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

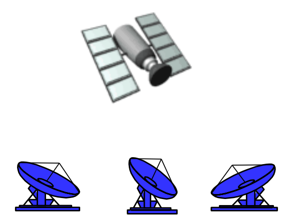
Multiple access links, protocols

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., 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 broadcast channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!
  - no out-of-band channel for coordination

analogy to human protocols

An ideal multiple access protocol

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
    - no synchronization of clocks, slots
4. Simple

Ideal performance:

1. No idle time (when there is traffic waiting)
2. No wasted time (collisions)
3. No access delay

MAC protocols: taxonomy

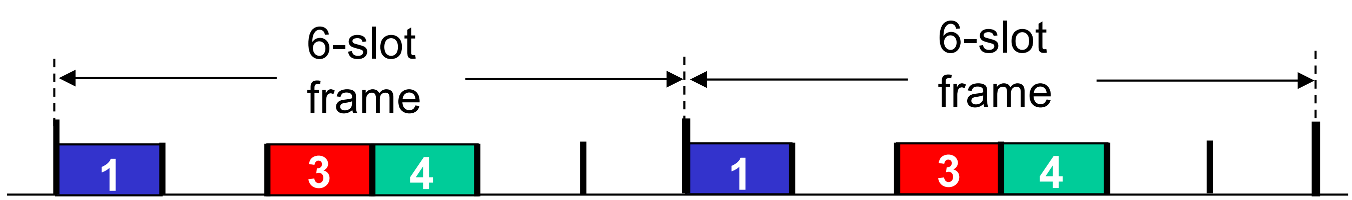
three broad classes:

- *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

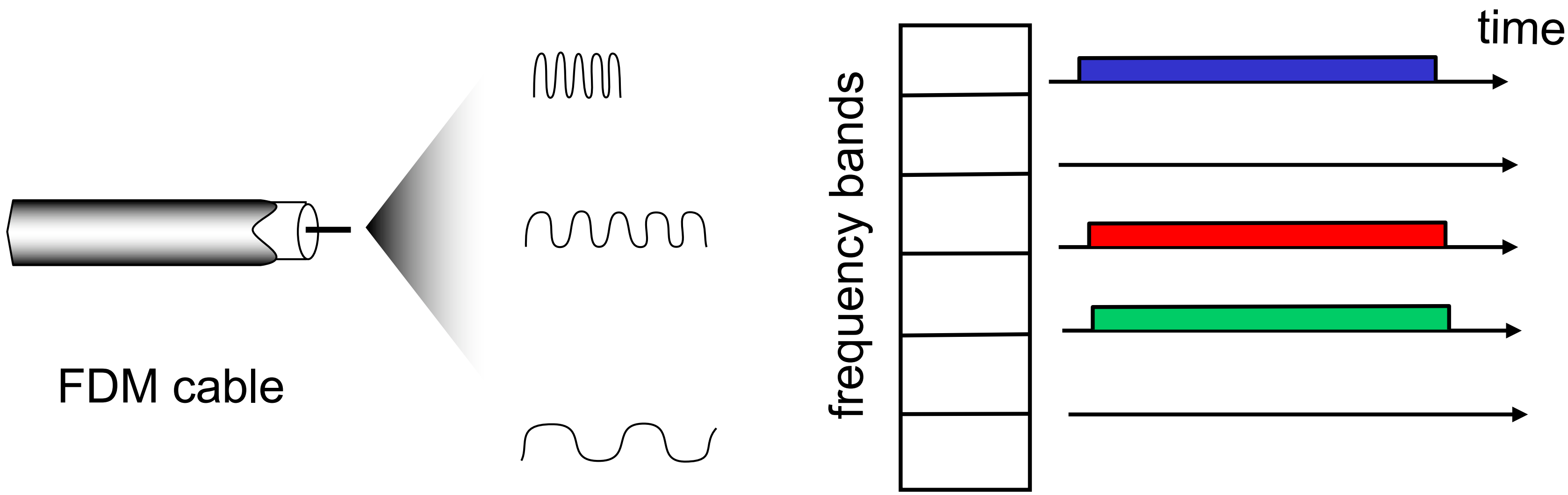
- access to channel in "rounds"
- each station gets fixed length slot (length = packet transmission time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have packets to send, slots 2,5,6 idle



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



Channel Partitioning MAC protocols: CDMA

CDMA: code division multiple access

- each station is assigned a different code
- each station uses its code to encode data bits
- stations can transmit simultaneously
- orthogonal codes→ stations can transmit simultaneously

