Chapter 6 The Link Layer and LANs

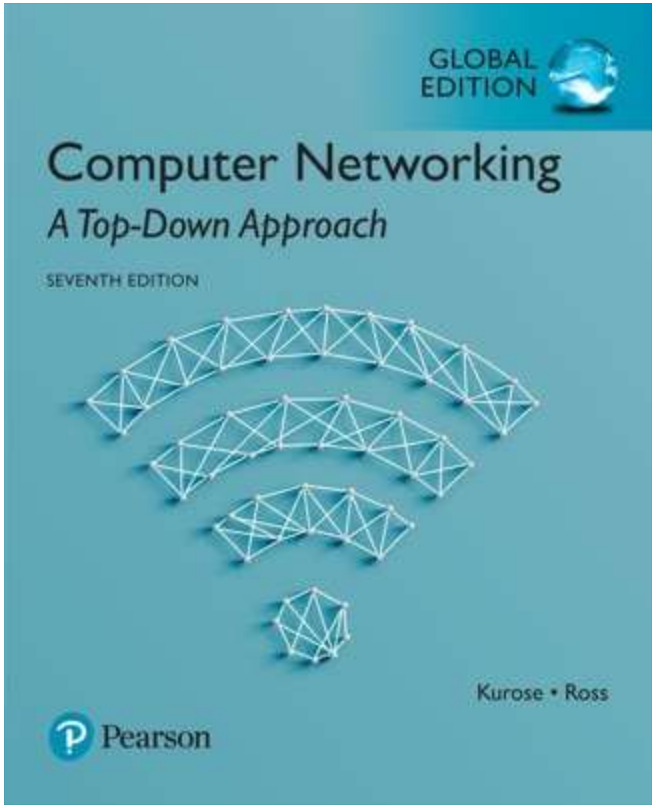
Modified by RenPing.Liu@uts.edu.au
19 May 2019

A note on the use of these Powerpoint slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR



Computer Networking: A Top Down Approach

7th edition
Jim Kurose, Keith Ross
Pearson/Addison Wesley
April 2016

Link layer, LANs: outline

6.1 introduction, services

6.2 error detection, correction

6.3 multiple access protocols

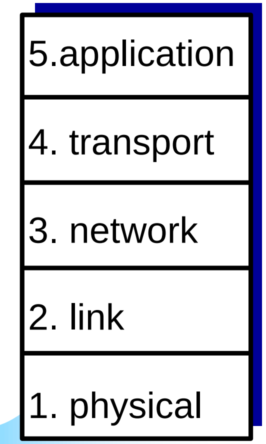
6.4 LANs
- addressing, ARP
- Ethernet
- switches
- VLANS

~~6.5 link virtualization: MPLS~~

~~6.6 data center networking~~

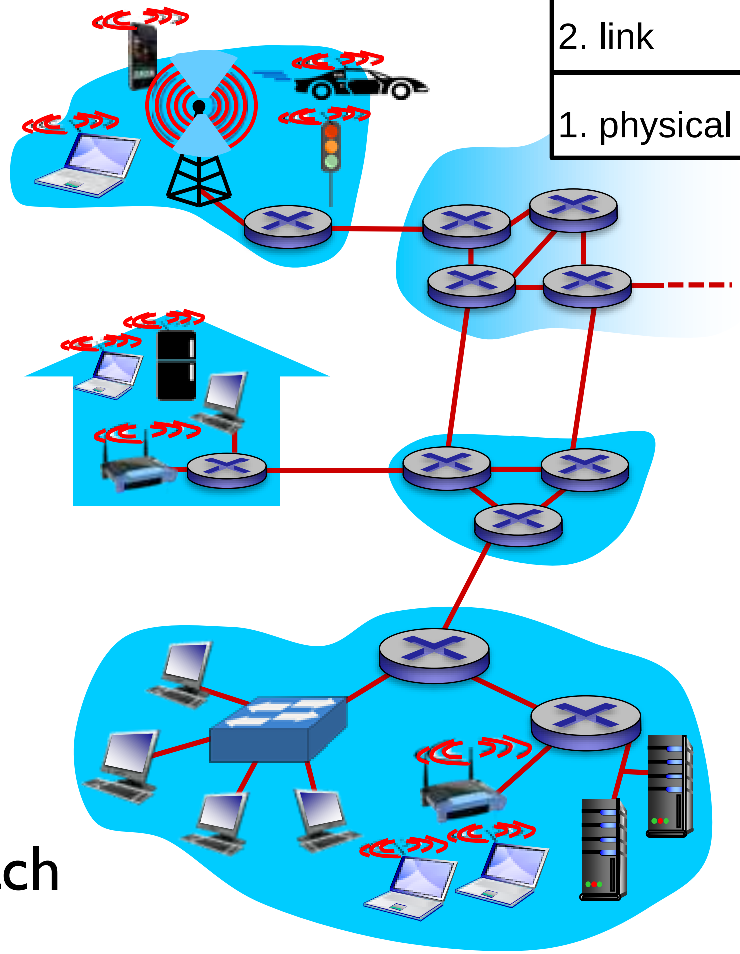
6.7 a day in the life of a web request

Link layer: introduction



terminology:

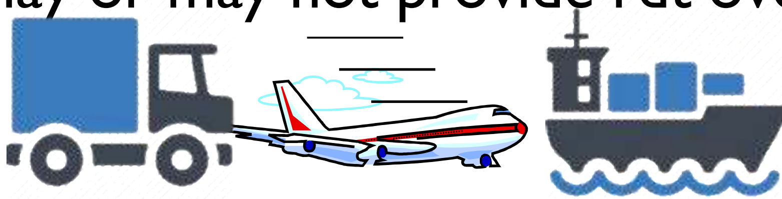
- hosts and routers: nodes
- communication channels that connect adjacent nodes along communication path: links
- Media:
  - wired links: fibre, Ethernet, coaxial, phone line
  - wireless links: WiFi, Cellular
- Topology
  - point-to-point
  - LANs: WiFi, Cell, Ethernet switch



Link layer: context

- layer-2 packet: frame, encapsulates datagram
- datagram transferred by different link protocols over different links:
  - e.g., WiFi on first link, coaxial and fibre on intermediate links, Ethernet on last link
- each link protocol provides different services
  - e.g., may or may not provide rdt over link

5.application
4. transport
3. network
2. link
1. physical



letter writing		letter reading	home
postman pick up		postman delivery	postbox
distribute		distribute	post office
truck delivery		truck delivery	road
	air transport	ship transport	sea/air

