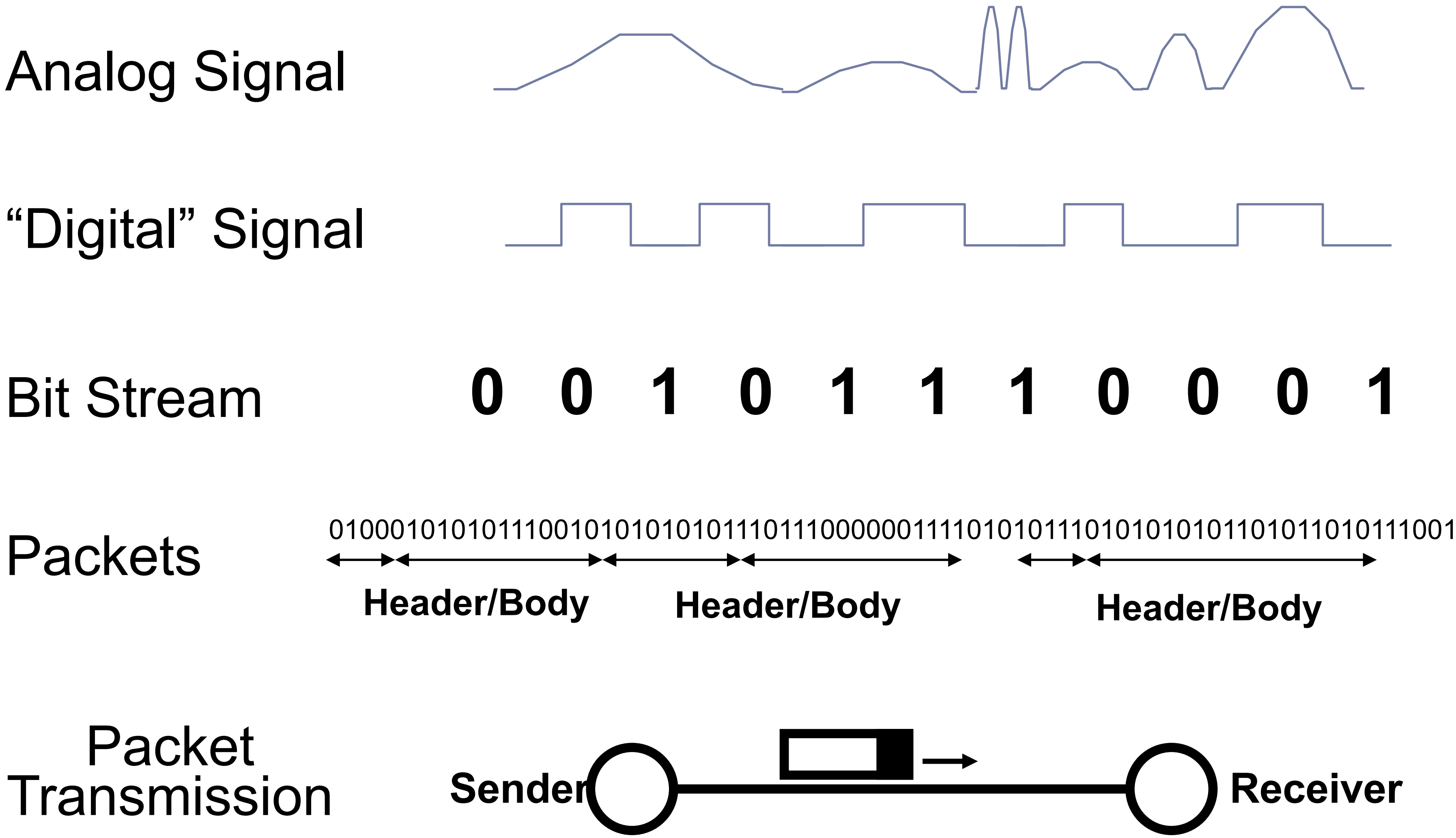
Cristina Nita-Rotaru

CS4700/5700: Network fundamentals

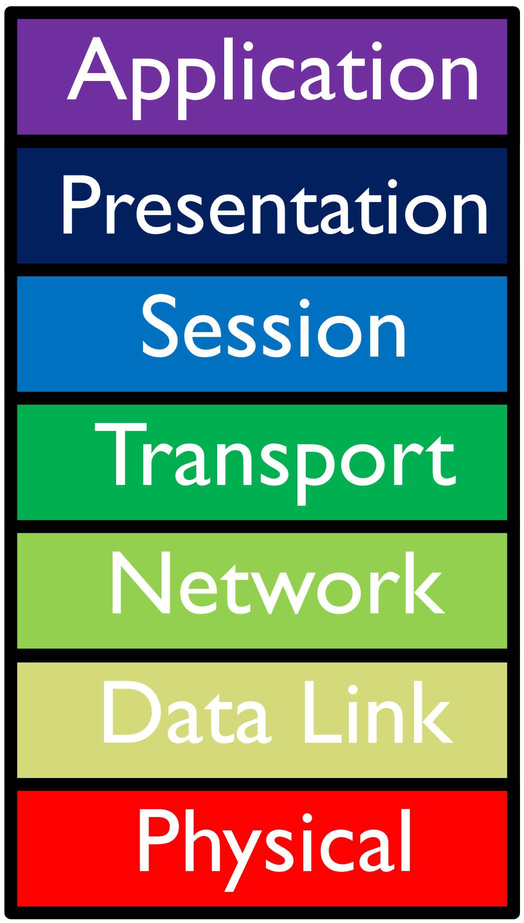
Data link layer.

