

# CS/ECE 439: Wireless Networking

MAC Layer – Road to Wireless

# Multiple Access Media

---

- ▶ **Media access**

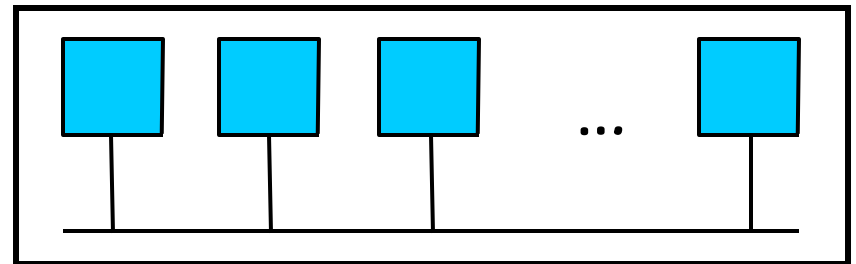
- ▶ Controlling which frame should be sent over the link next
  - ▶ Easy for point-to-point links; half versus full duplex
  - ▶ Harder for multi-access links: who gets to send?

- ▶ **Multiple senders on some media**

- ▶ Buses (Ethernet)
- ▶ Radio, Satellite

- ▶ **Goals**

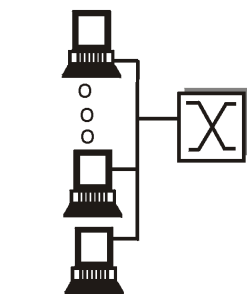
- ▶ Fair arbitration
- ▶ Good performance



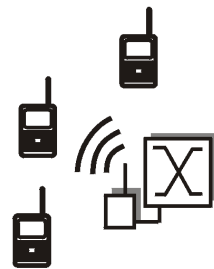
# Point-to-Point vs. Broadcast Media

---

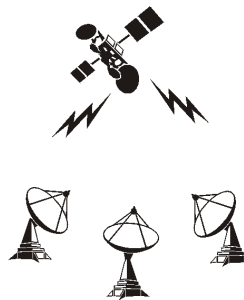
- ▶ **Point-to-point: dedicated pairwise communication**
  - ▶ Long-distance fiber link
  - ▶ Point-to-point link between Ethernet switch and host
- ▶ **Broadcast: shared wire or medium**
  - ▶ Traditional Ethernet
  - ▶ 802.11 wireless LAN



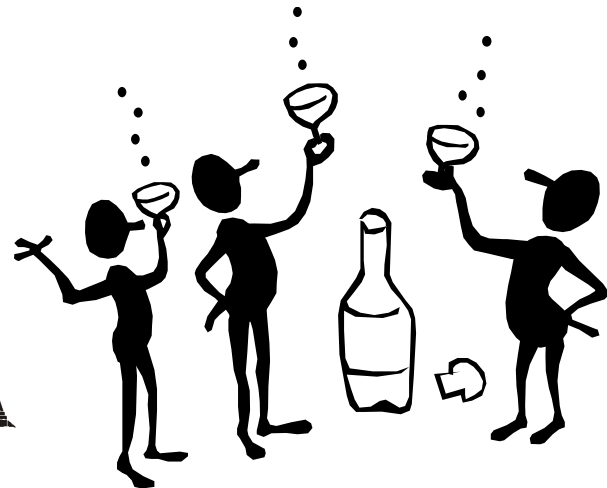
shared wire  
(e.g. Ethernet)



shared wireless  
(e.g. Wavelan)



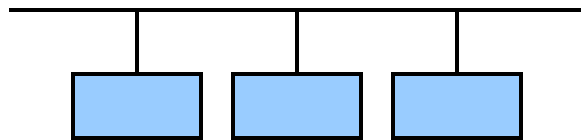
satellite



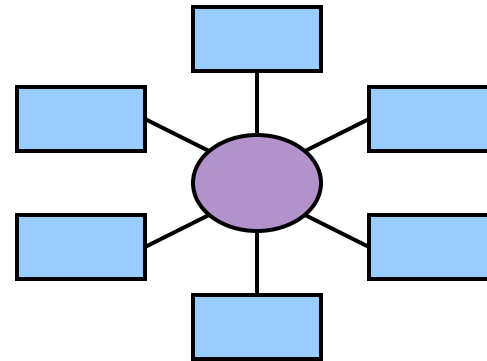
cocktail party

# Types of Shared Link Networks

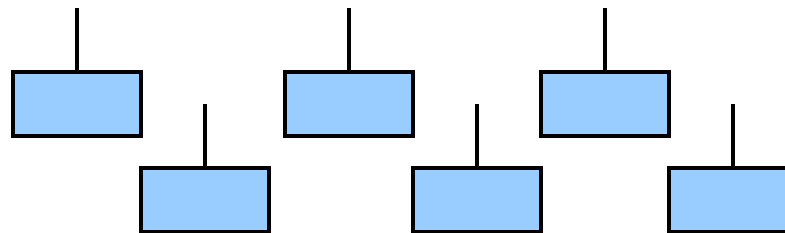
---



Bus Topology: Shared  
Ethernet



Star Topology: Active or Passive Hub



Wireless: Shared  
IEEE 802.11, BT, ZigBee

# Multiple Access Algorithm

---

- ▶ **Single shared broadcast channel**
  - ▶ Must avoid having multiple nodes speaking at once
  - ▶ Otherwise, collisions lead to garbled data
  - ▶ Need distributed algorithm for sharing the channel
  - ▶ Algorithm determines which node can transmit
- ▶ **Typical assumptions**
  - ▶ Communication needs vary
    - ▶ Over time
    - ▶ Between hosts
  - ▶ Network is not fully utilized
- ▶ [video](#)



# Multiple Access Media

---

- ▶ Which kind of multiplexing is best?
  - ▶ Channel partitioning: divide channel into pieces
    - ▶ Frequency-division multiplexing (FDM, separate bands)
  - ▶ Taking turns: scheme for trading off who gets to transmit
    - ▶ Time-division multiplexing (TDM, synchronous time slots)
    - ▶ Statistical time-division multiplexing (STDM, time slots on demand)
- ▶ These techniques are useful
  - ▶ But they have a number of limitations
  - ▶ They do not support bursty traffic efficiently
    - ▶ Lots of unused capacity, ...
    - ▶ ... while active users squeeze their bit stream through a very thin pipe
  - ▶ Work best in a provisioned service
    - ▶ Management of frequencies, time slots, placement of devices, etc.



# Multiple Access Media: Random Access

---

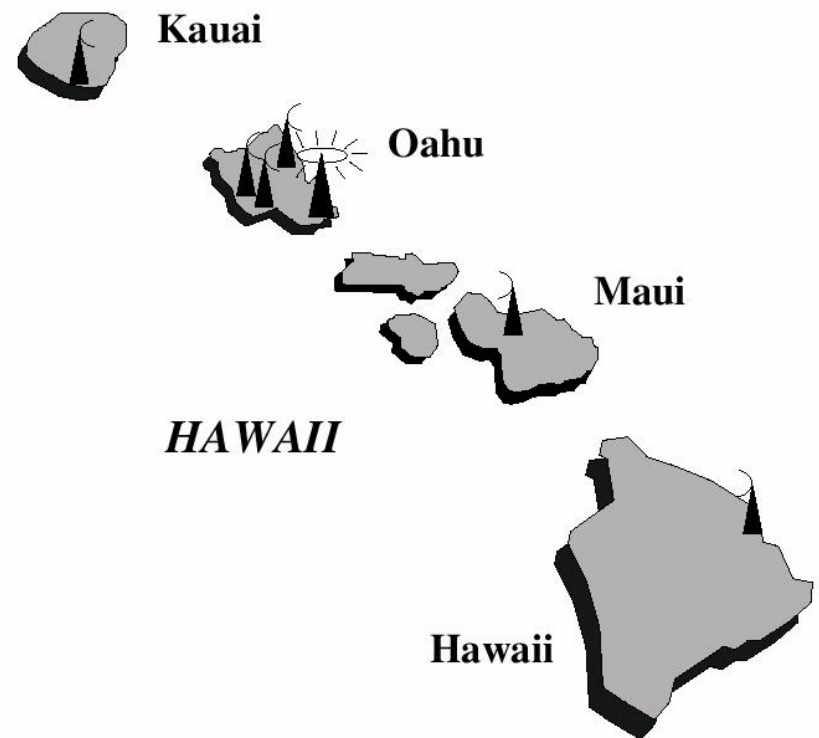
- ▶ **Random access**
  - ▶ Allow collisions, and then recover
  - ▶ Optimize for the common case (no collision)
  - ▶ Don't avoid collisions, just recover from them....
- ▶ **When node has packet to send**
  - ▶ Transmit at full channel data rate
  - ▶ No a priori coordination among nodes
- ▶ **Two or more transmitting nodes  $\Rightarrow$  collision**
  - ▶ Data lost
- ▶ **Random access MAC protocol specifies**
  - ▶ How to detect collisions
  - ▶ How to recover from collisions



# Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

---

- ▶ Aloha Packet Radio Network
  - ▶ First data communication system for Hawaiian islands
  - ▶ Hub at U. Hawaii, Oahu
  - ▶ Two radio channels
    - ▶ Random access: for sites sending data
    - ▶ Broadcast for hub rebroadcasting data
- ▶ Ethernet
  - ▶ CSMA/CD for LANs





# Pure ALOHA

---

- ▶ Developed in University of Hawaii in early 1970's
- ▶ Keep it simple
  - ▶ User transmits at will
  - ▶ If two or more messages overlap in time → collision
    - ▶ Receiver cannot decode packets
  - ▶ Wait roundtrip time plus a fixed increment → collision
    - ▶ Lack of ACK
  - ▶ After a collision
    - ▶ Colliding stations retransmit
    - ▶ Stagger attempts randomly to reduce repeat collisions
  - ▶ After several attempts, senders give up
- ▶ Simple but wasteful
  - ▶ Max efficiency of at most  $1/(2e) = 18\%$ !



# Pure ALOHA

---

- ▶ User model

- ▶ N transmitters

- ▶ Each transmitter hooked to one terminal
    - ▶ One person at each terminal

- ▶ Person types a line, presses return

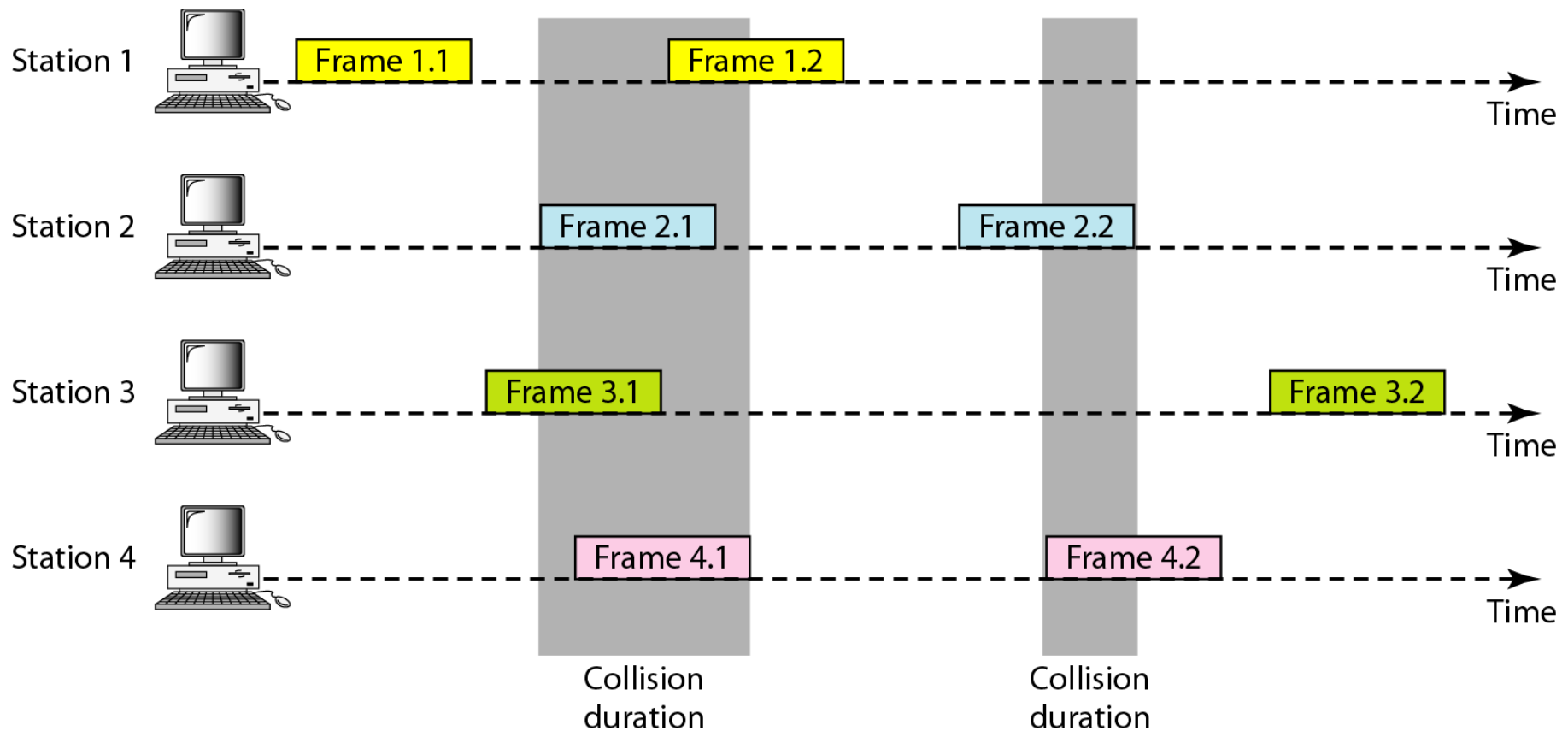
- ▶ Transmitter sends line
    - ▶ Each station transmits  $\lambda$  packets/sec on average based on a Poisson arrival process

- ▶ Checks for success (no interference)

- ▶ If collision occurred, wait random time and resend



# Pure ALOHA



# Pure ALOHA

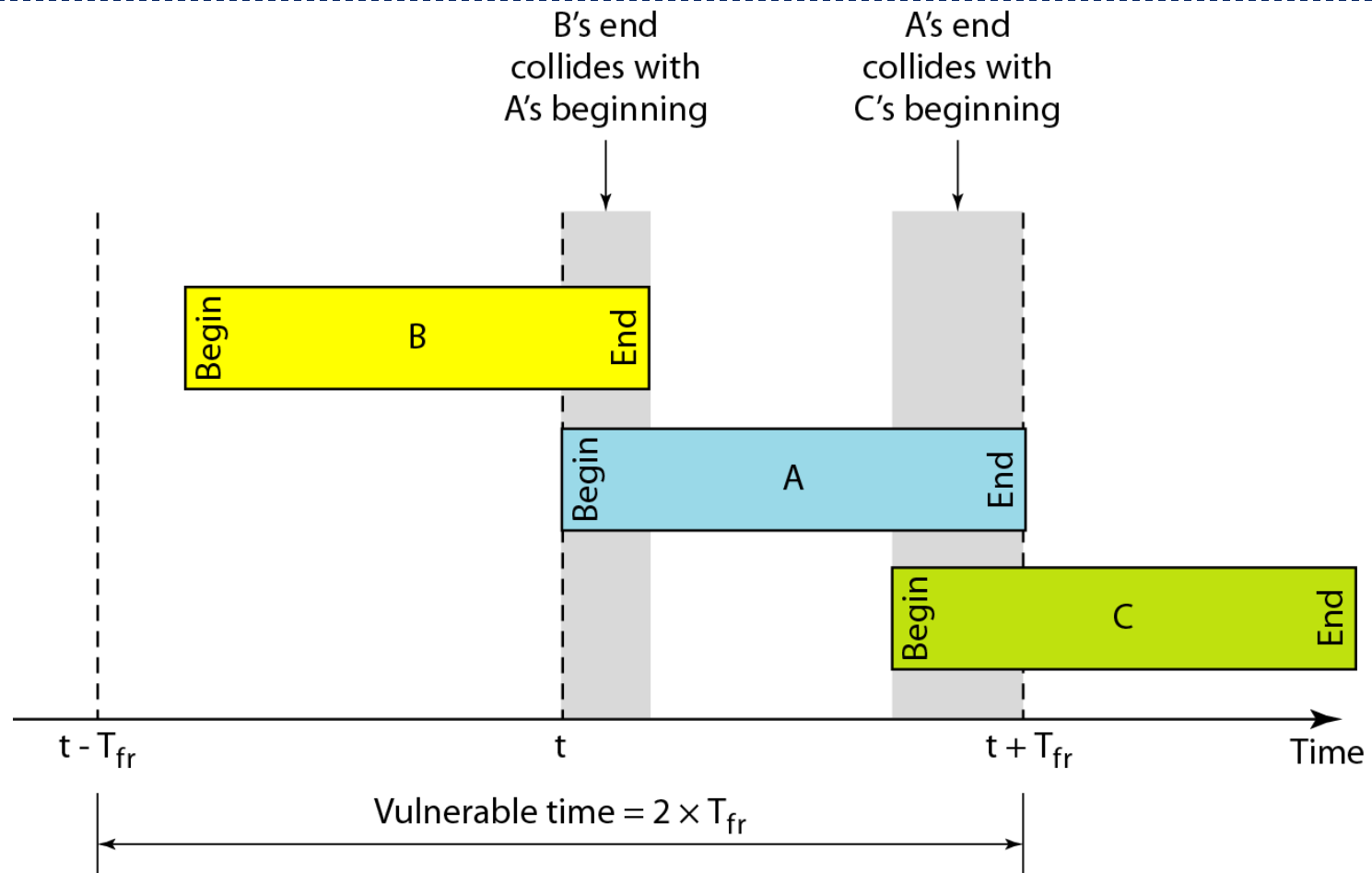
---

## ► Collisions

- A frame will not suffer a collision if no other frames are sent within one frame time of its start
- Let  $t$  = time to send a frame
- If any other user has generated a frame between time  $t_0$  and time  $t_0 + t$ , the end of that frame will collide with the beginning of our frame
- Similarly, any other frame started between time  $t_0 + t$  and time  $t_0 + 2t$  will collide with the end of our frame



