

CS/ECE 439: Wireless Networking

MAC Layer – Road to Wireless

Multiple Access Media

- ▶ **Media access**

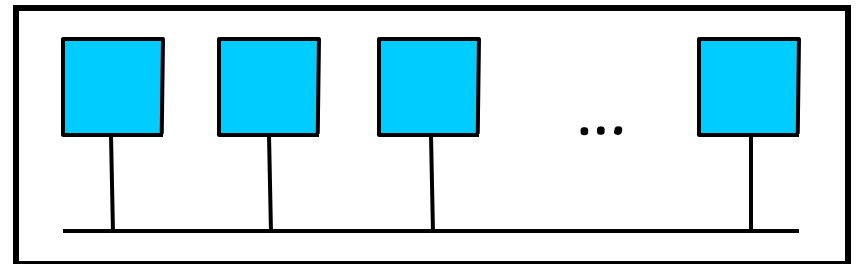
- ▶ Controlling which frame should be sent over the link next
 - ▶ Easy for point-to-point links; half versus full duplex
 - ▶ Harder for multi-access links: who gets to send?

- ▶ **Multiple senders on some media**

- ▶ Buses (Ethernet)
- ▶ Radio, Satellite

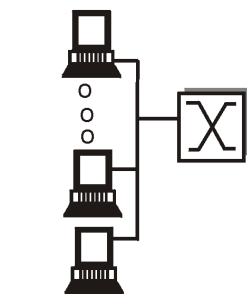
- ▶ **Goals**

- ▶ Fair arbitration
- ▶ Good performance

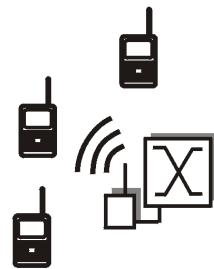


Point-to-Point vs. Broadcast Media

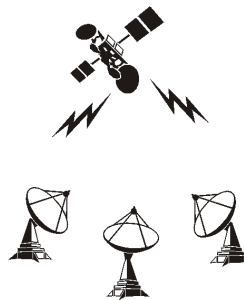
- ▶ **Point-to-point: dedicated pairwise communication**
 - ▶ Long-distance fiber link
 - ▶ Point-to-point link between Ethernet switch and host
- ▶ **Broadcast: shared wire or medium**
 - ▶ Traditional Ethernet
 - ▶ 802.11 wireless LAN



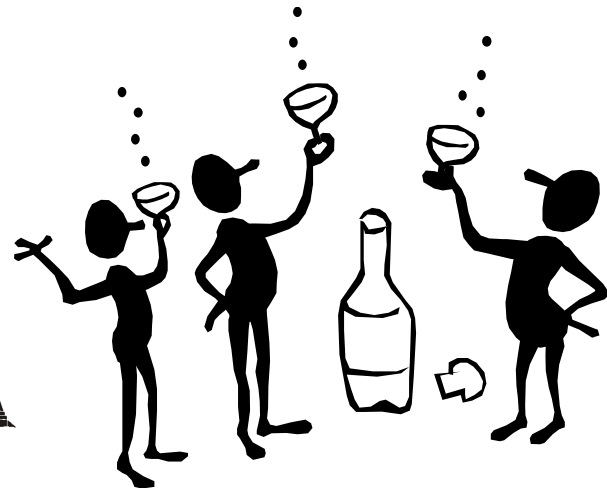
shared wire
(e.g. Ethernet)



shared wireless
(e.g. Wavelan)

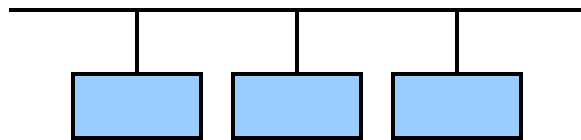


satellite

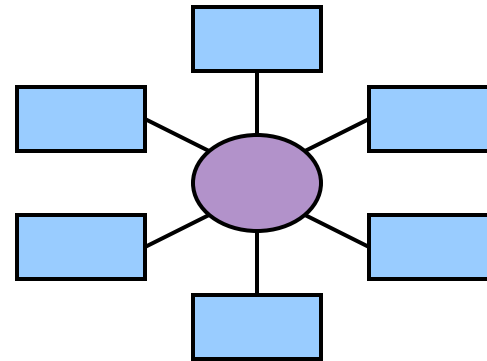


cocktail party

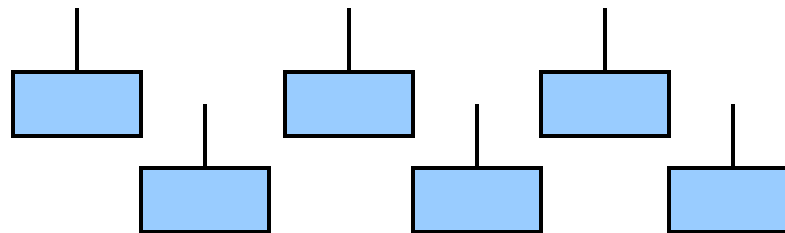
Types of Shared Link Networks



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.



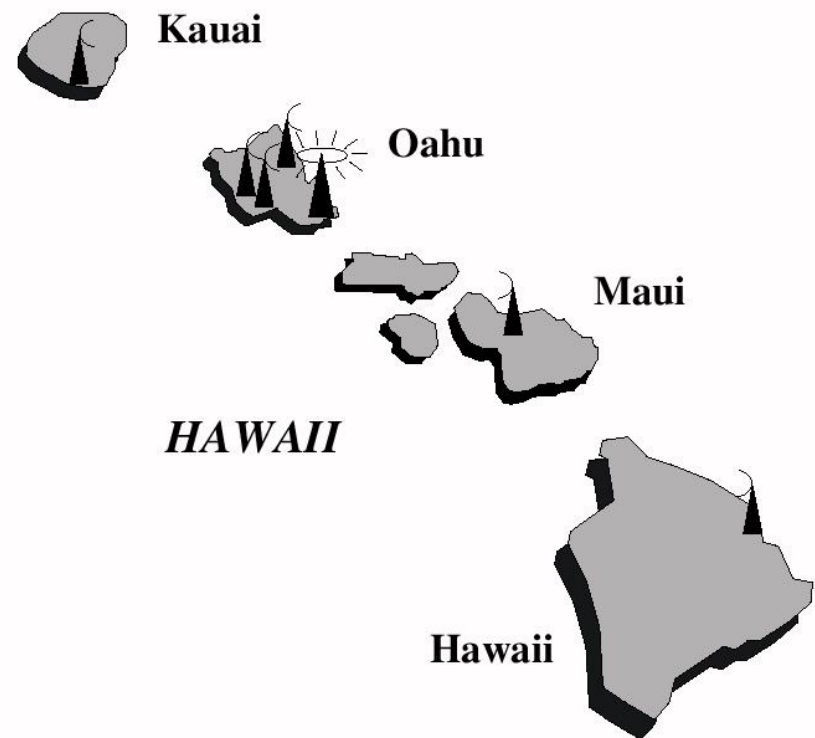
Multiple Access Media: Random Access

- ▶ **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

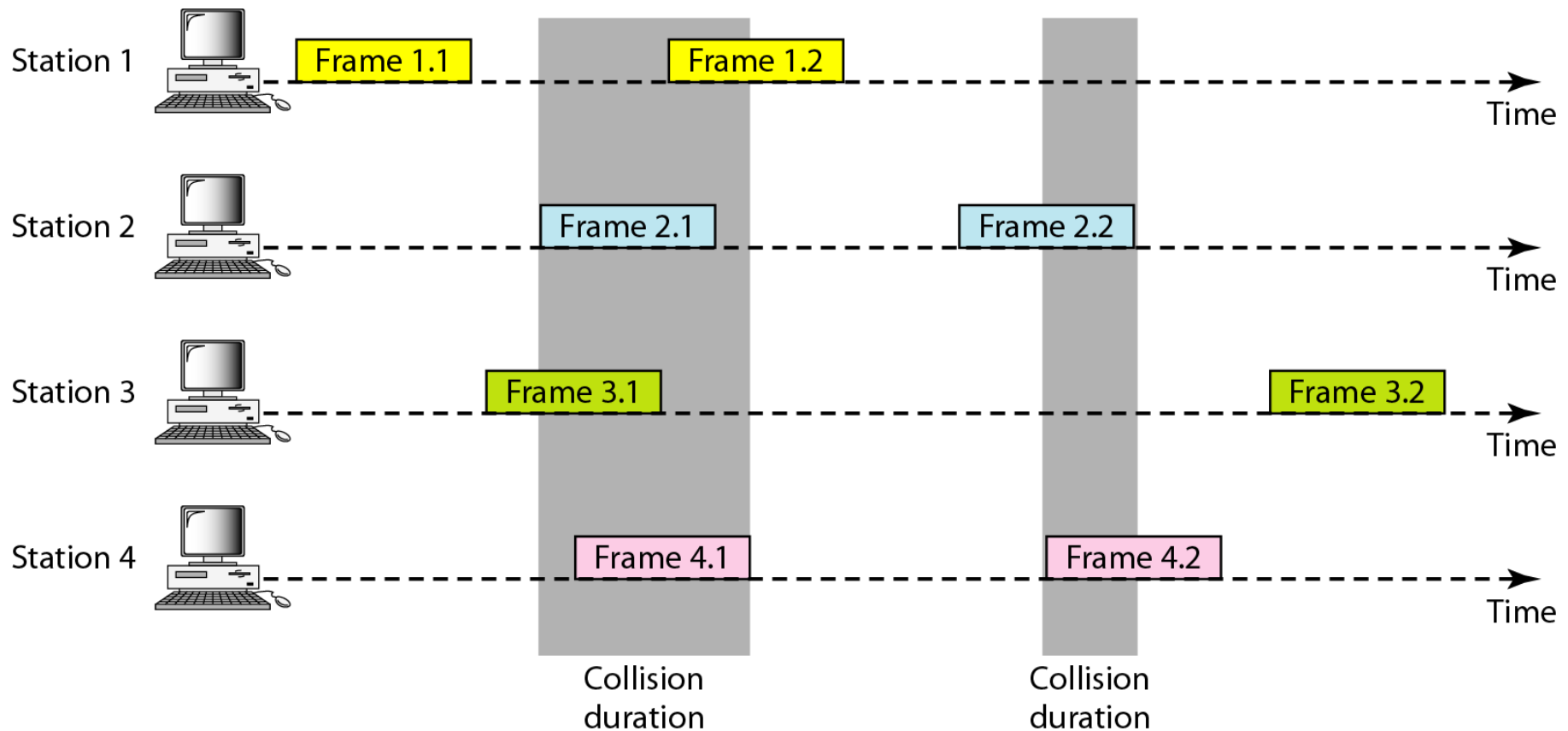
- ▶ Transmitter sends line
- ▶ Each station transmits λ packets/sec on average based on a Poisson arrival process

▶ Checks for success (no interference)

▶ If collision occurred, wait random time and resend



Pure ALOHA

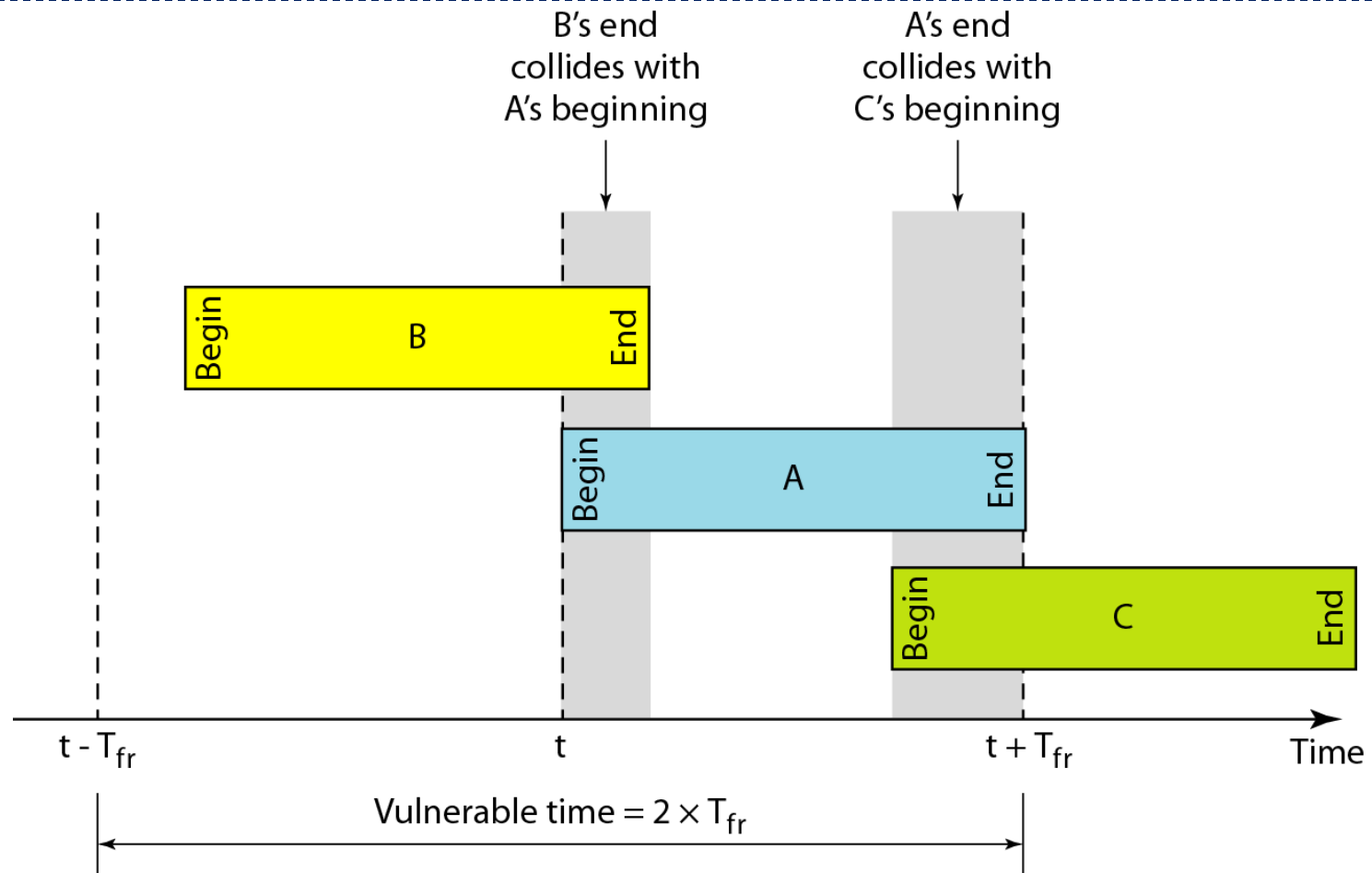


Pure ALOHA

► 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

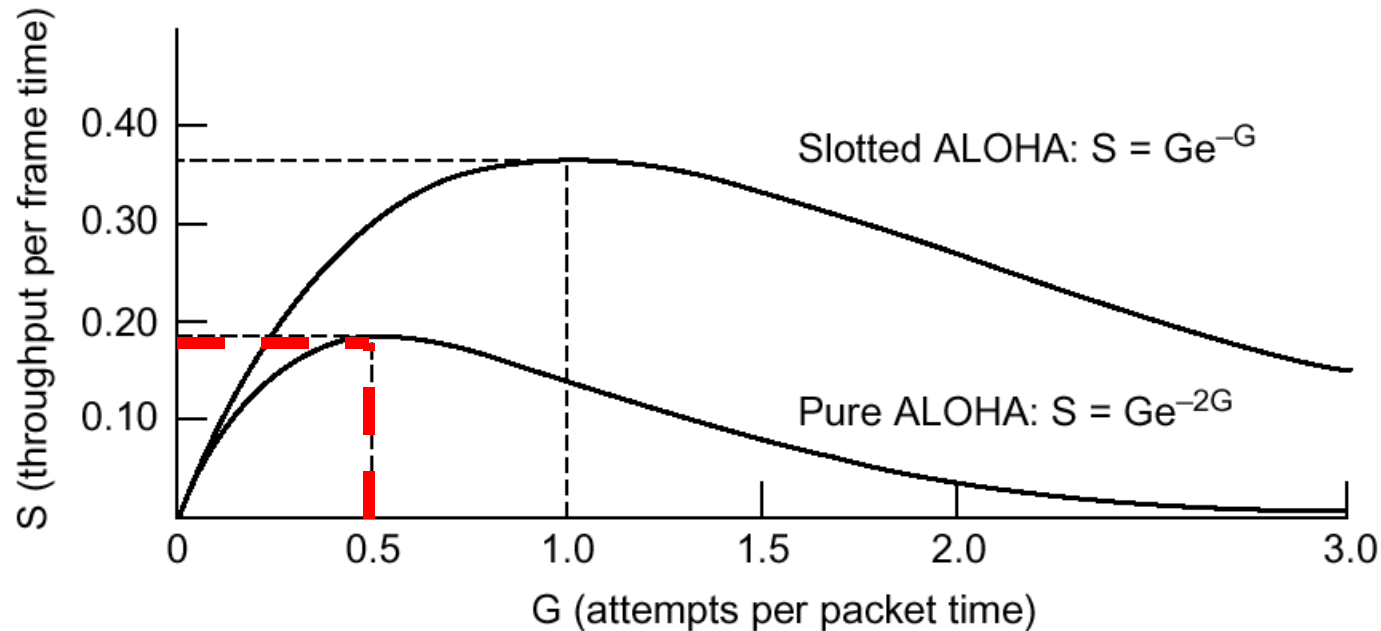
- ▶ Maximum throughput

- ▶ $G = 0.5$

- ▶ $S = 1/2e$

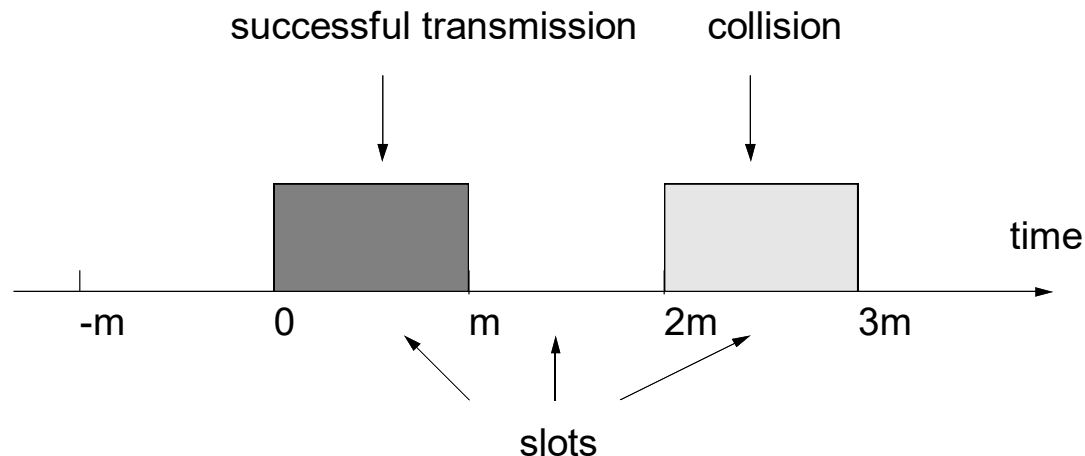
- Utilization

- Maximum of 0.184!

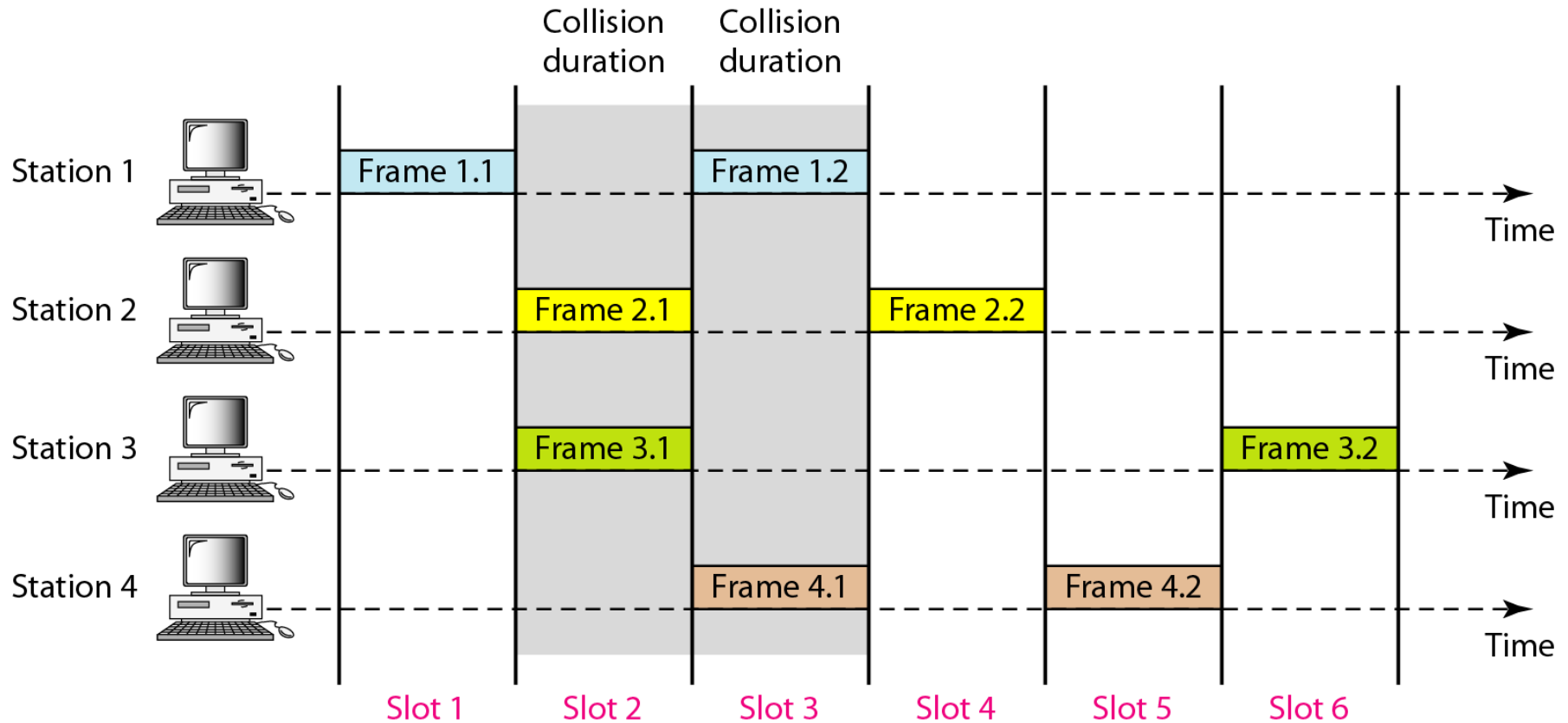


Slotted ALOHA

- ▶ 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 $G = 1$.

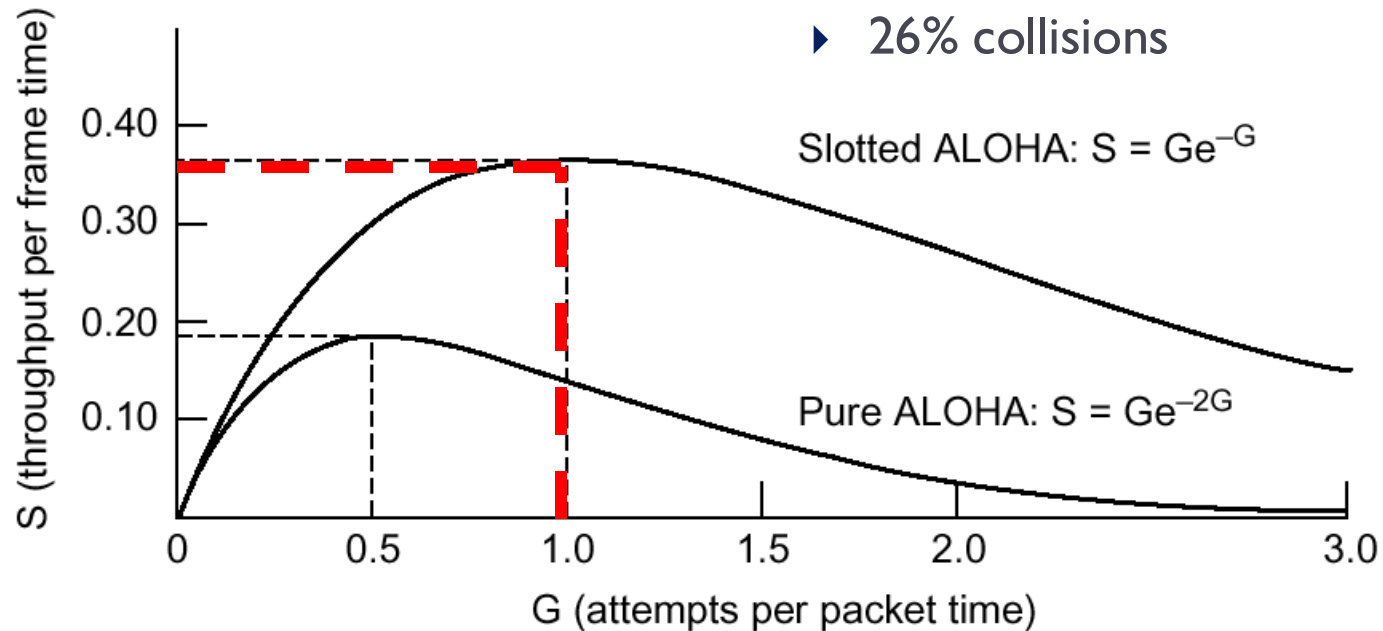
Slotted ALOHA

- ▶ 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

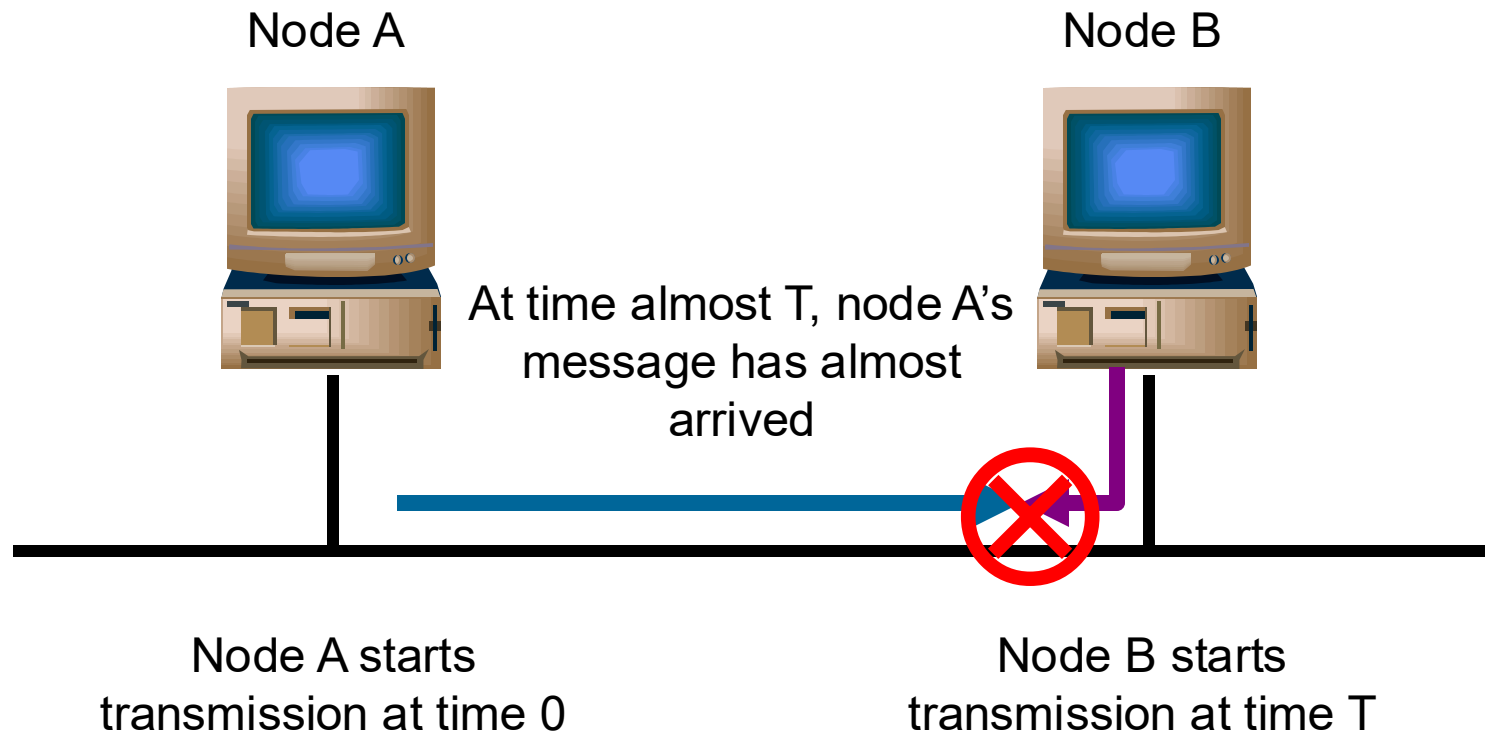
- ▶ **Tradeoff**

- ▶ 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

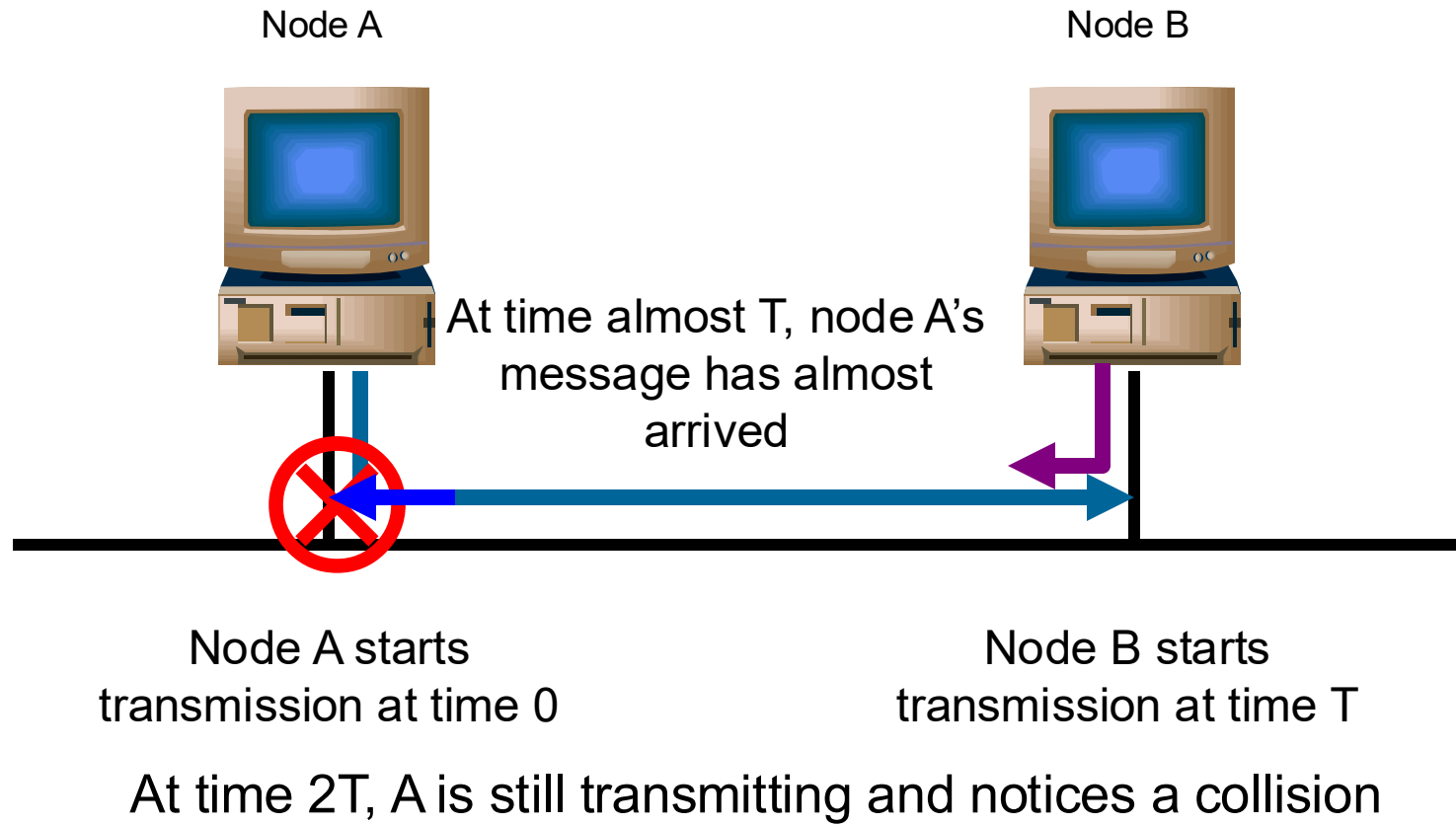
- ▶ A must still be transmitting when it receives B's transmission!