From Signals to Packets



1: Framing

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?

Framing

- Physical layer determines how bits are encoded
- Next step, how to encode blocks of data
  - Packet switched networks
  - Each packet includes routing information
  - Data boundaries must be known so headers can be read
- Types of framing
  - Byte oriented protocols
  - Bit oriented protocols
  - Clock based protocols

Byte Oriented: Sentinel Approach

- Add **START** and **END** sentinels to the data
- Problem: what if **END** appears in the data?
  - Add a special **DLE** (Data Link Escape) character before **END**
  - What if **DLE** appears in the data? Add **DLE** before it.
  - Similar to escape sequences in C
    - printf("You must \"escape\" quotes in strings");
    - printf("You must \\escape\\ forward slashes as well");
- Used by Point-to-Point protocol, e.g. modem, DSL, cellular

Byte Oriented: Byte Counting

- Sender: insert length of the data in bytes at the beginning of each frame
- Receiver: extract the length and read that many bytes

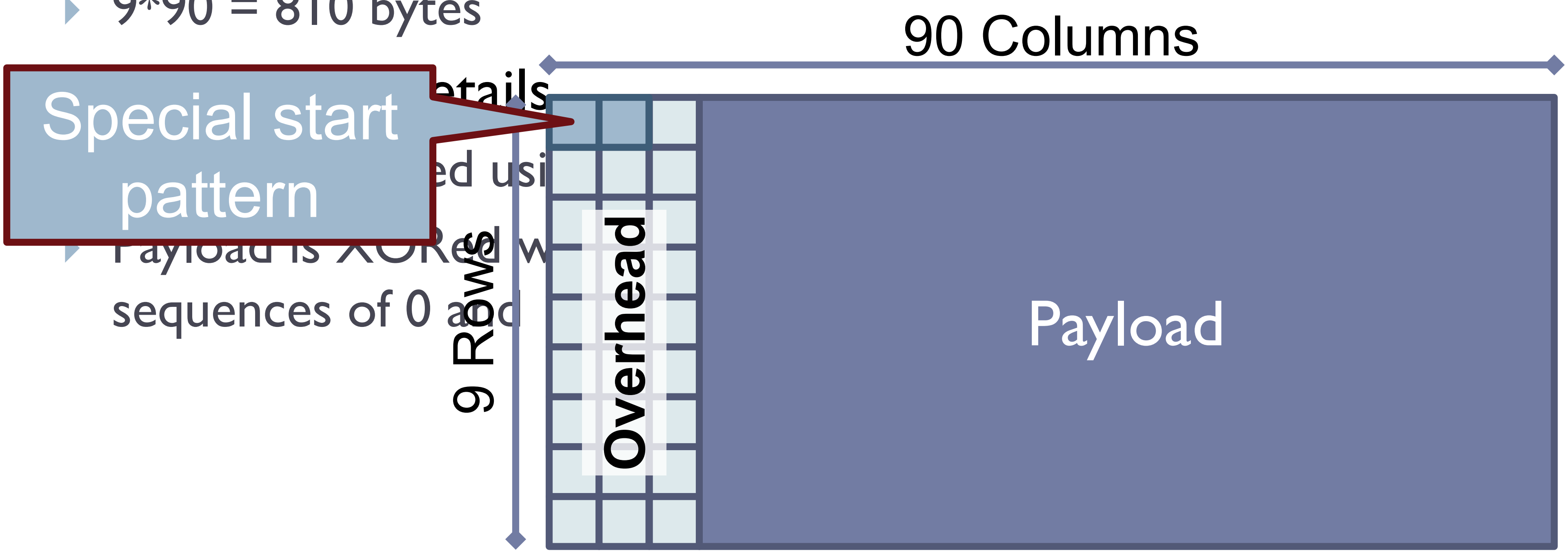
Bit Oriented: Bit Stuffing

01111110	Data	01111110

- Add sentinels to the start and end of data
  - Both sentinels are the same
  - Example: 01111110 in High-level Data Link Protocol (HDLC)
- Sender: insert a 0 after each 11111 in data
  - Known as "bit stuffing"
- Receiver: after seeing 11111 in the data…
  - 11111**0** → remove the 0 (it was stuffed)
  - 11111**1** → look at one more bit
    - 11111**10** → end of frame
    - 11111**11** → error! Discard the frame
- Disadvantage: 20% overhead at worst

