

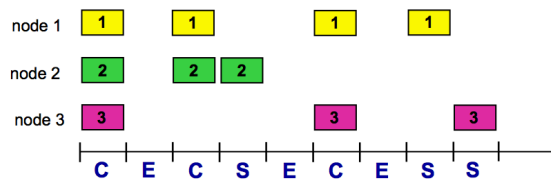
Link Layer (II)

Random access protocols

- when node has packet to send
 - transmit at full channel data rate R
 - no 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
 - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

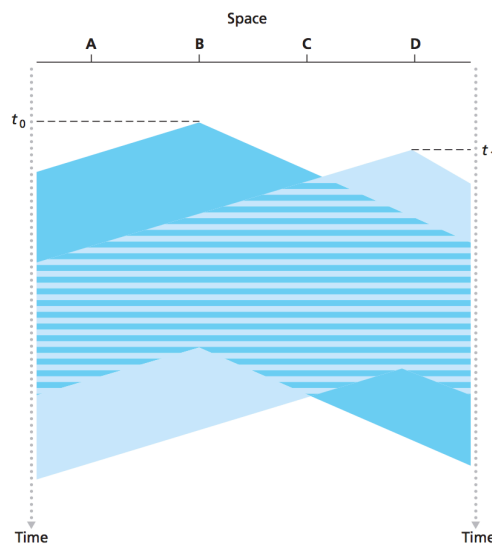
- **Assumption**
 - all packets (frames) same size
 - time divided into equal size slots (time to transmit 1 packet)
 - nodes start to transmit only at the beginning of a slot
 - nodes are synchronized
 - if 2 or more nodes transmit in slot, all nodes detect collision
- **Operation**
 - when node obtains fresh packet, 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



- Advantages
 - single active node can continuously transmit at full rate of channel
 - highly decentralized: only slots in nodes need to be in sync
 - simple
- Disadvantages
 - collisions, wasting slots
 - idle slots
 - clock synchronization

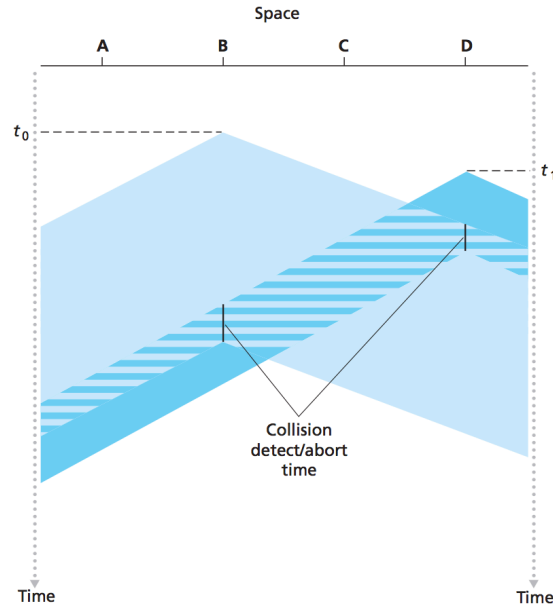
CSMA

- **CSMA: carrier sense multiple access**
- **listen before transmit:**
 - if channel sensed **idle**, **transmit** entire packet
 - if channel sensed **busy**, **defer** transmission
- human analogy: don't interrupt others!
- collisions:
 - **collisions can still occur:** propagation delay means two nodes may not hear each other's transmission
 - when occurs: **entire packet transmission time wasted**
 - distance & propagation delay play role in determining collision probability



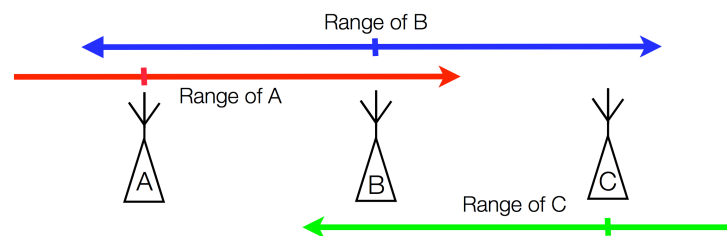
CSMA/CD (collision detection)

- 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 (hidden terminal)
- human analogy: the polite conversationalist



- If senses channel idle, starts frame transmission. If senses channel busy, waits until channel idle, then transmit.
- If detects another transmission while transmitting, aborts and enters exponential backoff stage.
 - longer backoff interval with more collisions.

