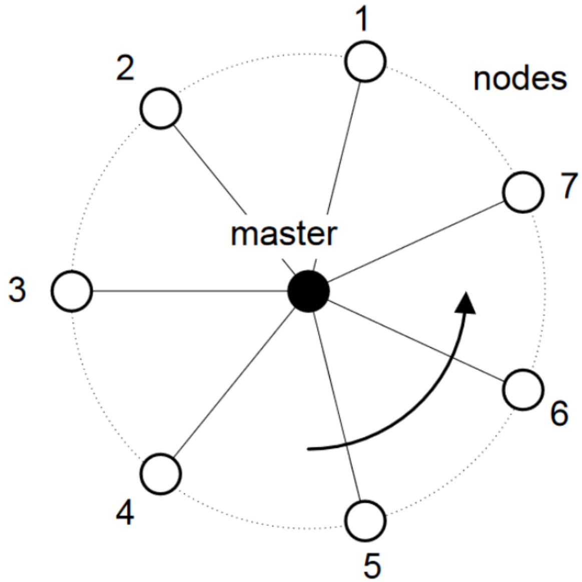
Medium Access with Demand-based Assignment and Random Access

Fixed assignment

- So far, we have assumed that the assignment of available resources is fixed
  - Timeslots assigned to stations in TDMA
  - Frequencies assigned to stations in FDMA
  - …
- Properties:
  - Easy to guarantee fairness
  - Requires some central manager who assigns resources
  - Waste of resources if node does not want to send
- Alternatives:
  - Demand-based assignment: polling, token-passing
  - Random access

Demand-based assignment

- **Polling:**
  - A server asks a node whether it has something to transmit
  - If not, server asks next node
  - Disadvantages:
    - Overhead! Not scalable to large number of nodes
    - Needs central coordination (server)



Source: Misic et al., 2014

Demand-based assignment (2)

- **Token-passing:**
  - Node can only send if it holds the “token”
  - Token = a special signal or packet
  - After transmitting its data, node gives token to next node
  - Example: IBM Token-ring network (wired!)
  - Advantage: no central coordinator needed
  - Challenges:
    - Node needs to know next node
    - Token can get lost: token regeneration needed after timeout
    - Fairness: what if node keeps token too long?

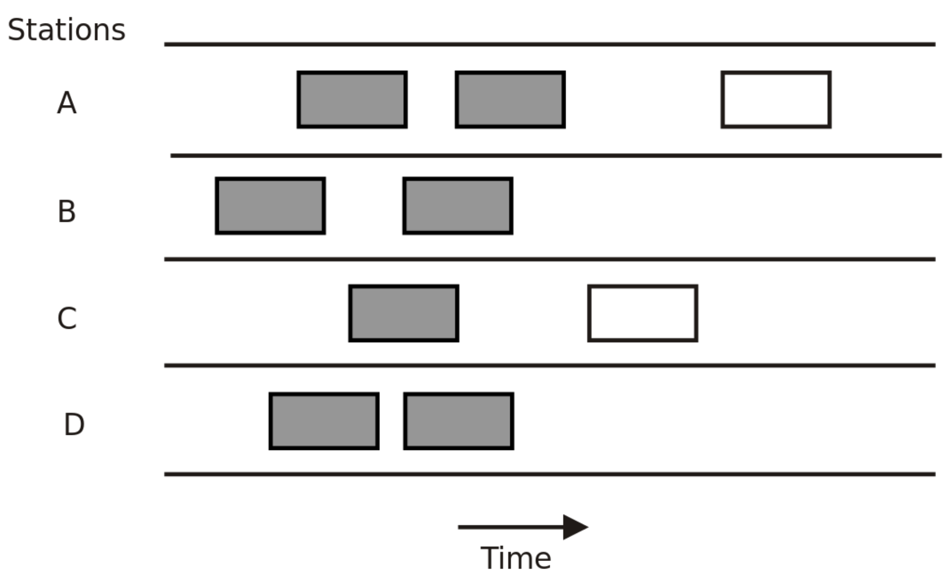
Random Access

- Node sends when it wants
- Collisions with other nodes that want to send at same time
- Example: WiFi !
- Advantages:
  - Fully distributed. No central coordinator/manager required
  - Low latency (if no collision)

- Disadvantage: number of collisions increases with number of nodes and traffic load

Example: ALOHA

- ALOHAnet = first data radio network 1970/71
- Designed to connect Hawaii islands with University of Hawaii on Honolulu
- All nodes send packets/frames of identical duration T on a single channel



Source: wikipedia.org

- Collisions possible if a node sends a packet while another node is still sending

Collision handling in ALOHA (and many other Random Access networks)

