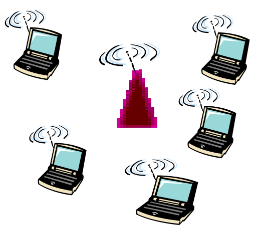
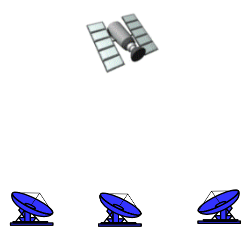
Multiple Access Links and Protocols

Two types of "links":

- point-to-point
  - E.g., point-to-point link between Ethernet switch and host
- broadcast (shared wire or medium)
  - E.g., 802.11 wireless LAN
  - (but also old-fashioned Ethernet)

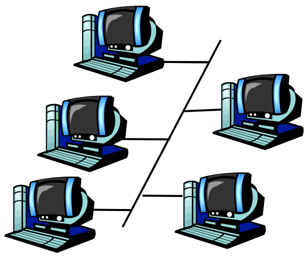


shared RF
(e.g., 802.11 WiFi)



shared RF
(satellite)

humans at a
cocktail party
(shared air, acoustical)

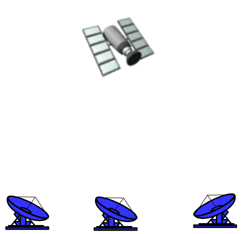


shared wire (e.g.,
cabled Ethernet)

Multiple Access protocols

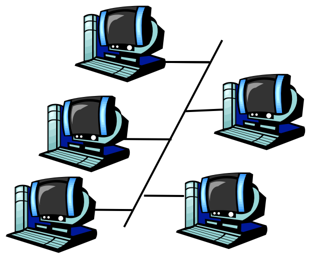
- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference
  - collision if node receives two or more signals at the same time

shared RF
(e.g., 802.11 WiFi)



shared RF
(satellite)

humans at a
cocktail party
(shared air, acoustical)



shared wire (e.g.,
cabled Ethernet)

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

Ideal Multiple Access Protocol

Broadcast channel of rate R bps

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

MAC Protocols: a 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

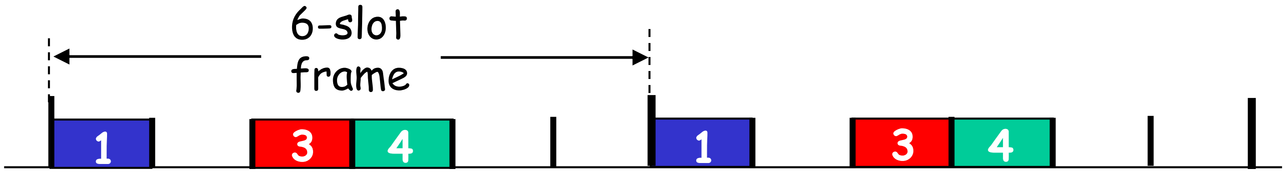
Channel Partitioning MAC protocols: TDMA

TDMA: time division multiple access

- access to channel in "rounds"
- each station gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle

Example 6-station LAN:

- 1,3,4 have pkt, slots 2,5,6 idle



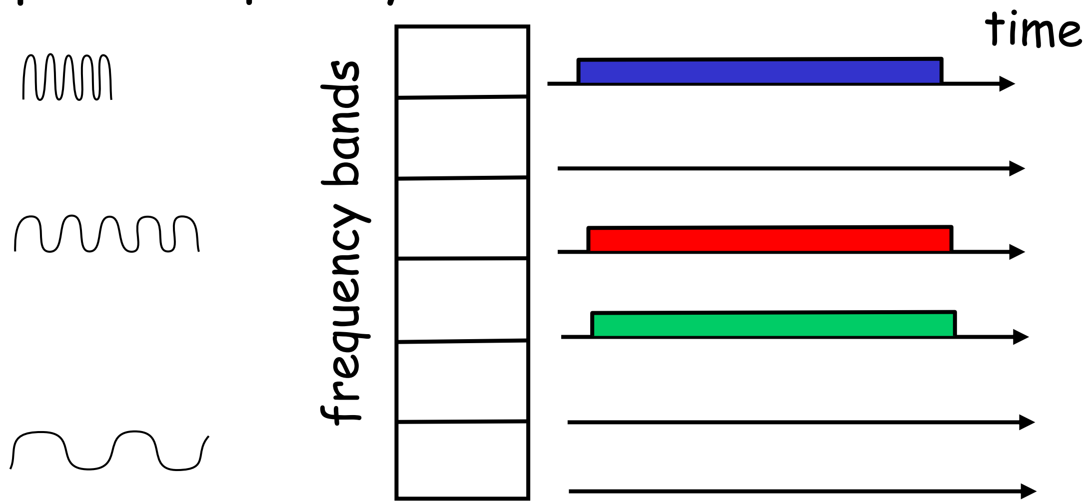
Channel Partitioning MAC protocols: FDMA

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 pkt, frequency bands 2,5,6 idle



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:
  1. how to detect collisions
  2. how to recover from collisions (e.g., via delayed retransmissions)

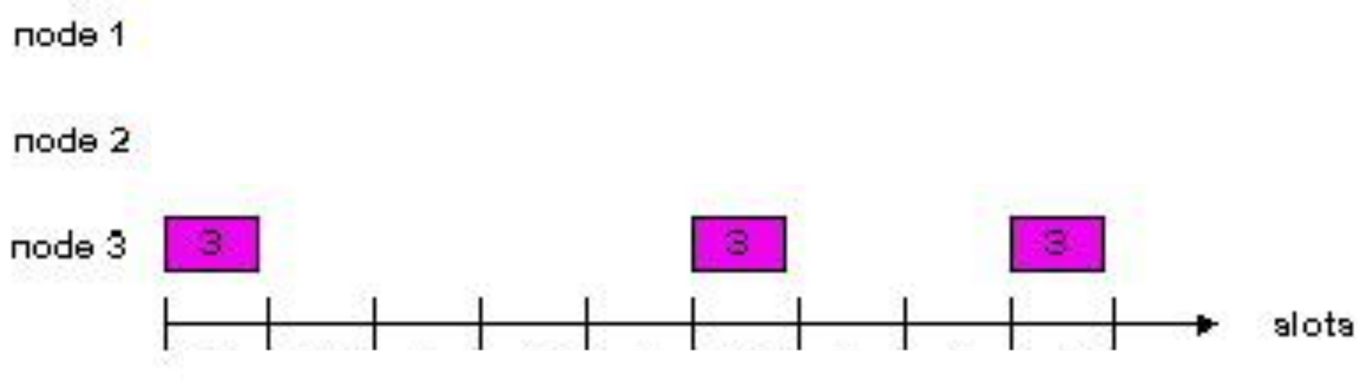
Examples of random access MAC protocols:
- slotted ALOHA
- ALOHA
- CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

Assumptions:

-
-
-
-
-

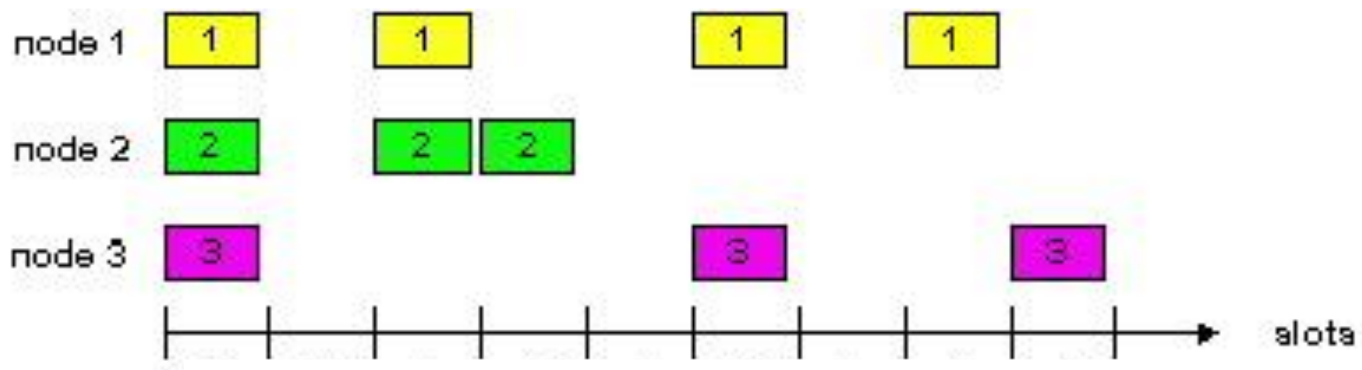
Slotted ALOHA



Assumptions:

- all frames same size
- time divided into equal size slots (time to transmit 1 frame)
- nodes start to transmit only slot beginning
-
-