- ▶ Example

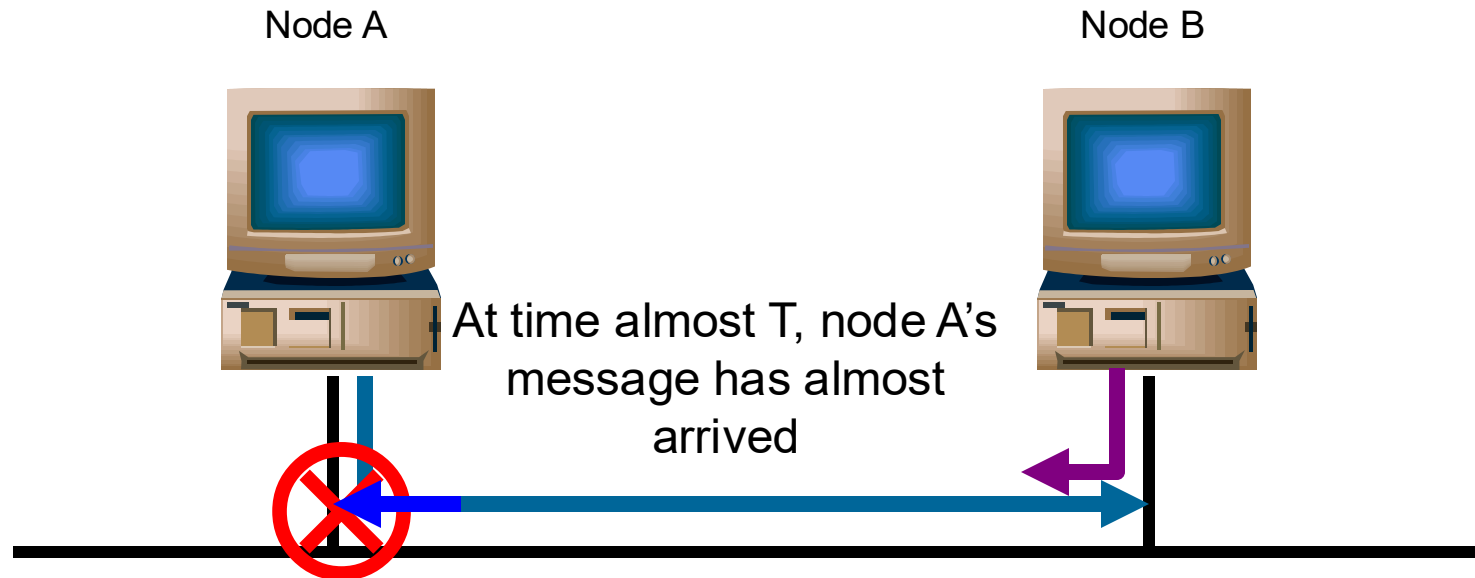
- ▶ Node A's message reaches node B at time T
 - ▶ Node B's message reaches node A at time $2T$
 - ▶ For node A to detect a collision, node A must still be transmitting at time $2T$



Ethernet MAC Algorithm



Ethernet MAC Algorithm



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 64B

Packet length $\geq 64\text{B}$

Ethernet MAC Algorithm

▶ Sender/Transmitter

- ▶ If line is idle (carrier sensed)
 - ▶ Send immediately
 - ▶ Send maximum of 1500B data (1527B total)
 - ▶ Wait 9.6 μ 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 μ s?

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

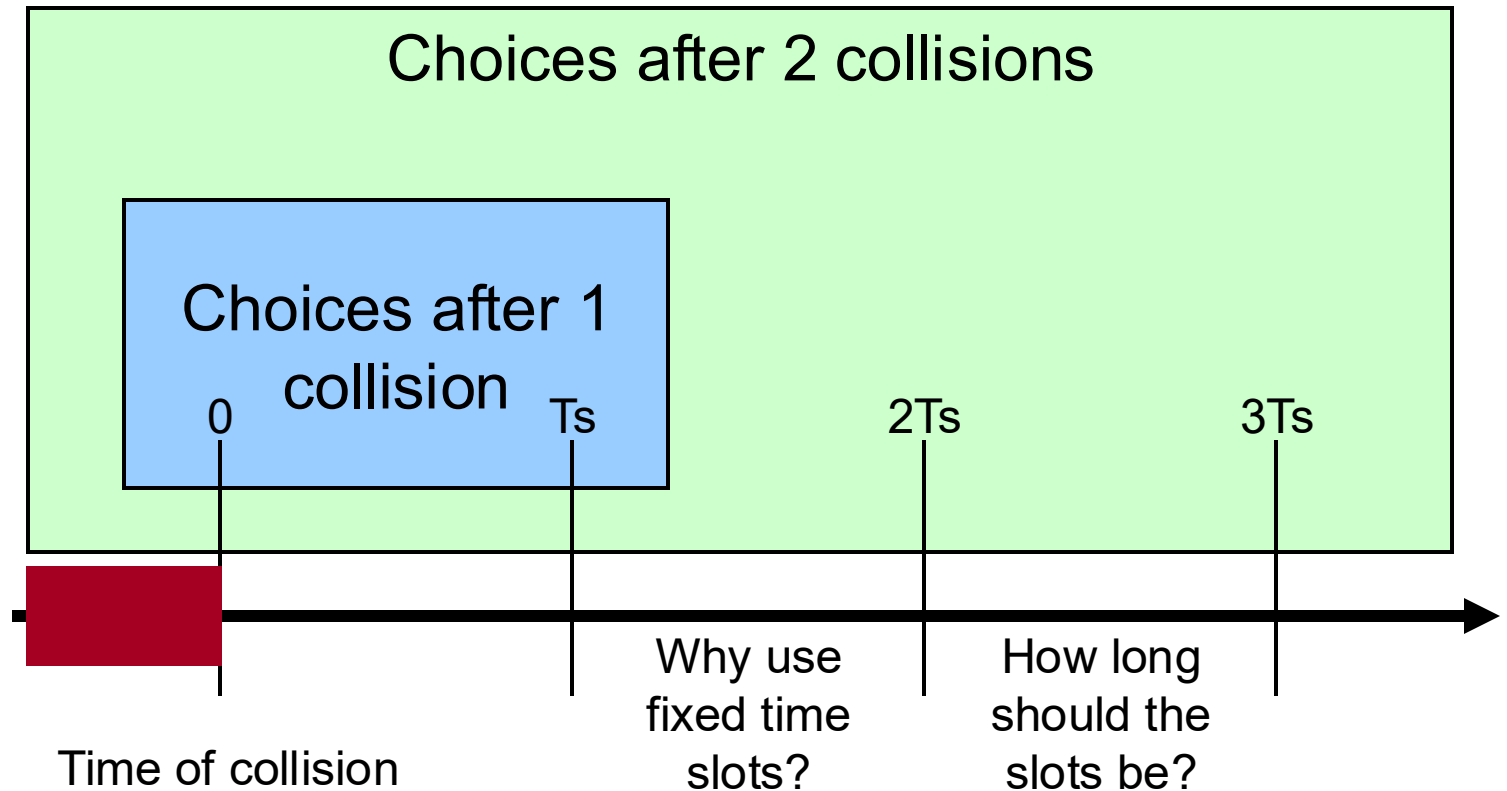
Incoming signal $\neq$
outgoing signal!



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 algorithm

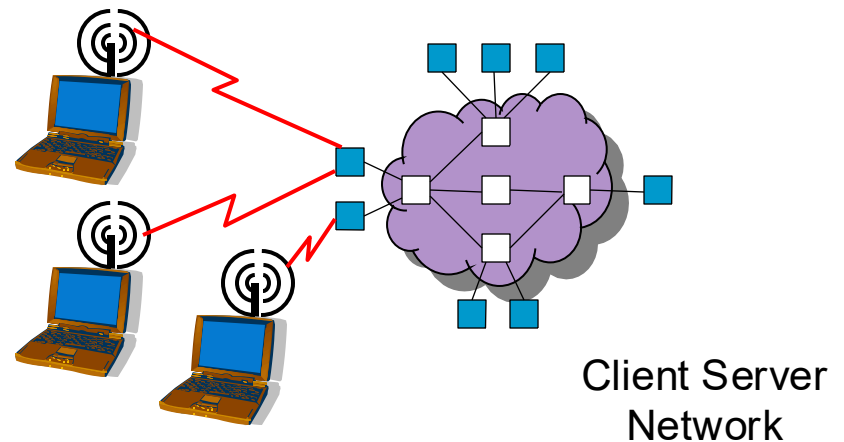
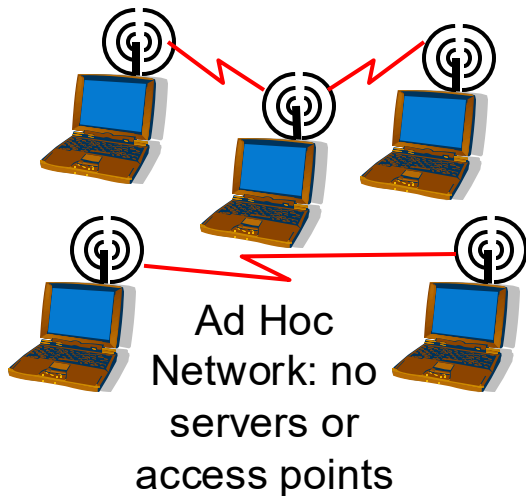
- ▶ When collision first occurs
 - ▶ Send a jamming signal to prevent further data being sent
- ▶ Resend a frame
 - ▶ After either 0 or T seconds, chosen at random
- ▶ If resend fails, resend the frame again
 - ▶ After either 0 , T , $2T$, or $3T$ seconds.
 - ▶ In other words, send after kT seconds, 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^{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

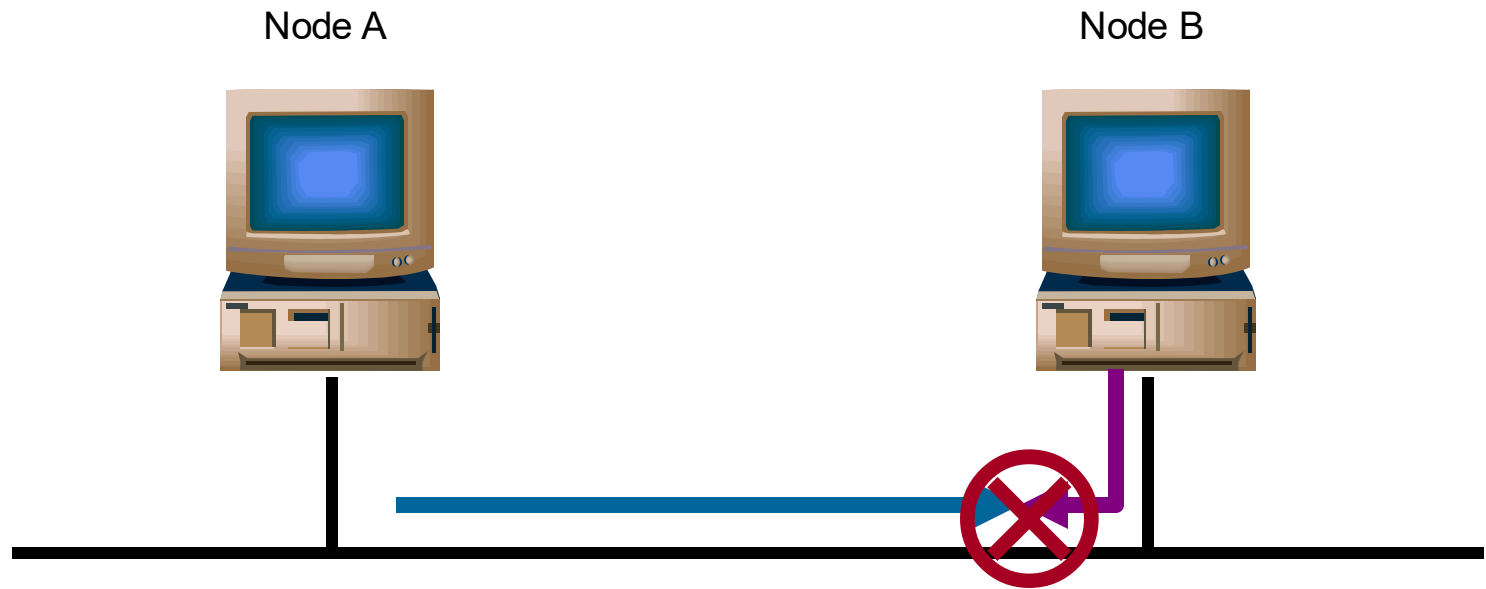
- ▶ A physical and multiple access layer standard for wireless local area networks (WLAN)



Medium Access Control

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

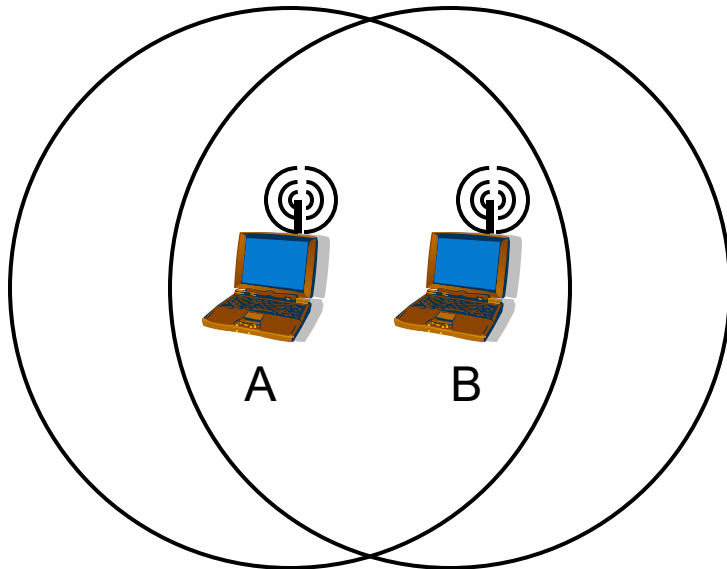
Ethernet MAC Algorithm



- 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

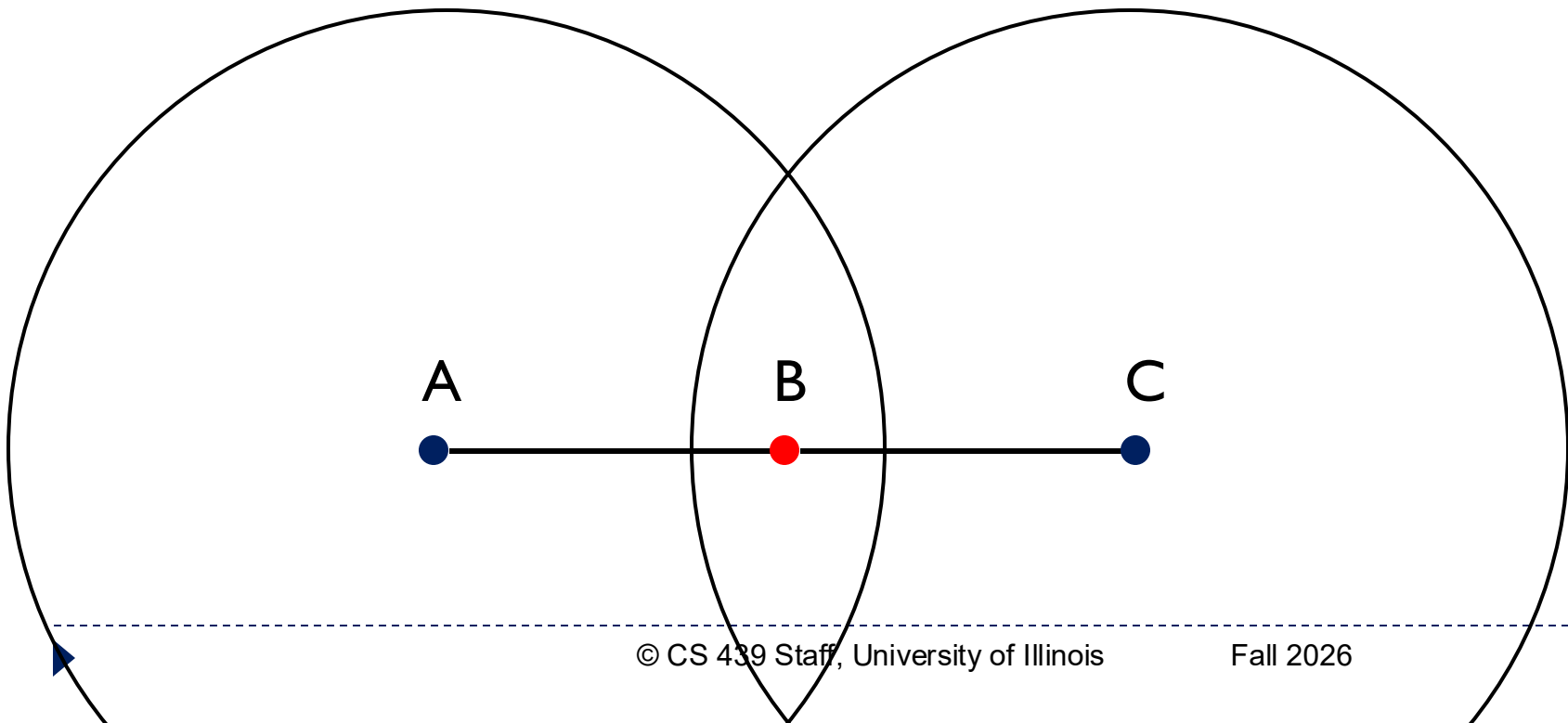
- ▶ A set of nodes send and receive frames over a shared link

