# Multiple access protocols

- A (distributed) algorithm that determines how nodes share channel

- Communication about channel sharing must use channel itself!
  – No out-of-band channel for coordination

- 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:
     1. No special node to coordinate transmissions
     2. No synchronization of clocks, slots
  4. Simple



# MAC protocols: taxonomy

Three broad classes:

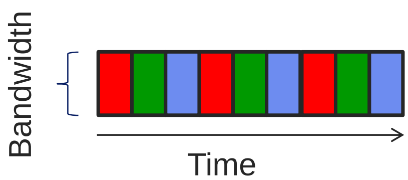
- Channel partitioning – tuning in
  – Divide channel into smaller "pieces" (time slots, frequency, code)
  – Allocate piece to node for exclusive use
- Random access – video chat
  – Channel not divided, allow collisions
  – "Recover" from collisions
- "Taking turns" – in-person conversation
  – Nodes take turns, but nodes with more to send can take longer turns

Broadcast channel networks:

# Channel partitioning MAC protocols

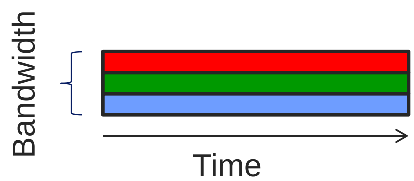
MONTANA STATE UNIVERSITY

## TDMA: Time Division MA




- Access to channel in "rounds"
- Each station assigned fixed length time slot in transmission schedule
- Unused slots go idle

## FDMA: Frequency Division MA




- Channel spectrum divided into frequency bands
- Each station assigned fixed frequency band
- Unused transmission time in frequency bands go idle



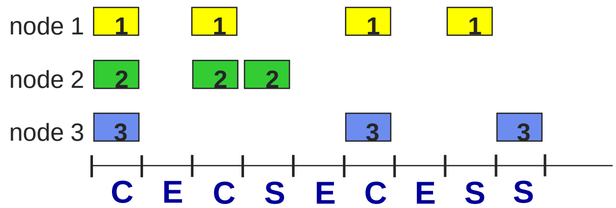
# Random access protocols

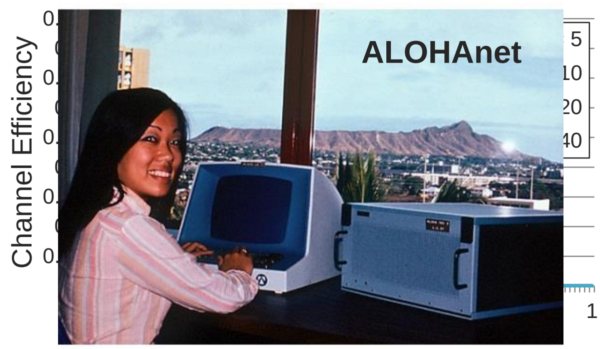
MONTANA STATE UNIVERSITY

- 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
  - nodes are synchronized
- Operation:
  - If 2 or more nodes transmit in slot, all nodes detect collision
  - If collision: node retransmits frame in each subsequent slot with probability $p$ until success
- Efficiency
  - Probability a node transmits is $p$
  - Probability other nodes don't is $(1-p)^{N-1}$
  - Probability of successful transmission $p(1-p)^{N-1}$
  - For $N$ nodes $Np(1-p)^{N-1}$