# Pure ALOHA



# Pure ALOHA

---

- ▶ Also assume fixed packet sizes (maximizes throughput)
- ▶ Arrival and success rates
  - ▶ Frames generated at rate  $S$ 
    - ▶ In steady state, must leave at  $S$  as well
  - ▶ Some frames retransmitted
    - ▶ Assume also Poisson with rate  $G$ ,  $G > S$
- ▶  $S = G P_0$ 
  - ▶  $P_0$  is the probability of successful transmission

# Pure Aloha Analysis

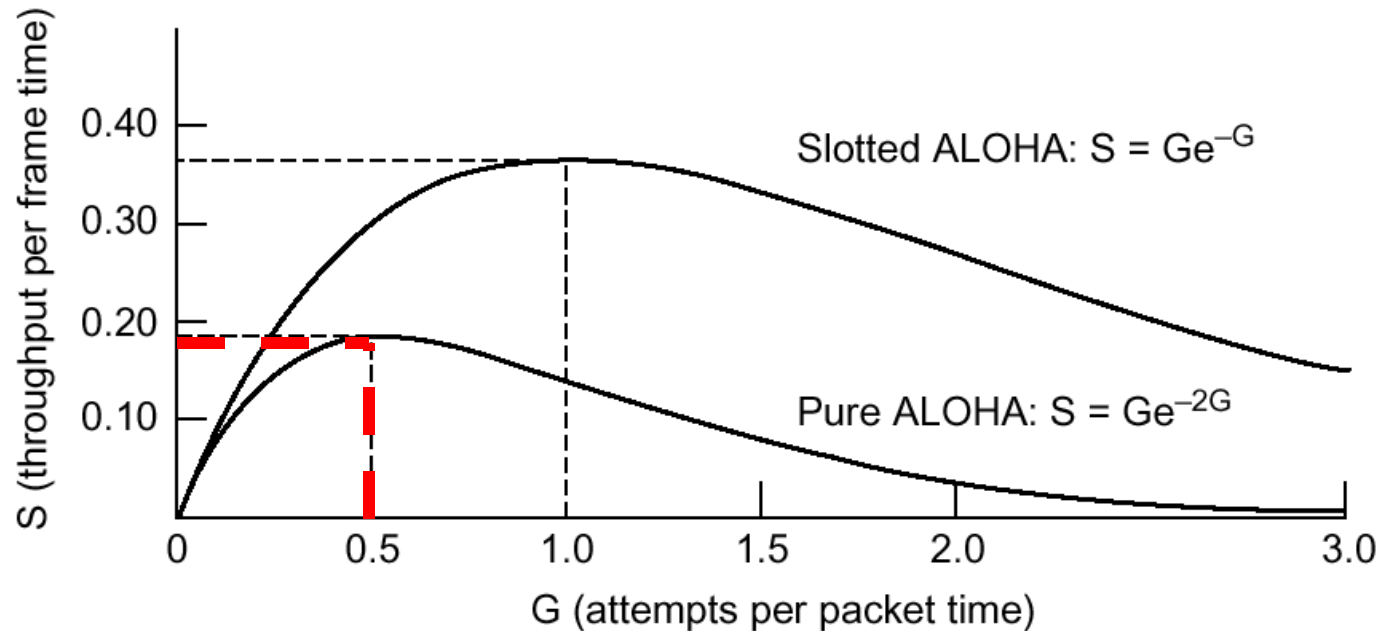
- ▶ Maximum throughput

- ▶  $G = 0.5$

- ▶  $S = 1/2e$

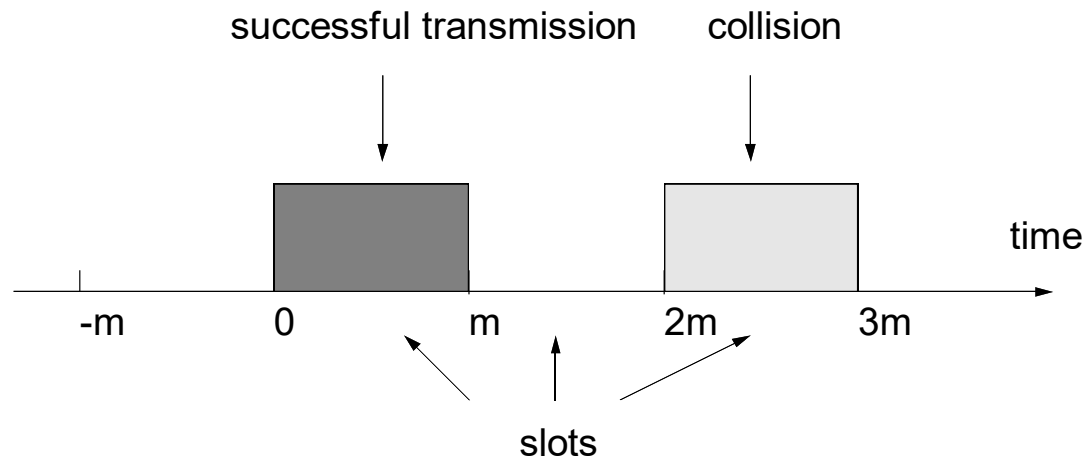
- Utilization

- Maximum of 0.184!



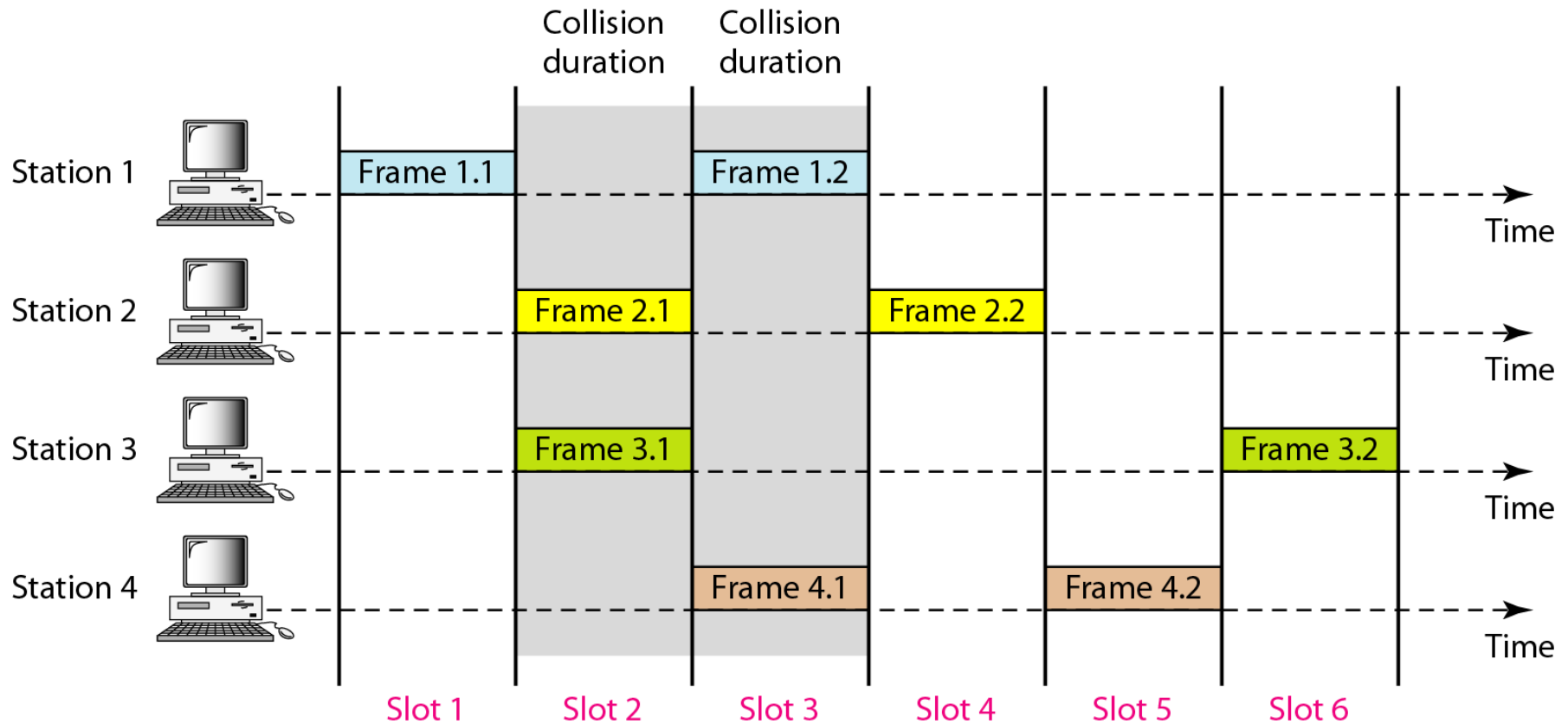
# Slotted ALOHA

- ▶ Hosts wait for next slot to transmit
  - ▶ Slot time units =  $m$  (message length)
  - ▶ Modify Aloha by allowing users to attempt transmission at the beginning of a time slot only
  - ▶ All users need to be synchronized in time.
- ▶ Vulnerable period is now cut in half ( $T$ )
  - ▶ Doubles max throughput





# Slotted Aloha



# Slotted ALOHA

---

- ▶ In each interval  $m$

- ▶ Mean number of frames generated is  $G$
- ▶ The probability of no other traffic being generated during the entire vulnerable period is
  - ▶  $P_0 = e^{-G}$
  - ▶  $S = Ge^{-G}$

Note: Not  $2G$

- ▶ Max  $S$   $1/e = 0.368$

- ▶ at  $G = 1$ .

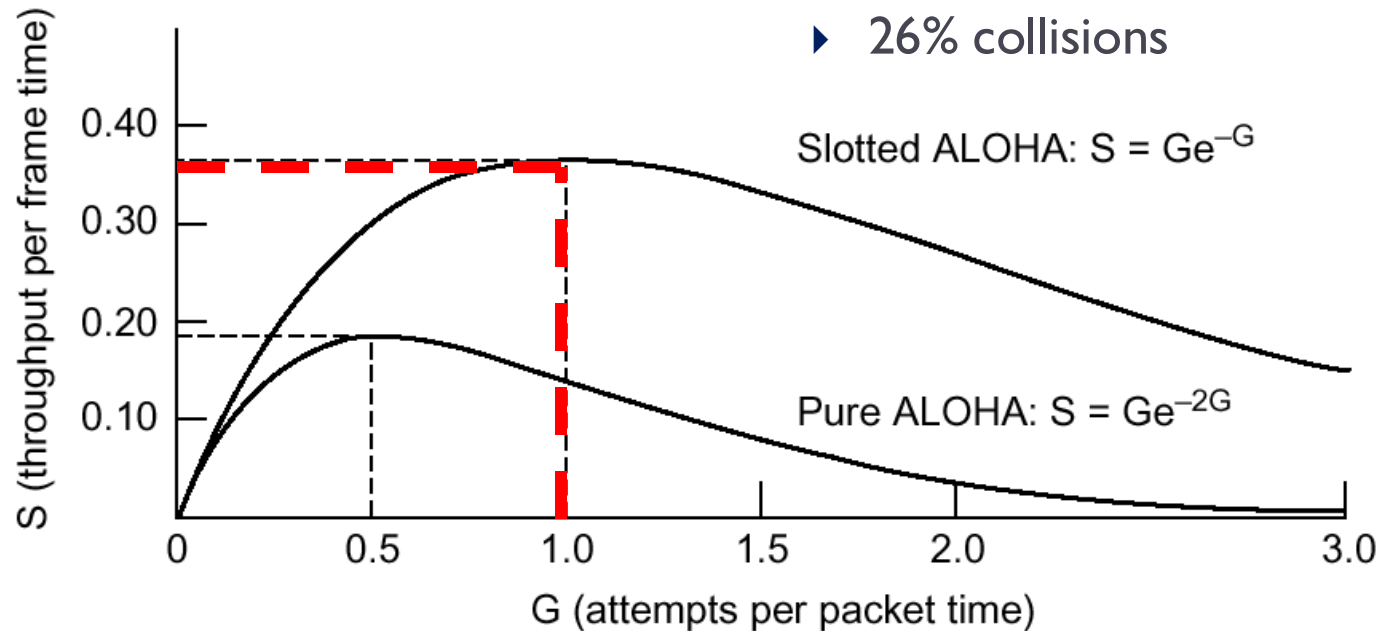
# Slotted ALOHA

- ▶ Maximum throughput

- ▶  $G = 1$
- ▶  $S = 1/e$

- ▶ Utilization

- ▶ Maximum of 0.368!
- ▶ 37% empty slots
- ▶ 37% successes
- ▶ 26% collisions



# Slotted ALOHA

---

## ▶ Pros

- ▶ Single active node can continuously transmit at full rate of channel
- ▶ Highly decentralized: only need slot synchronization
- ▶ Simple

## ▶ Cons

- ▶ Wasted slots:
  - ▶ Idle
  - ▶ Collisions
- ▶ Nodes should detect collision in less than time to transmit packet
- ▶ Clock synchronization



# Slotted ALOHA

---

- ▶ Performance

- ▶ Higher values of  $G$

- ▶ Reduces the number of empty slots
    - ▶ Increases the number of collisions exponentially

- ▶ Small increases in channel load can drastically reduce performance

- ▶ Limitations

- ▶ Slotted Alohas has twice the performance of basic Aloha, but performance is still poor

- ▶ Slotted design is also not very efficient when carrying variable sized packets!
    - ▶ Also (slightly) longer delay than pure Aloha

- ▶ Still, not bad for an absolutely minimal protocol!

- ▶ How do we go faster?



