Computer Networks Basic Protocols

Assoc. Prof. Dr. Berk CANBERK

27 September 2017

References:

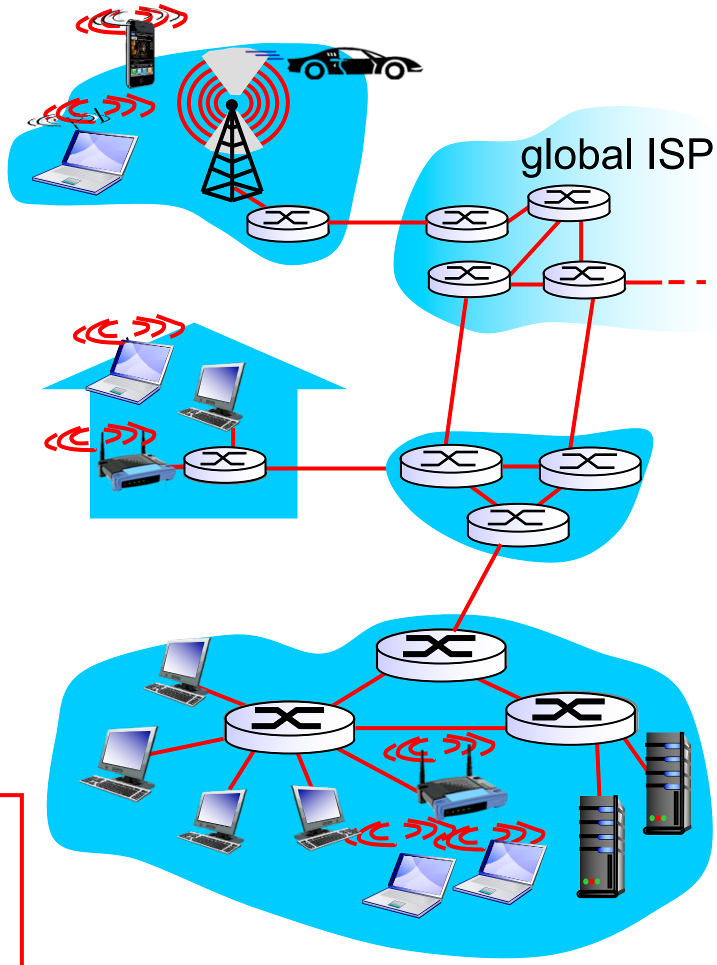
-*Data and Computer Communications*, William Stallings, Pearson-Prentice Hall, 9th Edition, 2010.

-*Computer Networking, A Top-Down Approach Featuring the Internet*, James F.Kurose, Keith W.Ross, Pearson-Addison Wesley, 6th Edition, 2012.

Link layer: introduction

terminology:

- hosts and routers: **nodes**
- communication channels that connect adjacent nodes along communication path: **links**
  - wired links
  - wireless links
  - LANs
- layer-2 packet: **frame,** encapsulates datagram



data-link layer has responsibility of transferring datagram from one node to *physically adjacent* node over a link

Link layer services

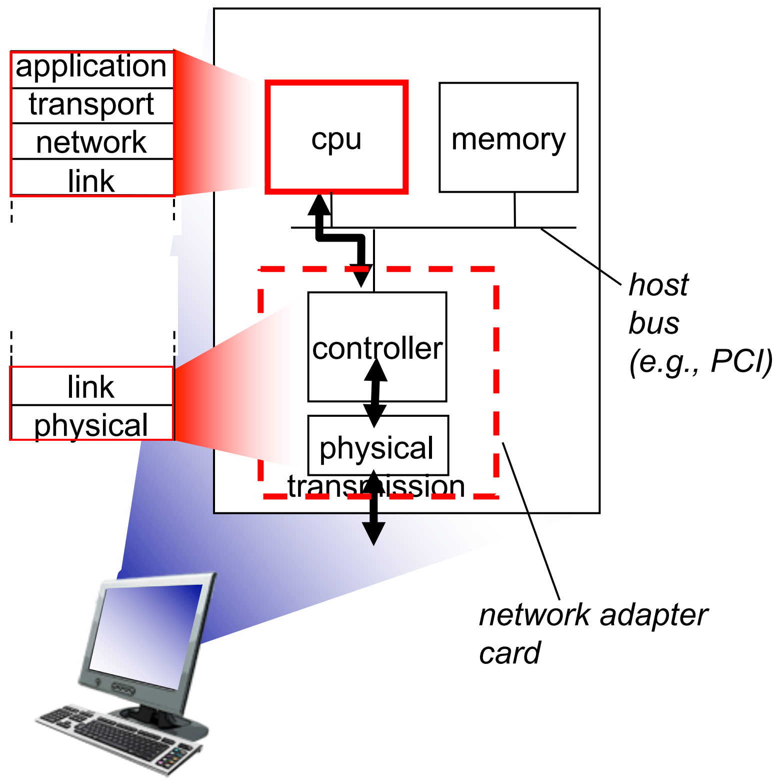
- *framing, link access:*
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - "MAC" addresses used in frame headers to identify source, dest
    - different from IP address!
- *reliable delivery between adjacent nodes*
  - seldom used on low bit-error link (fiber, some twisted pair)

Link layer services (more)