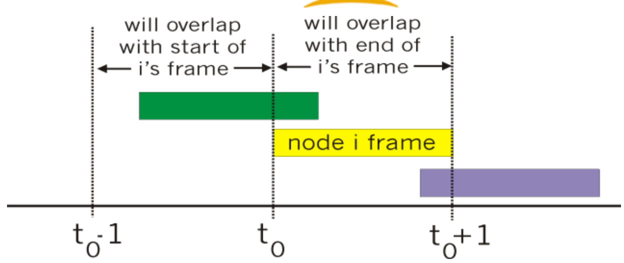



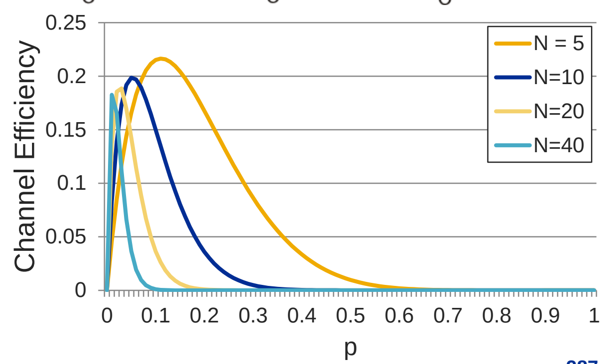

Mountains & Minds



# Pure ALOHA

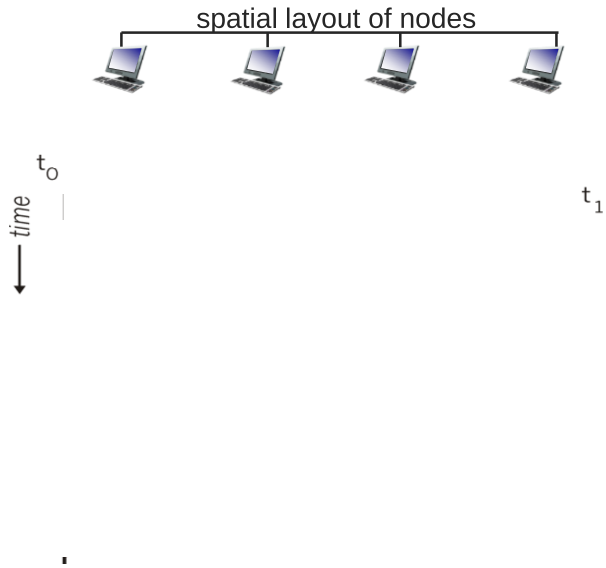
- No synchronization → unslotted
- Frames transmitted immediately
- More or less efficient than slotted ALOHA?
- Efficiency
  - Probability a node transmits is $p$
  - Probability other nodes don't is
    - $(1-p)^{N-1}$ in $[t_0-1, t_0]$
    - $(1-p)^{N-1}$ in $[t_0, t_0+1]$
  - Probability of successful transmission $p(1-p)^{2*(N-1)}$
  - For $N$ nodes $Np(1-p)^{2*(N-1)}$




Mountains & Minds

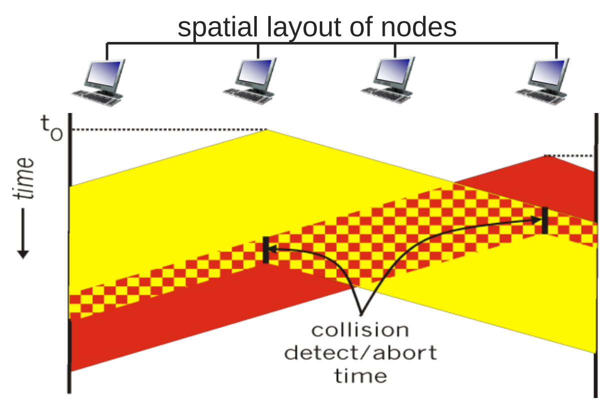
# Carrier Sense Multiple Access






# CSMA/CD (collision detection)

- Human analogy: the polite conversationalist
- 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 (even if full duplex)



# Ethernet CSMA/CD algorithm

MONTANA STATE UNIVERSITY

- NIC receives datagram from network layer, creates frame
- If NIC senses channel idle, starts frame transmission. If NIC senses channel busy, waits until channel idle, then transmits.
- If NIC transmits entire frame without detecting another transmission, NIC is done with frame.
- If NIC detects another transmission while transmitting, aborts and sends jam signal
- 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 $K \times 512$ bit times, returns to Step 2
  - Progressively longer backoff interval after additional collisions