- widely used in wireless

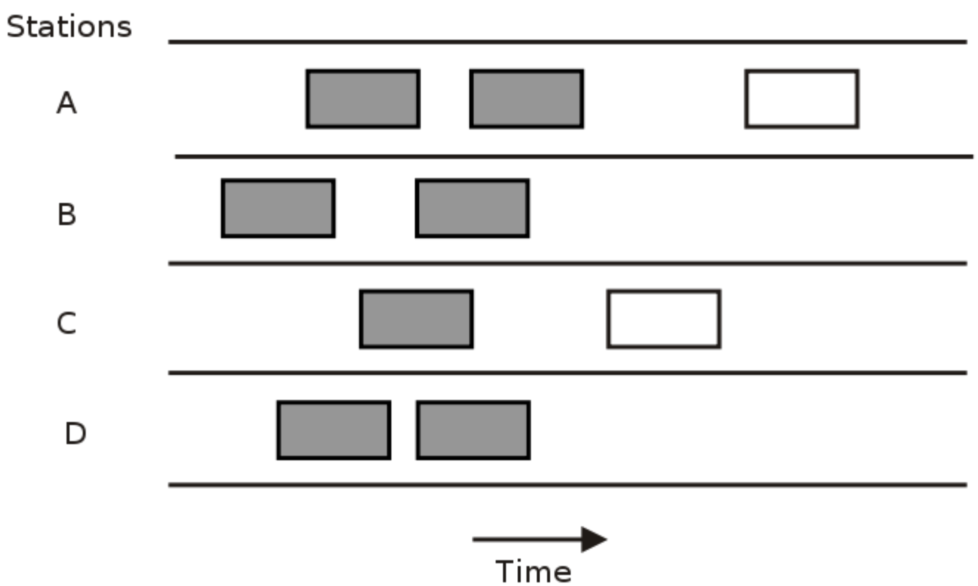
Channel Partitioning

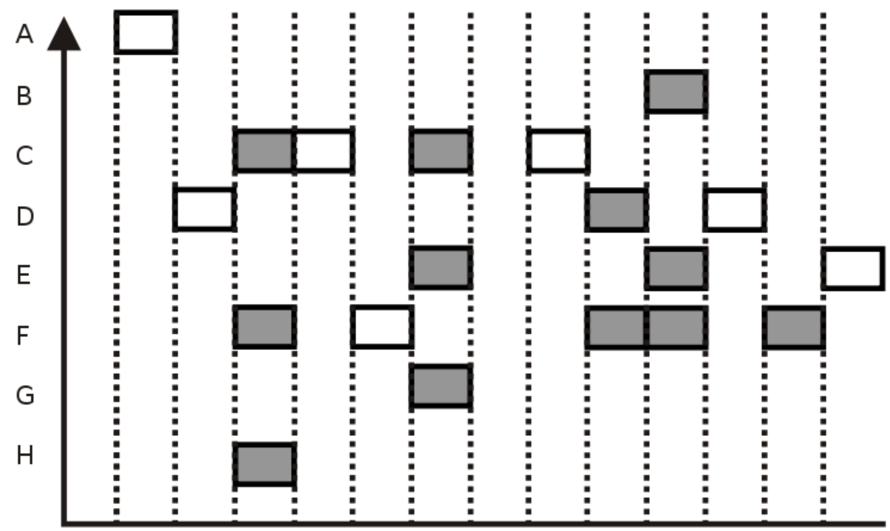
- Pros
  - No collisions
  - Perfectly fair
- Cons
  - Reserved resources (slots, frequencies) can remain idle even if there is traffic waiting
    - Use fraction of the bandwidth
    - Startup delay (for TDMA)
  - Some traffic may get denied access, although there are idle resources
  - A-priori coordination between nodes needed

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, pure ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Aloha





Slotted ALOHA protocol (shaded slots indicate collision)