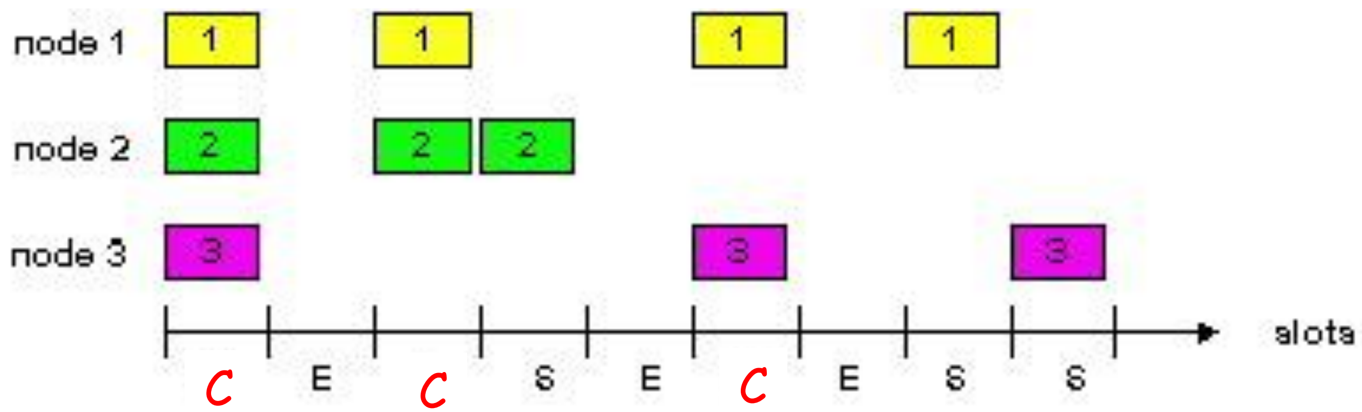
Slotted ALOHA



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

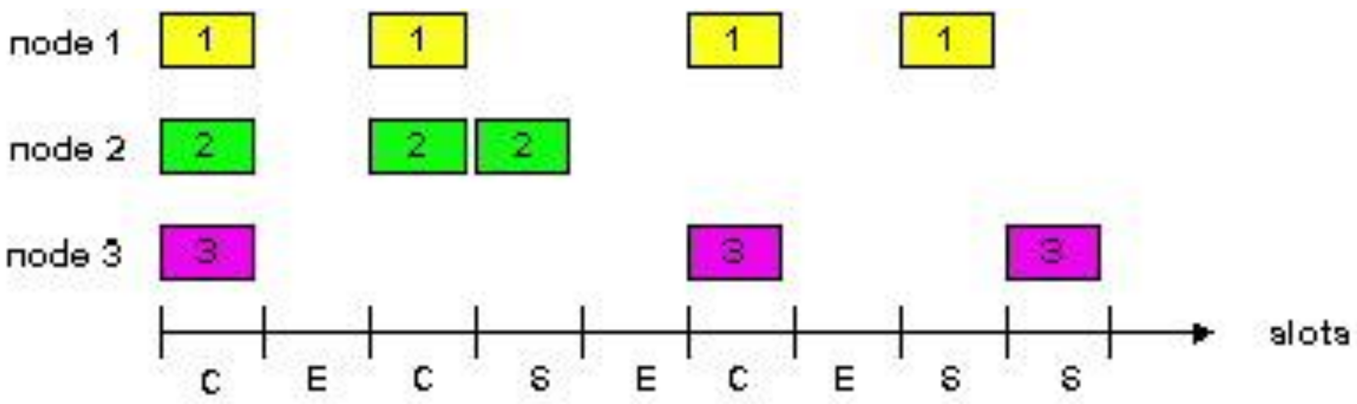
Slotted ALOHA



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

Slotted ALOHA



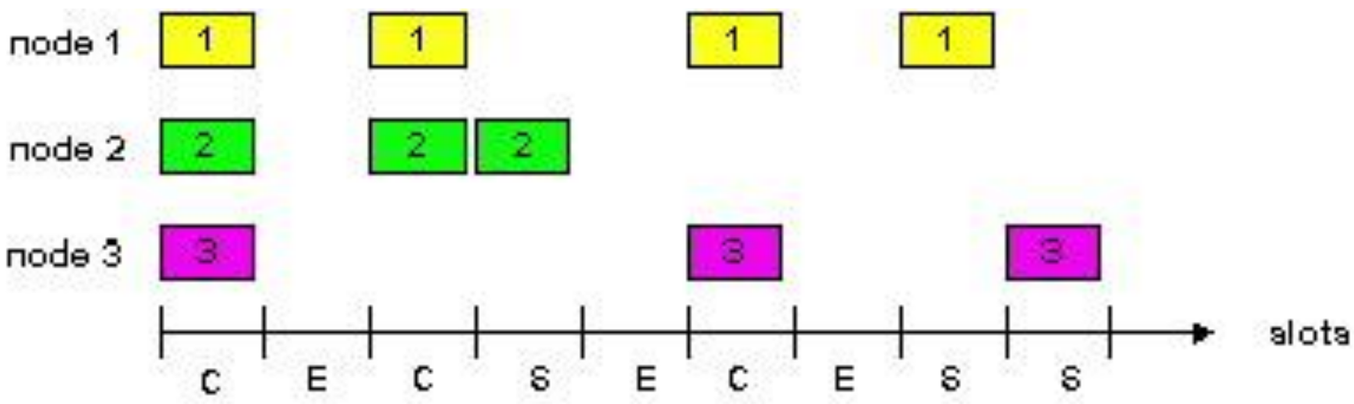
Assumptions:

- 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

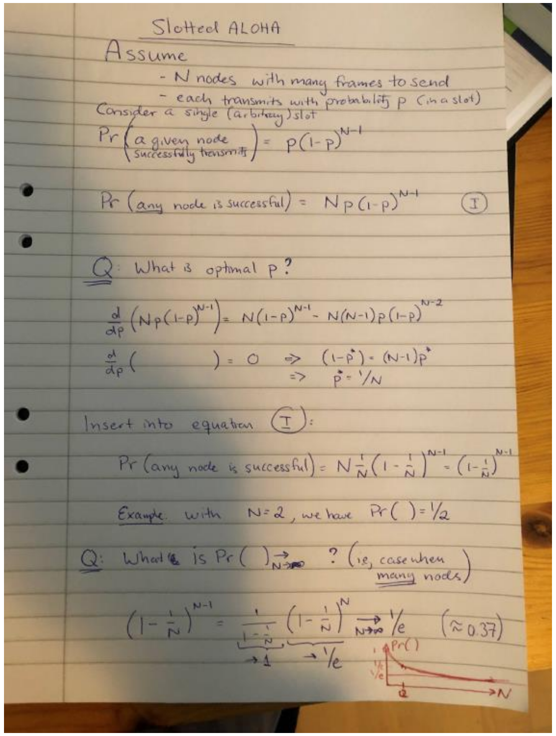
- collisions, wasting slots
- idle slots
- nodes may be able to detect collision in less than time to transmit packet
- clock synchronization

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!

!

Slotted ALOHA

Slotted ALOHA

Assume

- N nodes with many frames to send
- each transmits with probability p (in a slot)

Consider a single (arbitrary) slot

$$\Pr\left(\begin{array}{c}\text{a given node}\\ \text{successfully transmits}\end{array}\right) = p(1-p)^{N-1}$$

$$\Pr(\underline{\text{any}} \text{ node is successful}) = Np(1-p)^{N-1} \quad (\text{I})$$

Q: What is optimal p?

Slotted ALOHA

$$\Pr(\underline{\text{any}} \text{ node is successful}) = Np(1-p)^{N-1} \quad \text{(I)}$$

Q: What is optimal p?

Slotted ALOHA

$$Pr(\underline{\text{any}} \text{ node is successful}) = Np(1-p)^{N-1} \quad \text{(I)}$$

Q: What is optimal p?

$$\frac{d}{dp}\left(Np(1-p)^{N-1}\right) = N(1-p)^{N-1} - N(N-1)p(1-p)^{N-2}$$

$$\frac{d}{dp}\left(\qquad \right) = 0 \Rightarrow (1-p^*) = (N-1)p^*$$

$$\Rightarrow p^* = 1/N$$

Slotted ALOHA

$$\Pr(\underline{\text{any}} \text{ node is successful}) = Np(1-p)^{N-1} \qquad \text{(I)}$$

Q: What is optimal p?

$$\frac{d}{dp}\left(Np(1-p)^{N-1}\right) = N(1-p)^{N-1} - N(N-1)p(1-p)^{N-2}$$

$$\frac{d}{dp}\left(\qquad\right) = 0 \Rightarrow (1-p^*) = (N-1)p^*$$

$$\Rightarrow p^* = 1/N$$

Insert into equation (I):

$$\Pr(\text{any node is successful}) = N\frac{1}{N}\left(1-\frac{1}{N}\right)^{N-1} = \left(1-\frac{1}{N}\right)^{N-1}$$

Slotted ALOHA

$$Pr(\text{any node is successful}) = N \frac{1}{N}\left(1 - \frac{1}{N}\right)^{N-1} = \left(1 - \frac{1}{N}\right)^{N-1}$$

<u>Example:</u> with $N = 2$, we have $Pr(\ ) = 1/2$

Slotted ALOHA

$$\Pr(\text{any node is successful}) = N\frac{1}{N}\left(1-\frac{1}{N}\right)^{N-1} = \left(1-\frac{1}{N}\right)^{N-1}$$

Example: with $N=2$, we have $\Pr(\ ) = 1/2$

Q: What is $\Pr(\ )\underset{N\to\infty}{\longrightarrow}$? (i.e., case when many nods)

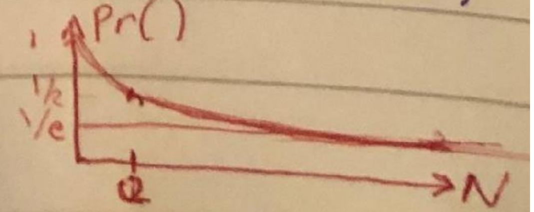
Slotted ALOHA

$$\Pr(\text{any node is successful}) = N\frac{1}{N}\left(1-\frac{1}{N}\right)^{N-1} = \left(1-\frac{1}{N}\right)^{N-1}$$

Example: with $N=2$, we have $\Pr(\ ) = 1/2$

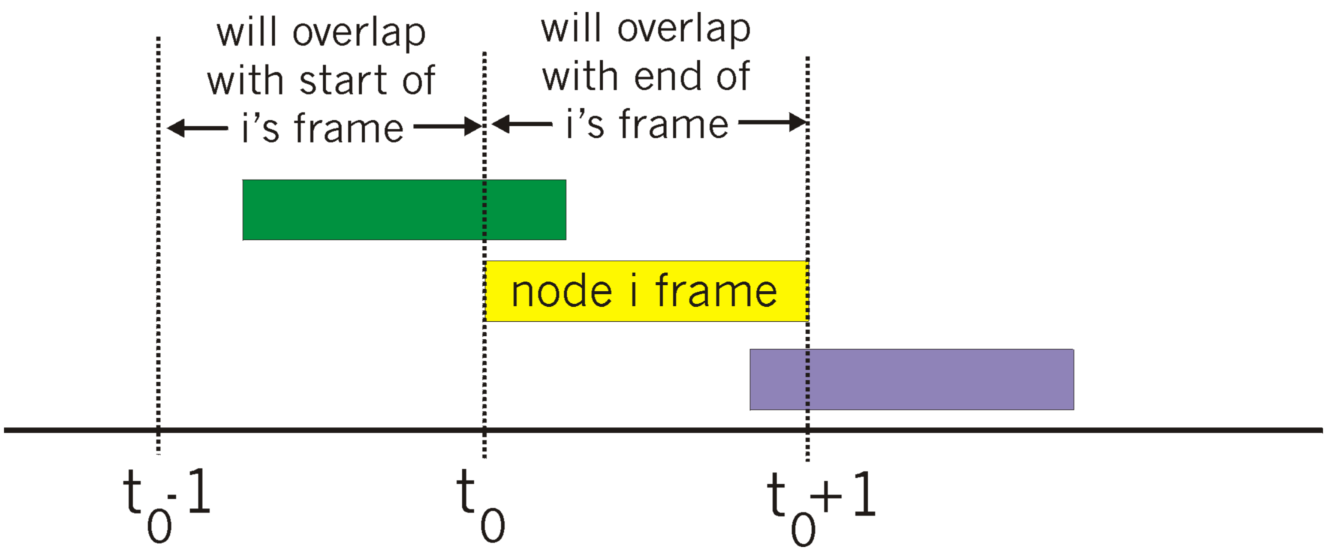
Q: What is $\Pr(\ )\xrightarrow[N\to\infty]{}$? (i.e., case when many nodes)

$$\left(1-\frac{1}{N}\right)^{N-1} = \underbrace{\frac{1}{1-\frac{1}{N}}}_{\to 1}\ \underbrace{\left(1-\frac{1}{N}\right)^{N}}_{\to 1/e} \xrightarrow[N\to\infty]{} 1/e \quad (\approx 0.37)$$



Pure (unslotted) ALOHA

- unslotted Aloha: simpler, no synchronization
- when frame first arrives
  - transmit immediately
- collision probability increases:
  - 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) ·

P(no other node transmits in $[t_0-1,t_0]$ ·

P(no other node transmits in $[t_0,t_0+1]$

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

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

… choosing optimum p* = 1/(2N) and then letting N -> infinity ...

