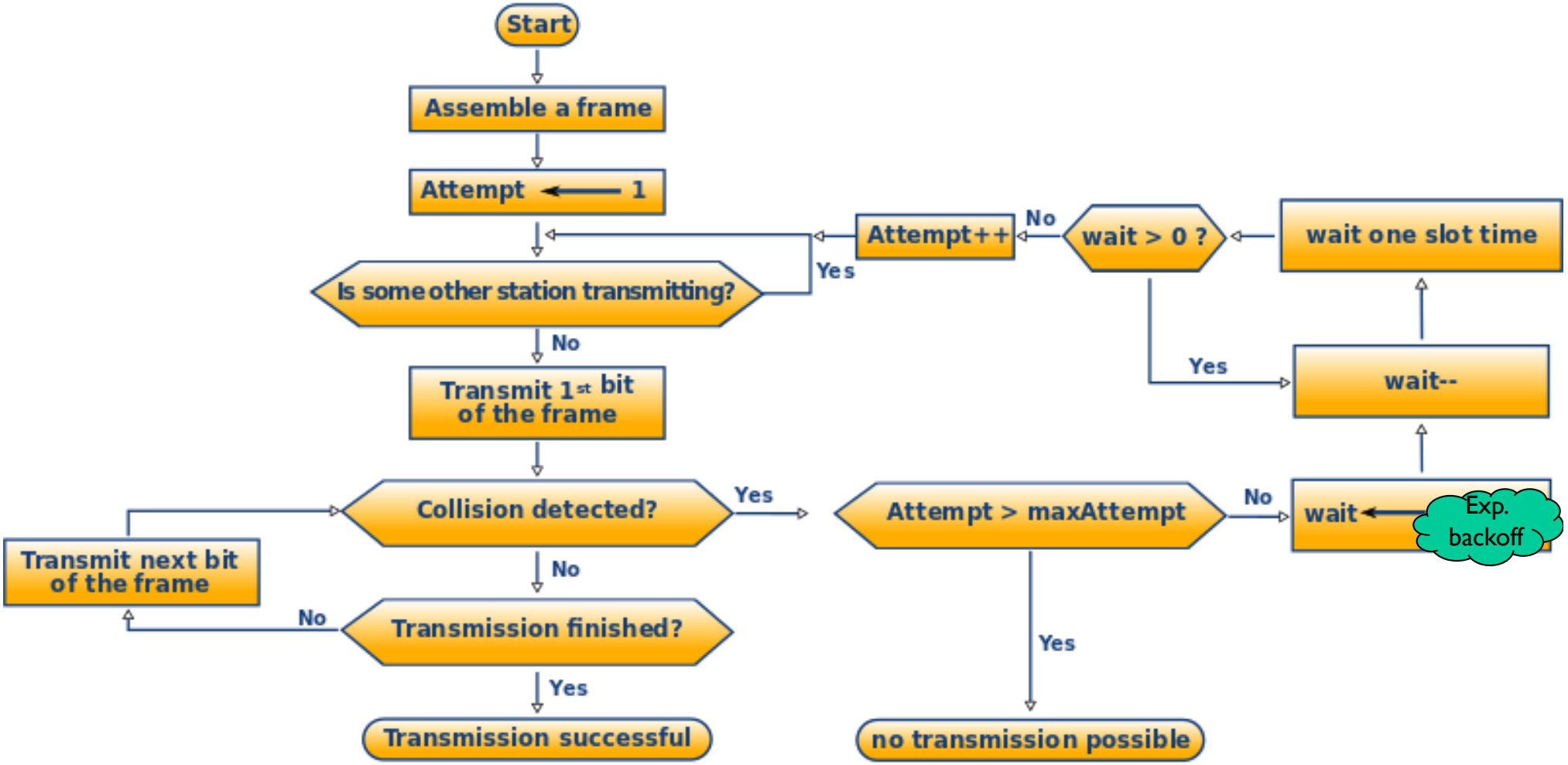
You are sure you "seized" the channel after $2d_{prop}$