# ALOHA Analysis

---

- ▶ **Tradeoff**

- ▶ Pure ALOHA provides smaller delays
- ▶ Slotted ALOHA provides higher throughput



# From Aloha comes Ethernet

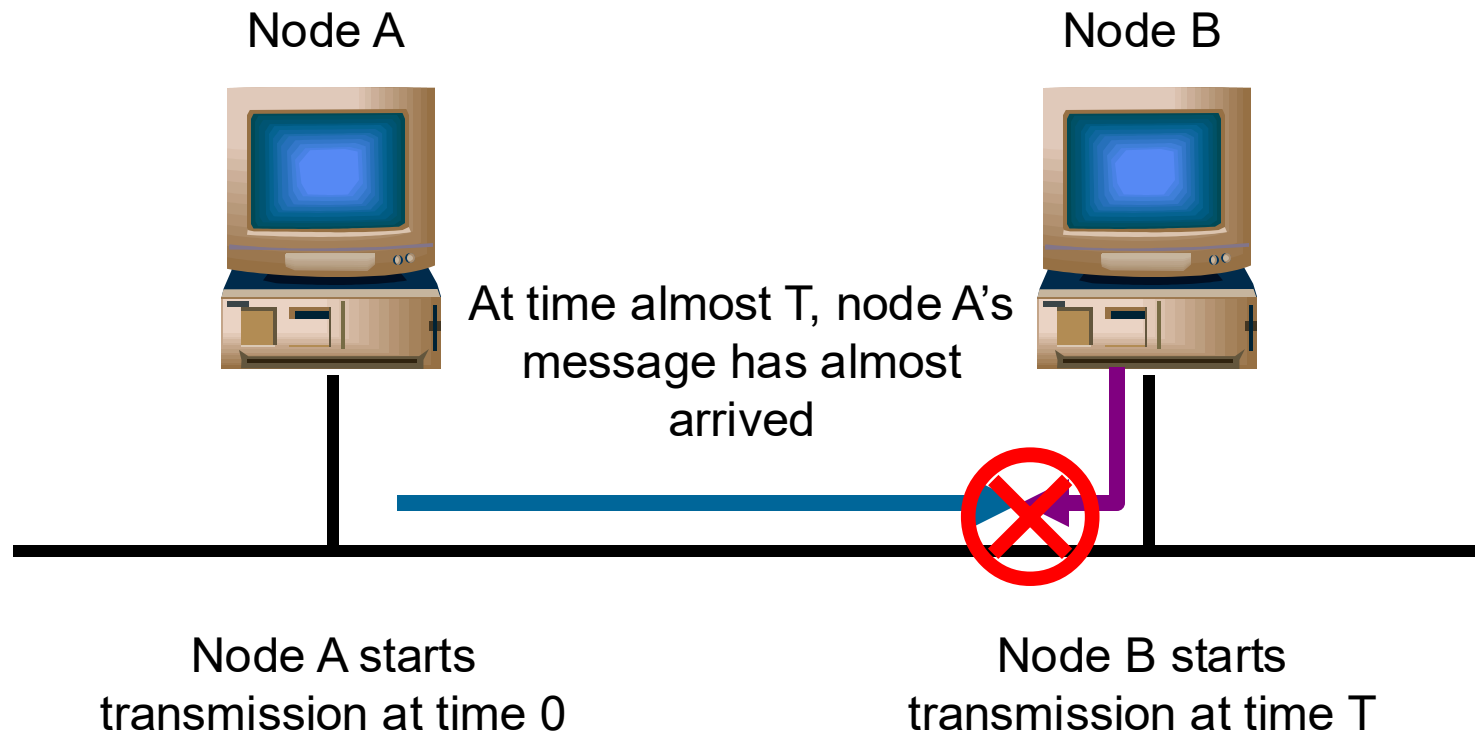
---

- ▶ Ethernet - CSMA/CD
- ▶ CS – Carrier Sense
  - ▶ Nodes can distinguish between an idle and a busy link
- ▶ MA - Multiple Access
  - ▶ A set of nodes send and receive frames over a shared link
- ▶ CD – Collision Detection
  - ▶ Nodes listen during transmission to determine if there has been interference



# Ethernet MAC Algorithm

---



How can we ensure that A knows about the collision?



# Collision Detection

---

- ▶ Problem

- ▶ How can A detect a collision?

- ▶ Solution

- ▶ A must still be transmitting when it receives B's transmission!

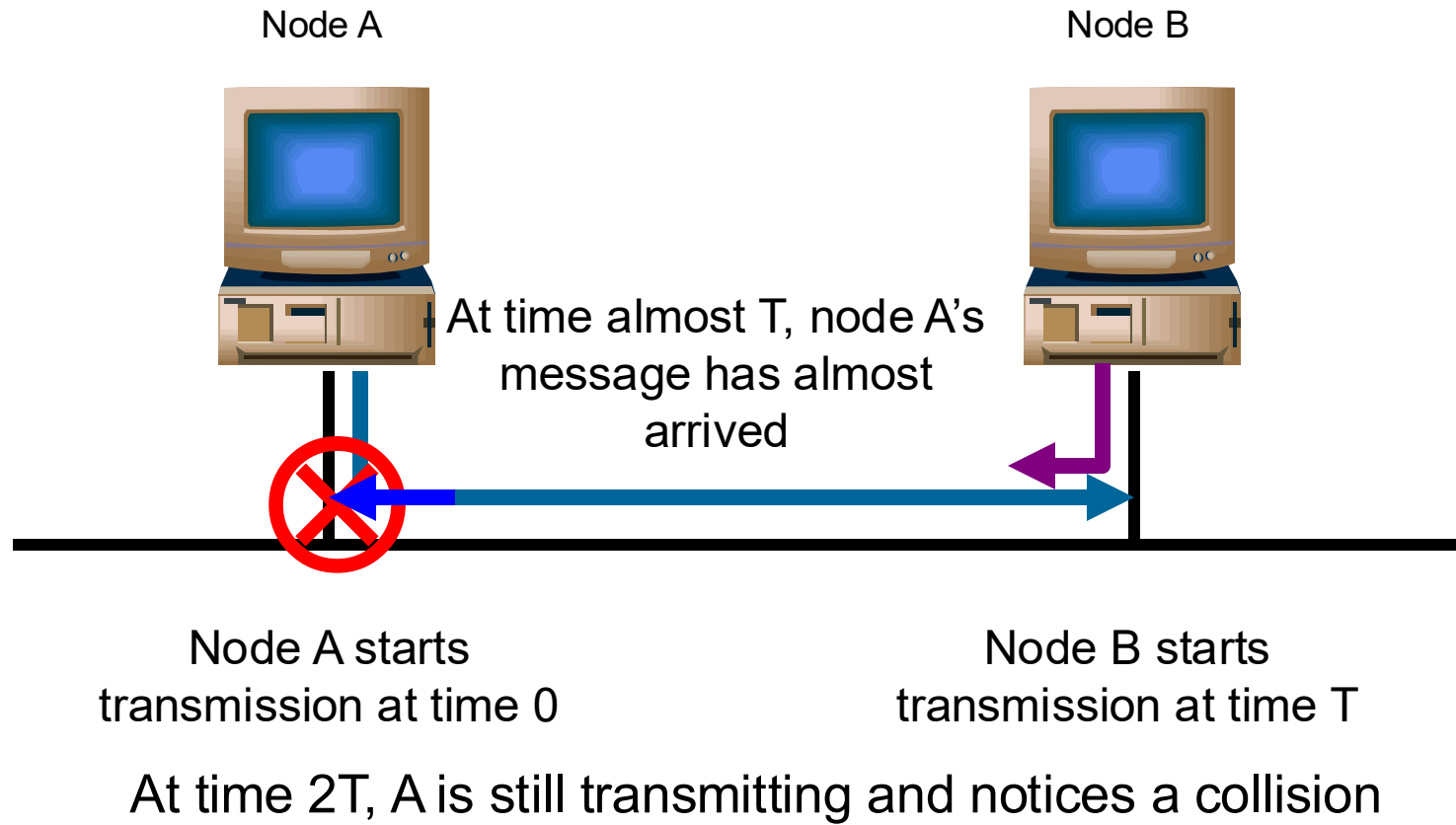
- ▶ Example

- ▶ Node A's message reaches node B at time  $T$
  - ▶ Node B's message reaches node A at time  $2T$
  - ▶ For node A to detect a collision, node A must still be transmitting at time  $2T$



# Ethernet MAC Algorithm

---



# Collision Detection

---

- ▶ **IEEE 802.3**

- ▶ 2T is bounded to  $51.2\mu\text{s}$
- ▶ At 10Mbps  $51.2\mu\text{s} = 512\text{b}$  or  $64 = 512\text{b}$  or  $64\text{B}$
- ▶ Packet length  $\geq 64\text{B}$

- ▶ **Jam after collision**

- ▶ Ensures that all hosts notice the collision



# Ethernet MAC Algorithm

---

## ▶ Sender/Transmitter

- ▶ If line is idle (carrier sensed)
  - ▶ Send immediately
  - ▶ Send maximum of 1500B data (1527B total)
  - ▶ Wait 9.6  $\mu$ s before sending again
- ▶ If line is busy (no carrier sense)
  - ▶ Wait until line becomes idle
  - ▶ Send immediately (1-persistent)
- ▶ If collision detected
  - ▶ Stop sending and jam signal
  - ▶ Try again later

Why have a max size?

Want to prevent one node from taking over completely

Why 9.6  $\mu$ s?

Too long: wastes time  
Too short: doesn't allow other nodes to transmit (fairness)

Incoming signal  $\neq$   
outgoing signal!



# Retransmission

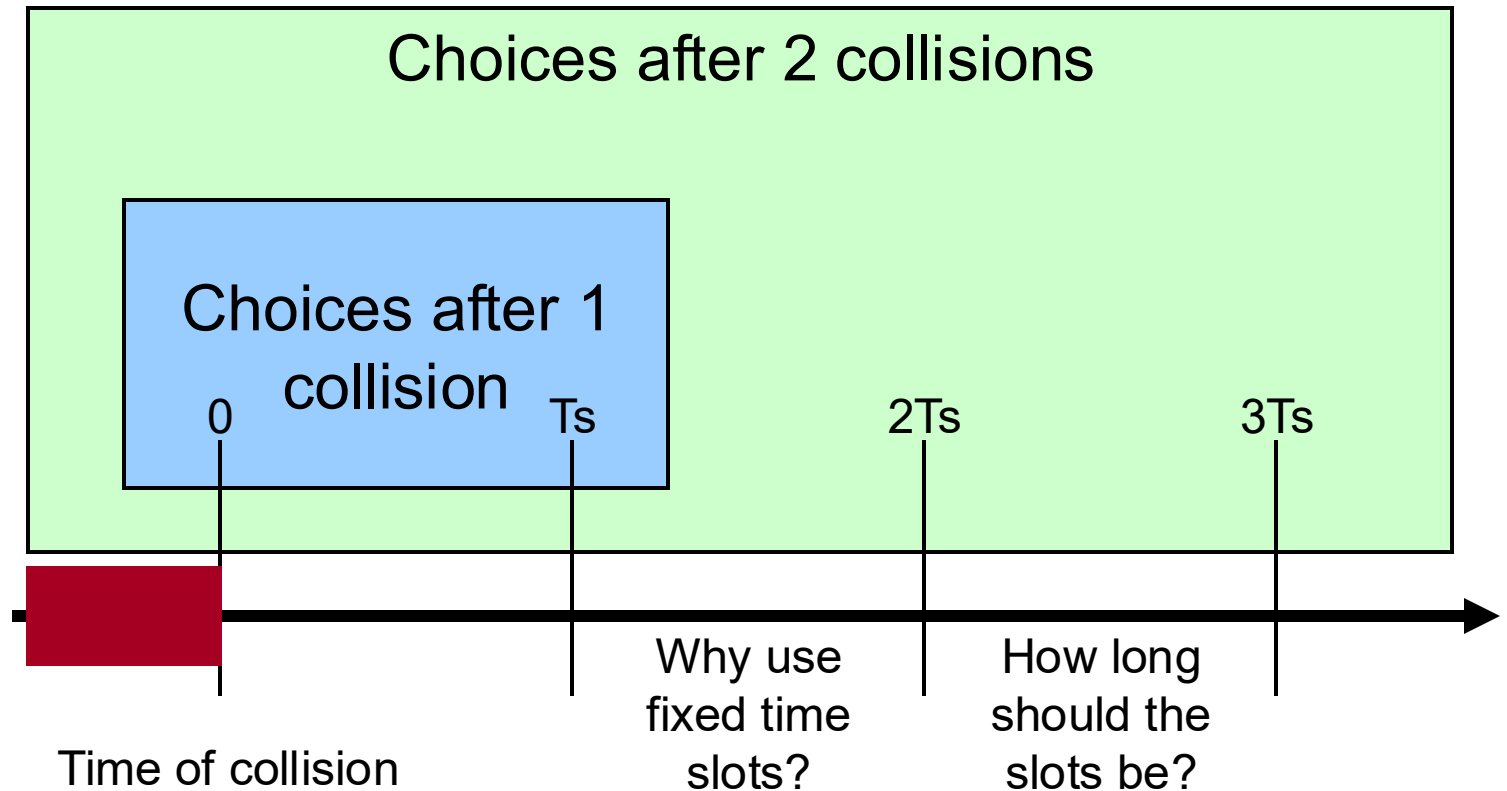
---

- ▶ How long should a host wait to retry after a collision?
- ▶ What happens if the host waits too long?
  - ▶ Wasted bandwidth
- ▶ What happens if the host doesn't wait long enough?
  - ▶ More collisions
- ▶ Ethernet Solution
  - ▶ Binary exponential backoff
    - ▶ Maximum backoff doubles with each failure
    - ▶ After N failures, pick an N-bit number
    - ▶  $2^N$  discrete possibilities from 0 to maximum



# Binary Exponential Backoff

---



# Binary Exponential Backoff

---

- ▶ For IEEE 802.3,  $T = 51.2 \mu\text{s}$
- ▶ Consider the following
  - ▶  $k$  hosts collide
  - ▶ Each picks a random number from 0 to  $2^{(N-1)}$
  - ▶ If the minimum value is unique
    - ▶ All other hosts see a busy line
    - ▶ Note: Ethernet RTT  $< 51.2 \mu\text{s}$
  - ▶ If the minimum value is not unique
    - ▶ Hosts with minimum value slot collide again!
    - ▶ Next slot is idle
    - ▶ Consider the next smallest backoff value



# Binary Exponential backoff algorithm

---

- ▶ When collision first occurs
  - ▶ Send a jamming signal to prevent further data being sent
- ▶ Resend a frame
  - ▶ After either  $0$  or  $T$  seconds, chosen at random
- ▶ If resend fails, resend the frame again
  - ▶ After either  $0$ ,  $T$ ,  $2T$ , or  $3T$  seconds.
  - ▶ In other words, send after  $kT$  seconds, where  $k$  is a random integer with  $0 \leq k < 2^2$
- ▶ If that still doesn't work, resend the frame again
  - ▶ After  $kT$ , where  $k$  is a random number with  $0 \leq k < 2^3$
- ▶ In general, after the  $n^{\text{th}}$  failed attempt, resend the frame after  $kT$ , where  $k$  is a random number and  $0 \leq k < 2^n$



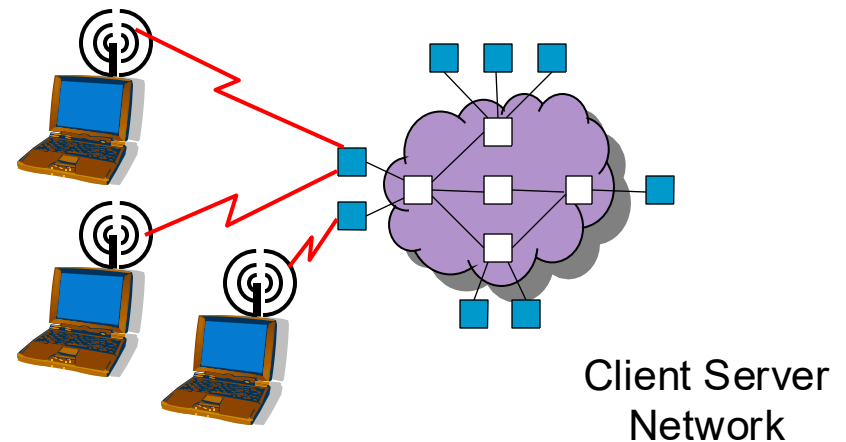
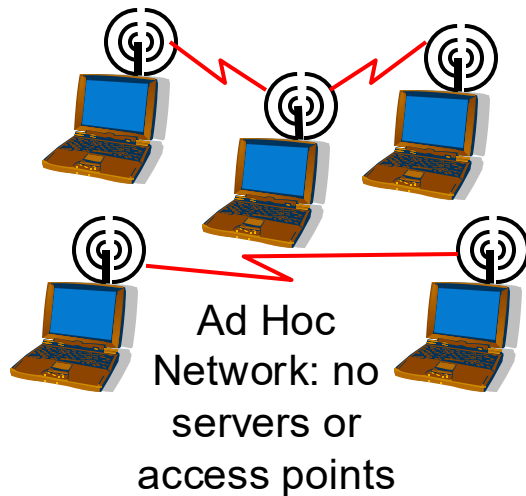