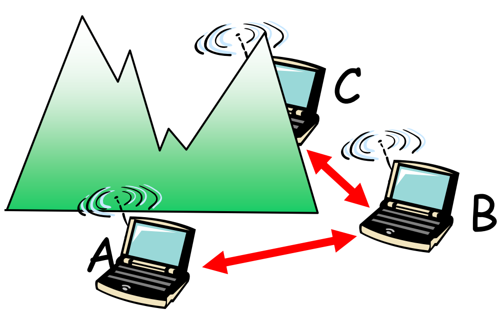
= 1/(2e) = .18

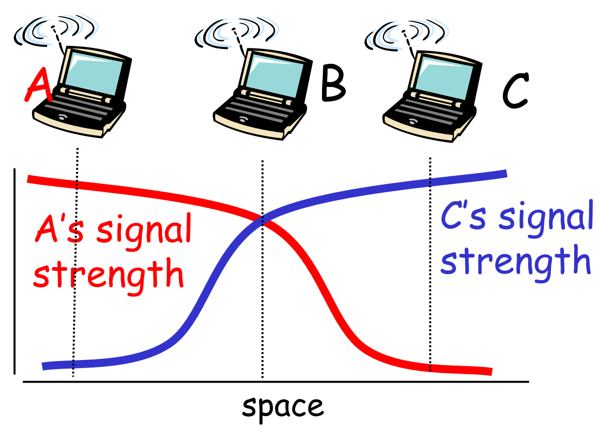
even worse than slotted Aloha!

Now, let's improve things …

IEEE 802.11: multiple access

- avoid collisions: 2⁺ nodes transmitting at same time
- 802.11: CSMA - sense before transmitting
  - don't collide with ongoing transmission by other node
- 802.11: *no collision detection!*
  - difficult to receive (sense collisions) when transmitting due to weak received signals (fading)
  - can't sense all collisions in any case: hidden terminal, fading
  - goal: *avoid collisions:* CSMA/C(ollision)A(voidance)





IEEE 802.11 MAC Protocol: CSMA/CA

802.11 sender

1 if sense channel idle for **DIFS** then
  - transmit entire frame (no CD)

2 if sense channel busy then
  - start random backoff time
  - timer counts down while channel idle
  - transmit when timer expires

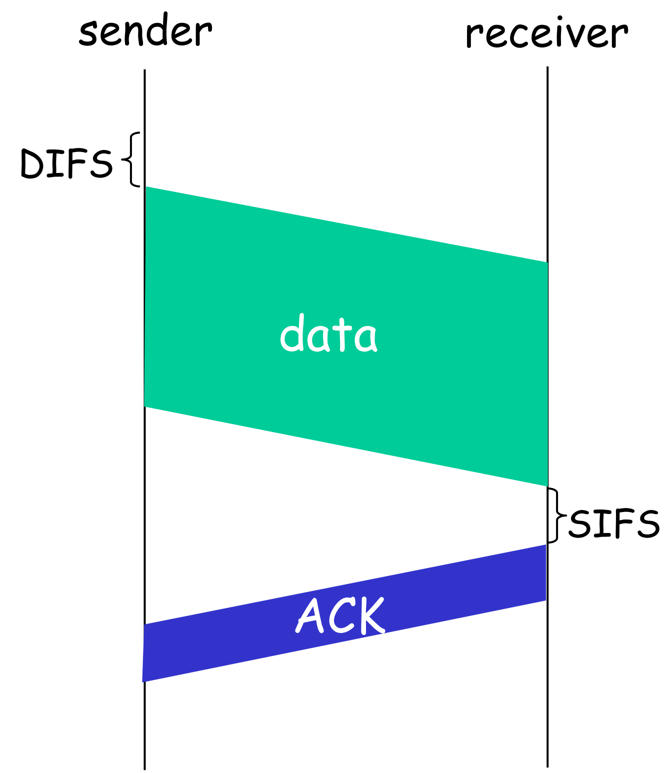
3 if no ACK then increase random backoff interval, repeat step 2

802.11 receiver

- if frame received OK

  return ACK after **SIFS**

  (service model is connectionless, acked)



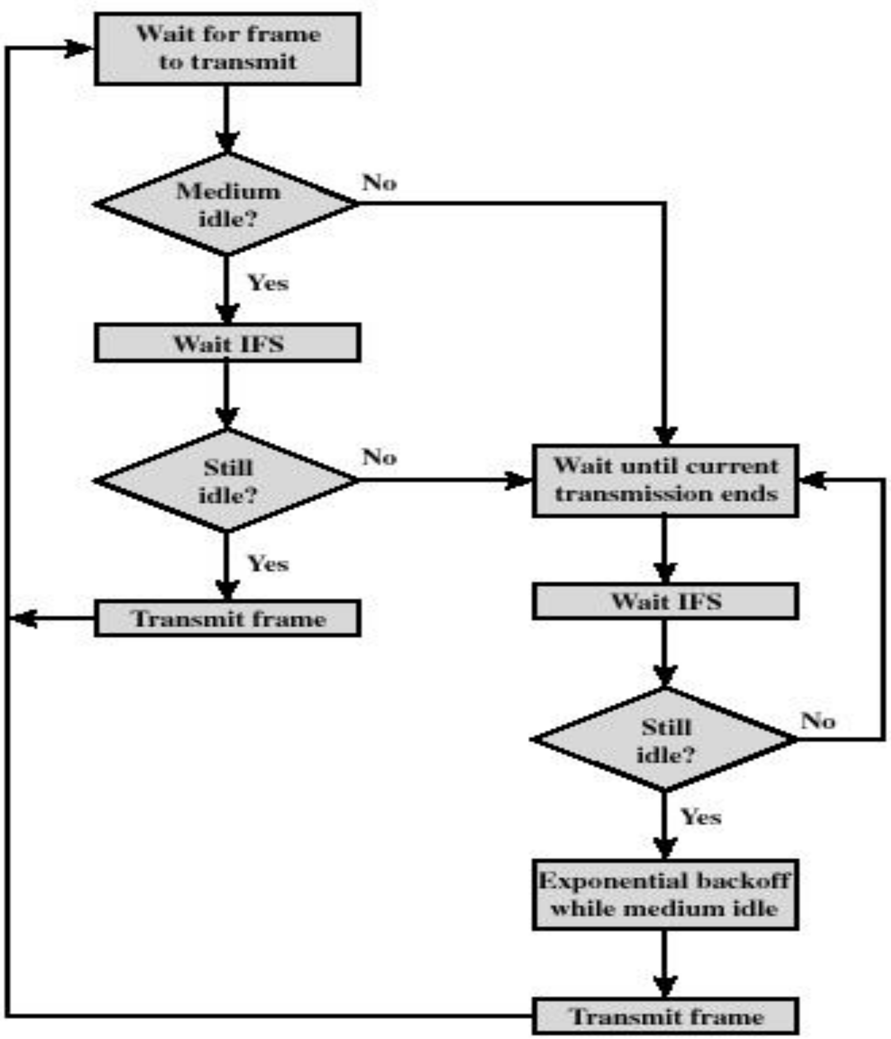
IEEE 802.11 MAC: Distributed Coordination Function (DCF)

- Make use of CSMA (carrier sense multiple access)
- Use set of delays generic called Interframe Space (IFS)

Algorithm Logic:

1. *Station sense the medium*
2. If medium idle, wait IFS, then if still idle transmit frame
3. If medium busy or become busy, defer and monitor the medium until idle
4. Then, delay IFS and sense medium
5. If medium idle, exponential backoff and if then idle, station transmit

- Binary exponential backoff -> handle heavy load



IEEE 802.11 MAC: Distributed Coordination Function (DCF)

- Make use of CSMA (carrier sense multiple access)
- Use set of delays generic called Interframe Space (IFS)

Algorithm Logic:

1*. Station sense the medium*

2. If medium idle, wait IFS, then if still idle transmit frame

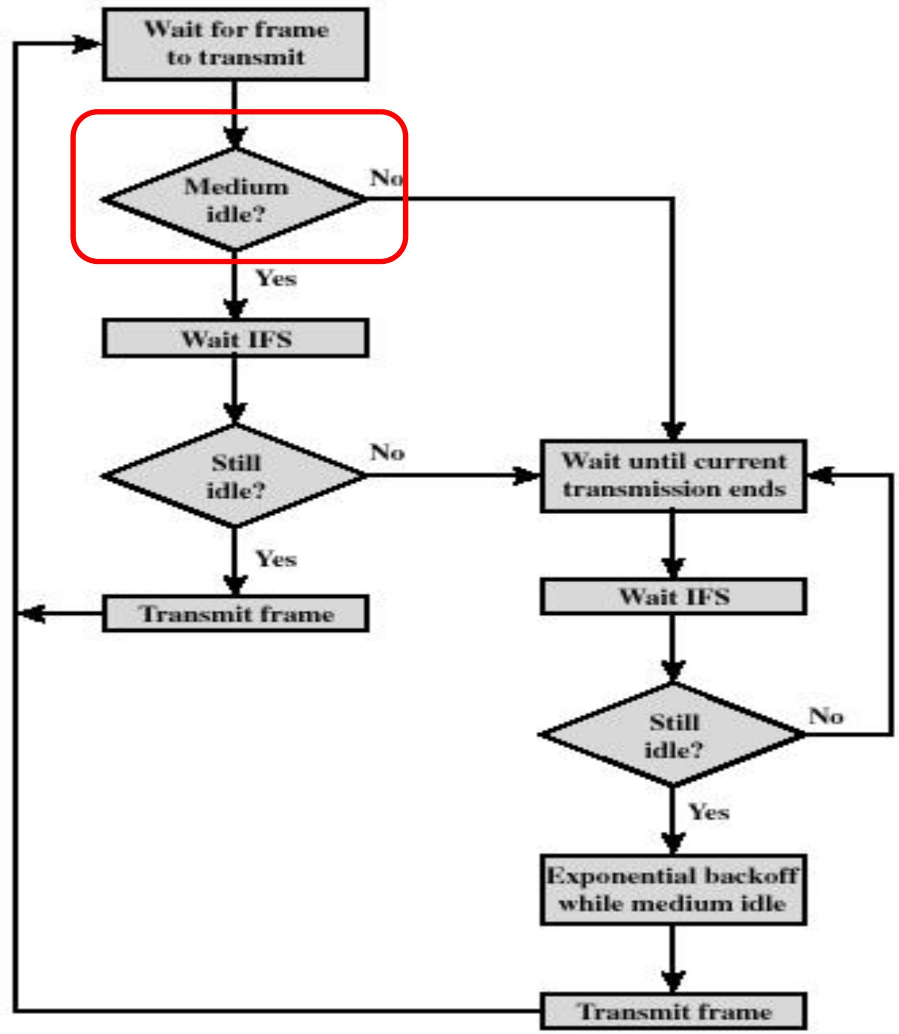
3. If medium busy or become busy, defer and monitor the medium until idle