- ▶ CA – Collision Avoidance

- ▶ Nodes use protocol to prevent collisions from occurring

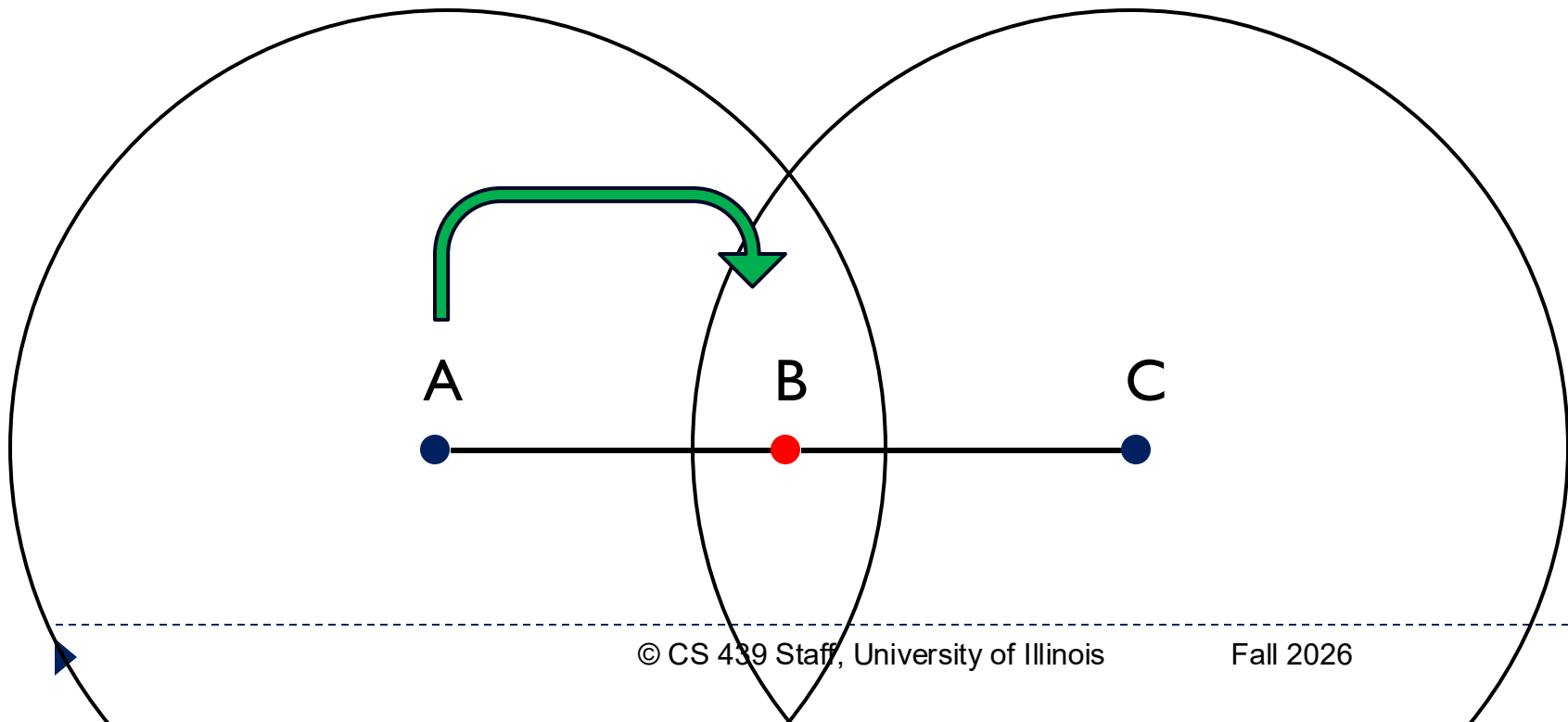
CSMA/CA

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



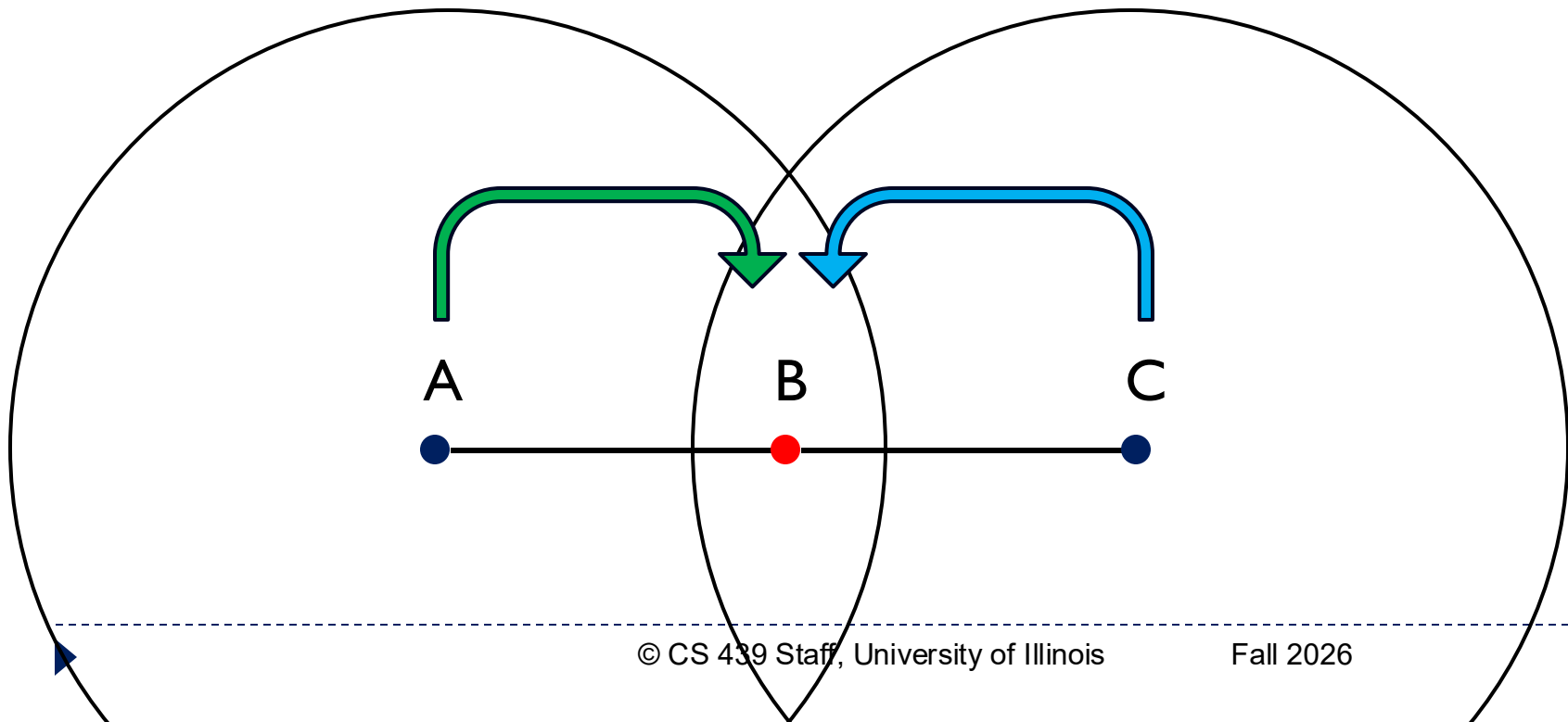
CSMA/CA

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



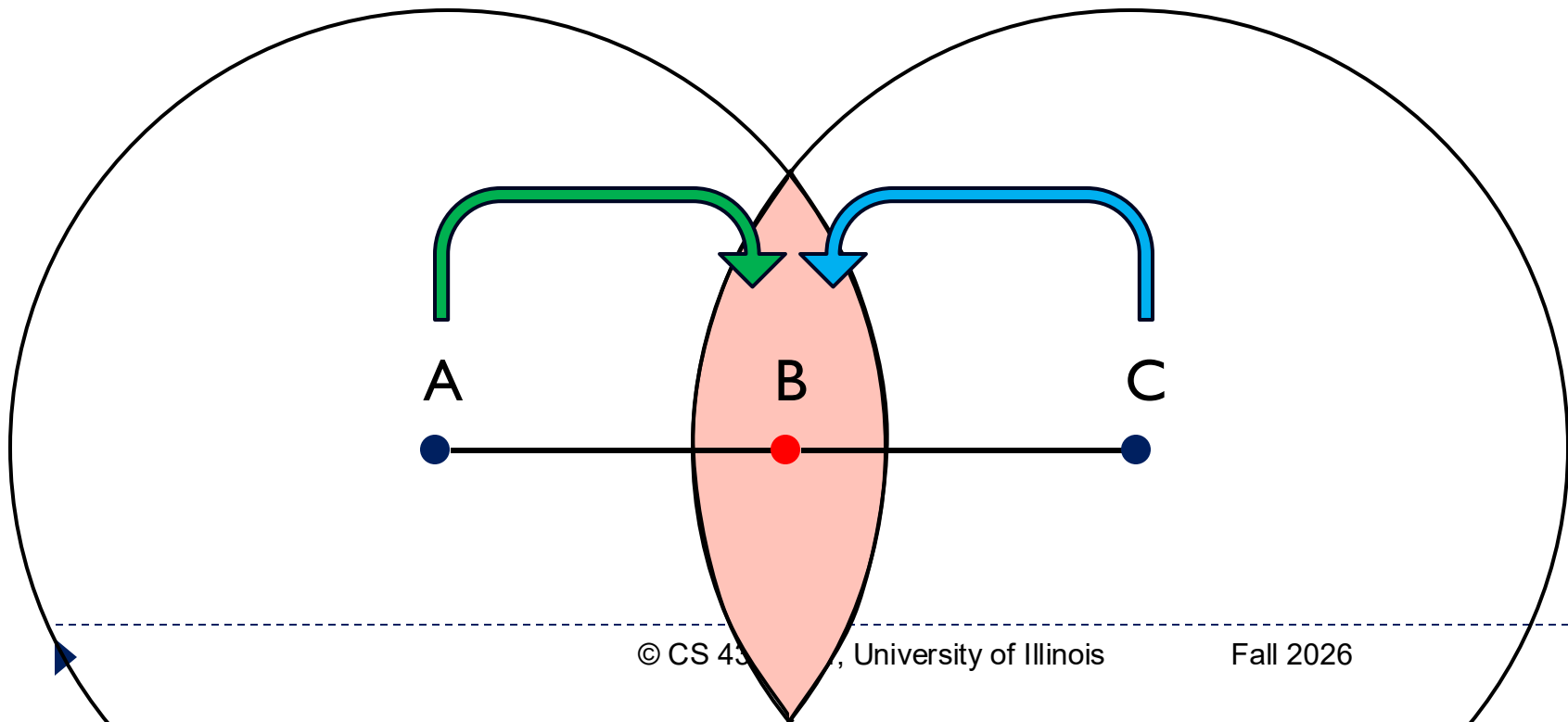
CSMA/CA

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



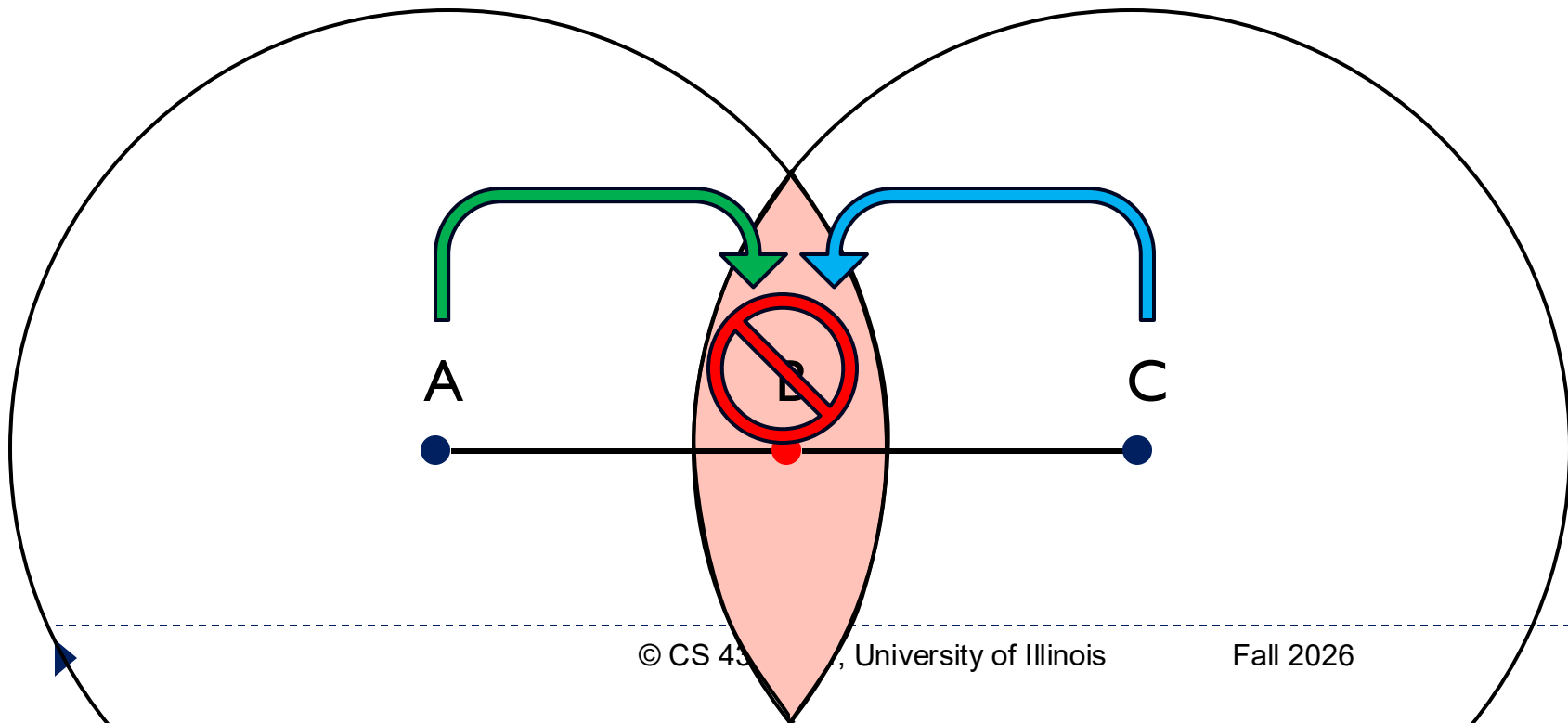
CSMA/CA

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



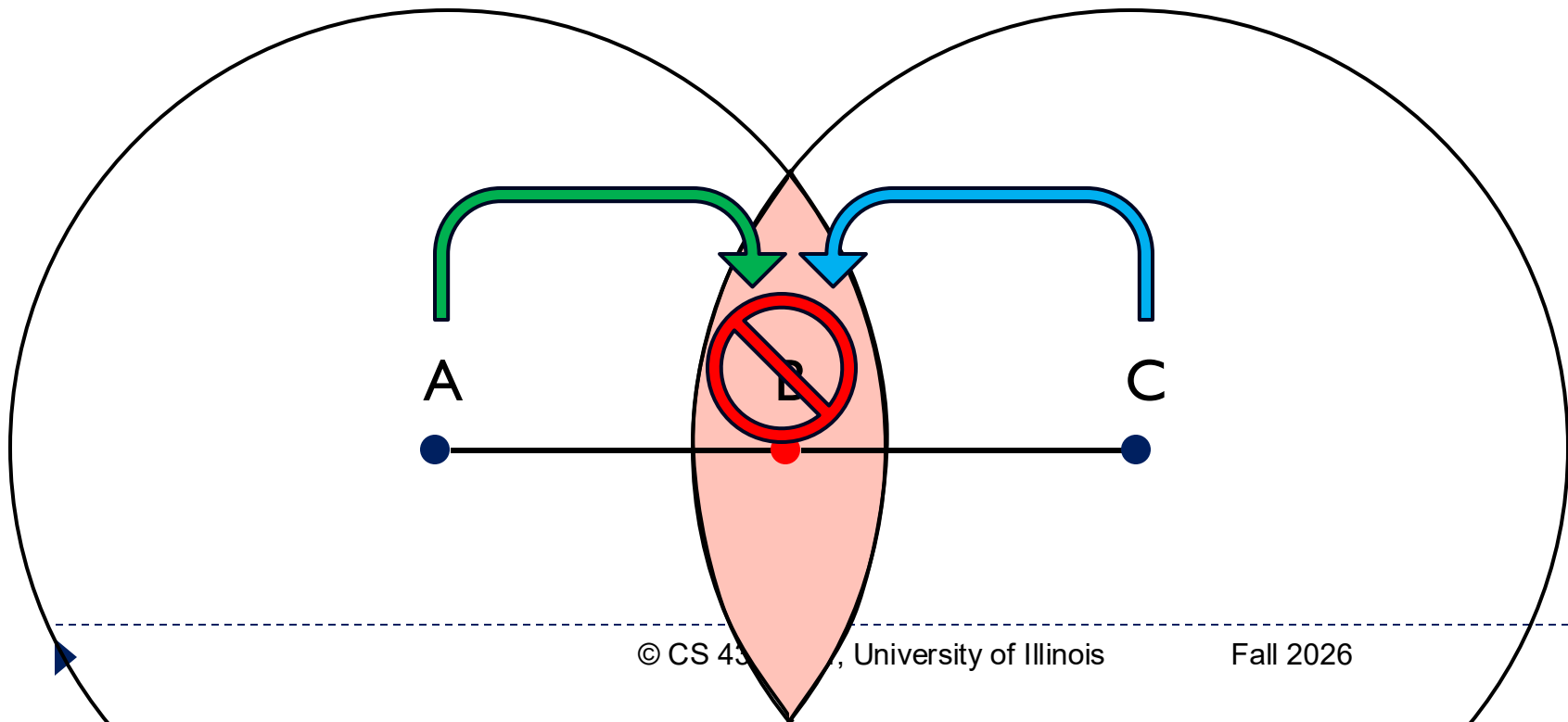
CSMA/CA

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



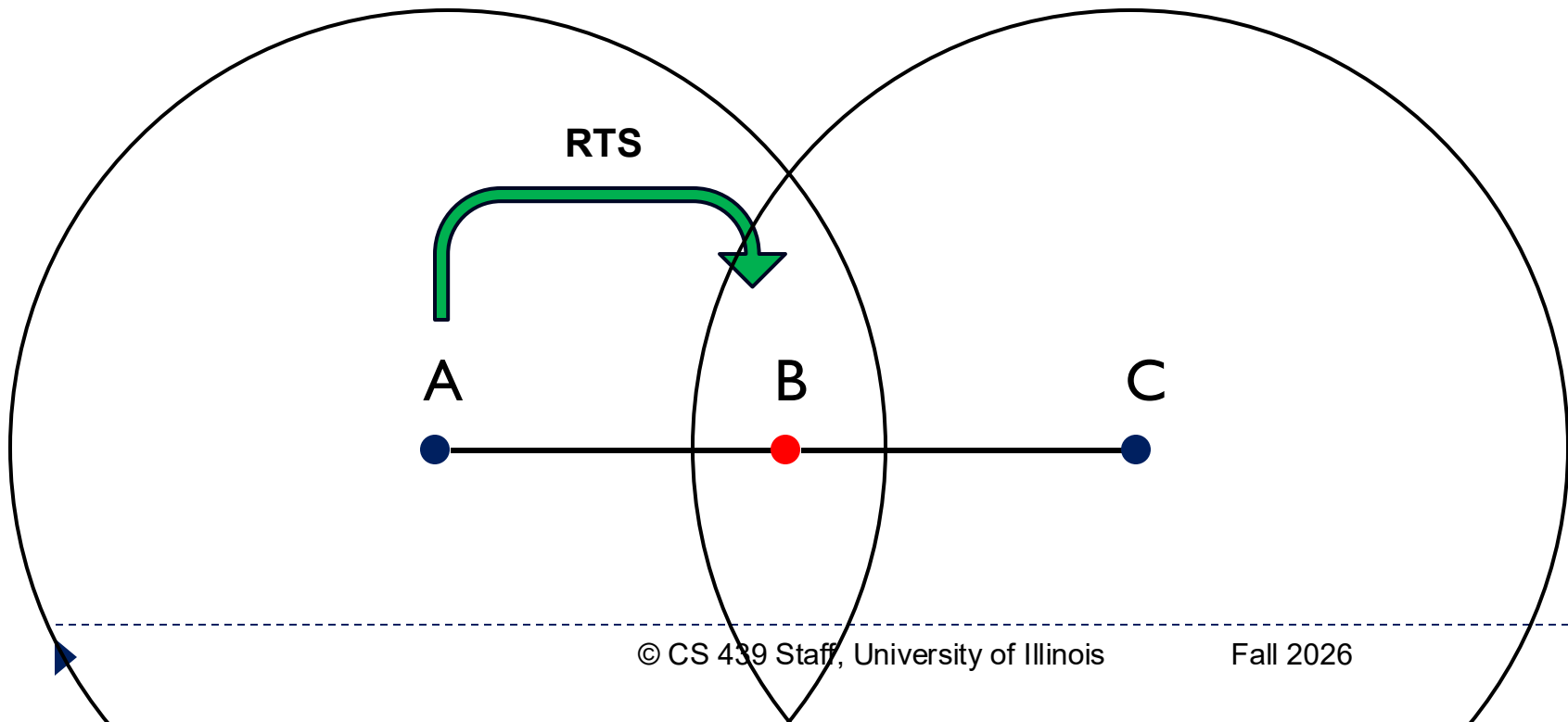
Hidden Terminal Problem

- ▶ 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
- ▶ If C transmits, collision will occur at B



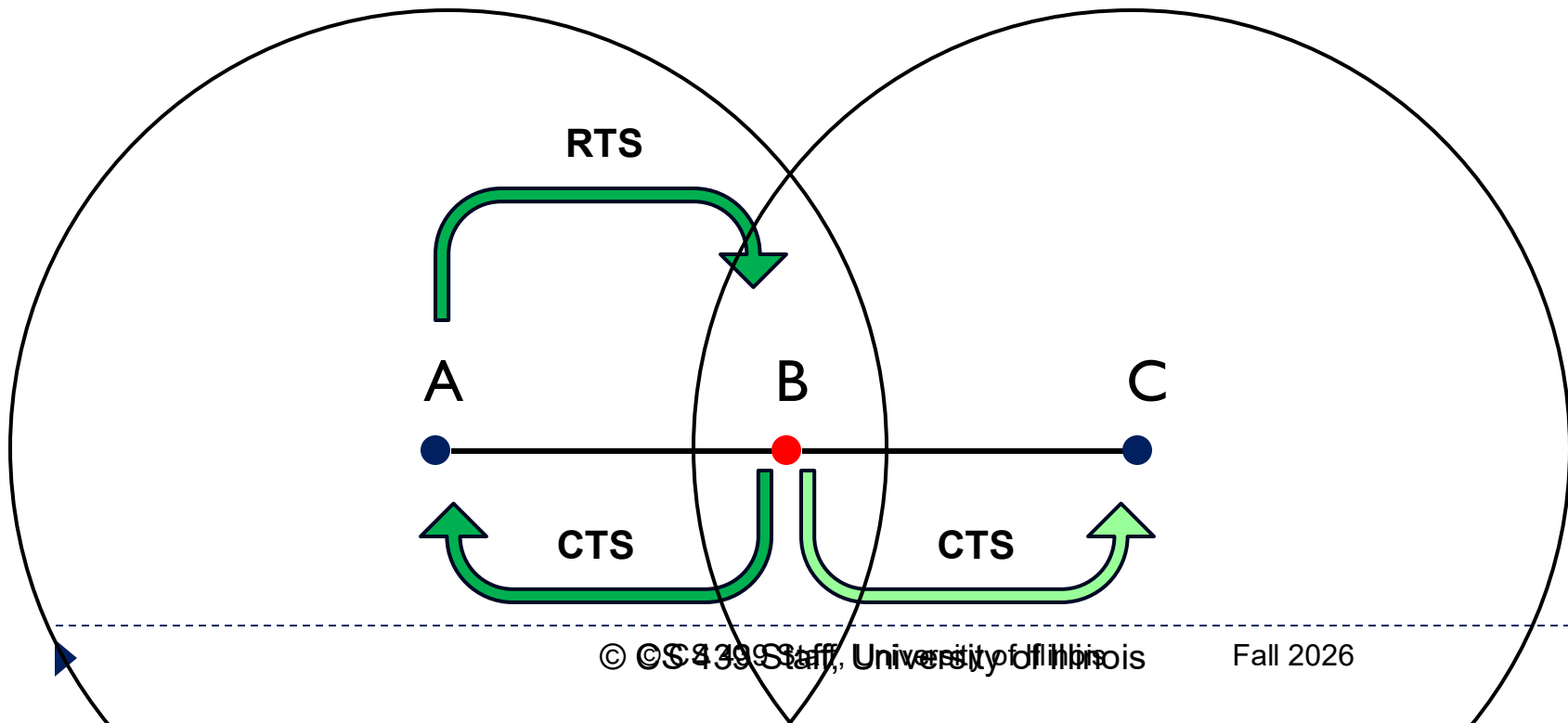
MACA Solution for Hidden Terminal Problem

- ▶ When A wants to send a packet to B
 - ▶ A first sends a Request-to-Send (RTS) to B



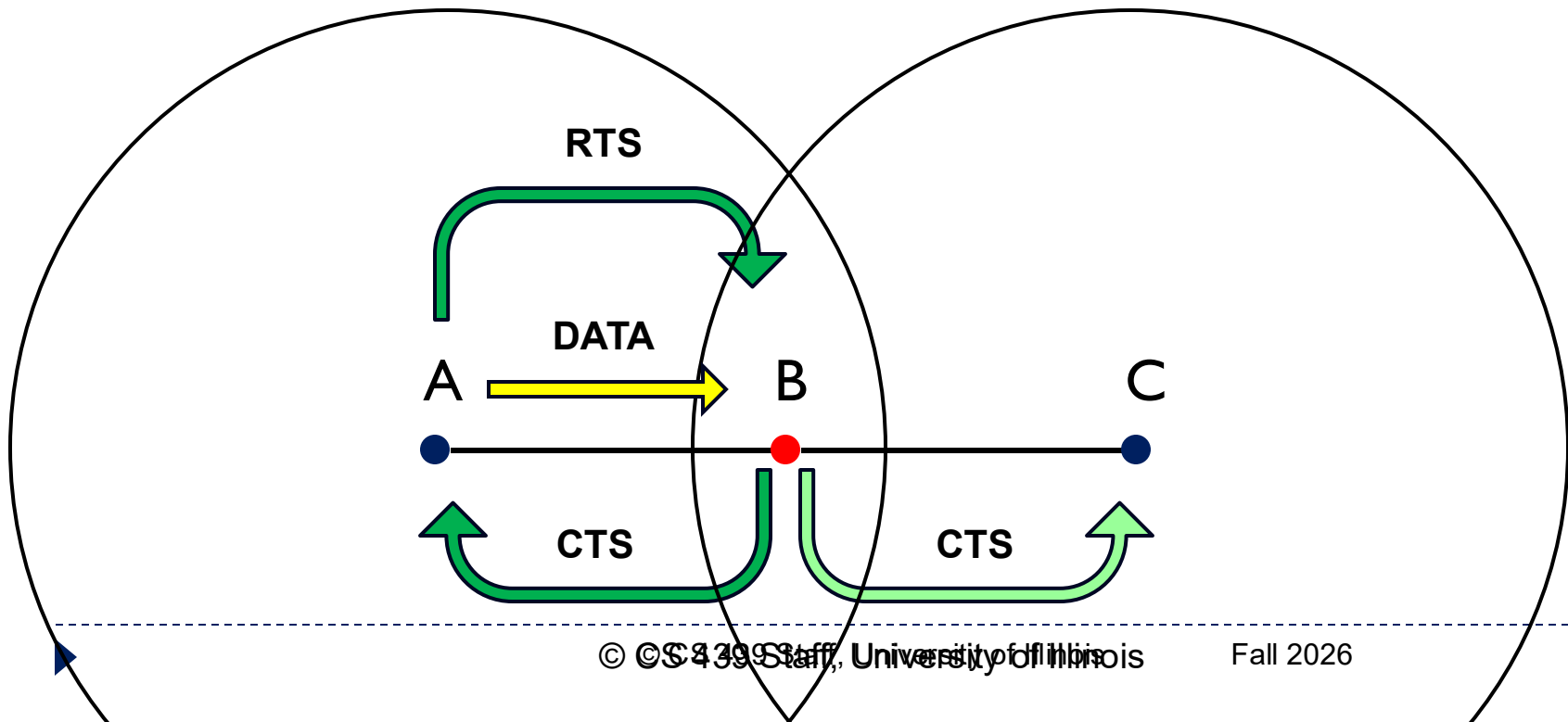
MACA Solution for Hidden Terminal Problem

- ▶ When A wants to send a packet to B
 - ▶ A first sends a Request-to-Send (RTS) to B
- ▶ On receiving RTS
 - ▶ B responds by sending Clear-to-Send (CTS) to A
 - ▶ provided B is able to receive the packet



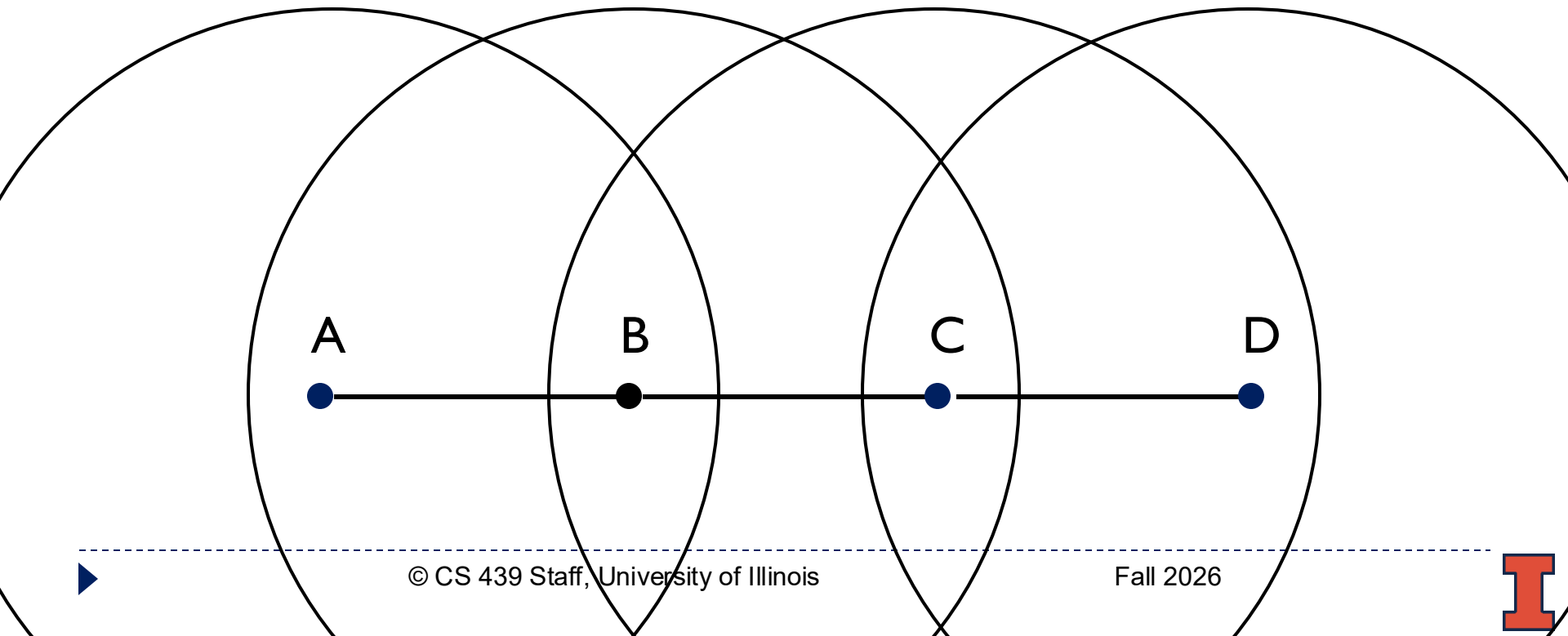
MACA Solution for Hidden Terminal Problem

- ▶ When a C overhears a CTS, it keeps quiet for the duration of the transfer
- ▶ A can now safely send data