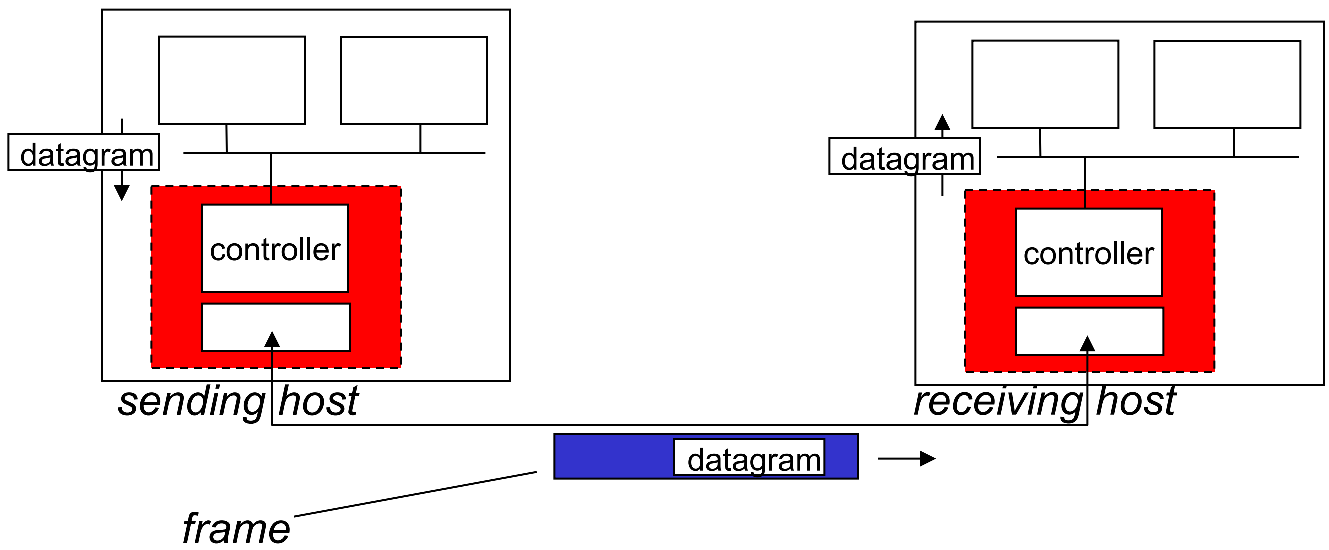
- *flow control:*
  - pacing between adjacent sending and receiving nodes
- *error detection:*
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- *error correction:*
  - receiver identifies *and corrects* bit 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, flow control, etc
  - extracts datagram, passes to upper layer at receiving side

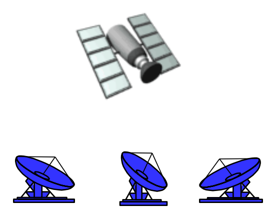
Multiple access links, protocols

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
  - 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)

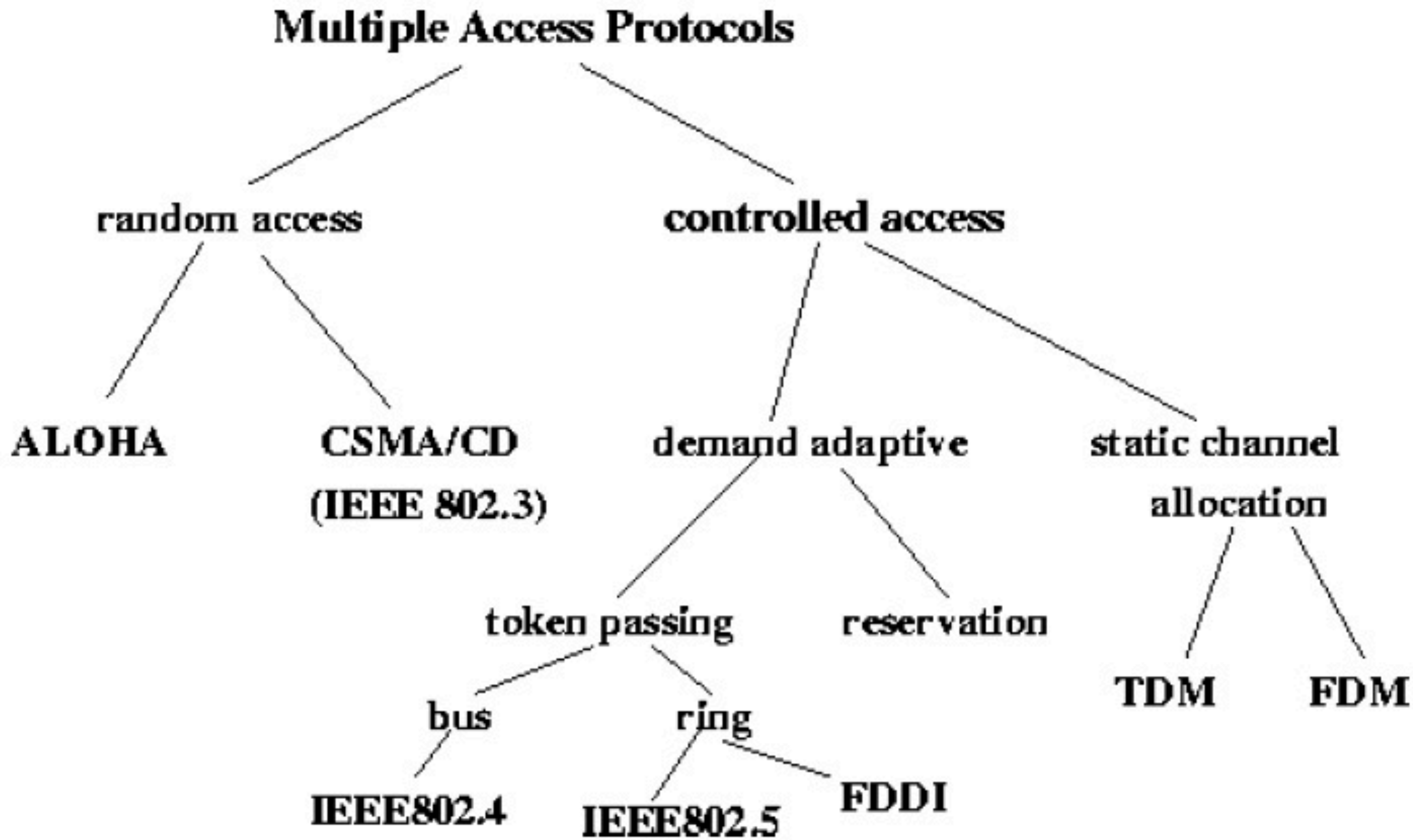
Multiple access protocols

- single shared broadcast channel
- 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