Link layer services

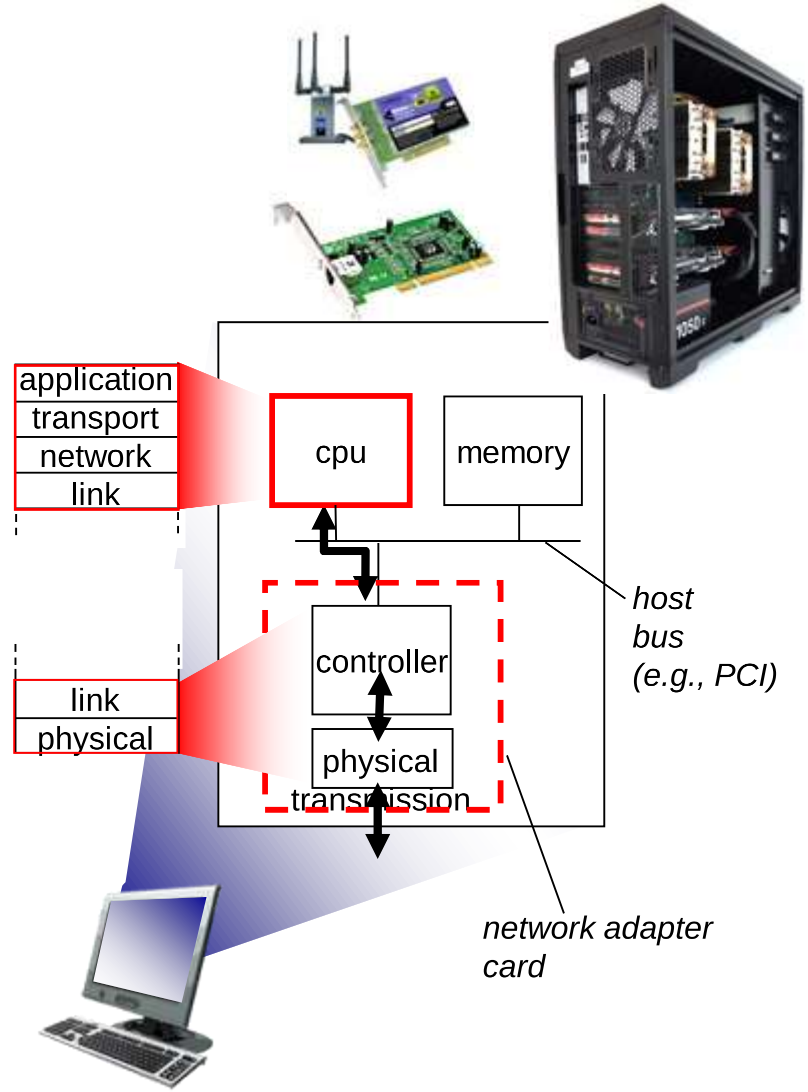
- *framing, link access:*
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - "MAC" addresses used in frame headers to identify source, destination
    - different from IP address!
- *reliable delivery between adjacent nodes*
  - we learned how to do this already (chapter 3)!
  - seldom used on low bit-error link (fiber, some twisted pair)
  - wireless links: high error rates
    - *Q:* why both link-level and end-end reliability?

Link layer services (more)

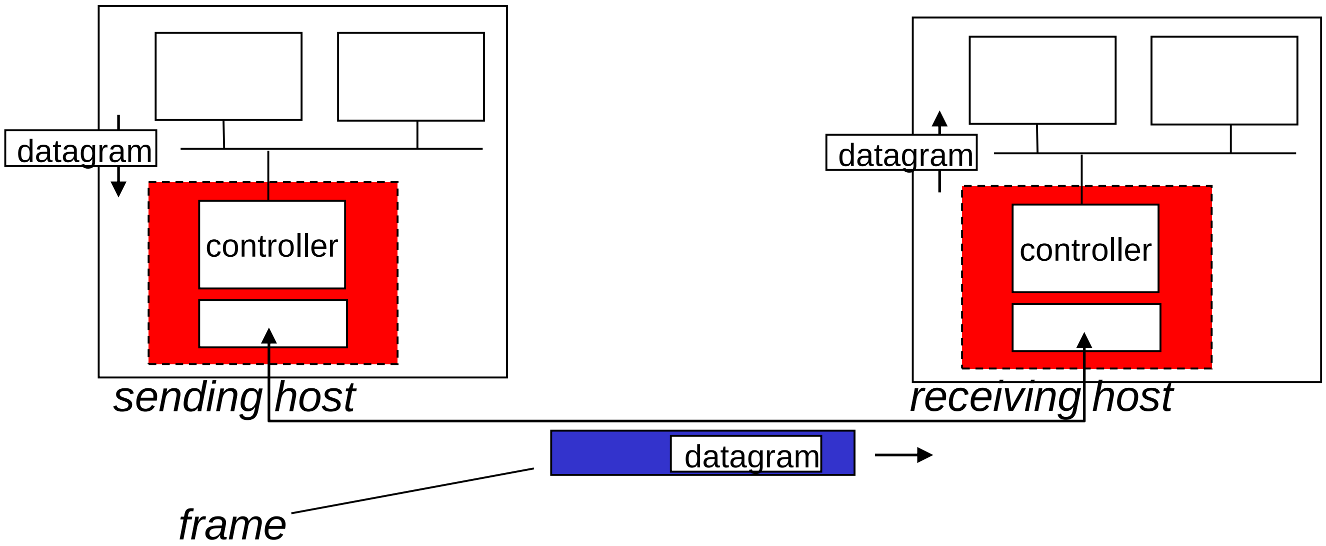
- *flow control:*
  - pacing between adjacent sending and receiving nodes
- *error detection:*
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- *error correction:*
  - receiver identifies *and corrects* bit error(s) without resorting to retransmission
- *half-duplex and full-duplex*
  - with half duplex, nodes at both ends of link can transmit, but not at same time

Where is the link layer implemented?

- in each and every host
- link layer implemented in "adaptor" (aka *network interface card* NIC) or on a chip
  - Ethernet card, 802.11 card; Ethernet chipset
  - implements link, physical layer
- attaches into host's system buses
- combination of hardware, software, firmware



Adaptors communicating



- sending side:
  - encapsulates datagram in frame
  - adds error checking bits, rdt, flow control, etc.
- receiving side
  - looks for errors, rdt, flow control, etc.
  - extracts datagram, passes to upper layer at receiving side