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 $\{0, 1, 2, \ldots, 2^m-1\}$. NIC waits $K \cdot 512$ bit times, returns to Step 2
   - longer backoff interval with more collisions

CSMA/CD Algorithm



http://en.wikipedia.org/wiki/Carrier_sense_multiple_access_with_collision_detection

CSMA/CD principles

- CSMA: 1-/p-/non- persistent
- CD
- Retransmission after exponential backoff:
  - Why random: to avoid synchronization
  - Why exponential: ~ TCP's multiplicative decrease
  - Why adjust after every collision: p essentially adapts to N
- Minimum frame=contention slot=512bits
  - =worst case RTT (for 10Mbps, length 2500m)

[Analysis of CSMA/CD]

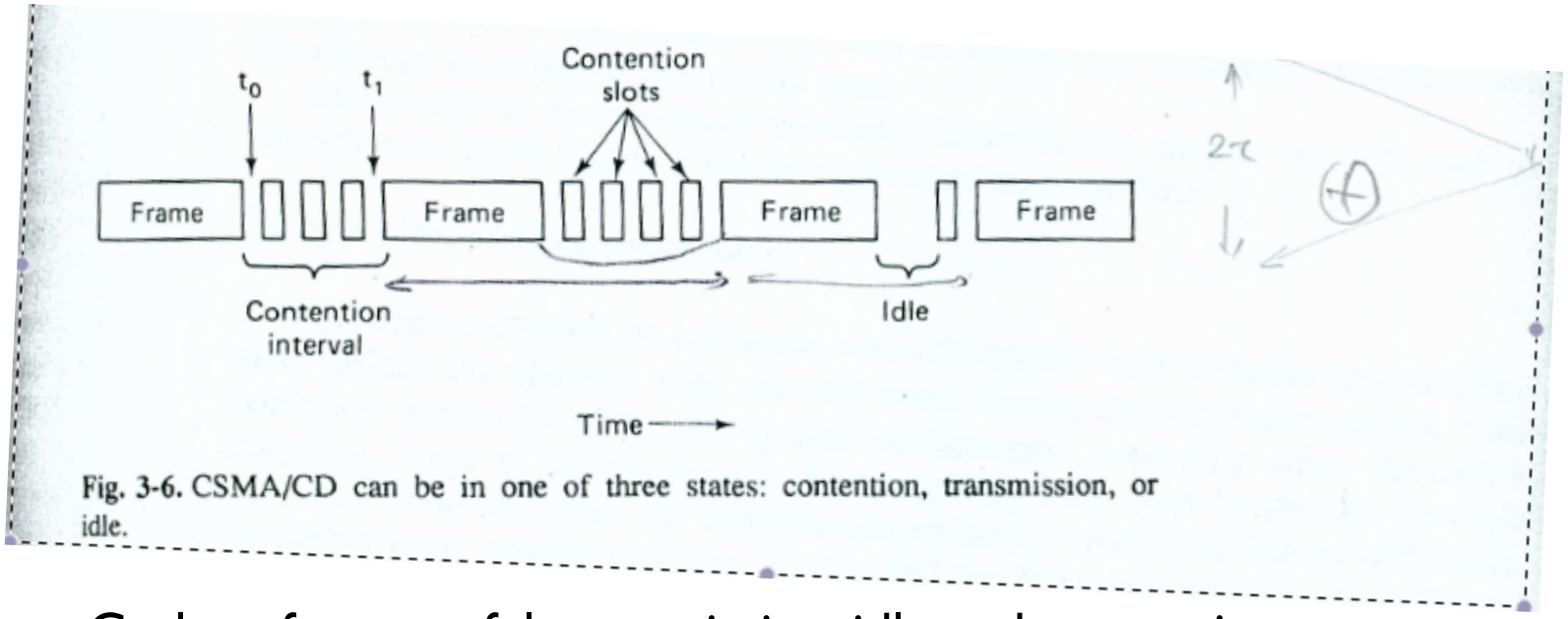


Fig. 3-6. CSMA/CD can be in one of three states: contention, transmission, or idle.