4. Then, delay IFS and sense medium

5. If medium idle, exponential backoff and if then idle, station transmit

- Binary exponential backoff

-> handle heavy load



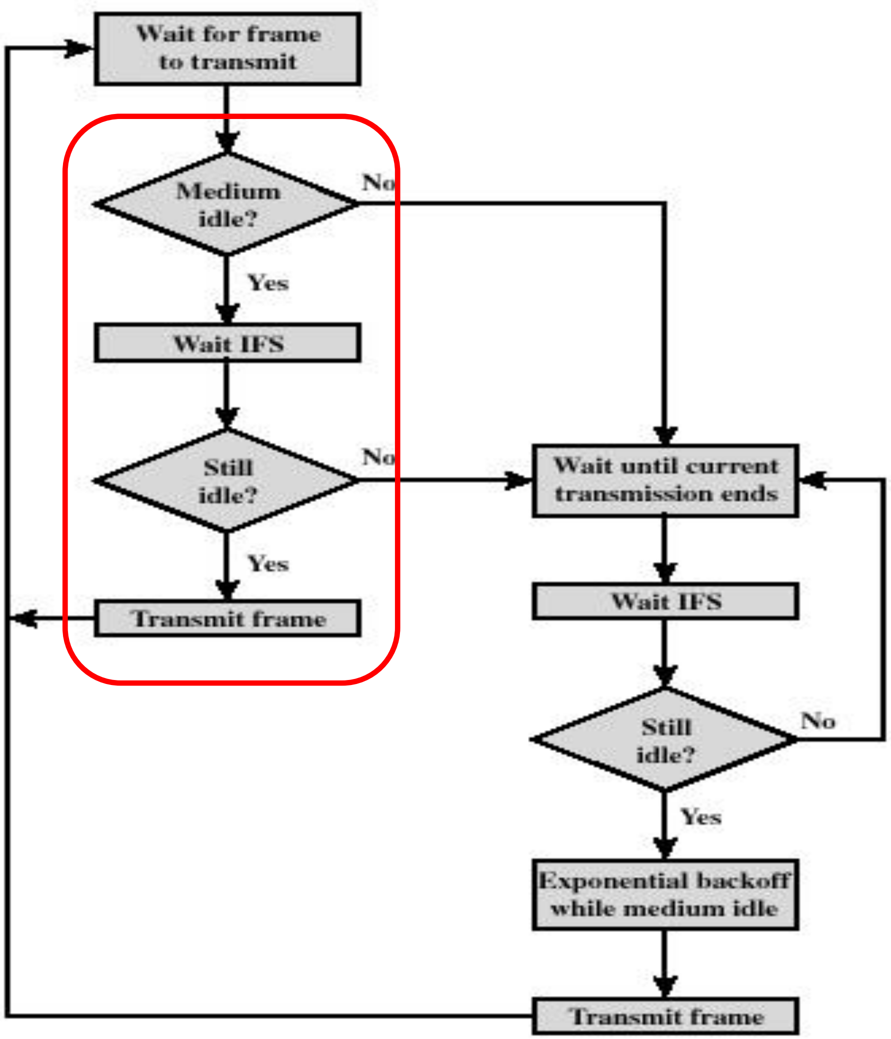
IEEE 802.11 MAC: Distributed Coordination Function (DCF)

- Make use of CSMA (carrier sense multiple access)
- Use set of delays generic called Interframe Space (IFS)

Algorithm Logic:

1. Station sense the medium
2. If medium idle, wait IFS, then if still idle transmit frame
3. If medium busy or become busy, defer and monitor the medium until idle
4. Then, delay IFS and sense medium
5. If medium idle, exponential backoff and if then idle, station transmit

- Binary exponential backoff -> handle heavy load



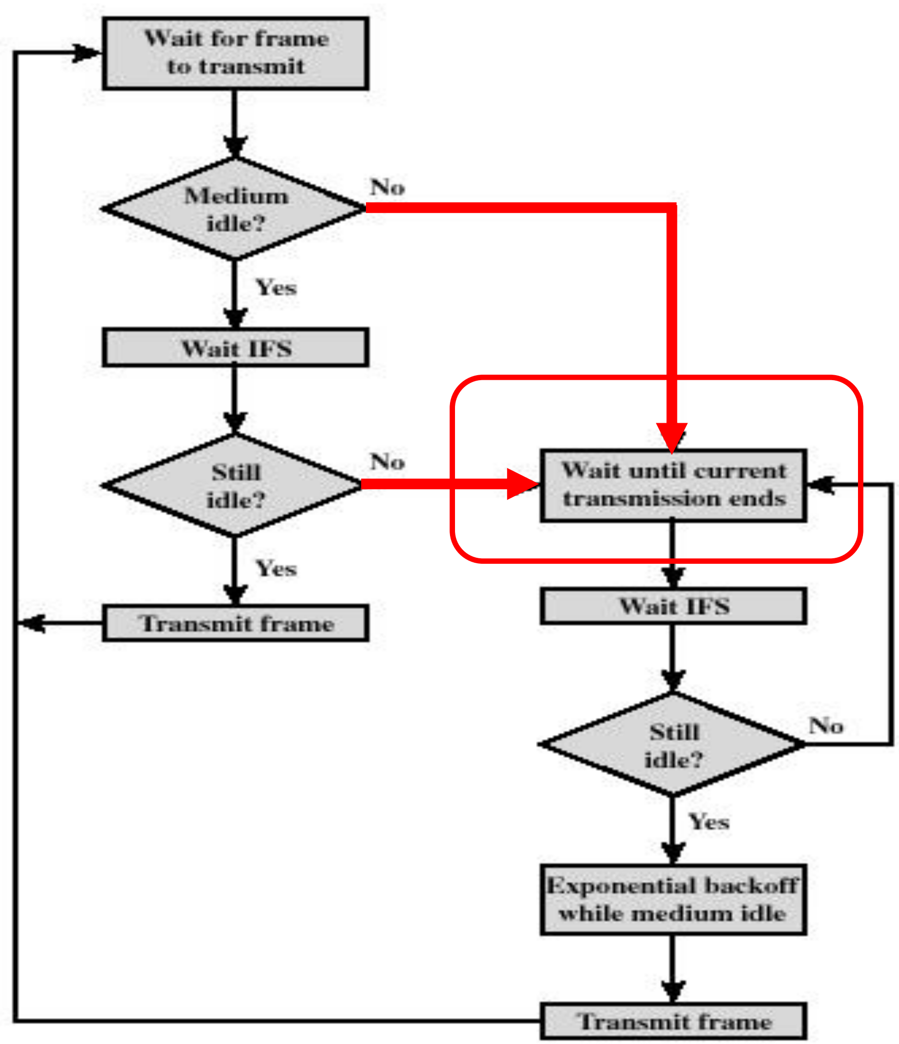
IEEE 802.11 MAC: Distributed Coordination Function (DCF)

- Make use of CSMA (carrier sense multiple access)
- Use set of delays generic called Interframe Space (IFS)

Algorithm Logic:

1. *Station sense the medium*
2. *If medium idle, wait IFS, then if still idle transmit frame*
3. *If medium busy or become busy, defer and monitor the medium until idle*
4. *Then, delay IFS and sense medium*
5. *If medium idle, exponential backoff and if then idle, station transmit*

- Binary exponential backoff -> handle heavy load

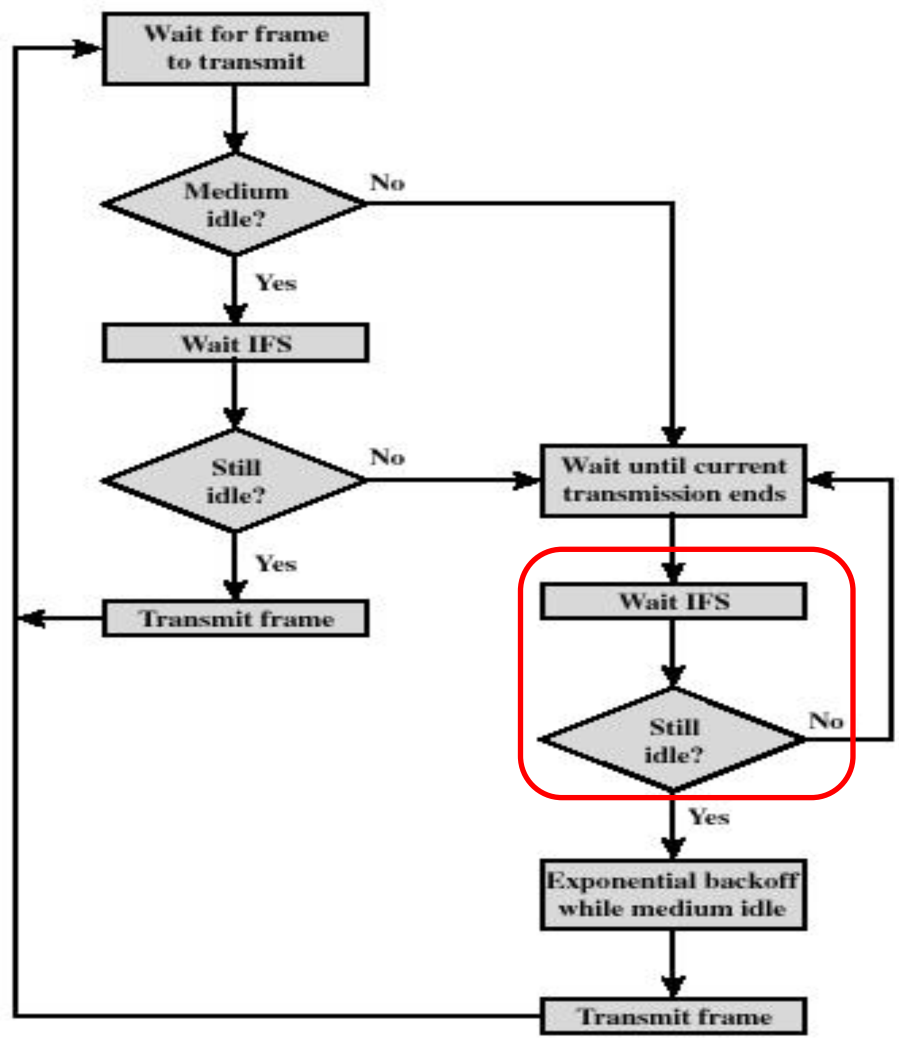


IEEE 802.11 MAC: Distributed Coordination Function (DCF)

- Make use of CSMA (carrier sense multiple access)
- Use set of delays generic called Interframe Space (IFS)

Algorithm Logic:

1. *Station sense the medium*
2. *If medium idle, wait IFS, then if still idle transmit frame*
3. *If medium busy or become busy, defer and monitor the medium until idle*
4. *Then, delay IFS and sense medium*
5. *If medium idle, exponential backoff and if then idle, station transmit*



