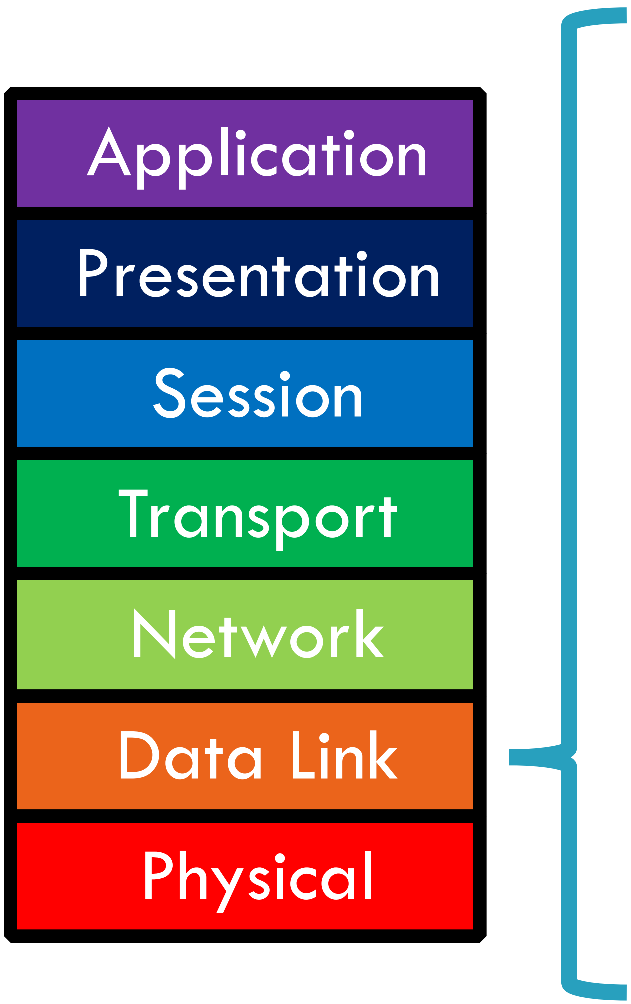
# CS 3700
## Networks and Distributed Systems

**Lecture 4: Ethernet/Media Access**

Revised 1/14/13

# Data Link Layer






- Function:
  - Send blocks of data (frames) between physical devices
  - Regulate access to the physical media
- Key challenge:
  - How to delineate frames?
  - How to detect errors?
  - How to perform media access control (MAC)?
  - How to recover from and avoid collisions?

# Outline

- Media Access Control
  - 802.3 Ethernet
  - 802.11 Wifi

# What is Media Access?



- Ethernet and Wifi are both multi-access technologies
  - Broadcast medium, shared by many hosts
  - Simultaneous transmissions cause collisions
    - This destroys the data

# What is Media Access?



- Ethernet and Wifi are both multi-access technologies
  - Broadcast medium, shared by many hosts
  - Simultaneous transmissions cause collisions
    - This destroys the data
- Media Access Control (MAC) protocols are required
  - Rules on how to share the medium
  - Strategies for detecting, avoiding, and recovering from collisions

# Strategies for Media Access



- Channel partitioning
  - Divide the resource into small pieces
  - Allocate each piece to one host
  - Example: Time Division Multi-Access (TDMA) cellular
  - Example: Frequency Division Multi-Access (FDMA) cellular

# Strategies for Media Access



- Channel partitioning
  - Divide the resource into small pieces
  - Allocate each piece to one host
  - Example: Time Division Multi-Access (TDMA) cellular
  - Example: Frequency Division Multi-Access (FDMA) cellular
- Taking turns
  - Tightly coordinate shared access to avoid collisions
  - Example: Token ring networks

# Strategies for Media Access



- Channel partitioning
  - Divide the resource into small pieces
  - Allocate each piece to one host
  - Example: Time Division Multi-Access (TDMA) cellular
  - Example: Frequency Division Multi-Access (FDMA) cellular
- Taking turns
  - Tightly coordinate shared access to avoid collisions
  - Example: Token ring networks
- Contention
  - Allow collisions, but use strategies to recover
  - Examples: Ethernet, Wifi

# Strategies for Media Access



- Channel partitioning
  - Divide the resource into small pieces
  - Allocate each piece to one host
  - Example: Time Division Multi-Access (TDMA) cellular
  - Example: Frequency Division Multi-Access (FDMA) cellular
- Taking turns
  - Tightly coordinate shared access to avoid collisions
  - Example: Token ring networks
- Contention
  - Allow collisions, but use strategies to recover
  - Examples: Ethernet, Wifi

# Contention MAC Goals



- Share the medium
  - Two hosts sending at the same time collide, thus causing interference
  - If no host sends, channel is idle
  - Thus, want one user sending at any given time

# Contention MAC Goals



- Share the medium
  - Two hosts sending at the same time collide, thus causing interference
  - If no host sends, channel is idle
  - Thus, want one user sending at any given time
- High utilization
  - TDMA is low utilization
  - Just like a circuit switched network

# Contention MAC Goals



- Share the medium
  - Two hosts sending at the same time collide, thus causing interference
  - If no host sends, channel is idle
  - Thus, want one user sending at any given time
- High utilization
  - TDMA is low utilization
  - Just like a circuit switched network
- Simple, distributed algorithm
  - Multiple hosts that cannot directly coordinate
  - No fancy (complicated) token-passing schemes

# Contention Protocol Evolution



- ALOHA
  - Developed in the 70's for packet radio networks

# Contention Protocol Evolution



- ALOHA
  - Developed in the 70's for packet radio networks
- Slotted ALOHA
  - Start transmissions only at fixed time slots
  - Significantly fewer collisions than ALOHA

# Contention Protocol Evolution



- ALOHA
  - Developed in the 70's for packet radio networks
- Slotted ALOHA
  - Start transmissions only at fixed time slots
  - Significantly fewer collisions than ALOHA
- Carrier Sense Multiple Access (CSMA)
  - Start transmission only if the channel is idle

# Contention Protocol Evolution

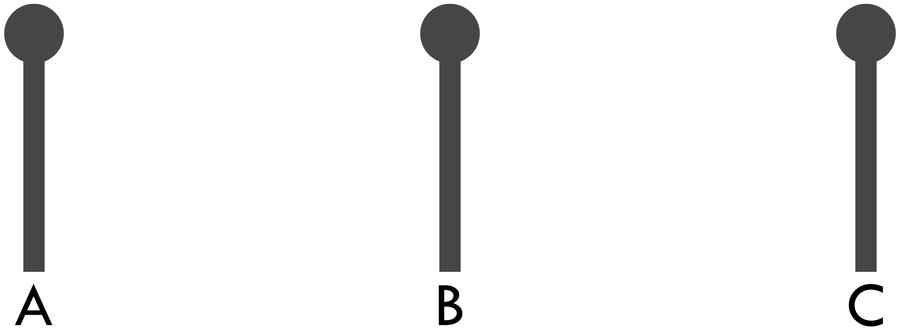


- ALOHA
  - Developed in the 70's for packet radio networks
- Slotted ALOHA
  - Start transmissions only at fixed time slots
  - Significantly fewer collisions than ALOHA
- Carrier Sense Multiple Access (CSMA)
  - Start transmission only if the channel is idle
- CSMA / Collision Detection (CSMA/CD)
  - Stop ongoing transmission if collision is detected

# ALOHA



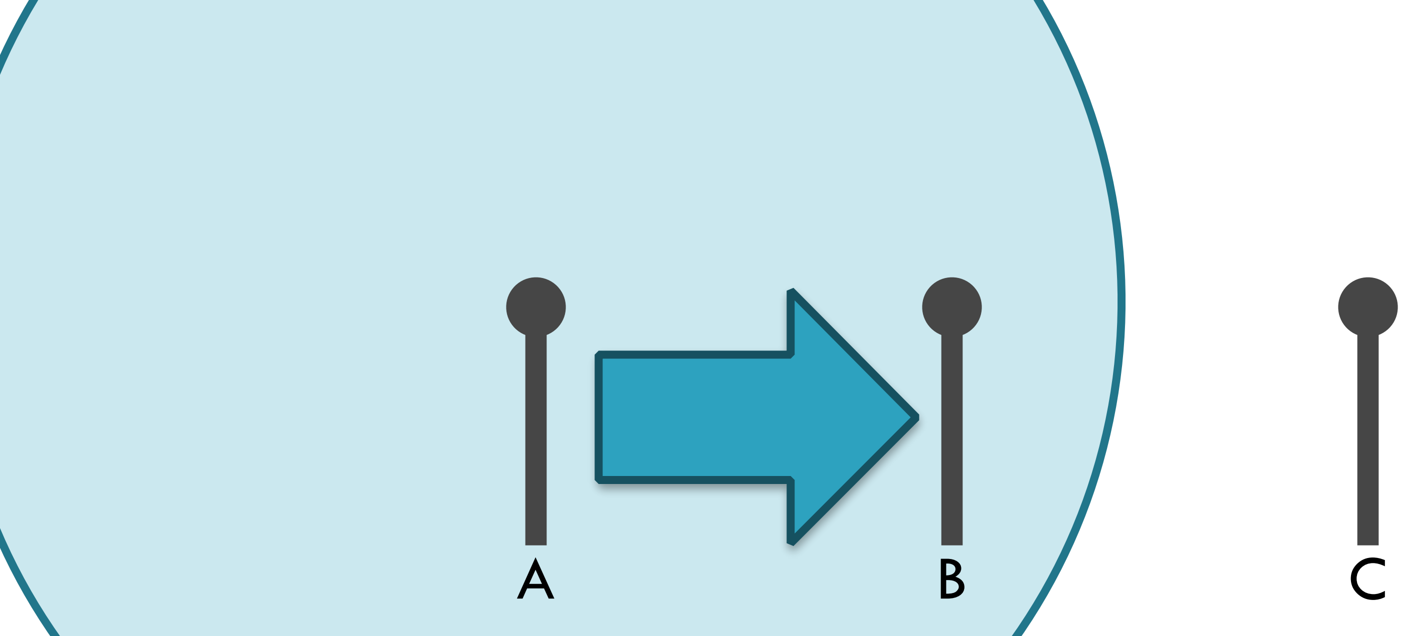
- Topology: radio broadcast with multiple stations
- Protocol:
  - Stations transmit data immediately
  - Receivers ACK all packets
  - No ACK = collision, wait a random time then retransmit



# ALOHA



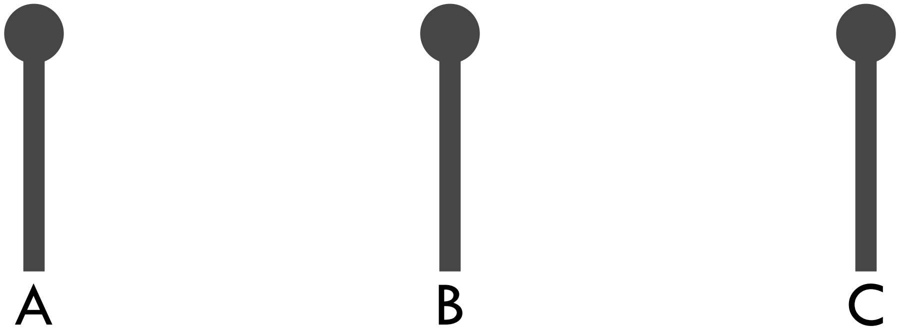
- Topology: radio broadcast with multiple stations
- Protocol:
  - Stations transmit data immediately
  - Receivers ACK all packets
  - No ACK = collision, wait a random time then retransmit



# ALOHA



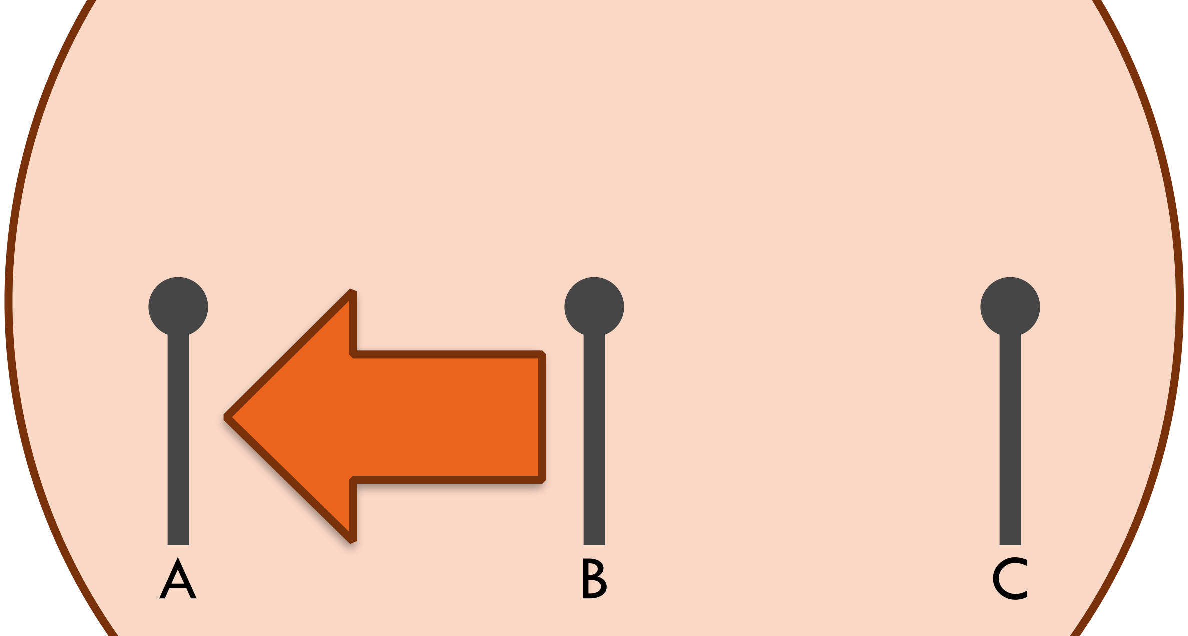
- Topology: radio broadcast with multiple stations
- Protocol:
  - Stations transmit data immediately
  - Receivers ACK all packets
  - No ACK = collision, wait a random time then retransmit



# ALOHA



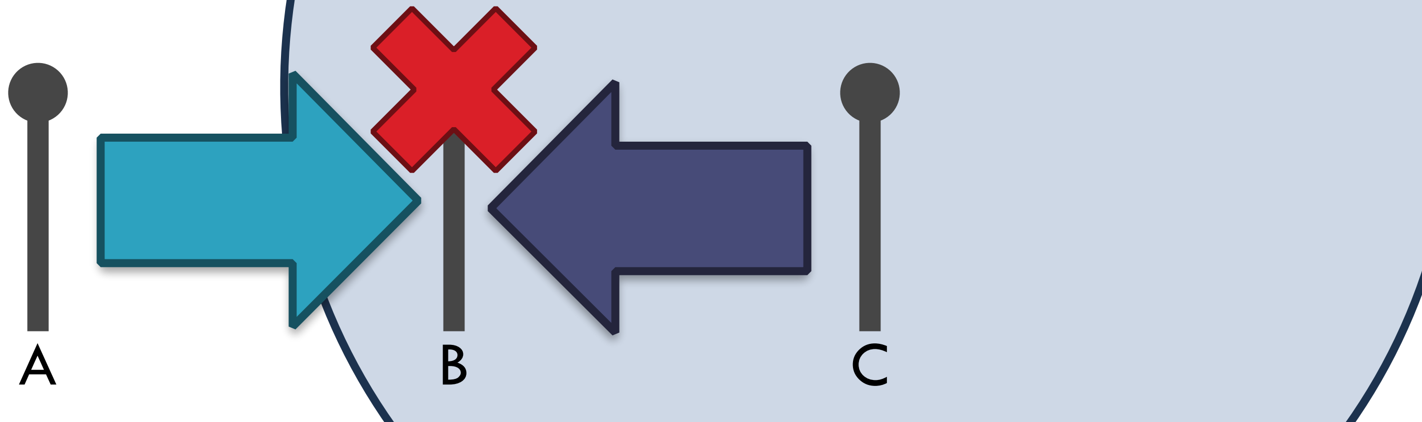
- Topology: radio broadcast with multiple stations
- Protocol:
  - Stations transmit data immediately
  - Receivers ACK all packets
  - No ACK = collision, wait a random time then retransmit



# ALOHA



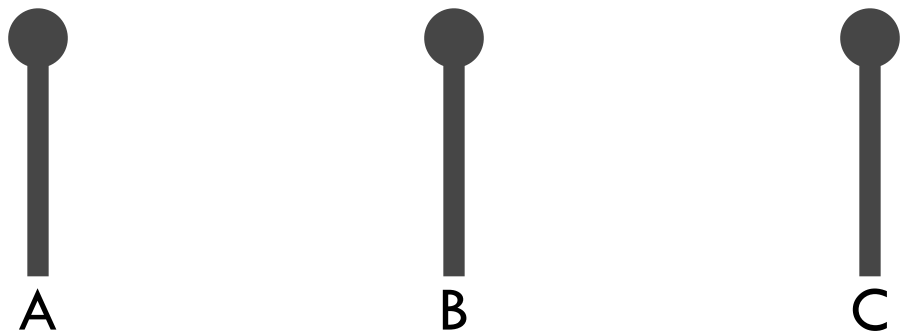
- Topology: radio broadcast with multiple stations
- Protocol:
  - Stations transmit data immediately
  - Receivers ACK all packets
  - No ACK = collision, wait a random time then retransmit



# ALOHA



- Topology: radio broadcast with multiple stations
- Protocol:
  - Stations transmit data immediately
  - Receivers ACK all packets
  - No ACK = collision, wait a random time then retransmit



# ALOHA



- Topology: radio broadcast with multiple stations
- Protocol:
  - Stations transmit data immediately
  - Receivers ACK all packets
  - No ACK = collision, wait a random time then retransmit

- Simple, but radical concept
- Previous attempts all divided the channel
  - TDMA, FDMA, etc.
- Optimized for the common case: few senders

# Tradeoffs vs. TDMA

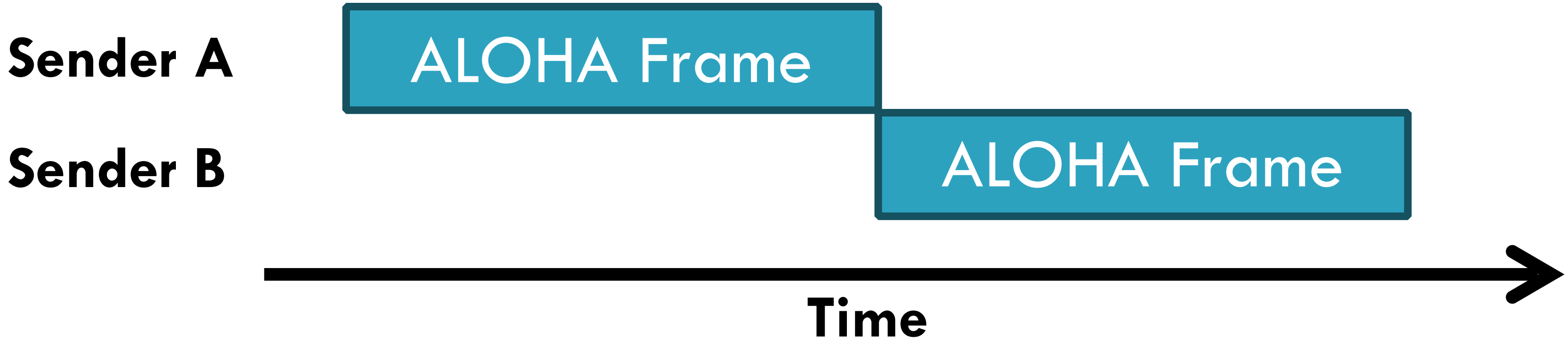


- In TDMA, each host must wait for its turn
  - Delay is proportional to number of hosts
- In Aloha, each host sends immediately
  - Much lower delay
  - But, much lower utilization

# Tradeoffs vs. TDMA



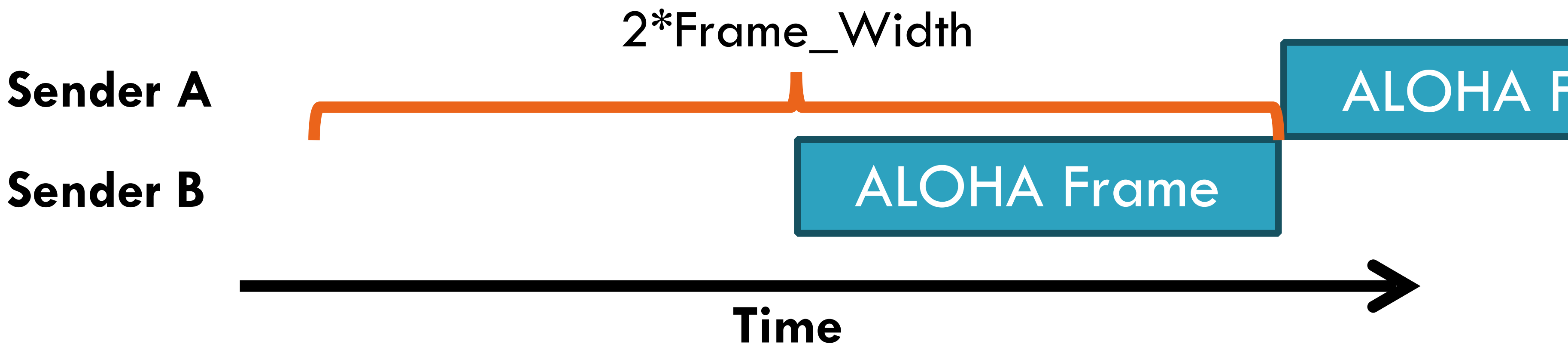
- In TDMA, each host must wait for its turn
  - Delay is proportional to number of hosts
- In Aloha, each host sends immediately
  - Much lower delay
  - But, much lower utilization



# Tradeoffs vs. TDMA



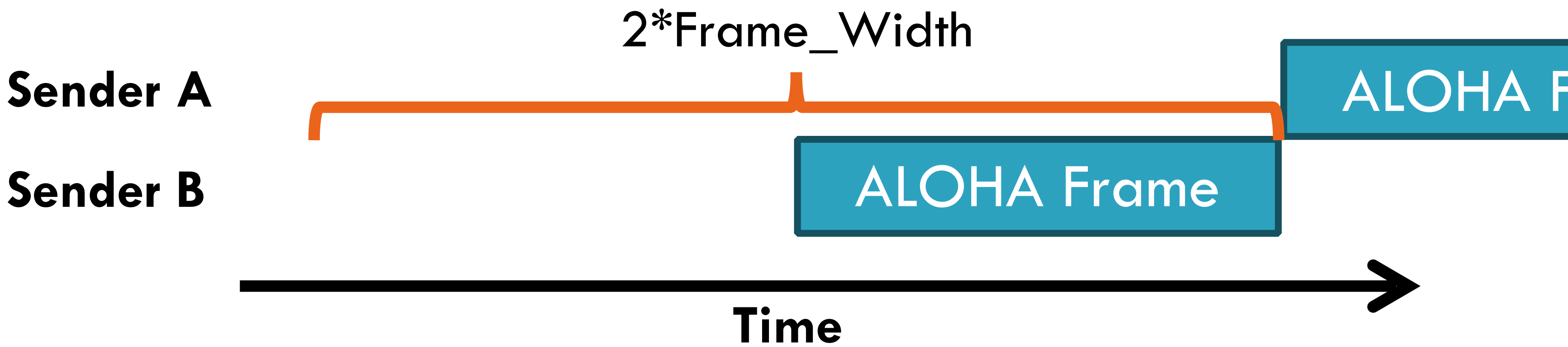
- In TDMA, each host must wait for its turn
  - Delay is proportional to number of hosts
- In Aloha, each host sends immediately
  - Much lower delay
  - But, much lower utilization



# Tradeoffs vs. TDMA



- In TDMA, each host must wait for its turn
  - Delay is proportional to number of hosts
- In Aloha, each host sends immediately
  - Much lower delay
  - But, much lower utilization