- Cycles of: successful transmission, idle and contention
- Contention slot: S>=2*(prop.delay)
- Channel efficiency= $\dfrac{\textit{Successful Tx time (in a cycle)}}{\textit{Duration of a cycle}}$
- Because analysis of exp. backoff is difficult
  - Ch.6, Problem 20's Simplification: k stations, each transmitting with prob. p in each slot [this looks like slotted Aloha]

CSMA/CD efficiency

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

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

- efficiency goes to 1
  - as t_{prop} goes to 0
  - as t_{trans} goes to infinity
- better performance than ALOHA
- and simple, cheap, decentralized!

WiFi(802.11) vs. Ethernet (802.3)

- Both CSMA-based
- Collision detection: possible in Ethernet not on WiFi
- Collision avoidance in WiFi:
  - "transmit after you *seize* the channel"
  - Seize the channel through sensing in Ethernet
  - Seize the channel through RTS/CTS in WiFi
  - http://en.wikipedia.org/wiki/Carrier_sense_multiple_access_with_collision_avoidance

Random Access Protocols so far

- when node has packet to send
  - transmit at full channel data rate R.
  - no coordination among nodes (distributed)
- two or more transmitting nodes ➜ "collision"
- random access MAC protocol specifies:
  - how each node detects collisions
  - how each node recovers from collisions (e.g., via delayed retransmissions)
- examples of random access MAC protocols:
  - ALOHA family: slotted ALOHA, "pure" ALOHA
  - CSMA family: CSMA, CSMA/CD, CSMA/CA

"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

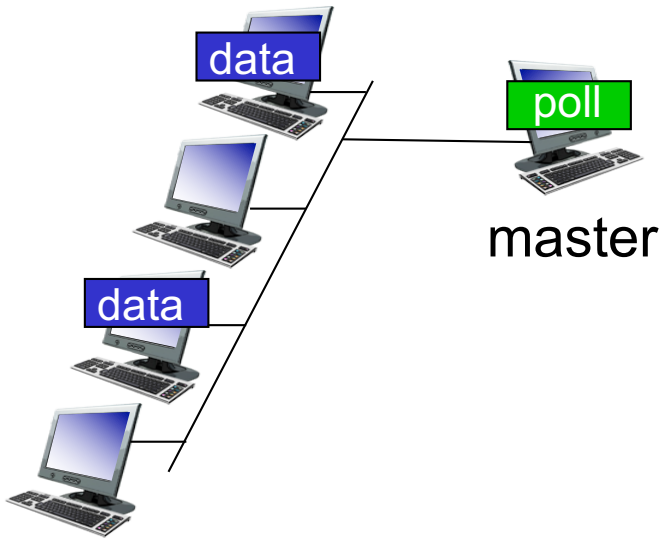
"taking turns" protocols

look for best of both worlds!

["Taking turns" MAC protocols]

polling:

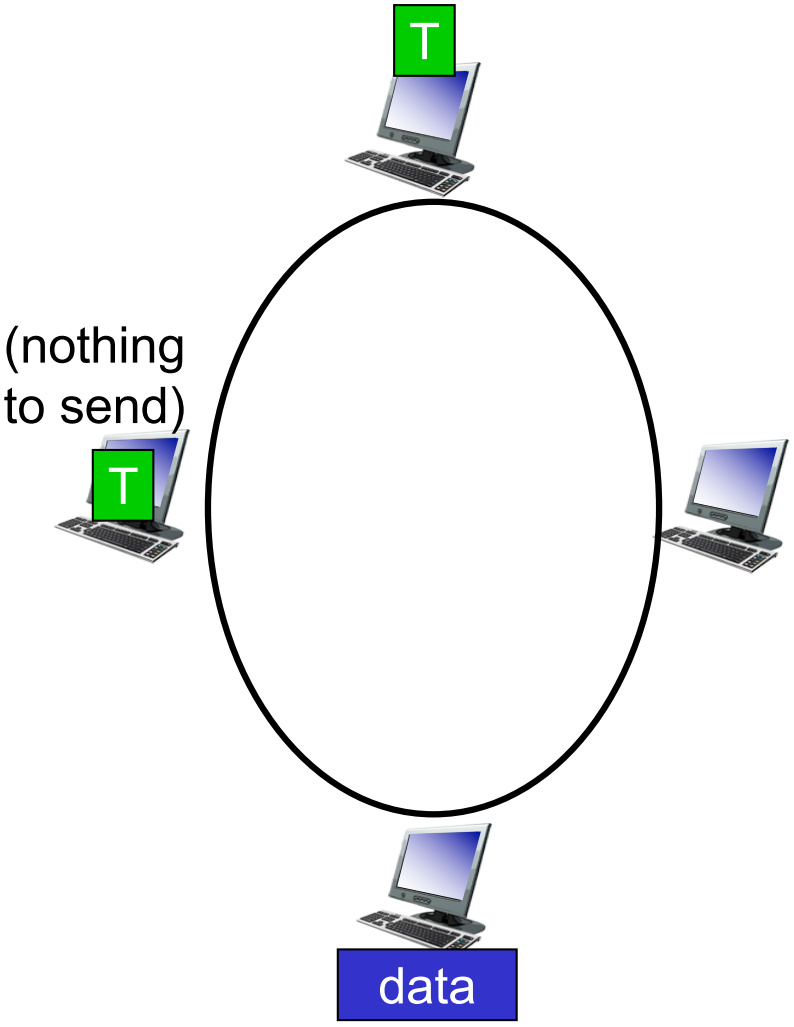
- master node "invites" slave nodes to transmit in turn
- typically used with "dumb" slave devices
- concerns:
  - polling overhead
  - latency
  - single point of failure (master)



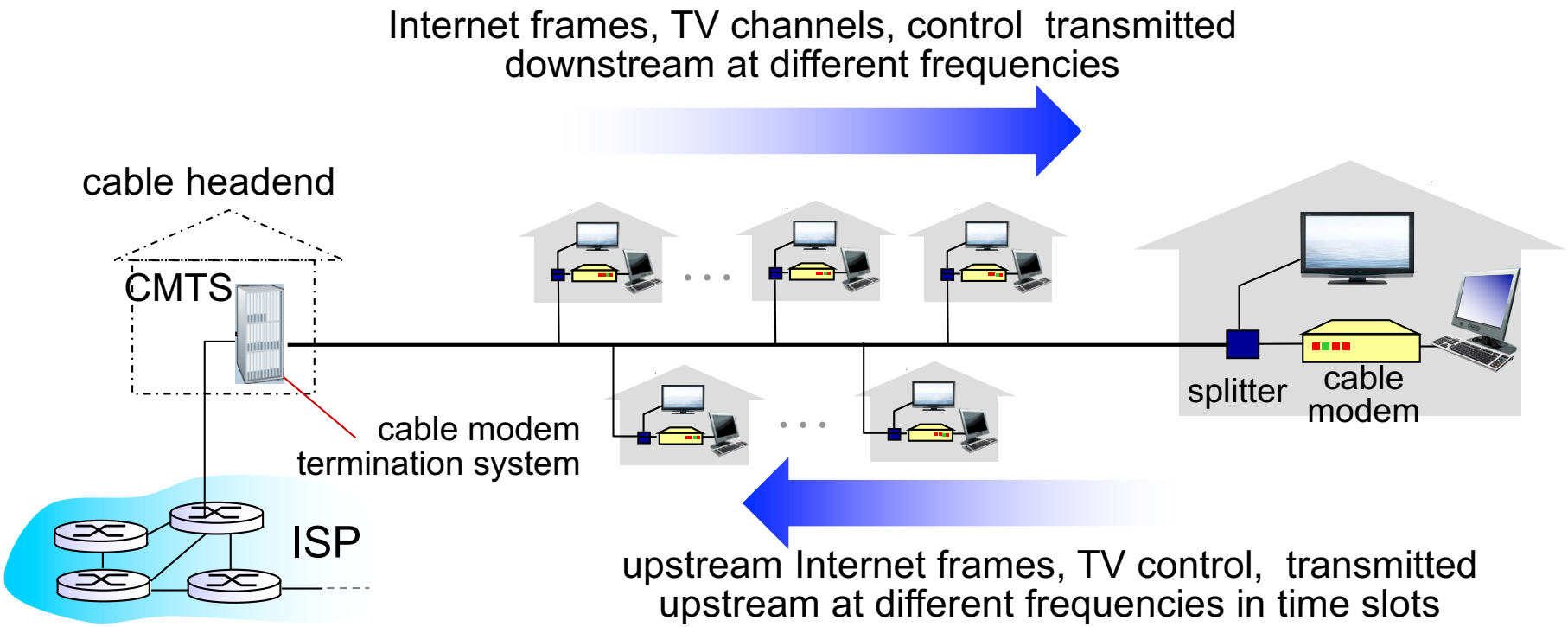
["Taking turns" MAC protocols]