- Binary exponential backoff -> handle heavy load

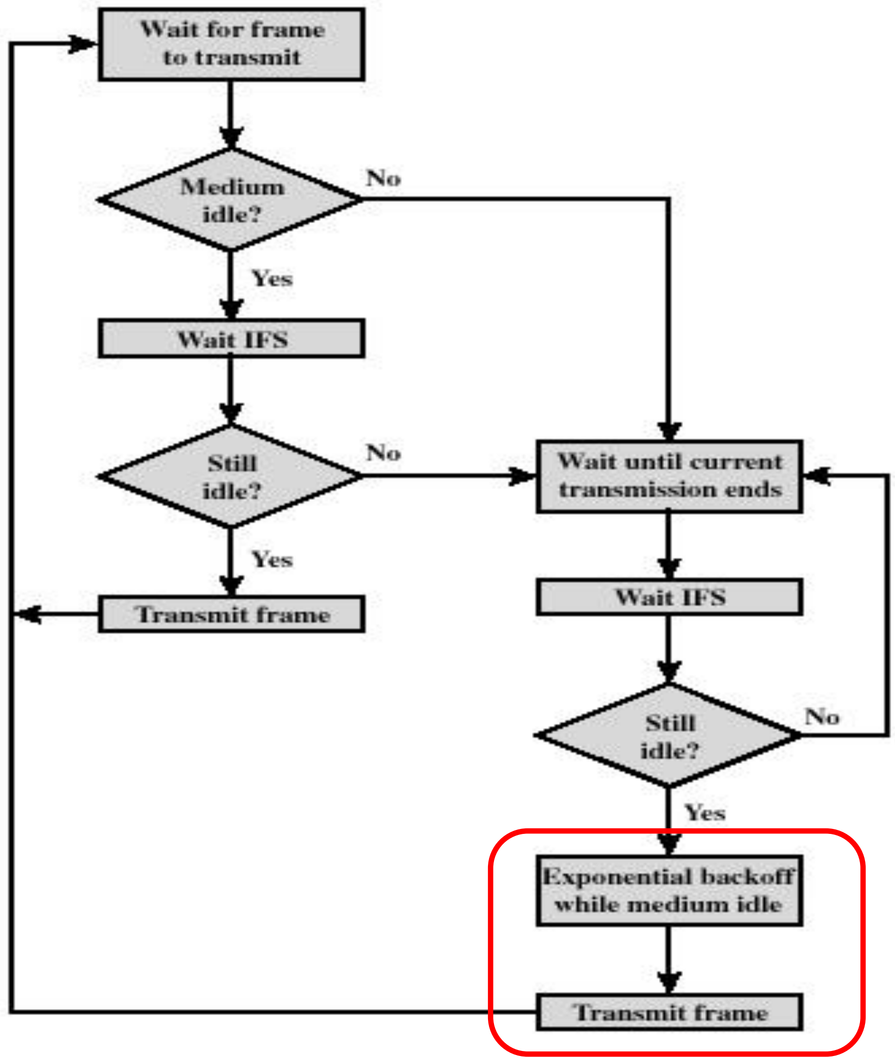
IEEE 802.11 MAC: Distributed Coordination Function (DCF)

- Make use of CSMA (carrier sense multiple access)
- Use set of delays generic called Interframe Space (IFS)

Algorithm Logic:

1. Station sense the medium
2. If medium idle, wait IFS, then if still idle transmit frame
3. If medium busy or become busy, defer and monitor the medium until idle
4. Then, delay IFS and sense medium
5. If medium idle, exponential backoff and if then idle, station transmit

- Binary exponential backoff
-> handle heavy load



IEEE 802.11 MAC: Distributed Coordination Function (DCF)

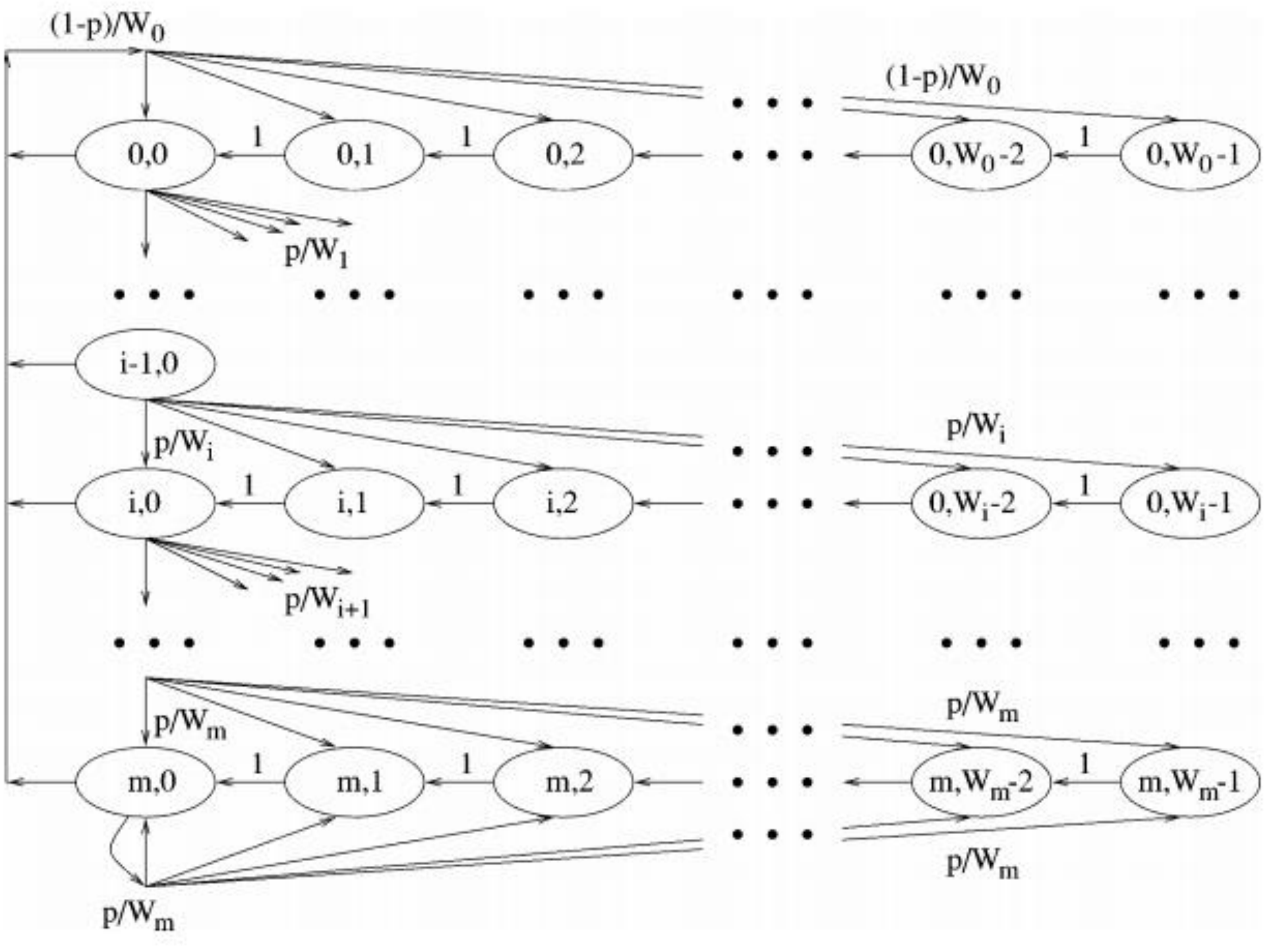


Fig. 4. Markov Chain model for the backoff window size.

G. Bianchi, **Performance analysis of the IEEE 802.11 distributed coordination function, JSAC 2000**

IEEE 802.11 MAC protocol: parameters

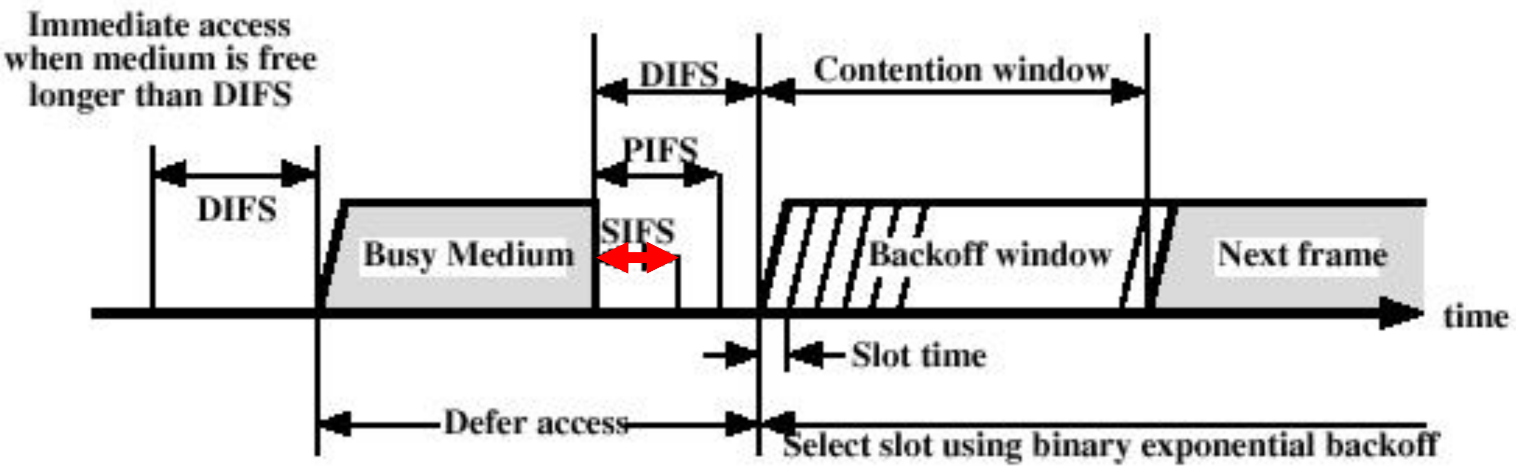
Table 2.1. IEEE 802.11 parameters

Parameter	**802.11 (FHSS)**	**802.11 (DSSS)**	**802.11 (IR)**	**802.11b**	**802.11a**
t_{slot}	50 μsec	20 μsec	8 μsec	20 μsec	9 μsec
SIFS	28 μsec	10 μsec	10 μsec	10 μsec	16 μsec
PIFS	SIFS $+t_{slot}$				
DIFS	SIFS$+(2 \times t_{slot})$				
Operating Frequency	2.4 GHz	2.4 GHz	850-950 nm	2.4 GHz	5 GHz
Maximum Data Rate	2 Mbps	2 Mbps	2 Mbps	11 Mbps	54 Mbps
CWmin	15	31	63	31	15
CWmax	1,023	1,023	1,023	1,023	1,023

IEEE 802.11 MAC: DCF, cont'd

Priority-based scheme - use 3 values for IFS:

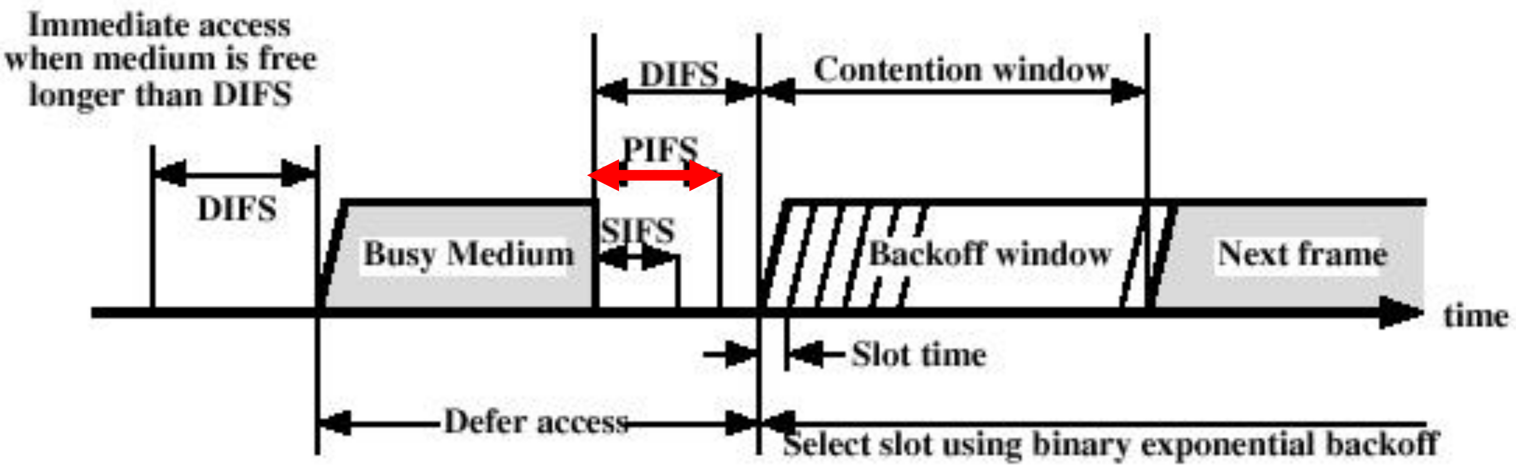
- SIFS (short IFS): shortest IFS used for immediate responses such as ACK, CTS, poll response
- PIFS (point coordination function IFS): middle length IFS used for issuing polls by a centralized controller
- DIFS (distributed coordination function IFS): longest IFS used for regular asynchronous frames



IEEE 802.11 MAC: DCF, cont'd

Priority-based scheme - use 3 values for IFS:

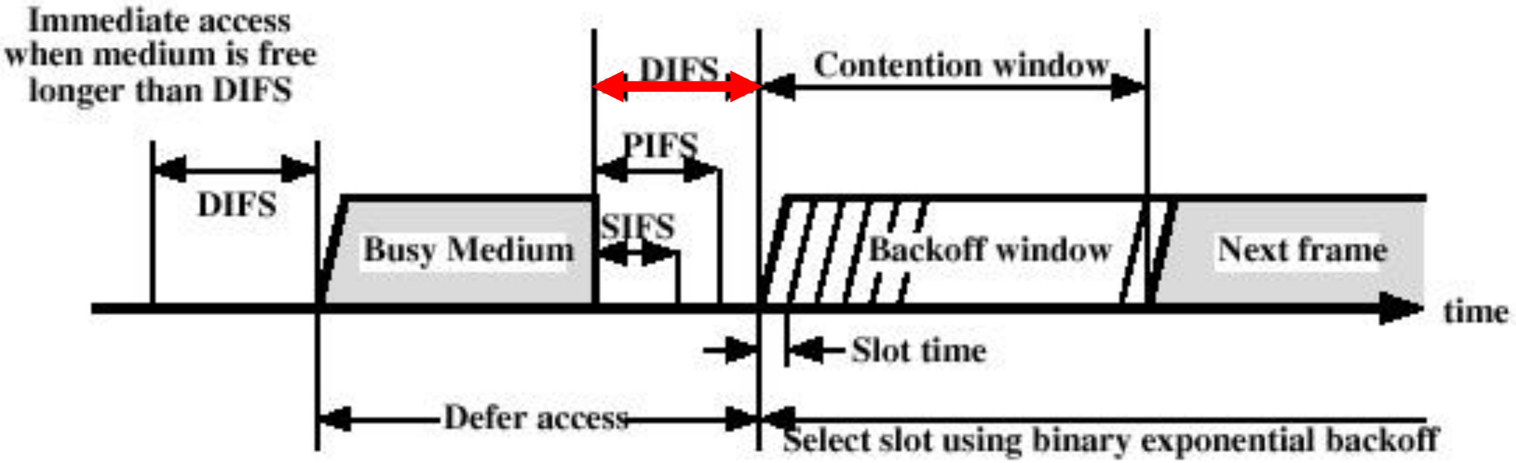
- SIFS (short IFS): shortest IFS used for immediate responses such as ACK, CTS, poll response
- PIFS (point coordination function IFS): middle length IFS used for issuing polls by a centralized controller
- DIFS (distributed coordination function IFS): longest IFS used for regular asynchronous frames



IEEE 802.11 MAC: DCF, cont'd

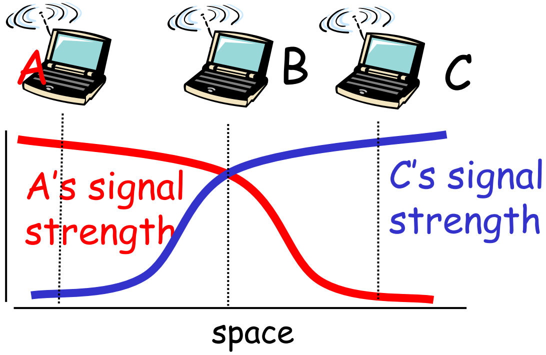
Priority-based scheme - use 3 values for IFS:

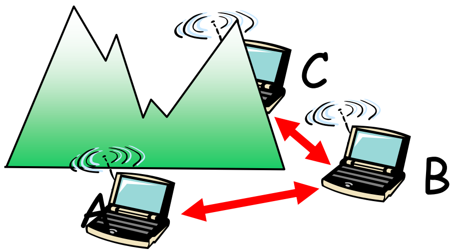
- SIFS (short IFS): shortest IFS used for immediate responses such as ACK, CTS, poll response
- PIFS (point coordination function IFS): middle length IFS used for issuing polls by a centralized controller
- DIFS (distributed coordination function IFS): longest IFS used for regular asynchronous frames



Hidden Terminal Problem

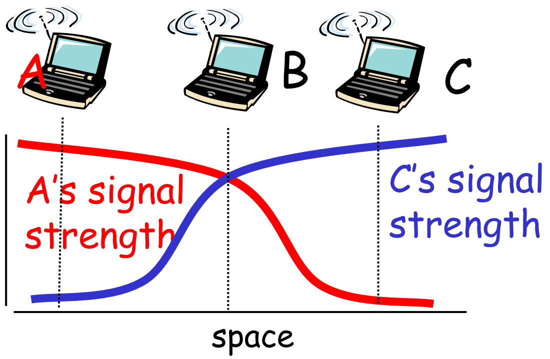
Hidden terminal problem (ad-hoc and WLAN)

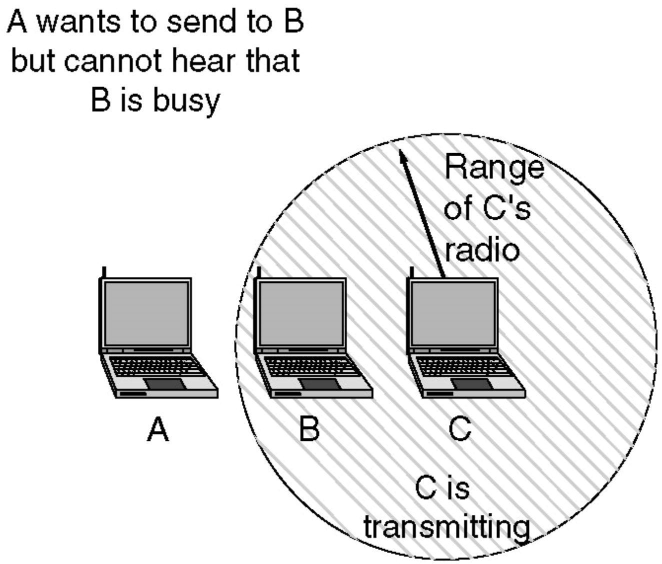


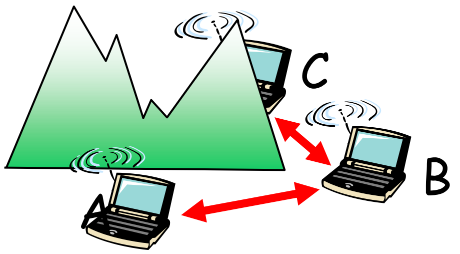


Hidden Terminal Problem

Hidden terminal problem (ad-hoc and WLAN)



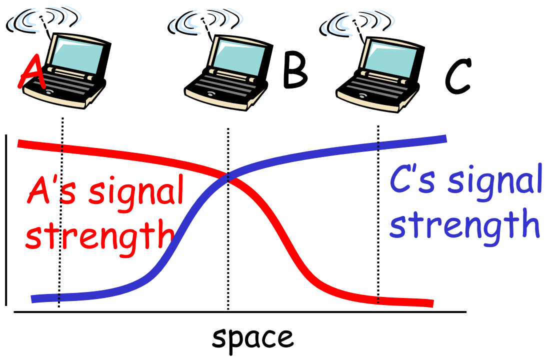


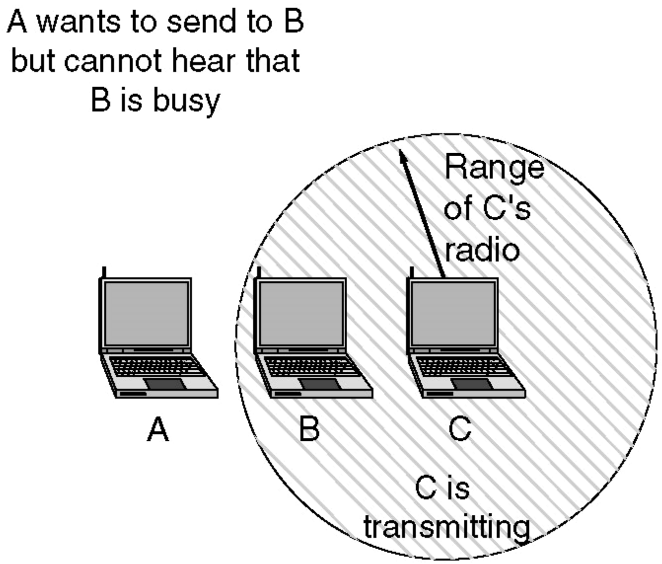


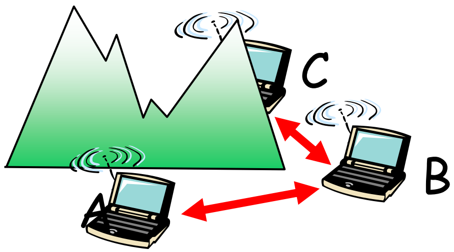
Hidden Terminal Problem

Hidden terminal problem (ad-hoc and WLAN)
- medium free near the transmitter
- medium not free near the receiver