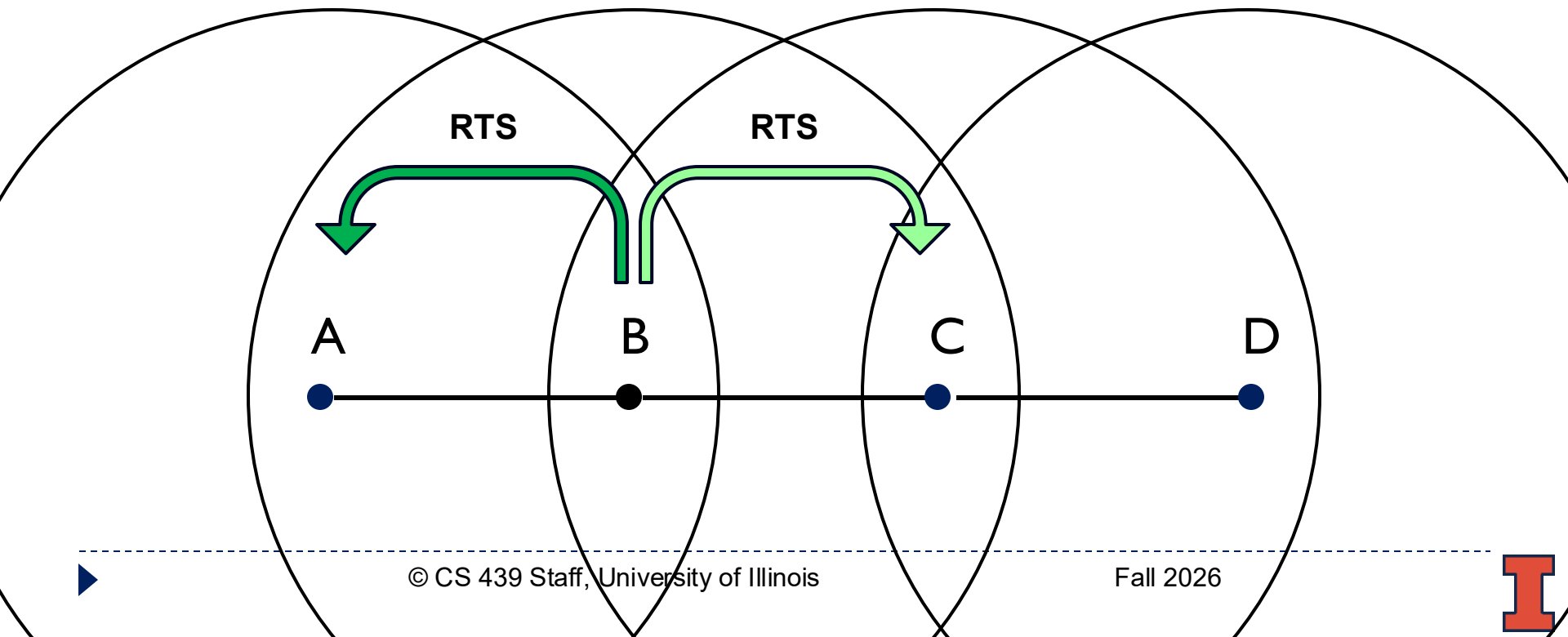
CSMA/CA

- ▶ But we still have a problem



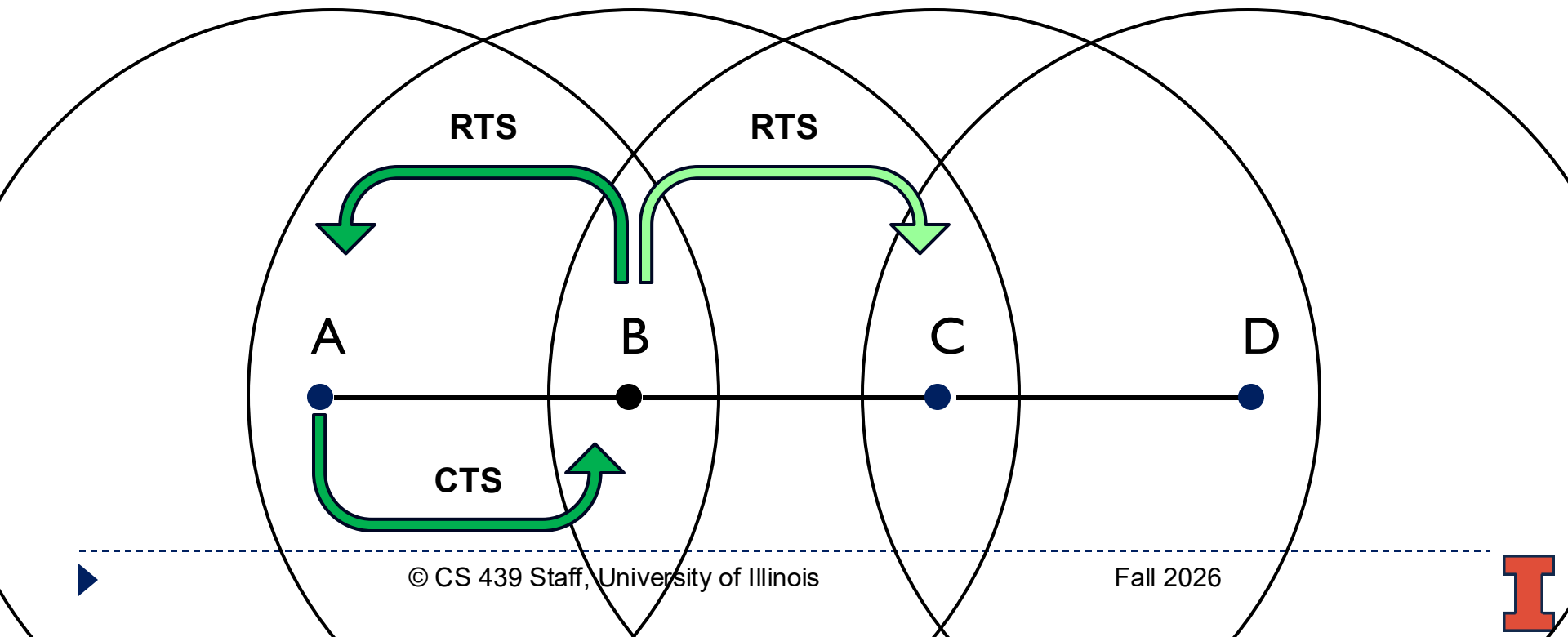
CSMA/CA

- ▶ But we still have a problem



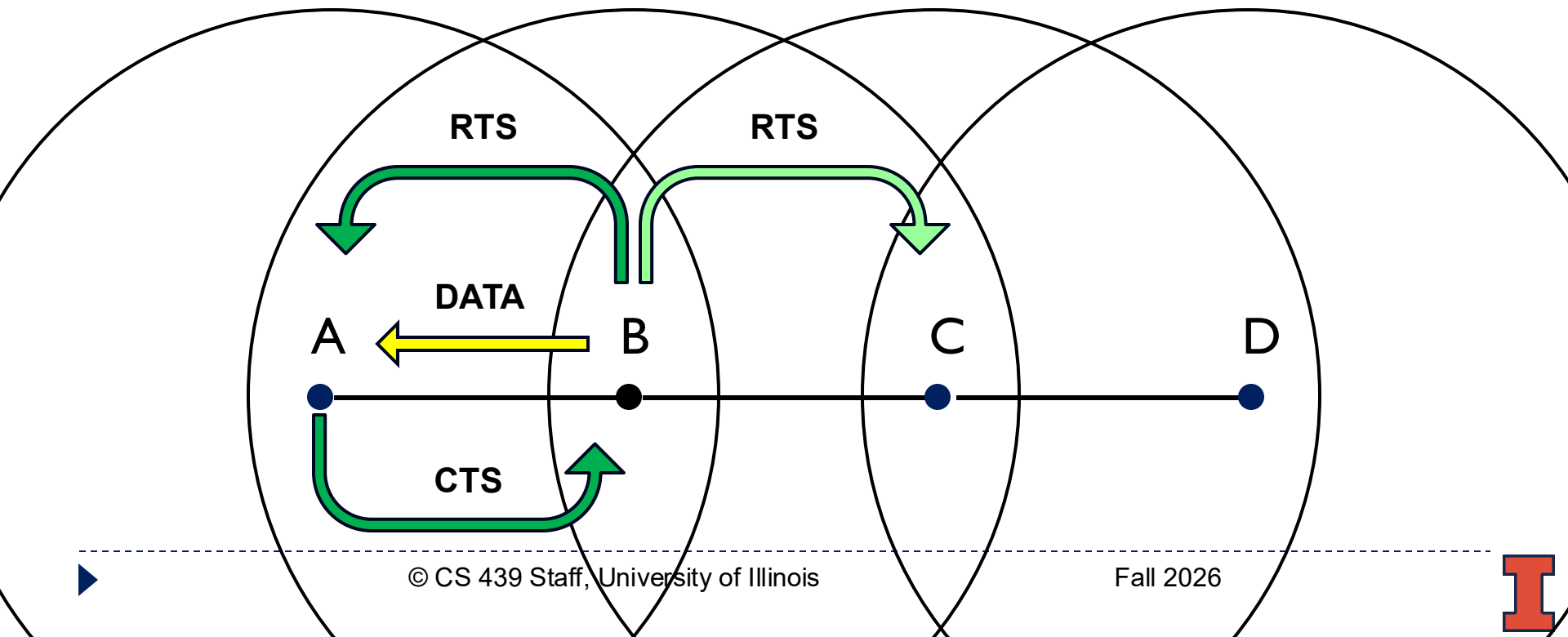
CSMA/CA

- ▶ But we still have a problem



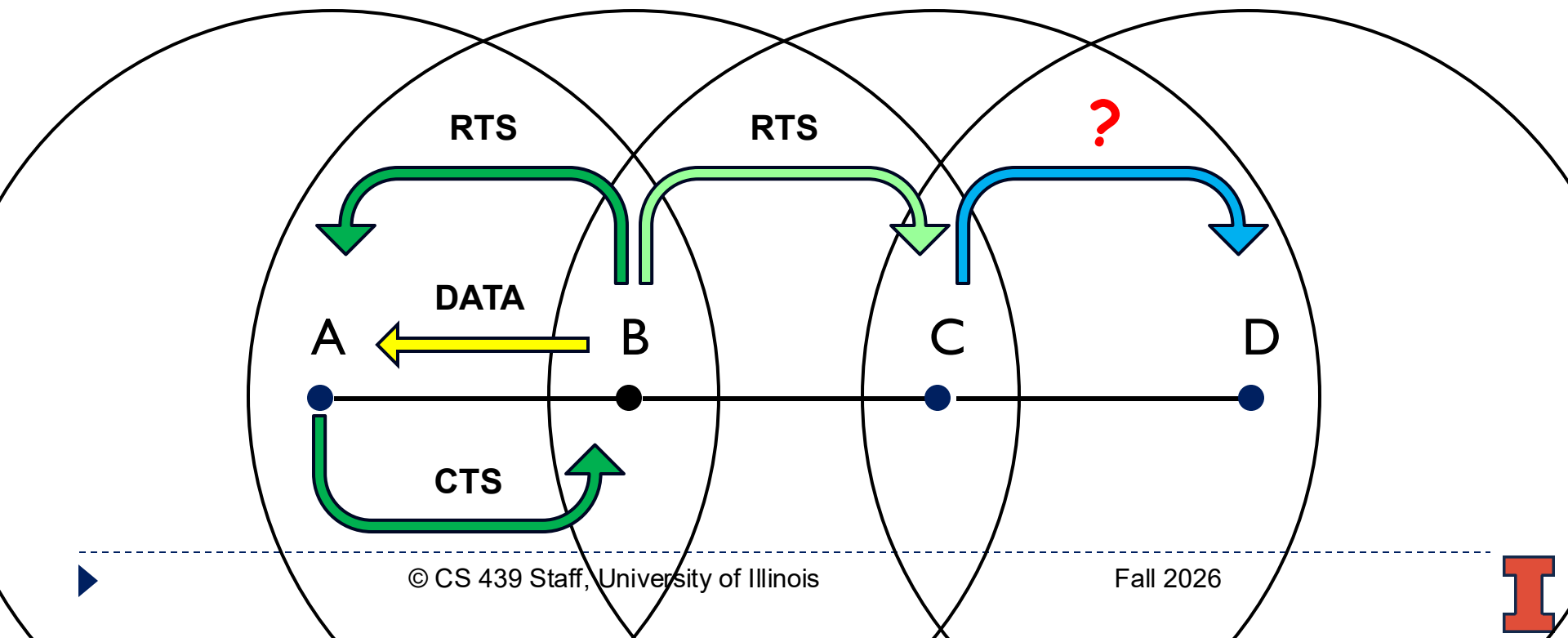
CSMA/CA

- ▶ But we still have a problem



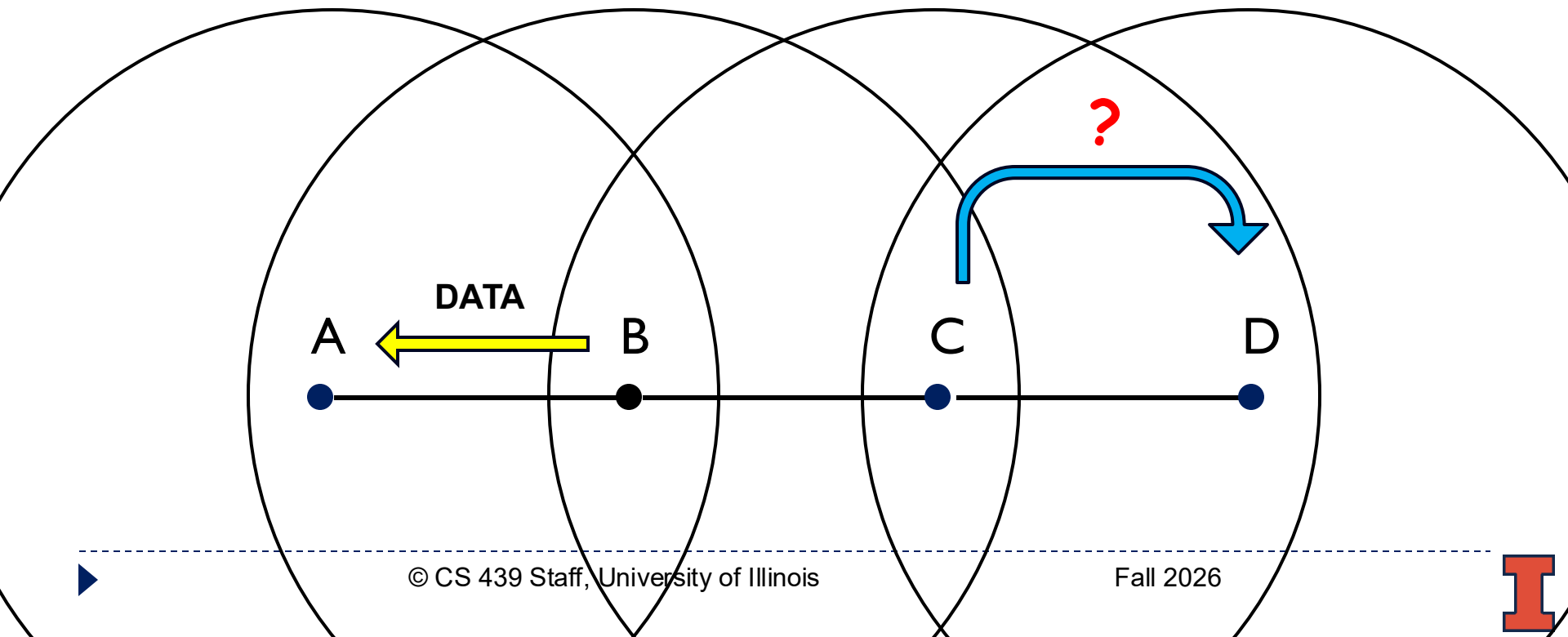
Exposed Terminal Problem

- ▶ But we still have a problem
 - ▶ What should C do if it has a message for D?



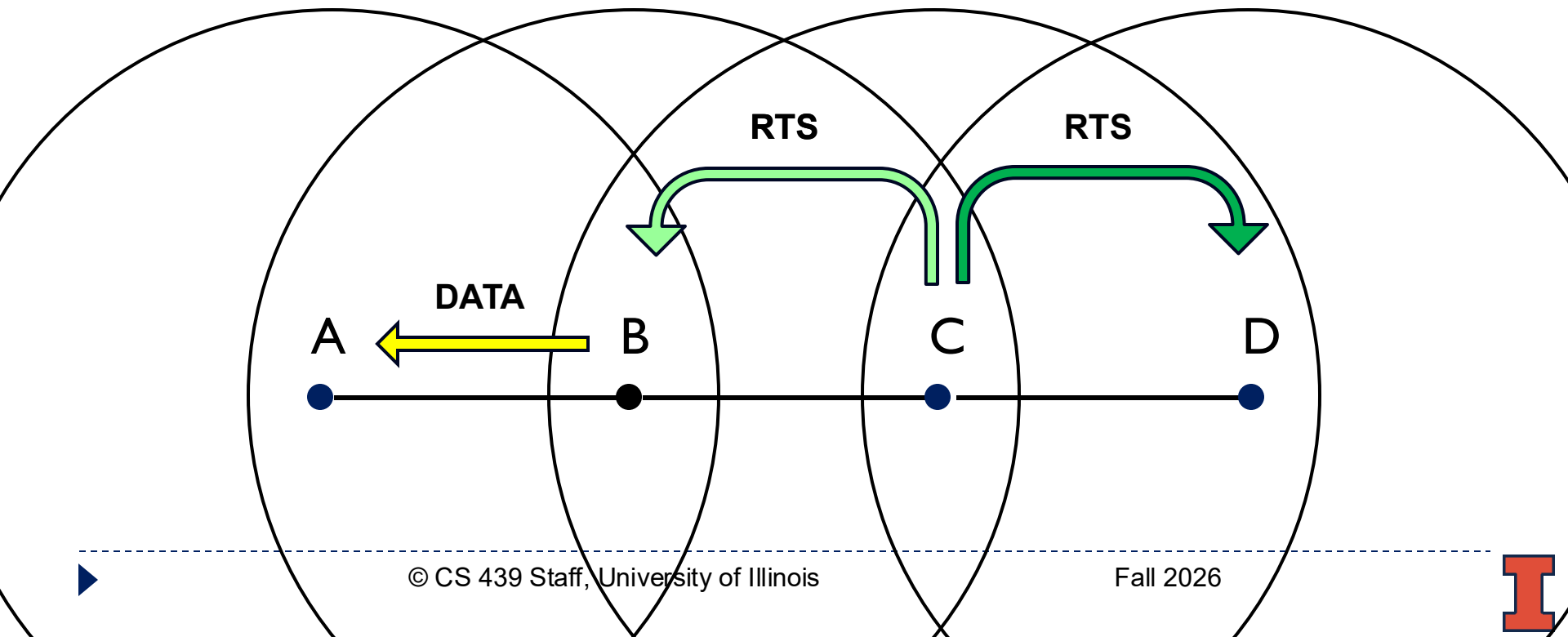
Exposed Terminal Problem

- ▶ But we still have a problem
 - ▶ What should C do if it has a message for D?
 - ▶ C can hear the data message from B to A



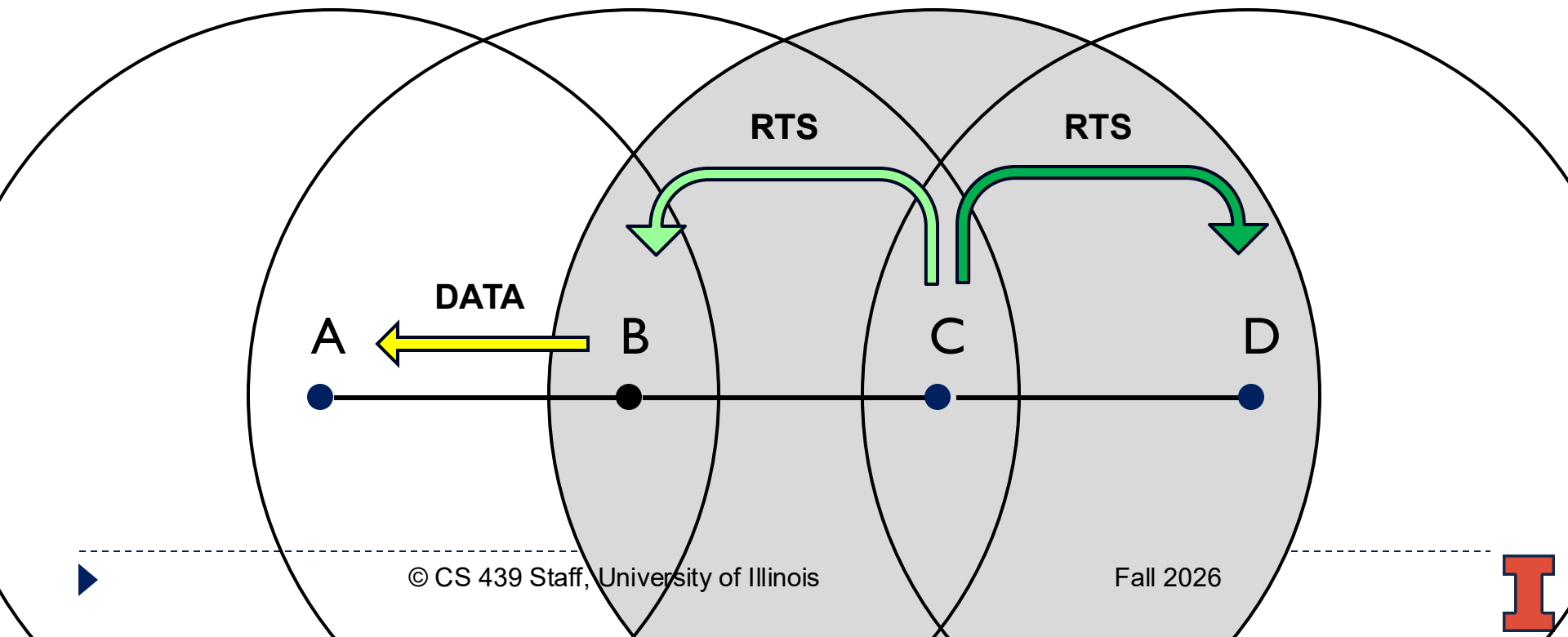
Exposed Terminal Problem

- ▶ But we still have a problem
 - ▶ What happens if C transmits an RTS?



Exposed Terminal Problem

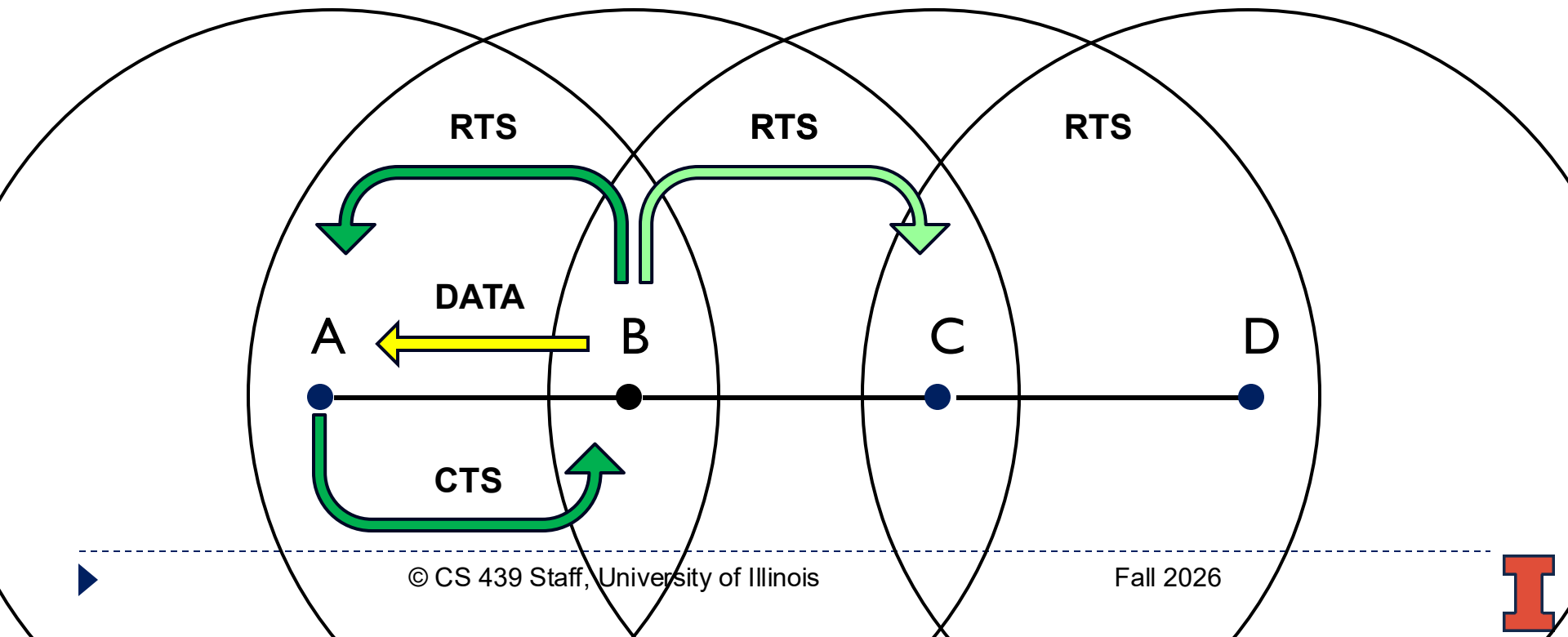
- ▶ But we still have a problem
 - ▶ What happens if C transmits an RTS?



MACA Solution for Exposed Terminal Problem

► Node C

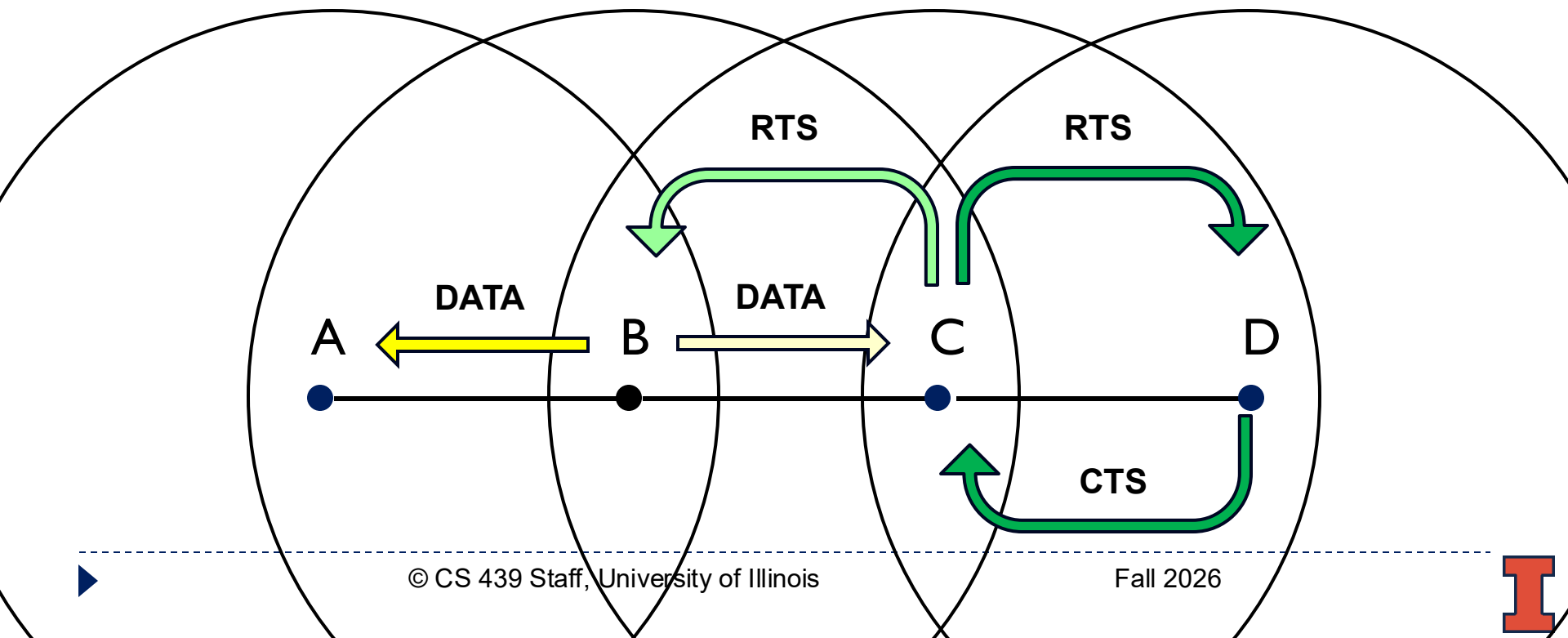
- Hear CTS - Stay quiet
- Hear RTS, but no CTS - OK to transmit



Digging into the Exposed Terminal Problem

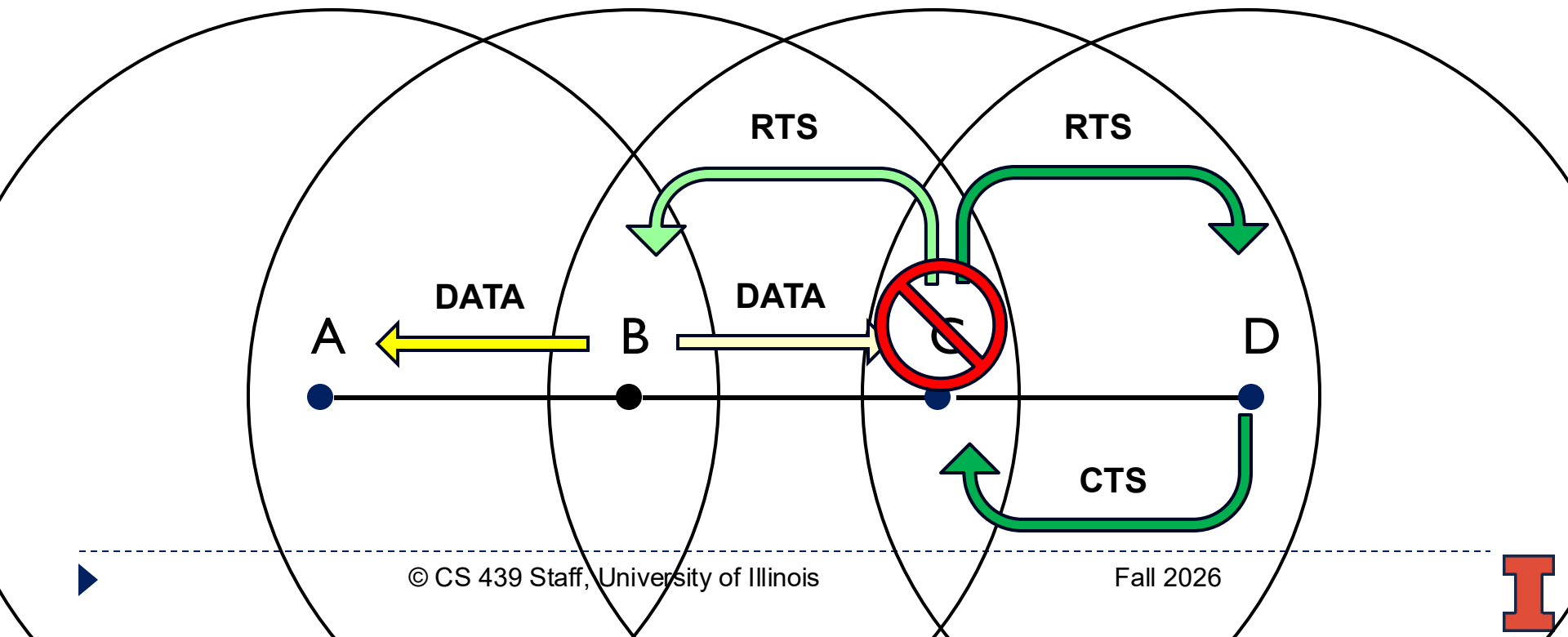
► Problem

- Exposed terminal solution doesn't consider CTS at node C



Digging into the Exposed Terminal Problem

- ▶ Problem
 - ▶ Exposed terminal solution doesn't consider CTS at node C
- ▶ 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

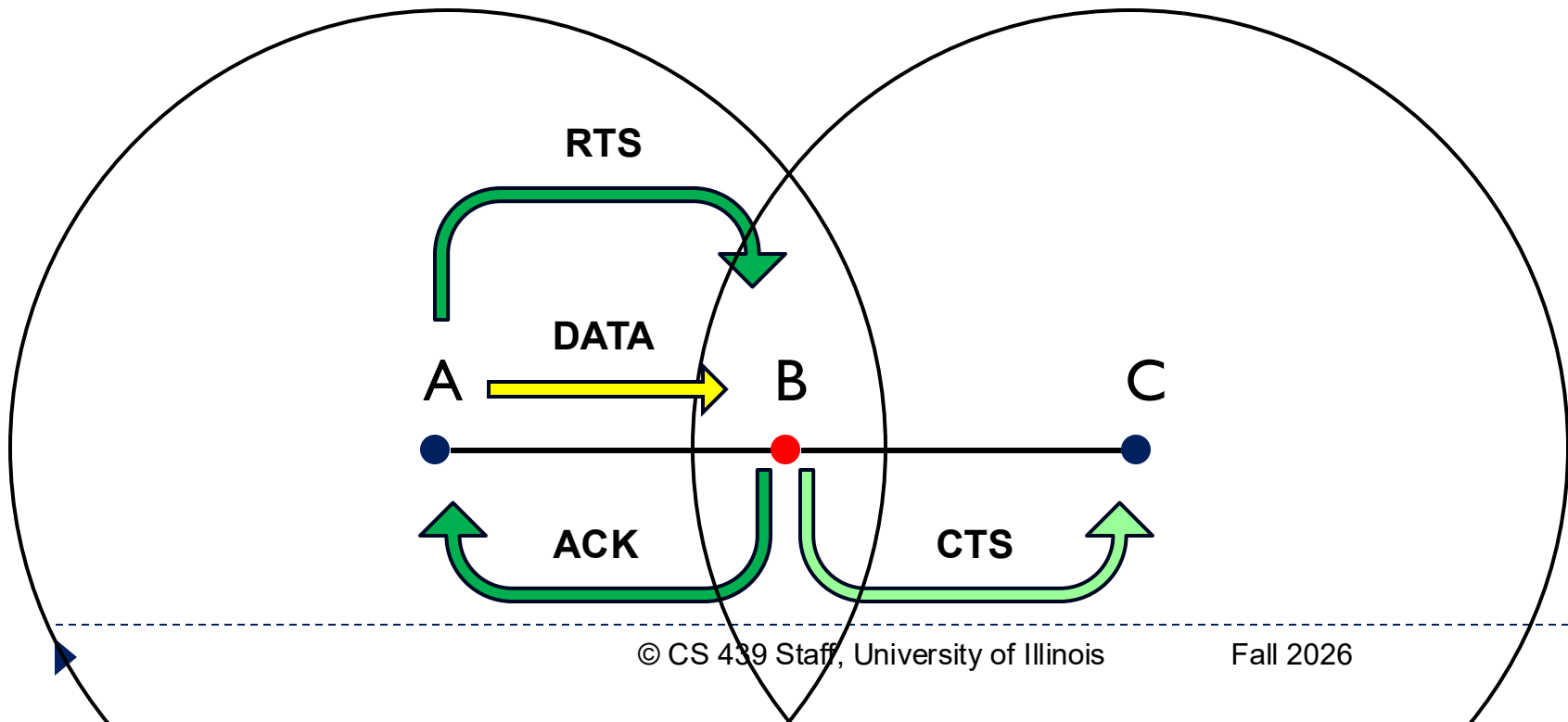


Reliability

- ▶ 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
- ▶ MACAW – Multiple Access with Collision Avoidance for Wireless

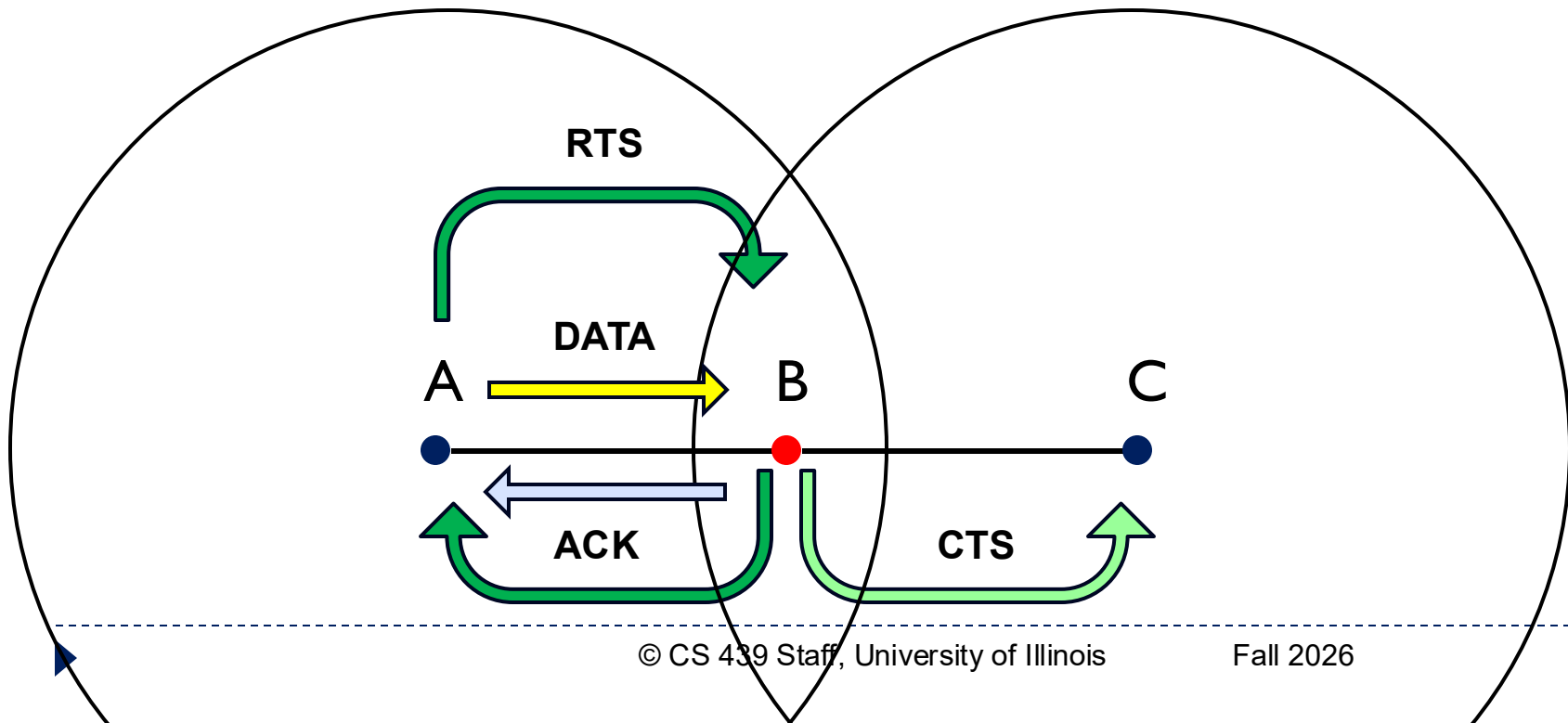
A Simple Solution to Improve Reliability - MACAW

- ▶ When B receives a data packet from A, B sends an Acknowledgement (ACK)



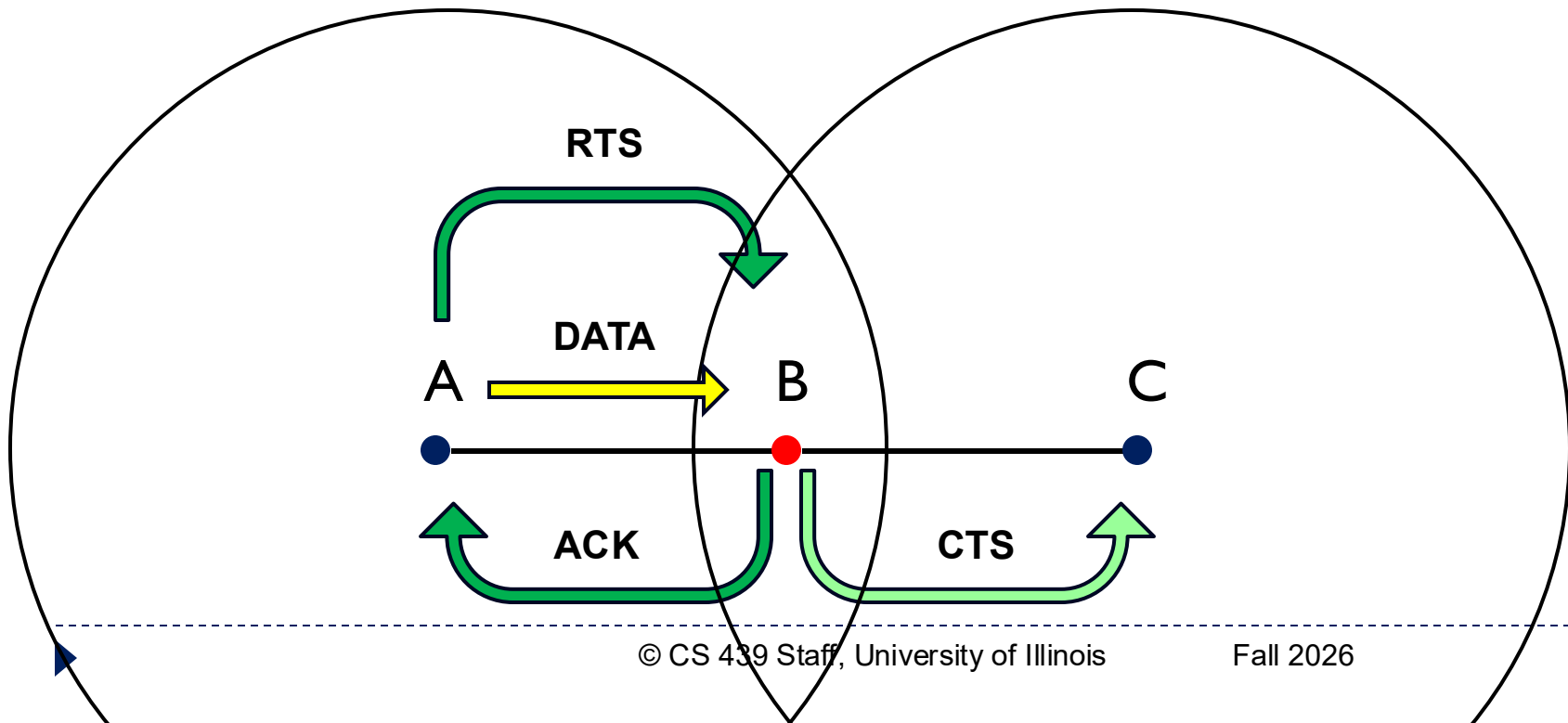
A Simple Solution to Improve Reliability - MACAW

