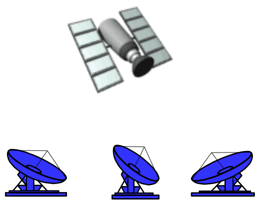
Multiple Access Links: MAC Protocols

two types of "links":

- point-to-point
  - PPP for dial-up access
  - point-to-point link between Ethernet switch, host (PPPoE)
- *broadcast (shared wire or medium)*
  - old-fashioned Ethernet
  - upstream HFC
  - 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)

Broadcast LAN: Media Access Control

- Broadcast LAN: single shared broadcast channel
  - two or more simultaneous transmissions by nodes: interference!
    - *collision* if node receives two or more signals at the same time
  - only one node can send successfully at a time!
- How to share a broadcast channel?
  - Humans use multi-access protocols all the time

Multiple Access Protocol

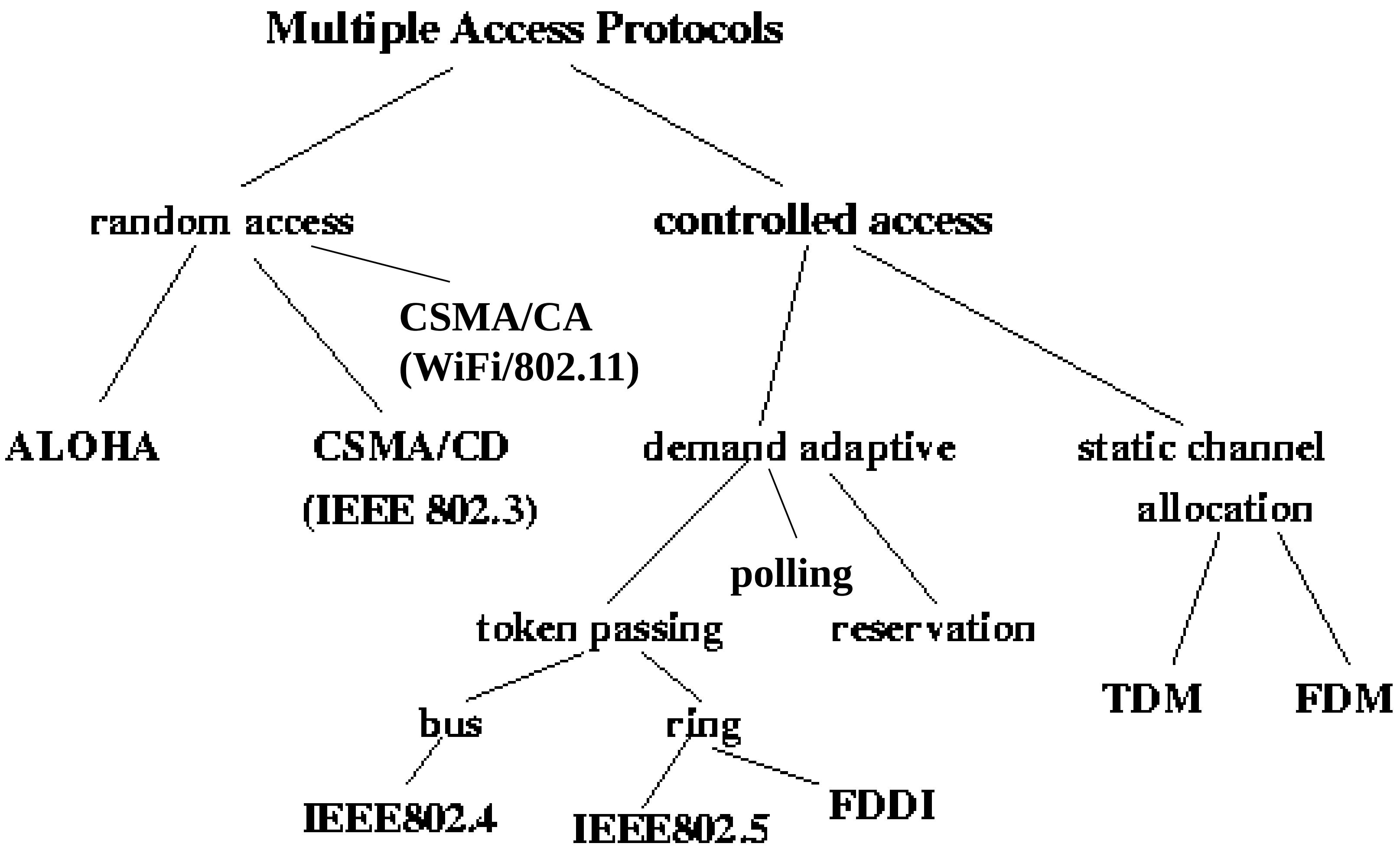
- distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!
- what to look for in multiple access protocols:
  - synchronous or asynchronous
  - information needed about other stations
  - robustness
  - performance: access delay and throughput

MAC Protocols: a Taxonomy

Three broad classes:

- Channel Partitioning (static controlled access)
  - divide channel into smaller "pieces" (e.g., time slots -> TDMA, frequency->FDMA, code->CDMA)
  - allocate piece to node for exclusive use
- "Demand Adaptive" Controlled Access: e.g., Polling or Taking Turns
  - tightly coordinate shared access to avoid collisions
- Random Access
  - channel not divided, allow collisions
  - "recover" from collisions