=> *Packet collision*





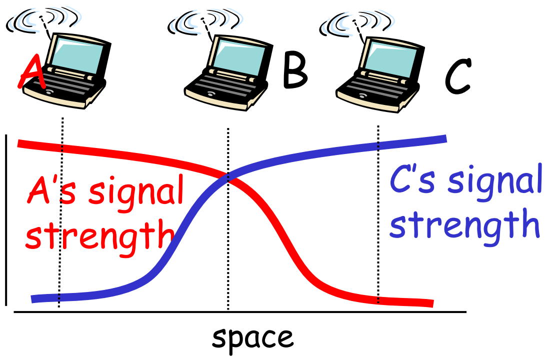


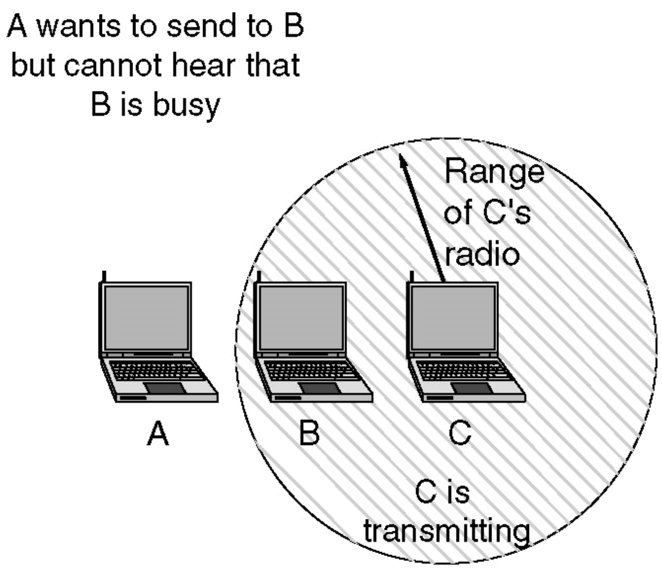
Hidden Terminal Problem

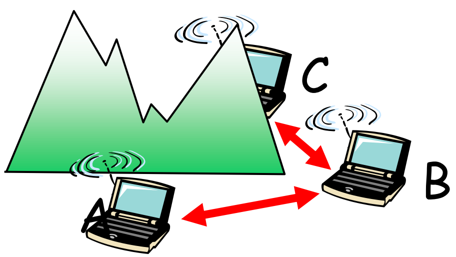
Hidden terminal problem (ad-hoc and WLAN)
- medium free near the transmitter
- medium not free near the receiver

=> *Packet collision*

Possible solution:
- MAC scheme using RTS-CTS scheme







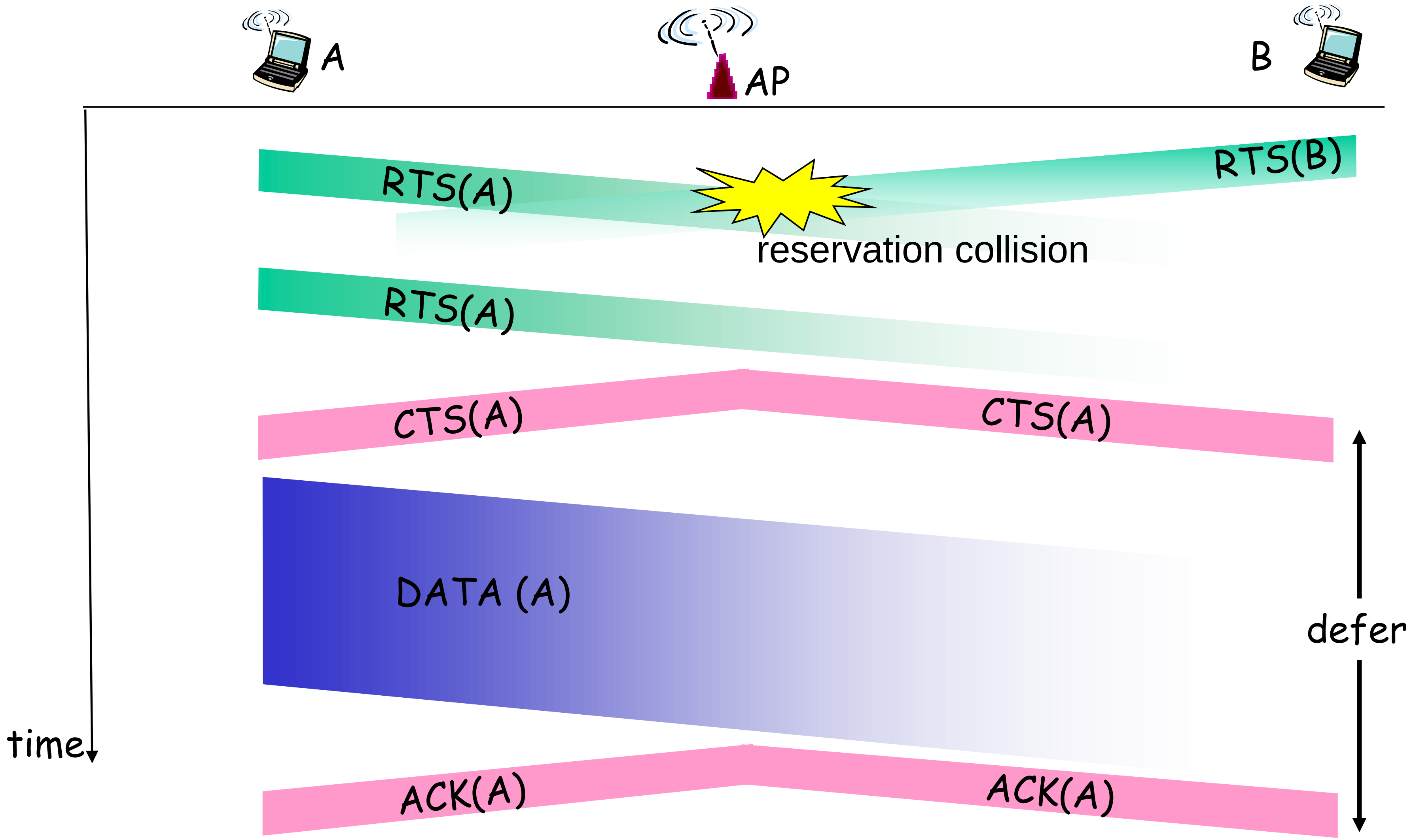
Avoiding collisions (more)

idea: allow sender to "reserve" channel rather than random access of data frames: avoid collisions of long data frames

- sender first transmits *small* request-to-send (RTS) packets to base station using CSMA
  - RTS may still collide with each other (but they're short)
- BS broadcasts clear-to-send CTS to host in response to RTS
- RTS heard by all nodes because of broadcast property
  - sender transmits (large) data frame
  - other stations defer transmissions until it is done

Avoid data frame collisions completely using small reservation packets!

Collision Avoidance: RTS-CTS exchange



Hidden Terminal Problem [2]

Problems with RTS-CTS solution:

- possible collisions between CTS and RTS
- collisions between data packets due to multiple different CTS granted to different neighboring nodes

- Also, adds delay and overhead …
  - Only beneficial used for very large packets
  - Typically, not used in practice (unless very large packets)…

802.11 frame: addressing

frame control	duration	address 1	address 2	address 3	seq control	address 4	payload	CRC
2	2	6	6	6	2	6	0 - 2312	4

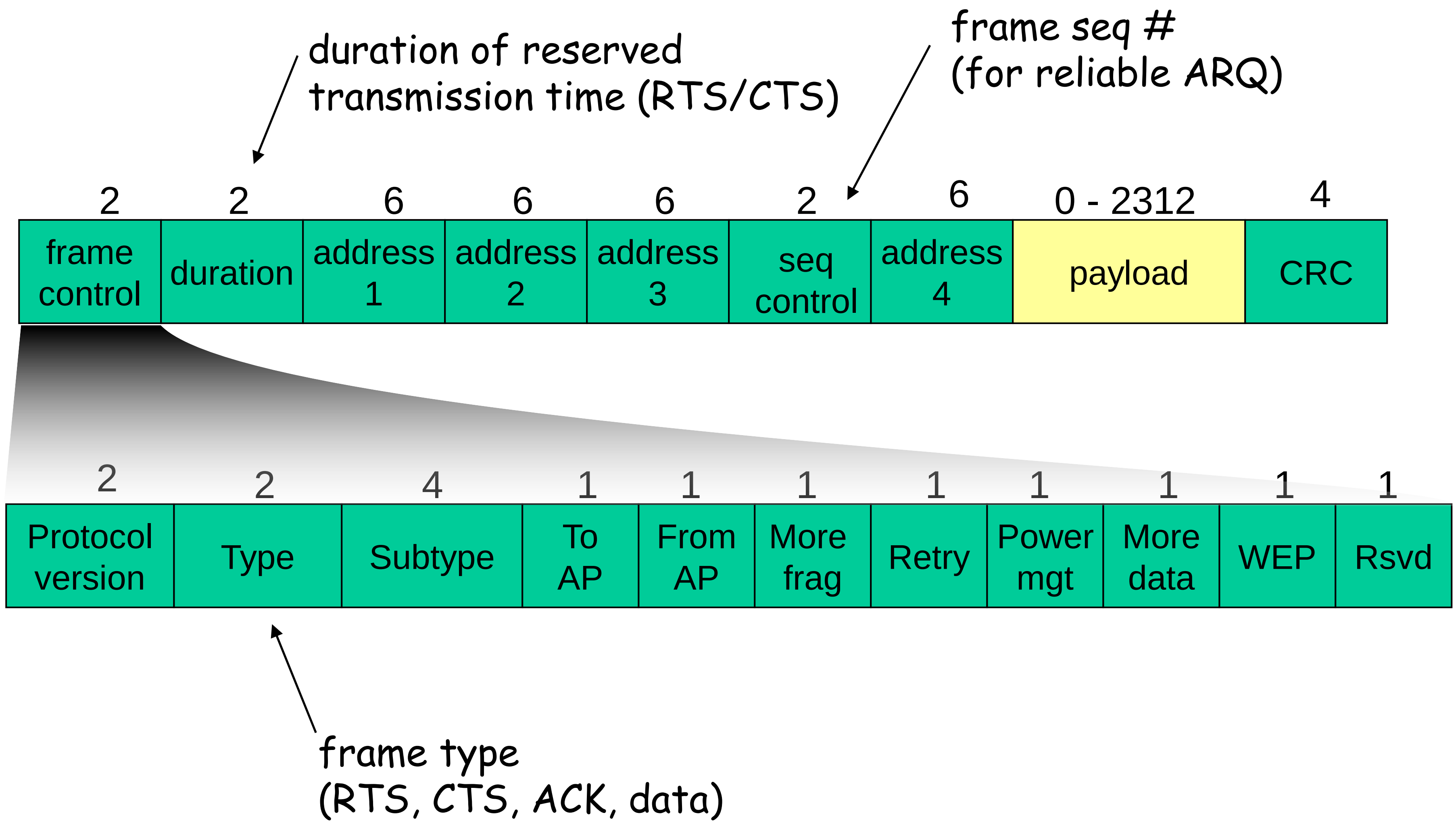
Address 1: MAC address of wireless host or AP to receive this frame