- ▶ When B receives a data packet from A, B sends an Acknowledgement (ACK)



A Simple Solution to Improve Reliability - MACAW

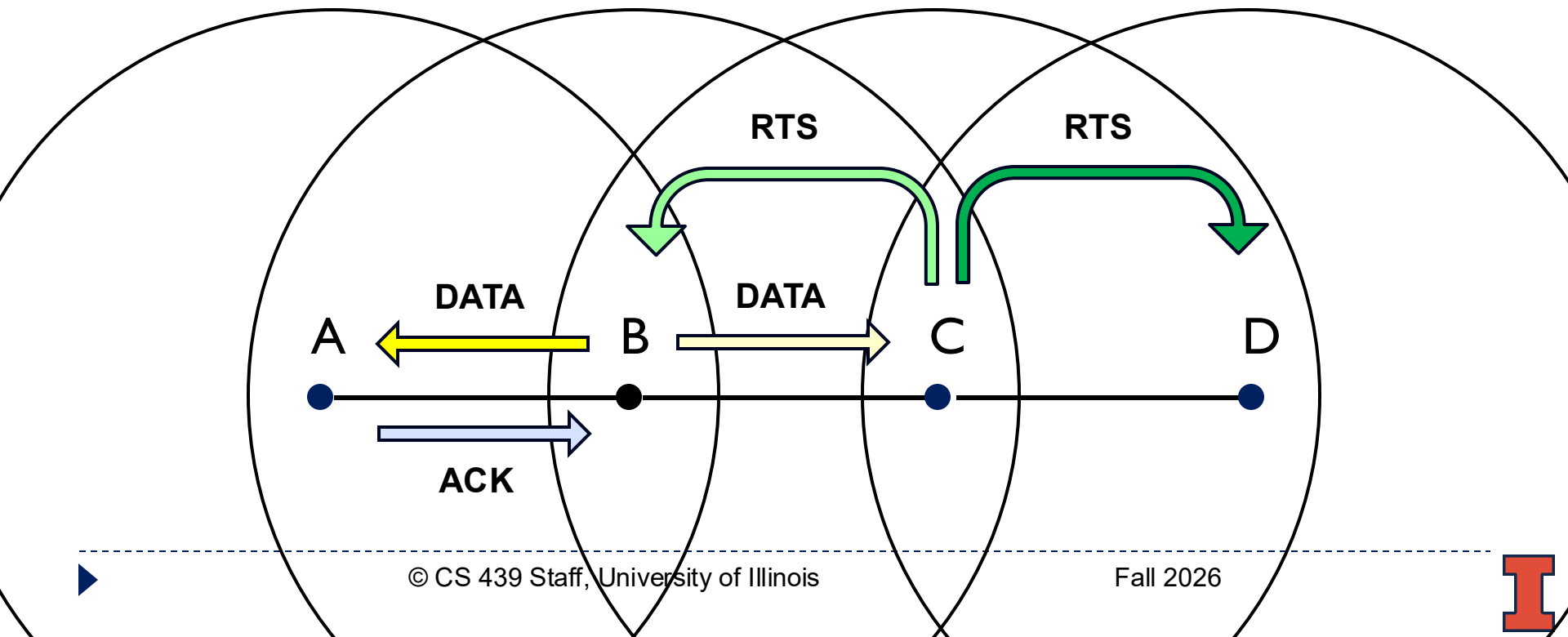
- ▶ When B receives a data packet from A, B sends an Acknowledgement (ACK)
- ▶ If node A fails to receive an ACK
 - ▶ Retransmit the packet using RTS/CTS/DATA/ACK sequence



Digging further into the Exposed Terminal Problem

► Problem

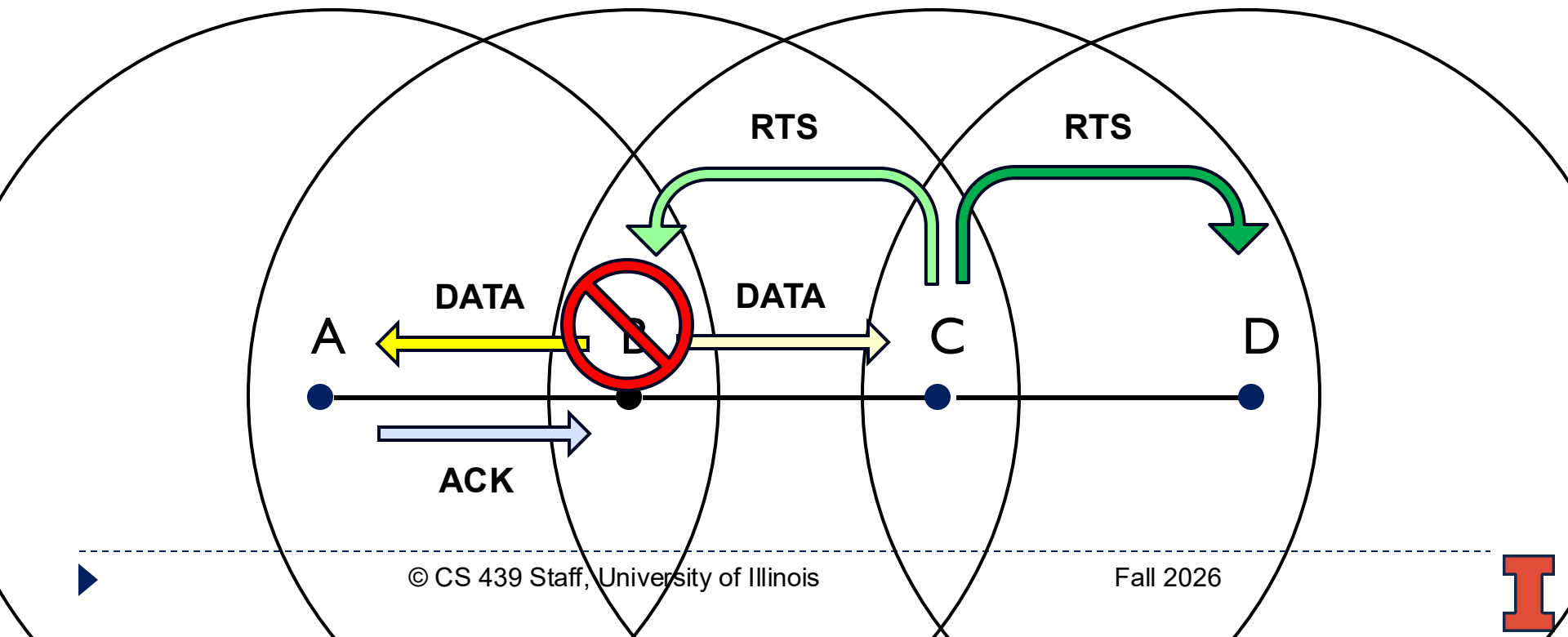
- Exposed terminal solution doesn't consider ACK at node B



Digging further into the Exposed Terminal Problem

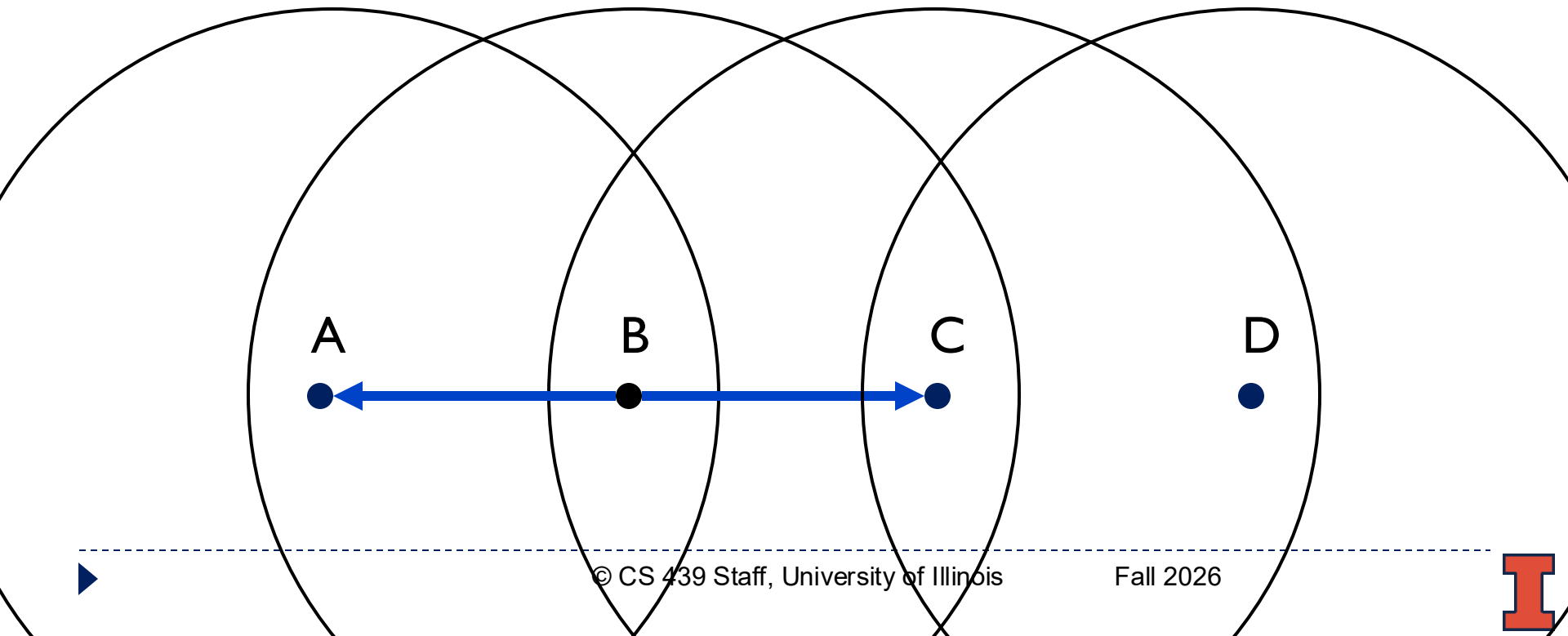
► Problem

- Exposed terminal solution doesn't consider ACK at node B
- C doesn't wait since it doesn't hear A's CTS
 - C's RTS could collide with A's ACK
 - Failure even though DATA was received correctly



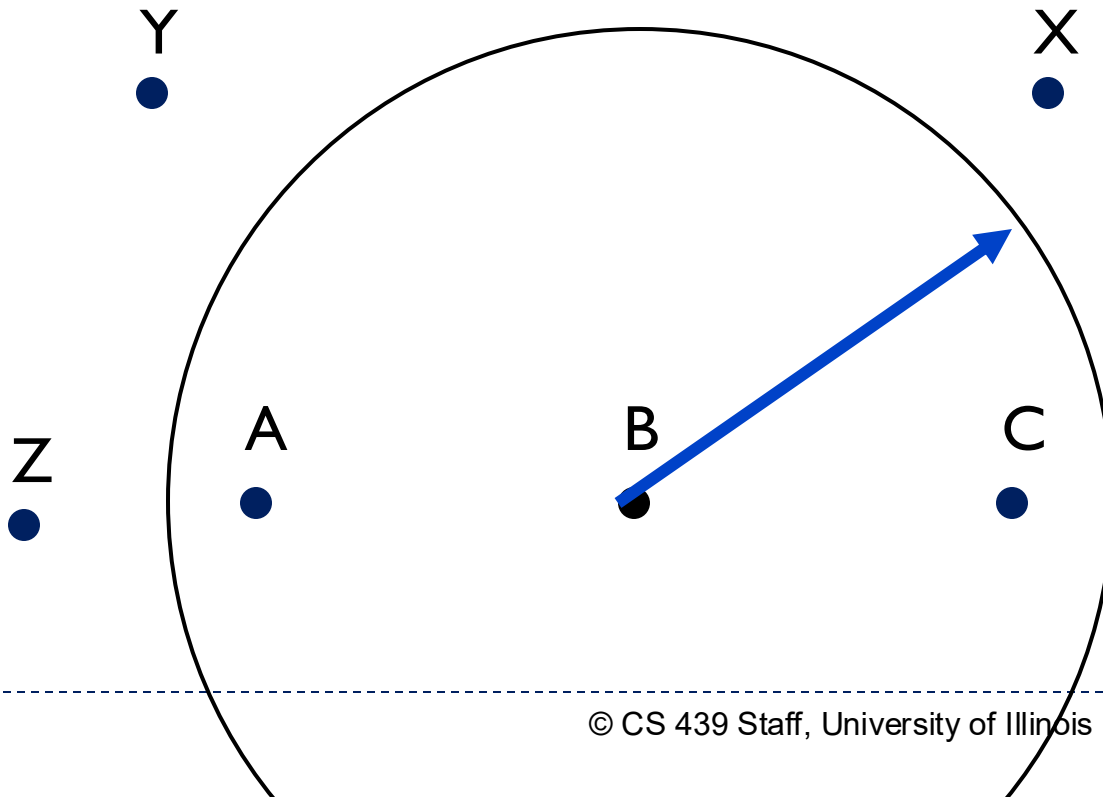
Exposed Terminal Problem Assumptions

- ▶ What is wrong with this diagram?



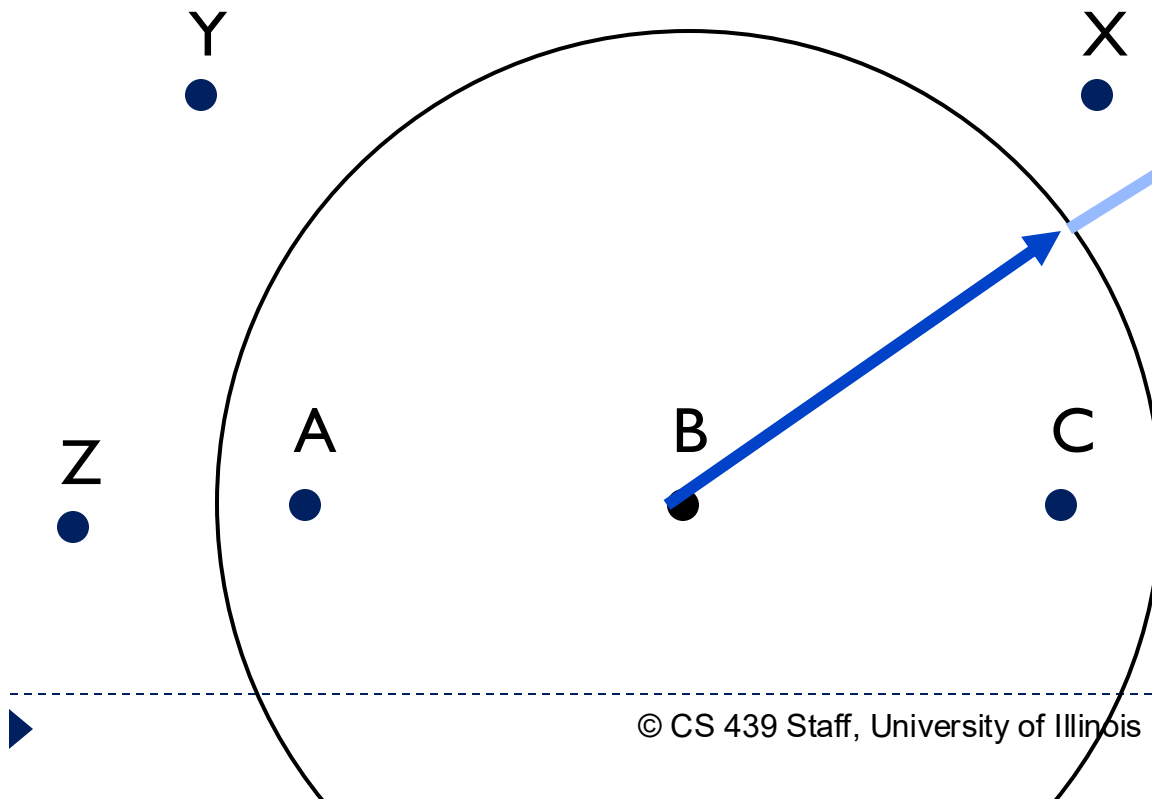
Exposed Terminal Problem Assumptions

- ▶ What is wrong with this diagram?



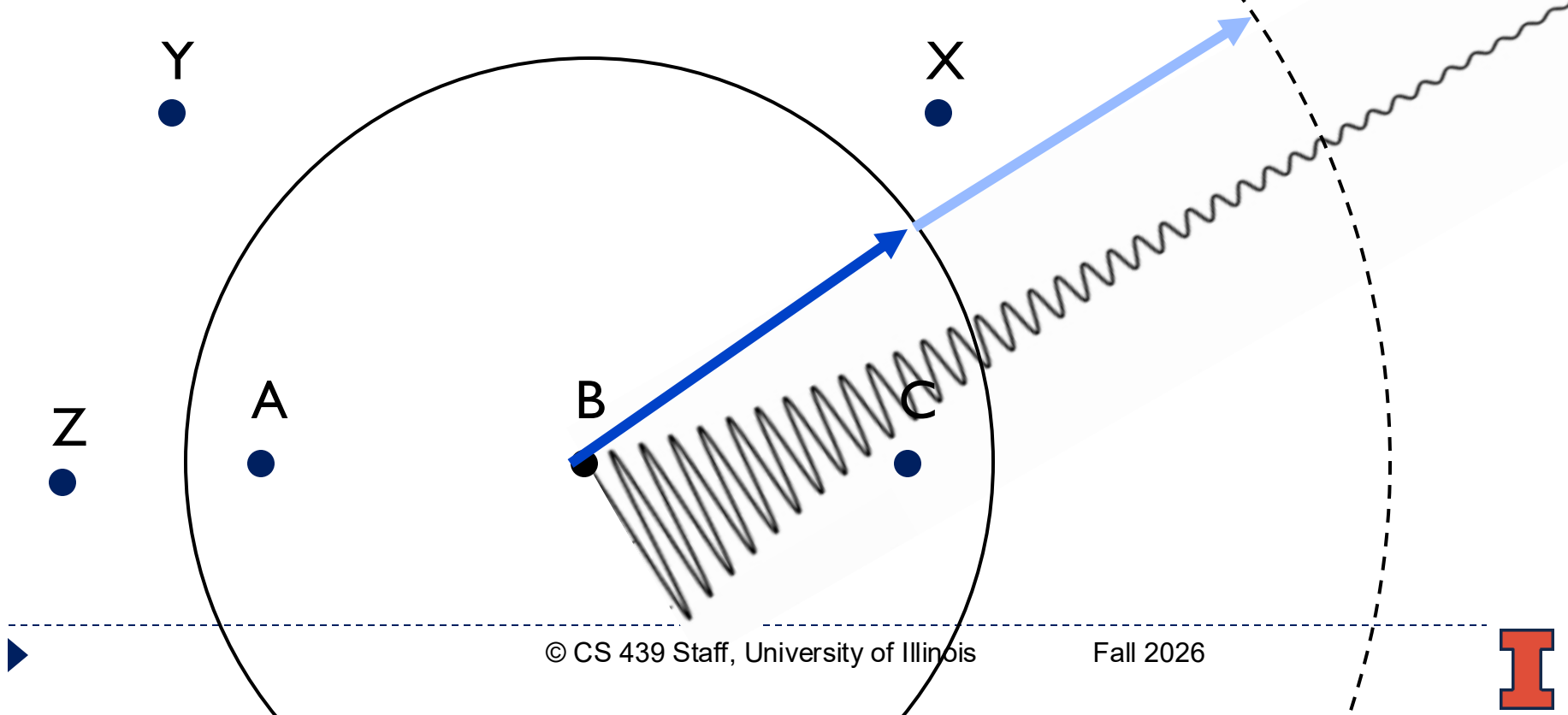
Exposed Terminal Problem Assumptions

- ▶ The signal does not abruptly stop at the solid line!



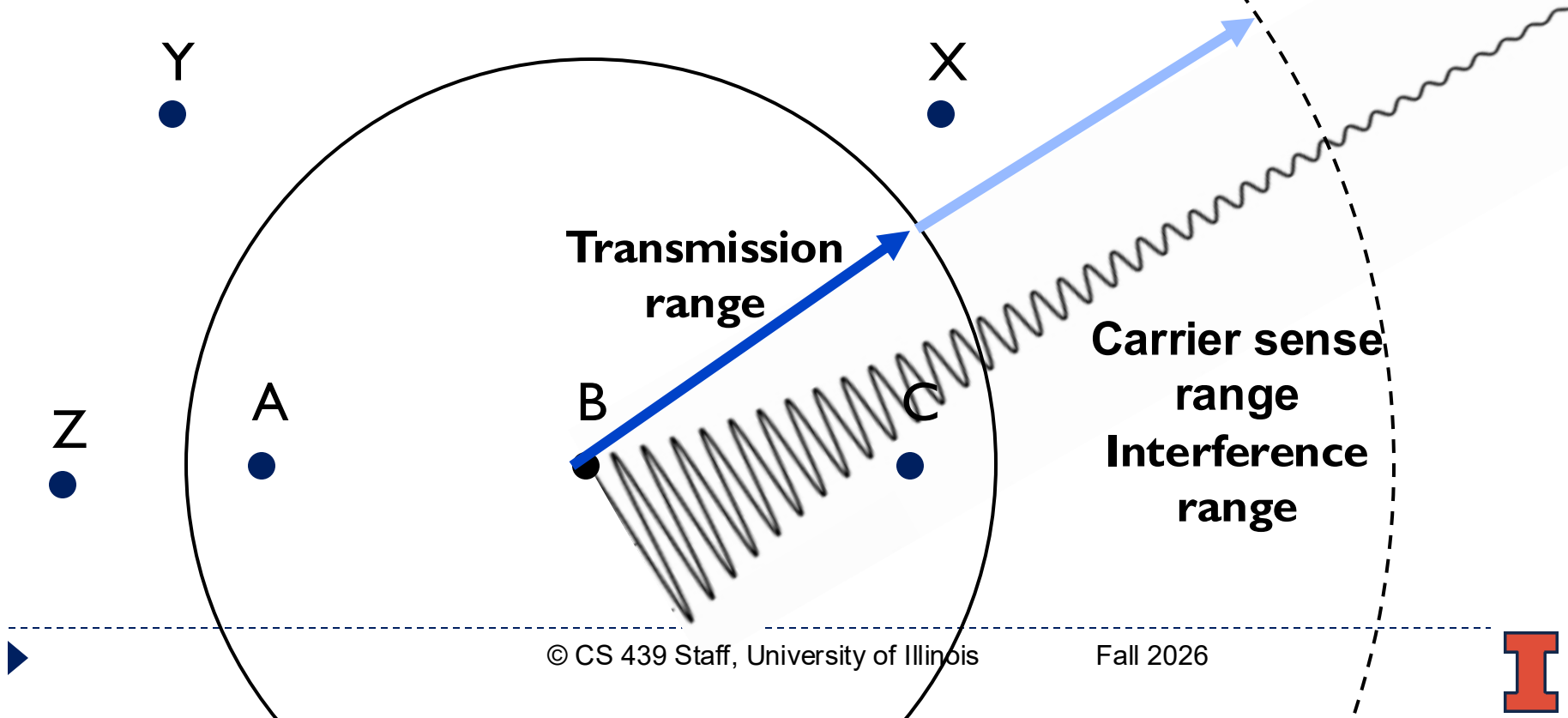
Exposed Terminal Problem Assumptions

- ▶ The signal does not abruptly stop at the solid line!



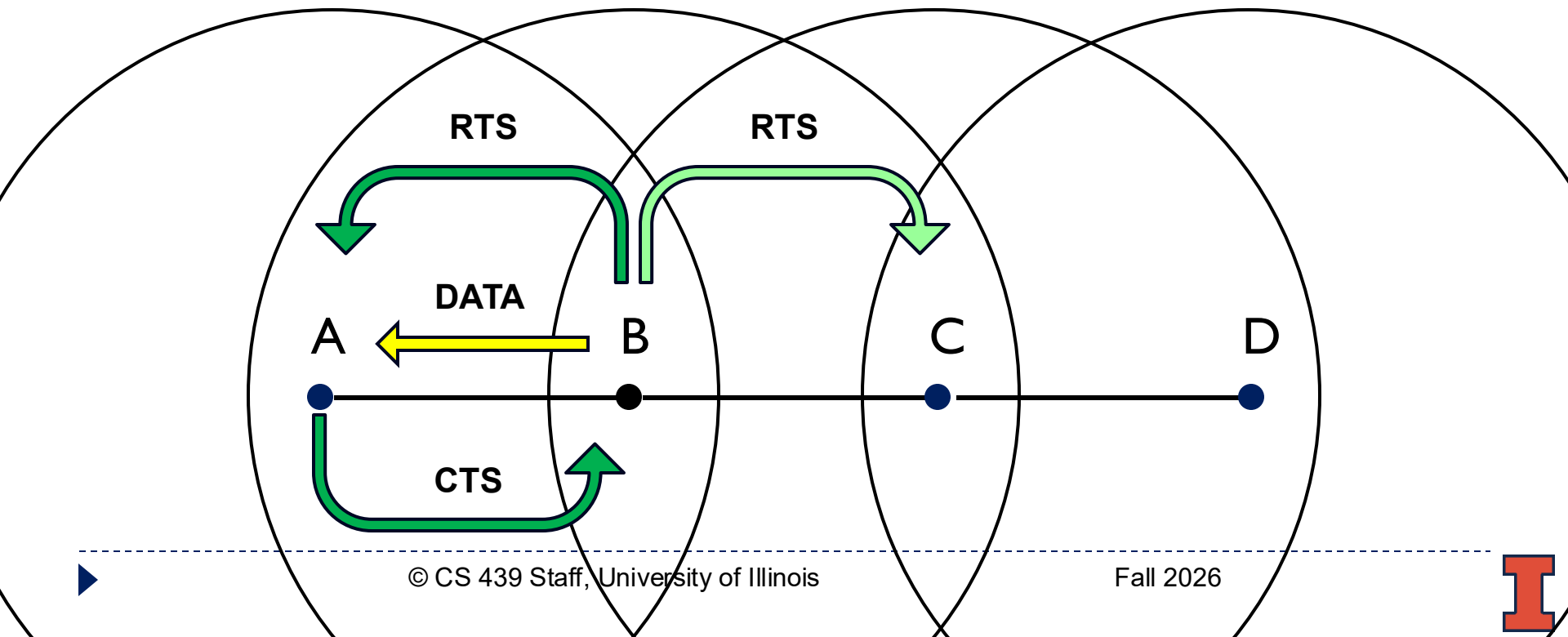
Exposed Terminal Problem Assumptions

- ▶ How does this affect the Exposed terminal Problem?



Exposed Terminal Problem

- ▶ Although D cannot decode the DATA from A, it can detect the message at a lower signal level



Another MACAW Proposal

- ▶ **Reduce collisions**

- ▶ Have all nodes use carrier sense before RTS
 - ▶ This really is the best way to go!

- ▶ **Additional proposal**

- ▶ A node that is deferring based on carrier sensing cannot decode the message
- ▶ Includes length of DATA in RTS and CTS so other nodes know how long to wait
 - We'll see something like this in 802.11 NAV

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



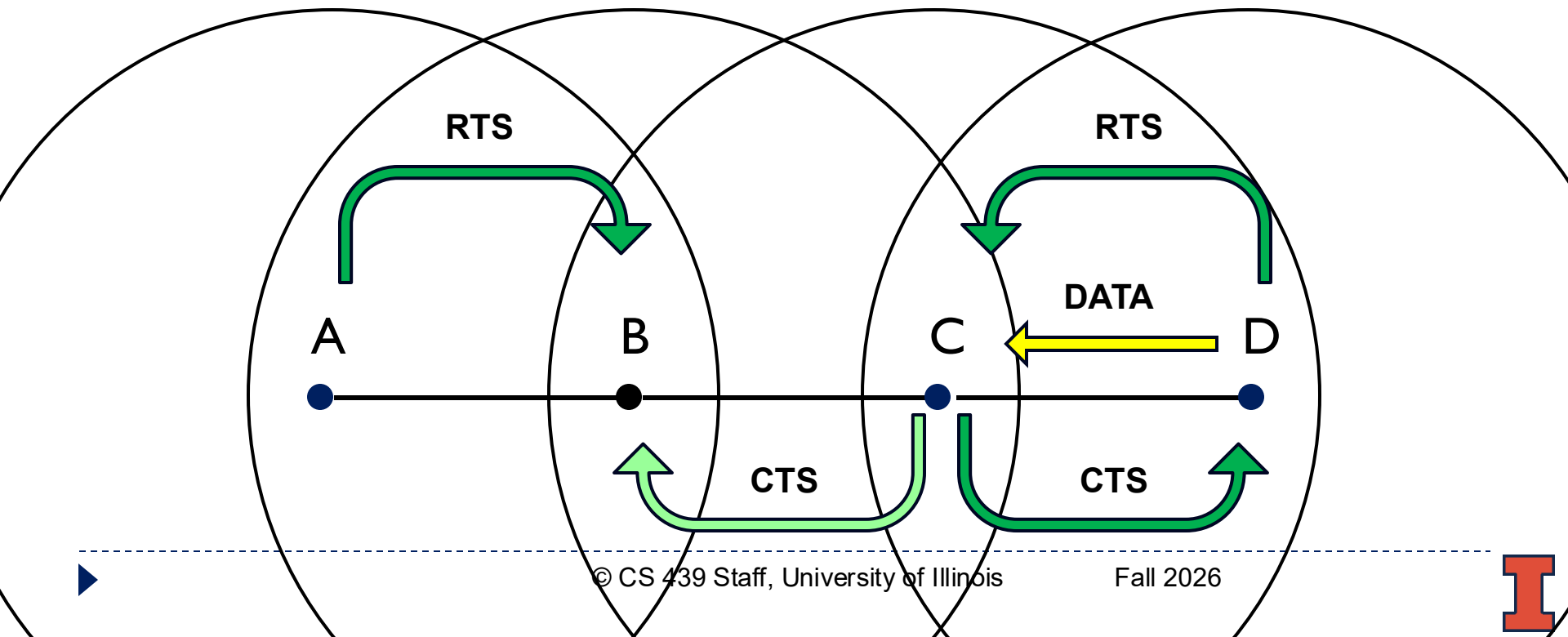
Backoff Algorithm

- ▶ **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!