A Simple Taxonomy



MAC protocols: taxonomy

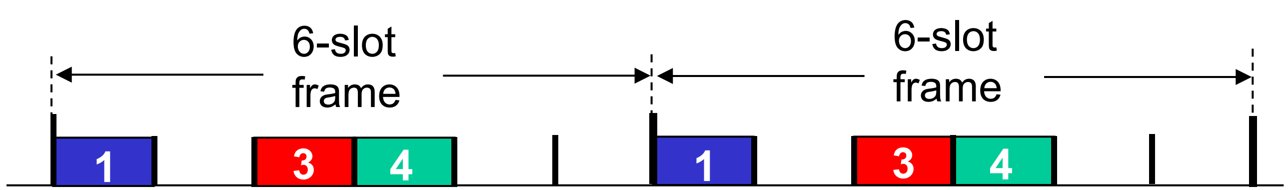
three broad classes:

- *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

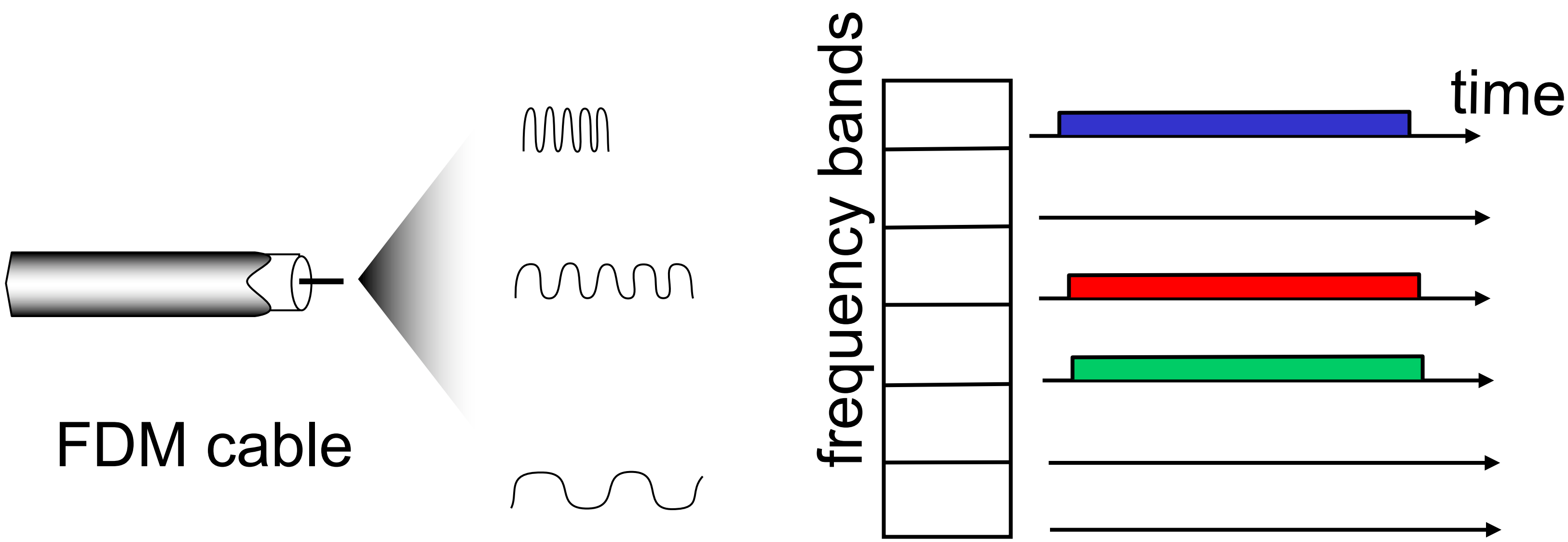
- access to channel in "rounds"
- each station gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle



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

Slotted ALOHA

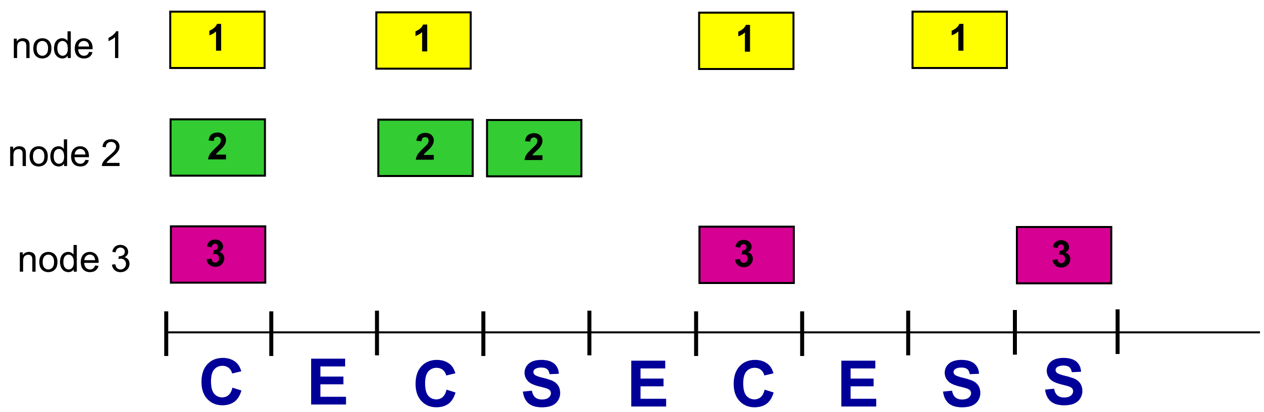
assumptions:

- all frames same size
- time divided into equal size slots (time to transmit 1 frame)
- nodes start to transmit only 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

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:

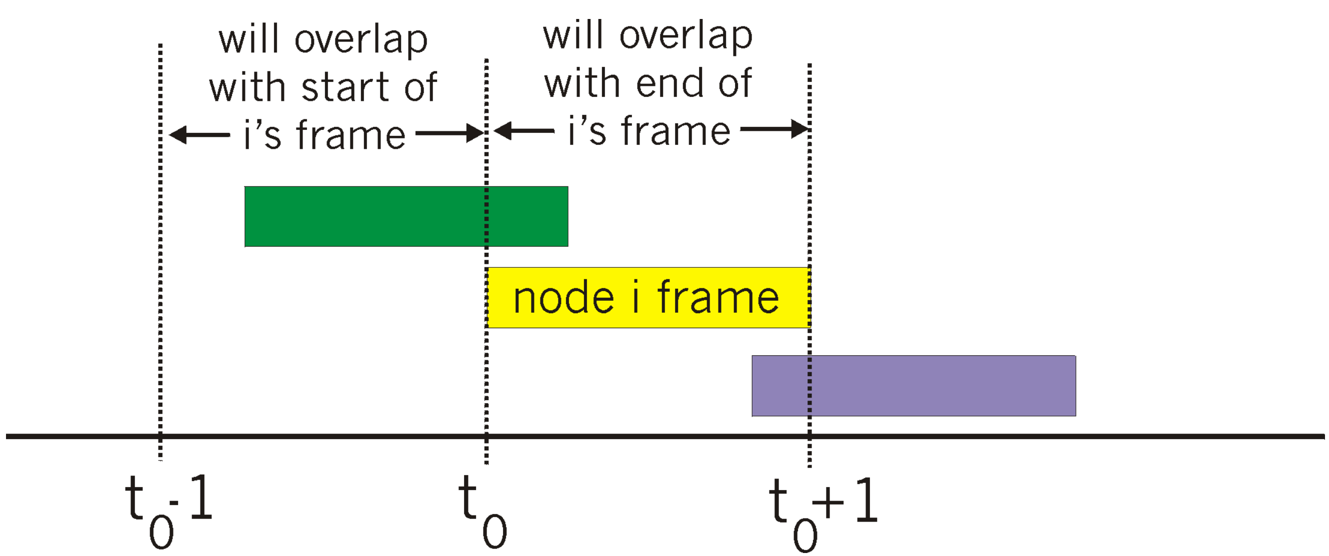
max efficiency = 1/e = .37

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 $[t_0, t_0+1]$

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

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

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

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

even *worse* than slotted Aloha!

CSMA (carrier sense multiple access)

CSMA: listen before transmit:

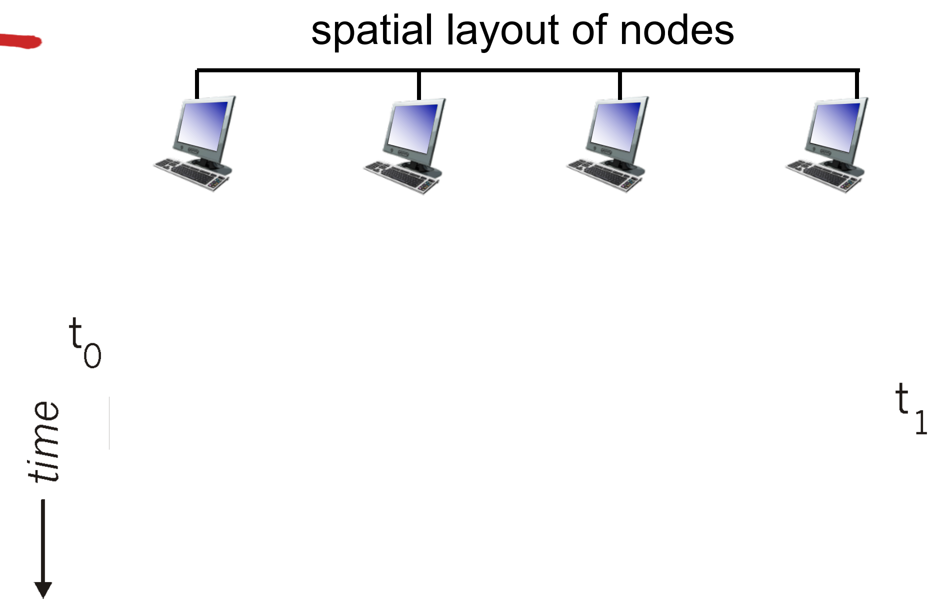
if channel sensed idle: transmit entire frame

- 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



CARRIER SENSE MULTIPLE ACCESS (CSMA)

- CSMA protocol was developed to overcome the problem found in ALOHA i.e. to minimize the chances of collision, so as to improve the performance.

- CSMA protocol is based on the principle of ‘carrier sense’.

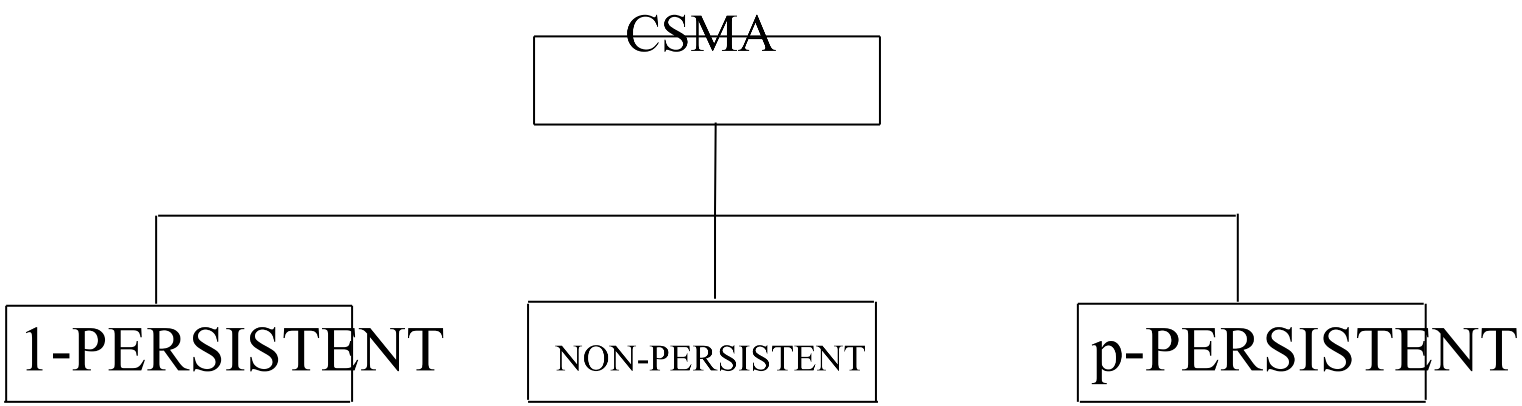
- The chances of collision can be reduce to great extent if a station senses the channel before trying to use it.