Link layer, LANs: outline

6.1 introduction, services

6.2 error detection, correction

6.3 multiple access protocols

6.4 LANs
- addressing, ARP
- Ethernet
- switches
- VLANS

6.5 link virtualization: MPLS

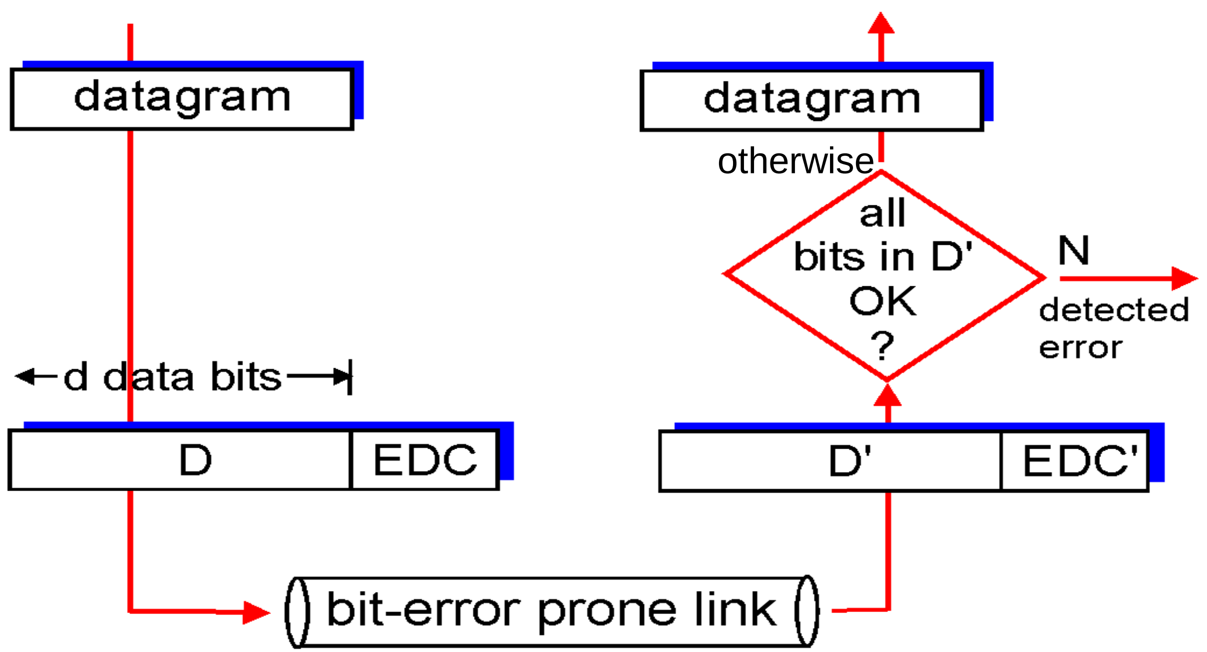
6.6 data center networking

6.7 a day in the life of a web request

Error detection

EDC= Error Detection and Correction bits (redundancy)
D = Data protected by error checking, may include header fields

- Error detection not 100% reliable!
  - protocol may miss some errors, but rarely
  - larger EDC field yields better detection and correction



Internet checksum (review)

goal: detect “errors” (e.g., flipped bits) in transmitted packet (note: used at transport layer only)

sender:

- treat segment contents as sequence of 16-bit integers
- checksum: addition (1’s complement sum) of segment contents
- sender puts checksum value into UDP checksum field

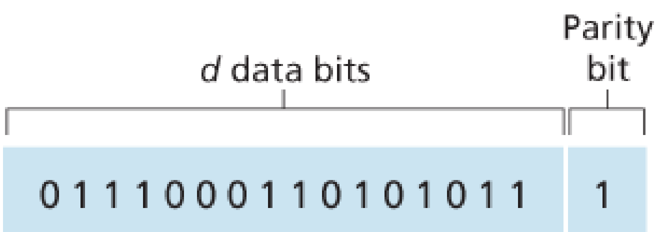
receiver:

- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - NO - error detected
  - YES - no error detected. *But maybe errors nonetheless?*

Parity checking

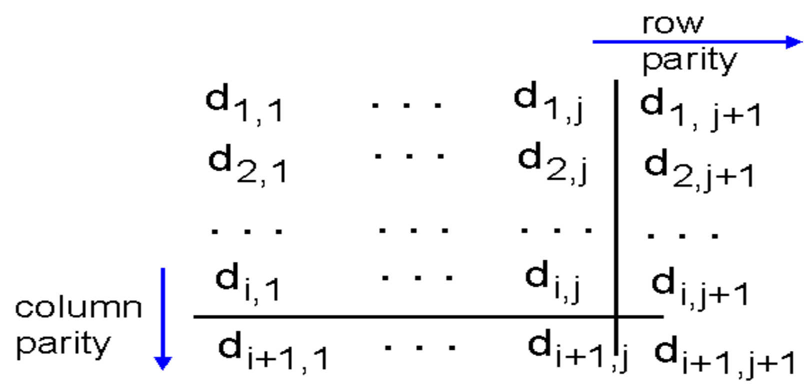
single bit parity:

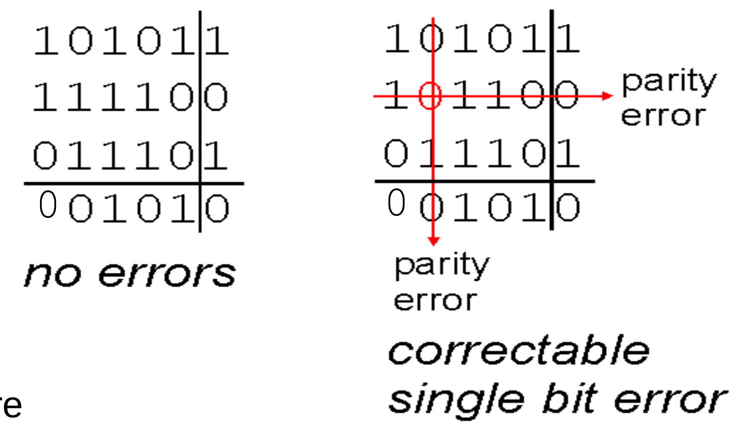
- *d*etect single bit errors
- one-bit even parity



two-dimensional bit parity:

- detect and correct single bit errors



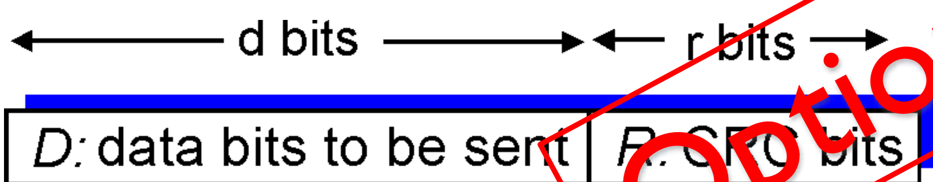


* Check out the online interactive exercises for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

Cyclic redundancy check

- more powerful error-detection coding
- view data bits, D, as a binary number
- choose r+1 bit pattern (generator), G
- goal: choose r CRC bits, R, such that
  - <D,R> exactly divisible by G (modulo 2)
  - receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
  - can detect all burst errors less than r+1 bits
- widely used in practice (Ethernet, 802.11 WiFi, ATM)

Optional – not tested



bit pattern

$$D * 2^r \text{ XOR } R$$

mathematical formula

CRC example

want:

$D \cdot 2^r$ XOR $R = nG$

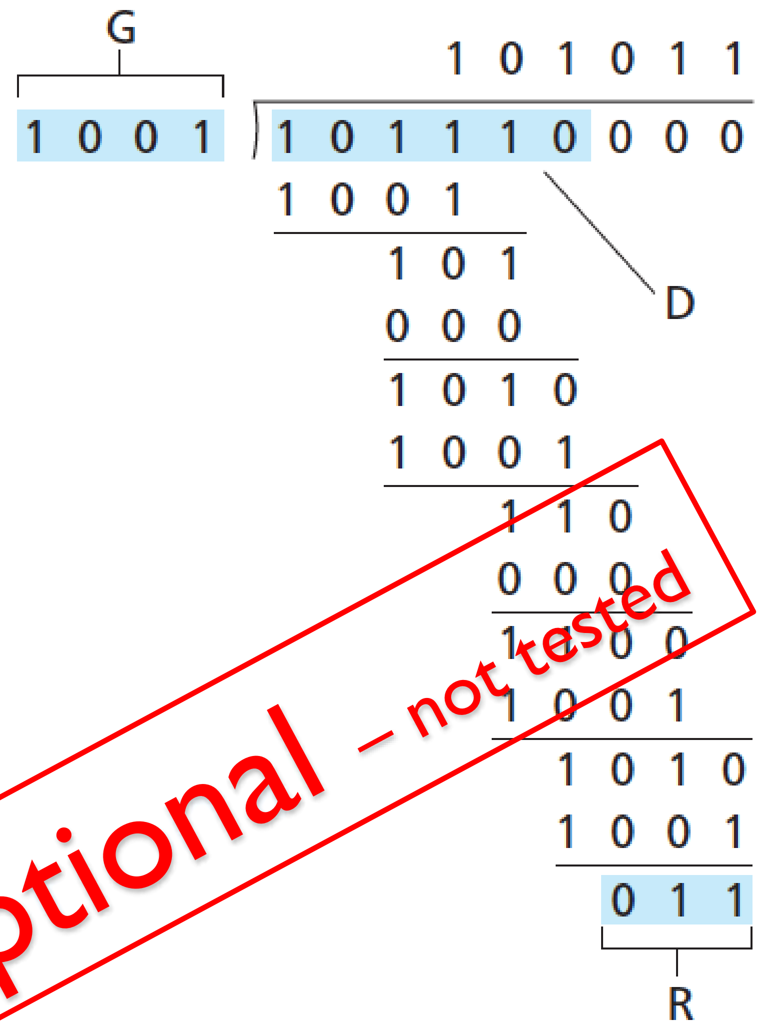
equivalently:

$D \cdot 2^r = nG$ XOR R

equivalently:

if we divide $D \cdot 2^r$ by G, want remainder R to satisfy:

$$R = remainder\left[\frac{D \cdot 2^r}{G}\right]$$



Optional – not tested

* Check out the online interactive examples for more examples: http://gaia.cs.umass.edu/kurose_ross/interactive/

Link layer, LANs: outline

6.1 introduction, services

6.2 error detection, correction

6.3 multiple access protocols

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

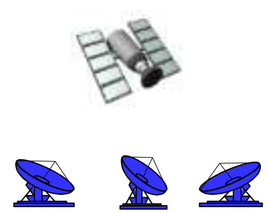
Multiple access links, protocols

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
  - 802.11 wireless LAN

shared wire (e.g., old 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
- 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

MAC protocols: taxonomy

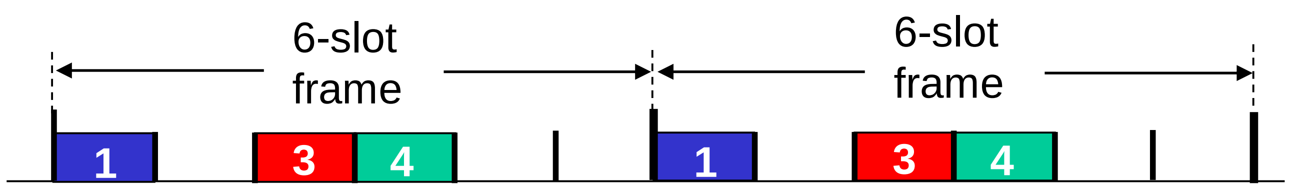
three broad classes:

- *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 MAC protocols: TDMA

TDMA: time division multiple access

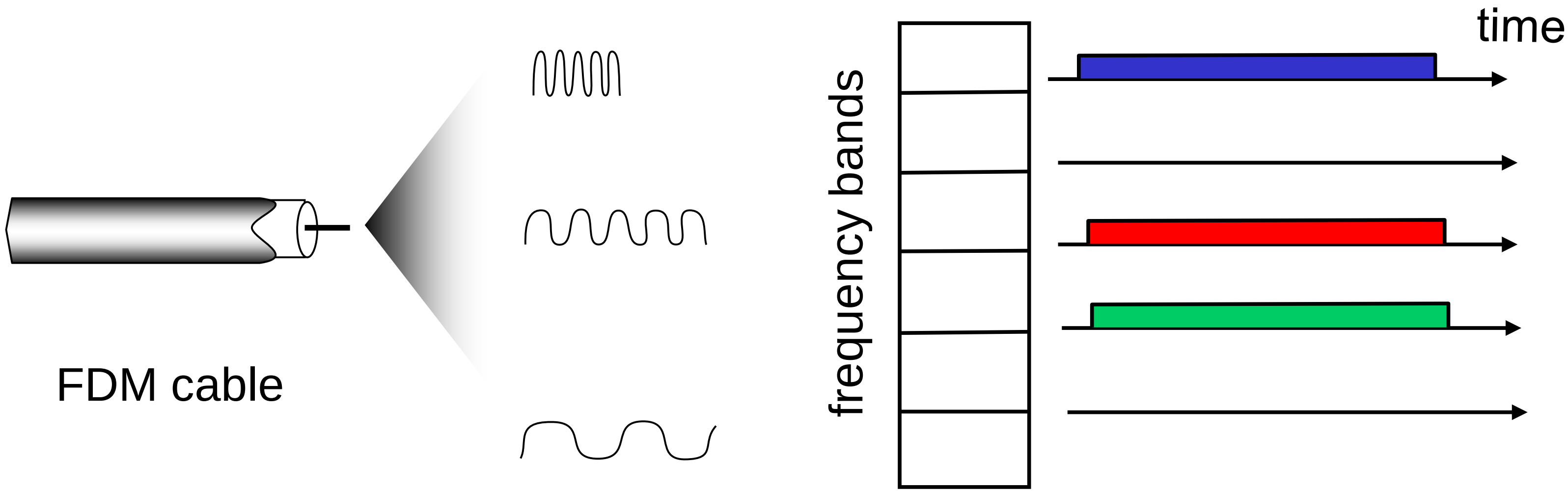
- access to channel in "rounds"
- each station gets fixed length slot (length = packet transmission time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have packets to send, slots 2,5,6 idle



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



An ideal multiple access protocol

Cellular Net: TDMA/FDMA

TDMA or FDMA satisfactory? No

- Only use a fraction of the link, even no other traffic
- Heavy coordination processing/traffic