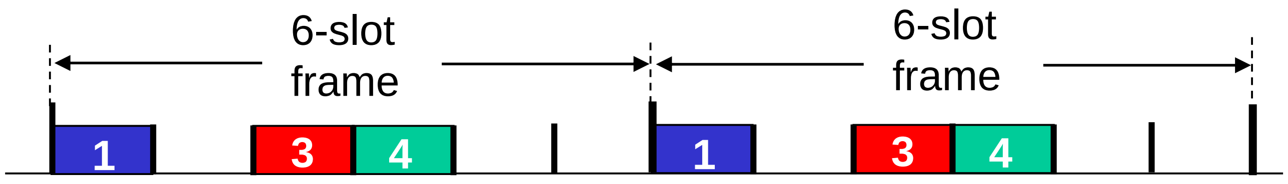
Taxonomy of MAC Protocols



Channel Partitioning MAC protocols: TDMA

TDMA: time division multiple access

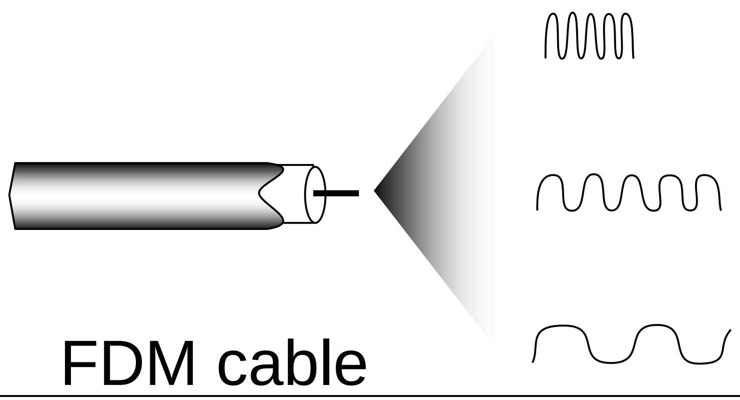
- access to channel in "rounds"
- each station gets fixed length slot (length = packet transmission time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have packets to send, slots 2,5,6 idle

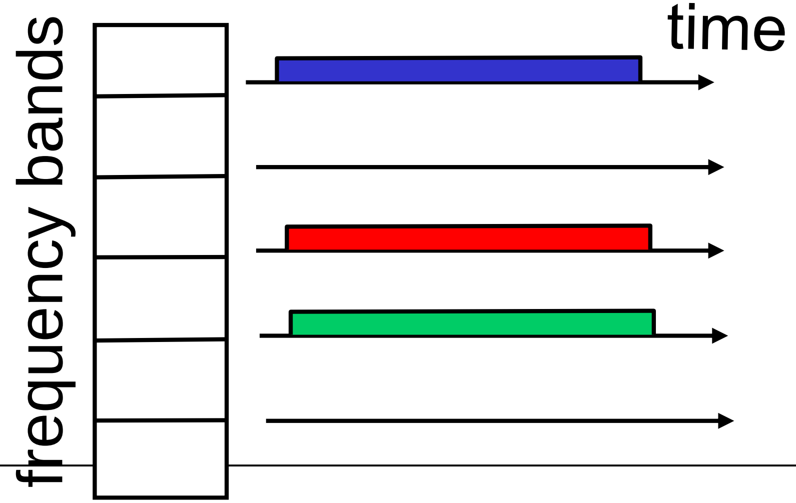


Channel Partitioning MAC Protocols: FDMA

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





"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!

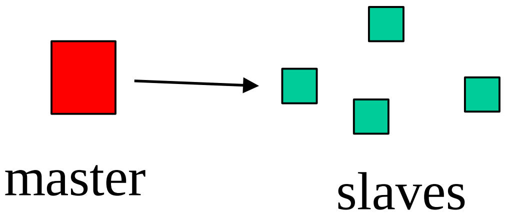
"Demand-Adaptive" Controlled Protocols

- Human analogy:
  - traffic control with green/red light
    - fixed time vs. adaptive time vs. no lights at all
- (Master-Slave based) Polling:
  - e.g., in a classroom: I am the "master" ;-)
- "Taking Turns" via token-passing:
  - e.g., a round-table panel with a single microphone

"Taking Turns" MAC Protocols

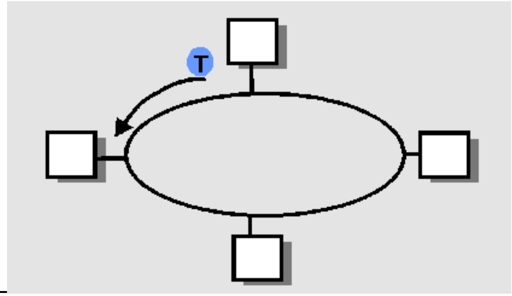
Polling:

- centralized
- master node "invites" slave nodes to transmit in turn
- concerns:
  - polling overhead
  - latency
  - single point of failure (master)