# Medium Access Control

---

## ▶ IEEE 802.11

- ▶ A physical and multiple access layer standard for wireless local area networks (WLAN)



# Medium Access Control

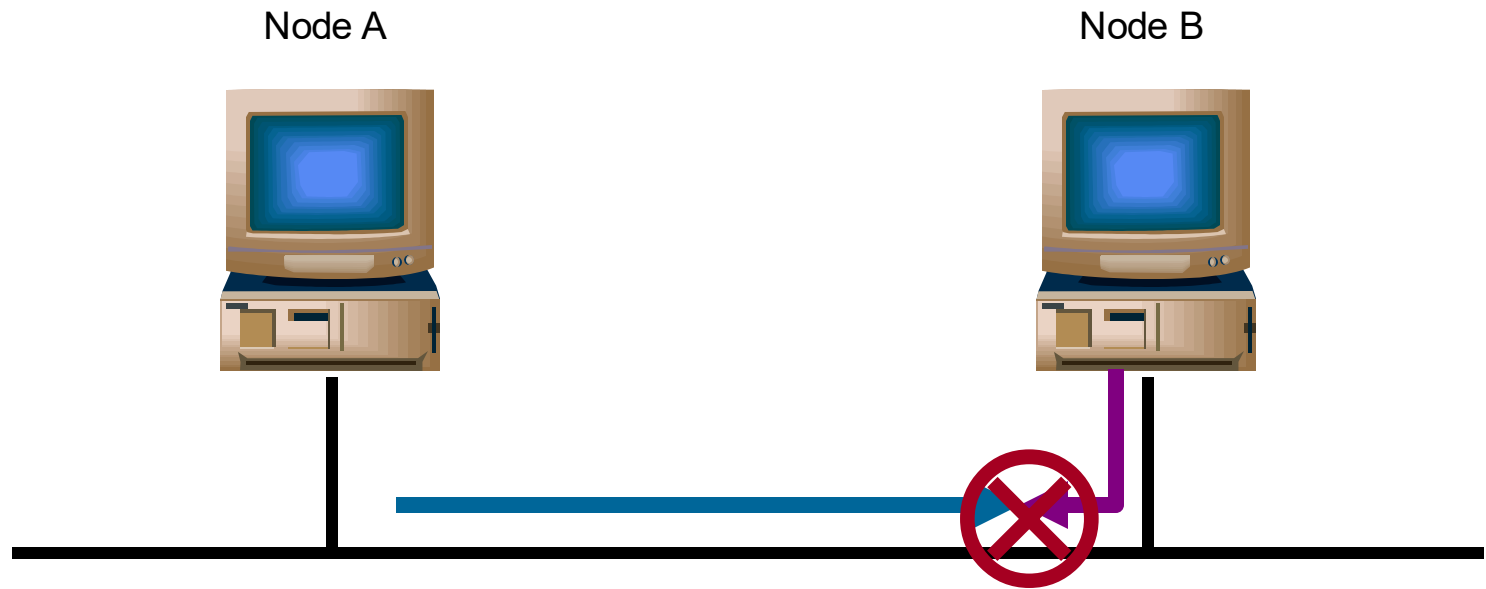
---

- ▶ Wireless channel is a shared medium
- ▶ Need access control mechanism to avoid interference
- ▶ Why not CSMA/CD?



# Ethernet MAC Algorithm

---

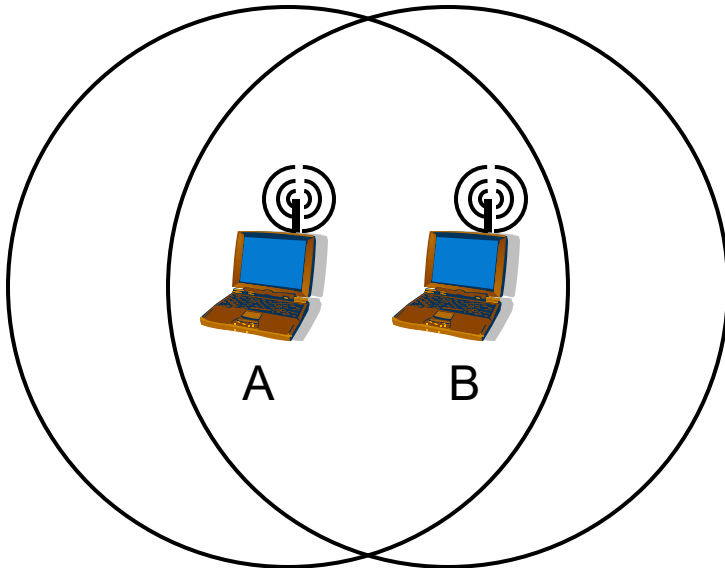


- Listen for carrier sense before transmitting
- Collision: What you hear is not what you sent!

# CSMA/CD in WLANs?

---

- ▶ Most radios are functionally half-duplex
  - ▶ Listening while transmitting is not possible
  - ▶ Ratio of transmitted signal power to received power is too high at the transmitter
  - ▶ Transmitter cannot detect competing transmitters (is deaf while transmitting)
- ▶ Collision might not occur at sender
  - ▶ Collision at receiver might not be detected by sender!



- ▶ Why do collisions happen?
  - ▶ Near simultaneous transmissions
    - ▶ Period of vulnerability: propagation delay

# Wireless Ethernet - CSMA/CA

---

- ▶ CS – Carrier Sense

- ▶ Nodes can distinguish between an idle and a busy link

- ▶ MA - Multiple Access

- ▶ A set of nodes send and receive frames over a shared link

- ▶ CD – Collision **Detection**

- ▶ Nodes listen during transmission to determine if there has been interference



# Wireless Ethernet - CSMA/CA

---

- ▶ CS – Carrier Sense

- ▶ Nodes can distinguish between an idle and a busy link

- ▶ MA - Multiple Access

- ▶ A set of nodes send and receive frames over a shared link

- ▶ CA – Collision Avoidance

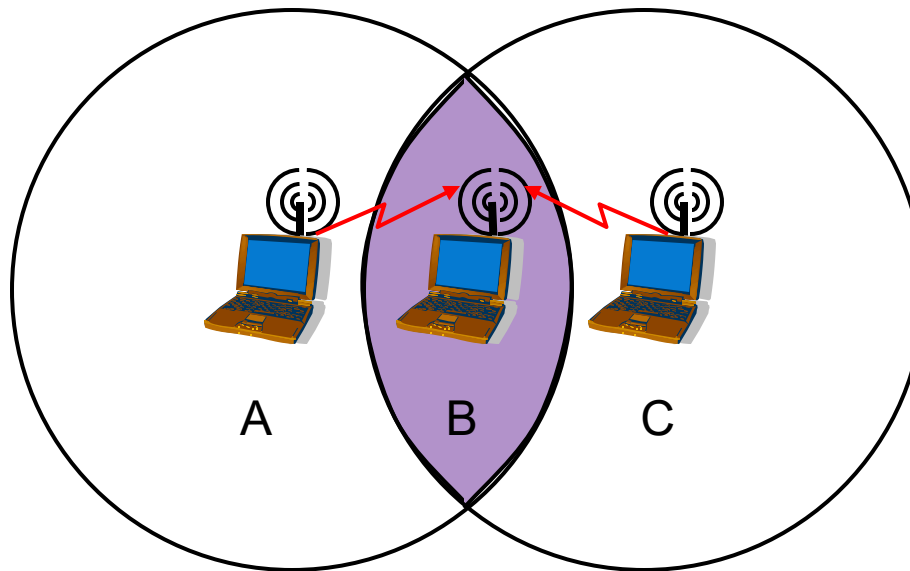
- ▶ Nodes use protocol to prevent collisions from occurring



# IEEE 802.11 MAC Layer Standard

---

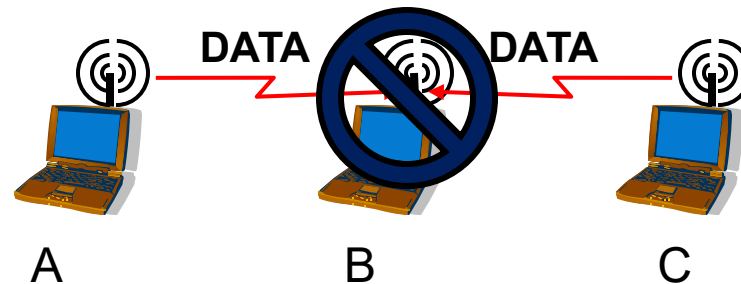
- ▶ Similar to Ethernet
- ▶ But consider the following:



# Hidden Terminal Problem

---

- ▶ Node B can communicate with both A and C
- ▶ A and C cannot hear each other
- ▶ When A transmits to B, C cannot detect the transmission using the carrier sense mechanism
- ▶ If C transmits, collision will occur at node B

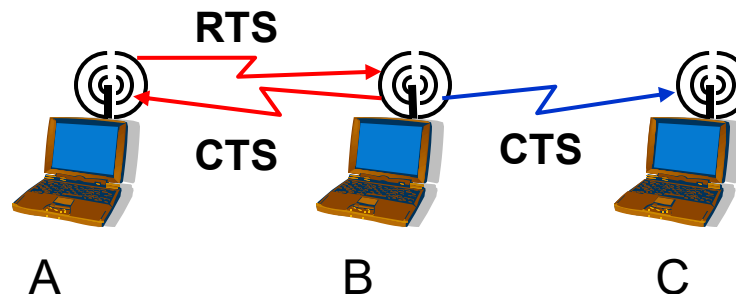




# MACA Solution for Hidden Terminal Problem

---

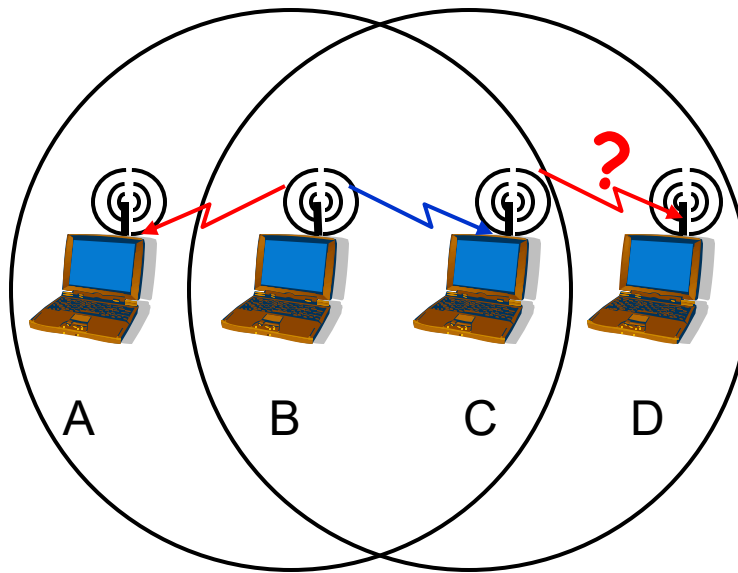
- ▶ When node A wants to send a packet to node B
  - ▶ Node A first sends a Request-to-Send (RTS) to B
- ▶ On receiving RTS
  - ▶ Node B responds by sending Clear-to-Send (CTS) to A
  - ▶ provided node B is able to receive the packet
- ▶ When a node C overhears a CTS, it keeps quiet for the duration of the transfer



# IEEE 802.11 MAC Layer Standard

---

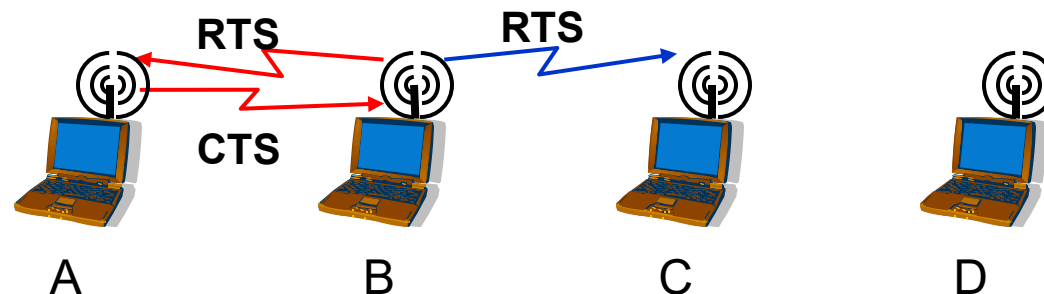
- ▶ But we still have a problem



# Exposed Terminal Problem

---

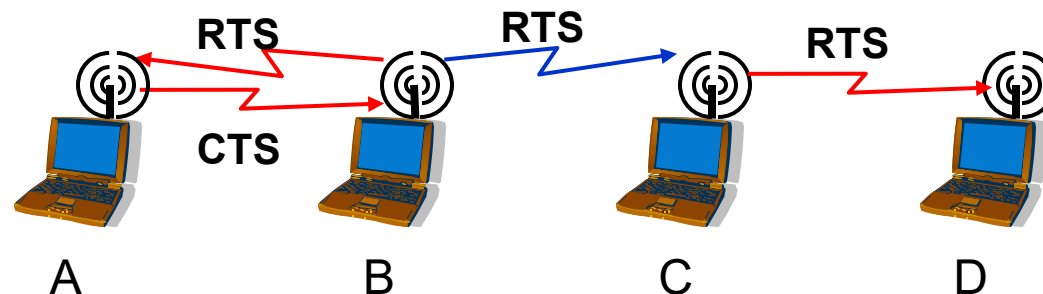
- ▶ B talks to A
- ▶ C wants to talk to D
- ▶ C senses channel and finds it to be busy
- ▶ C stays quiet (when it could have ideally transmitted)



# MACA Solution for Exposed Terminal Problem

---

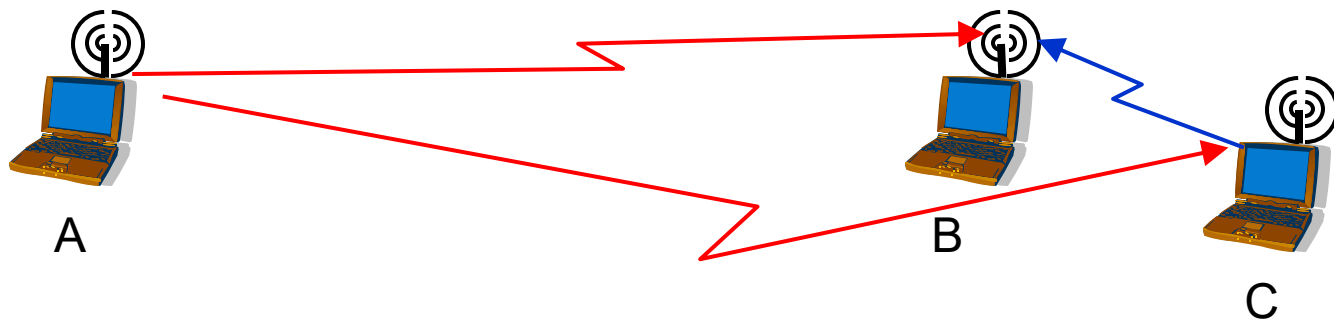
- ▶ Sender transmits Request to Send (RTS)
- ▶ Receiver replies with Clear to Send (CTS)
- ▶ Neighbors
  - ▶ See CTS - Stay quiet
  - ▶ See RTS, but no CTS - OK to transmit



# Capture Effect

---

- ▶ C will almost always “win” if there is a collision at B
  - ▶ Can lead to extreme unfairness and even starvation
- ▶ Solution is power control
  - ▶ Very difficult to manage in a non-provisioned environment!