token passing:

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

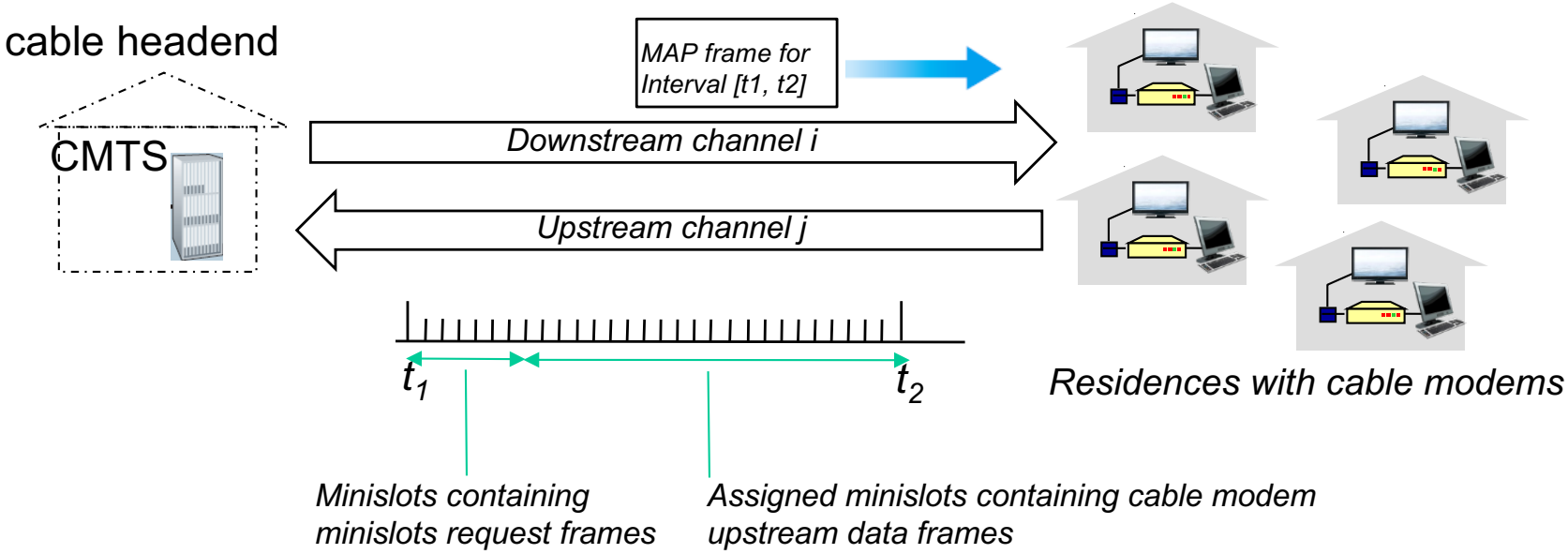


[Cable access network]



- 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 spec

- 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
- *random access* (dynamic),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - carrier sensing: easy in some technologies (wire), hard in others (wireless)
  - CSMA/CD used in Ethernet
  - CSMA/CA used in 802.11
- *taking turns*
  - Passing a token: FDDI, token ring
  - polling from central site: e.g. bluetooth
- Practical protocols mix and match these ideas
  - [E.g. protocols for cable internet access]