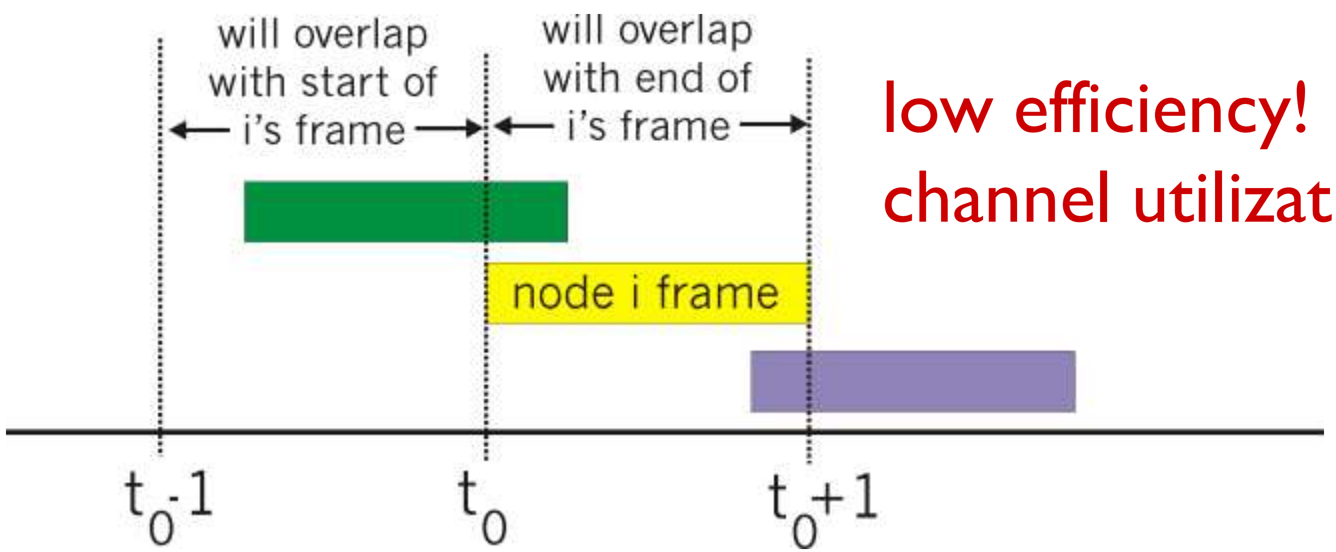
Ideal: given broadcast channel of rate R bps

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
4. simple

Pure (unslotted) ALOHA

- unslotted Aloha: simple, no coordination
- when frame first arrives
  - transmit immediately
- collision problem:
  - frame sent at t_0 collides with other frames sent in $[t_0-1, t_0+1]$



low efficiency!

channel utilization = 18%

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 $[t_0-1,t_0]$

define p as the probability that a device transmits in one slot

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

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

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

even *worse* than slotted Aloha!

Optional – not tested

Slotted ALOHA

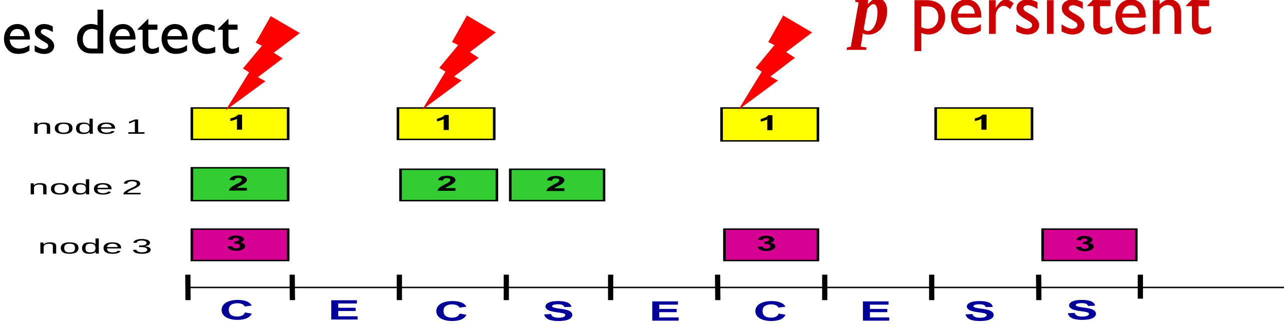
assumptions:

- 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

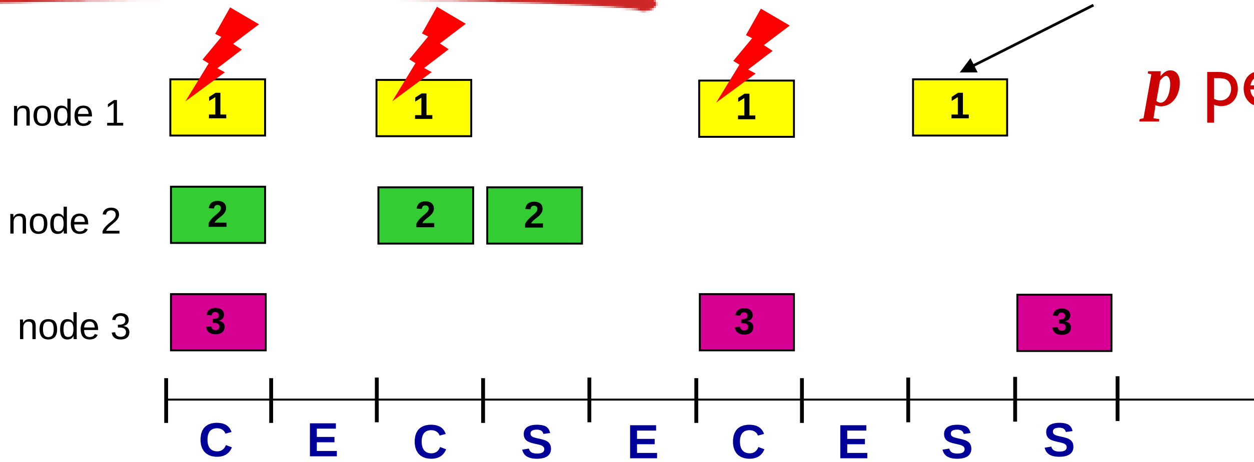
p persistent



Slotted ALOHA

Total number of nodes: N

Each node transmit in a slot with prob: p



***p* persistent**

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 $= Np(1-p)^{N-1}$

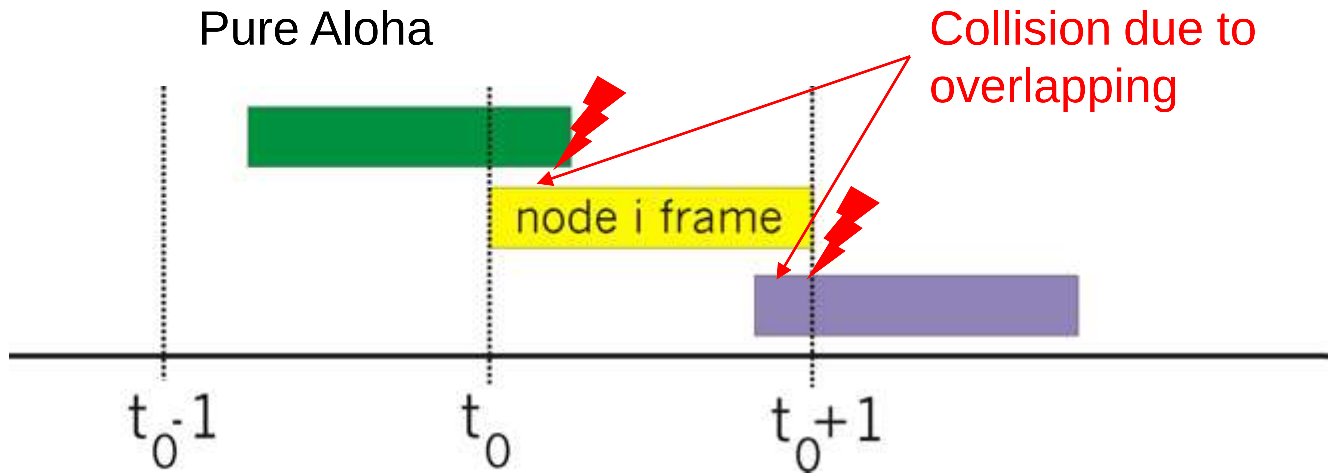
- max efficiency: find p^* that maximizes $Np(1-p)^{N-1}$
- for many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as N goes to infinity, gives:

  max efficiency $= 1/e = .37$

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

!

Can we avoid collision?



CSMA (carrier sense multiple access)

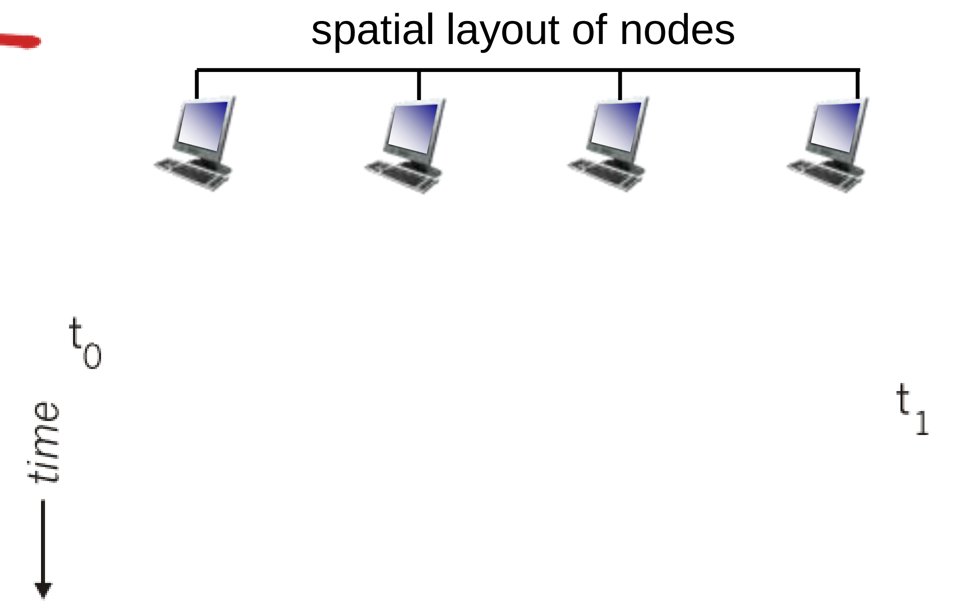
listen before talk:

if channel sensed idle: transmit entire frame

- if channel sensed busy, defer transmission
- human analogy: don't interrupt others!
  - the polite conversationalist

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

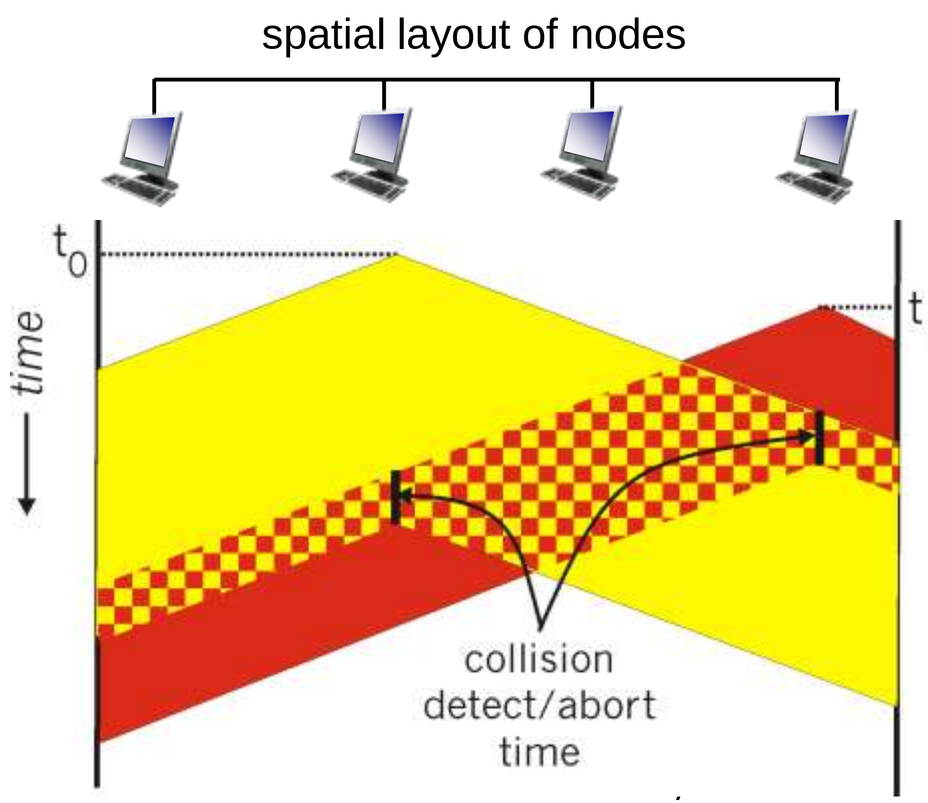


CSMA/CD (collision detection)

CSMA/CD: carrier sensing, deferral as in CSMA

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

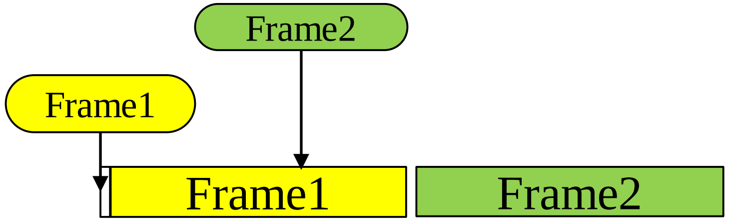
- collision detection:

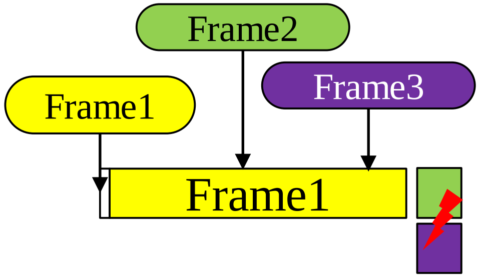


- easy in wired LANs: measure signal strengths, compare transmitted, received signals
- difficult in wireless LANs: received signal strength overwhelmed by local transmission strength
  In WiFi, it's CSMA/CA (Collision Avoidance)

Ethernet CSMA/CD algorithm

1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel idle, starts frame transmission.
   - If successful, NIC is done with frame!
3. If NIC senses channel busy,
   - defer until channel idle, then transmit
4. If detects another transmission while transmitting - collision
   - Collision: abort transmission, go to Backoff Mode





Ethernet CSMA/CD algorithm -continue

5. After collision → ***binary (exponential) backoff:***

after m^{th} collision, *- chooses K at random from {0,1,2, …, 2^m-1}.*

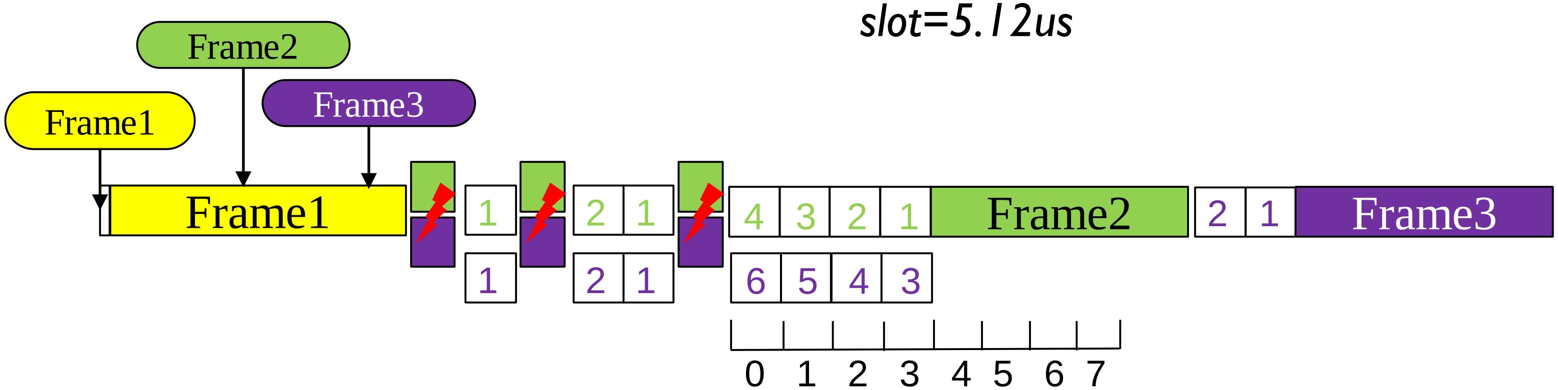
*- waits for K slots, slot=512*bit time*

1) after 1^{st} collision: m=1, k is chosen in {0,1} → k=1, 1
2) after 2^{nd} collision: m=2, k is chosen in {0,1,2,3} → k=2, 2
3) after 3^{rd} collision: m=3, k is chosen in {0,1,2,…,7} → k=4, 6

**slot=512*bit time, bit time=1bit/Rate*

Ex: 100Mbps - bit time=1/10M=0.01us (microsecond)

slot=5.12us



CSMA/CD efficiency

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

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

Good for LAN

- efficiency depends on
  - as t_{prop} increases, efficiency goes down (t_{prop}=0→eff=100%)
  - as t_{trans} increases, efficiency goes up
- better performance than ALOHA: and simple, cheap, decentralized!
- CSMA/CD was standardized in Ethernet

“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

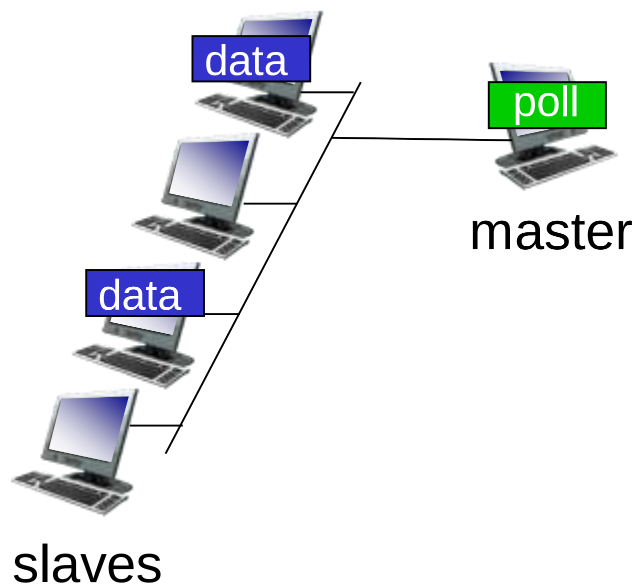
“taking turns” protocols

look for best of both worlds!

"Taking turns" MAC protocols

polling:

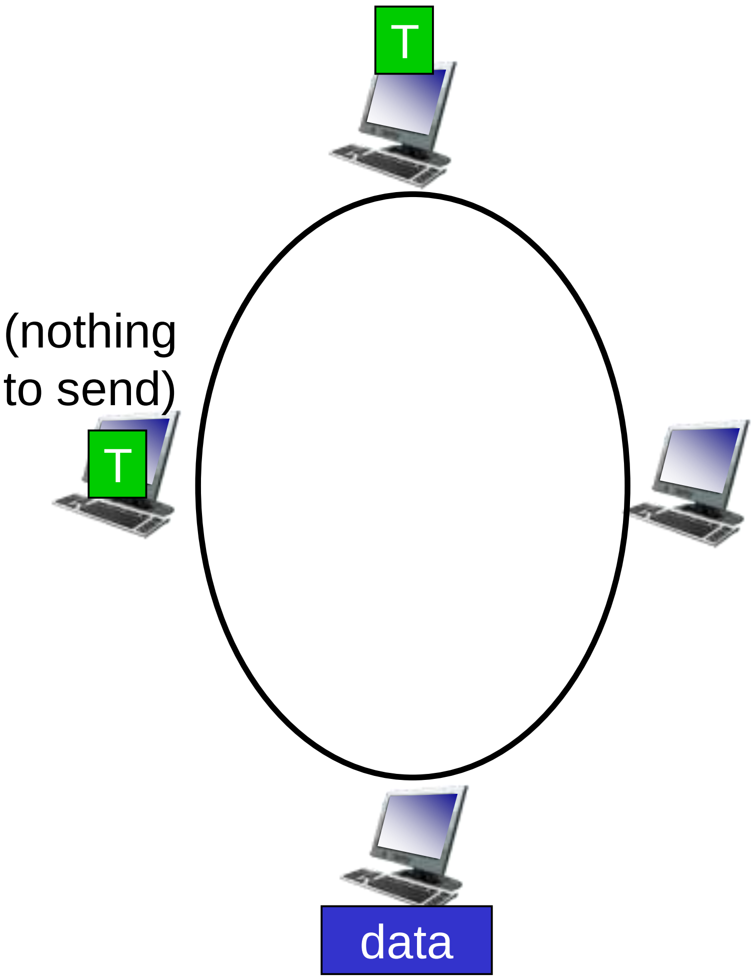
- master node "invites" slave nodes to transmit in turn
- typically used with "dumb" slave devices
  - Bluetooth
- concerns:
  - polling overhead
  - latency
  - single point of failure (master)