- Although CSMA can reduce the possibility of collision, but it cannot eliminate it completely.

- The chances of collision still exist because of propagation delay.

•There are three different types of CSMA protocols :-

(i) 1-Persistent CSMA
(ii) Non-Persistent CSMA
(iii) P-Persistent CSMA

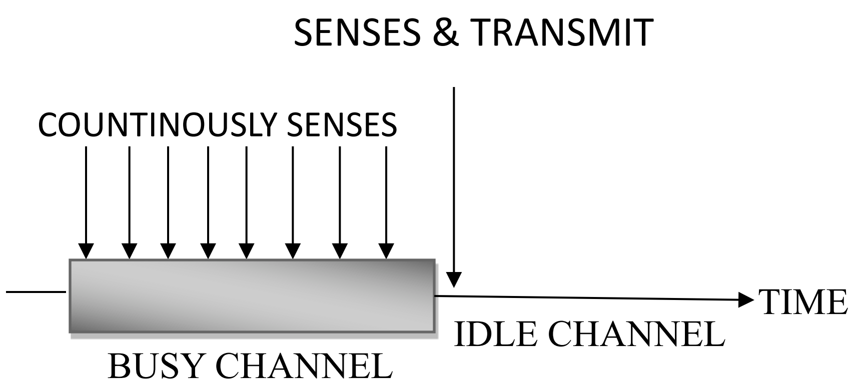


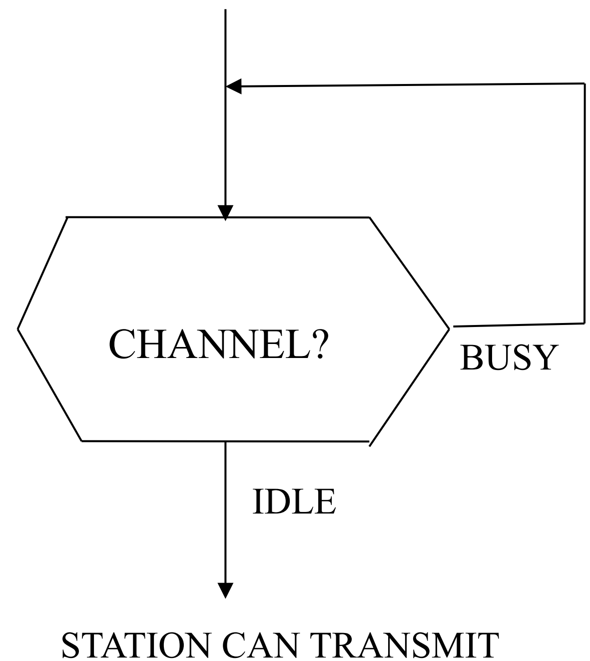
1-Persistent CSMA

- In this method, station that wants to transmit data continuously sense the Channel to check whether the channel is idle or busy.
- If the channel is busy , the station waits until it becomes idle.
- When the station detects an idle channel, it immediately transmits the frame with probability 1. Hence it is called 1-persistent CSMA.
- This method has the highest chance of collision because two or more station may find channel to be idle at the same time and transmit their frames.
- When the collision occurs, the stations wait a random amount of time and start all over again.

Drawback of 1-persistent

- The propagation delay time greatly affects this protocol. Let us suppose, just after the station 1 begins its transmission, station 2 also become ready to send its data and sense the channel. If the station 1 signal has not yet reached station 2, station 2 will sense the channel to be idle and will begin its transmission. This will result in collision.
- Even if propagation delay time is zero, collision will still occur. If two stations become ready in the middle of third station's transmission both stations will wait until the transmission of first station ends and both will begin their transmission exactly simultaneously. This will also result in collision.





Non –persistent CSMA

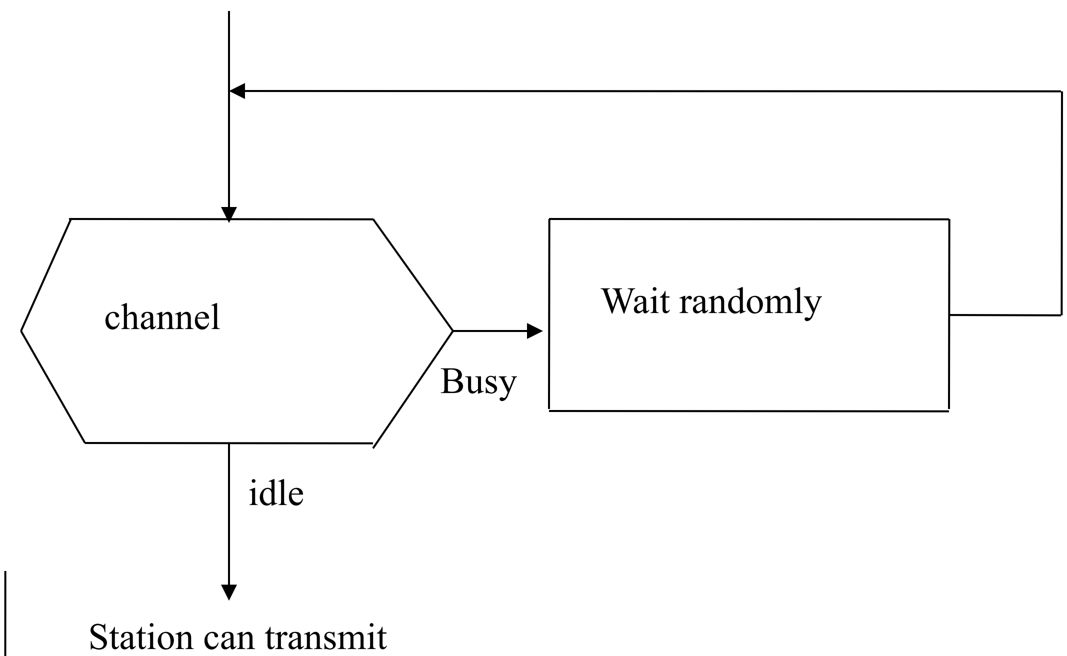
- A station that has a frame to send senses the channel.
- If the channel is idle, it sense immediately.
- If the channel is busy, it waits a random amount of time and then senses the channel again.
- In non-persistent CSMA the station does not continuously sense the channel for purpose of capturing it when it defects the end of precious transmission .