  - Maximum throughput is ~18% of channel capacity

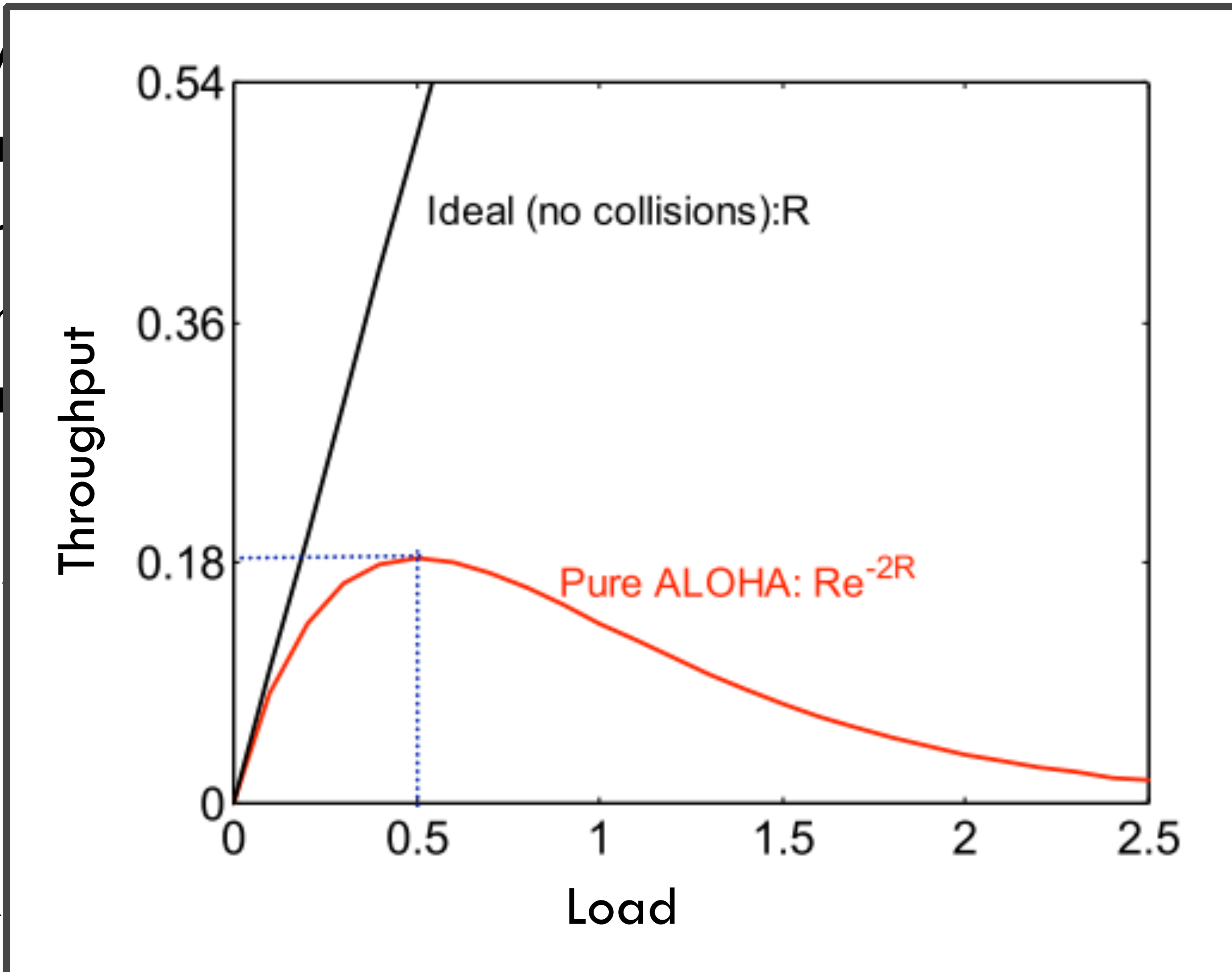
# Tradeoffs vs. TDMA



- In TDM
  - Dela
- In Aloh
  - Much
  - But,

**Sender**

**Sender**




LOHA F

- Max

# Slotted ALOHA

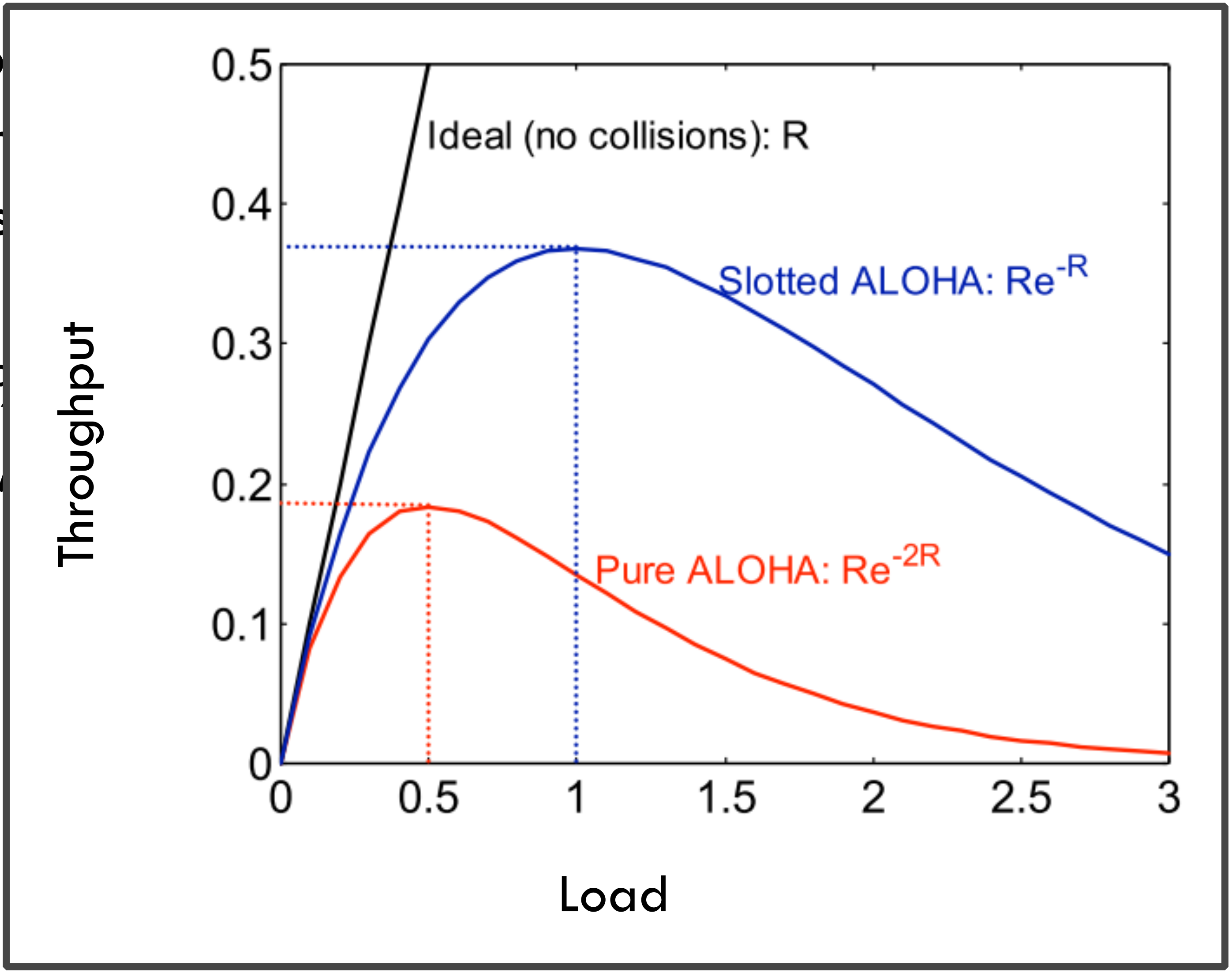


- Protocol
  - Same as ALOHA, except time is divided into slots
  - Hosts may only transmit at the beginning of a slot
- Thus, frames either collide completely, or not at all
  - 37% throughput vs. 18% for ALOHA
  - But, hosts must have synchronized clocks

# Slotted ALOHA



- Proto
  - Sar
  - Hos
- Thus,
  - 37
  - But



# 802.3 Ethernet



| Bytes | 7 | 1 | 6 | 6 | 2 | 0-1500 | 0-46 | 4 |
|---|---|---|---|---|---|---|---|---|
| | Preamble | SF | Source | Dest. | Length | Data | Pad | Checksum |

# 802.3 Ethernet



| Bytes | 7 | 1 | 6 | 6 | 2 | 0-1500 | 0-46 | 4 |
|---|---|---|---|---|---|---|---|---|
| | Preamble | SF | Source | Dest. | Length | Data | Pad | Checksum |

- Preamble is 7 bytes of 10101010. Why?

# 802.3 Ethernet

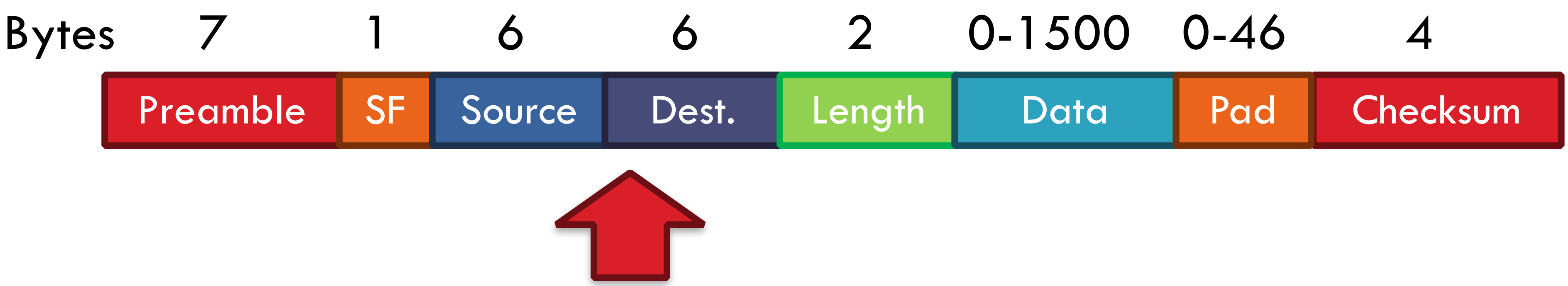


| Bytes | 7 | 1 | 6 | 6 | 2 | 0-1500 | 0-46 | 4 |
|---|---|---|---|---|---|---|---|---|
| | Preamble | SF | Source | Dest. | Length | Data | Pad | Checksum |

- Preamble is 7 bytes of 10101010. Why?
- Start Frame (SF) is 10101011

# 802.3 Ethernet






- Preamble is 7 bytes of 10101010. Why?
- Start Frame (SF) is 10101011
- Source and destination are MAC addresses
  - E.g. 00:45:A5:F3:25:0C
  - Broadcast: FF:FF:FF:FF:FF:FF

# 802.3 Ethernet

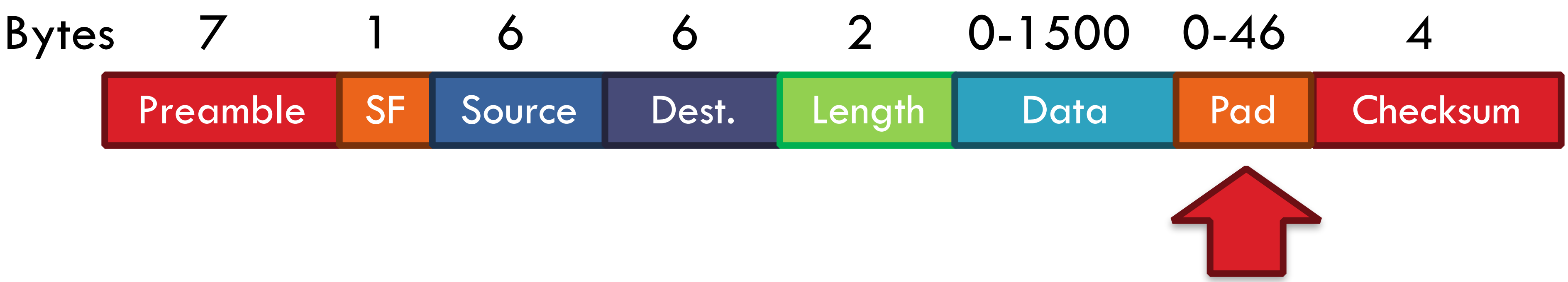


| Bytes | 7 | 1 | 6 | 6 | 2 | 0-1500 | 0-46 | 4 |
|---|---|---|---|---|---|---|---|---|
| | Preamble | SF | Source | Dest. | Length | Data | Pad | Checksum |

- Preamble is 7 bytes of 10101010. Why?
- Start Frame (SF) is 10101011
- Source and destination are MAC addresses
  - E.g. 00:45:A5:F3:25:0C
  - Broadcast: FF:FF:FF:FF:FF:FF

# 802.3 Ethernet






- Preamble is 7 bytes of 10101010. Why?
- Start Frame (SF) is 10101011
- Source and destination are MAC addresses
  - E.g. 00:45:A5:F3:25:0C
  - Broadcast: FF:FF:FF:FF:FF:FF
- Minimum packet length of 64 bytes, hence the pad

# 802.3 Ethernet



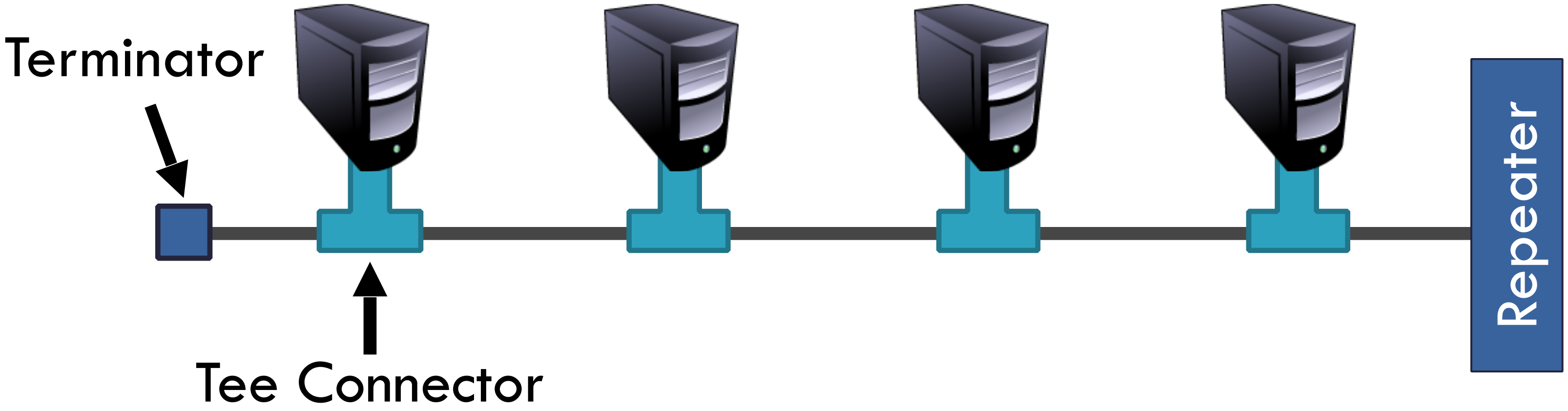
| Bytes | 7 | 1 | 6 | 6 | 2 | 0-1500 | 0-46 | 4 |
|---|---|---|---|---|---|---|---|---|
| | Preamble | SF | Source | Dest. | Length | Data | Pad | Checksum |

- Preamble is 7 bytes of 10101010. Why?
- Start Frame (SF) is 10101011
- Source and destination are MAC addresses
  - E.g. 00:45:A5:F3:25:0C
  - Broadcast: FF:FF:FF:FF:FF:FF
- Minimum packet length of 64 bytes, hence the pad

# Broadcast Ethernet



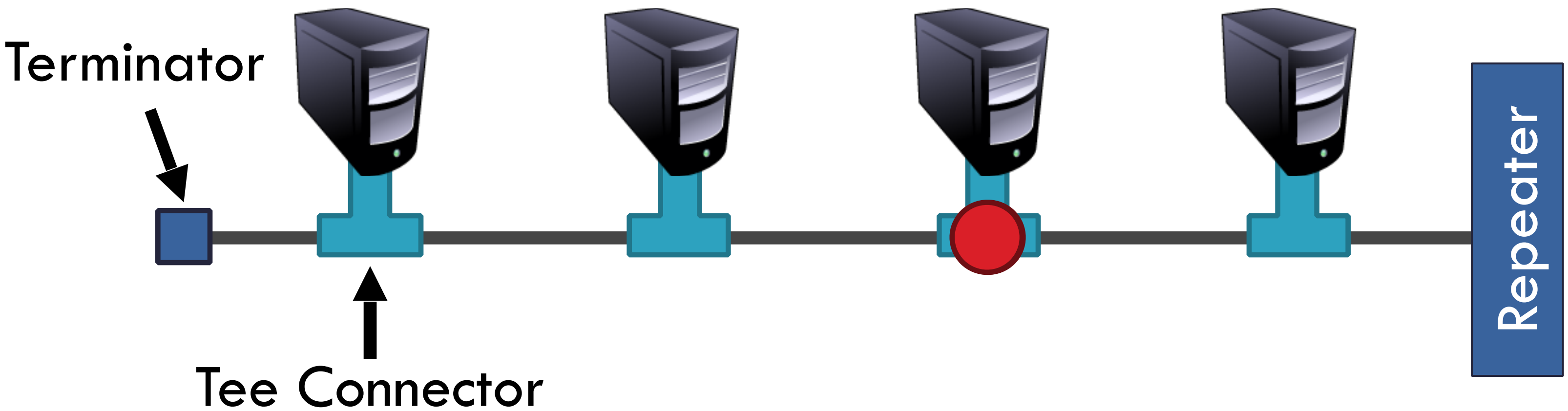
- Originally, Ethernet was a broadcast technology




# Broadcast Ethernet



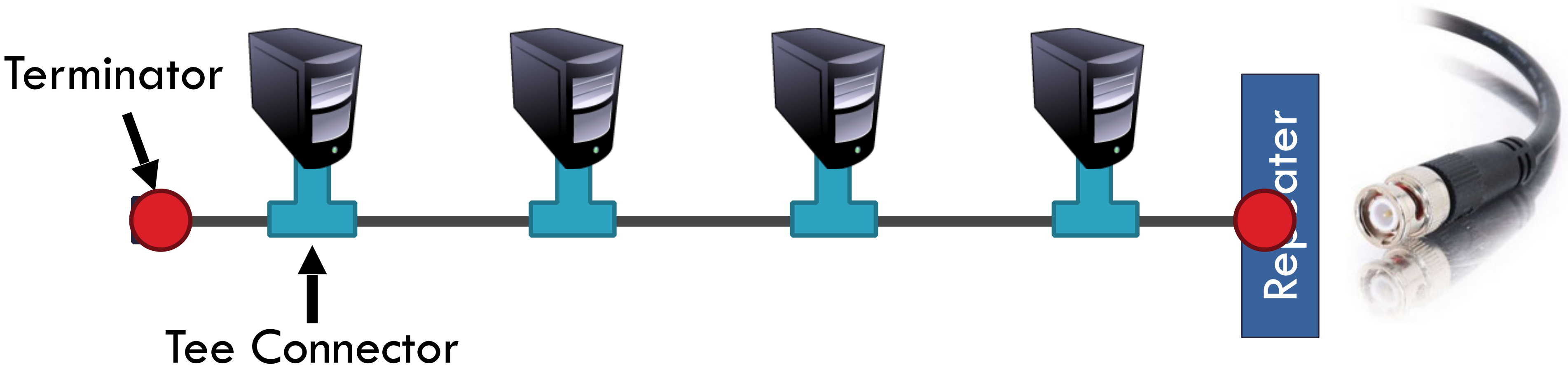
- Originally, Ethernet was a broadcast technology




# Broadcast Ethernet



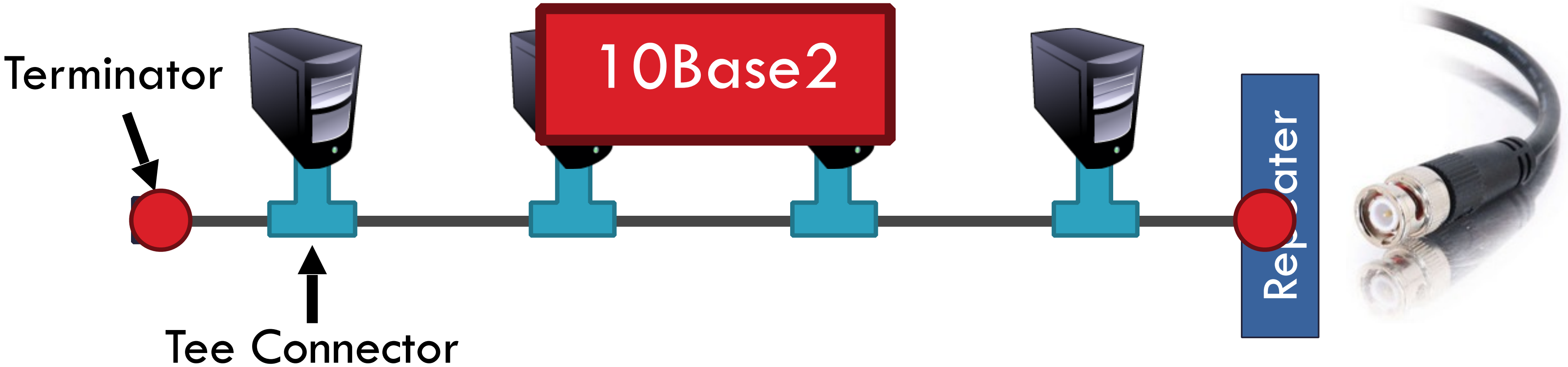
- Originally, Ethernet was a broadcast technology



# Broadcast Ethernet



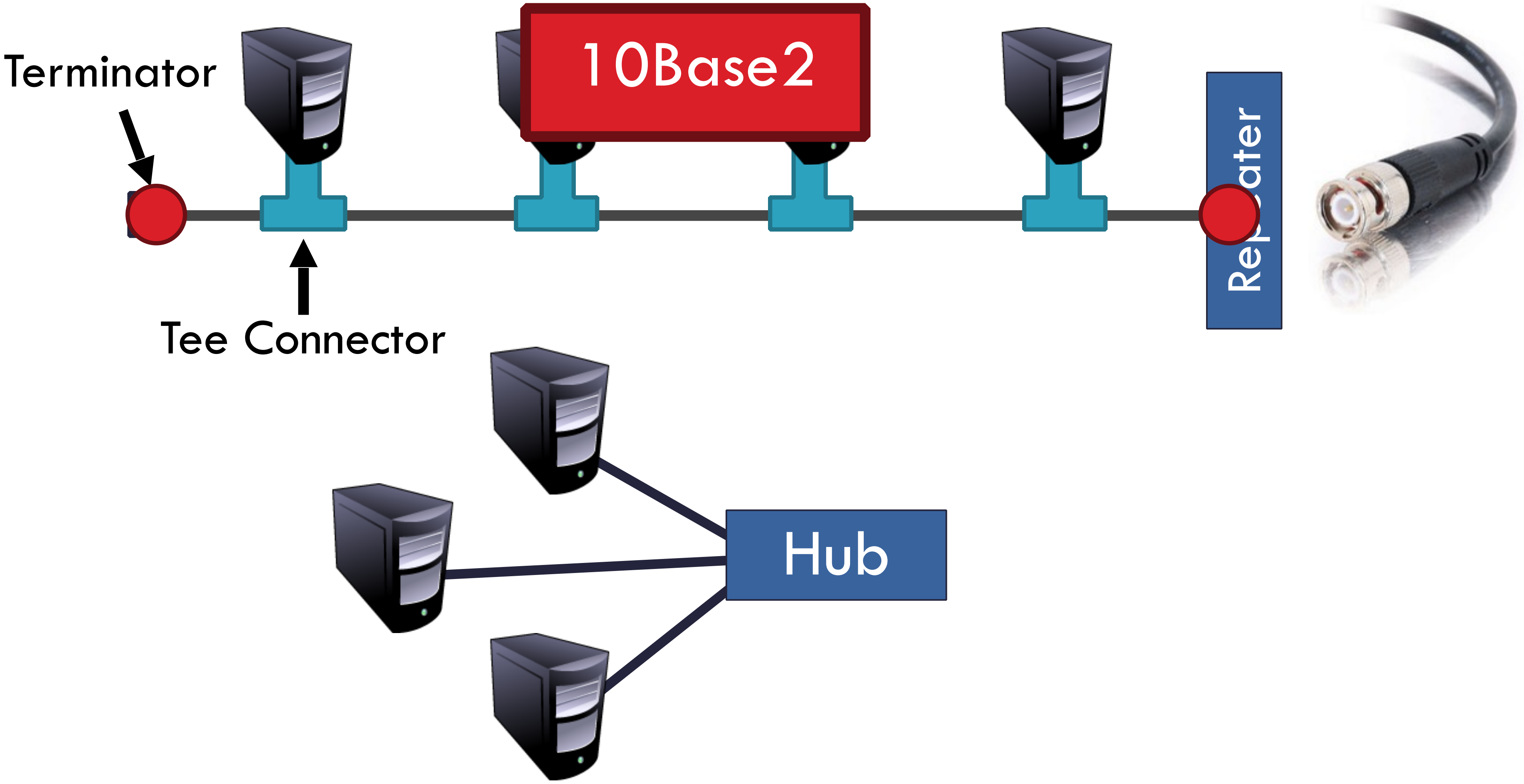
- Originally, Ethernet was a broadcast technology