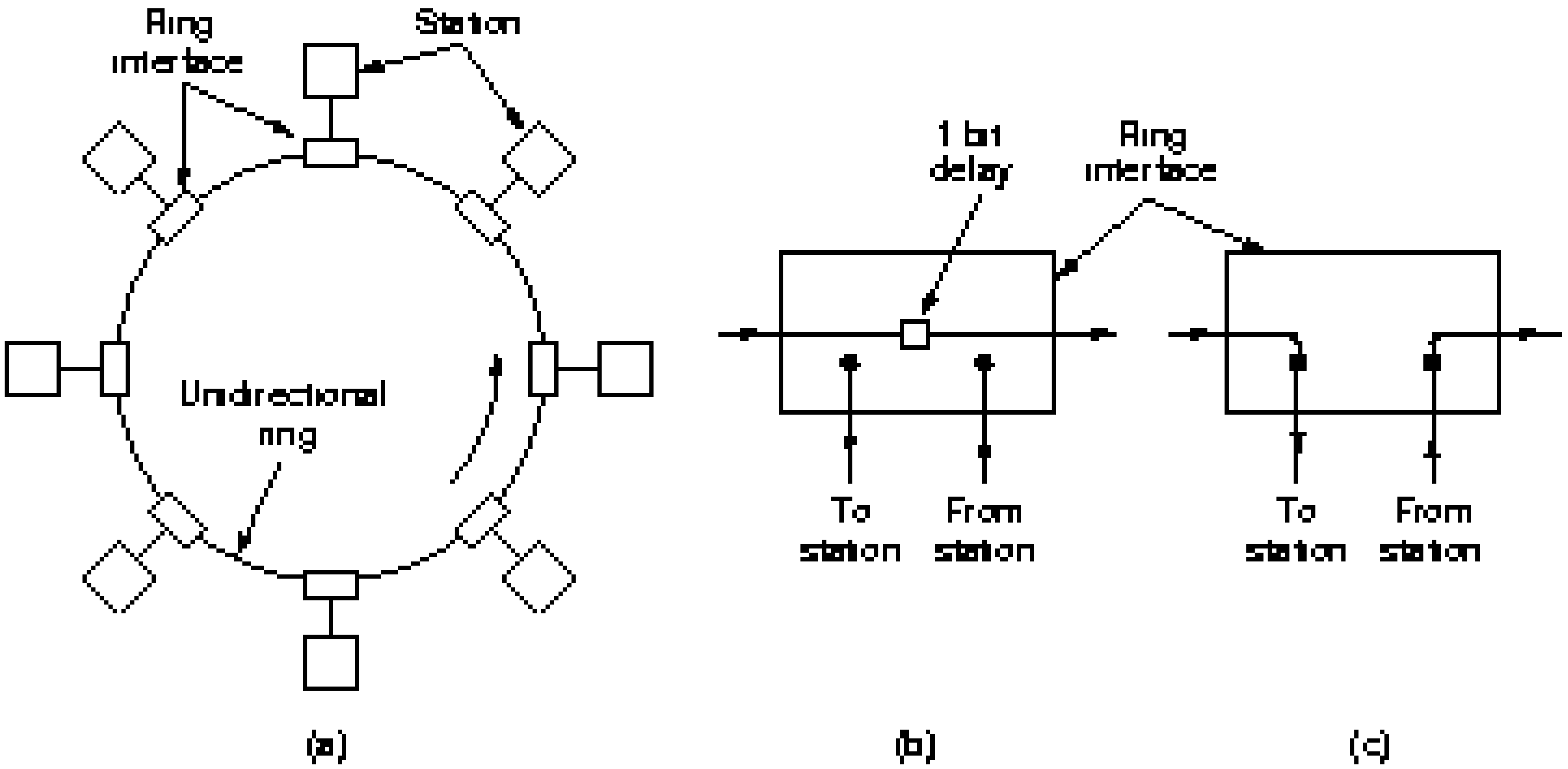


Token passing:

- distributed
- control **token** passed from one node to next sequentially.
- what is a **token**? a special control message
- concerns:
  - token overhead
  - latency
  - single point of failure (token)

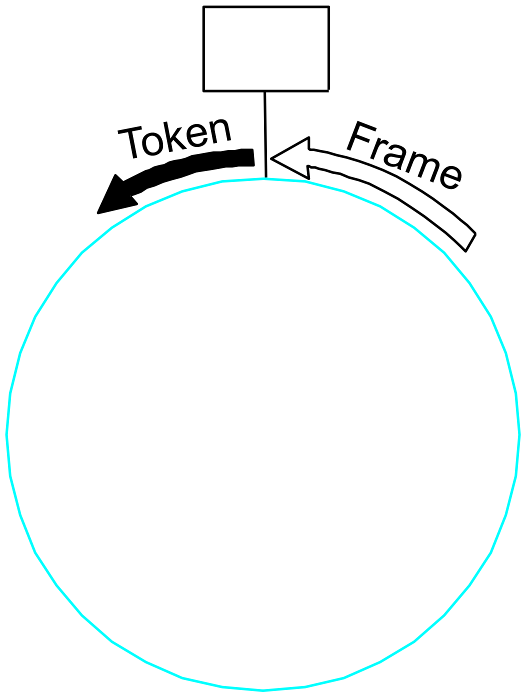


Token Ring Topology

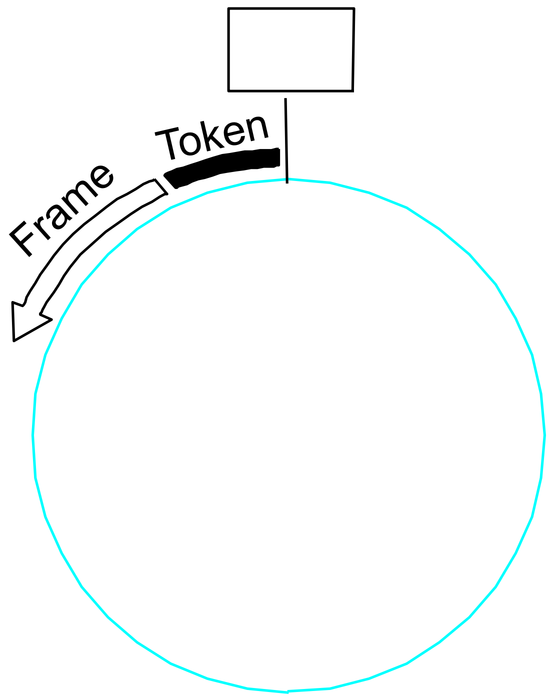


Using token-passing, nodes do not have to form a physical ring! E.g., token bus: all nodes connected via a bus, forming a *logical* ring!)

Token Release



Release after Reception
(used by Token Ring)



Release after Transmission
(used by FDDI)

Token Ring Performance

- Efficiency with "release after reception"