Slotted ALOHA

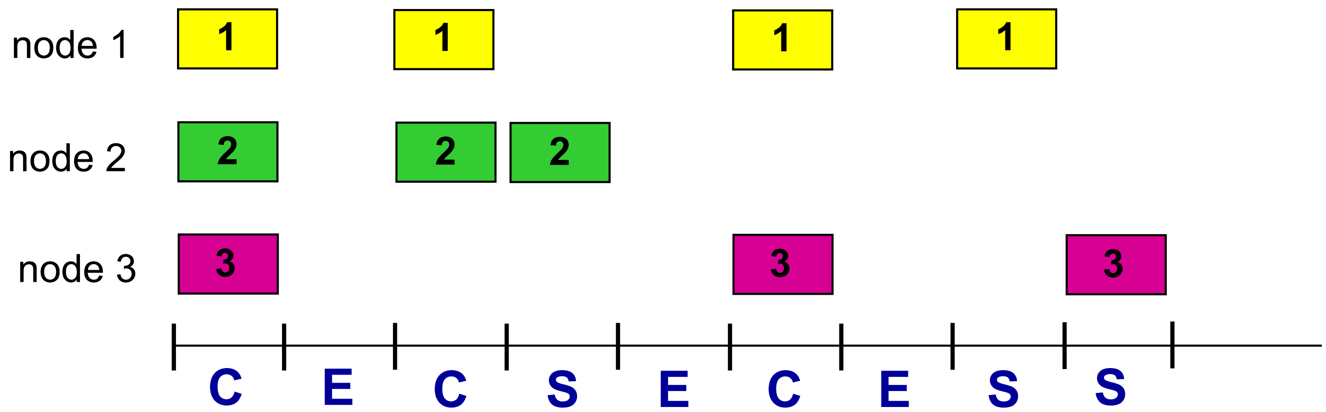
assumptions:

- 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



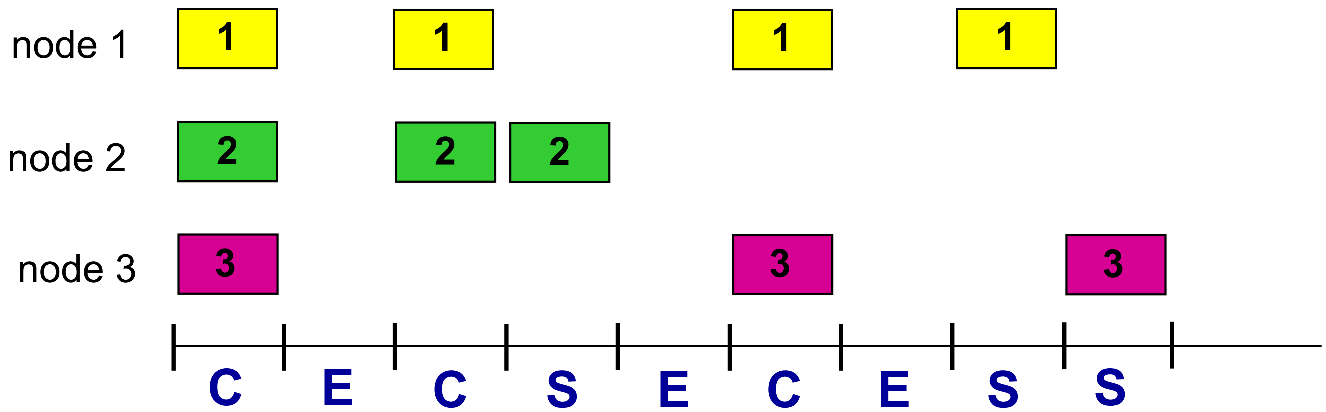
Pros:

- 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 → wasting slots
- clock synchronization
- nodes may be able to detect collision in less than time to transmit packet

Slotted ALOHA: input/output



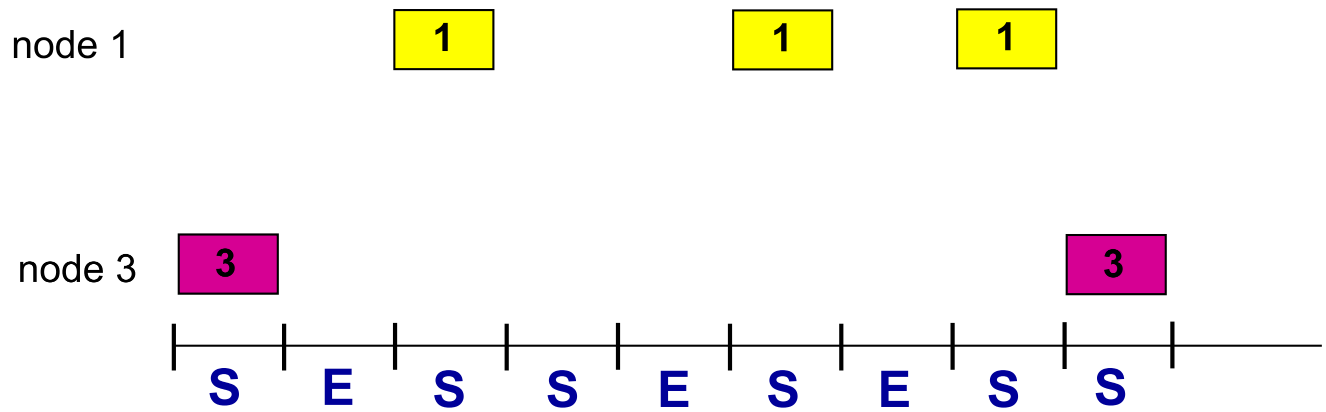
Input:

- *N=1,…:* #nodes with many frames to send
- each node transmits in slot with probability *p in [0,1]*
- *Total offered load: Np*

Output:

- Throughput/efficiency:
  - = long-term fraction of successful slots
  - = prob. for one given slot lead to a successful transmission

Slotted ALOHA: low load regime



Input:

- *Low N:* #nodes with many frames to send

Or

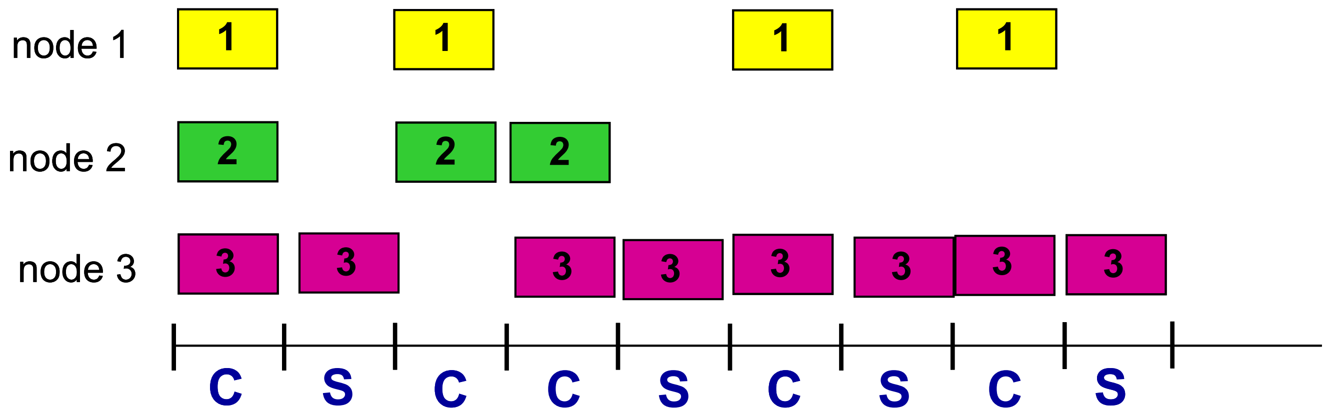
- each node transmits in slot with low prob. p

- *Total offered load: Np*
  - is low

Output:

- Throughput = offered load
  - low
  - No collisions
  - Almost all packets get through
  - But may be many idle slots

Slotted ALOHA: congestion regime



Input:

- *High N:* # nodes with many frames to send

Or

- each node transmits in slot with high prob. *p*
- *Total offered load: Np*
  - is high

Output:

- low throughput
  - More (re)transmissions than slots
  - Many collisions
  - Almost no packets get through

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:

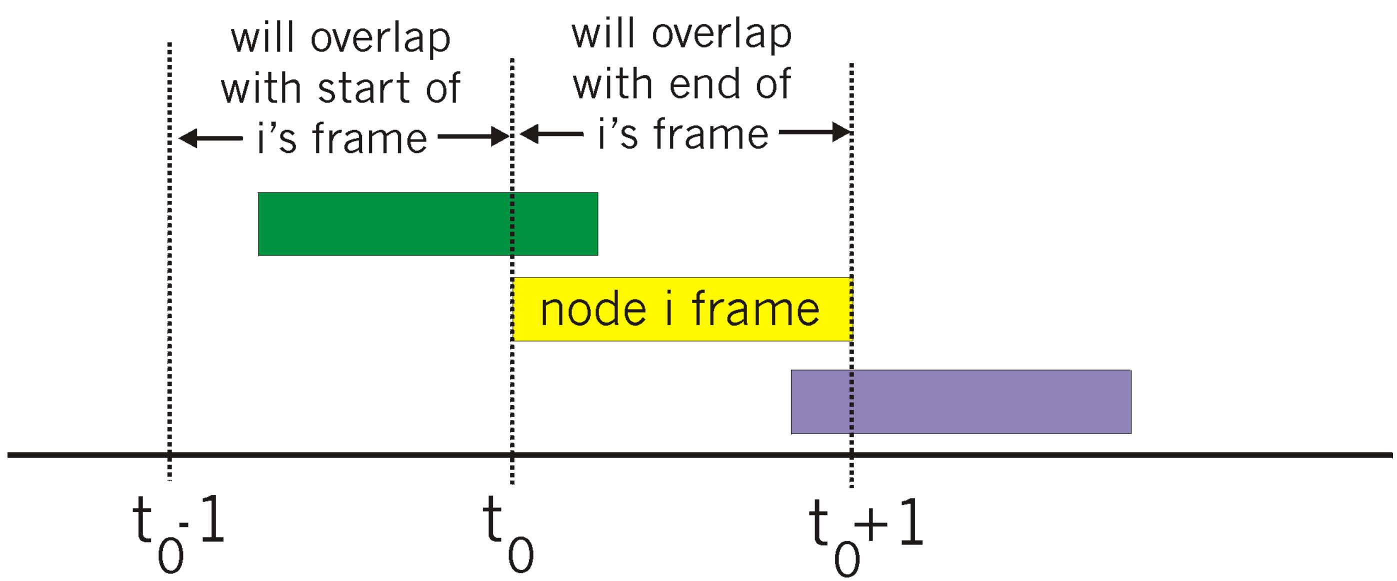
max efficiency = 1/e = .37

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

!

Pure (unslotted) ALOHA

- unslotted Aloha: simpler, no synchronization
- when frame first arrives
  - transmit immediately
- If the message collides, try sending "later"
  - Wait a random time ~= transmit in the next time frame w.p. p
- Q: is collision probability higher than in slotted Aloha?
  - A: 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 $[t_0\text{-}1, t_0]$ $\cdot$

P(no other node transmits in $[t_0\text{-}1, t_0]$

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

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

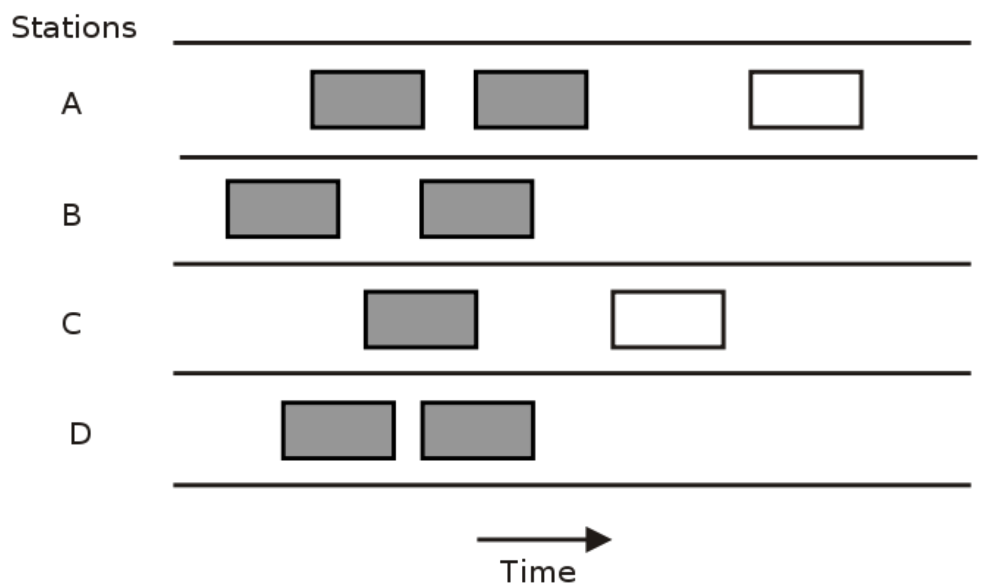
… choosing optimum p and then letting $n \longrightarrow \infty$

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

even *worse* than slotted Aloha!

A different type of analysis for Aloha

- Finite and infinite population analysis
- http://en.wikipedia.org/wiki/ALOHAnet



Aloha and Slotted Aloha efficiency