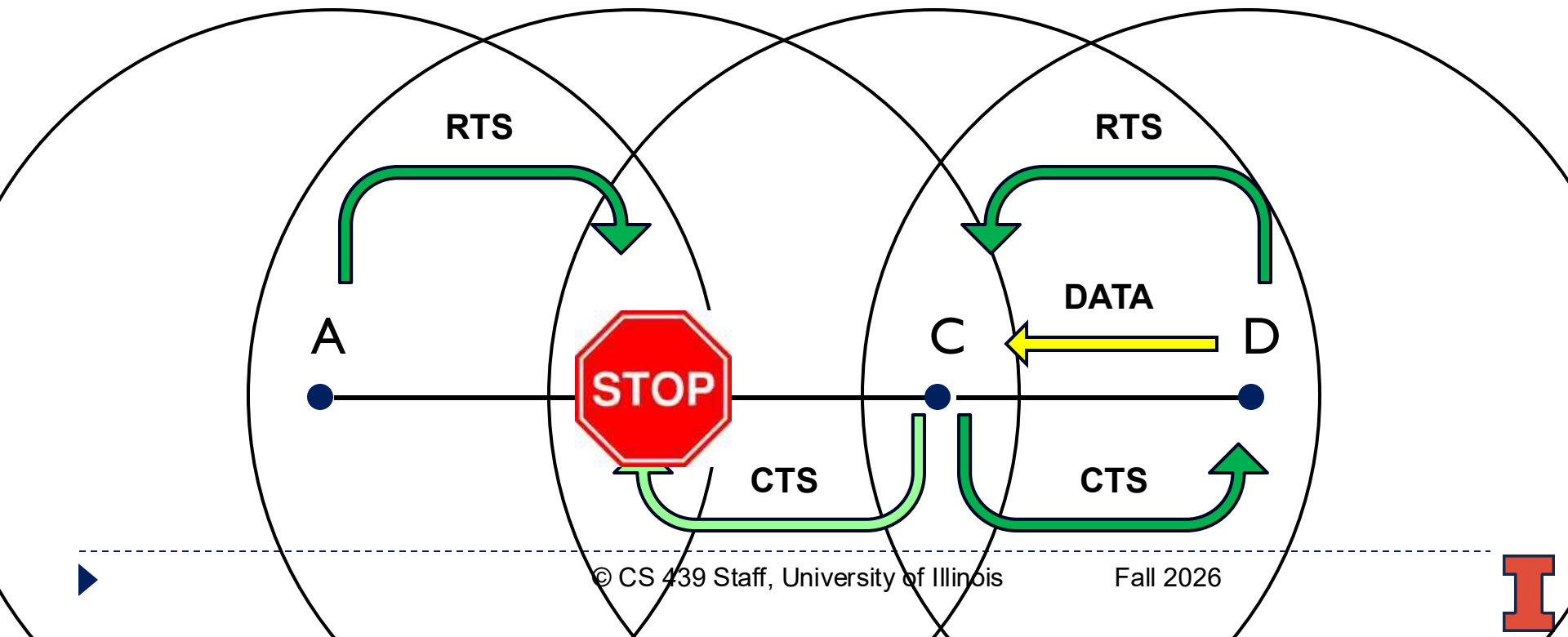
New Problem: Deafness

- ▶ A sends an RTS to B, while C is receiving from D



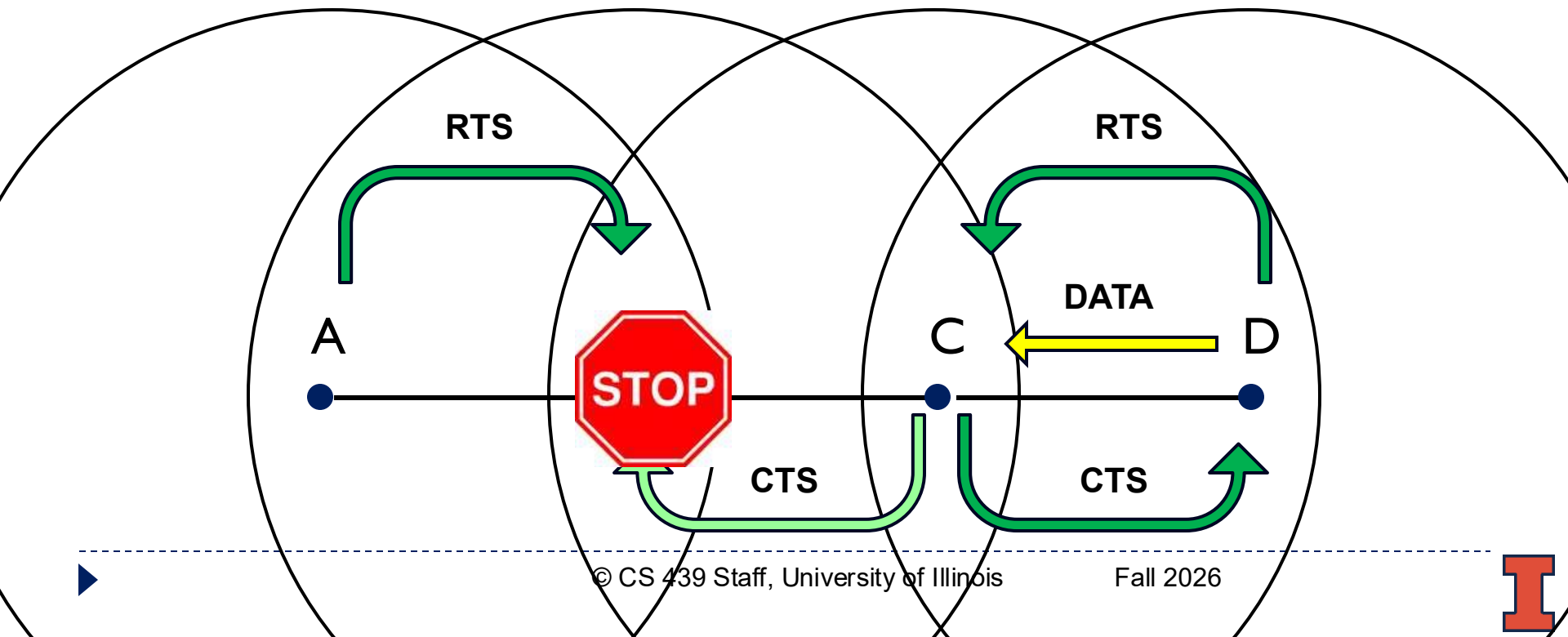
New Problem: Deafness

- ▶ A sends an RTS to B, while C is receiving from D
- ▶ B cannot reply with a CTS
 - ▶ B knows that D is sending to C



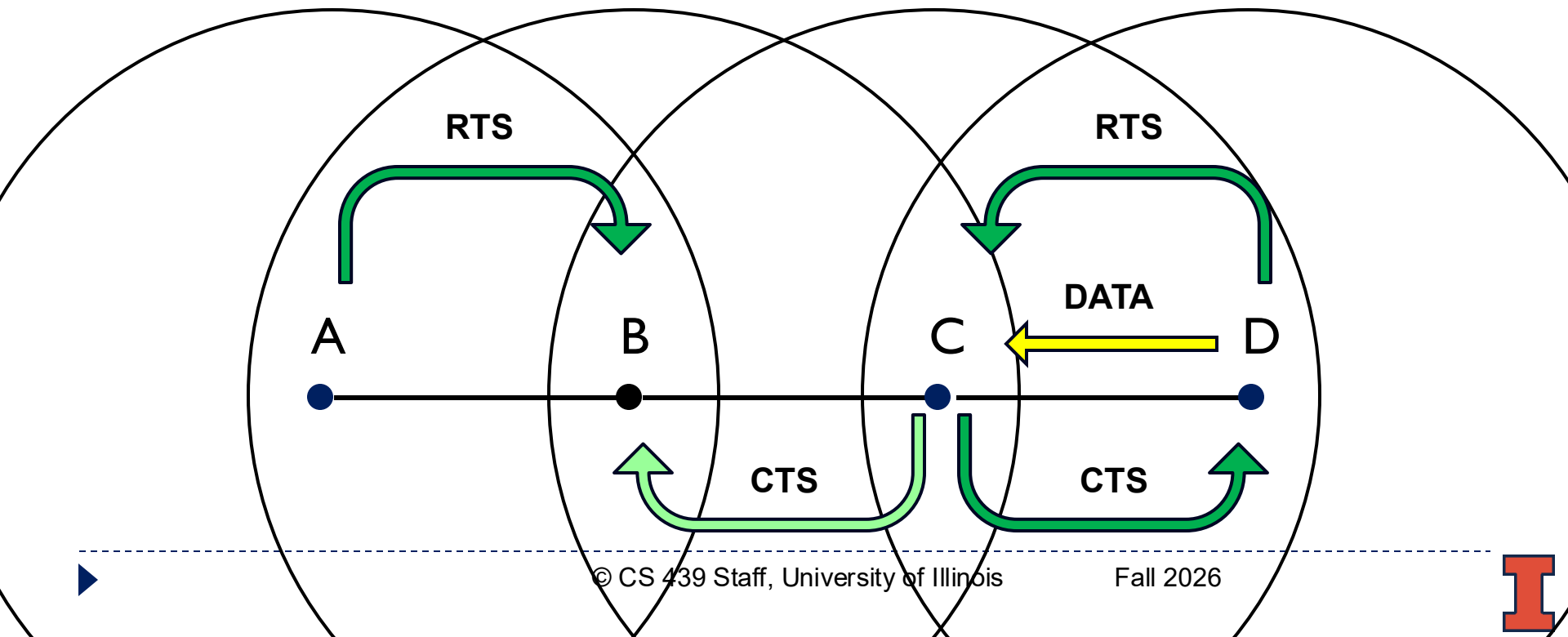
New Problem: Deafness

- ▶ A sends an RTS to B, while C is receiving from D
- ▶ B cannot reply with a CTS
 - ▶ B knows that D is sending to C
 - ▶ A keeps retransmitting RTS, increasing its own BO timeout



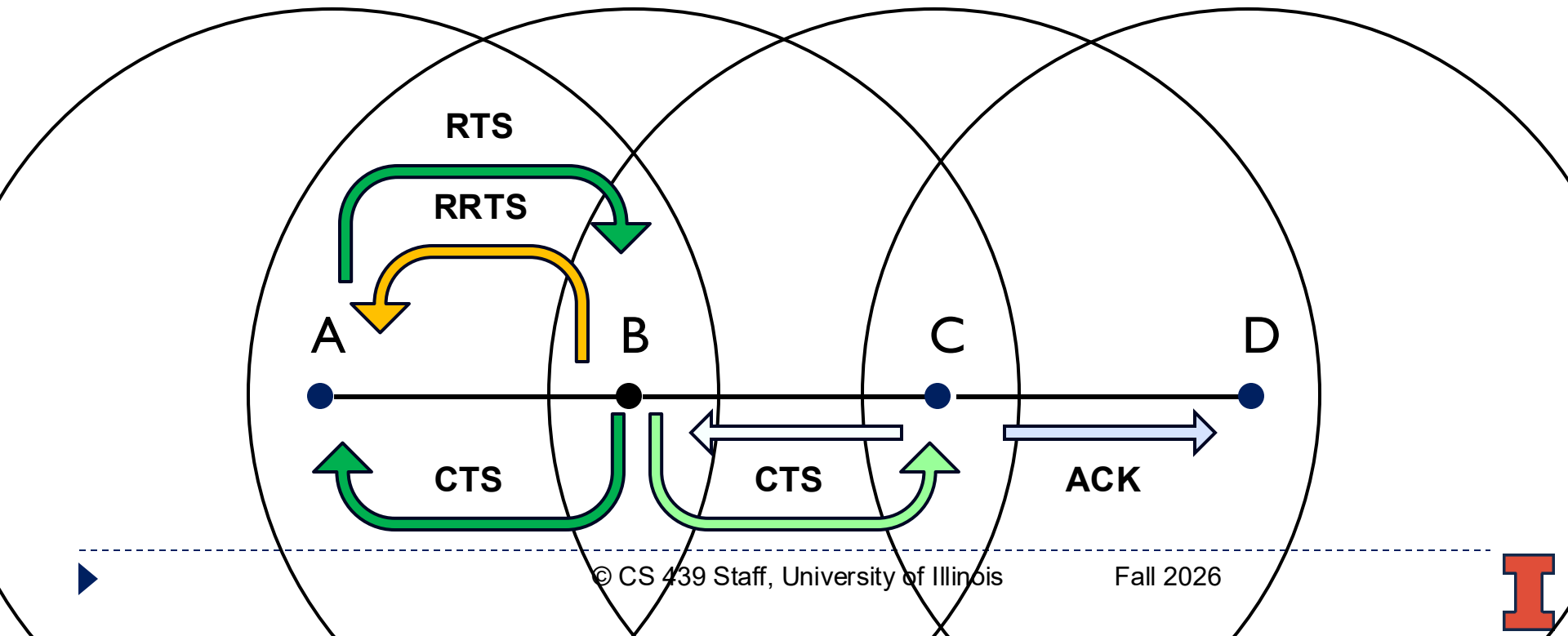
Proposal: Request for RTS - MACAW

- ▶ Have B do contention on behalf of A
 - ▶ If B receives RTS, it must defer CTS reply



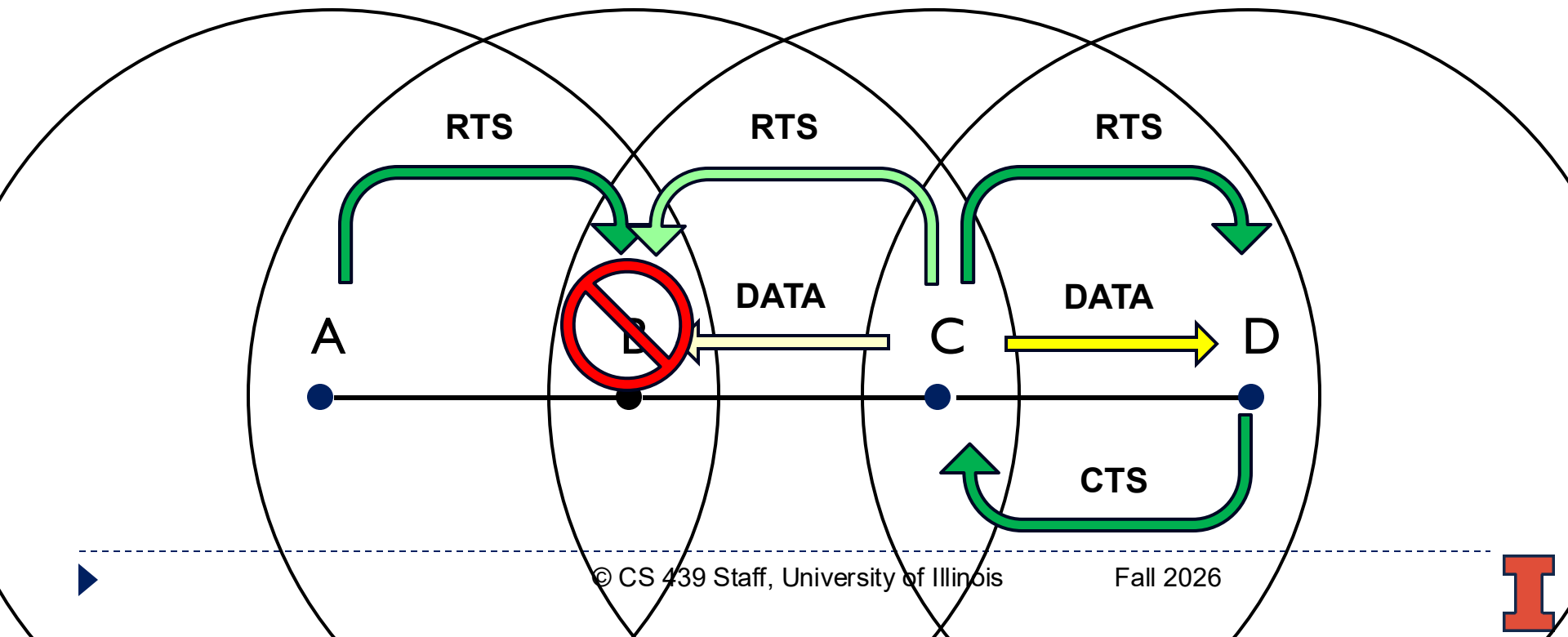
Proposal: Request for RTS - MACAW

- ▶ Have B do contention on behalf of A
 - ▶ If B receives RTS, it must defer CTS reply
 - ▶ After hearing C's ACK to D, B sends RRTS to A
 - ▶ A responds by starting normal RTS-CTS
 - ▶ Others hearing RRTS defer long enough for RTS-CTS



Problem: Request for RTS - MACAW

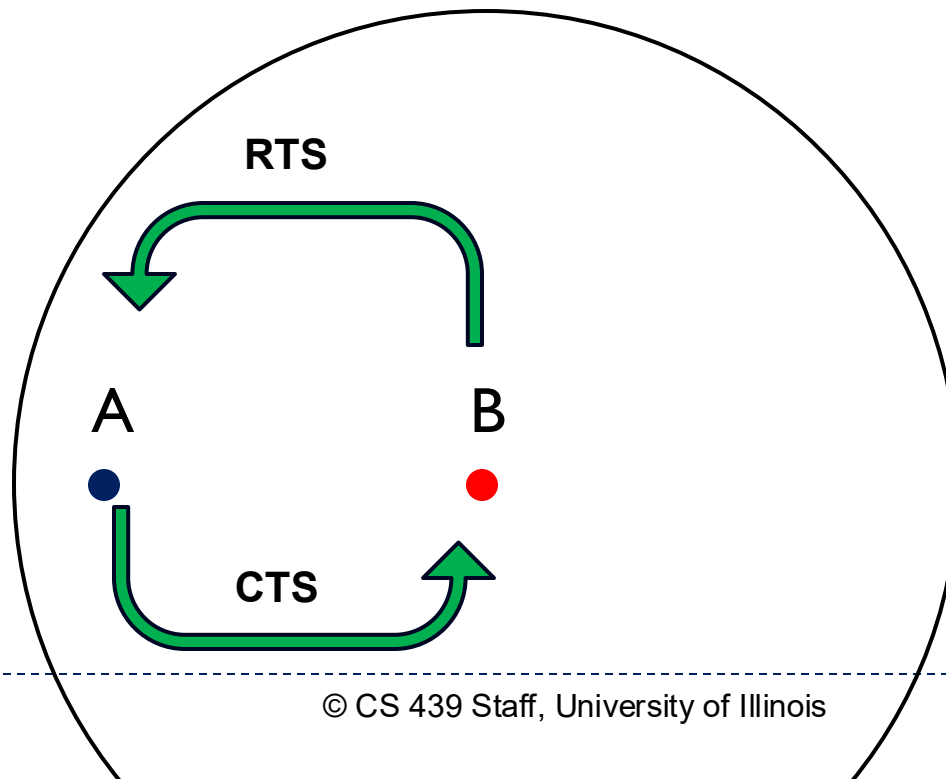
- ▶ B may not receive the RTS from A at all, due to interference with transmission from C
- ▶ A waits for RRTS that never comes



Broadcast/Multicast

► Problem

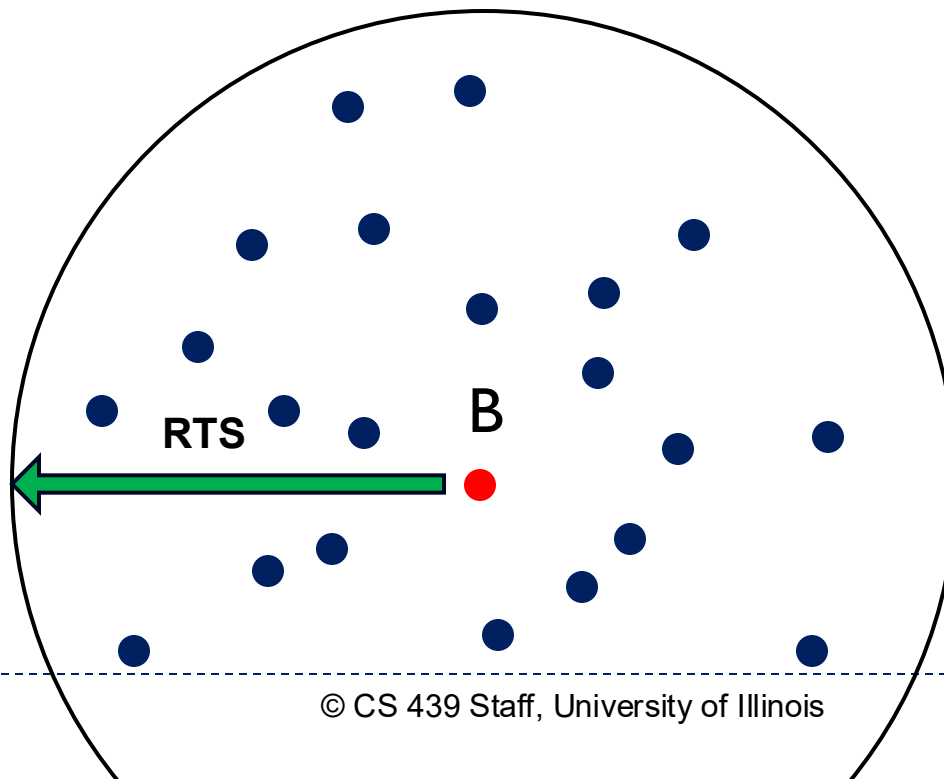
- Basic RTS-CTS works for unicast transmissions



Broadcast/Multicast

► Problem

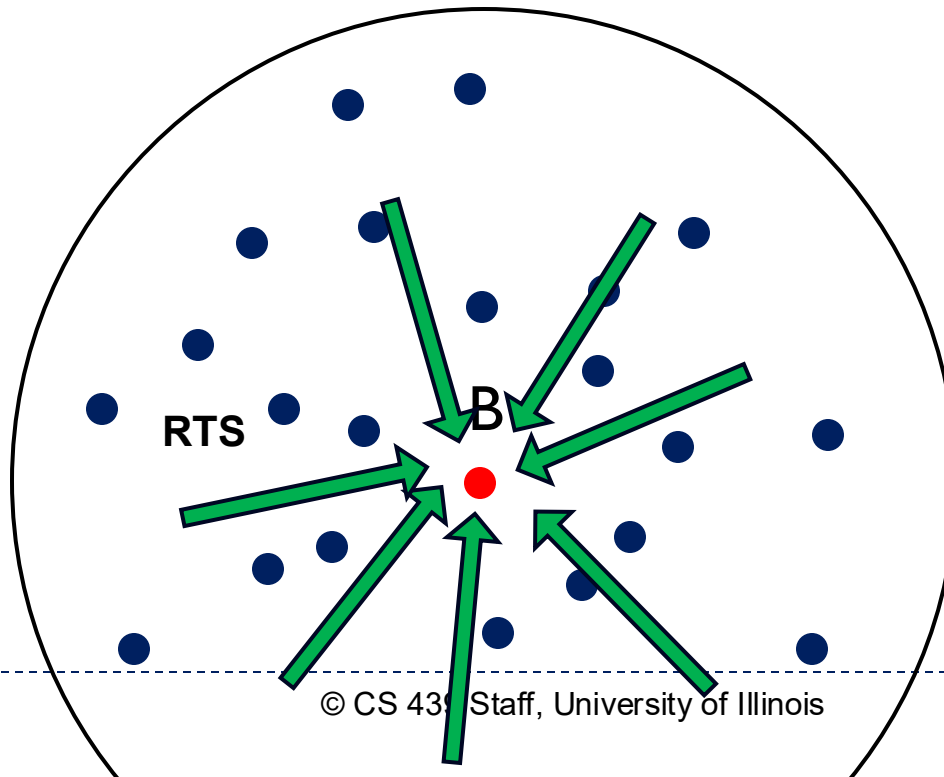
- What would happen here?



Broadcast/Multicast

► Problem

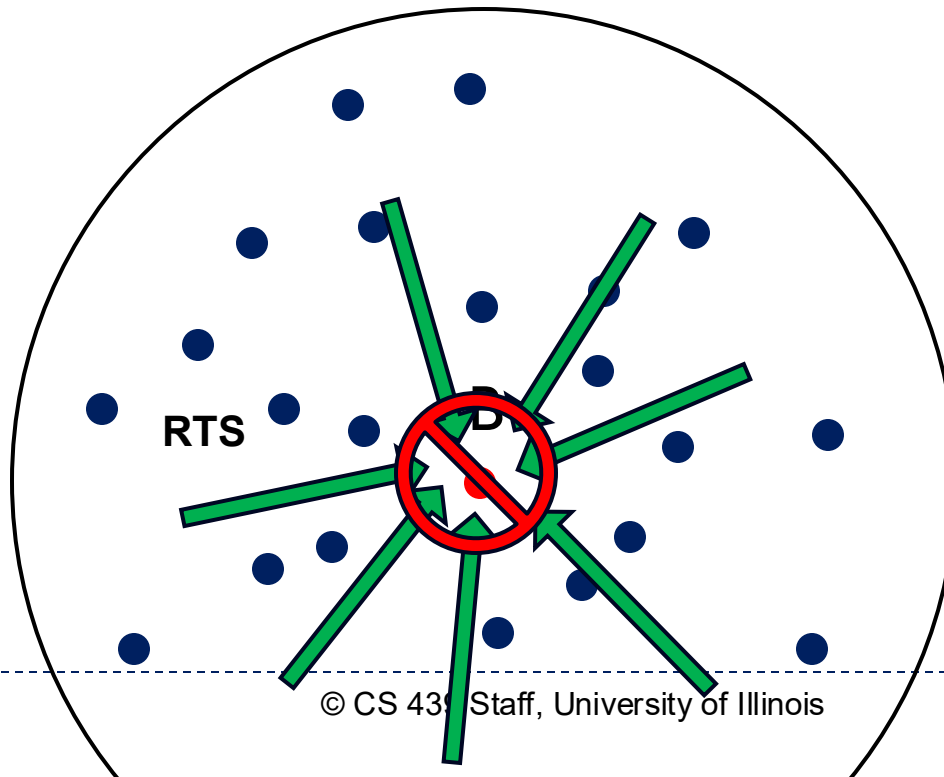
- RTS would trigger CTS from all receivers



Broadcast/Multicast

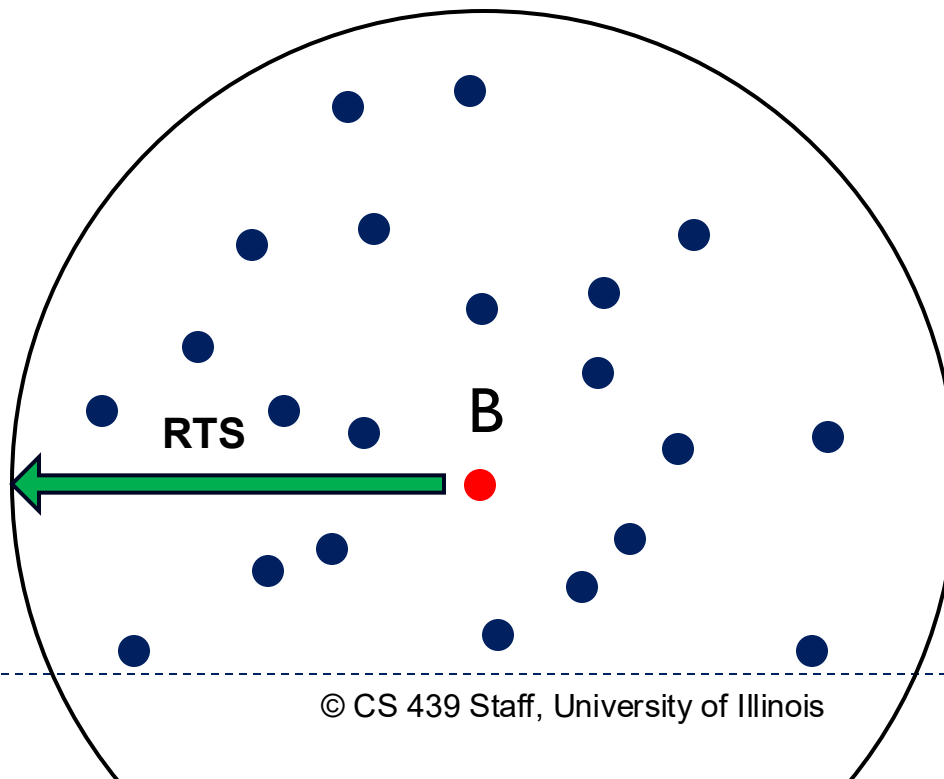
► Problem

- RTS would trigger CTS from all receivers
- Ack implosion!



Broadcast/Multicast

► Thoughts for solutions?