# Broadcast Ethernet



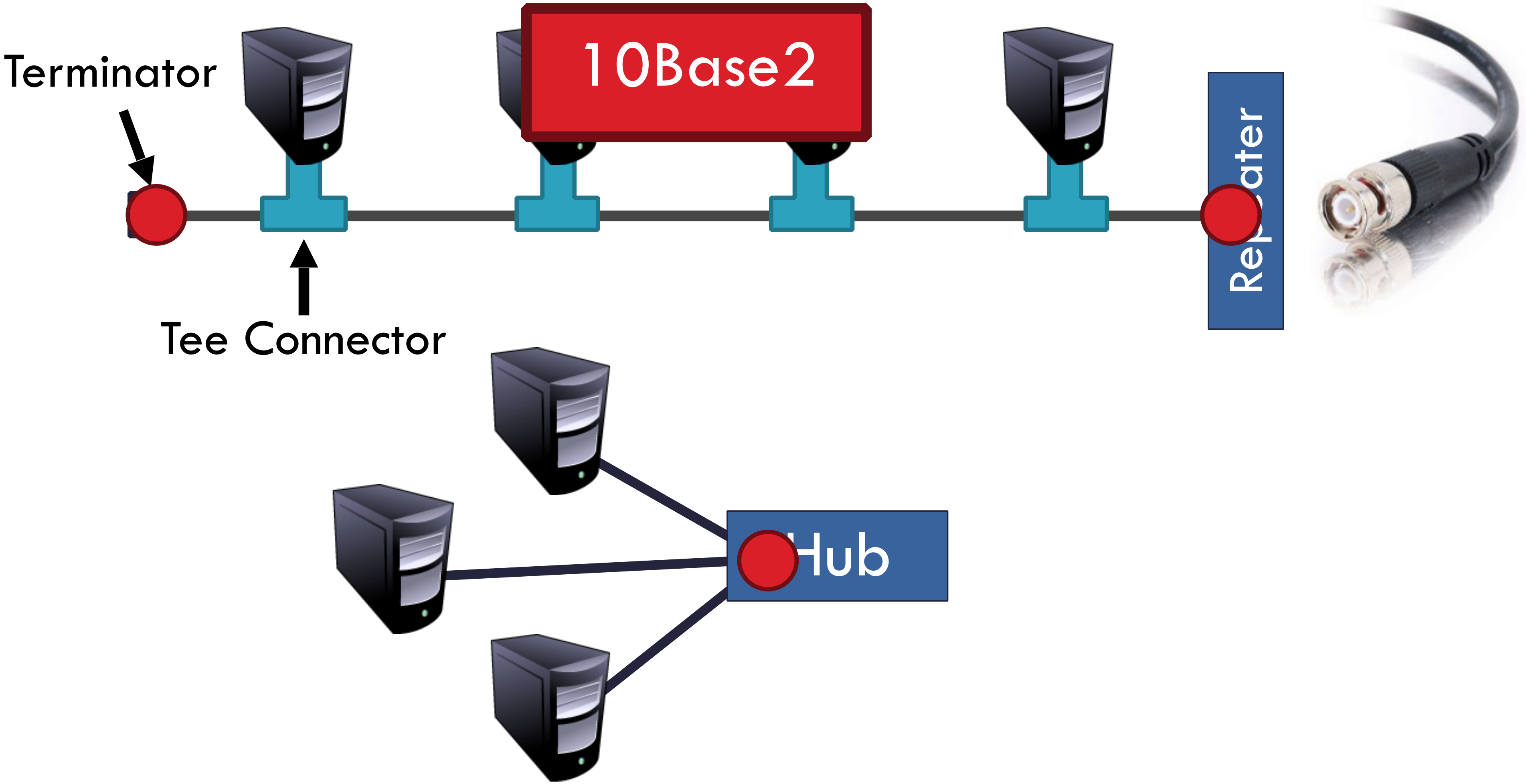
- Originally, Ethernet was a broadcast technology



# Broadcast Ethernet



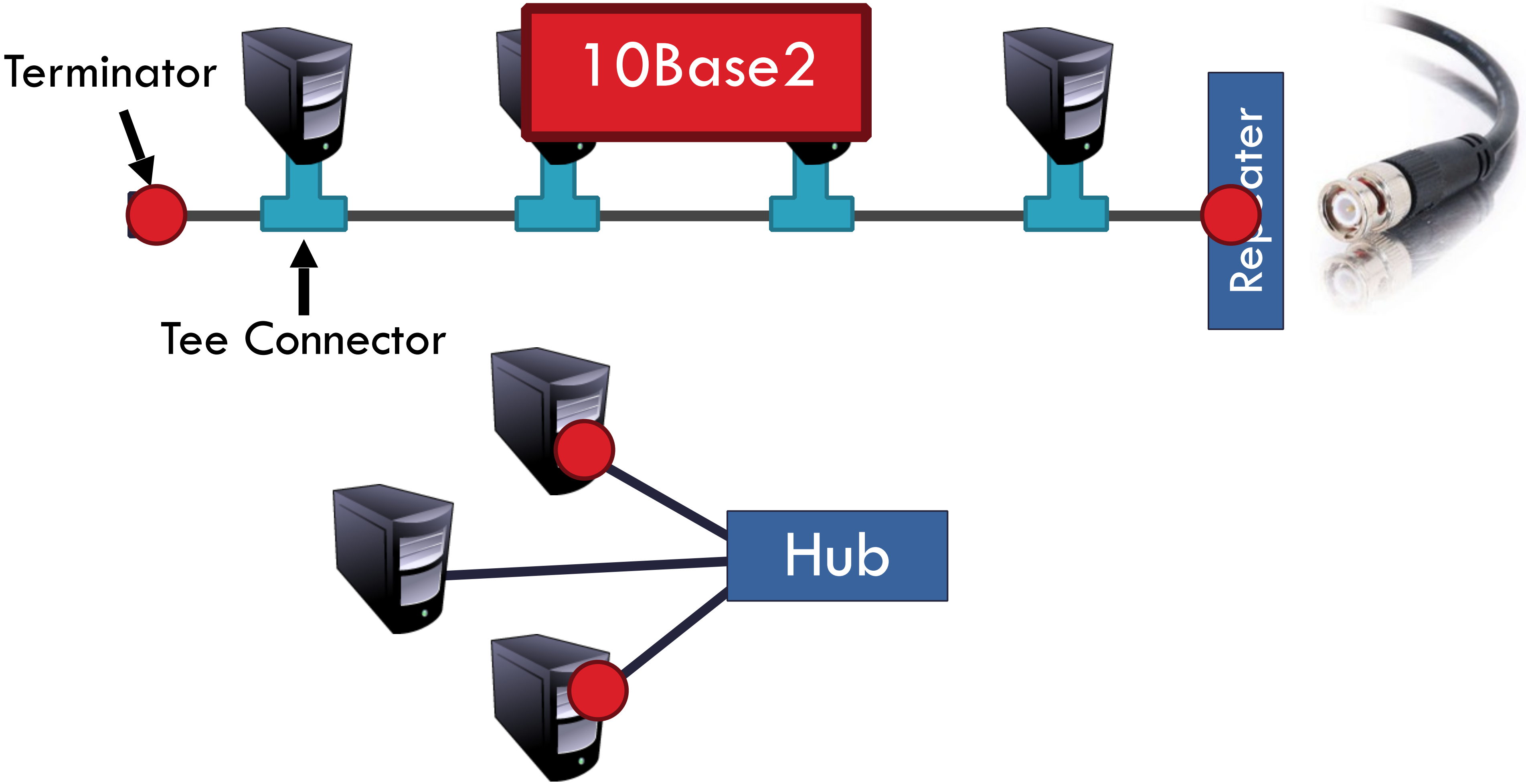
- Originally, Ethernet was a broadcast technology



# Broadcast Ethernet



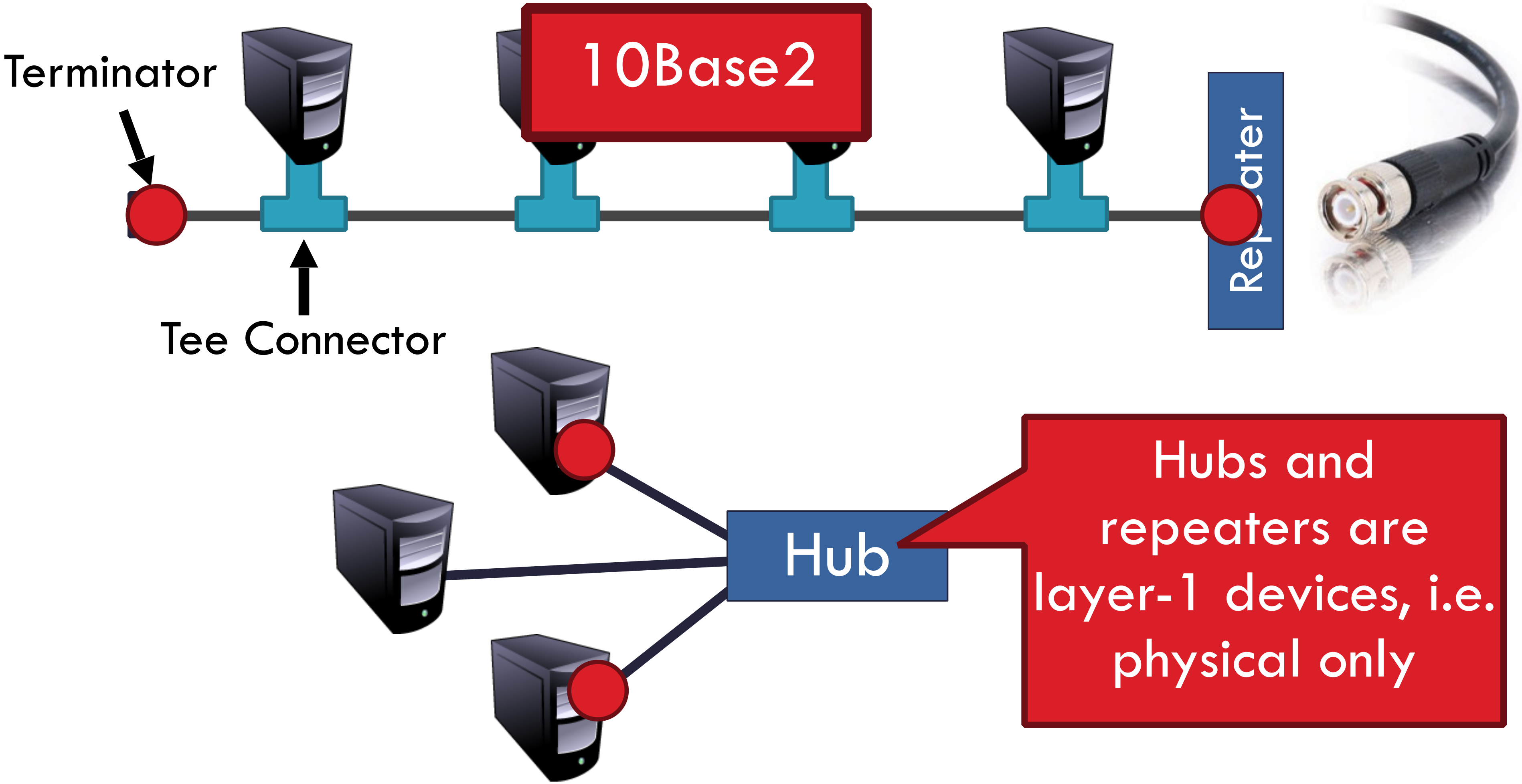
- Originally, Ethernet was a broadcast technology



# Broadcast Ethernet



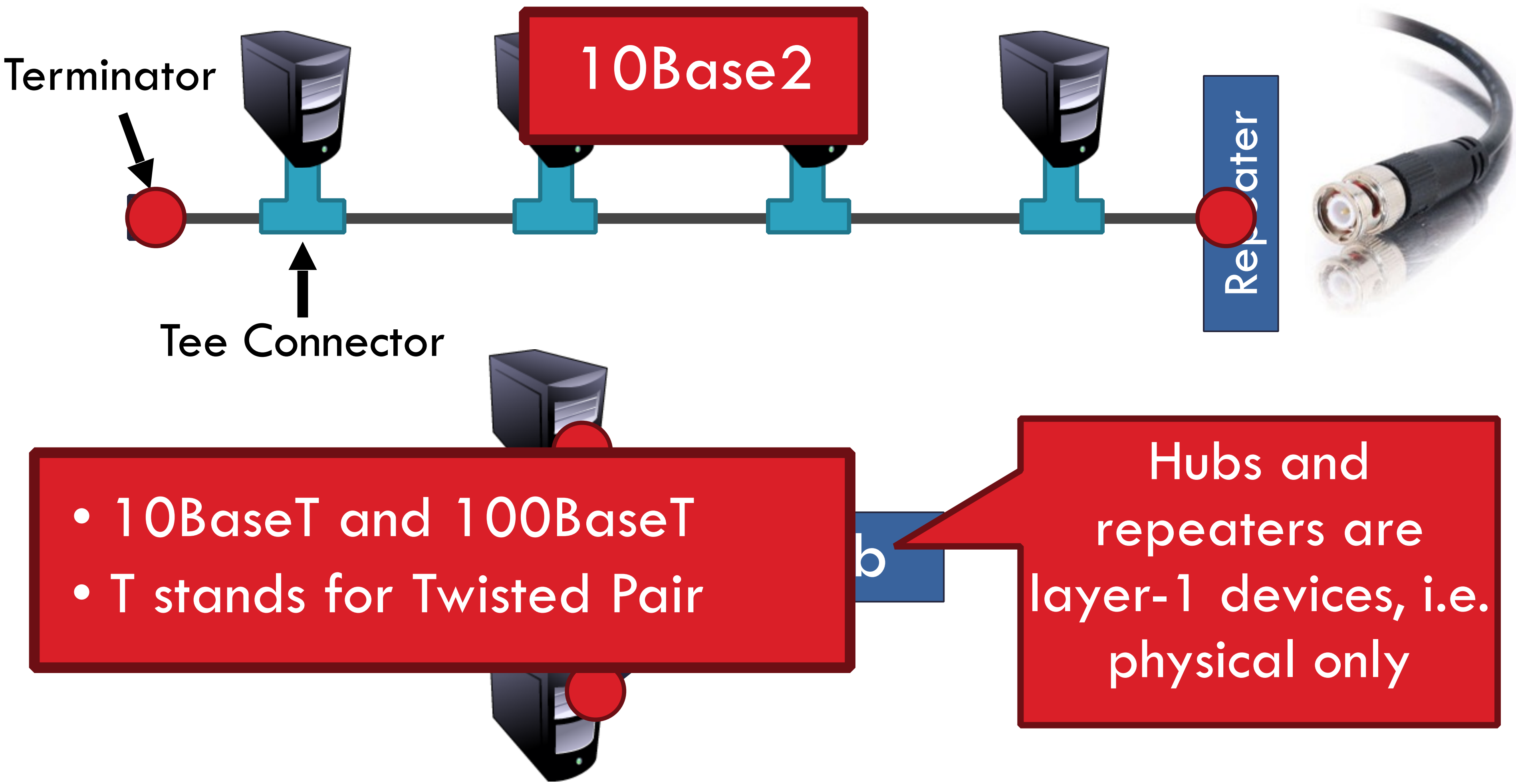
- Originally, Ethernet was a broadcast technology



# Broadcast Ethernet



- Originally, Ethernet was a broadcast technology



# CSMA/CD



- Carrier sense multiple access with collision detection
- Key insight: wired protocol allows us to sense the medium

# CSMA/CD



- Carrier sense multiple access with collision detection
- Key insight: wired protocol allows us to sense the medium
- Algorithm
  1. Sense for carrier

# CSMA/CD



- Carrier sense multiple access with collision detection
- Key insight: wired protocol allows us to sense the medium
- Algorithm
  1. Sense for carrier
  2. If carrier is present, wait for it to end
     - Sending would cause a collision and waste time

# CSMA/CD



- Carrier sense multiple access with collision detection
- Key insight: wired protocol allows us to sense the medium
- Algorithm
  1. Sense for carrier
  2. If carrier is present, wait for it to end
     - Sending would cause a collision and waste time
  3. Send a frame and sense for collision

# CSMA/CD



- Carrier sense multiple access with collision detection
- Key insight: wired protocol allows us to sense the medium
- Algorithm
  1. Sense for carrier
  2. If carrier is present, wait for it to end
     - Sending would cause a collision and waste time
  3. Send a frame and sense for collision
  4. If no collision, then frame has been delivered

# CSMA/CD



- Carrier sense multiple access with collision detection
- Key insight: wired protocol allows us to sense the medium
- Algorithm
  1. Sense for carrier
  2. If carrier is present, wait for it to end
     - Sending would cause a collision and waste time
  3. Send a frame and sense for collision
  4. If no collision, then frame has been delivered
  5. If collision, abort immediately
     - Why keep sending if the frame is already corrupted?

# CSMA/CD

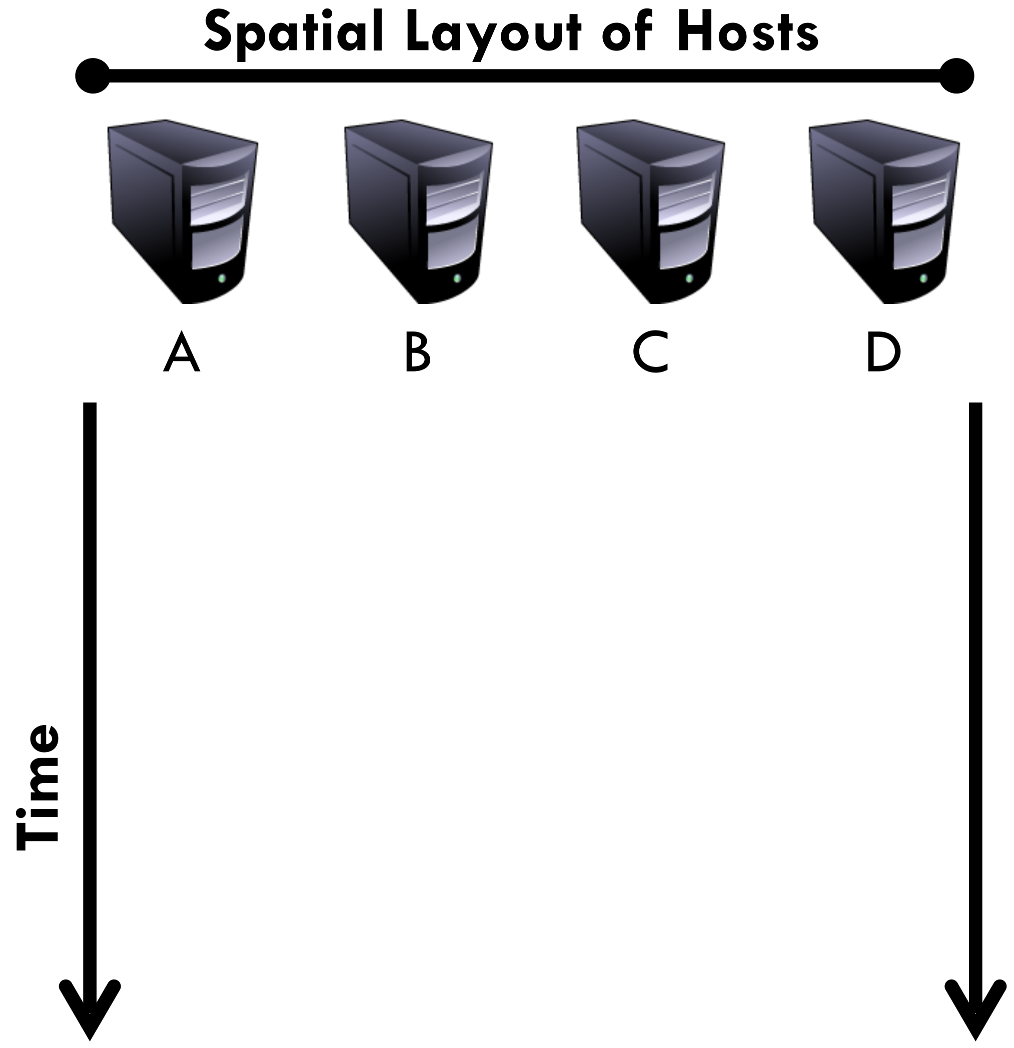


- Carrier sense multiple access with collision detection
- Key insight: wired protocol allows us to sense the medium
- Algorithm
  1. Sense for carrier
  2. If carrier is present, wait for it to end
     - Sending would cause a collision and waste time
  3. Send a frame and sense for collision
  4. If no collision, then frame has been delivered
  5. If collision, abort immediately
     - Why keep sending if the frame is already corrupted?
  6. Perform exponential backoff then retransmit