$$\approx \frac{1}{1+a}$$

where $$a = \frac{PROP}{TRANS}$$

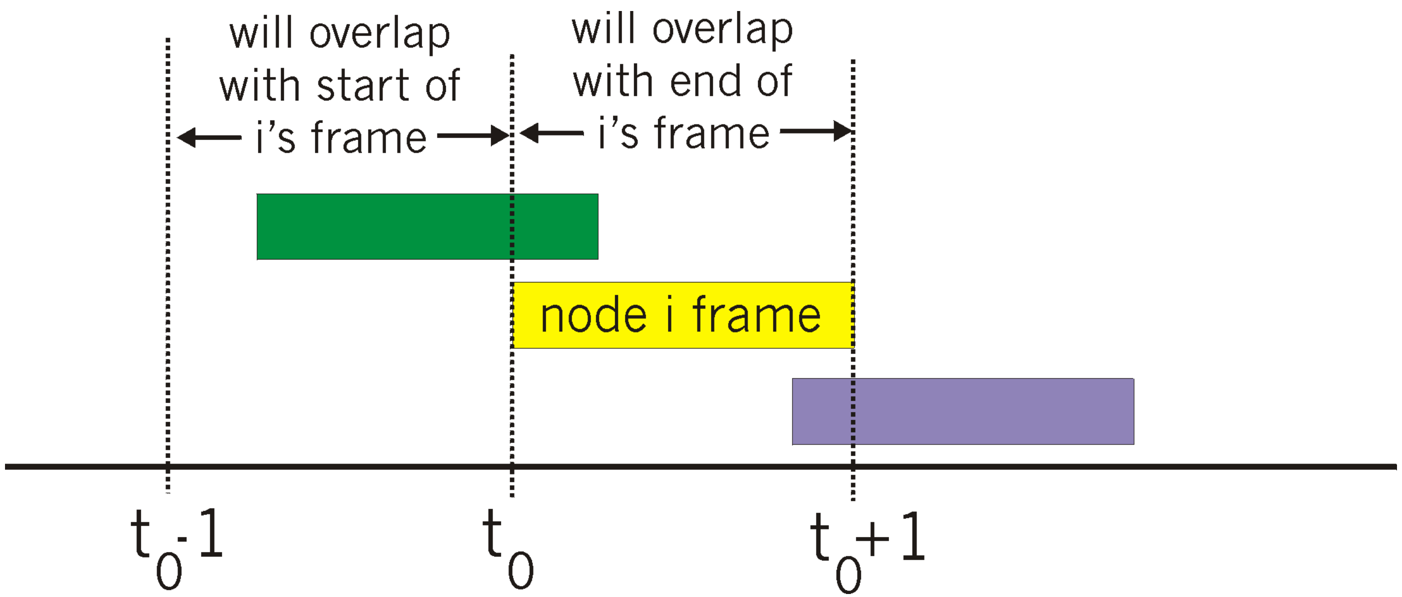
- What is the efficiency with "release after transmission" ?

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 or avoid 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

Pure (unslotted) ALOHA

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



Slotted ALOHA

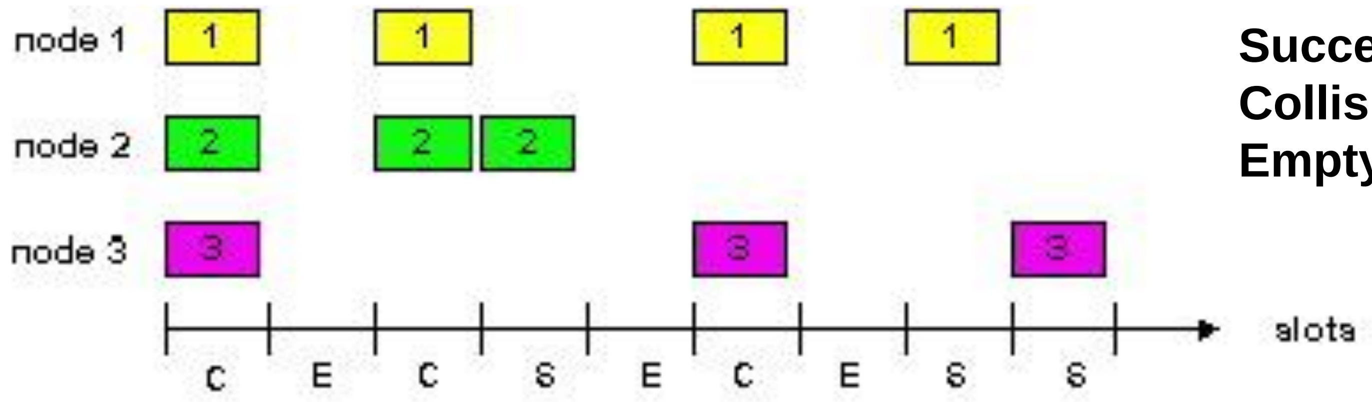
Assumptions

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

Operation

- when node obtains fresh frame, it transmits in next slot
- 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



Success (S), Collision (C), Empty (E) slots

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

Slotted Aloha efficiency

Efficiency is the long-run fraction of successful slots when there's many nodes, each
with many frames to send

- Suppose N nodes with many frames to send, each transmits in slot with probability *p*
- prob that 1st node has success in a slot = $p(1-p)^{N-1}$
- prob that any node has a success = $Np(1-p)^{N-1}$

- For max efficiency with N nodes, 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 1/e = .37