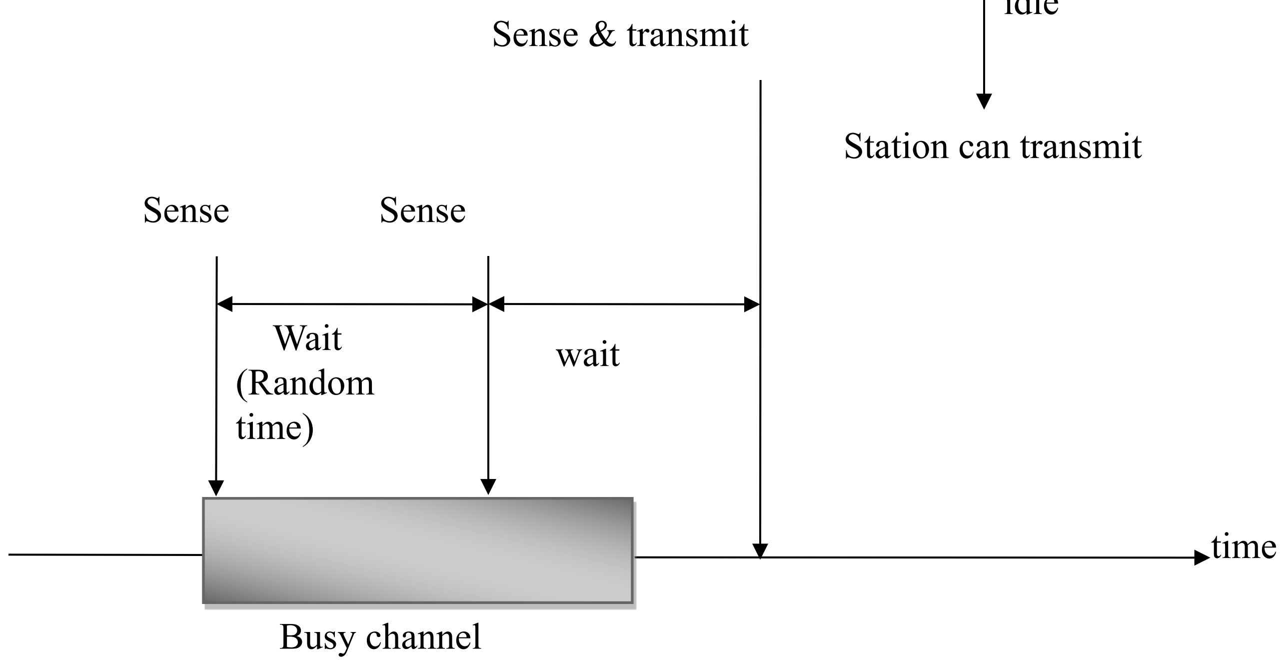
Advantages of non-persistent

- It reduces the chances of collision because the stations wait a random amount of time. It is unlikely that two or more stations Will wait for same amount of time and will retransmit at the same time.

Disadvantages of non-persistent

- It reduces the efficiency of network because the channel remains idle when there may be station with frames to send. This is due to the fact that the stations wait a random amount of time after the collision.