# CSMA/CD Collisions



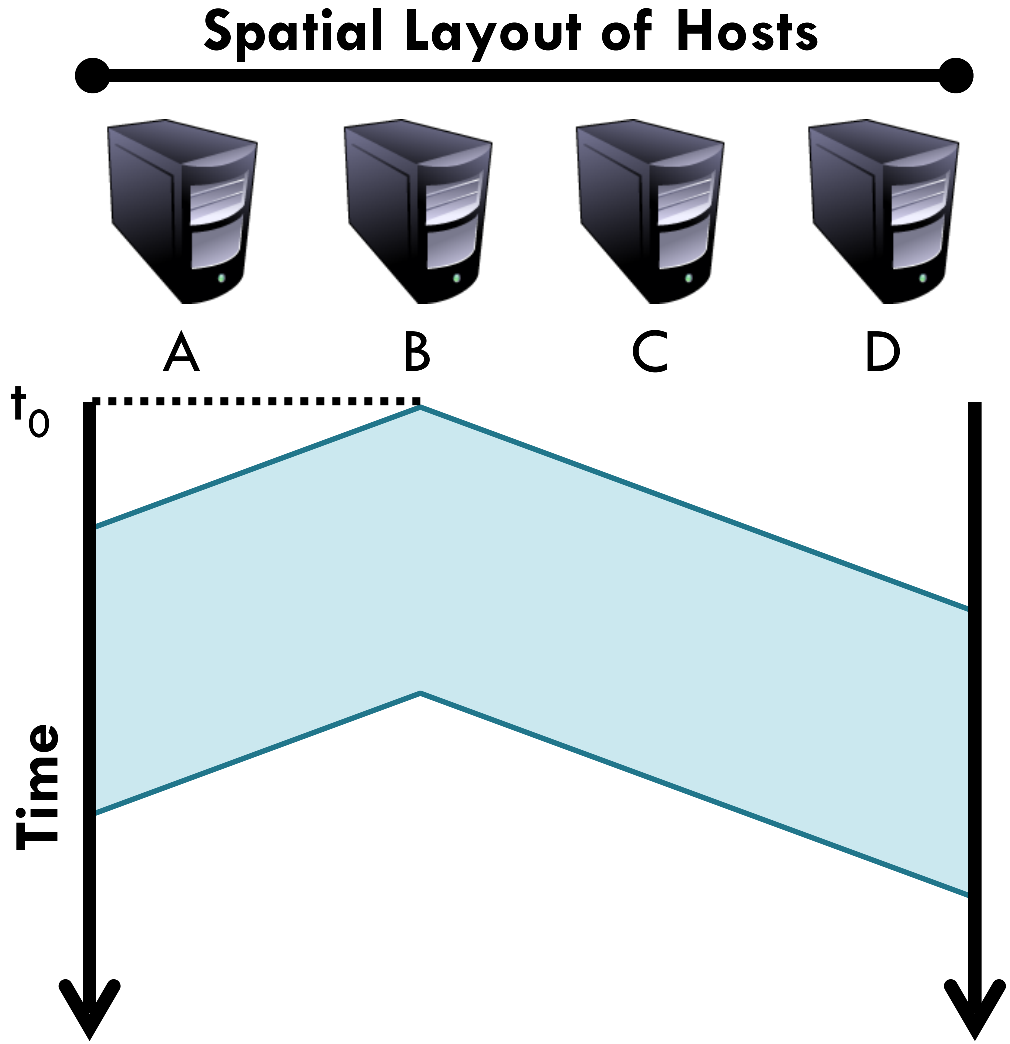
- Collisions can occur



# CSMA/CD Collisions



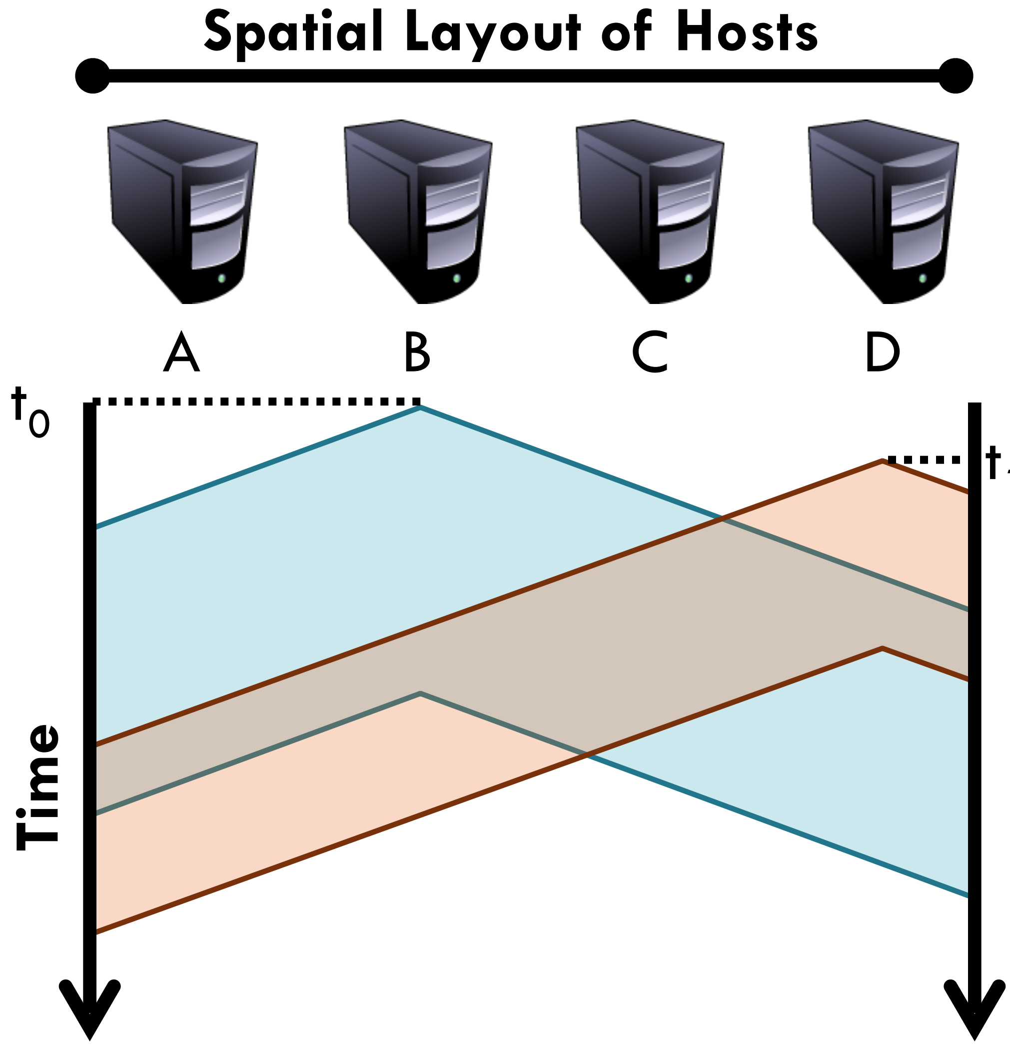
- Collisions can occur



# CSMA/CD Collisions



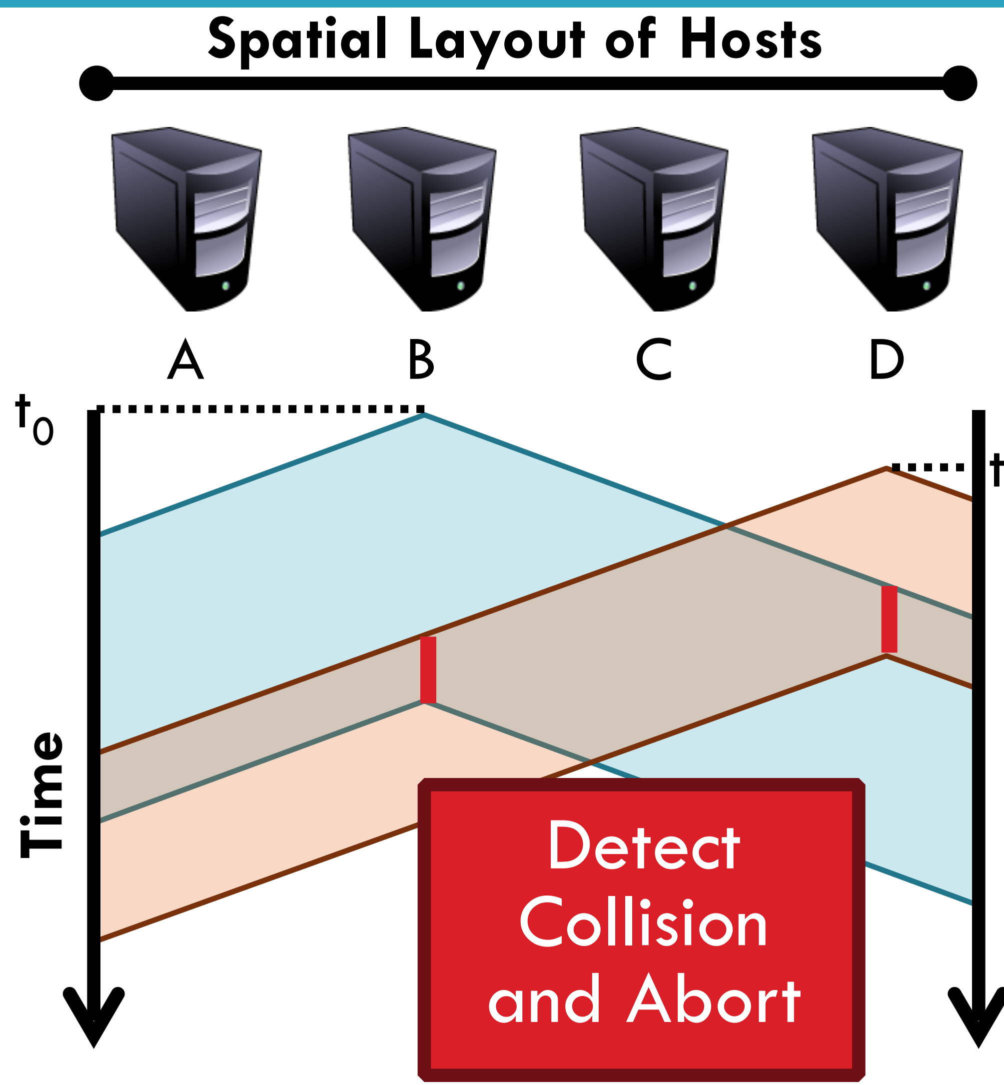
- Collisions can occur



# CSMA/CD Collisions



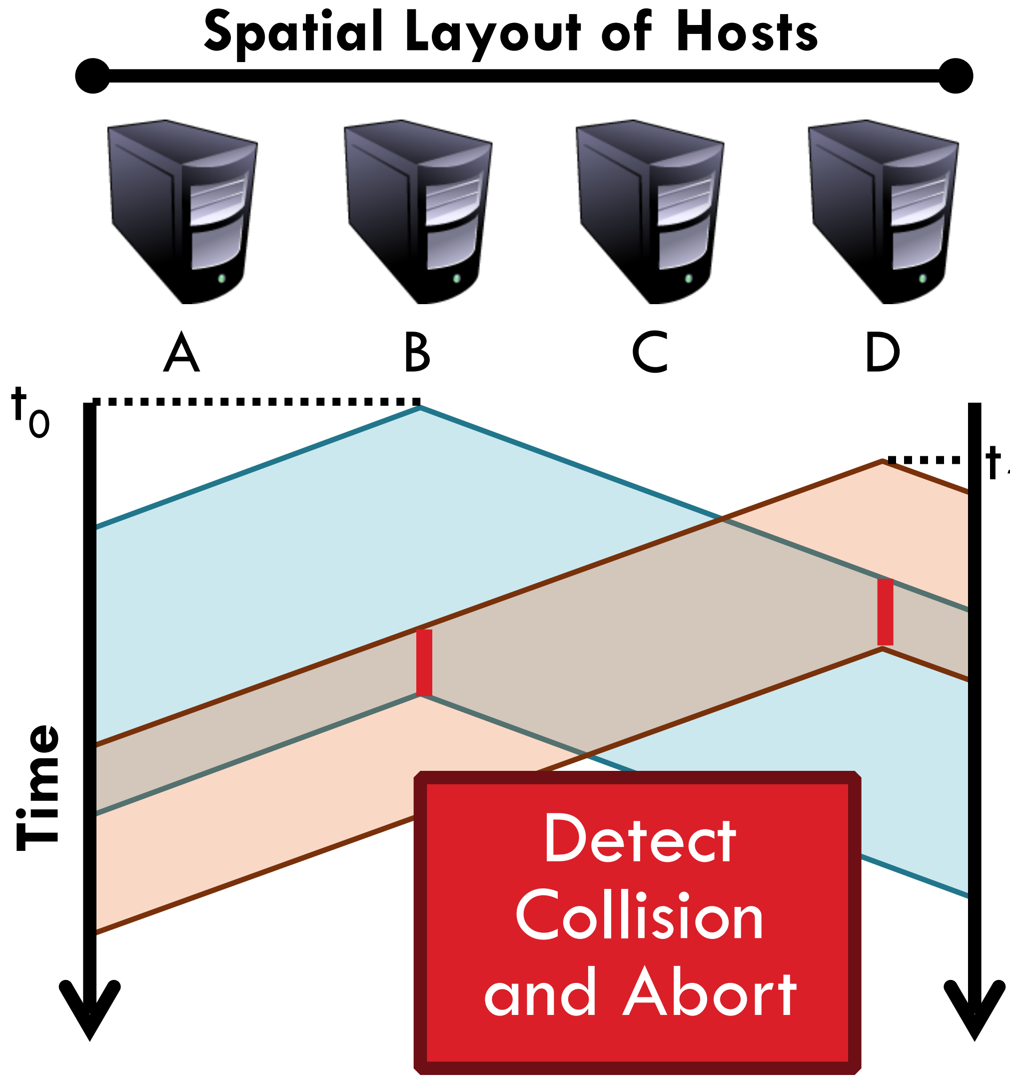
- Collisions can occur
- Collisions are quickly detected and aborted



# CSMA/CD Collisions



- Collisions can occur
- Collisions are quickly detected and aborted
- Note the role of distance, propagation delay, and frame length



# Exponential Backoff



- When a sender detects a collision, send "jam signal"
  - Make sure all hosts are aware of collision
  - Jam signal is 32 bits long (plus header overhead)
- Exponential backoff operates in multiples of 512 bits
  - Select $k \in [0, 2^n - 1]$, where $n$ = number of collisions
  - Wait $k$ * 51.2μs before retransmission
  - $n$ is capped at 10, frame dropped after 16 collisions
- Backoff time is divided into contention slots

# Exponential Backoff



- When a sender detects a collision, send "jam signal"
  - Make sure all hosts are aware of collision
  - Jam signal is 32 bits long (plus header overhead)
- Exponential backoff operates in multiples of 512 bits
  - Select $k \in [0, 2^n - 1]$, where $n$ = number of collisions
  - Wait $k$ * 51.2μs before retransmission
  - $n$ is capped at 10, frame dropped after 16 collisions
- Backoff time is divided into contention slots

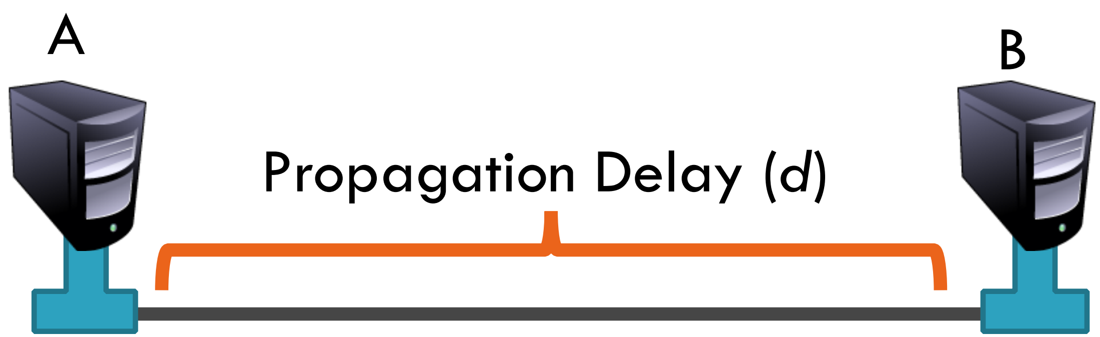
Remember this number

# Minimum Packet Sizes

# Minimum Packet Sizes



- Why is the minimum packet size 64 bytes?
  - To give hosts enough time to detect collisions
- What is the relationship between packet size and cable length?

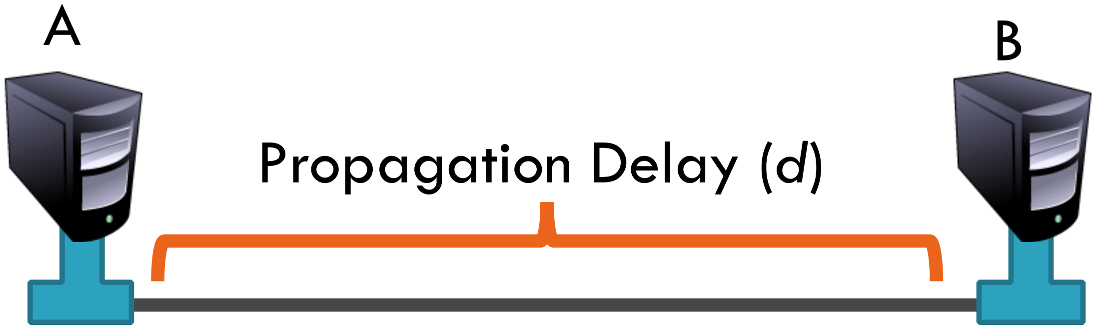


# Minimum Packet Sizes



- Why is the minimum packet size 64 bytes?
  - To give hosts enough time to detect collisions
- What is the relationship between packet size and cable length?

1. Time *t*: Host A starts transmitting

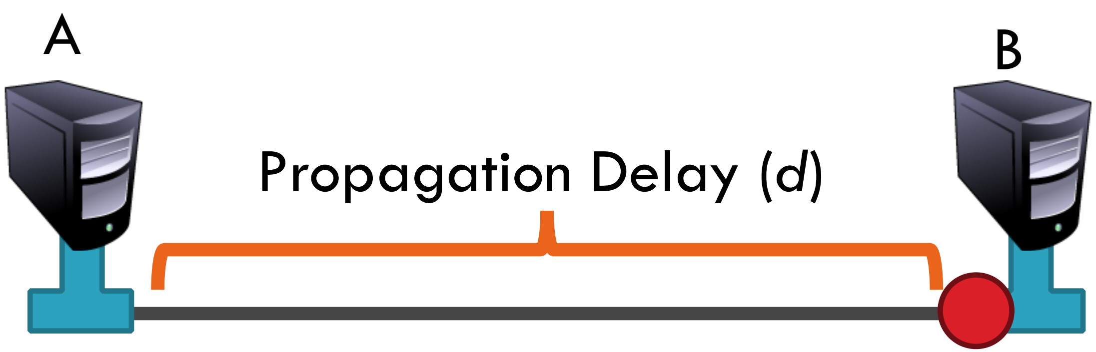


# Minimum Packet Sizes



- Why is the minimum packet size 64 bytes?
  - To give hosts enough time to detect collisions
- What is the relationship between packet size and cable length?

1. Time *t*: Host A starts transmitting

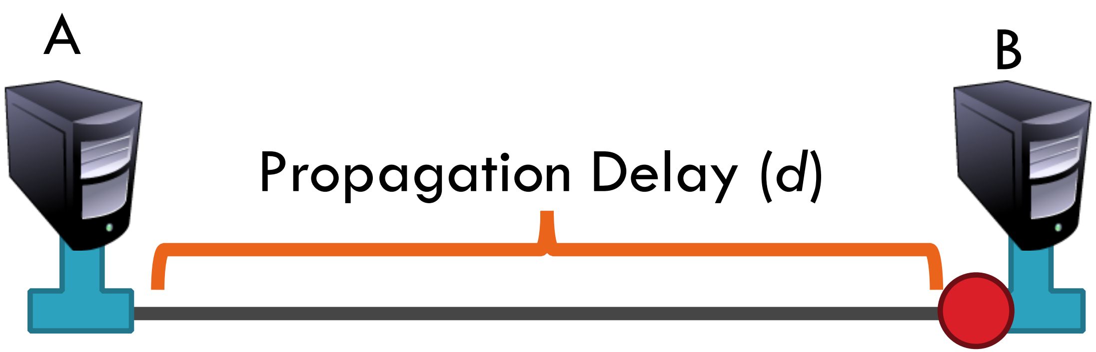


# Minimum Packet Sizes



- Why is the minimum packet size 64 bytes?
  - To give hosts enough time to detect collisions
- What is the relationship between packet size and cable length?

1. Time *t*: Host A starts transmitting
2. Time *t* + *d*: Host B starts transmitting

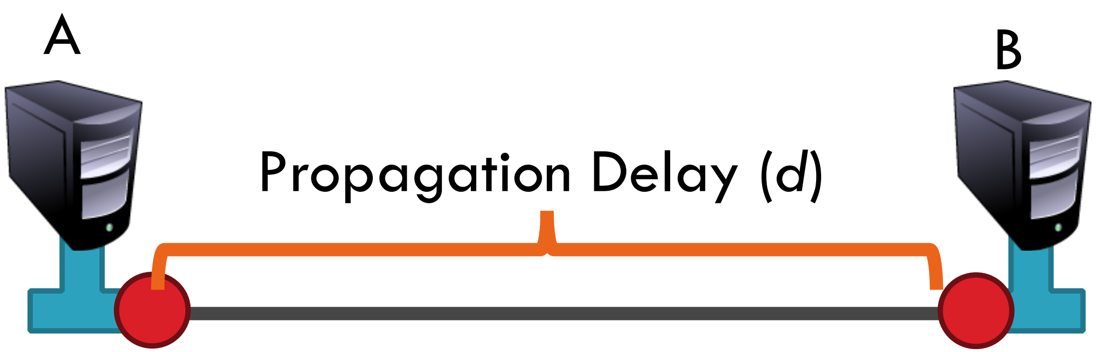


# Minimum Packet Sizes



- Why is the minimum packet size 64 bytes?
  - To give hosts enough time to detect collisions
- What is the relationship between packet size and cable length?

1. Time *t*: Host A starts transmitting
2. Time *t* + *d*: Host B starts transmitting

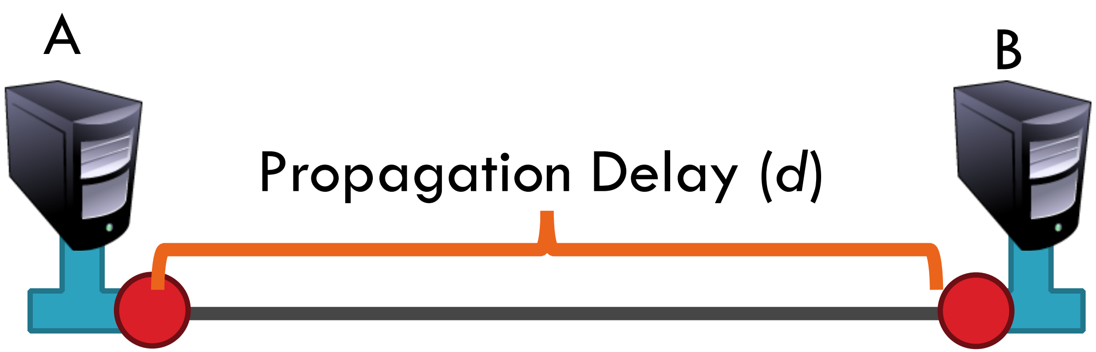


# Minimum Packet Sizes



- Why is the minimum packet size 64 bytes?
  - To give hosts enough time to detect collisions
- What is the relationship between packet size and cable length?

1. Time *t*: Host A starts transmitting
2. Time *t* + *d*: Host B starts transmitting
3. Time *t* + 2**d*: collision detected

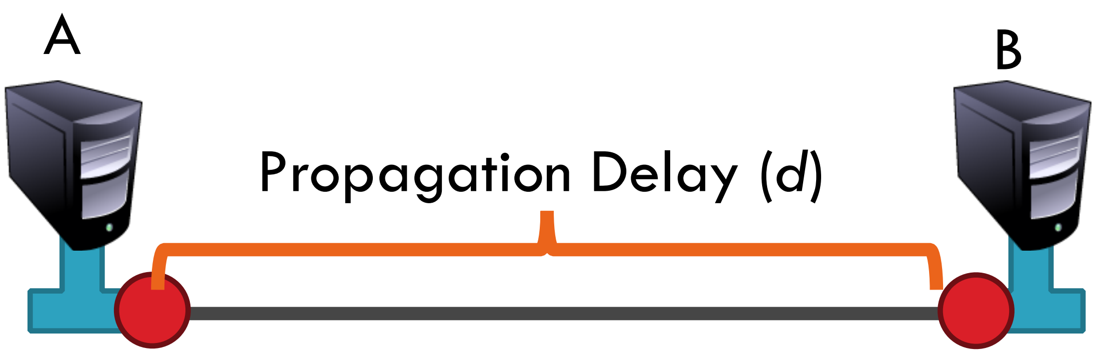


# Minimum Packet Sizes



- Why is the minimum packet size 64 bytes?
  - To give hosts enough time to detect collisions
- What is the relationship between packet size and cable length?