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

Pure Aloha Efficiency

P(success by given node) = P(node transmits) $\cdot$

P(no other node transmits in $[p_0\text{-}1,p_0]$ $\cdot$

P(no other node transmits in $[p_0,p_0+1]$

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

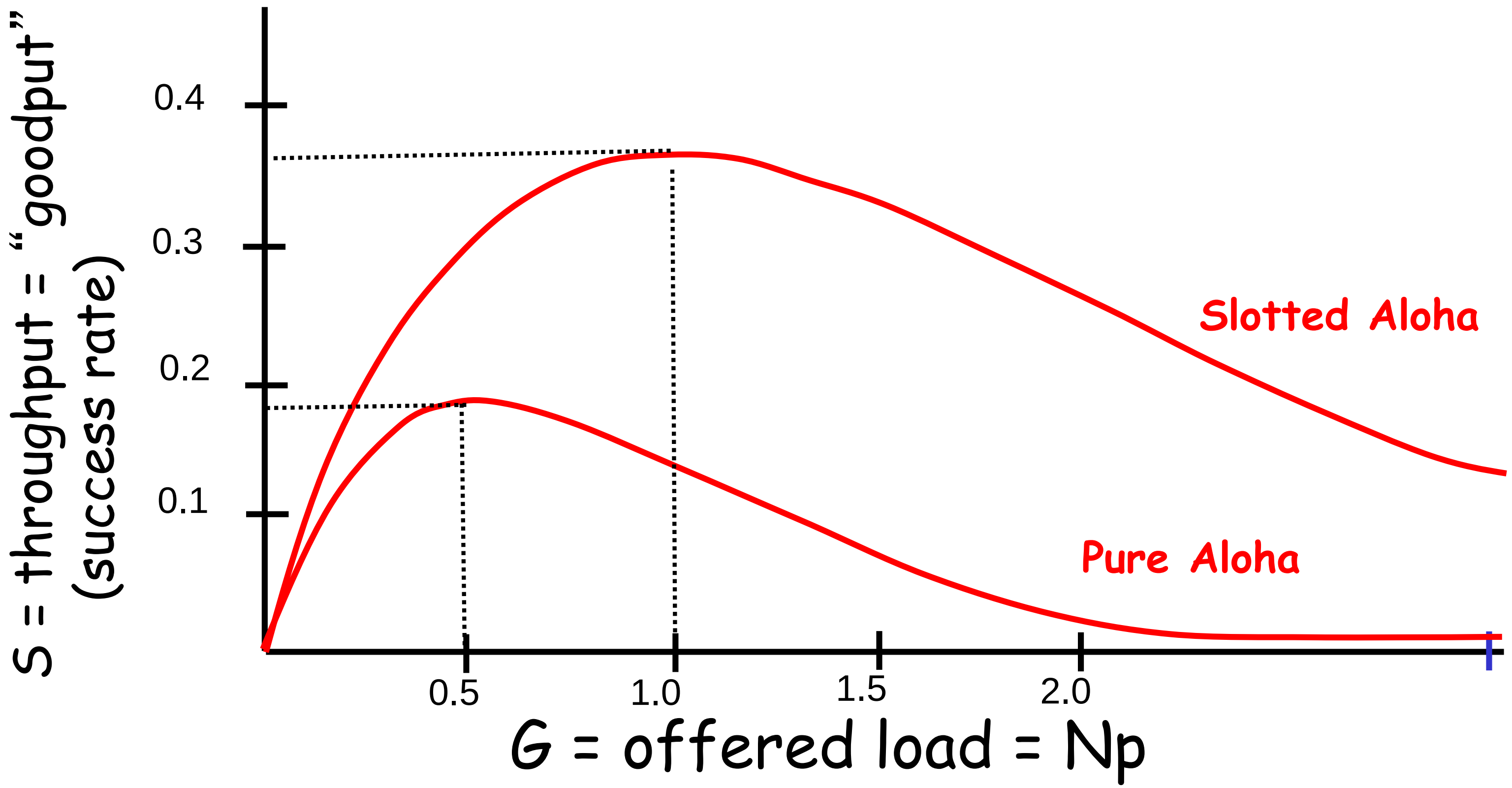
$= p \cdot (1-p)^{2(N-1)}$

… choosing optimum p and then letting n -> infty ...

= 1/(2e) = .18

Efficiency is even worse !

Performance of Aloha Protocols

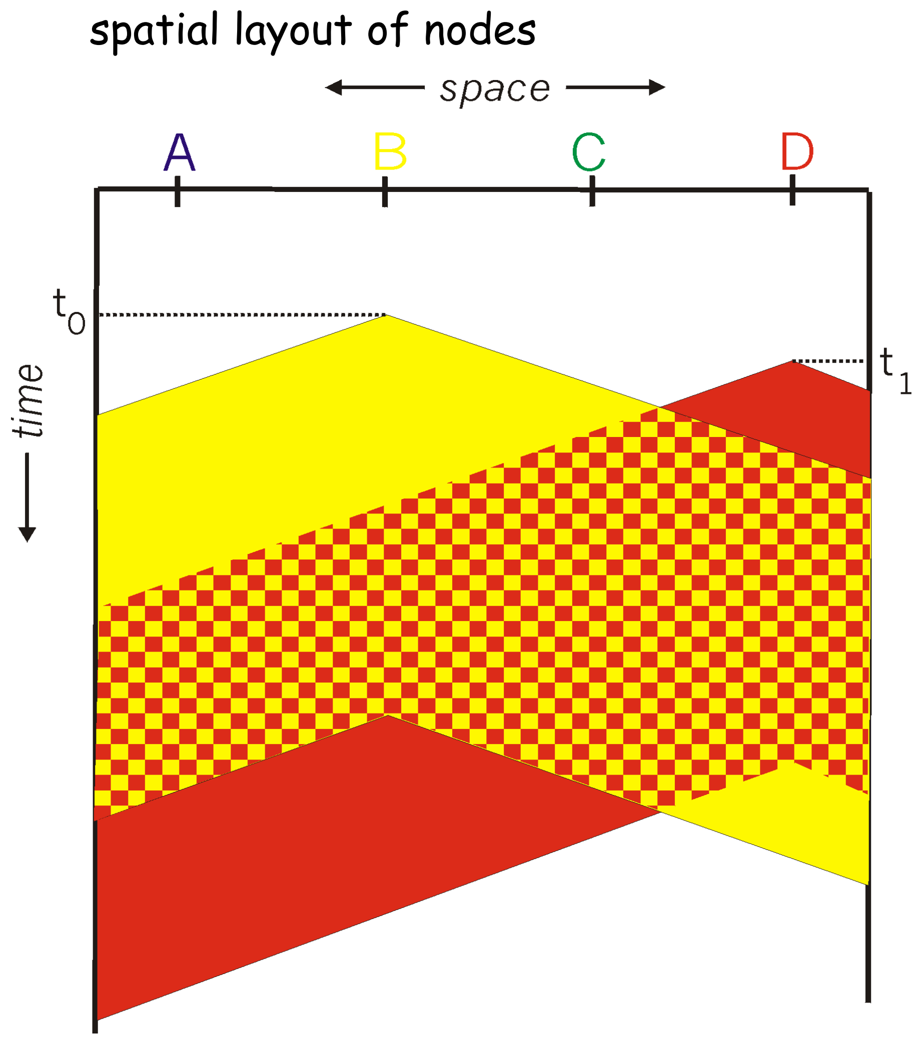


Can we do better with random access?

Carrier Sense Multiple Access

- Aloha is inefficient (and rude):
  - doesn't listen before talking
- CSMA: Listen before transmit
  - Human analogy: don't interrupt others!
  - If channel idle, transmit entire packet
  - If busy, defer transmission
    - How long should we wait?
- Persistent vs. Nonpersistent CSMA
  - Nonpersistent:
    - if idle, transmit
    - if busy, wait random amount of time
  - p-persistent
    - If idle, transmit with probability p
    - If busy, wait till it becomes idle
    - If collision, wait random amount of time
- Can carrier sense avoid collisions completely?