# IEEE 802.11 MAC Layer Standard

---

- ▶ **MACAW – Multiple Access with Collision Avoidance for Wireless**
  - ▶ Sender transmits Request to Send (RTS)
  - ▶ Receiver replies with Clear to Send (CTS)
  - ▶ Neighbors
    - ▶ See CTS
      - Stay quiet
    - ▶ See RTS, but no CTS
      - OK to transmit
  - ▶ Receiver sends ACK for frame
    - ▶ Neighbors stay silent until they hear ACK



# Collisions

---

- ▶ Still possible
  - ▶ RTS packets can collide!
- ▶ Binary exponential backoff
  - ▶ Backoff counter doubles after every collision and reset to minimum value after successful transmission
  - ▶ Performed by stations that experience RTS collisions
- ▶ RTS collisions not as bad as data collisions in CSMA
  - ▶ Since RTS packets are typically much smaller than DATA packets



# Reliability

---

- ▶ **Wireless links are prone to errors**
  - ▶ High packet loss rate detrimental to transport-layer performance
- ▶ **Mechanisms needed to reduce packet loss rate experienced by upper layers**

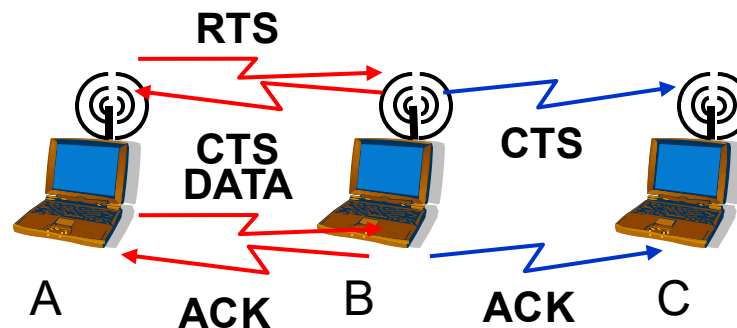




# A Simple Solution to Improve Reliability - MACAW

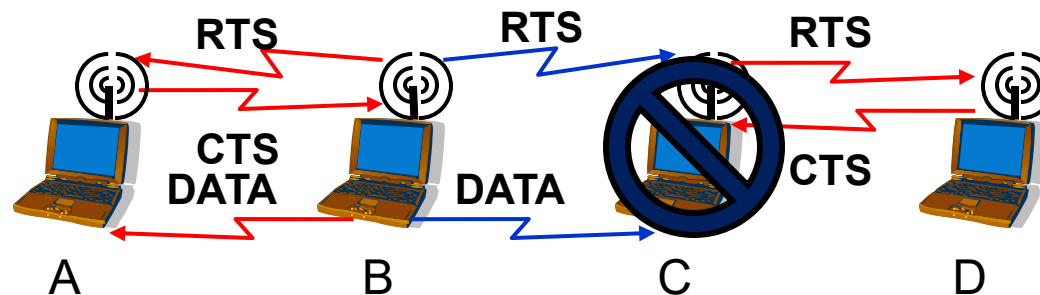
---

- ▶ When node B receives a data packet from node A, node B sends an Acknowledgement (ACK)
- ▶ If node A fails to receive an ACK
  - ▶ Retransmit the packet



# Revisiting the Exposed Terminal Problem

- ▶ Problem
  - ▶ Exposed terminal solution doesn't consider CTS at node C
- ▶ With RTS-CTS, C doesn't wait since it doesn't hear A's CTS
  - ▶ With B transmitting DATA, C can't hear intended receiver's CTS
  - ▶ C trying RTS while B is transmitting is useless



# Revisiting the Exposed Terminal Problem - MACAW

---

- ▶ **One solution**

- ▶ Have C use carrier sense before RTS

- ▶ **Alternative**

- ▶ B sends DS (data sending) packet before DATA
    - ▶ Short packet lets C know that B received A's CTS
    - ▶ Includes length of B's DATA so C knows how long to wait



# Backoff Algorithm

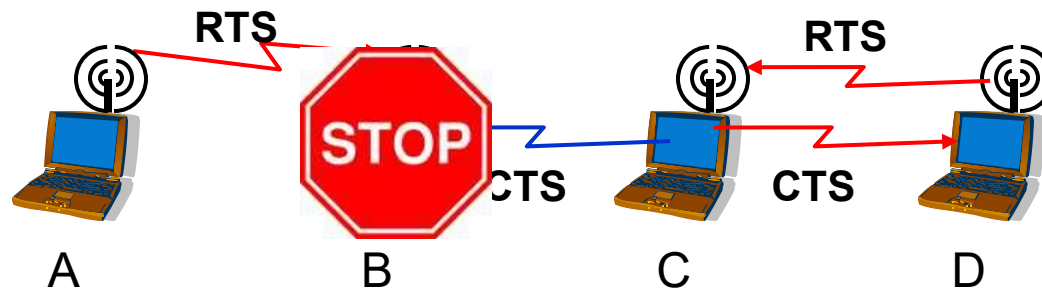
---

- ▶ Binary exponential backoff (BEB)
  - ▶ Backoff counter doubles after every collision and reset to minimum value after successful transmission
- ▶ Unfair channel allocation!
  - ▶ Successful transmitters reset backoff counter to minimum value
    - ▶ It is more likely that successful transmitters continue to be successful
  - ▶ If there is no maximum backoff
    - ▶ One station can get the entire channel bandwidth
- ▶ Ideally
  - ▶ The backoff counter should reflect the ambient congestion level which is the same for all stations involved!



# Deafness

- ▶ For the scenario below
  - ▶ Node A sends an RTS to B
    - ▶ While node C is receiving from D,
  - ▶ Node B cannot reply with a CTS
    - ▶ B knows that D is sending to C
    - ▶ A keeps retransmitting RTS and increasing its own BO timeout



# Revisiting the Exposed Terminal Problem - MACAW

---

- ▶ **One solution**

- ▶ Have C use carrier sense before RTS

- ▶ **Alternative**

- ▶ B sends DS (data sending) packet before DATA
    - ▶ Short packet lets C know that B received A's CTS
    - ▶ Includes length of B's DATA so C knows how long to wait



# Backoff Algorithm

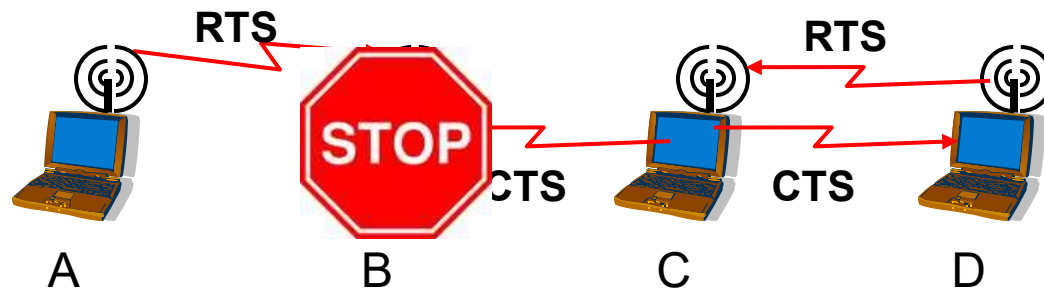
---

- ▶ Binary exponential backoff (BEB)
  - ▶ Backoff counter doubles after every collision and reset to minimum value after successful transmission
- ▶ Unfair channel allocation!
  - ▶ Successful transmitters reset backoff counter to minimum value
    - ▶ It is more likely that successful transmitters continue to be successful
  - ▶ If there is no maximum backoff
    - ▶ One station can get the entire channel bandwidth
- ▶ Ideally
  - ▶ The backoff counter should reflect the ambient congestion level which is the same for all stations involved!



# Deafness

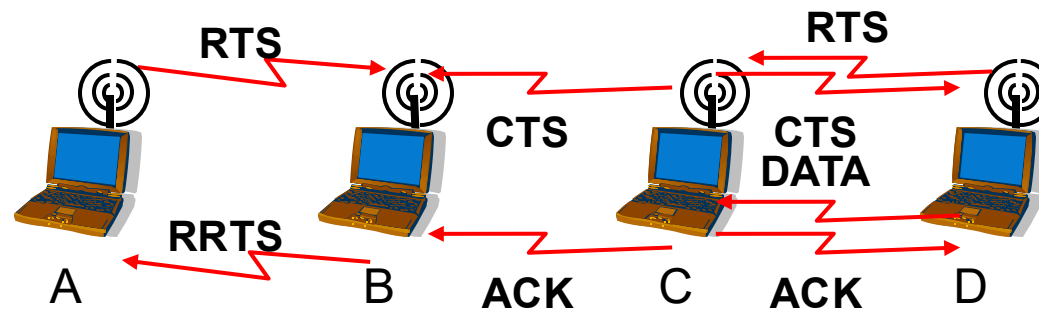
- ▶ For the scenario below
  - ▶ Node A sends an RTS to B
    - ▶ While node C is receiving from D,
  - ▶ Node B cannot reply with a CTS
    - ▶ B knows that D is sending to C
    - ▶ A keeps retransmitting RTS and increasing its own BO timeout





# Request for RTS - MACAW

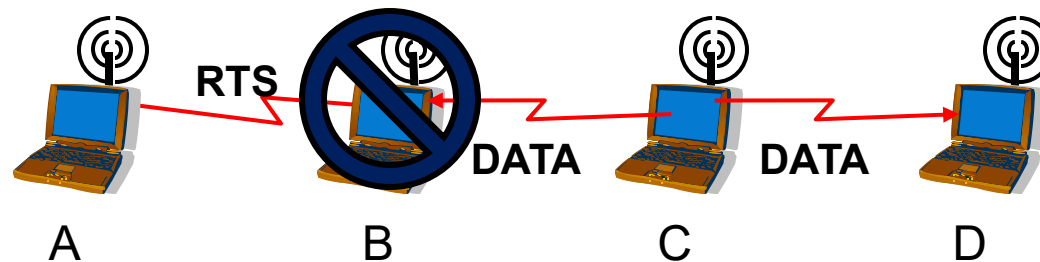
- ▶ Have B do contention on behalf of A
  - ▶ If B receives RTS for which it must defer CTS reply
  - ▶ Then B later sends RRTS to A when it can send
  - ▶ A responds by starting normal RTS-CTS
  - ▶ Others hearing RRTS defer long enough for RTS-CTS



# Another MACAW Proposal

---

- ▶ This approach, however, does not work in the scenario below
- ▶ Node B may not receive the RTS from A at all, due to interference with transmission from C



# Broadcast/Multicast

---

- ▶ **Problem**

- ▶ Basic RTS-CTS only works for unicast transmissions

- ▶ **For multicast**

- ▶ RTS would get CTS from each intended receiver
  - ▶ Likely to cause (many) collisions back at sender



# Multicast - MACAW

---

- ▶ **Sort-of solution**
  - ▶ Don't use CTS for multicast data
- ▶ **Receivers recognize multicast destination in RTS**
  - ▶ Don't return CTS
  - ▶ Sender follows RTS immediately by DATA
  - ▶ After RTS, all receivers defer for long enough for DATA
- ▶ **Helps, but doesn't fully solve problem**
  - ▶ Like normal CSMA, only those in range of sender will defer
  - ▶ Others in range of receiver will not defer



# IEEE 802.11

---

- ▶ MAC functionality
  - ▶ Addressing
  - ▶ CSMA/CA
- ▶ Error detection (FCS)
- ▶ Error correction (ACK frame)
- ▶ Flow control: stop-and-wait
- ▶ Fragmentation (More Frag)
- ▶ Collision Avoidance (RTS-CTS)



# IEEE 802.11 Wireless MAC

---

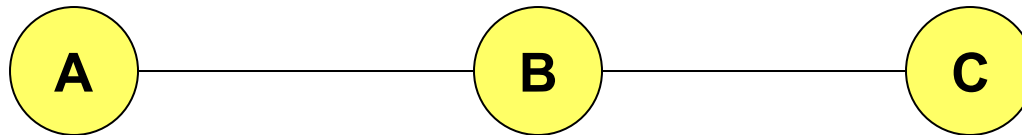
- ▶ Distributed and centralized MAC components
  - ▶ Distributed Coordination Function (DCF)
  - ▶ Point Coordination Function (PCF)
- ▶ DCF suitable for multi-hop ad hoc networking
- ▶ DCF is a Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) protocol



# IEEE 802.11 DCF

---

- ▶ Uses RTS-CTS exchange to avoid hidden terminal problem
  - ▶ Any node overhearing a CTS cannot transmit for the duration of the transfer
- ▶ Uses ACK to achieve reliability
- ▶ Any node receiving the RTS cannot transmit for the duration of the transfer
  - ▶ To prevent collision with ACK when it arrives at the sender
  - ▶ When B is sending data to C, node A keeps quite



# IEEE 802.11 CSMA/CA

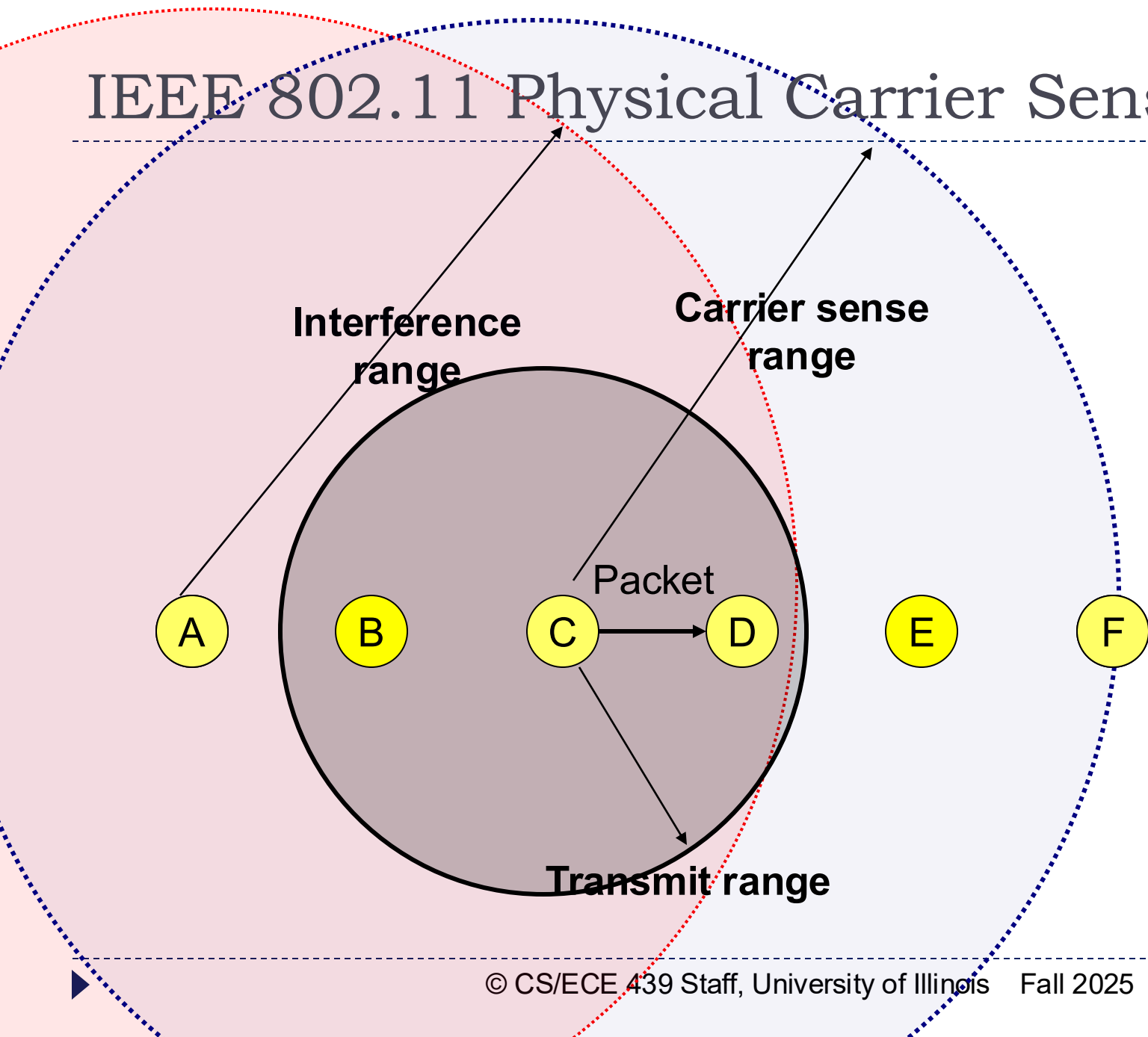
---

- ▶ Nodes stay silent when carrier sensed
  - ▶ Physical carrier sense
  - ▶ Virtual carrier sense
    - ▶ Network Allocation Vector (NAV)
    - ▶ NAV is updated based on overheard RTS/CTS/DATA/ACK packets, each of which specified duration of a pending transmission
- ▶ Backoff intervals used to reduce collision probability