1. Time *t*: Host A starts transmitting
2. Time *t* + *d*: Host B starts transmitting
3. Time *t* + 2**d*: collision detected




min_frame_size*light_speed/(2*bandwidth) = max_cable_length

$(64B*8)*(2.5*10^8 mps)/(2*10^7 bps) = 6400$ meters

# Minimum Packet Sizes



- Why is the minimum packet size 64 bytes?
  - To give hosts enough time to detect collisions
- What is the relationship between packet size and cable length?

- 10 Mbps Ethernet
- Packet and cable lengths change for faster Ethernet standards

1. Time *t*: Host A st transmitting
2. Time *t* + *d*: Host transmitting
3. Time *t* + 2**d*: collision detected

min_frame_size*light_speed/(2*b dwidth) = max_cable_length

$(64B*8)*(2.5*10^8 mps)/(2*10^7 bps) = 6400$ meters

# Cable Length Examples



min_frame_size*light_speed/(2*bandwidth) = max_cable_length

$(64B*8)*(2.5*10^8 mps)/(2*10Mbps) = 6400$ meters

- What is the max cable length if min packet size were changed to 1024 bytes?

# Cable Length Examples



min_frame_size*light_speed/(2*bandwidth) = max_cable_length

$(64B*8)*(2.5*10^8 mps)/(2*10 Mbps) = 6400$ meters

- What is the max cable length if min packet size were changed to 1024 bytes?
  - 102.4 kilometers

# Cable Length Examples



min_frame_size*light_speed/(2*bandwidth) = max_cable_length

(64B*8)*(2.5*$10^8$mps)/(2*10Mbps) = 6400 meters

- What is the max cable length if min packet size were changed to 1024 bytes?
  - 102.4 kilometers
- What is max cable length if bandwidth were changed to 1 Gbps ?

# Cable Length Examples



min_frame_size*light_speed/(2*bandwidth) = max_cable_length

$(64B*8)*(2.5*10^8 mps)/(2*10Mbps) = 6400$ meters

- What is the max cable length if min packet size were changed to 1024 bytes?
  - 102.4 kilometers
- What is max cable length if bandwidth were changed to 1 Gbps ?
  - 64 meters

# Cable Length Examples



min_frame_size*light_speed/(2*bandwidth) = max_cable_length

$(64B*8)*(2.5*10^8 mps)/(2*10Mbps) = 6400$ meters

- What is the max cable length if min packet size were changed to 1024 bytes?
  - 102.4 kilometers
- What is max cable length if bandwidth were changed to 1 Gbps ?
  - 64 meters
- What if you changed min packet size to 1024 bytes and bandwidth to 1 Gbps?

# Cable Length Examples



min_frame_size*light_speed/(2*bandwidth) = max_cable_length

$(64B*8)*(2.5*10^8 mps)/(2*10Mbps) = 6400$ meters

- What is the max cable length if min packet size were changed to 1024 bytes?
  - 102.4 kilometers
- What is max cable length if bandwidth were changed to 1 Gbps ?
  - 64 meters
- What if you changed min packet size to 1024 bytes and bandwidth to 1 Gbps?
  - 1024 meters

# Exponential Backoff, Revisited



- Remember the 512 bit backoff timer?
- Minimum Ethernet packet size is also 512 bits
  - 64 bytes * 8 = 512 bits
- Coincidence? Of course not.
  - If the backoff time was <512 bits, a sender who waits and another who sends immediately can still collide

# Maximum Packet Size



- Maximum Transmission Unit (MTU): 1500 bytes
- Pros:
  - Bit errors in long packets incur significant recovery penalty
- Cons:
  - More bytes wasted on header information
  - Higher per packet processing overhead
- Datacenters shifting towards Jumbo Frames
  - 9000 bytes per packet

# Long Live Ethernet



- Today's Ethernet is switched
  - More on this later
- 1Gbit and 10Gbit Ethernet now common
  - 100Gbit on the way
  - Uses same old packet header
  - Full duplex (send and receive at the same time)
  - Auto negotiating (backwards compatibility)
  - Can also carry power

# Outline

- Framing
- Error Checking and Reliability
- Media Access Control
  - 802.3 Ethernet
  - 802.11 Wifi

# 802.3 vs. Wireless

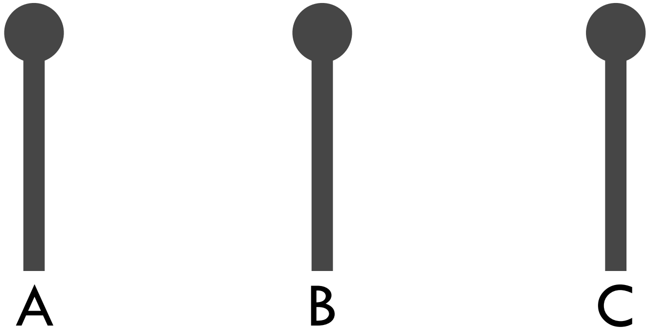


- Ethernet has one shared collision domain
  - All hosts on a LAN can observe all transmissions
- Wireless radios have small range compared to overall system
  - Collisions are local
  - Collision are at the receiver, not the sender
  - Carrier sense (CS in CSMA) plays a different role
- 802.11 uses CSMA/CA not CSMA/CD
  - Collision avoidance, rather than collision detection

# Hidden Terminal Problem



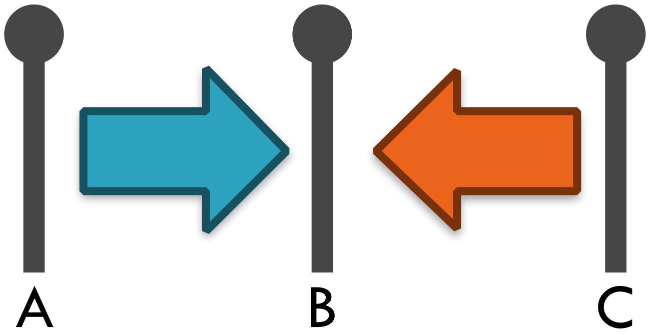
- Radios on the same network cannot always hear each other



# Hidden Terminal Problem



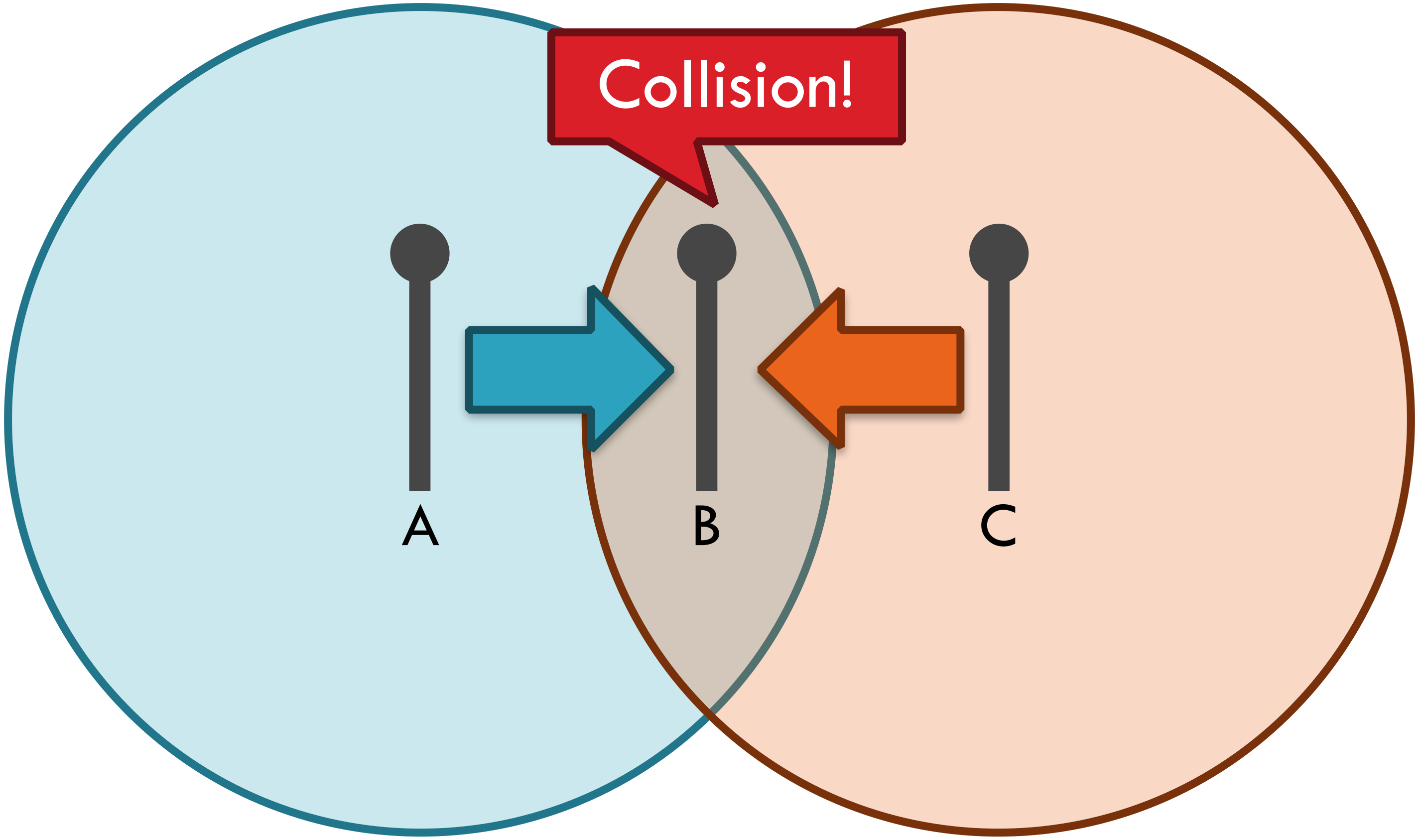
- Radios on the same network cannot always hear each other



# Hidden Terminal Problem



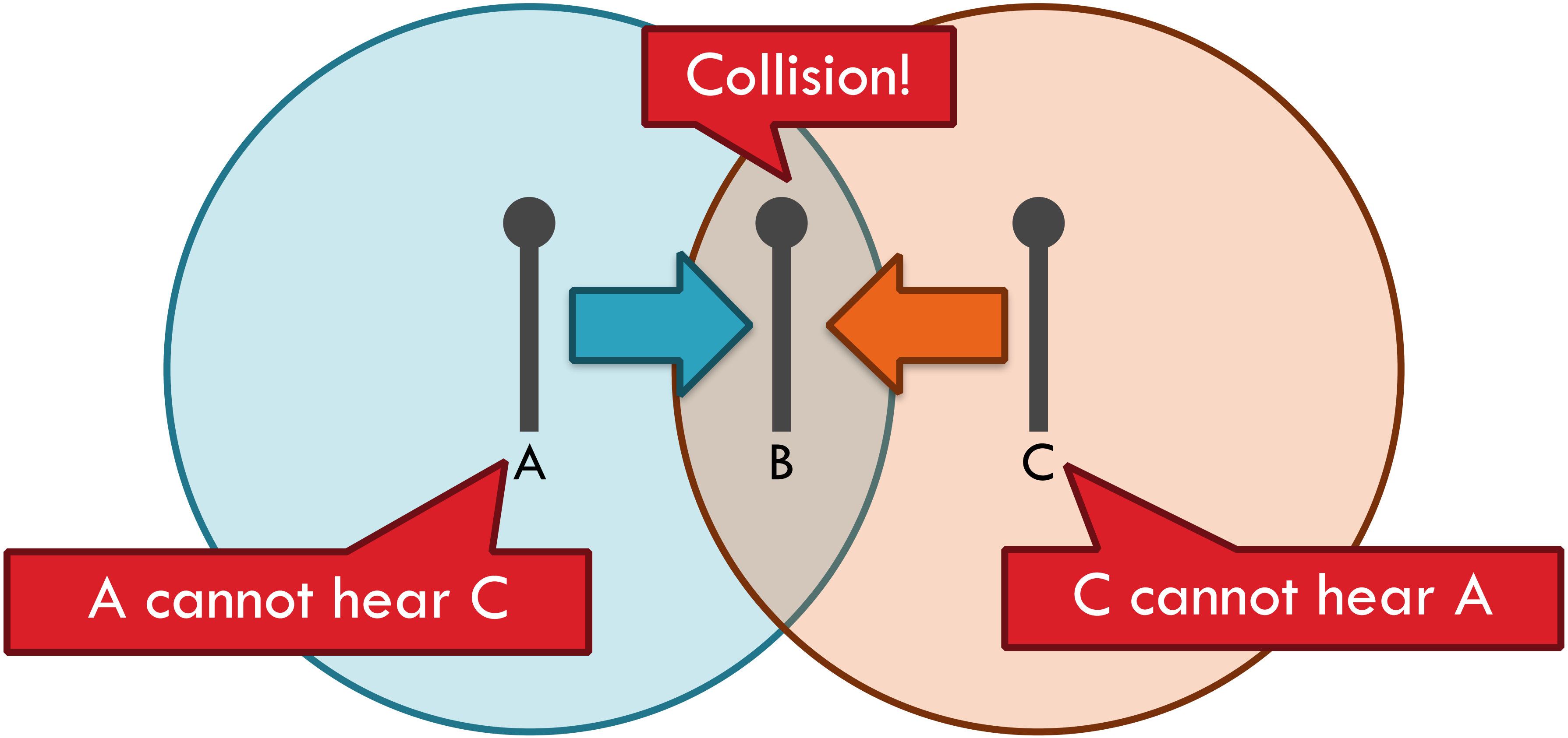
- Radios on the same network cannot always hear each other



# Hidden Terminal Problem



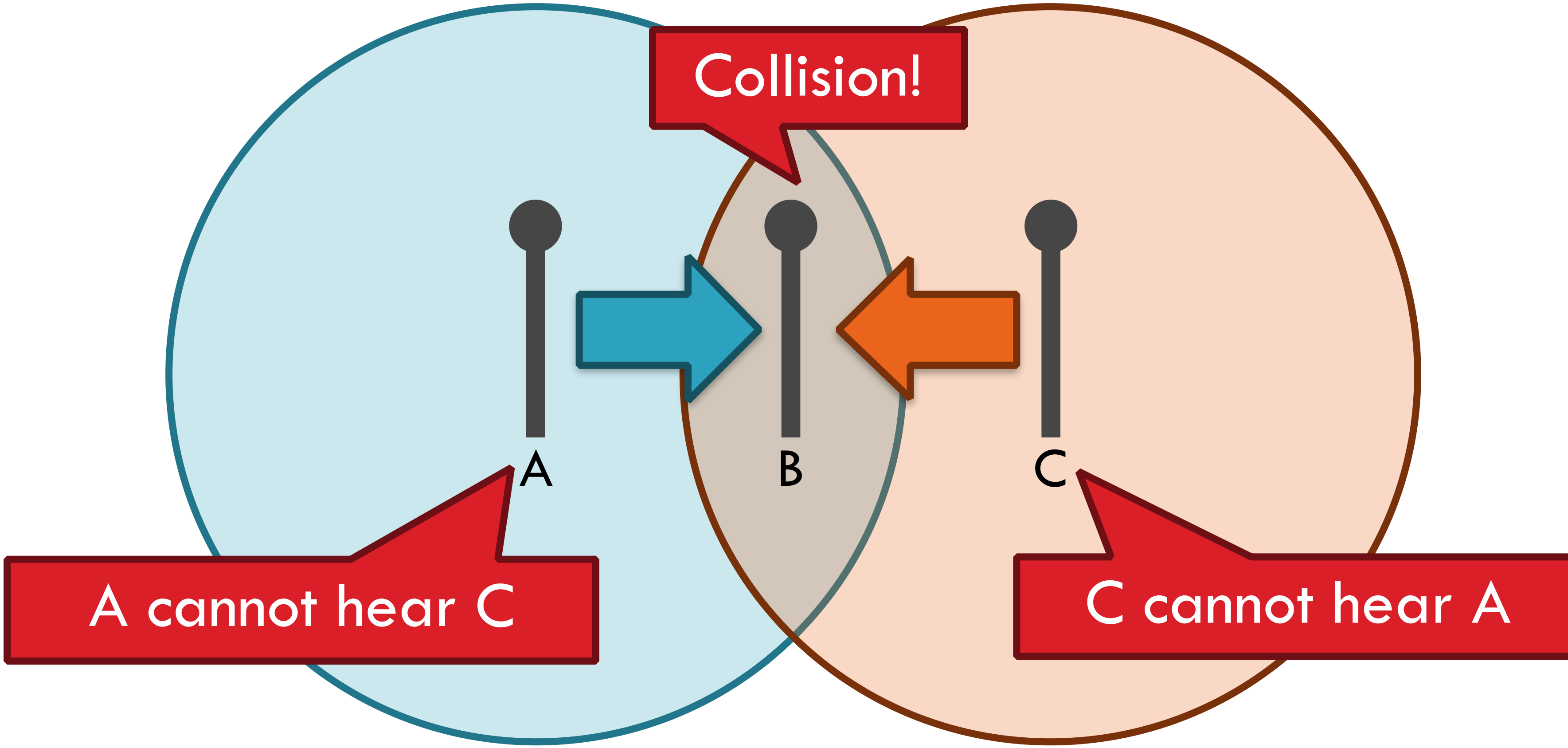
- Radios on the same network cannot always hear each other



# Hidden Terminal Problem



- Radios on the same network cannot always hear each other

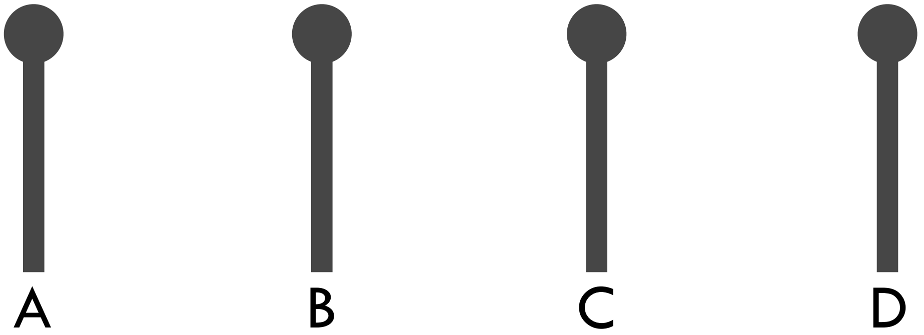



- Hidden terminals mean that sender-side collision detection is useless

# Exposed Terminal Problem



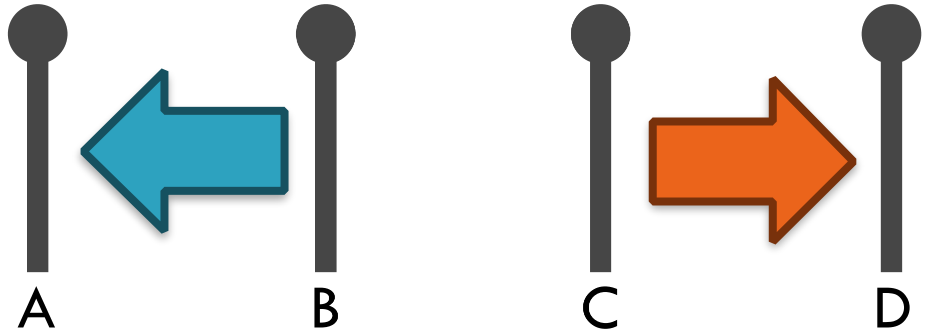
- Carrier sensing is problematic in wireless



# Exposed Terminal Problem



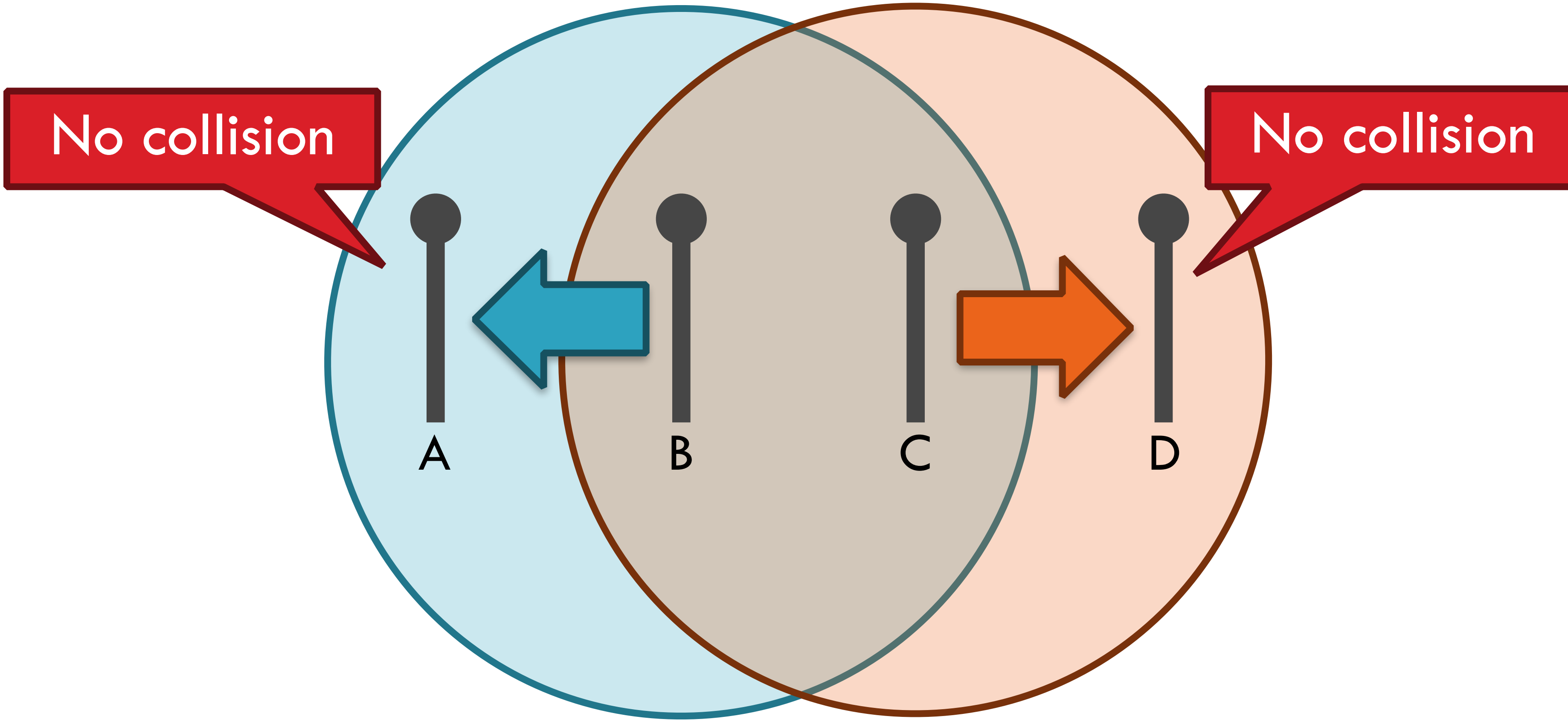
- Carrier sensing is problematic in wireless



# Exposed Terminal Problem



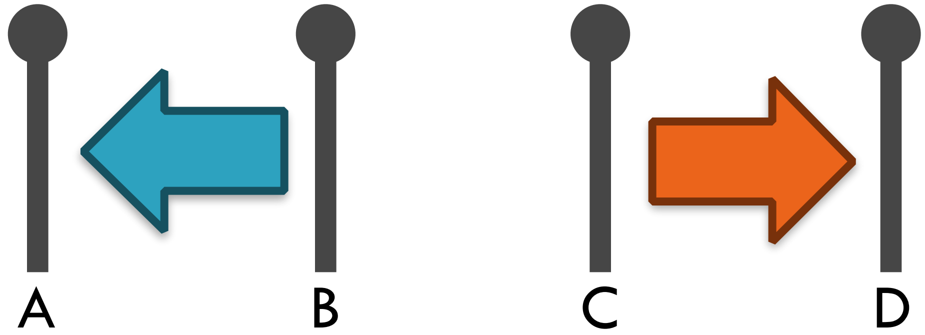
- Carrier sensing is problematic in wireless