Mountains & Minds

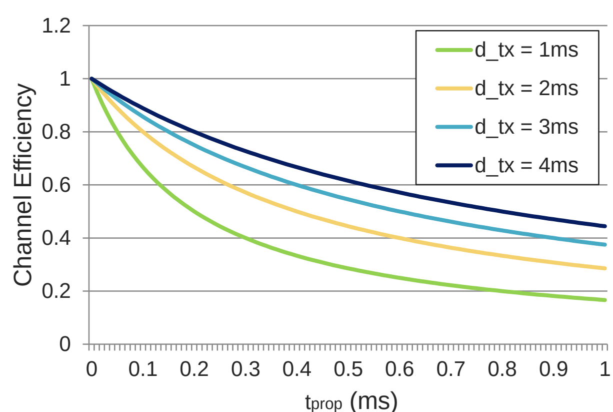
290

# CSMA/CD efficiency

MONTANA STATE UNIVERSITY

- Distance & propagation delay play role in in determining collision probability
- $t_{prop}$ = max prop delay between 2 nodes in LAN
- $t_{tx}$ = time to transmit max-size frame
- Efficiency goes to 1
  - as $t_{prop}$ goes to 0
  - as $t_{tx}$ goes to infinity
- Better performance than ALOHA: and simple, cheap, decentralized!

$$\text{Efficiency} = \frac{1}{1+5*t_{prop}/t_{tx}}$$




Mountains & Minds

291

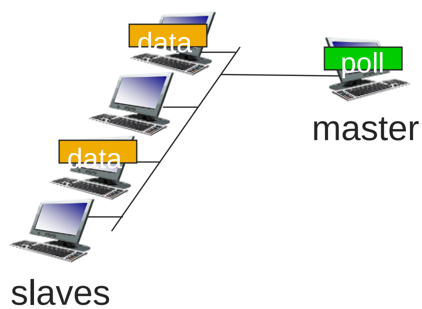
# "Taking turns" MAC protocols

- 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: overhead of collisions
- "Taking turns" protocols
  - look for best of both worlds!



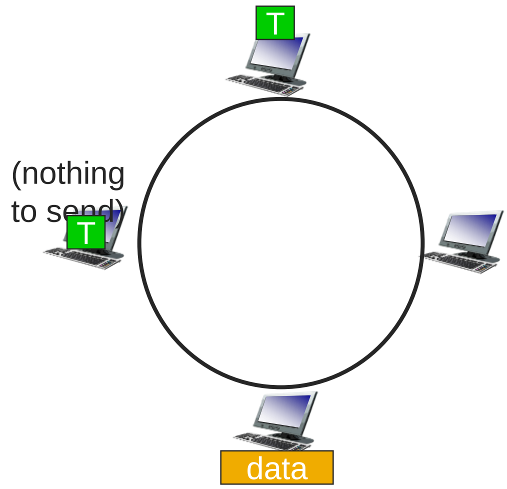
# "Taking turns" MAC protocols

- 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)



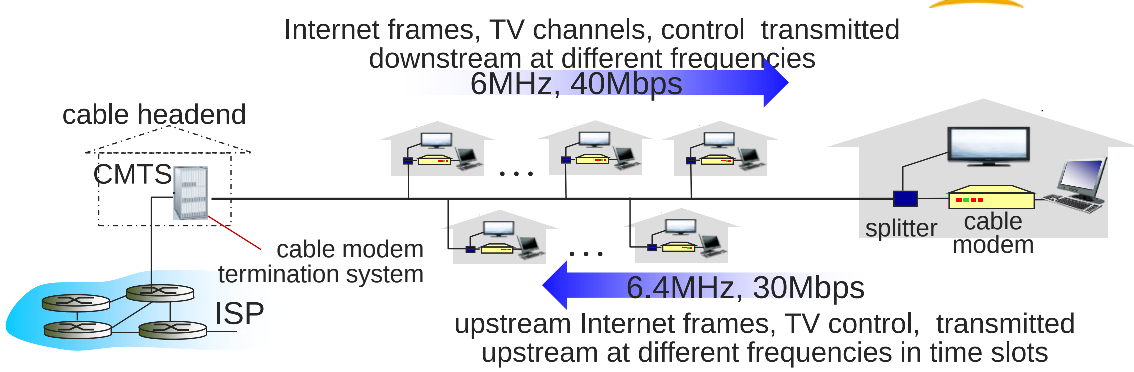
# "Taking turns" MAC protocols

- Token passing:
  - Control token passed from one node to next sequentially.
- Can transmit when holding the token (message)
- Efficient when all nodes have data to send
- Concerns:
  - Token overhead
  - Latency
  - Single point of failure (token)






# Cable access network




- DOCSIS: data over cable service interface spec
  - 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