CSMA Collisions



collisions can still occur:
propagation delay means two nodes may not hear each other's transmission

collision:
entire packet transmission time wasted

note:
role of distance & propagation delay in determining collision probability

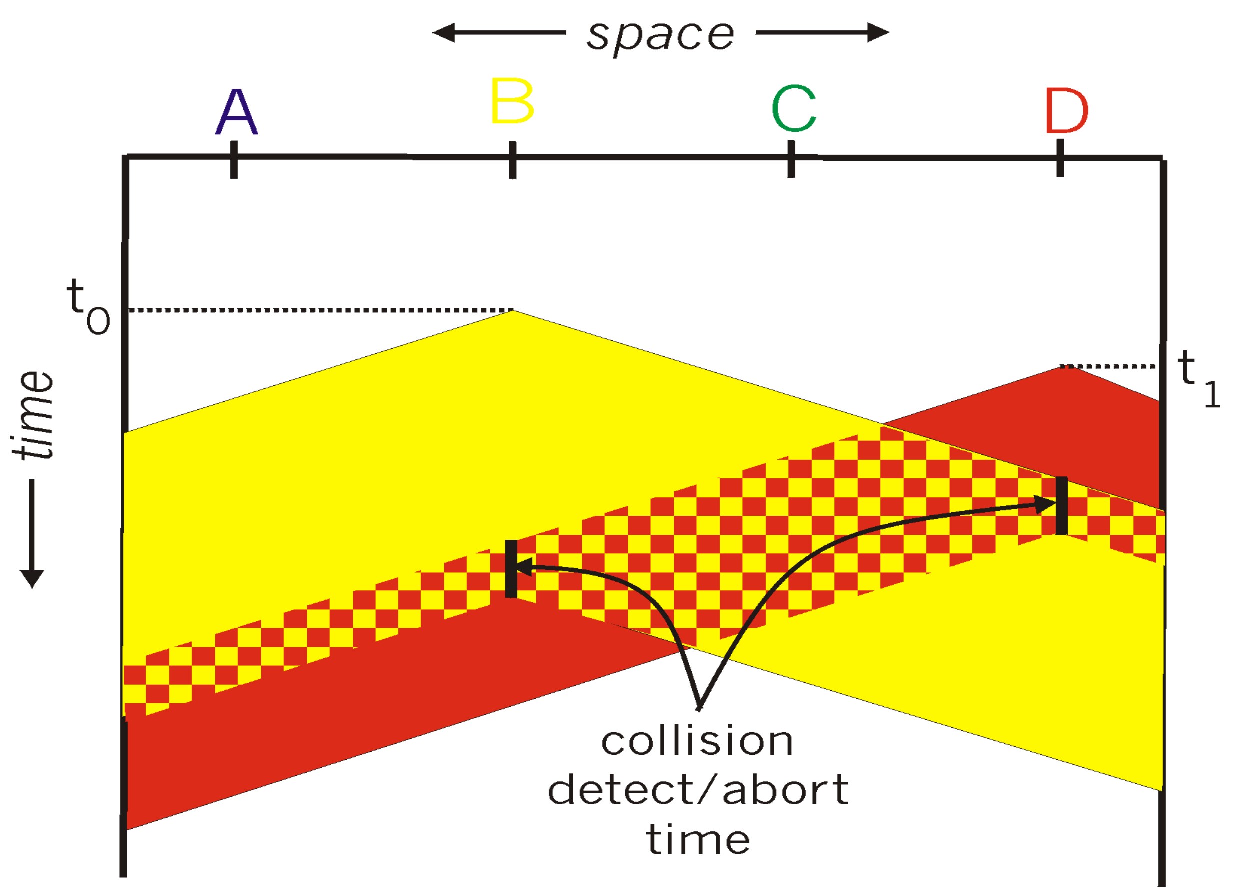
CSMA/CD (Collision Detection)

CSMA/CD: carrier sensing, deferral as in CSMA

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

- human analogy: the polite conversationalist
  - talking while keep listening, stop if collision detected
- How to detect collision?
  - easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - difficult in wireless LANs: receiver shut off while transmitting