# Exposed Terminal Problem



- Carrier sensing is problematic in wireless

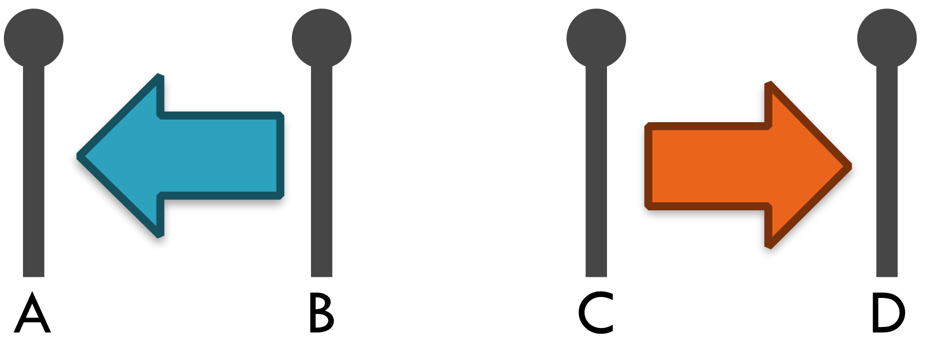


# Exposed Terminal Problem



- Carrier sensing is pr

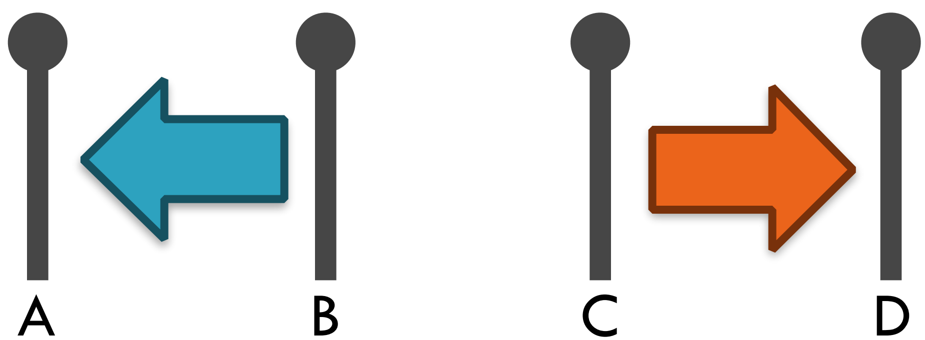
Carrier sense detects a busy channel



# Exposed Terminal Problem



- Carrier sensing is pr

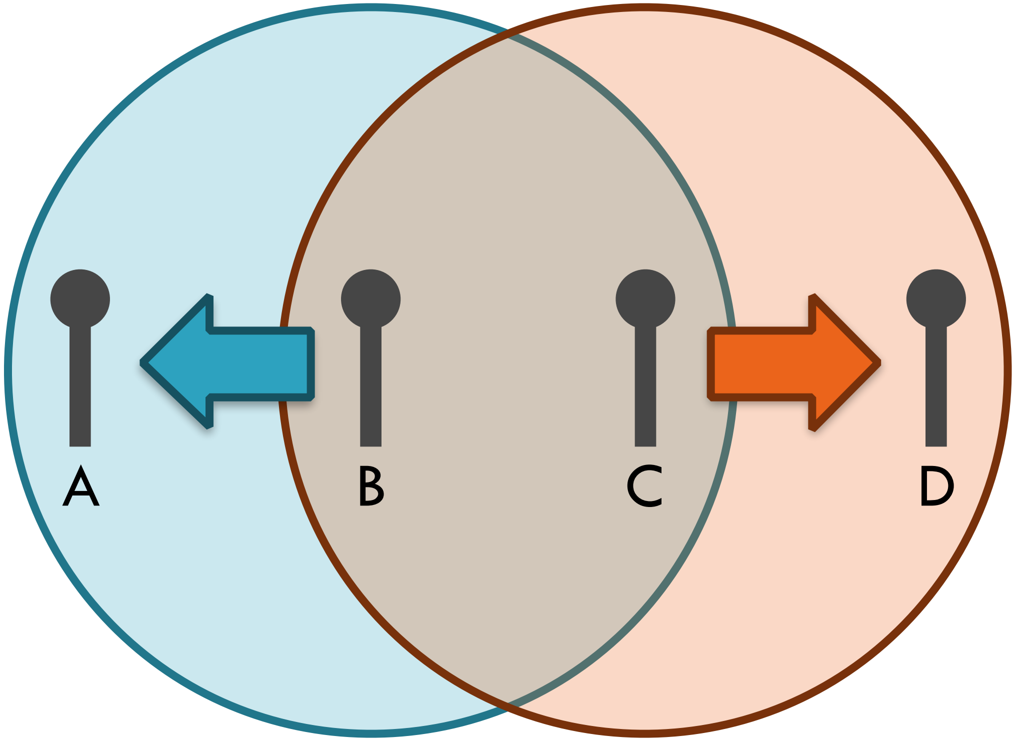



- Carrier sense can erroneously reduce utilization

# Reachability in Wireless



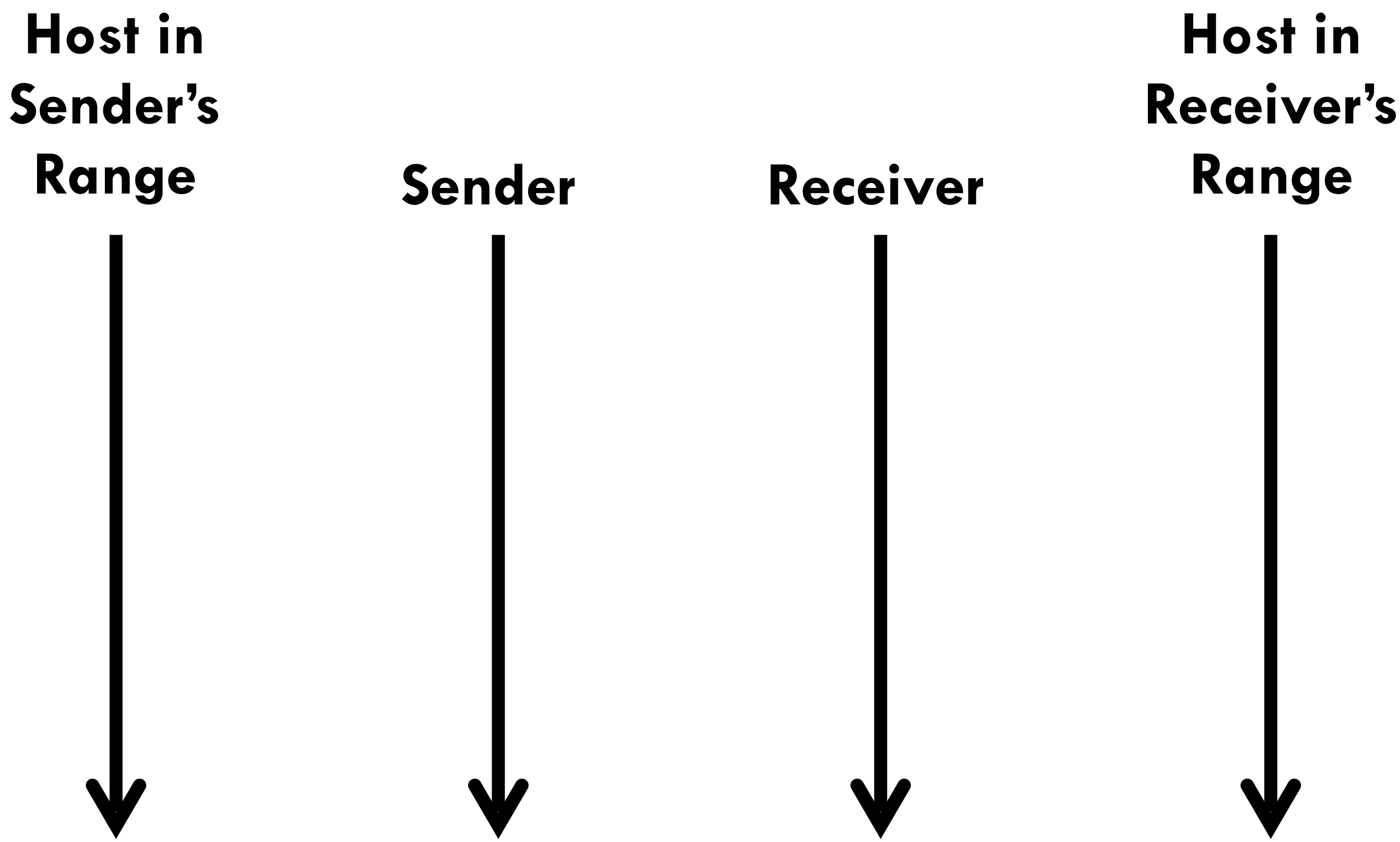
- High level problem:
  - Reachability in wireless is not transitive
  - Just because A can reach B, and B can reach C, doesn't mean A can reach C



# MACA



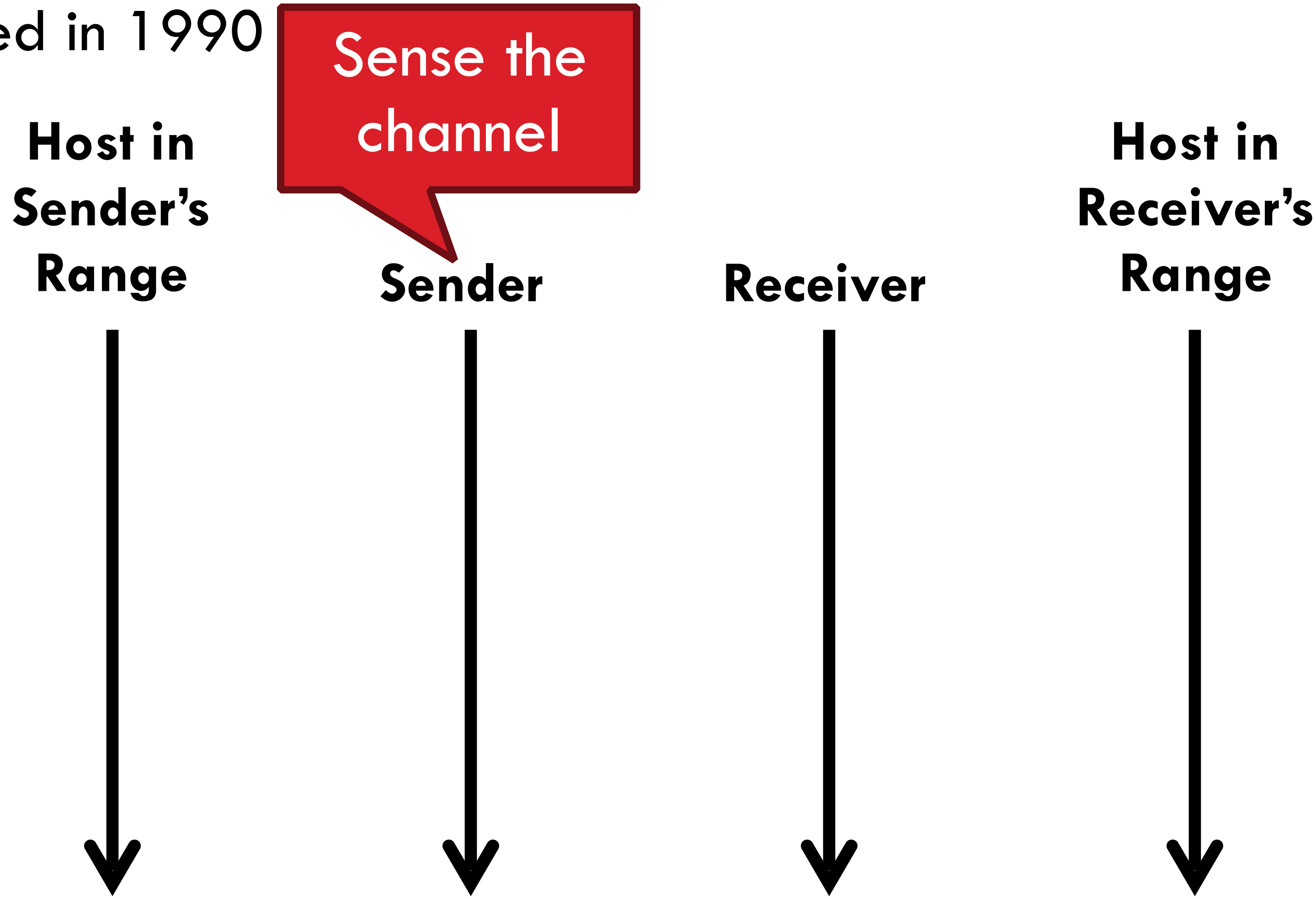
- **M**ultiple **A**ccess with **C**ollision **A**voidance
  - Developed in 1990



# MACA



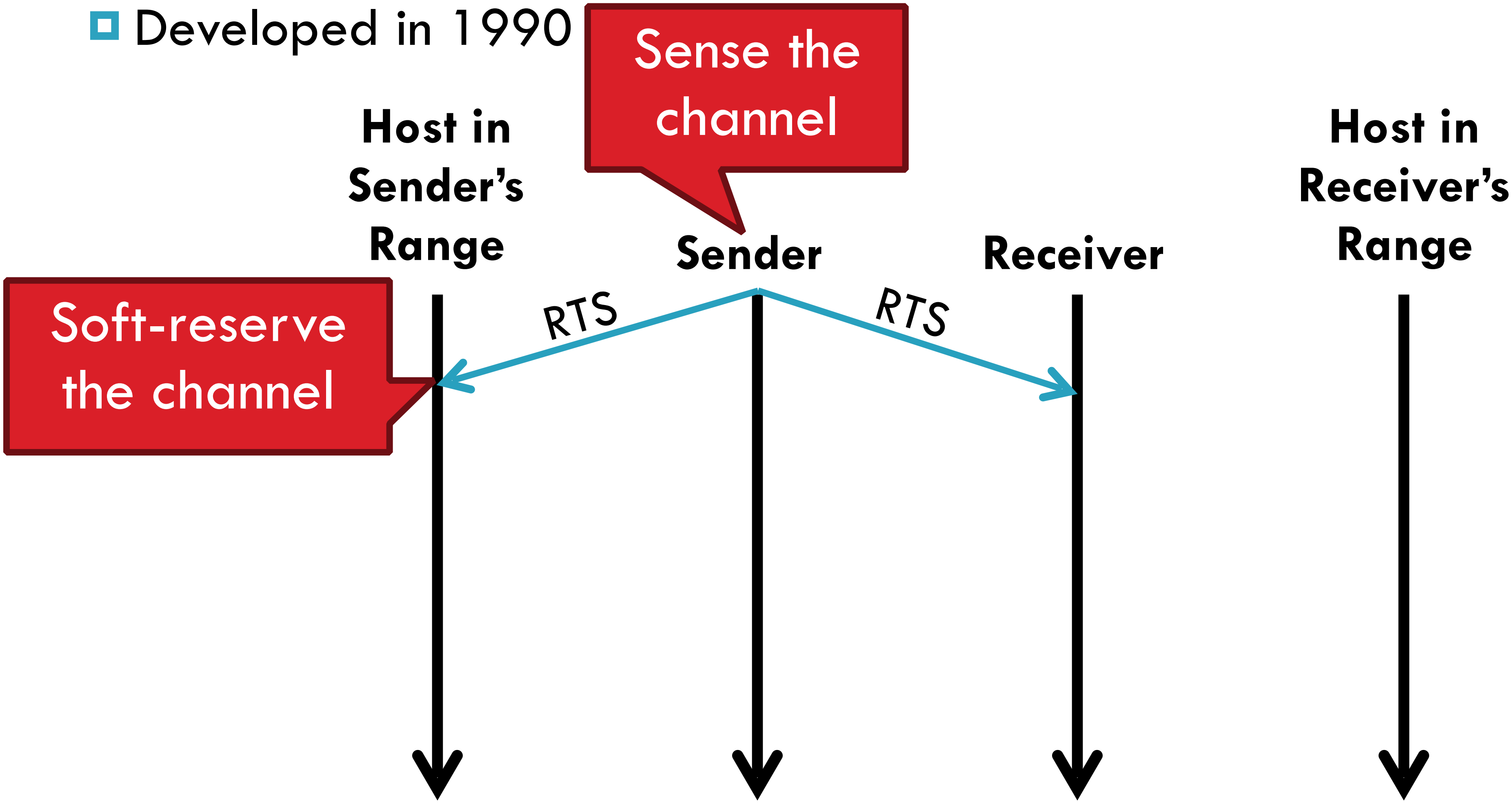
- **M**ultiple **A**ccess with **C**ollision **A**voidance
  - Developed in 1990



# MACA



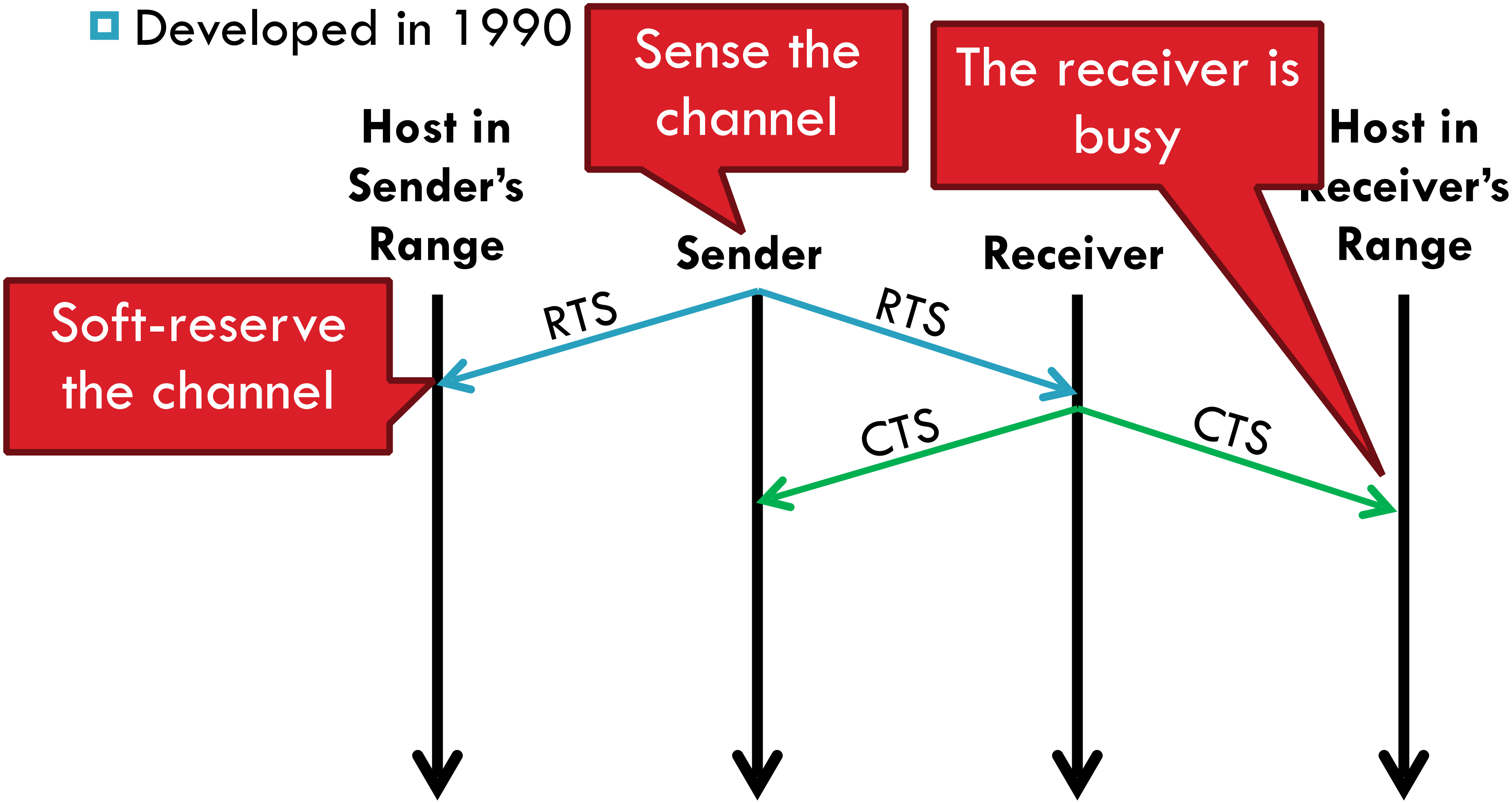
- **M**ultiple **A**ccess with **C**ollision **A**voidance
  - Developed in 1990



# MACA



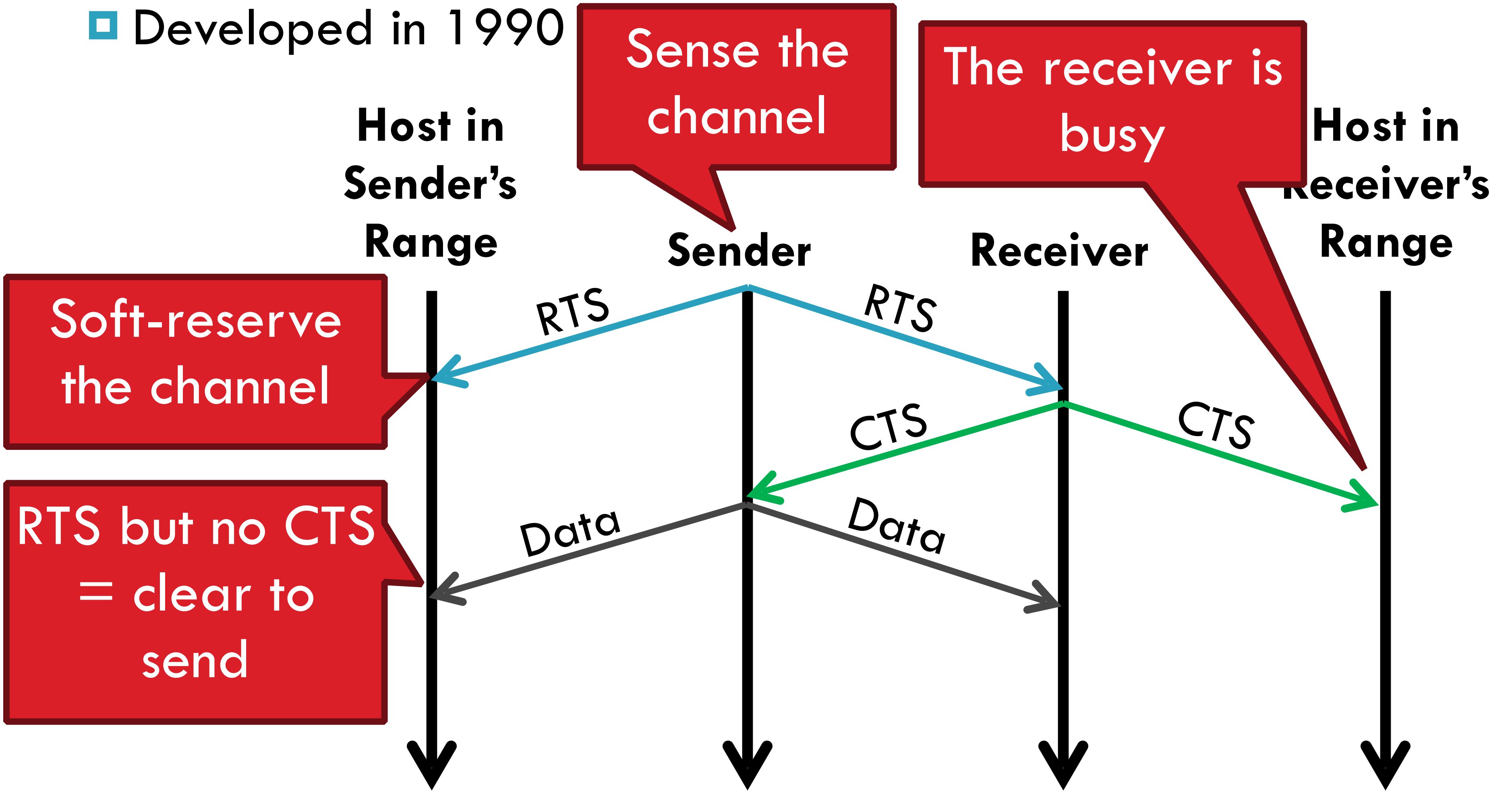
- **M**ultiple **A**ccess with **C**ollision **A**voidance
  - Developed in 1990