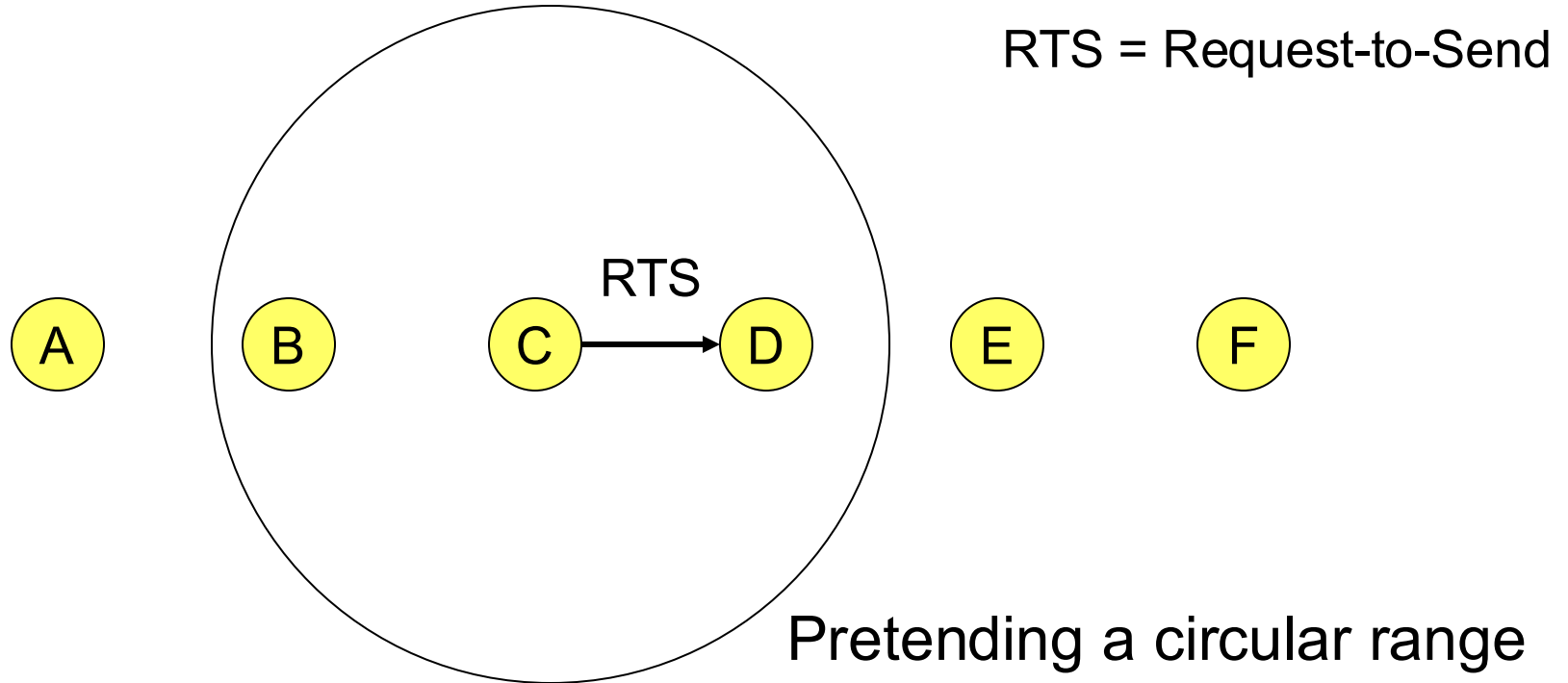


# IEEE 802.11 Physical Carrier Sense



# IEEE 802.11 Virtual Carrier Sense

---

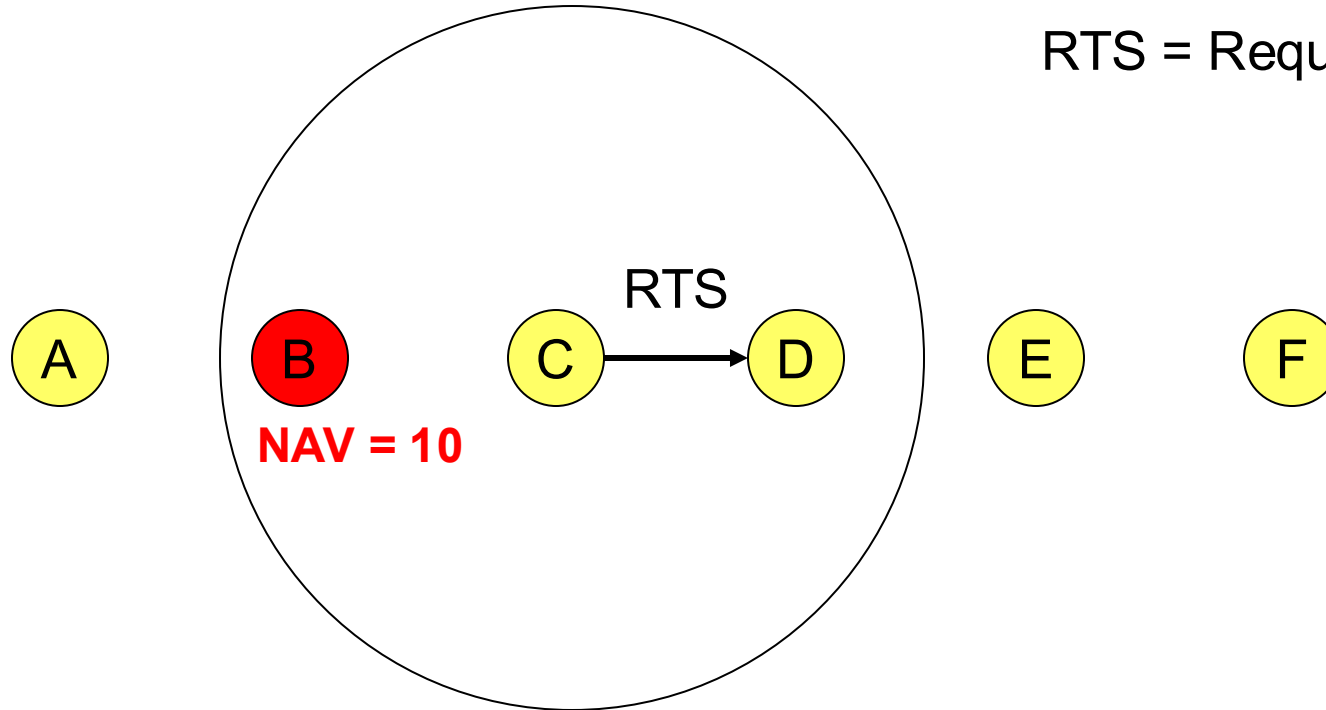


# IEEE 802.11 Virtual Carrier Sense

---

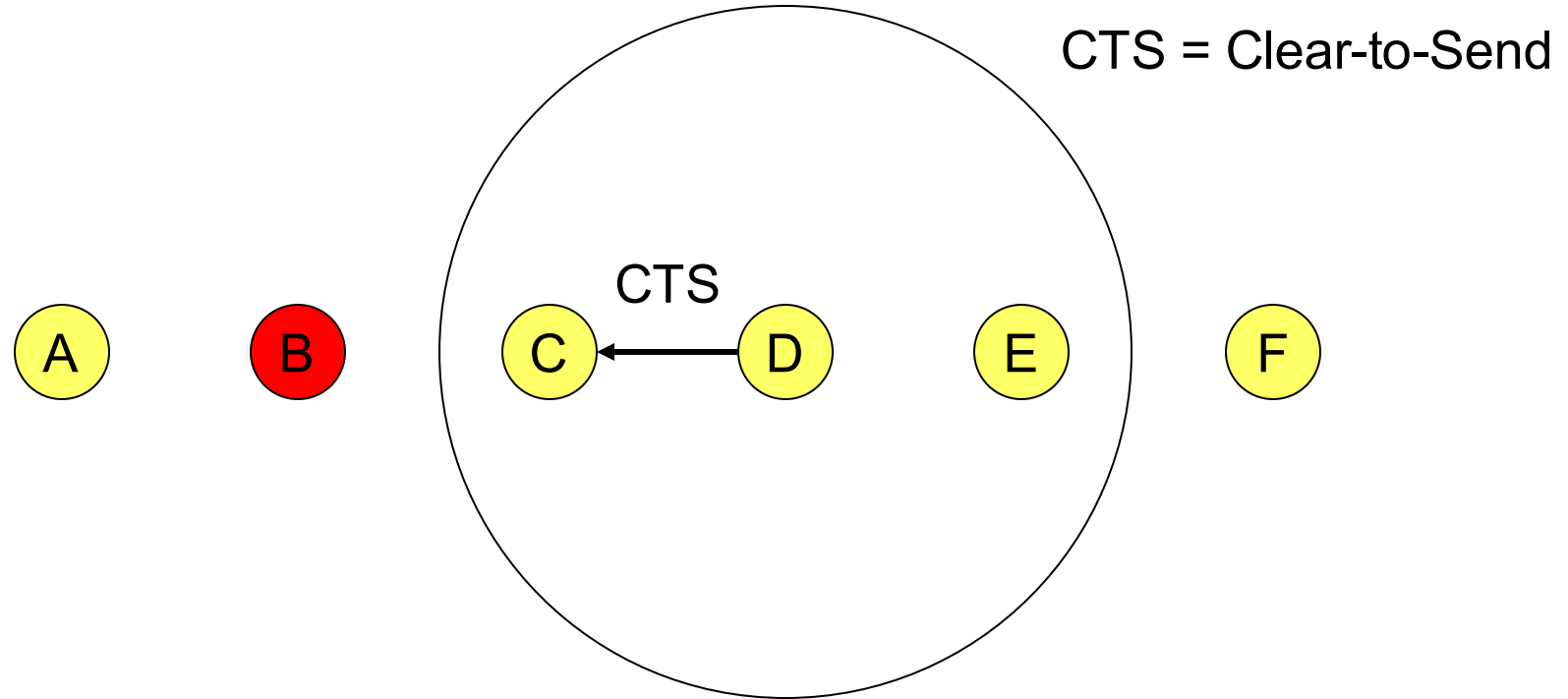
**NAV** = remaining duration to keep quiet