Address 2: MAC address of wireless host or AP transmitting this frame

Address 3: MAC address of router interface to which AP is attached

Address 4: used only in ad hoc mode

802.11 frame: more



Exposed Terminal Problems

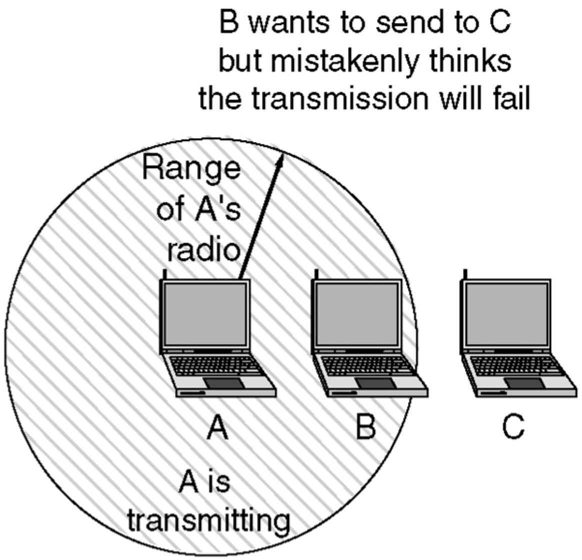
Exposed terminal problem (ad-hoc and WLAN)

- medium free near the receiver
- medium busy near the transmitter

    => *Waist of bandwidth*

Possible solutions:

- directional antennas
- separate channels for control and data



Issues, medium access schemes

- Distributed operation
- Synchronization
- Hidden terminals
- Exposed terminals
- Throughput
- Access delay
- Fairness
- Real-time traffic support
- Resource reservation
- Ability to measure resource availability
- Capability for power control
- Adaptive rate control
- Use of directional antennas

Issues, medium access schemes

- Distributed operation
- Synchronization
- Hidden terminals
- Exposed terminals
- Throughput
- Access delay
- Fairness
- Real-time traffic support
- Resource reservation
- Ability to measure resource availability
- Capability for power control
- Adaptive rate control
- Use of directional antennas

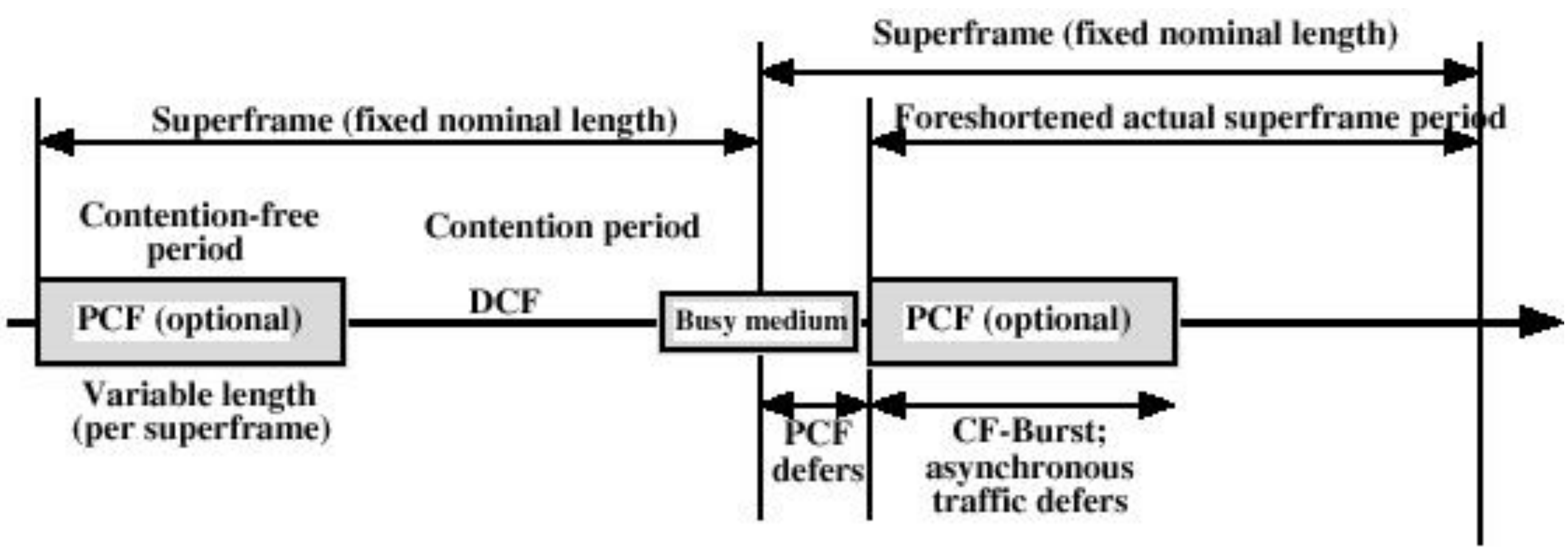
Issues, medium access schemes

- Distributed operation
- Synchronization
- Hidden terminals
- Exposed terminals
- Throughput
- Access delay
- Fairness
- Real-time traffic support
- Resource reservation
- Ability to measure resource availability
- Capability for power control
- Adaptive rate control
- Use of directional antennas

A few more words about QoS

IEEE 802.11 MAC: Point Coordination Function (PCF)

- Alternative access method on top of DCF
- Polling operation by a centralized master
- Use PIFS when issuing polls
- To avoid locking out the asynchronous traffic the superframe is used



IEEE 802.11 MAC Frame Types

- *Six types of control frames*

- Power save - poll (PS-poll)
- Request to send (RTS)
- Clear to send (CTS)
- Acknowledgment (ACK)
- Contention-free (CF)-end
- CF-end + CF-Ack

- *Eight types of data frames*

➔ *Carry user data*

- Data
- Data + CF-Ack
- Data + CF-poll
- Data + CF-Ack + CF-poll

➔ *Do not carry user data*

- Null Function
- CF-Ack
- CF-Poll
- CF-Ack + CF-Poll

- ***Management frames***

- association request and association response
- reassociation request and reassociation response
- probe request and probe response
- beacon
- announcement traffic indication message
- disassociation
- authentication and deauthentication

IEEE 802.11e: Enhanced Distribution Coordination Function (EDCF)

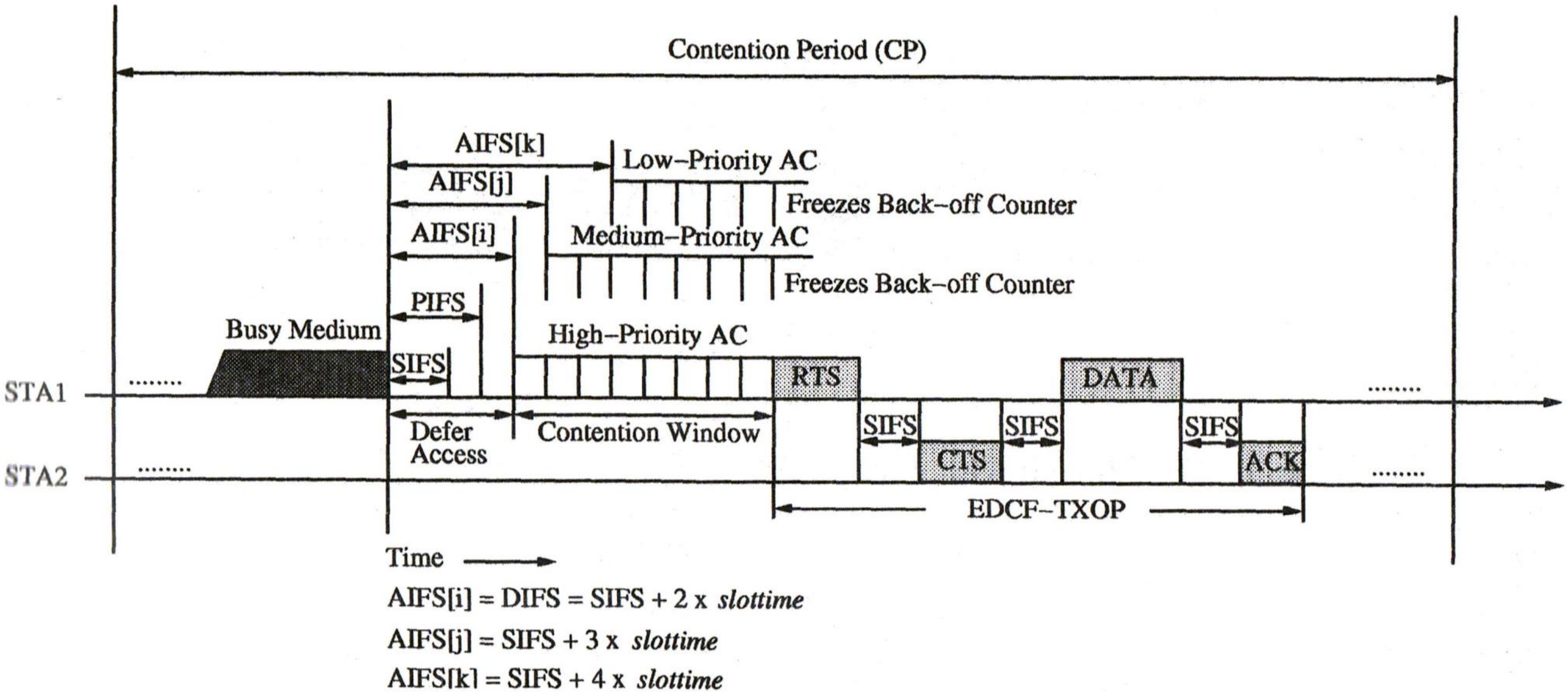


Figure 10.5. An example of EDCF access mechanism.

IEEE 802.11 MAC protocol: parameters

Table 2.1. IEEE 802.11 parameters

Parameter	**802.11 (FHSS)**	**802.11 (DSSS)**	**802.11 (IR)**	**802.11b**	**802.11a**
t_{slot}	50 μsec	20 μsec	8 μsec	20 μsec	9 μsec
SIFS	28 μsec	10 μsec	10 μsec	10 μsec	16 μsec
PIFS	SIFS $+t_{slot}$				
DIFS	SIFS$+(2 \times t_{slot})$				
Operating Frequency	2.4 GHz	2.4 GHz	850-950 nm	2.4 GHz	5 GHz
Maximum Data Rate	2 Mbps	2 Mbps	2 Mbps	11 Mbps	54 Mbps
CWmin	15	31	63	31	15
CWmax	1,023	1,023	1,023	1,023	1,023