# MACA



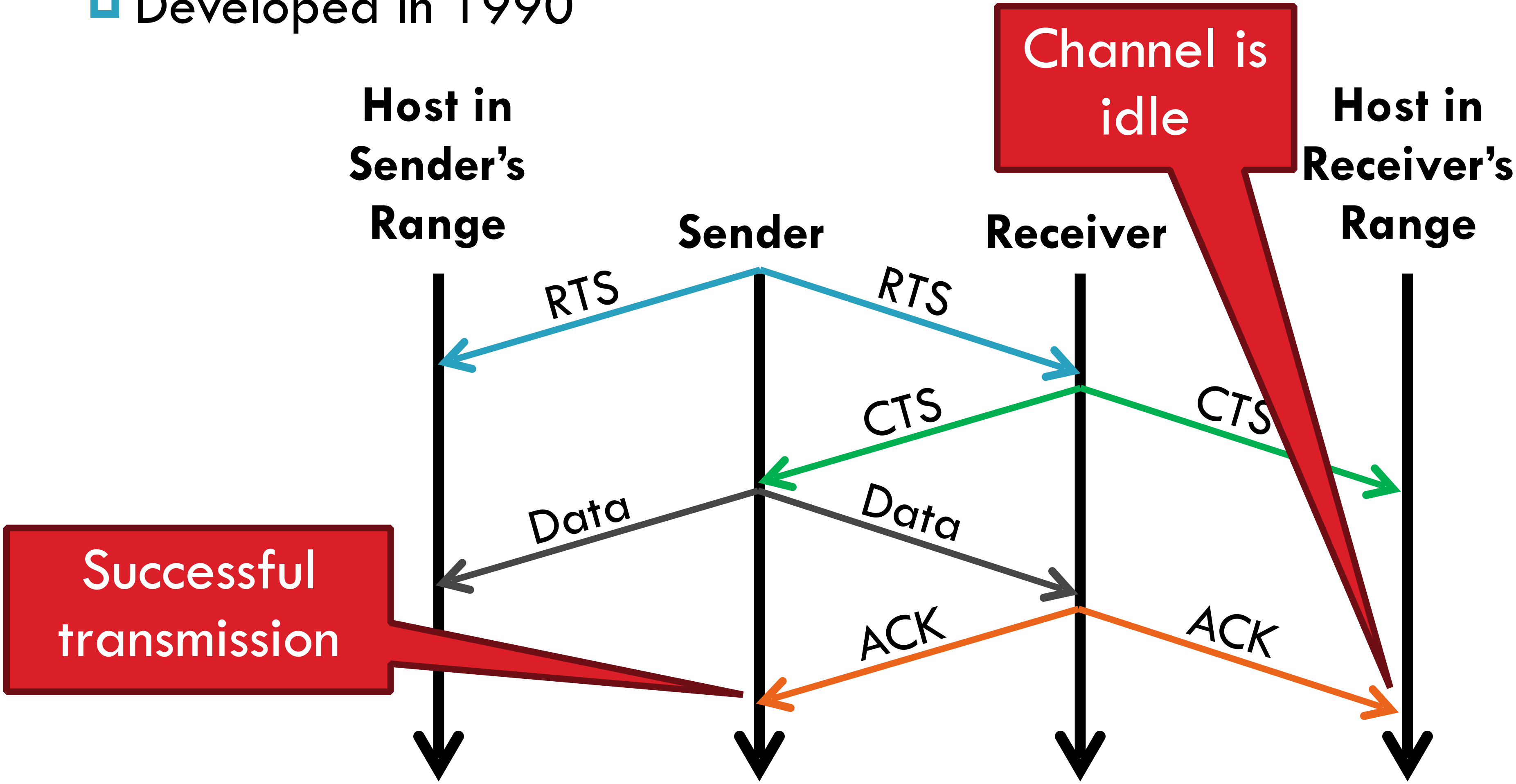
- **M**ultiple **A**ccess with **C**ollision **A**voidance
  - Developed in 1990



# MACA



- **M**ultiple **A**ccess with **C**ollision **A**voidance
  - Developed in 1990



# Collisions in MACA



- What if sender does not receive CTS or ACK?
  - Assume collision
  - Enter exponential backoff mode

# 802.11b

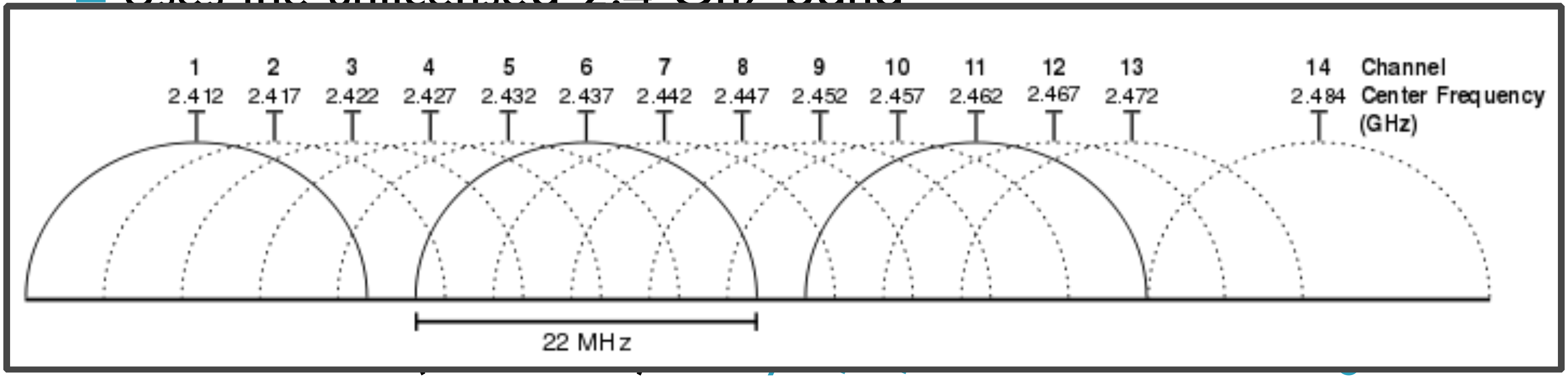


- 802.11
  - Uses CSMA/CA, not MACA
- 802.11b
  - Introduced in 1999
  - Uses the unlicensed 2.4 Ghz band
    - Same band as cordless phones, microwave ovens
  - Complementary code keying (CCK) modulation scheme
  - 5.5 and 11 Mbps data rates
    - Practical throughput with TCP is only 5.9 Mbps
  - 11 channels (in the US). Only 1, 6, and 11 are orthogonal

# 802.11b



- 802.11
  - Uses CSMA/CA, not MACA
- 802.11b
  - Introduced in 1999
  - Uses the unlicensed 2.4 Ghz band



# 802.11a/g



- 802.11a
  - Uses the 5 Ghz band
  - 6, 9, 12, 18, 24, 36, 48, 54 Mbps
  - Switches from CCK to Orthogonal Frequency Division Multiplexing (OFDM)
    - Each frequency is orthogonal
- 802.11g
  - Introduced in 2003
  - Uses OFDM to improve performance (54 Mbps)
  - Backwards compatible with 802.11b
    - Warning: b devices cause g networks to fall back to CCK

# 802.11n/ac



- 802.11n
  - Introduced in 2009
  - Multiple Input Multiple Output (MIMO)
    - Multiple send and receive antennas per devices (up to four)
    - Data stream is multiplexed across all antennas
  - Maximum 600 Mbps transfer rate (in a 4x4 configuration)
  - 300 Mbps is more common (2x2 configuration)
- 802.11ac
  - Final approval in Feb 2014
  - 8x8 MIMO in the 5 GHz band, 500 Mbps – 1 GBps rates

# 802.11 Media Access



- MACA-style RTS/CTS is optional
- Distributed Coordination Function (DCF) based on…
  - Inter Frame Spacing (IFS)
    - DIFS – low priority, normal data packets
    - PIFS – medium priority, used with Point Coordination Function (PCF)
    - SIFS – high priority, control packets (RTS, CTS, ACK, etc.)
  - Contention interval: random wait time

# 802.11 Media Access

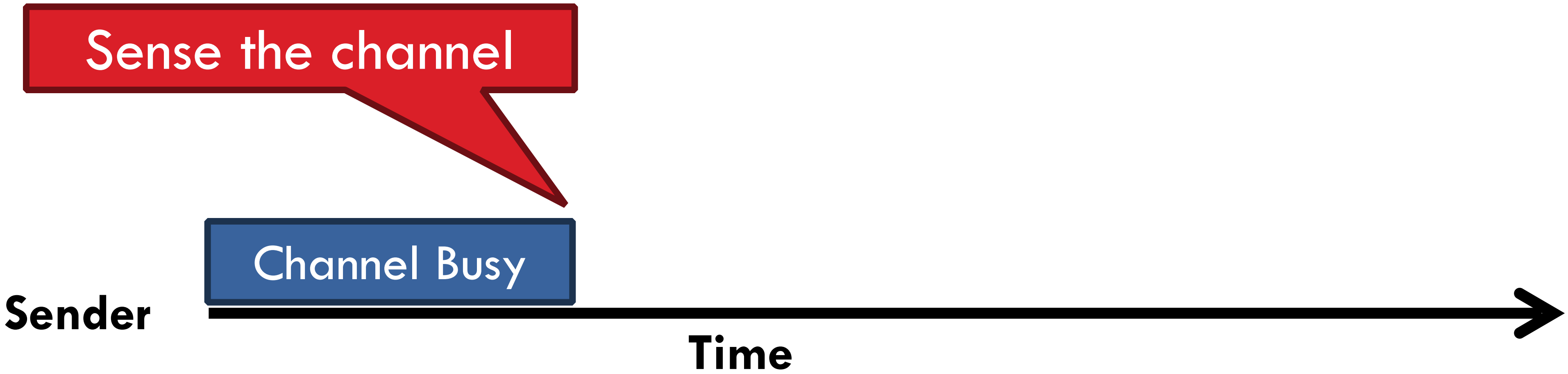


- MACA-style RTS/CTS is optional
- Distributed Coordination Function (DCF) based on…
  - Inter Frame Spacing (IFS)
    - DIFS – low priority, normal data packets
    - PIFS – medium priority, used with Point Coordination Function (PCF)
    - SIFS – high priority, control packets (RTS, CTS, ACK, etc.)
  - Contention interval: random wait time

# 802.11 Media Access



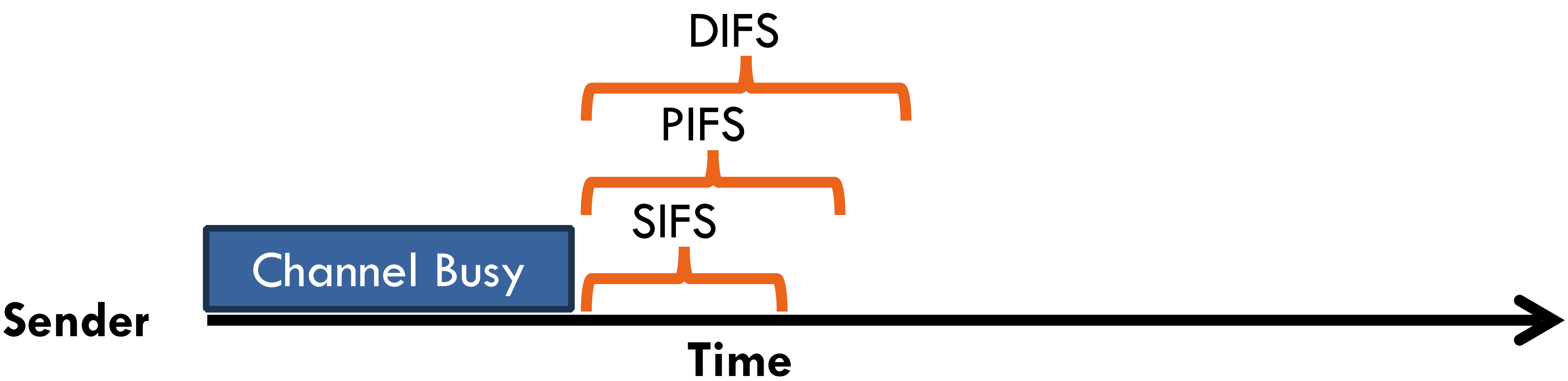
- MACA-style RTS/CTS is optional
- Distributed Coordination Function (DCF) based on…
  - Inter Frame Spacing (IFS)
    - DIFS – low priority, normal data packets
    - PIFS – medium priority, used with Point Coordination Function (PCF)
    - SIFS – high priority, control packets (RTS, CTS, ACK, etc.)
  - Contention interval: random wait time



# 802.11 Media Access



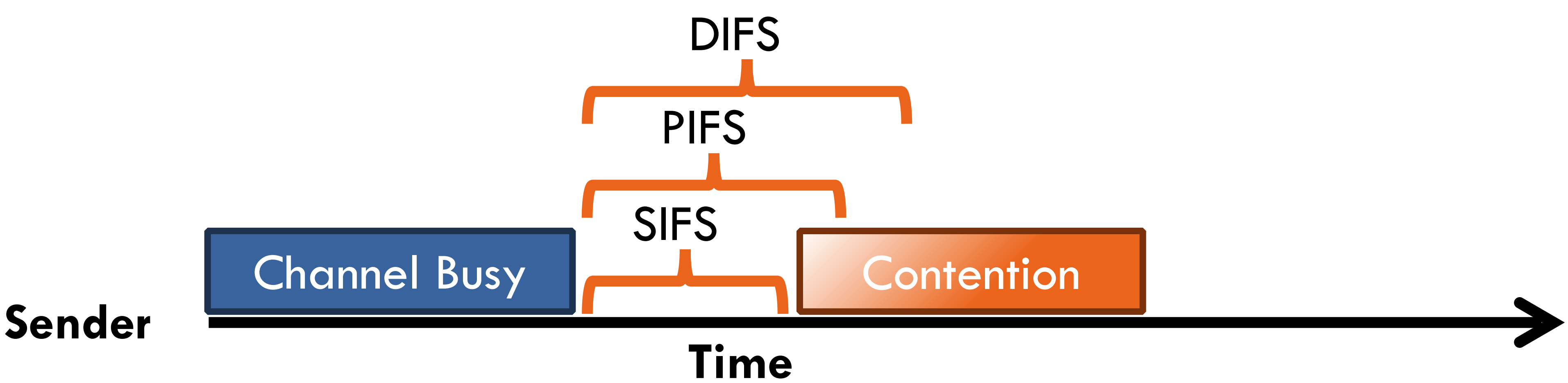
- MACA-style RTS/CTS is optional
- Distributed Coordination Function (DCF) based on…
  - Inter Frame Spacing (IFS)
    - DIFS – low priority, normal data packets
    - PIFS – medium priority, used with Point Coordination Function (PCF)
    - SIFS – high priority, control packets (RTS, CTS, ACK, etc.)
  - Contention interval: random wait time



# 802.11 Media Access



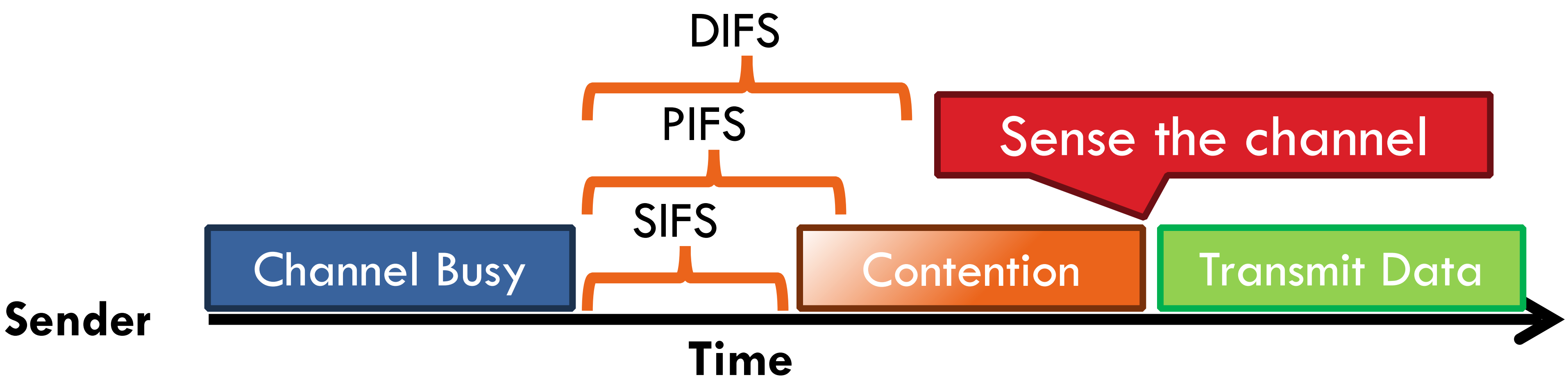
- MACA-style RTS/CTS is optional
- Distributed Coordination Function (DCF) based on…
  - Inter Frame Spacing (IFS)
    - DIFS – low priority, normal data packets
    - PIFS – medium priority, used with Point Coordination Function (PCF)
    - SIFS – high priority, control packets (RTS, CTS, ACK, etc.)
  - Contention interval: random wait time



# 802.11 Media Access

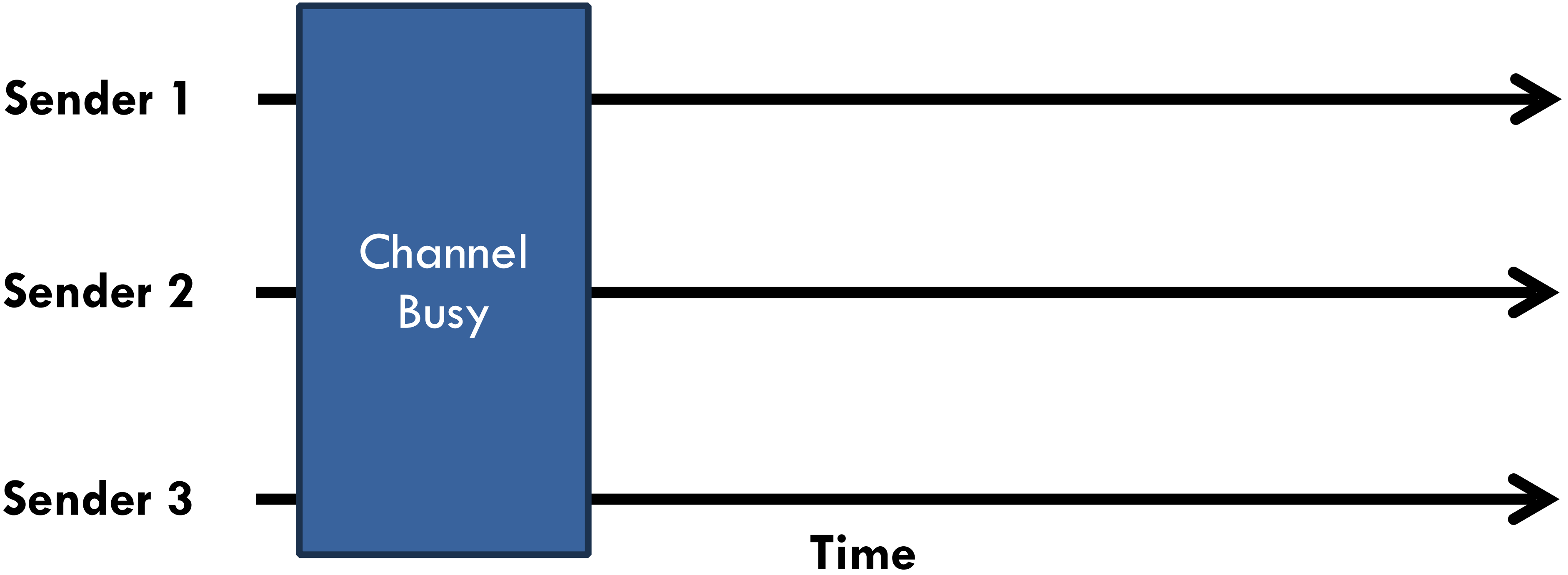


- MACA-style RTS/CTS is optional
- Distributed Coordination Function (DCF) based on…
  - Inter Frame Spacing (IFS)
    - DIFS – low priority, normal data packets
    - PIFS – medium priority, used with Point Coordination Function (PCF)
    - SIFS – high priority, control packets (RTS, CTS, ACK, etc.)
  - Contention interval: random wait time