channel
Wait randomly
Busy
idle
Station can transmit

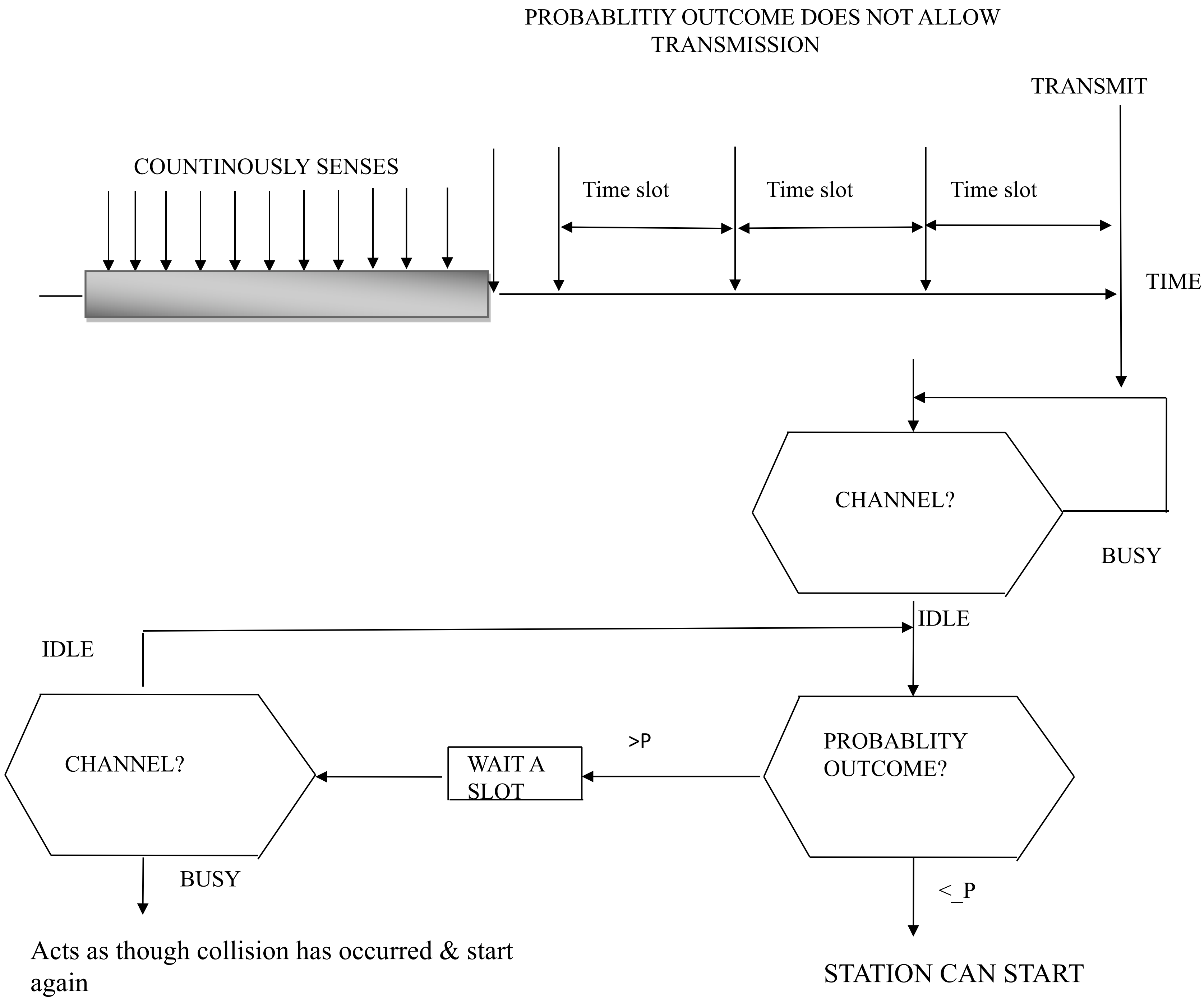

Sense & transmit
Sense
Sense
Wait
(Random
time)
wait
time
Busy channel

p-persistent CSMA

- This method is used when channel has time slots such that the time slot duration is equal to or greater than the maximum propagation delay time.
- Whenever a station becomes ready to send the channel.
- If channel is busy, station waits until next slot.
- If the channel is idle, it transmits with a probability p.
- With the probability q=1-p, the station then waits for the beginning of the next time slot.
- If the next slot is also idle, it either transmits or wait again with probabilities p and q.
- This process is repeated till either frame has been transmitted or another station has begun transmitting.
- In case of the transmission by another station, the station act as though a collision has occurred and it waits a random amount of time and starts again.

Advantages of p-persistent

- it reduce the chances of collision and improve the efficiency of the network.


PROBABLITIY OUTCOME DOES NOT ALLOW TRANSMISSION
TRANSMIT
COUNTINOUSLY SENSES
Time slot
Time slot
Time slot
TIME
CHANNEL?
BUSY
IDLE
IDLE
CHANNEL?
WAIT A SLOT
>P
PROBABLITY OUTCOME?
BUSY
<_P
Acts as though collision has occurred & start again
STATION CAN START

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

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·512 bit times, returns to Step 2
   - longer backoff interval with more collisions

“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

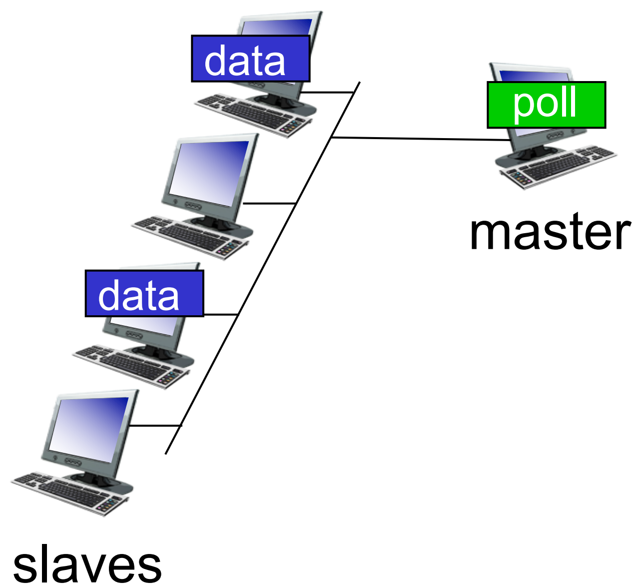
“taking turns” protocols

look for best of both worlds!

"Taking turns" MAC protocols

polling:

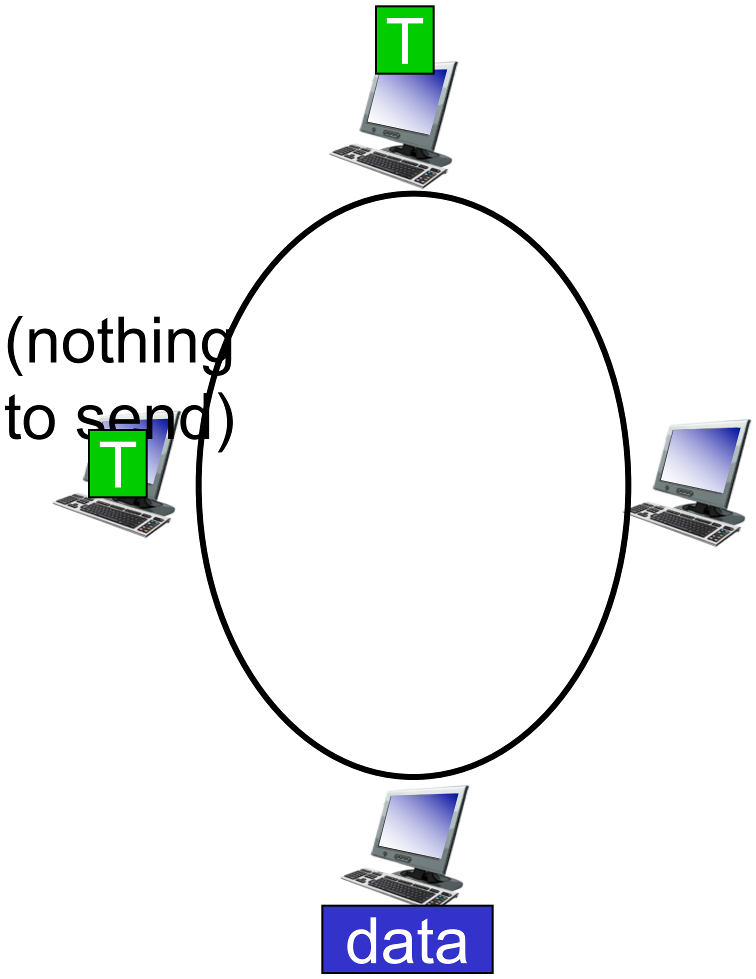
- master node "invites" slave nodes to transmit in turn
- typically used with "dumb" slave devices
- concerns:
  - polling overhead
  - latency
  - single point of failure (master)