RTS = Request-to-Send



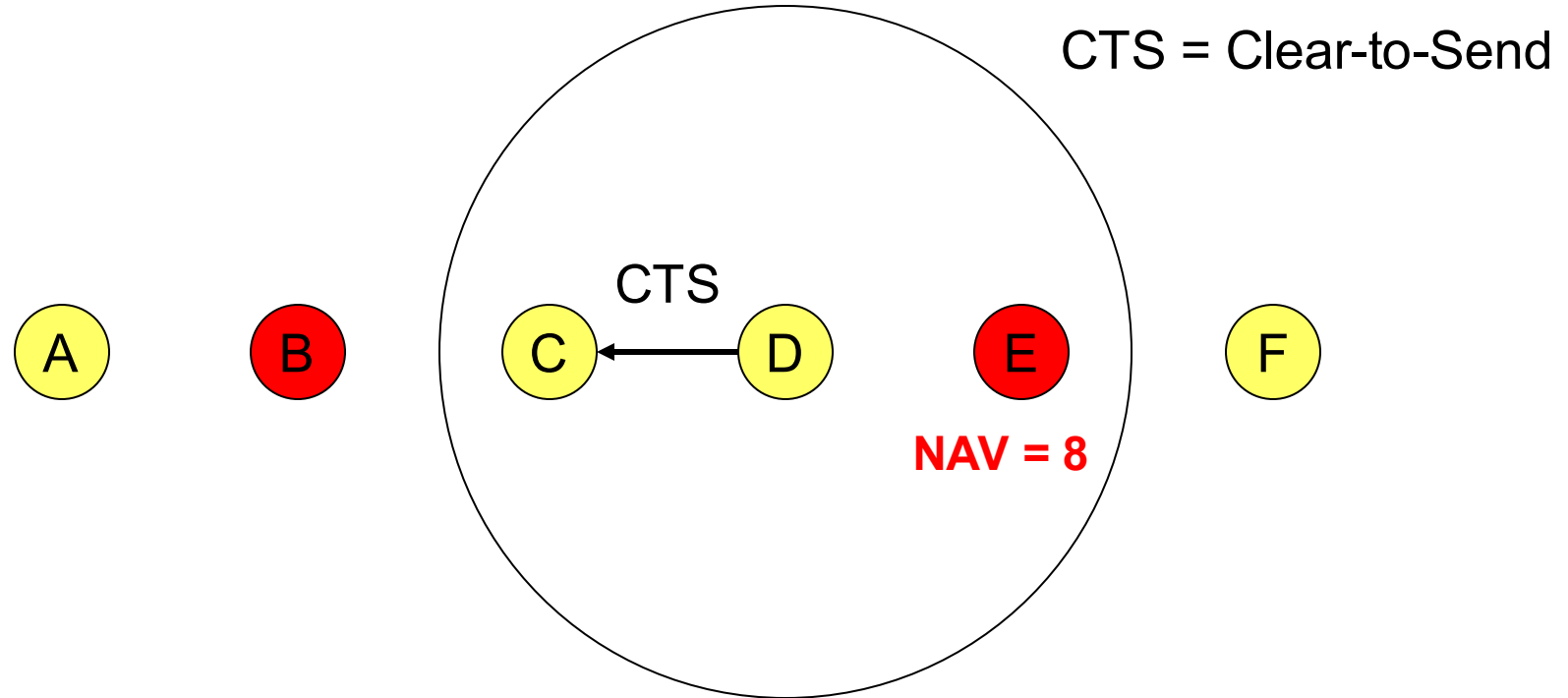
# IEEE 802.11 Virtual Carrier Sense

---



# IEEE 802.11 Virtual Carrier Sense

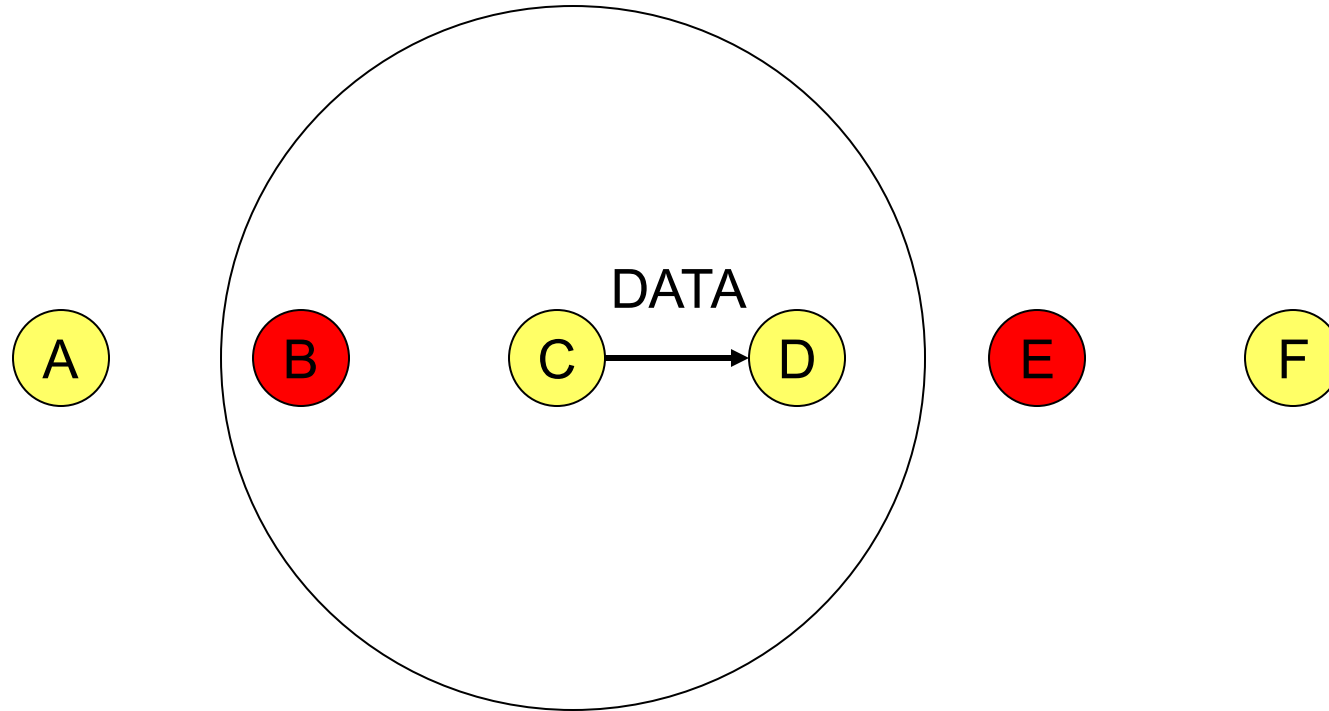
---



# IEEE 802.11 Virtual Carrier Sense

---

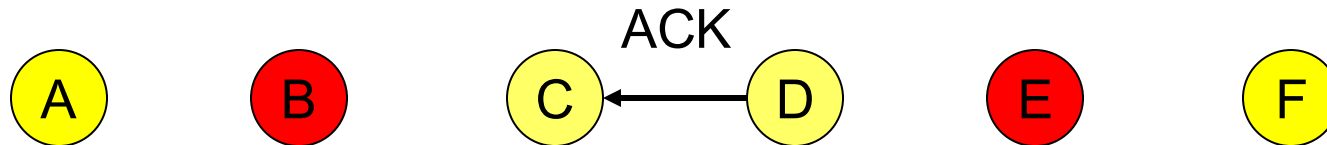
- ▶ DATA packet follows CTS



# IEEE 802.11 Virtual Carrier Sense

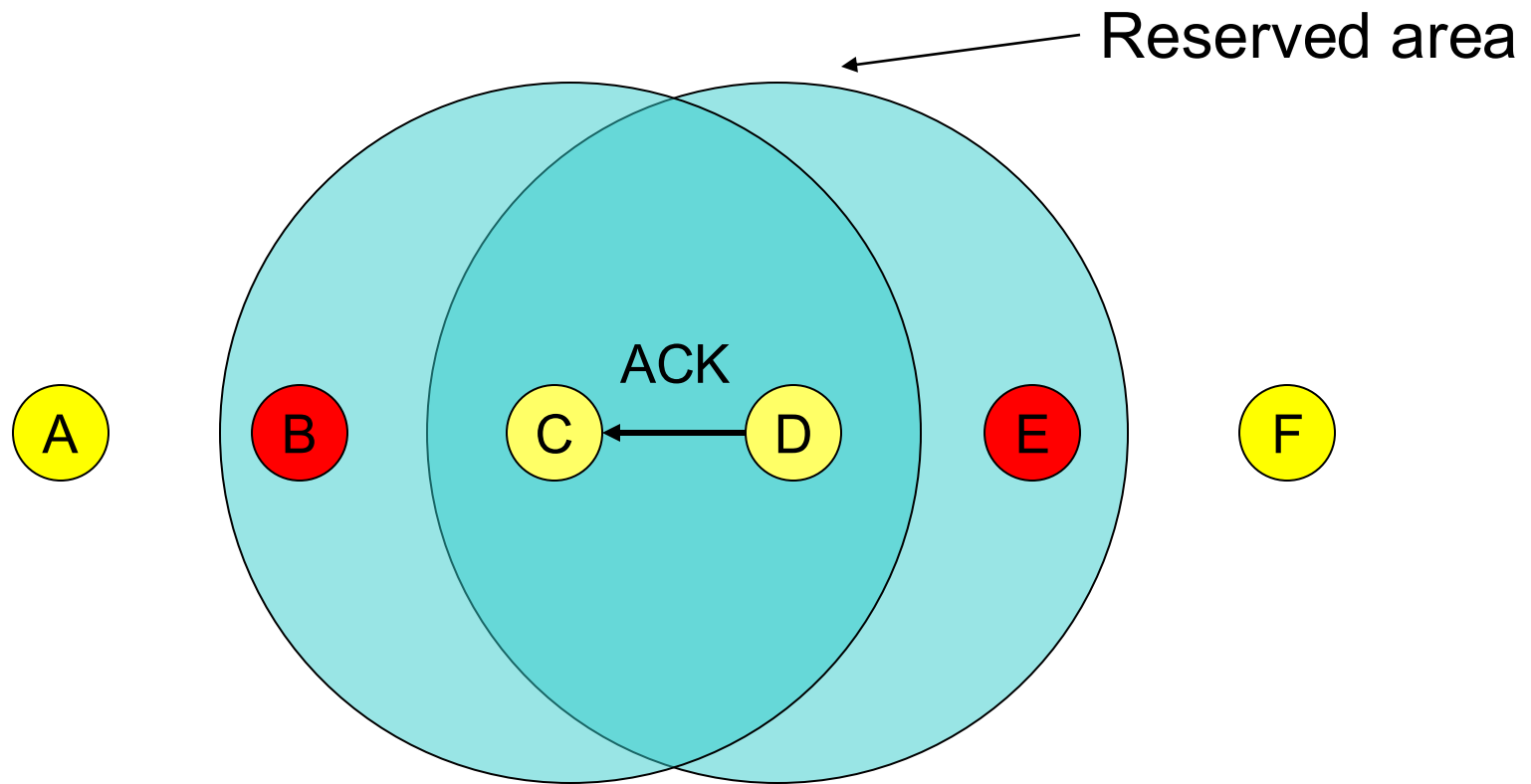
---

- ▶ Successful data reception acknowledged using ACK



# IEEE 802.11

---





# More features

---

- ▶ Use of RTS/CTS is controlled by an RTS threshold
  - ▶ Only used for data packets  $>$  threshold
  - ▶ Pointless to use RTS/CTS for short data packets
    - ▶ High overhead!
- ▶ Number of retries is limited by a Retry Counter
  - ▶ Short retry counter
    - ▶ For packets shorter than RTS threshold
  - ▶ Long retry counter
    - ▶ For packets longer than RTS threshold
- ▶ Packets can be fragmented.
  - ▶ Each fragment is acknowledged
  - ▶ But all fragments are sent in one sequence
  - ▶ Sending shorter frames can reduce impact of bit errors
  - ▶ Lifetime timer: maximum time for all fragments of frame



# Ethernet vs. IEEE 802.11

---

## ▶ If carrier is sensed

- ▶ Send immediately
- ▶ Send maximum of 1500B data (1527B total)
- ▶ Wait 9.6  $\mu$ s before sending again

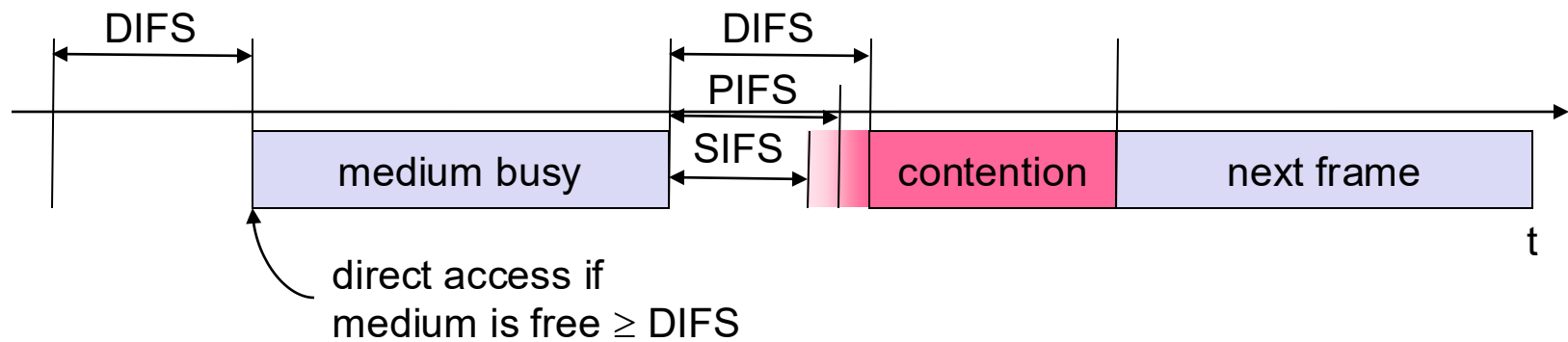
## ▶ If carrier is sensed

- ▶ When should a node transmit?



# Interframe Spacing

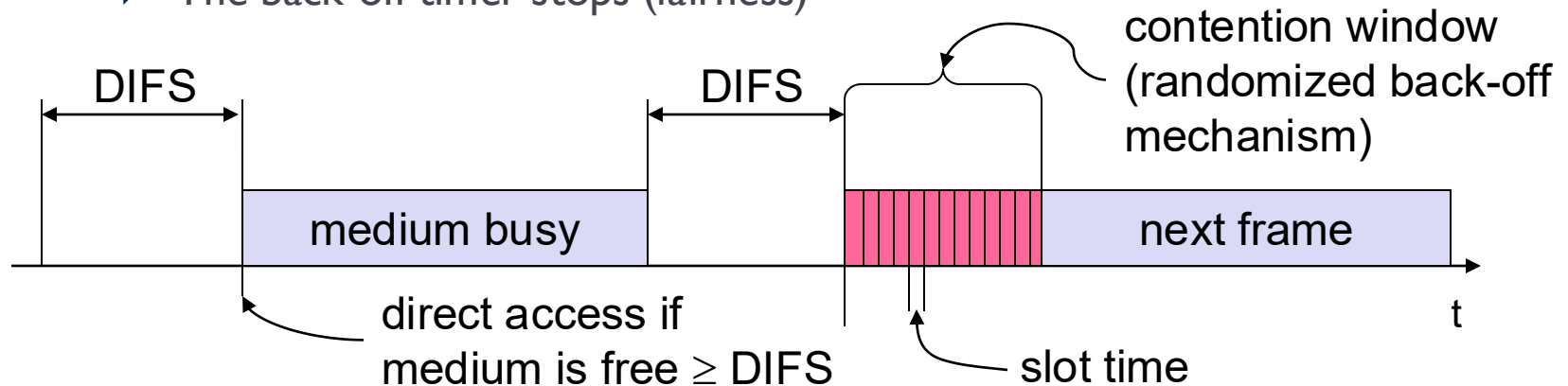
- ▶ Interframe spacing
  - ▶ Plays a large role in coordinating access to the transmission medium
- ▶ Varying interframe spacings
  - ▶ Creates different priority levels for different types of traffic!
- ▶ 802.11 uses 4 different interframe spacings



# IEEE 802.11 - CSMA/CA

---

- ▶ Sensing the medium
- ▶ If free for an Inter-Frame Space (IFS)
  - ▶ Station can start sending (IFS depends on service type)
- ▶ If busy
  - ▶ Station waits for a free IFS, then waits a random back-off time (collision avoidance, multiple of slot-time)
- ▶ If another station transmits during back-off time
  - ▶ The back-off timer stops (fairness)



# Types of IFS

---

- ▶ **SIFS**

- ▶ Short interframe space
- ▶ Used for highest priority transmissions
- ▶ RTS/CTS frames and ACKs

- ▶ **DIFS**

- ▶ DCF interframe space
- ▶ Minimum idle time for contention-based services (> SIFS)

# Types of IFS

---

- ▶ **PIFS**

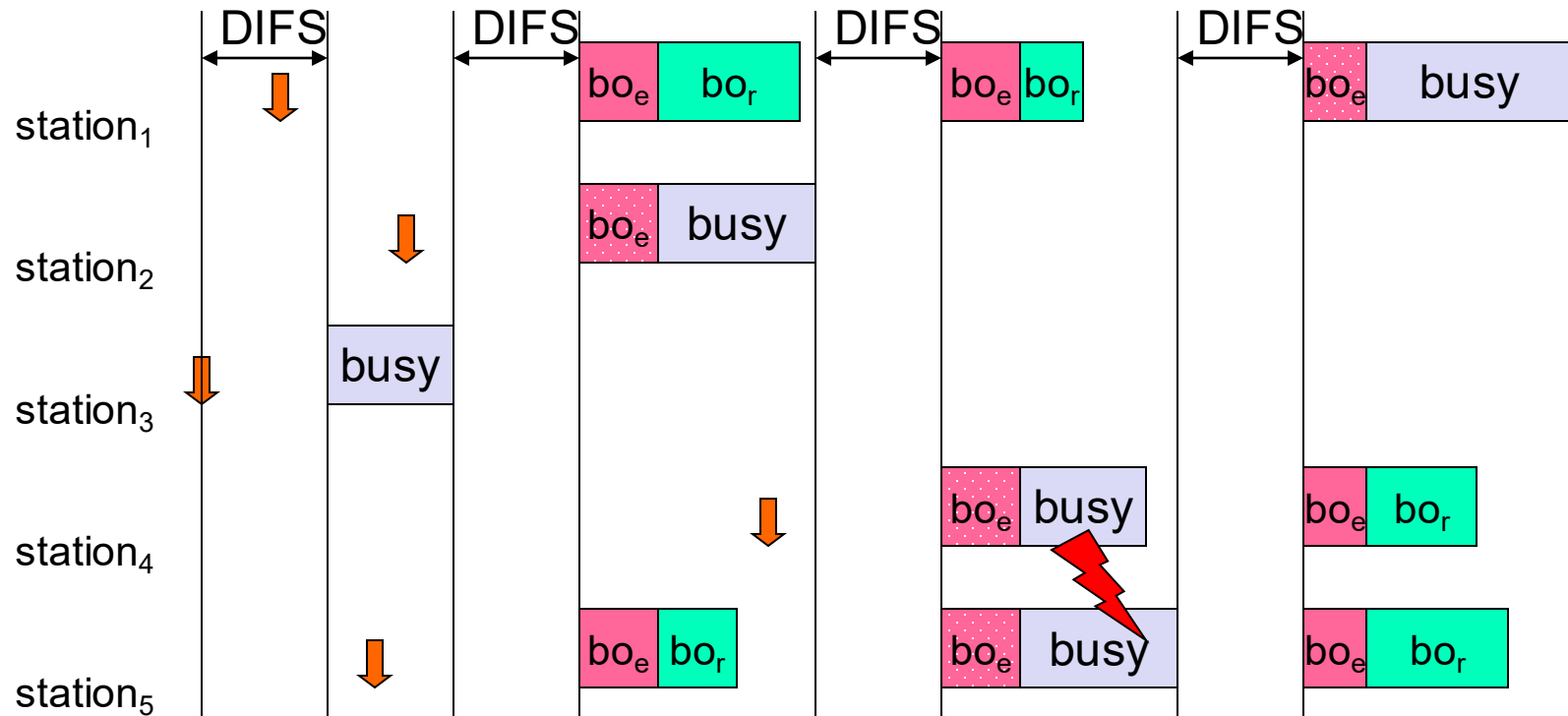
- ▶ PCF interframe space
- ▶ Minimum idle time for contention-free service ( $>\text{SIFS}$ ,  $<\text{DIFS}$ )