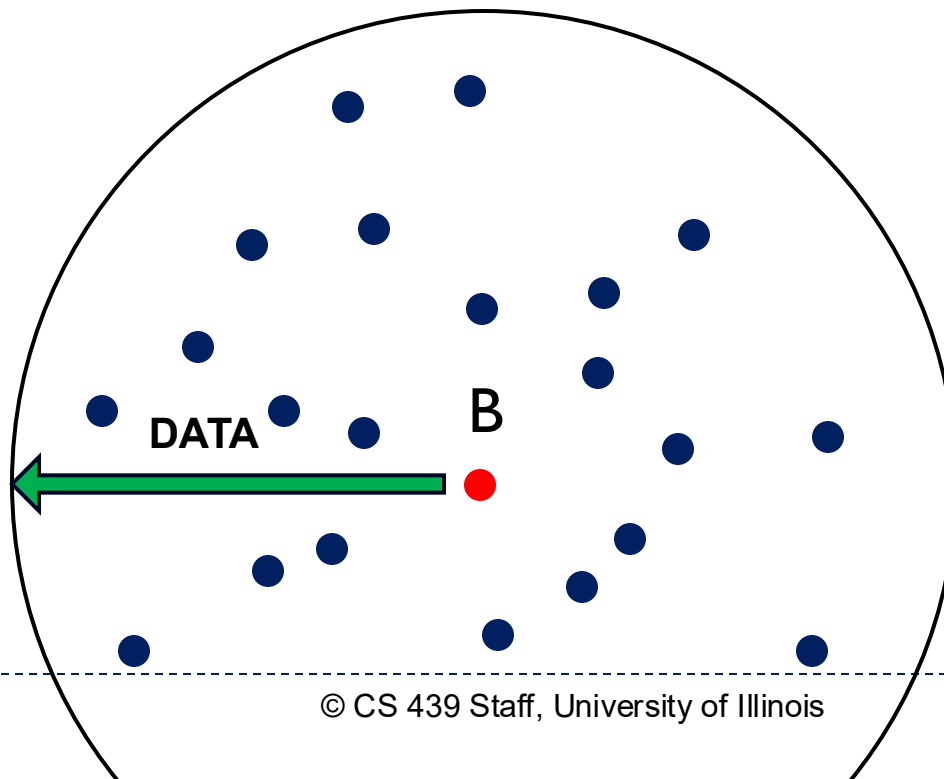
Broadcast - MACAW

- ▶ **Sort-of solution**
 - ▶ Don't use CTS for broadcast
- ▶ **Receivers recognize broadcast 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



Broadcast/Multicast

- ▶ **Better solution (but not perfect)**
 - ▶ Don't use RTS or CTS for broadcast
 - ▶ Immediately send DATA



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)

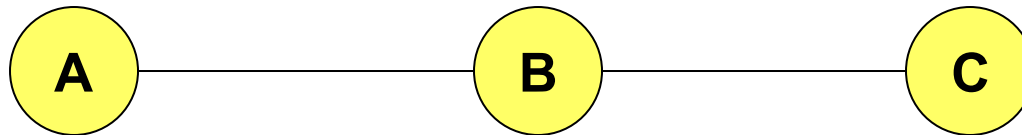
IEEE 802.11 Wireless MAC

- ▶ 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

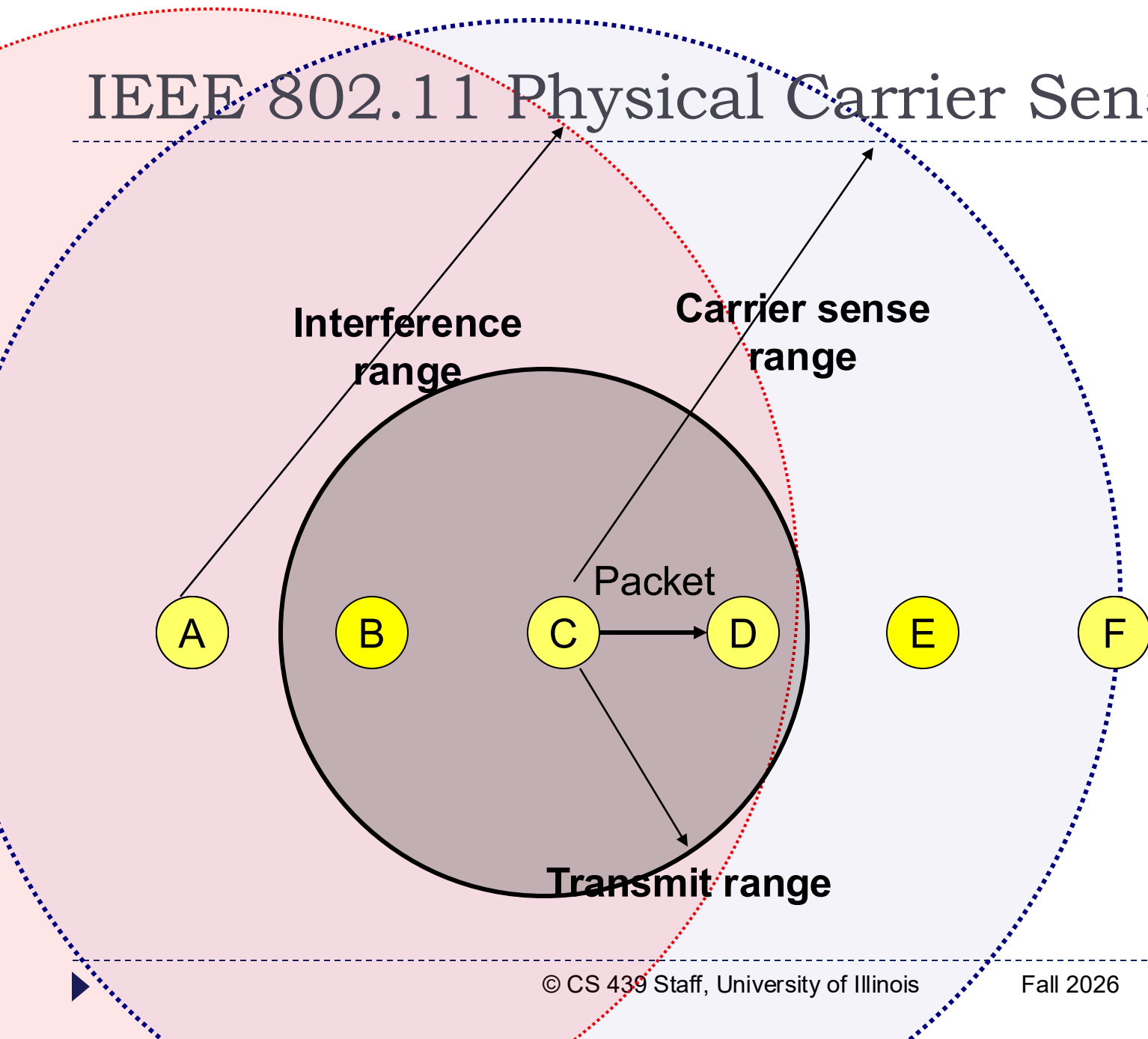


IEEE 802.11 CSMA/CA

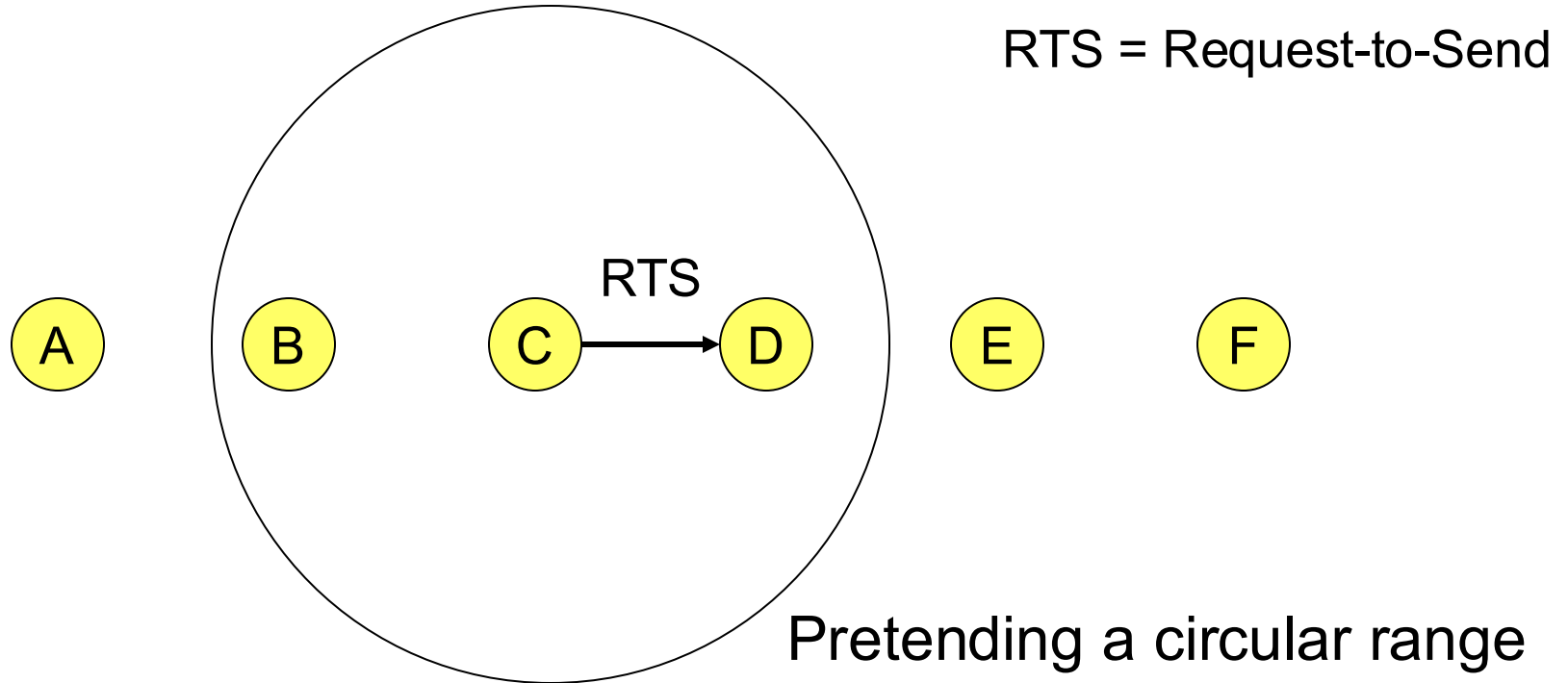
- ▶ Nodes stay silent when carrier sensed
 - ▶ Physical carrier sense
 - ▶ Virtual carrier sense
 - ▶ Network Allocation Vector (NAV)
 - ▶ NAV is updated based on overheard RTS/CTS/DATA/ACK packets, each of which specified duration of a pending transmission
- ▶ Backoff intervals used to reduce collision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



IEEE 802.11 Physical Carrier Sense



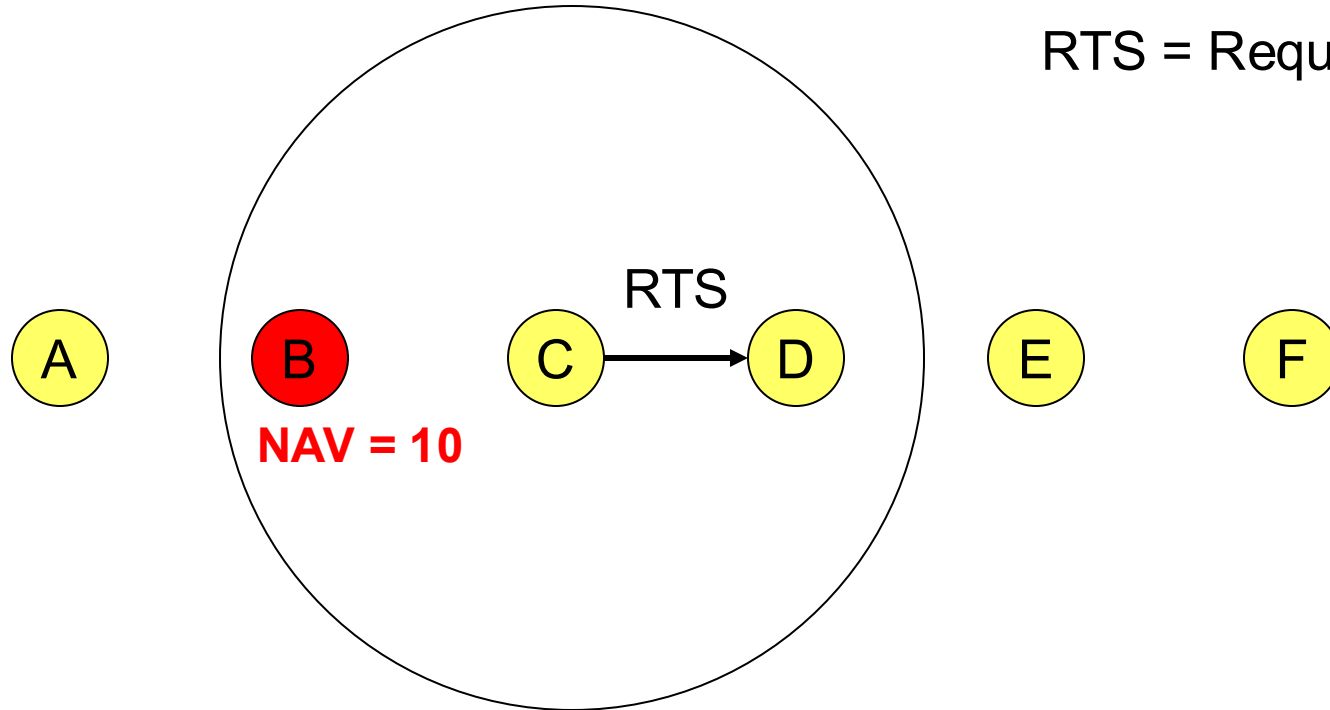
IEEE 802.11 Virtual Carrier Sense



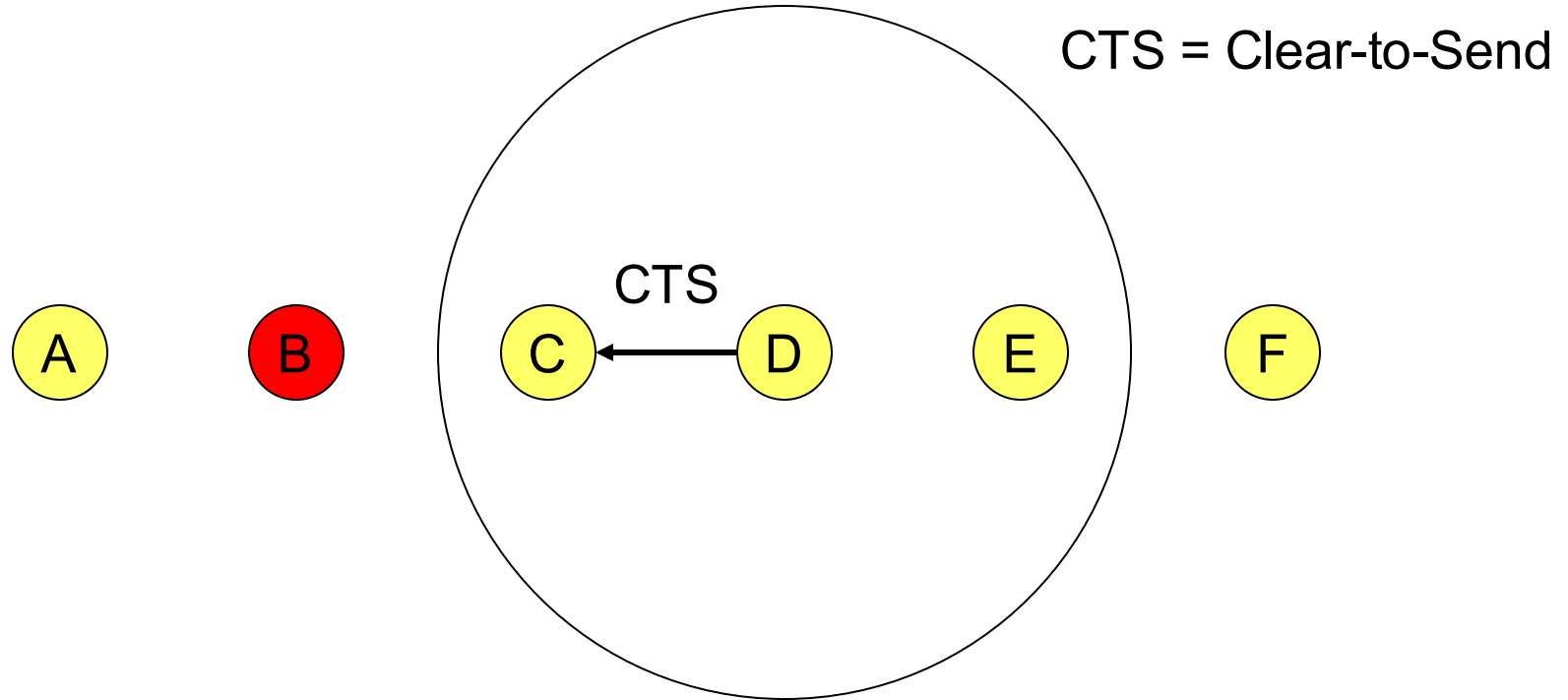
IEEE 802.11 Virtual Carrier Sense

NAV = remaining duration to keep quiet

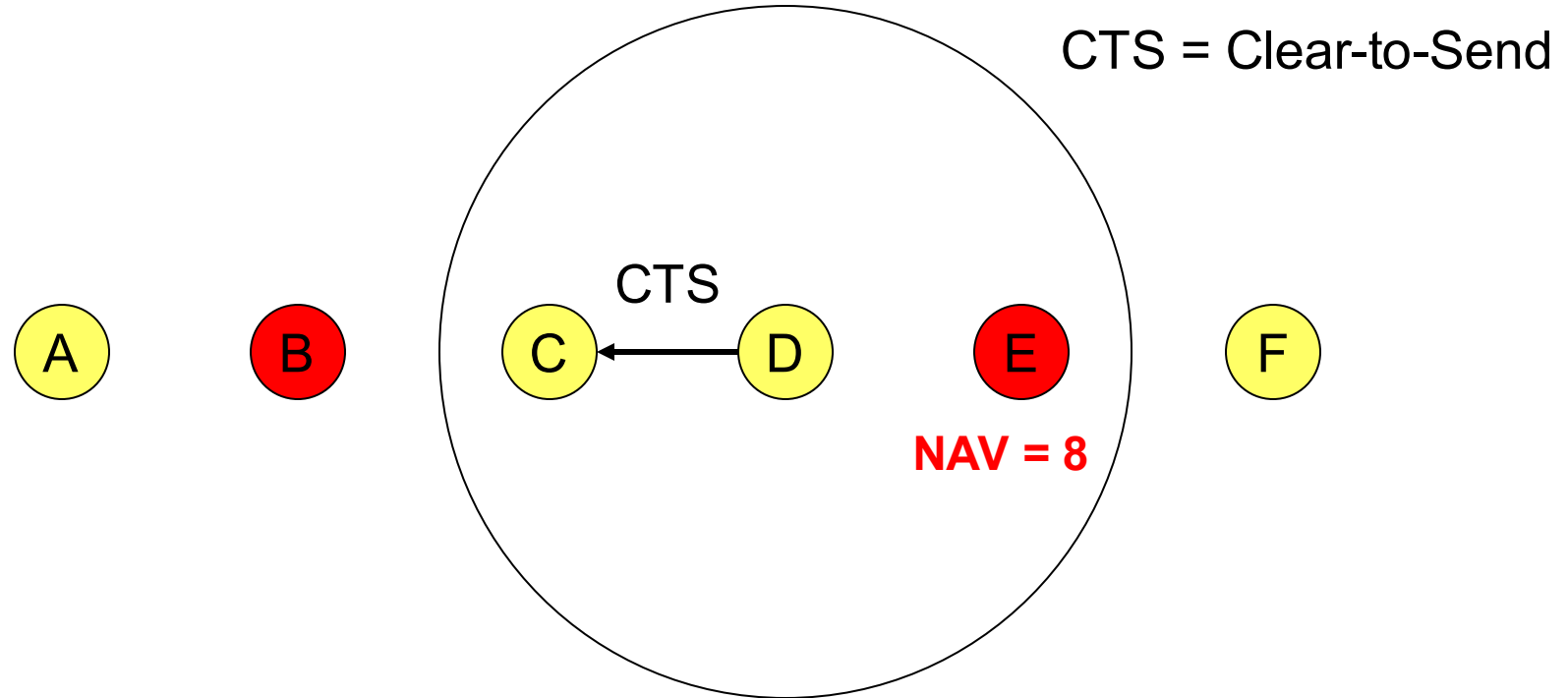
RTS = Request-to-Send



IEEE 802.11 Virtual Carrier Sense

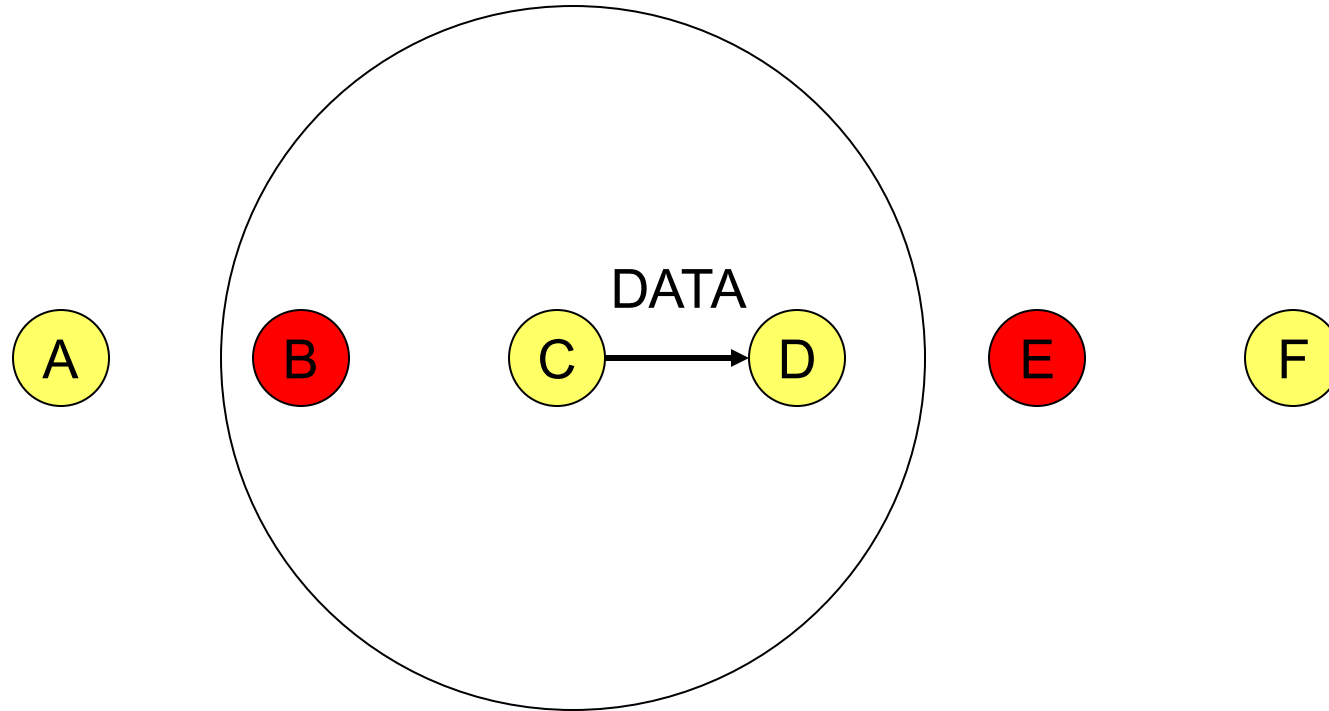


IEEE 802.11 Virtual Carrier Sense



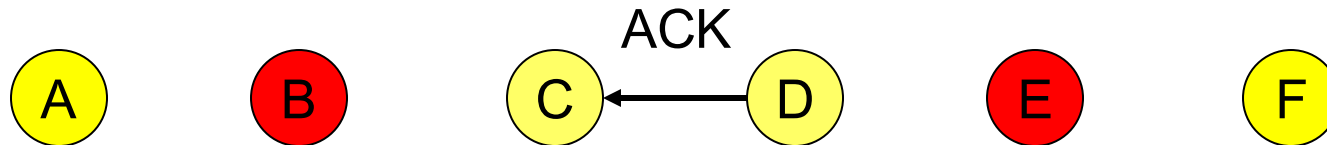
IEEE 802.11 Virtual Carrier Sense

- ▶ DATA packet follows CTS

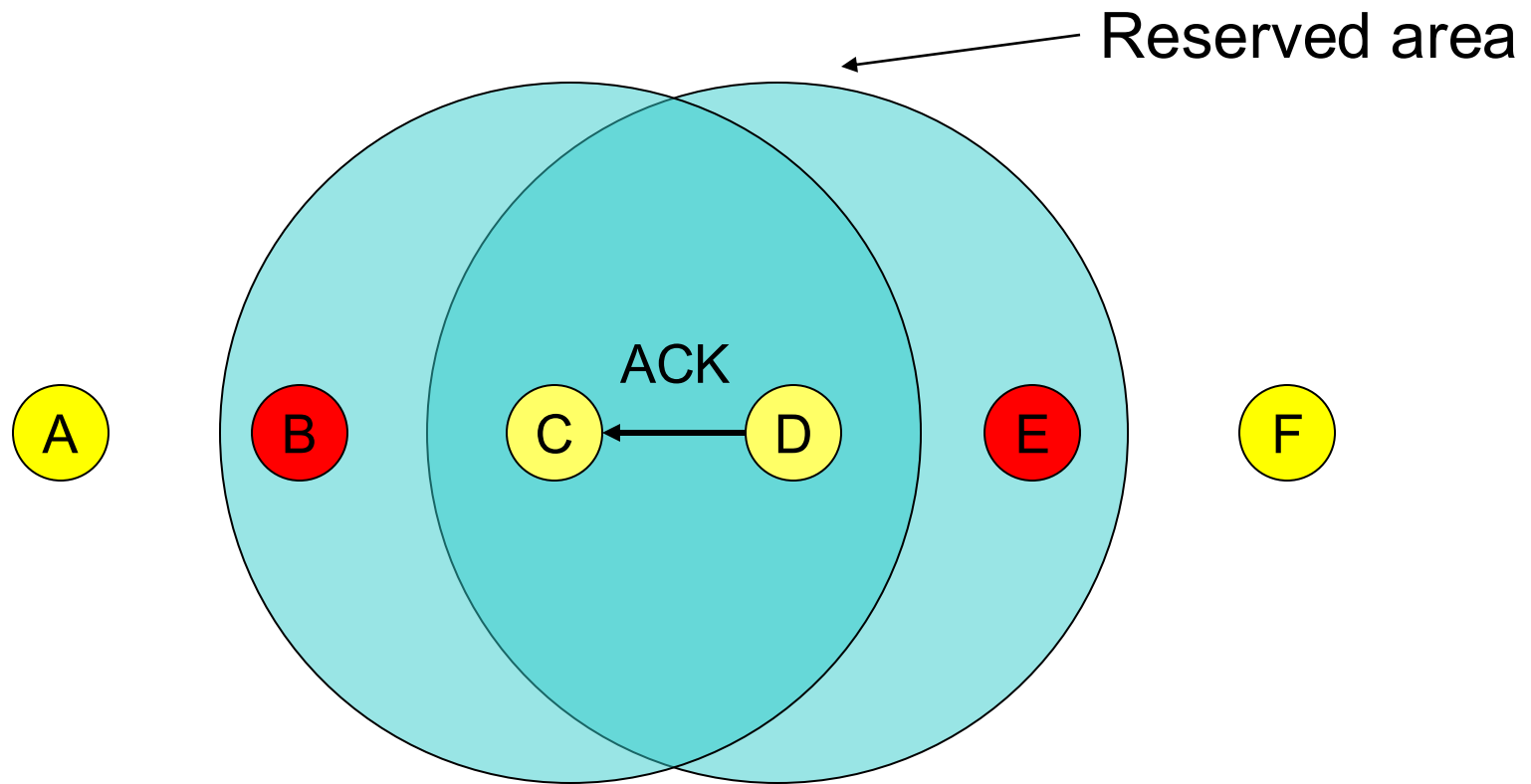


IEEE 802.11 Virtual Carrier Sense

- ▶ 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 μ s before sending again

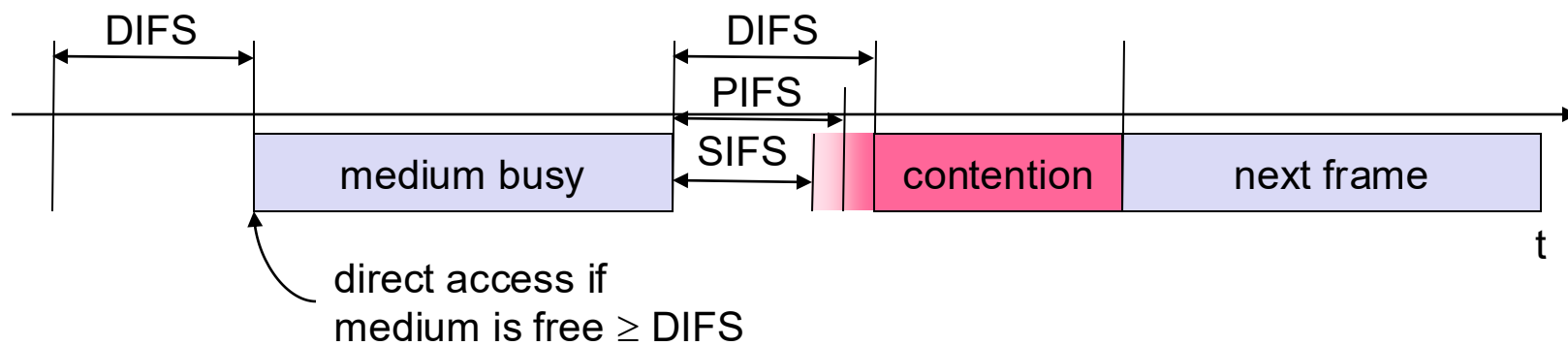
▶ If carrier is sensed

- ▶ When should a node transmit?