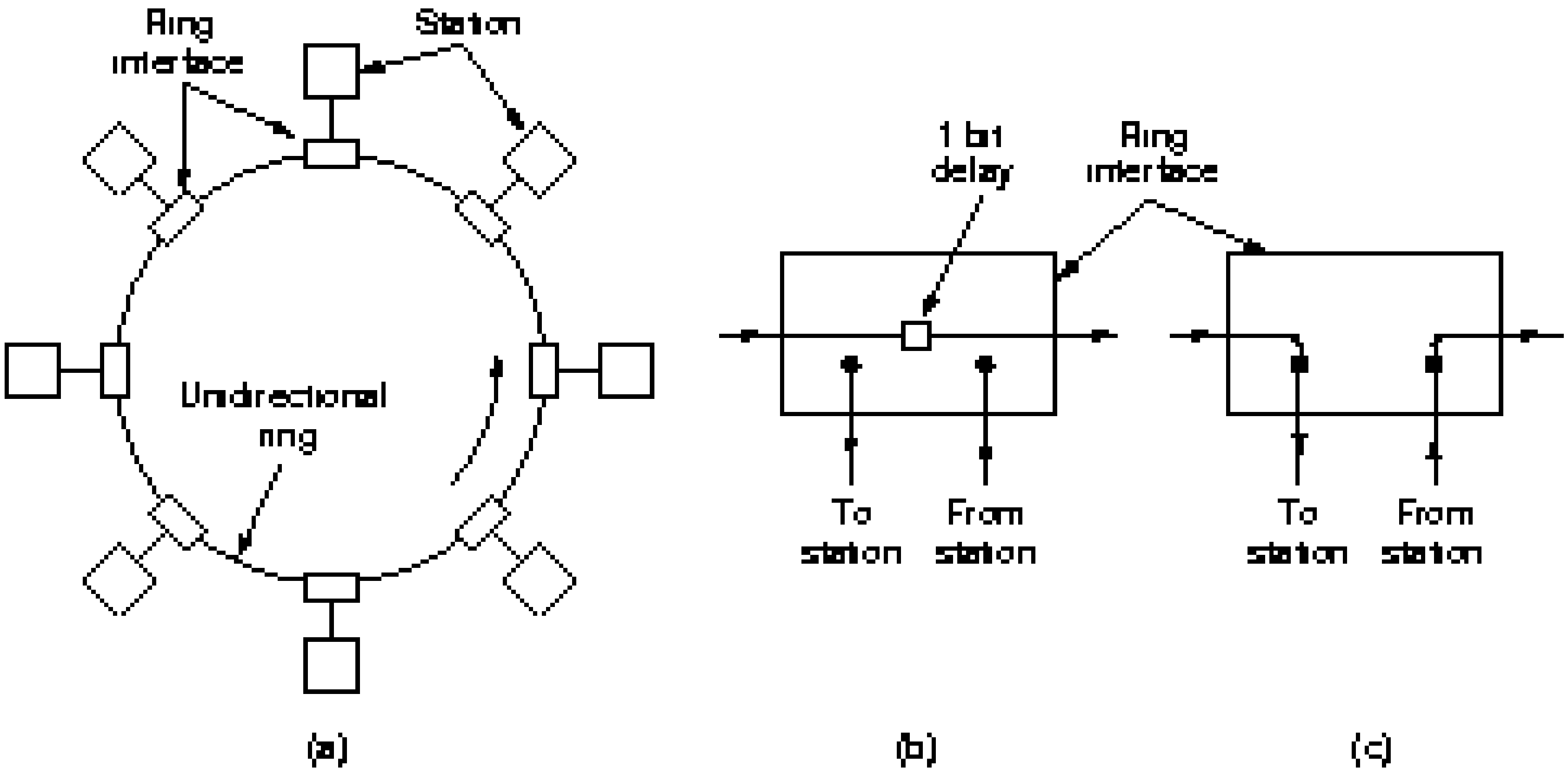
CSMA/CD: Illustration



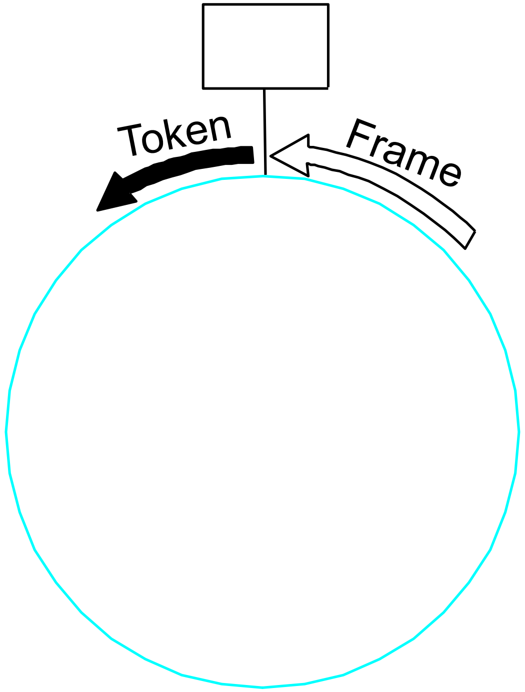
Token Ring (IEEE 802.5)

- Station
  - Wait for token to arrive
  - Hold the token and start data transmission
    - Maximum token holding time ➔ max packet size
  - Strip the data frame off the ring
    - After it has gone around the ring
  - When done, release the token to next station
- When no station has data to send
  - Token circulates continuously
  - Ring must have sufficient delay to contain the token

Ring Topology



Token Release after Reception



Release after Reception

In token passing protocols, sender is always responsible for removing the frame it has transmitted! (Why?)

Tokens and Data Frames

8	8	8	48	48	Variable	32	8	8
Start delimiter	Access control	Frame control	Dest addr	Src addr	Body	Checksum	End delimiter	Frame status

Token Ring Frame Fields

- Access Control
  - Token bit: 0 → token 1 → data
  - Monitor bit: used for monitoring ring
  - Priority and reservation bits: multiple priorities
- Frame Status
  - Set by destination, read by sender
- Frame control
  - Various control frames for ring maintenance

Priority and Reservation

- Token carries priority bits
  - Only stations with frames of equal or higher priority can grab the token
- A station can make reservation
  - When a data frame goes by
  - If a higher priority has not been reserved
- A station raising the priority is responsible for lowering it again

Ring Maintenance

- Each ring has a monitor station
- How to select a monitor?
  - Election/self-promotion: CLAIM_TOKEN
- Responsibilities
  - Insert additional delay
    - To accommodate the token
  - Check for lost token
    - Regenerate token
  - Watch for orphan frames
    - Drain them off the ring
  - Watch for garbled frames
    - Clean up the ring and regenerate token

Fault Scenarios

- What to do if ring breaks?
  - Everyone participates in detecting ring breaks
  - Send beacon frames
  - Figure out which stations are down
  - By-pass them if possible
- What happens if monitor dies?
  - Everyone gets a chance to become the new king
- What if monitor goes berserk?