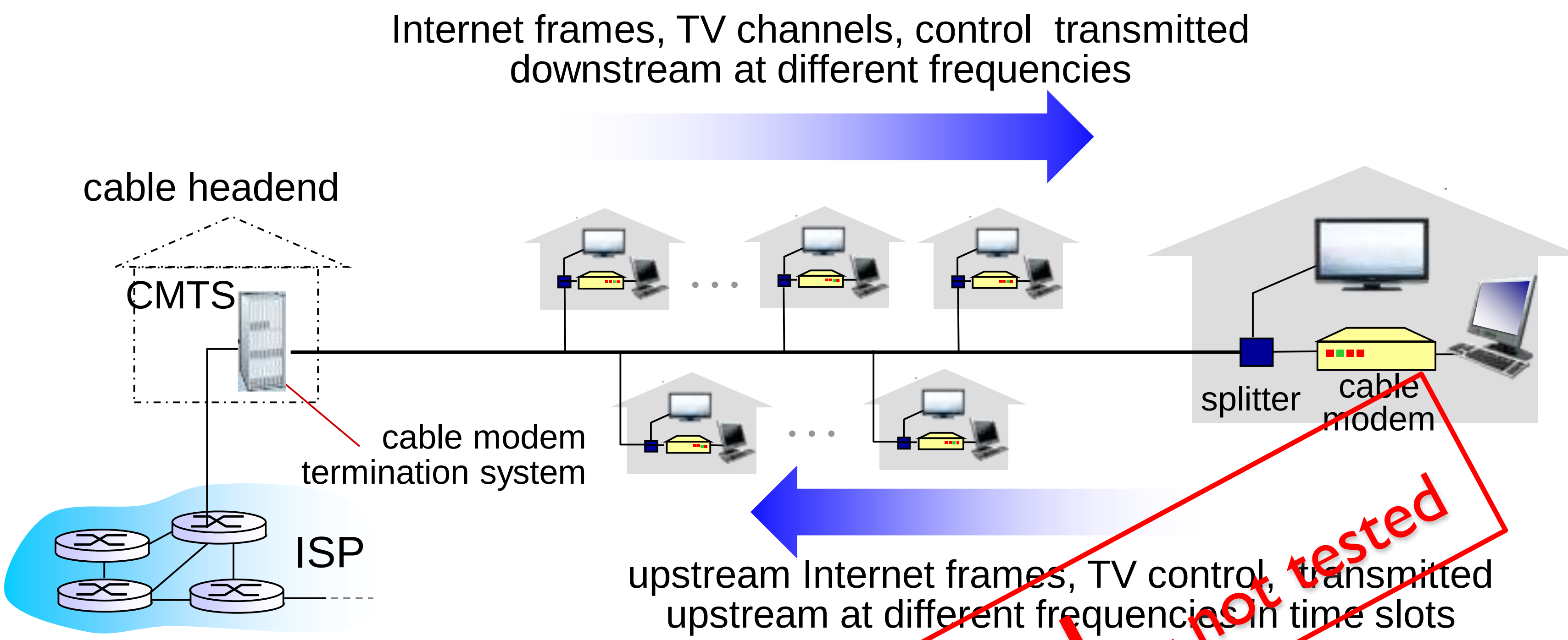
“Taking turns” MAC protocols

token passing:

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

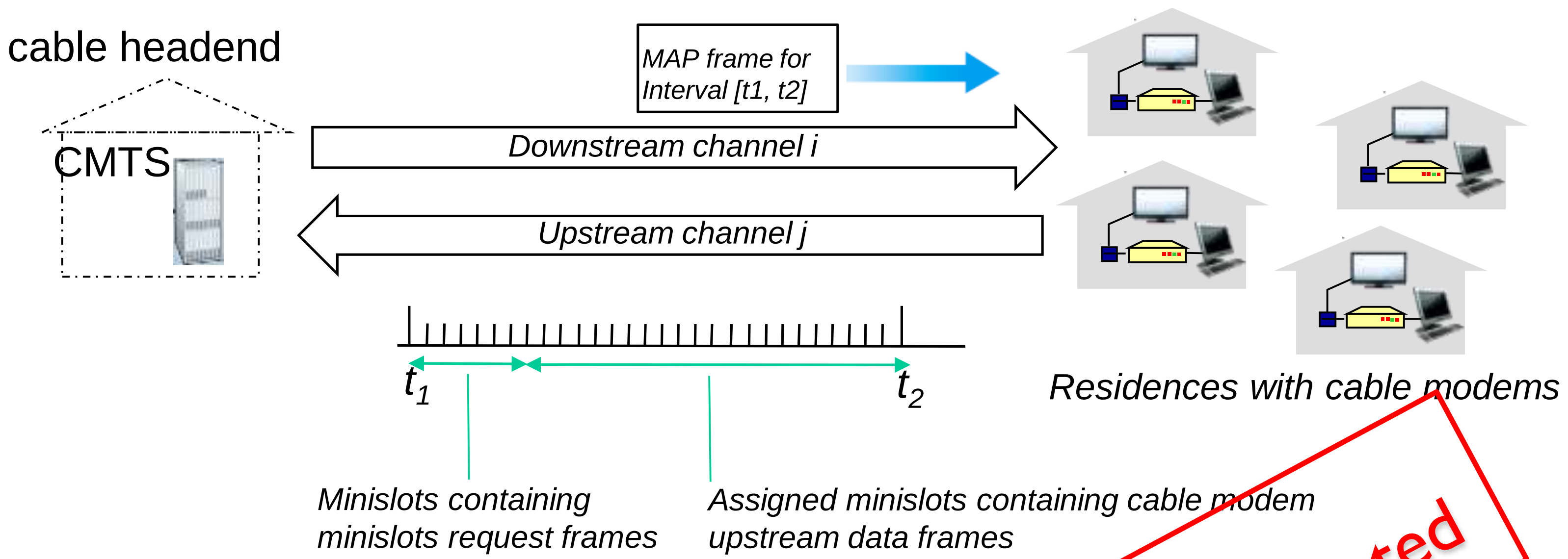


Cable access network



- 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



Optional – not tested

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*
  - polling from central site, token passing
  - Bluetooth, FDDI, token ring