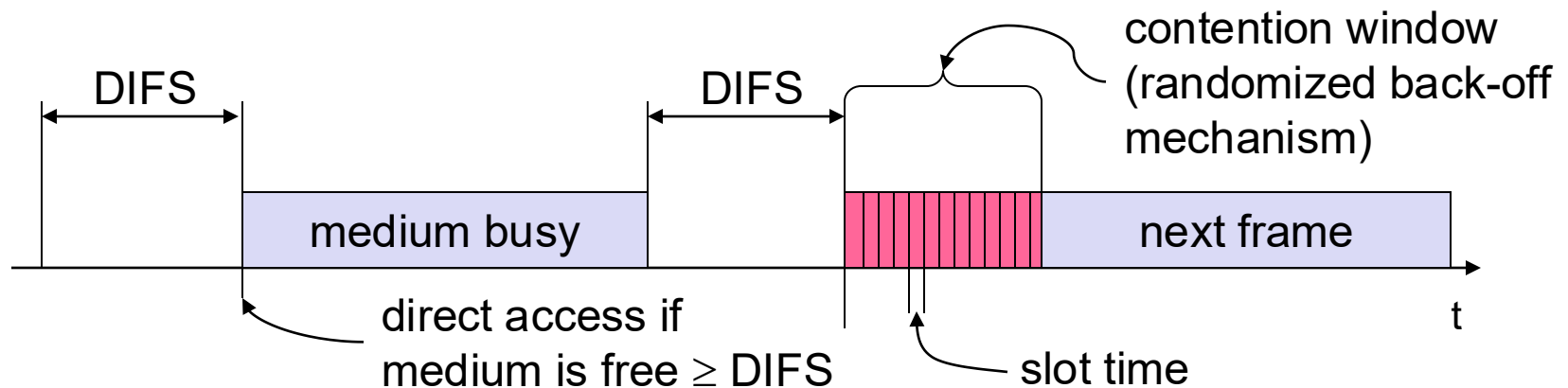
Interframe Spacing

- ▶ **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 ($> \text{SIFS}$)

▶ PIFS

▶ PCF interframe space

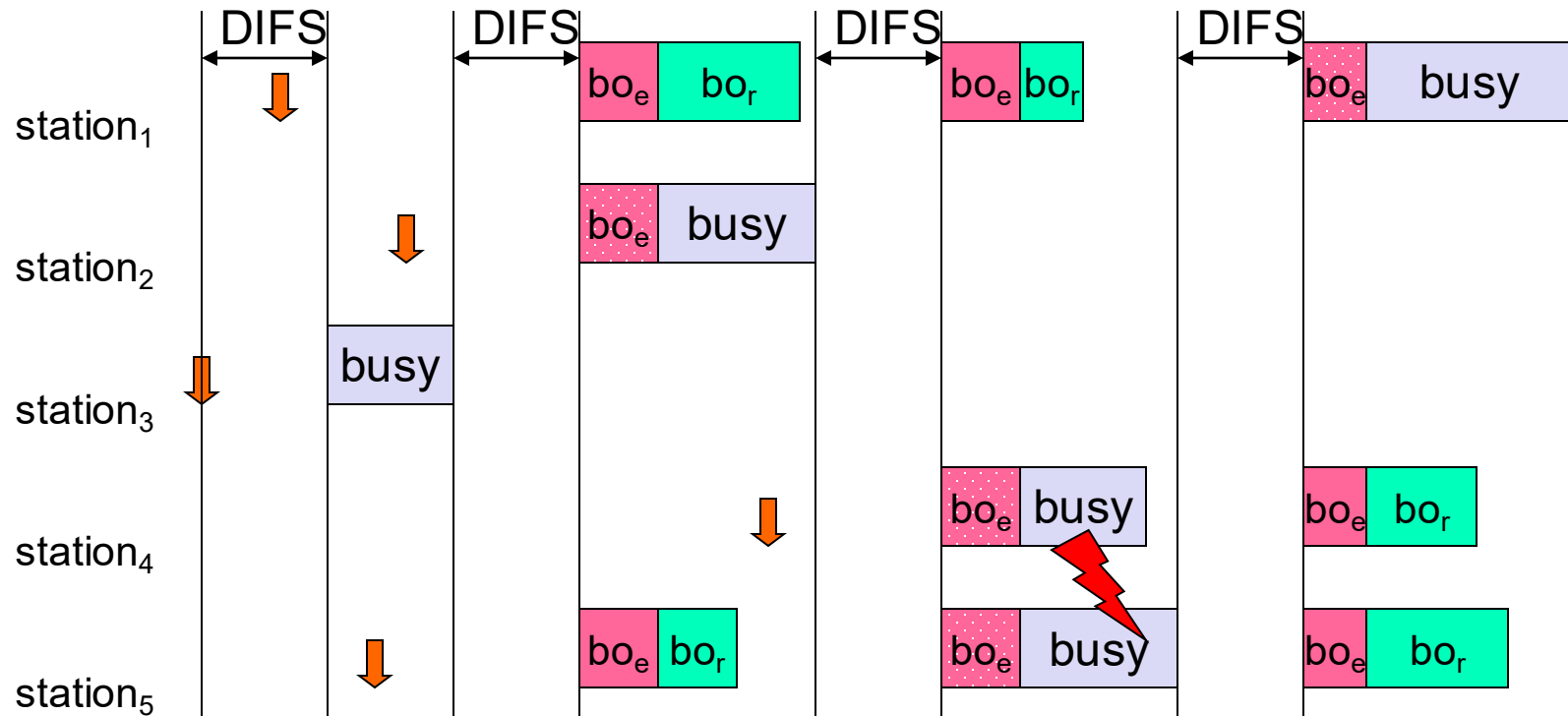
- ▶ Minimum idle time for contention-free service ($> \text{SIFS}$, $< \text{DIFS}$)

▶ EIFS

▶ Extended interframe space

- ▶ Used when there is an error in transmission

IEEE 802.11 - Competing Stations



busy

medium not idle (frame, ack etc.)

bo_e

elapsed backoff time



packet arrival at MAC

bo_r

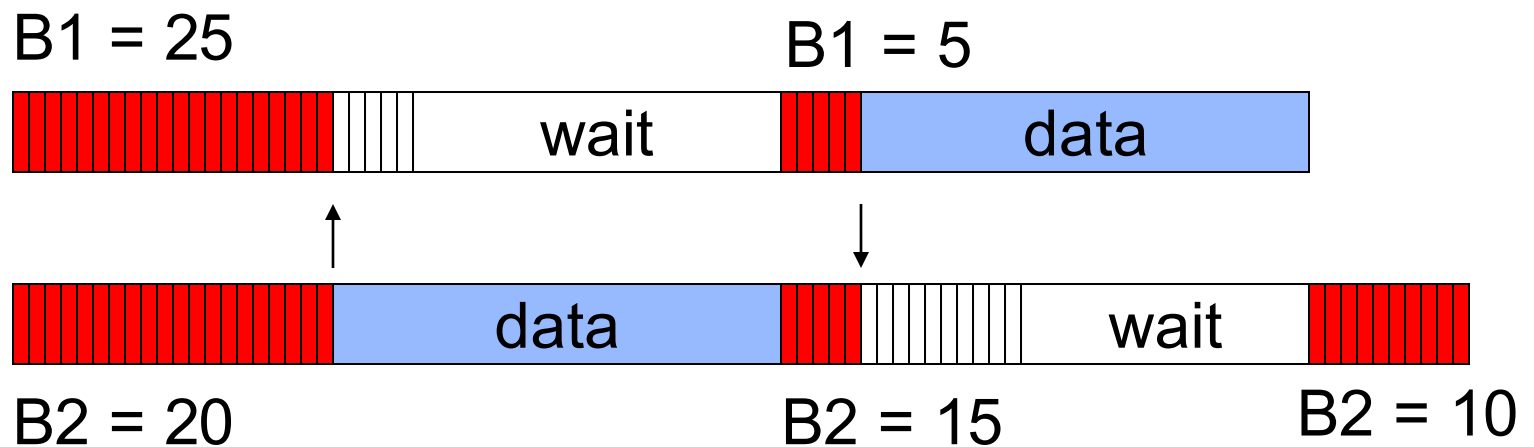
residual backoff time

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

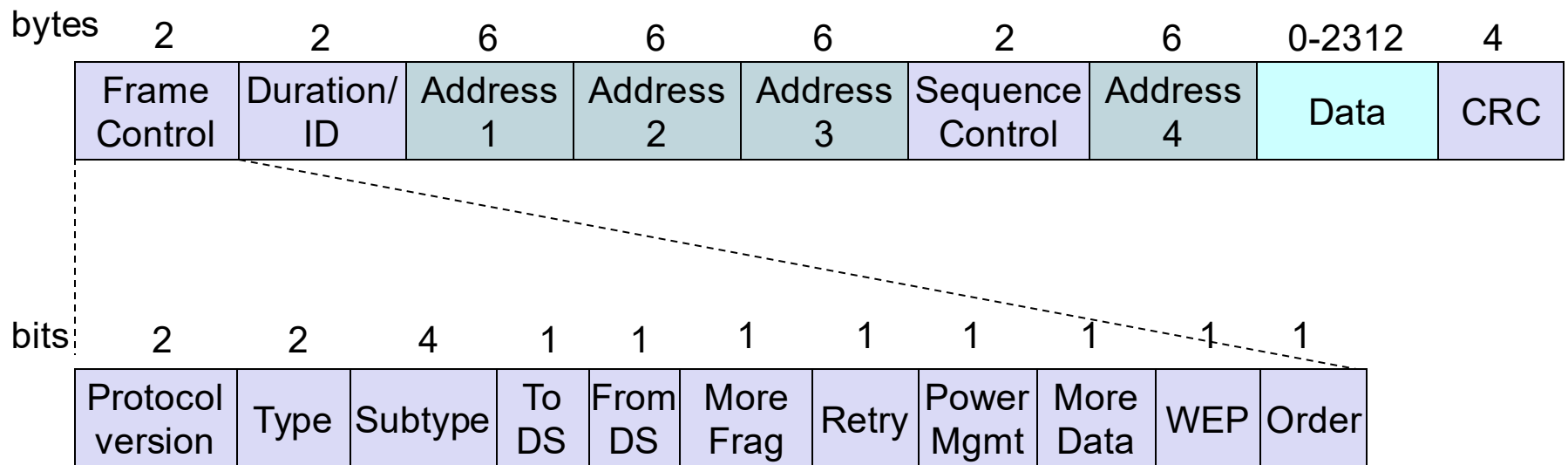
▶ Addresses

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

▶ Miscellaneous

- ▶ sending time, checksum, frame control, data

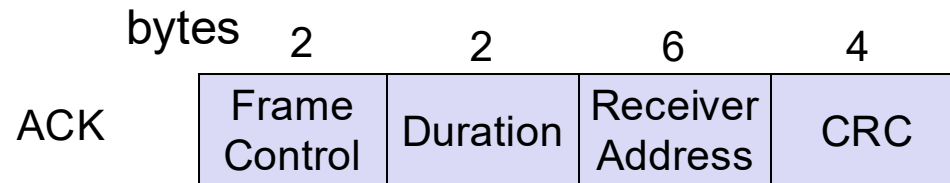
IEEE 802.11 Data Frame Format



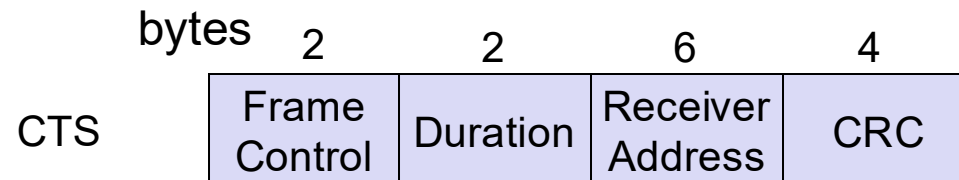
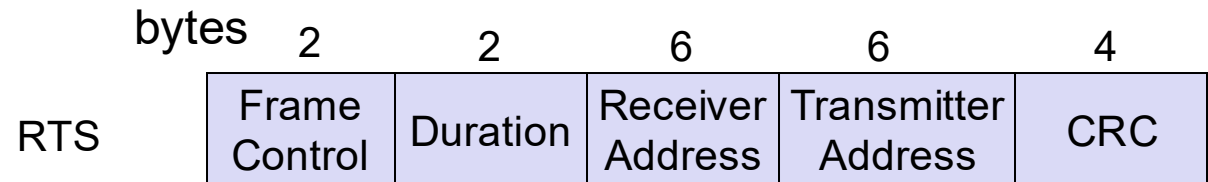
IEEE 802.11 Control Frame Format

- ▶ Acknowledgement

- ▶ Request To Send

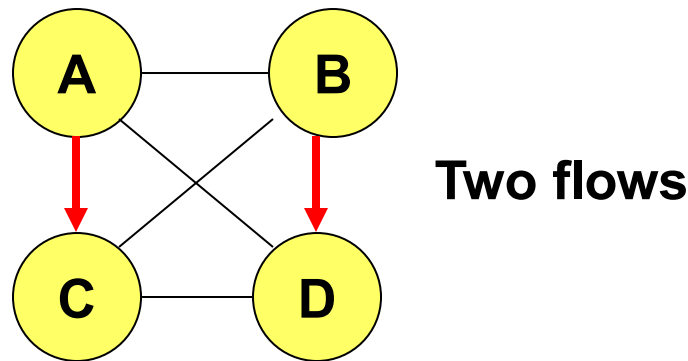


- ▶ Clear To Send



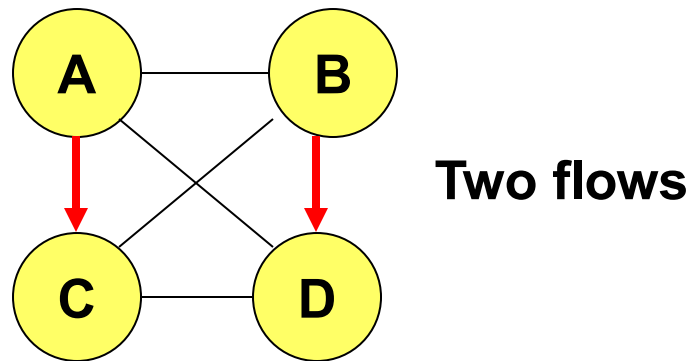
Fairness Issue

- ▶ 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

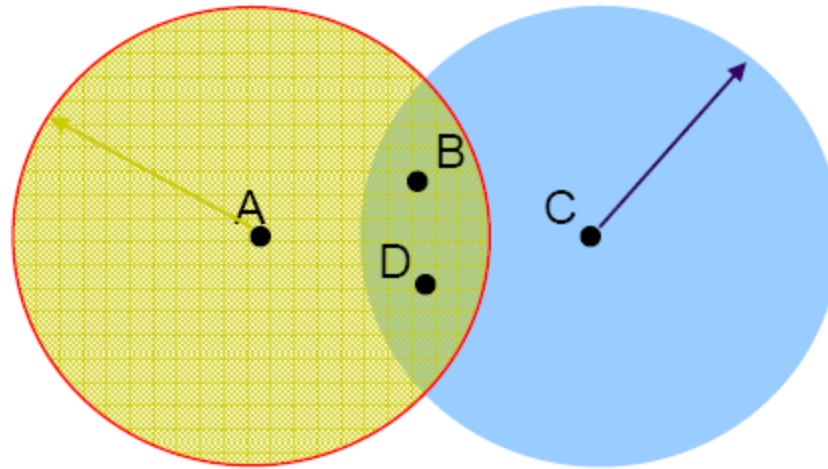
▶ Unfairness

- ▶ 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

Preventing Collisions Altogether



- ▶ **Frequency Spectrum partitioned into several channels**
 - ▶ Nodes within interference range can use separate channels
 - ▶ Now A can send to B while C sends to D without any interference!
 - ▶ Aggregate Network throughput doubles

Using Multiple Channels

- ▶ **802.11:AP's on different channels**
 - ▶ Usually manually configured by administrator
 - ▶ Automatic Configuration may cause problems
- ▶ **Most cards have only 1 transceiver**
 - ▶ Not Full Duplex: Cannot send and receive at the same time
- ▶ **Multichannel MAC Protocols**
 - ▶ Automatically have nodes negotiate channels
 - ▶ Channel coordination amongst nodes is necessary
 - ▶ Introduces negotiation and channel-switching latency that reduce throughput

802.11ax: High Efficiency WLANs

- ▶ **OFDM**

- ▶ Orthogonal Frequency Division Multiplexing

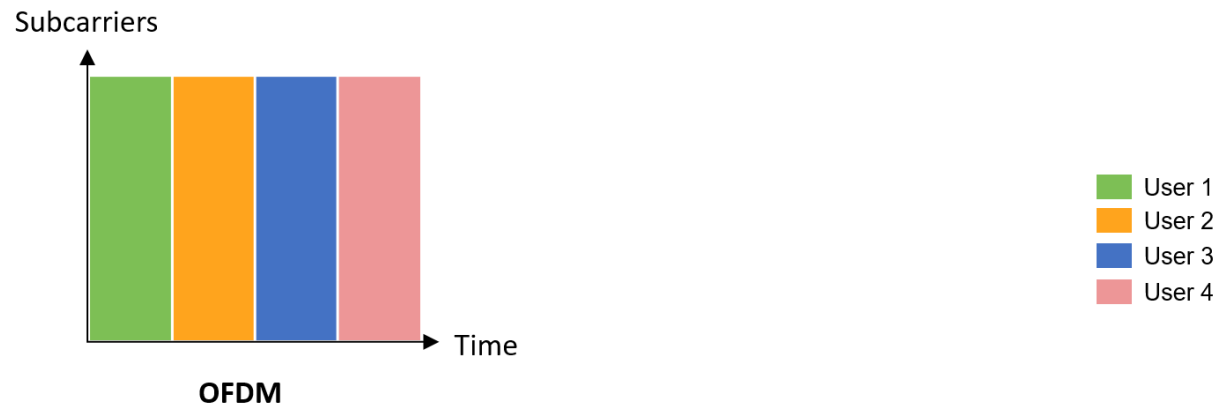
- ▶ **OFDMA**

- ▶ Orthogonal Frequency Division Multiple Access
 - ▶ Divides subcarriers into Resource Units (RUs)
 - ▶ Shares the channel among multiple devices simultaneously

Resource Units

- ▶ RUs are the allocation blocks of bandwidth in OFDMA
- ▶ Vary in size depending on channel width
- ▶ Adjacent subcarriers (tones) are grouped together into a resource unit (RU)

OFDM v OFDMA

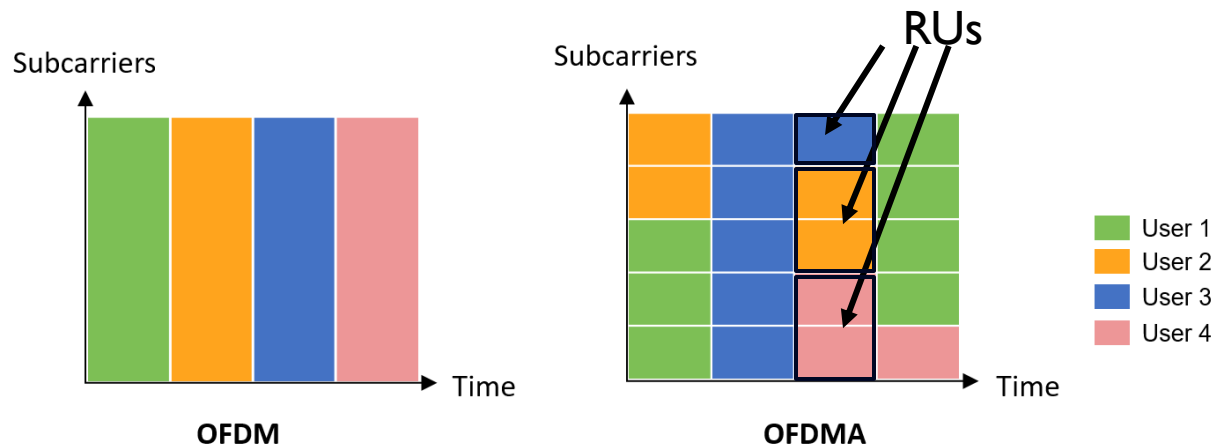


Single user on channel
at a time

<https://oringnet.com/en/knowledge-base/how-does-wifi6-ofdma-work>



OFDM v OFDMA



Single user on channel
at a time

Multiple users,
Various bandwidths

<https://oringnet.com/en/knowledge-base/how-does-wifi6-ofdma-work>

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)



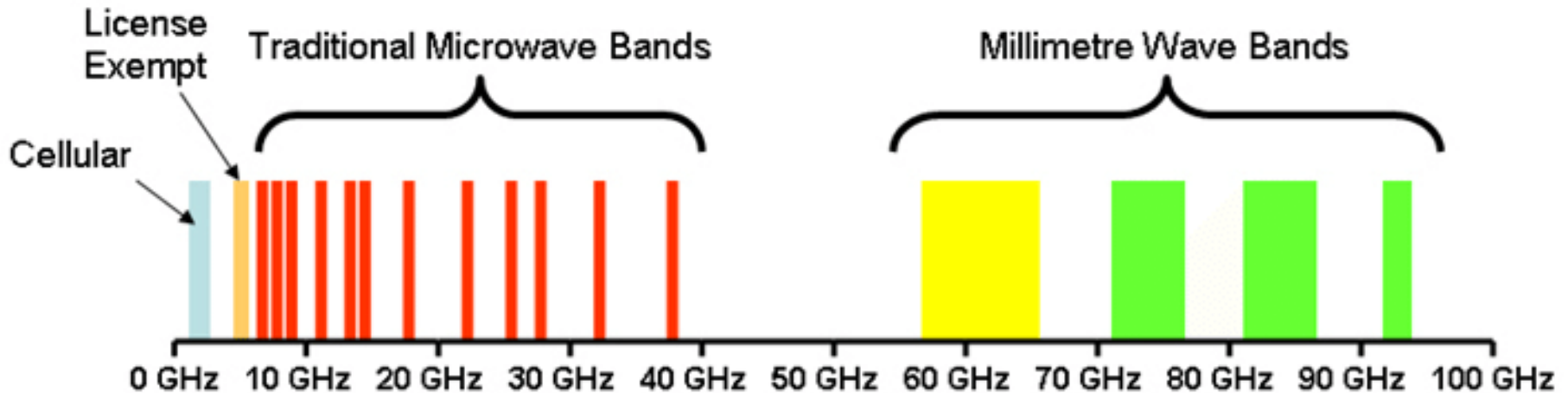
IEEE 802.11 Amendments

- ▶ 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



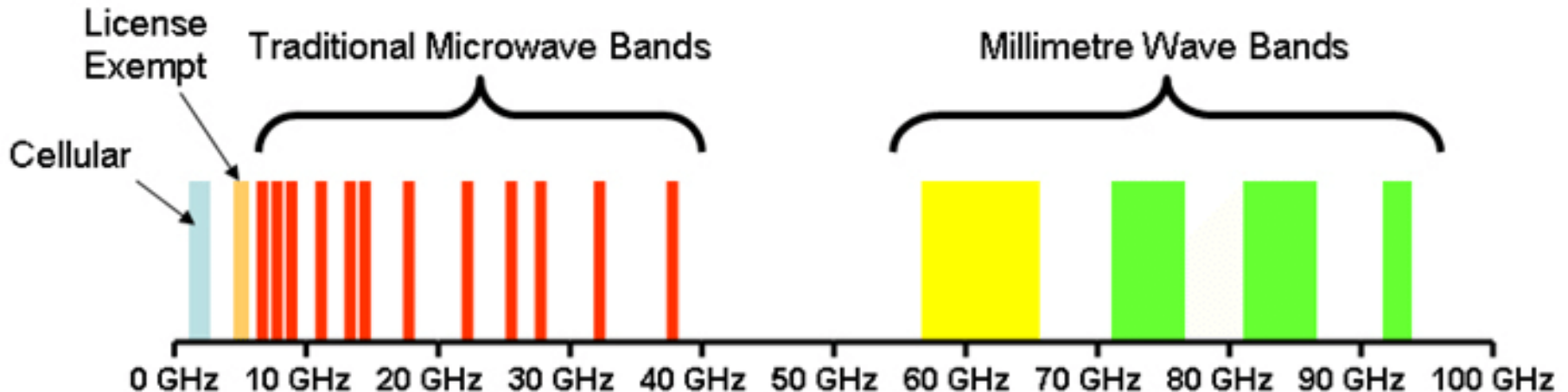
802.11ad/ay: WiGig (60 GHz)

- ▶ Uses the unlicensed 60 GHz band
 - ▶ Huge bandwidth available (multiple GHz-wide channels)
 - ▶ Very high throughput, but very short range
 - ▶ Signal barely penetrates walls; often line-of-sight only



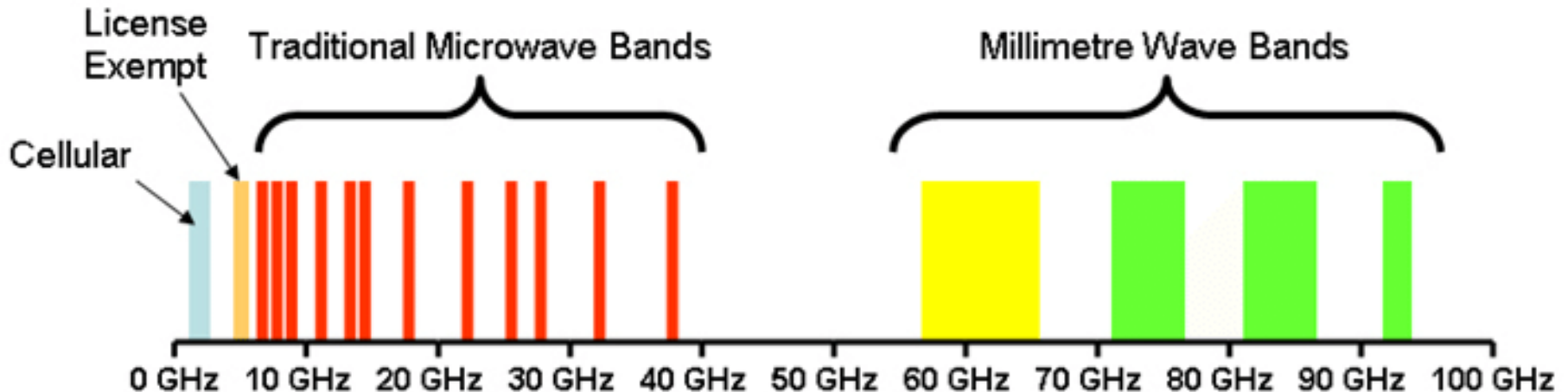
802.11ad/ay: WiGig (60 GHz)

- ▶ 802.11ad (2012)
 - ▶ Defines the original 60 GHz PHY/MAC
 - ▶ Up to ~7 Gbps
- ▶ 802.11ay (2021)
 - ▶ Extends 802.11ad — channel bonding, MIMO
 - ▶ Up to ~40+ Gbps



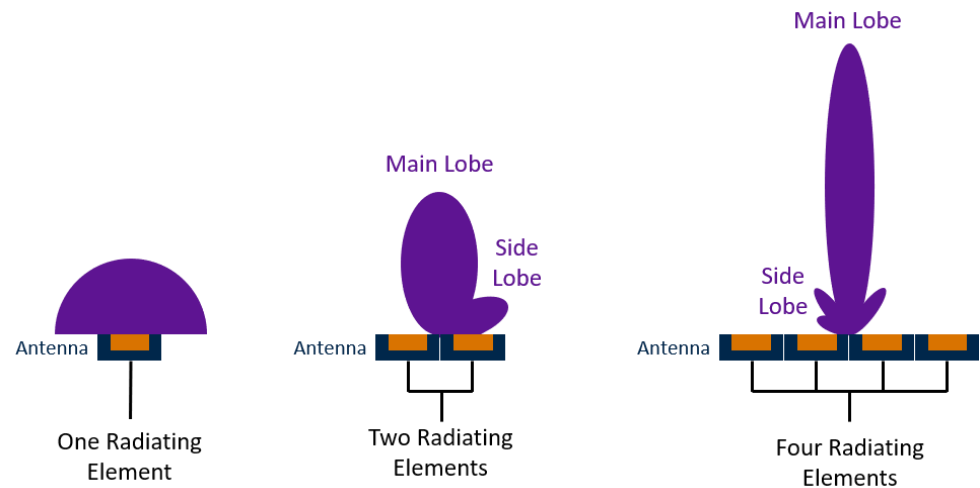
802.11ad/ay: WiGig (60 GHz)

- ▶ Typical uses
 - ▶ Wireless docking stations
 - ▶ VR headset streaming
 - ▶ Short-range "wireless HDMI" replacement
 - ▶ Some 5G backhaul links
- ▶ Not a replacement for 2.4/5/6 GHz Wi-Fi
 - ▶ Used alongside it for specific short-range, high-bandwidth links



802.11ad/ay: WiGig MAC

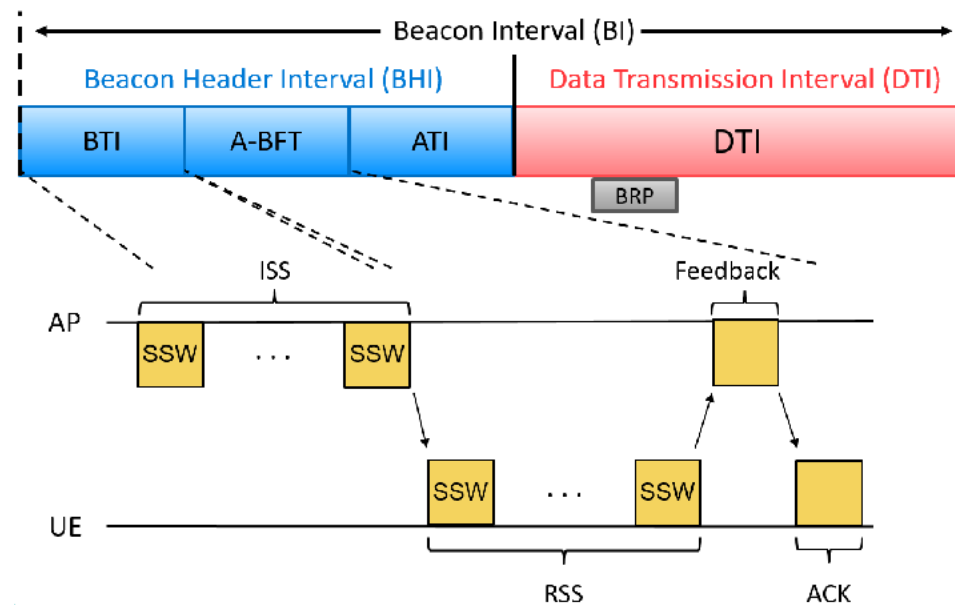
- ▶ 60 GHz uses directional beamforming
 - ▶ Contention-based access alone doesn't work
 - ▶ Path loss is severe
 - ▶ Nodes must physically steer narrow antenna beams
 - ▶ Communication requires precise alignment



802.11ad/ay: WiGig MAC

▶ Beacon Header Interval

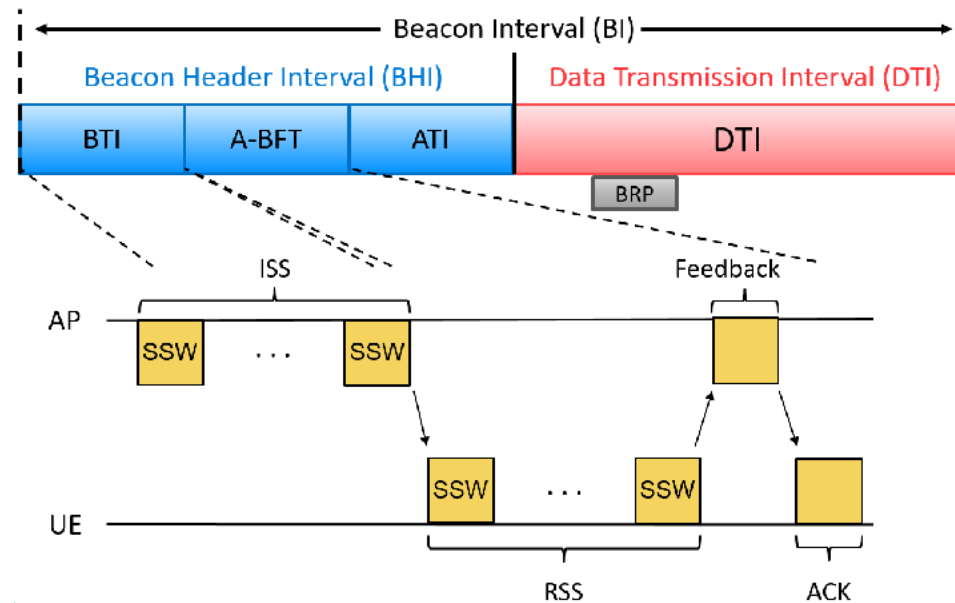
- ▶ Management and control window at the beginning of each cycle
- ▶ Dedicated to antenna alignment and scheduling
- ▶ SSW (Sector Sweep)
 - ▶ Directional wireless beams are swept sequentially to determine the optimal link path
- ▶ A-BFT (Association Beamforming Training)
 - ▶ User equipment responds to the AP's sweeps
 - ▶ Both sides lock in mmwave antenna steering directions



802.11ad/ay: WiGig MAC

► Access mode

- Contention-Based Access Periods (CBAP)
 - standard CSMA-CA
- Service Periods (SP)
 - TDMA-like, contention-free slots
 - scheduled by a central coordinator



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)