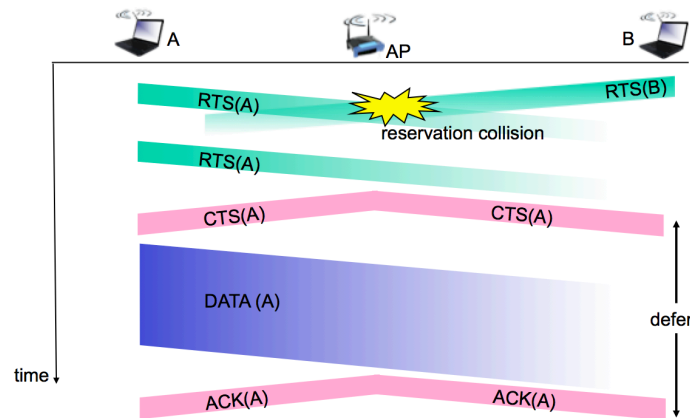
Hidden Terminal Problem



- A & C are in B's communication range. So, B can send messages to both A & C and can sense A and C's transmission.
- A can transmit to B and can sense B's transmission, but cannot detect C.
- C can transmit to B and can sense B's transmission, but cannot detect A.
- So, A doesn't know whether C is transmitting to B, even if using CSMA/CD. So as for C.
- Therefore, the collisions can happen when both A and C transmit to B simultaneously.
- We call this issue Hidden Terminals.

CSMA/CA (collision avoidance)

- avoid data frame collisions completely using small reservation packets



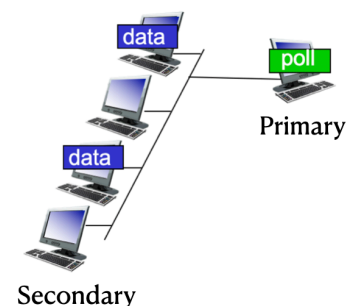
- RTS (request-to-send): sent by the sender
- CTS (clear-to-send): AP broadcasts in response to RTS
- all nodes hear CTS: sender transmits data frame, others defer transmissions

“Taking turns” MAC protocols

- channel partitioning MAC protocols
 - share channel **efficiently and fairly at high load**
 - **inefficient at low load**: delay in channel access, or R/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!

Polling

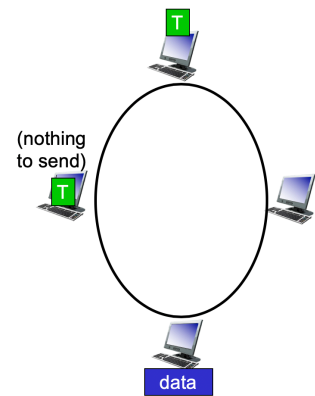
- primary node “invites” secondary nodes to transmit in turn
 - the primary sends a message to secondary1 and lets it know that it can transmit up to some maximum number of packets, after secondary1 transmits some packets, the primary sends a message to secondary2 and lets it know it can transmit up to some maximum number of packets.
- typically used with “dumb” slave devices
- concerns:
 - polling overhead



- latency
 - single point of failure (master)
- examples: 802.15 protocol and Bluetooth protocol

Token passing

- a control token (a small, special-purpose packet) passed from one node to next in some fixed order
- When a node receives a token, it holds onto the token only if it has some packets to transmit; otherwise, it immediately forwards the token to the next node.
- If a node has packets to transmit when it receives the token, it sends up to a maximum number frames and then forwards the token to the next node.
- concerns:
 - token overhead
 - latency
 - single point of failure (some recovery procedure must be invoked to get the token back in circulation)



Disclaimer: Notes adapted from the textbook and online resources.