1. Sender sends a packet
2. Collision happens. Receiver receives a garbled packet
3. Receiver does not send ACK to sender. In that way, sender knows that the packet has not arrived
4. Sender waits a random time in the range $[0, \ldots, B]$ ("backoff period")
5. Sender tries again ($\rightarrow$ Step 1)

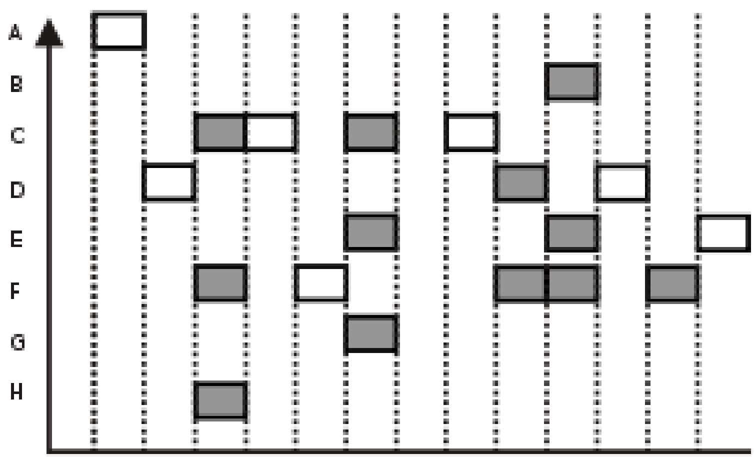
- The Random Backoff Period avoids that two (or more stations) try to send again and again at the same time

Efficiency of ALOHA

- Let's assume that the N stations send packets with an exponential distributed inter-packet time with rate λ
- Number of sending attempts in a time slot is Poisson distributed with rate $\mathrm{G} = \lambda N$
- A packet of duration T sent at time t can collide with packets sent during $]t - T, t + T[$
- Therefore, a sending attempt is only successful if nobody else is sending during a time period of duration $2T$
  - Number of sending attempts during $2T$ = $2G$
  - $P(k\ stations\ want\ to\ send\ in\ two\ time\ slots) = \frac{(2G)^k e^{-2G}}{k!}$
  - $P(nobody\ is\ sending) = P(k = 0) = e^{-2G}$
- That means of G sending attempts per time slot only $G \cdot e^{-2G}$ are successful on average

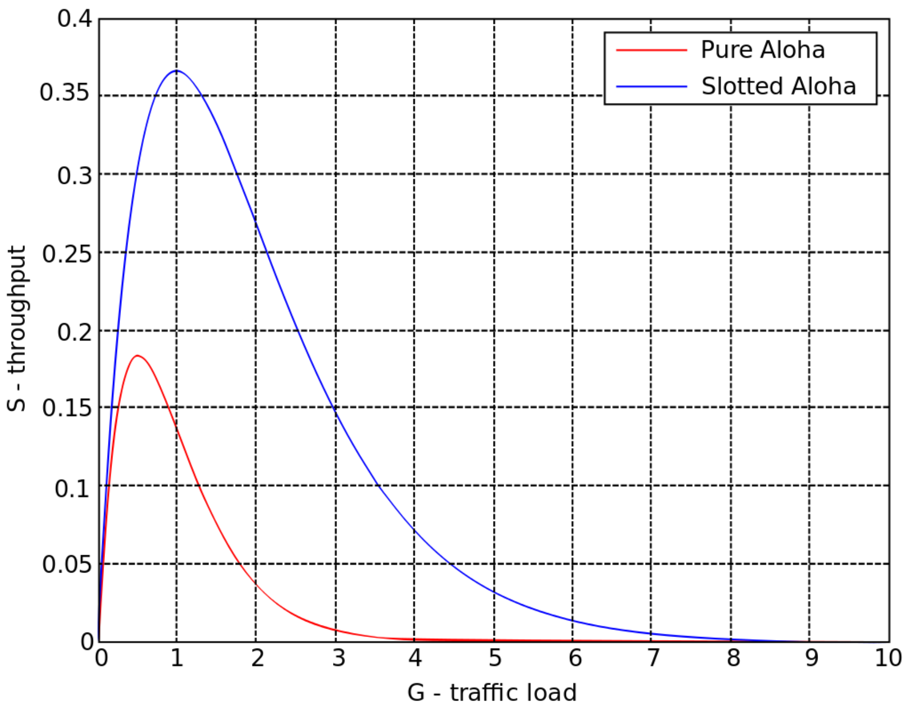
Slotted ALOHA

- Improvement: Slotted ALOHA
  - Use time slots → collision limited to *one* timeslot
  - Of G sending attempts per time slot $G \cdot e^{-G}$ are successful
  - Requires synchronized clock on all nodes



Slotted ALOHA protocol (shaded slots indicate collision)