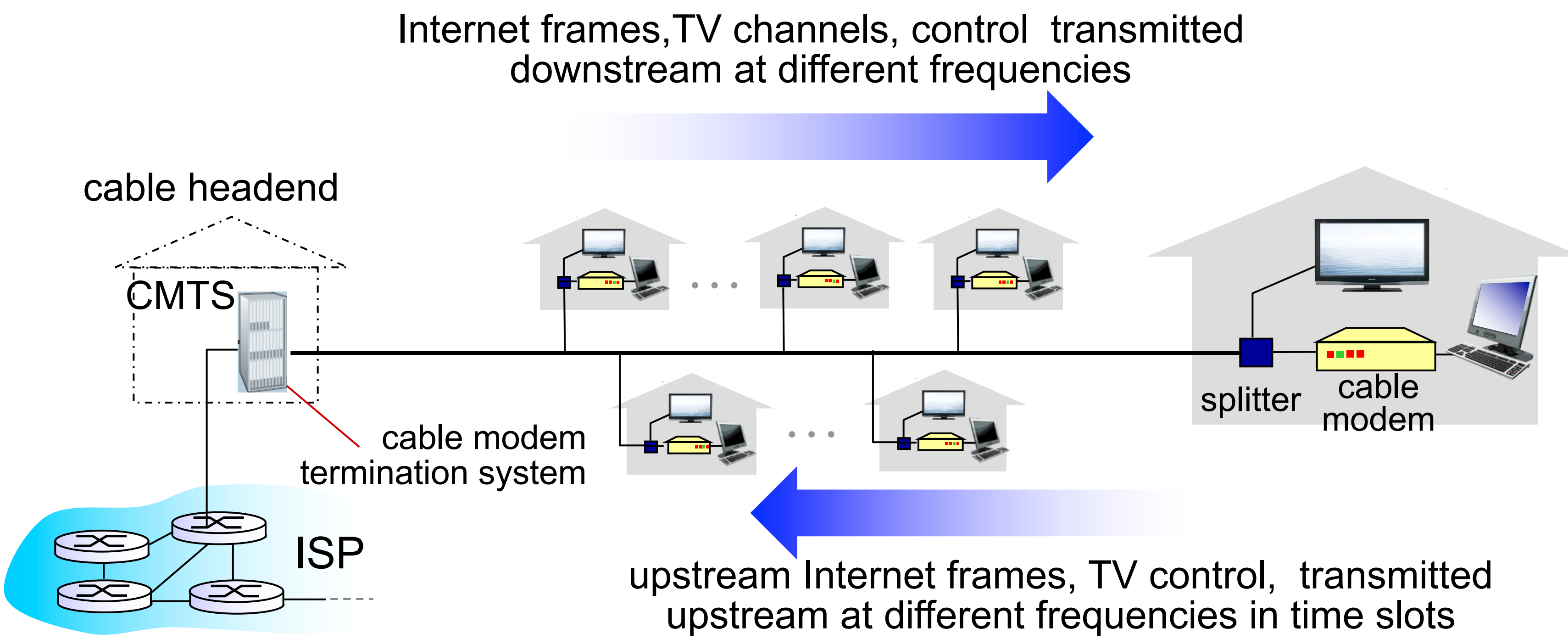
"Taking turns" MAC protocols

token passing:

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

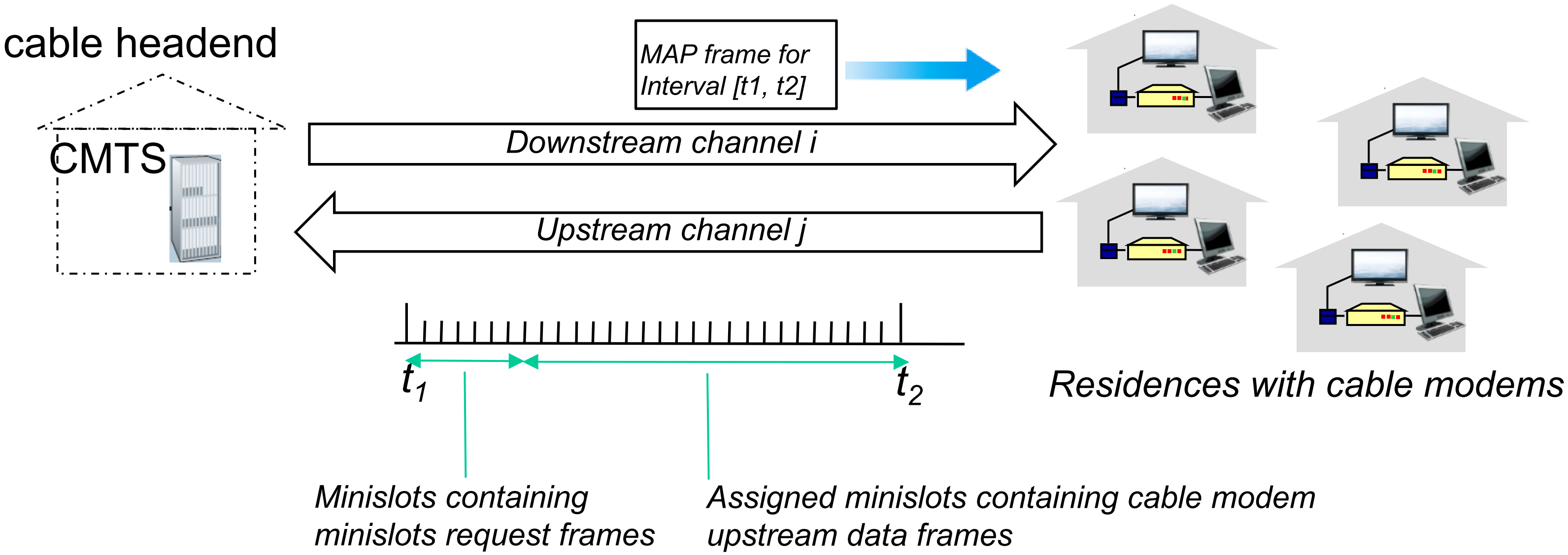


Cable access network



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