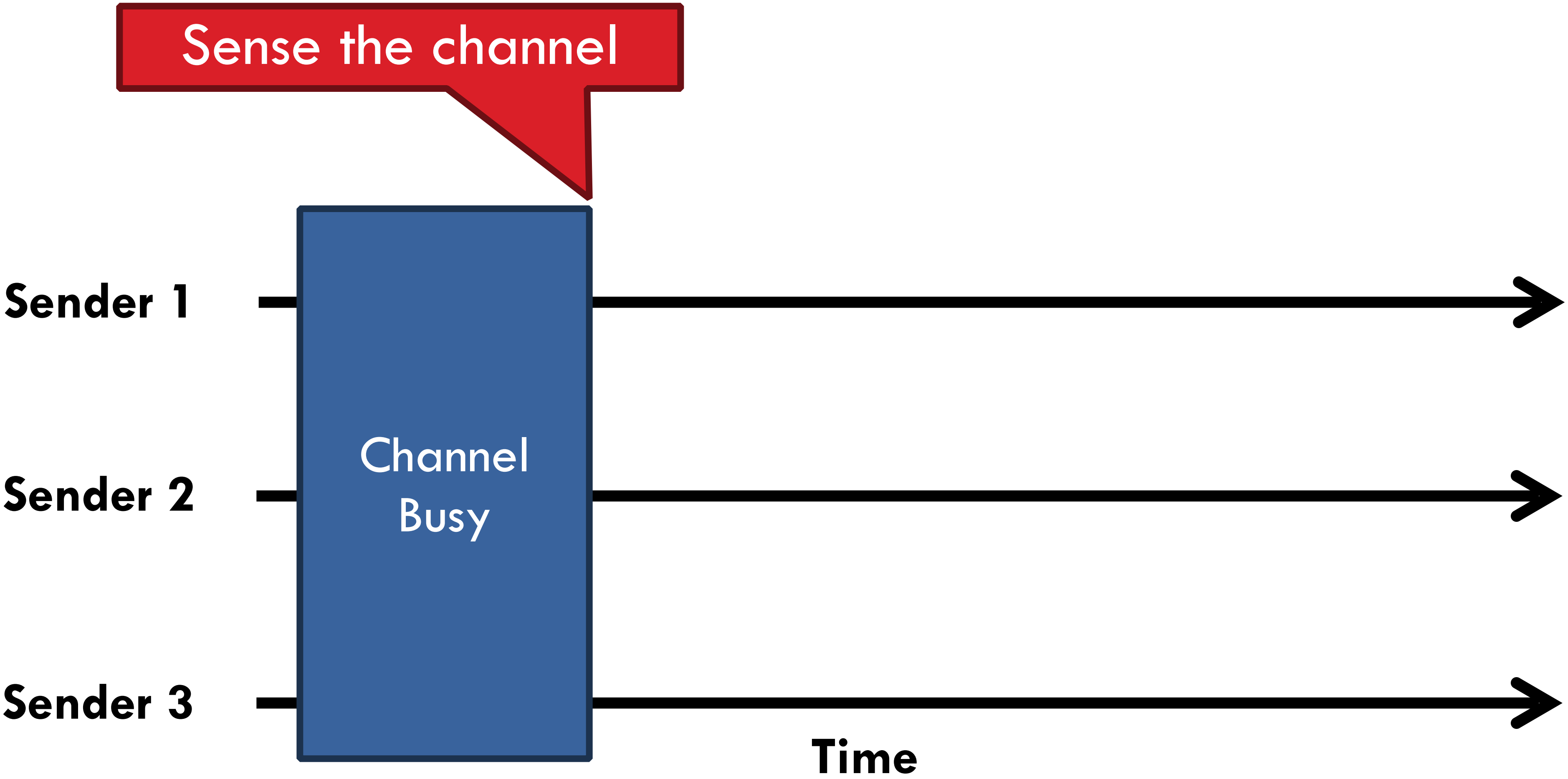
# 802.11 DCF Example





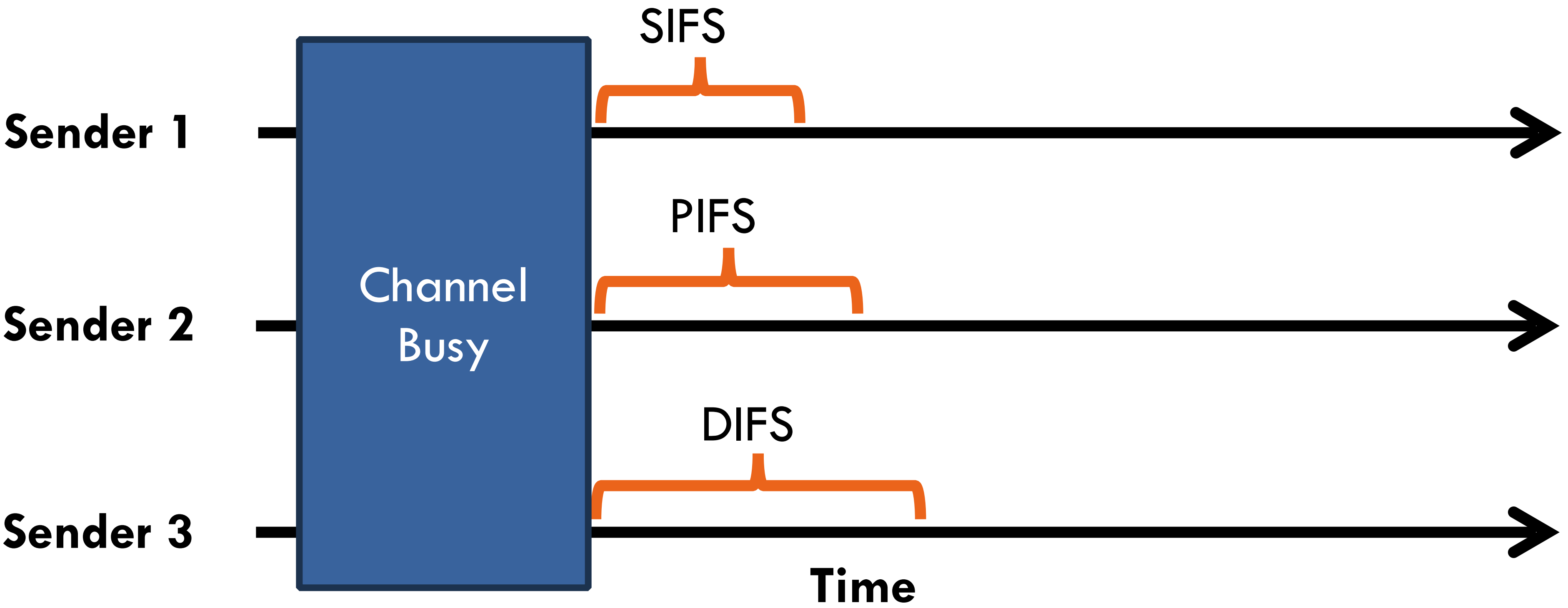
# 802.11 DCF Example





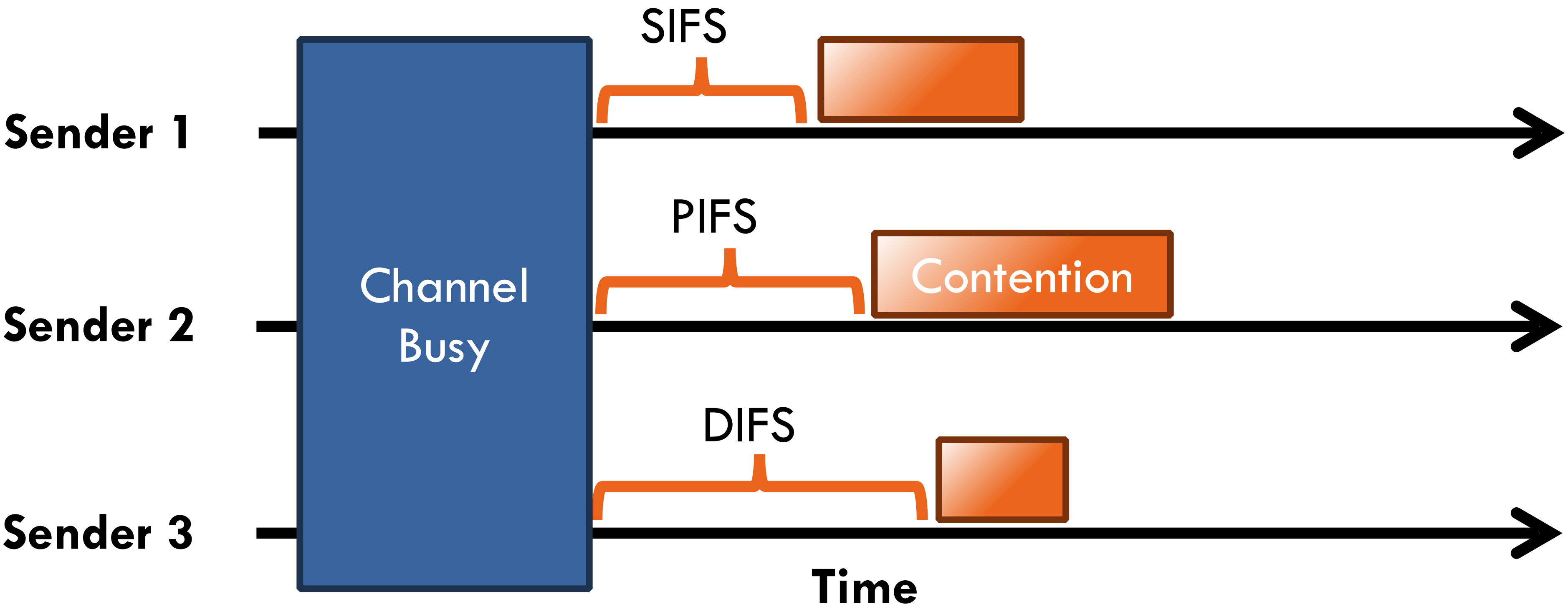
# 802.11 DCF Example





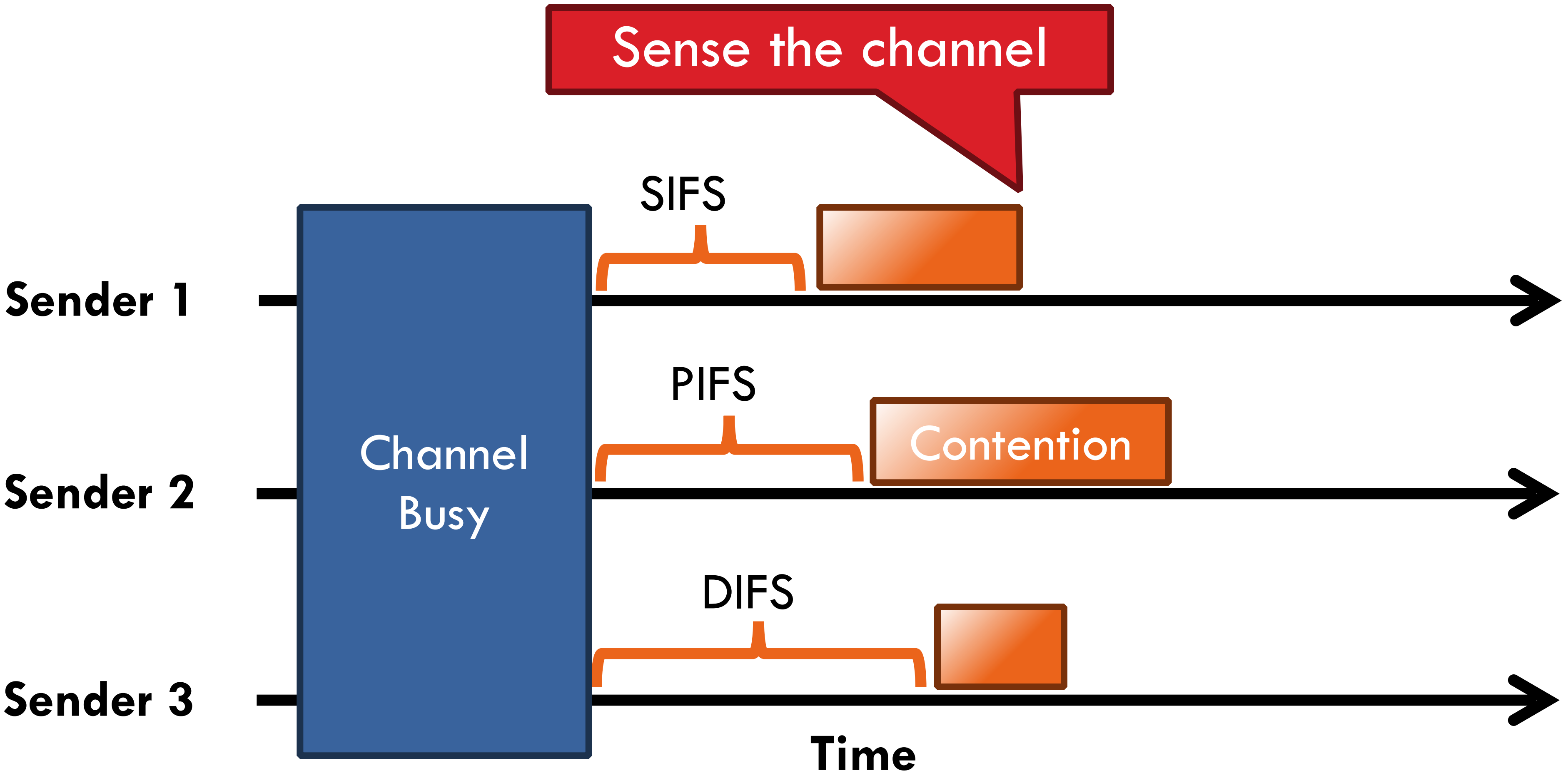
# 802.11 DCF Example





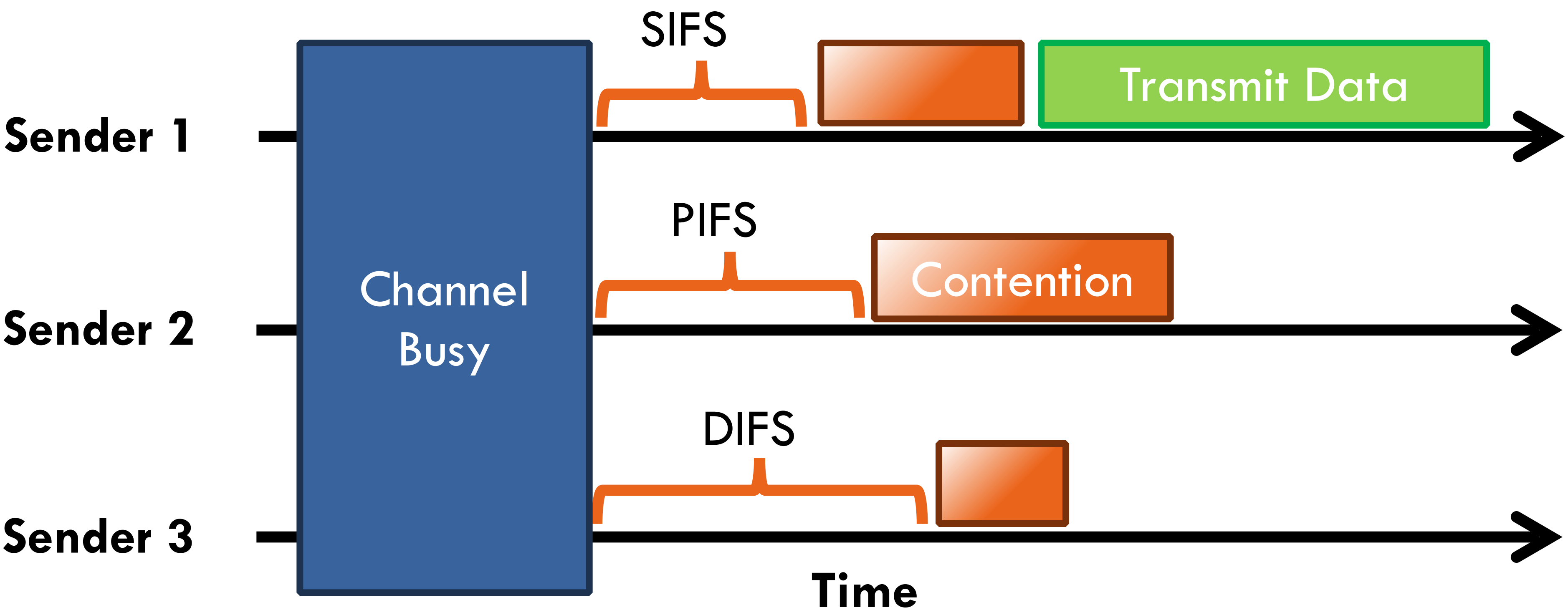
# 802.11 DCF Example





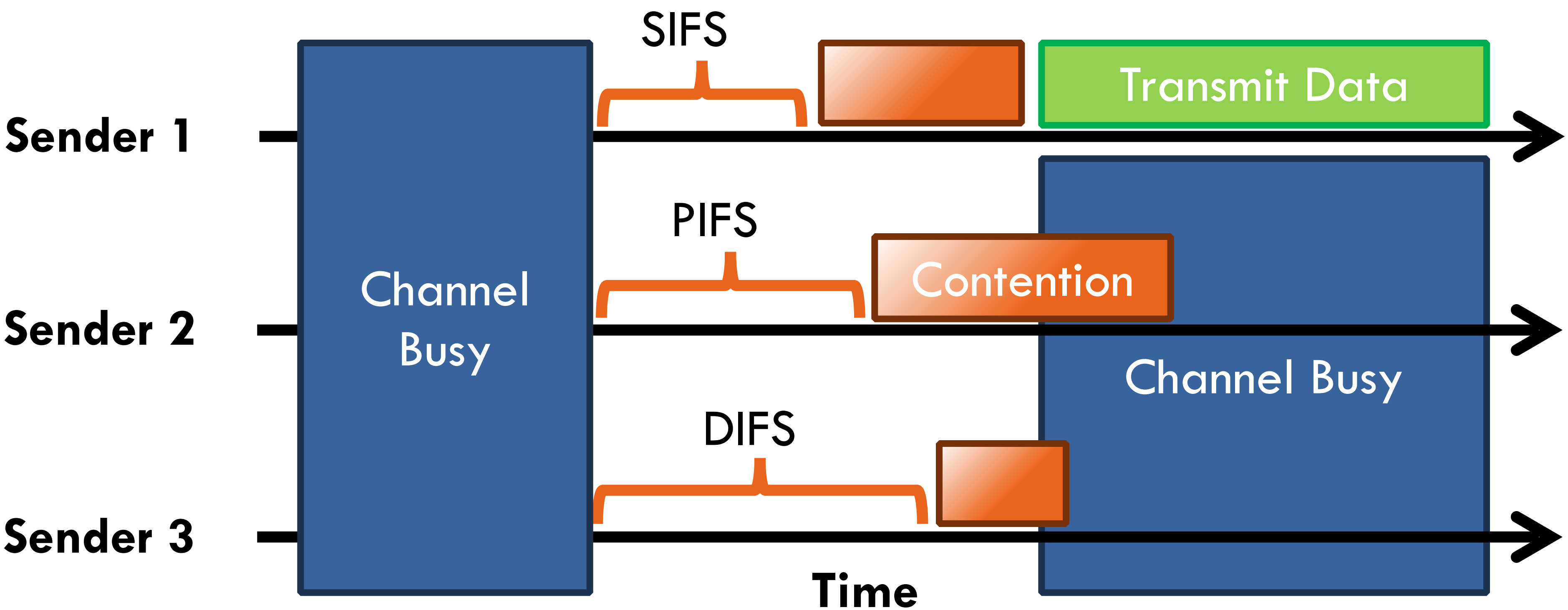
# 802.11 DCF Example





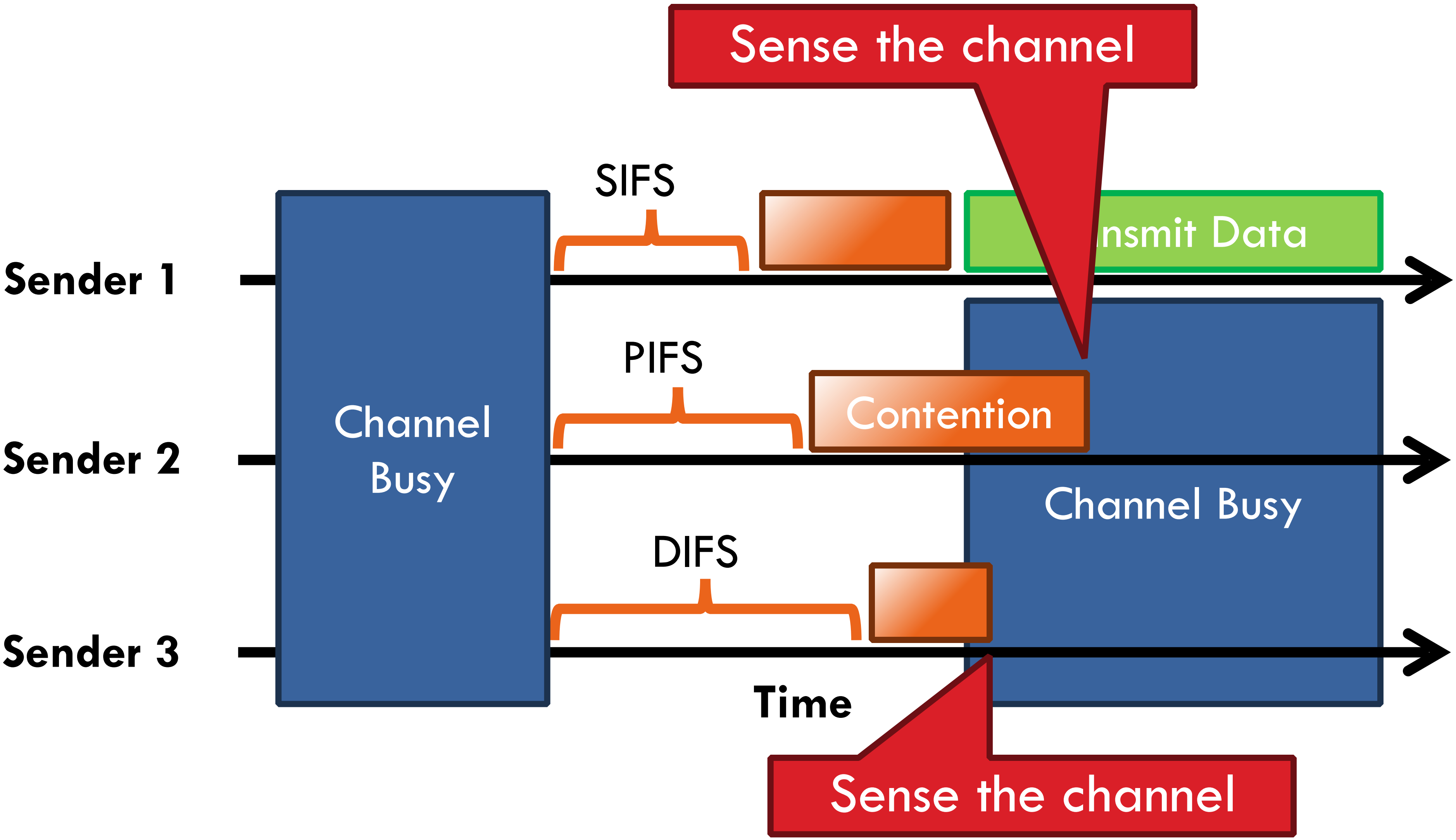
# 802.11 DCF Example





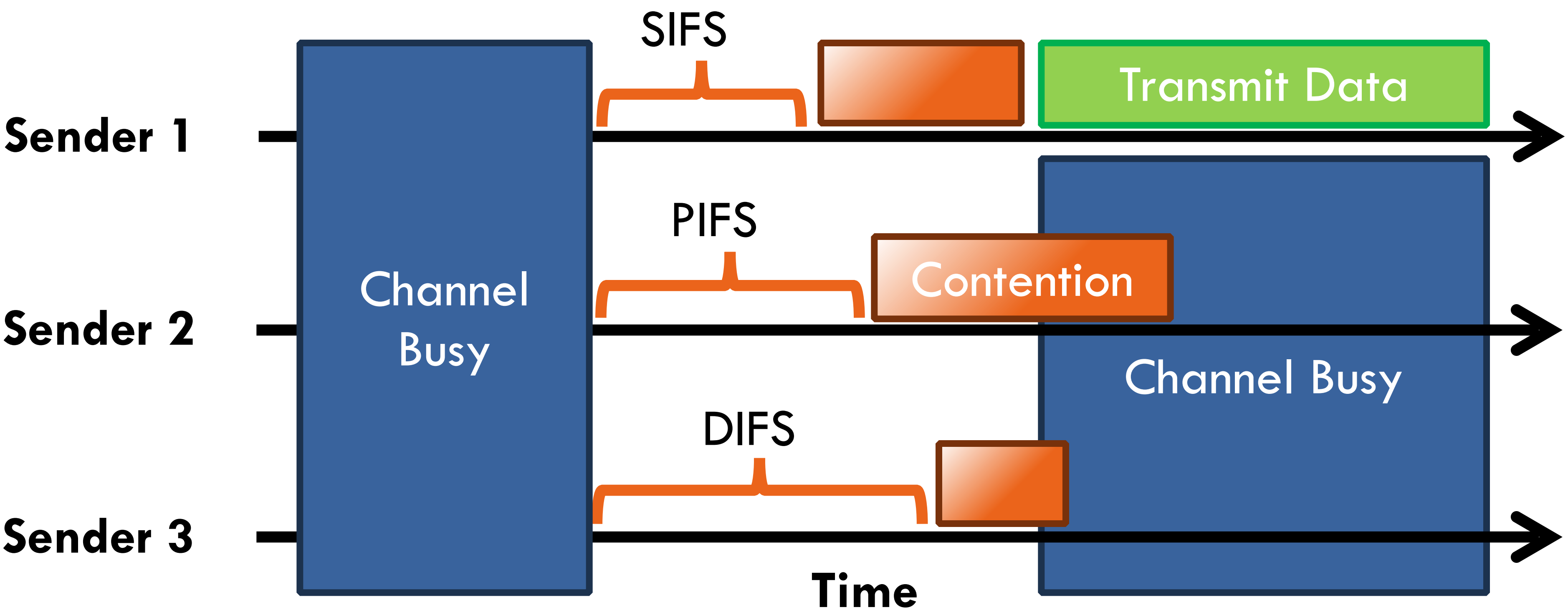
# 802.11 DCF Example





# 802.11 DCF Example





# 801.11 is Complicated



- We've only scratched the surface of 802.11
  - Association – how do clients connect to access points?
    - Scanning
    - What about roaming?
  - Variable sending rates to combat noisy channels
  - Infrastructure vs. ad-hoc vs. point-to-point
    - Mesh networks and mesh routing
  - Power saving optimizations
    - How do you sleep and also guarantee no lost messages?
  - Security and encryption (WEP, WAP, 802.11x)
- This is why there are courses on wireless networking