Source: wikipedia.org



CSMA

- Main limitation of ALOHA: Collisions only detected by receiver
- Improvement: CSMA = *Carrier-Sense Multiple Access*
  - Node listens to channel before sending
  - If a signal ("carrier") from another sending node is detected, node waits until transmission is over (with random backoff)

- However, collisions can still happen. Reason:
  - Propagation delay: signal needs time to reach other nodes

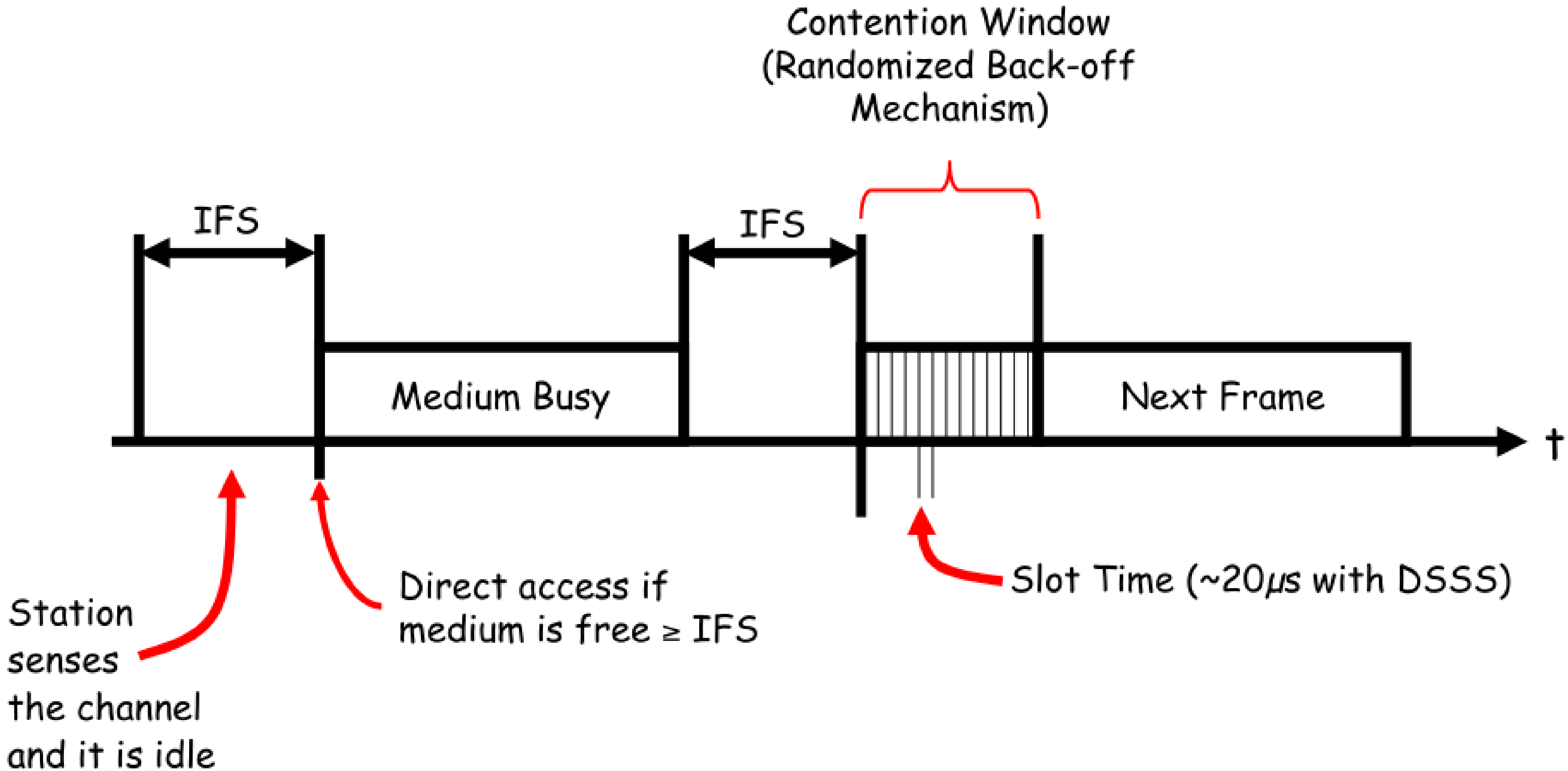
CSMA/CD

- CSMA with *Collision Detection*
- Collision detection:
  - Node continues listening while sending its frame
  - If a collision happens, node immediately stops sending
  - Random backoff time before next attempt
- Advantage: node does not waste time finishing sending its frame when collision happens → better use of channel
- Needs more complex hardware: sending *and* listening at the same time (full duplex capable hardware)
- Used in the original Ethernet with *Binary Exponential* Backoff:
  - Station randomly waits either 0μs or 51.2μs
  - Again collision? Wait randomly either 0 μs, 51.2μs, 102.4μs or 153.6μs
  - Again collision? Wait randomly 0 μs,…,7*51.2μs
  - …