- ▶ **EIFS**

- ▶ Extended interframe space
- ▶ Used when there is an error in transmission



# IEEE 802.11 - Competing Stations



# Backoff Interval

---

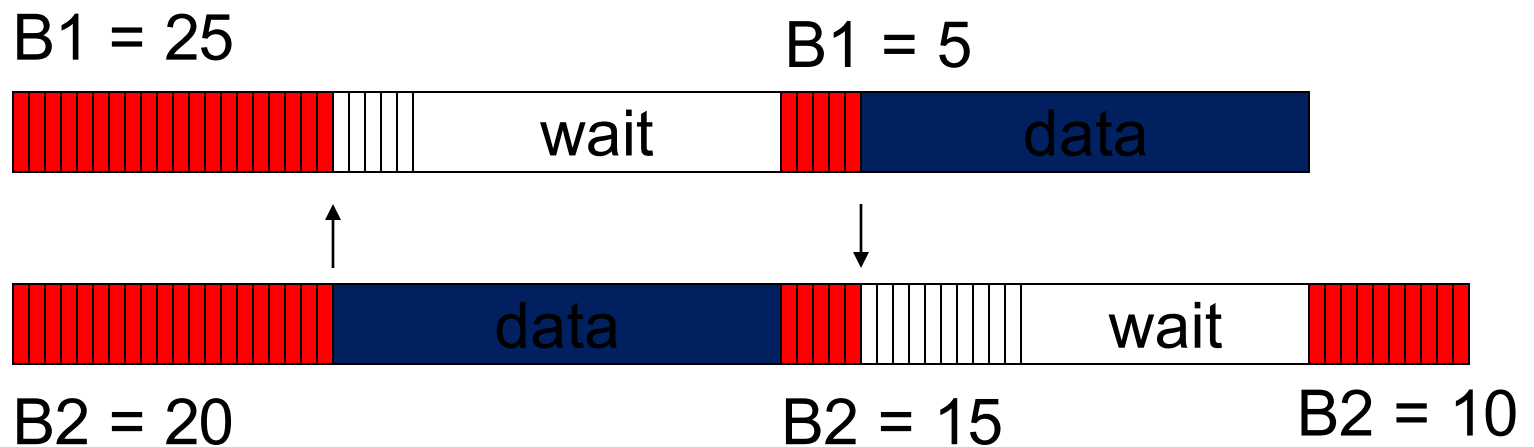
- ▶ When transmitting a packet, choose a backoff interval in the range  $[0, CW]$ 
  - ▶  $CW$  is contention window
- ▶ Count down the backoff interval when medium is idle
  - ▶ Count-down is suspended if medium becomes busy
- ▶ When backoff interval reaches 0, transmit RTS





# DCF Example

---



CW = 31

B1 and B2 are backoff intervals  
at nodes 1 and 2

# Backoff Interval

---

- ▶ The time spent counting down backoff intervals is a part of MAC overhead
- ▶ Large CW
  - ▶ Large backoff intervals
  - ▶ Can result in larger overhead
- ▶ Small CW
  - ▶ Larger number of collisions (when two nodes count down to 0 simultaneously)



# Backoff Interval

---

- ▶ The number of nodes attempting to transmit simultaneously may change with time
  - ▶ Some mechanism to manage contention is needed
- ▶ IEEE 802.11 DCF
  - ▶ Contention window CW is chosen dynamically depending on collision occurrence



# Binary Exponential Backoff in DCF

---

- ▶ When a node fails to receive CTS in response to its RTS, it increases the contention window
  - ▶ cw is doubled (up to an upper bound)
- ▶ When a node successfully completes a data transfer, it restores cw to  $CW_{\min}$ 
  - ▶ cw follows a sawtooth curve



# IEEE 802.11 Frame Format

---

## ▶ Types

- ▶ control frames, management frames, data frames

## ▶ Sequence numbers

- ▶ important against duplicated frames due to lost ACKs

## ▶ Addresses

- ▶ receiver, transmitter (physical), BSS identifier, sender (logical)

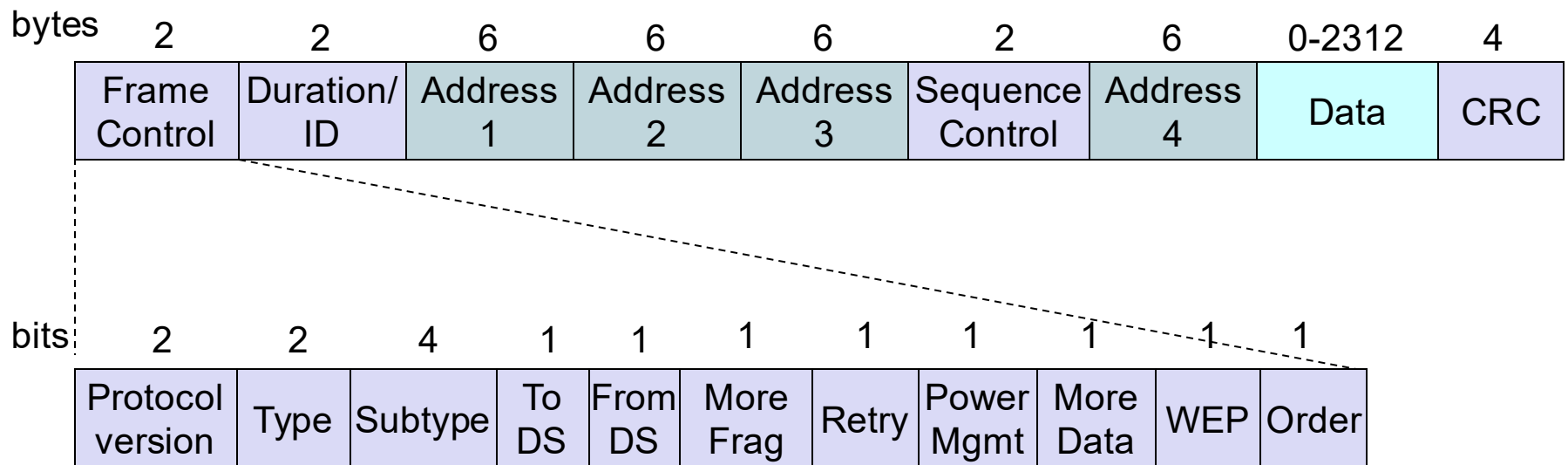
## ▶ Miscellaneous

- ▶ sending time, checksum, frame control, data



# IEEE 802.11 Data Frame Format

---

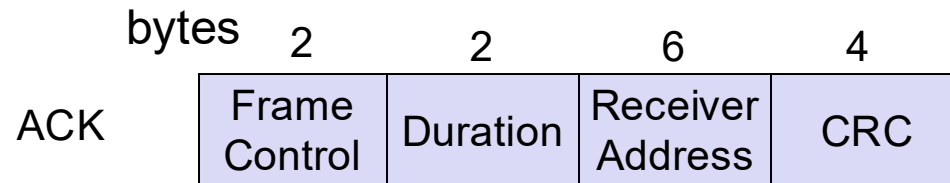


# IEEE 802.11 Control Frame Format

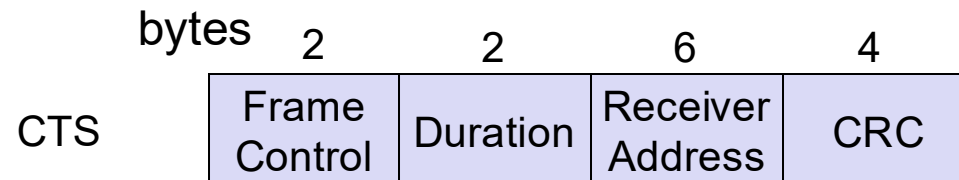
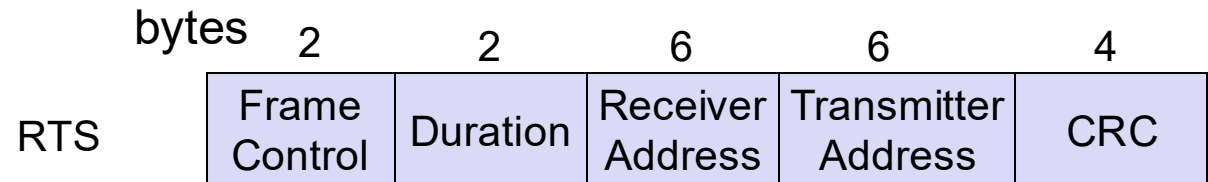
---

- ▶ Acknowledgement

- ▶ Request To Send



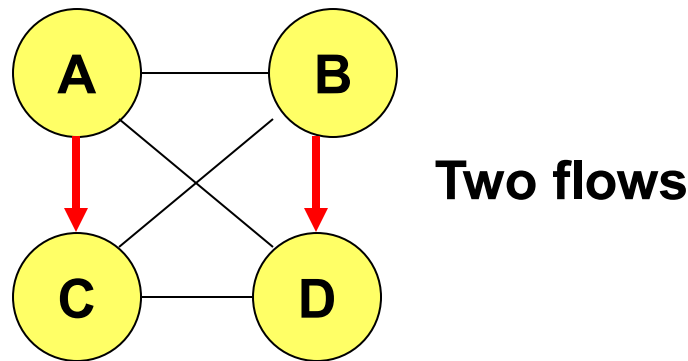
- ▶ Clear To Send



# Fairness Issue

---

- ▶ Many definitions of fairness plausible
- ▶ Simplest definition
  - ▶ All nodes should receive equal bandwidth

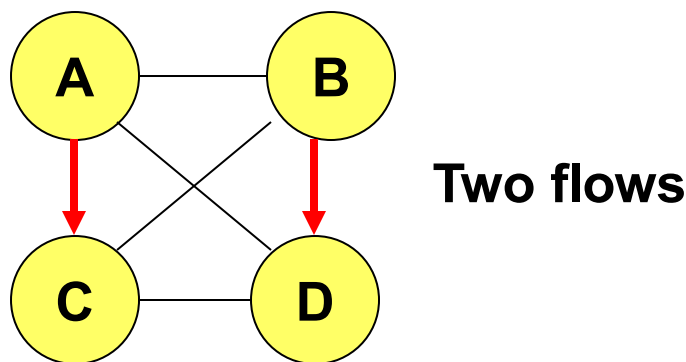




# Fairness Issue

---

- ▶ Assume that initially, A and B both choose a backoff interval in range  $[0, 31]$  but their RTSs collide
- ▶ Nodes A and B then choose from range  $[0, 63]$ 
  - ▶ Node A chooses 4 slots and B chooses 60 slots
  - ▶ After A transmits a packet, it next chooses from range  $[0, 31]$
  - ▶ It is possible that A may transmit several packets before B transmits its first packet



# Fairness Issue

---

- ▶ **Unfairness**

- ▶ When one node has backed off much more than some other node

- ▶ **MACAW Solution**

- ▶ When a node transmits a packet
    - ▶ Append the CW value to the packet
    - ▶ All nodes hearing that CW value use it for their future transmission attempts
  - ▶ CW is an indication of the level of congestion in the vicinity of a specific receiver node
    - ▶ MACAW proposes maintaining CW independently for each receiver
  - ▶ Per-receiver CW is particularly useful in multi-hop environments
    - ▶ Congestion level at different receivers can be very different

# IEEE 802.11 Amendments

---

- ▶ **IEEE 802.11-1997:**
  - ▶ Originally 1 Mbit/s and 2 Mbit/s
  - ▶ 2.4 GHz RF and infrared (IR)
- ▶ **IEEE 802.11a:**
  - ▶ 54 Mbit/s, 5 GHz standard (2001)
- ▶ **IEEE 802.11b:**
  - ▶ Enhancements to support 5.5 and 11 Mbit/s (1999)
- ▶ **IEEE 802.11c:**
  - ▶ Bridge operation procedures;
  - ▶ Included in the IEEE 802.1D standard (2001)
- ▶ **IEEE 802.11d:**
  - ▶ International (country-to-country) roaming extensions (2001)
- ▶ **IEEE 802.11e:**
  - ▶ Enhancements: QoS, including packet bursting (2005)
- ▶ **IEEE 802.11g:**
  - ▶ 54 Mbit/s, 2.4 GHz standard (backwards compatible with b) (2003)
- ▶ **IEEE 802.11h:**
  - ▶ Spectrum Managed 802.11a (5 GHz) for European compatibility (2004)
- ▶ **IEEE 802.11i:**
  - ▶ Enhanced security (2004)
- ▶ **IEEE 802.11j:**
  - ▶ Extensions for Japan (2004)
- ▶ **IEEE 802.11-2007:**
  - ▶ Updated standard including a, b, d, e, g, h, i and j. (2007)



# IEEE 802.11 Amendments

---

- ▶ **IEEE 802.11k:**
  - ▶ Radio resource measurement enhancements (2008)
- ▶ **IEEE 802.11n:**
  - ▶ Higher throughput improvements using MIMO (multiple input, multiple output antennas) (September 2009)
- ▶ **IEEE 802.11p:**
  - ▶ WAVE—Wireless Access for the Vehicular Environment (such as ambulances and passenger cars) (2010)
- ▶ **IEEE 802.11r:**
  - ▶ Fast BSS transition (FT) (2008)
- ▶ **IEEE 802.11s:**
  - ▶ Mesh Networking, Extended Service Set (ESS) (2011)
- ▶ **IEEE 802.11u:**
  - ▶ Improvements related to HotSpots and 3rd party authorization of clients, e.g. cellular network offload (2011)
- ▶ **IEEE 802.11v:**
  - ▶ Wireless network management (2011)
- ▶ **IEEE 802.11w:**
  - ▶ Protected Management Frames (2009)
- ▶ **IEEE 802.11y:**
  - ▶ 3650–3700 MHz Operation in the U.S. (2008)
- ▶ **IEEE 802.11z:**
  - ▶ Extensions to Direct Link Setup (DLS) (2010)



# IEEE 802.11 Amendments

---

- ▶ **IEEE 802.11-2012:**
  - ▶ New release including k, n, p, r, s, u, v, w, y and z (2012)
- ▶ **IEEE 802.11aa:**
  - ▶ Robust streaming of Audio Video Transport Streams (2012)
- ▶ **IEEE 802.11ac:**
  - ▶ Very High Throughput < 6GHz
  - ▶ Potential improvements over 802.11n: better modulation scheme (expected ~10% throughput increase), wider channels (estimate in future time 80 to 160 MHz), multi user MIMO (2012)
- ▶ **IEEE 802.11ad:**
  - ▶ Very High Throughput 60 GHz (~ February 2014)
- ▶ **IEEE 802.11ae:**
  - ▶ Prioritization of Management Frames (2012)
- ▶ **IEEE 802.11af:**
  - ▶ TV Whitespace (February 2014)



# IEEE 802.11 Amendments

---

- ▶ IEEE 802.11ah:
  - ▶ Sub 1 GHz sensor network, smart metering
- ▶ IEEE 802.11ai:
  - ▶ Fast Initial Link Setup
- ▶ IEEE 802.11aj:
  - ▶ China MM Wave
- ▶ IEEE 802.11aq:
  - ▶ Pre-association Discovery
- ▶ IEEE 802.11ak:
  - ▶ General Links
- ▶ IEEE 802.11mc:
  - ▶ Maintenance of the standard
- ▶ IEEE 802.11ax:
  - ▶ High Efficiency WLAN
- ▶ IEEE 802.11ay:
  - ▶ Enhancements for Ultra High Throughput in and around the 60 GHz Band
- ▶ IEEE 802.11az:
  - ▶ Next Generation Positioning
- ▶ IEEE 802.11ba
  - ▶ Wake Up Radio
- ▶ IEEE 802.11bb:
  - ▶ Light Communications
- ▶ IEEE 802.11be:
  - ▶ Extremely High Throughput



# Other Technologies

---

- ▶ IEEE 802.15 Wireless PAN

- ▶ IEEE 802.15.1

- ▶ Bluetooth certification

- ▶ IEEE 802.15.2

- ▶ IEEE 802.15 and IEEE 802.11 coexistence

- ▶ IEEE 802.15.3

- ▶ High-Rate wireless PAN (e.g., UWB, etc)

- ▶ IEEE 802.15.4

- ▶ Low-Rate wireless PAN (e.g., ZigBee, WirelessHART, MiWi, etc.)

- ▶ IEEE 802.15.5

- ▶ Mesh networking for WPAN

- ▶ IEEE 802.15.6

- ▶ Body area network

- ▶ IEEE 802.16

- ▶ Broadband Wireless Access (WiMAX certification)