Token Ring Summary

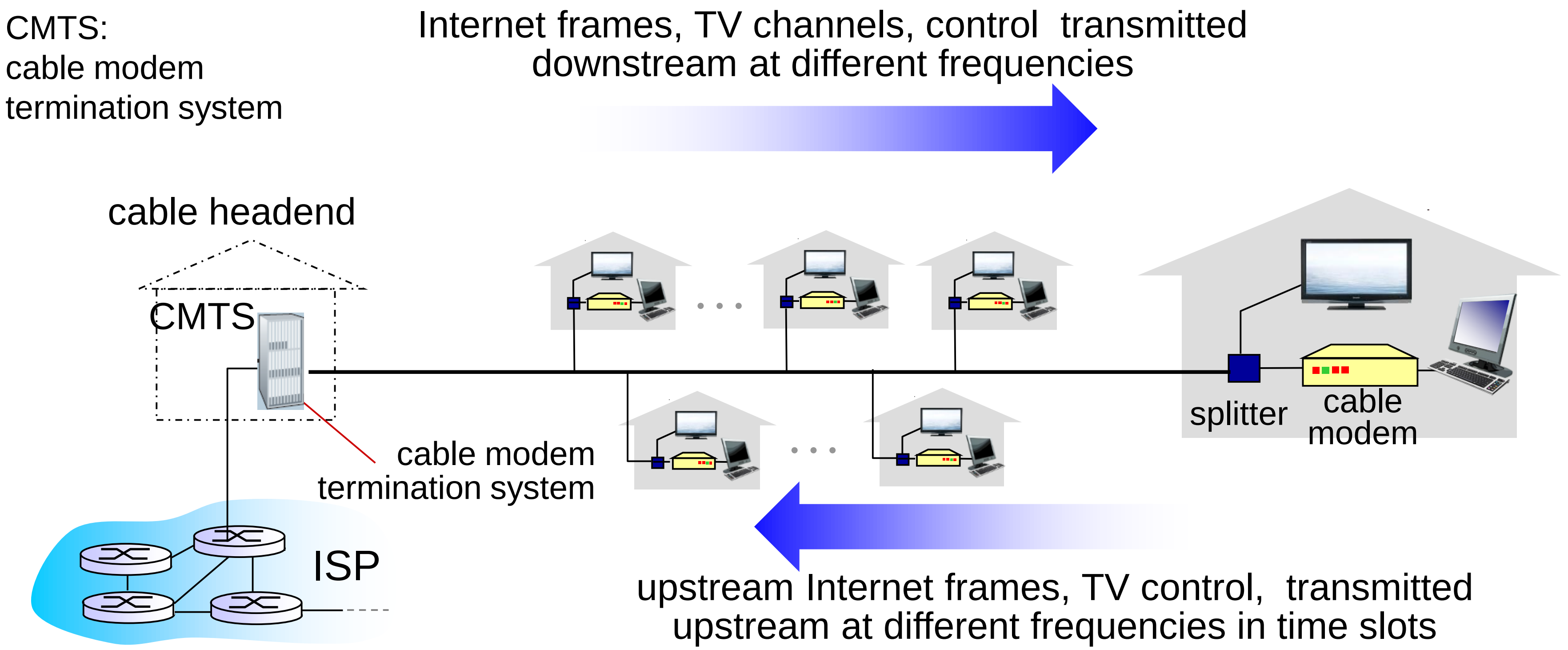
- Stations take turns to transmit
- Only the station with the token can transmit
- Sender receives its own transmission
  - Drains its frame off the ring
- Releases token after transmission/reception
- Deterministic delivery possible
- High throughput under heavy load

Ethernet vs Token Ring

- Non-deterministic
- No delays at low loads
- Low throughput under heavy load
- No priorities
- No management overhead
- Large minimum size

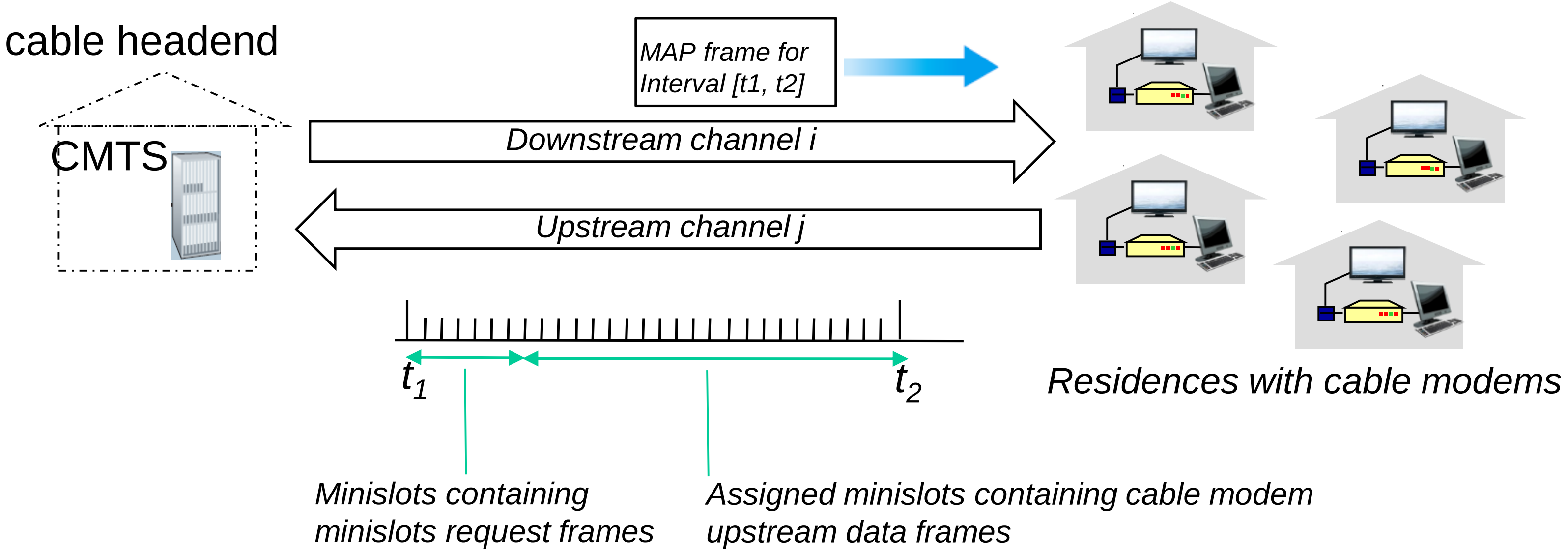
- Deterministic
- Substantial delays at low loads
- High throughput under heavy load
- Multiple priorities
- Complex management
- Small frames possible

Cable Access Network



- multiple 40Mbps downstream (broadcast) channels (each: 6MHz)
  - single CMTS transmits into channels
- multiple 30 Mbps upstream channels (each: 6.4MHz)
  - 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 (“content slots”)

Summary of MAC Protocols

- Why media access control?
  - Shared media: only one user can send at a time
  - Media access control: determine who has access
- MAC issues:
  - distributed, using the same channel for regulating access
- What do you do with a shared media?
  - Channel Partitioning, by time, frequency or code
    - Time Division, Code 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 WiFi/802.11
  - Taking Turns
    - polling from a central site, token passing (Bluetooth, Token Ring, FDDI)