CSMA/CA

- Tries to avoid collisions without using full-duplex capable hardware as needed for CSMA/CD
- Basic procedure:
  a) If channel is free
     1. Node waits for channel to stay free for the IFS time (*Inter-Frame Spacing* time)
     2. Node sends
  b) If channel is in use
     1. Node waits until current transmission is over
     2. Node waits for channel to stay free for the IFS time
     3. Node waits additional random backoff time

CSMA/CA



Source: Freie Universität Berlin

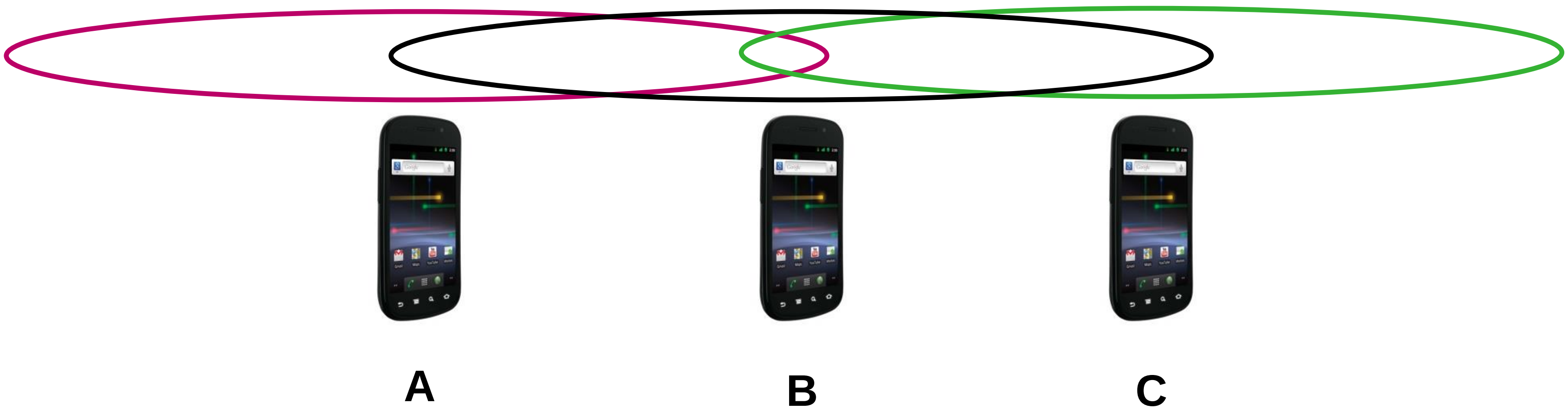
CSMA/CA with ACK

- Some CSMA/CA implementations support ACK control frames
- ACK frames are given higher priority than normal data frames
  - Receiver only waits Shorter IFS (SIFS) < IFS before sending

- Similarly, priorities between stations and for other control frame can be implemented

CSMA for wireless networks

- CSMA/CA and CSMA/CD are unreliable in a wireless network because of the Hidden Station problem
  - Node A and C want to send data to B
  - Node A cannot correctly sense node C and vice versa



CSMA/CA with RTS/CTS

- Protocol
  1. Before sending data, node A sends RTS (Request to Send) control frame to node B after medium was free for IFS time
  2. Node B sends CTS (Clear to Send) to node A after medium was free for SIFS time
  3. Node C also hears the CTS frame -> node C waits
  4. Node A can send now

- How long should node C wait?

  → Size (or duration) of data frame is included in RTS and CTS frames

CSMA/CA with RTS/CTS

- Note:
  - Collisions still possible: RTS packets can collide
  - But: Collisions of control frames (RTS, CTS, ACK) not as bad as collisions of data frames because control frames are typically much smaller

- Used in 802.11 (WiFi)

Other MAC protocols

- Many other MAC protocols exist

  μ-MAC, AI-LMAC, B-MAC, Bit, BMA, CC-MAC, CMAC, Crankshaft, CSMA-MPS, CSMA/ARC, DMAC, E2-MAC, EMACs, f-MAC, FLAMA, Funneling-MAC, G-MAC, HMAC, LMAC, LPL,MMAC, MR-MAC, nanoMAC, O-MAC, PACT, PEDAMACS, PicoRadio, PMAC, PMAC, Q-MAC, Q-MAC, QMAC, RATE EST, RL-MAC, RMAC, RMAC, S-MAC, S- MAC/AL, SCP-MAC, SEESAW, Sift, SMACS, SS-TDMA, STEM, T-MAC, TA-MAC, TICER, TRAMA, U-MAC, WiseMAC, X-MAC, Z-MAC, …