Clock-based Framing: SONET

- **S**ynchronous **O**ptical **N**etwork
  - Transmission over very fast optical links
  - STS-*n*, e.g. STS-1: 51.84 Mbps, STS-768: 36.7 Gbps
- STS-1 frames based on fixed sized frames
  - 9*90 = 810 bytes
  - …etails
  - …ed usi…
  - Payload is XORed w…
    sequences of 0 and…



2: Error checking

Dealing with Noise

- The physical world is inherently noisy
  - Interference from electrical cables
  - Cross-talk from radio transmissions, microwave ovens
  - Solar storms
- How to detect bit-errors in transmissions?
- How to recover from errors?

Naïve Error Detection

- Idea: send two copies of each frame
  - if (memcmp(frame1, frame2) != 0) { OH NOES, AN ERROR! }
- Why is this a bad idea?
  - Extremely high overhead
  - Poor protection against errors
    - Twice the data means twice the chance for bit errors

Parity Bits

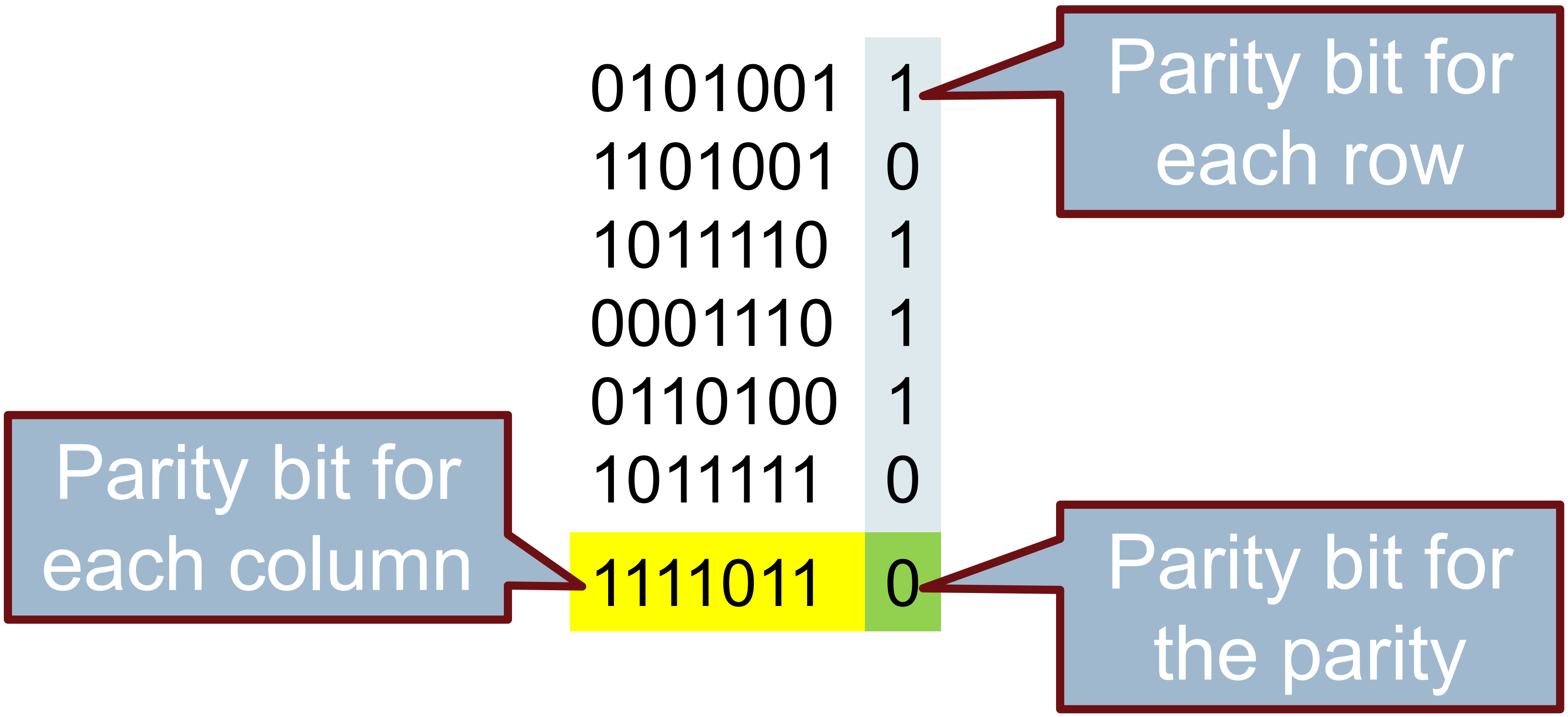
- Idea: add extra bits to keep the number of 1s even
  - Example: 7-bit ASCII characters + 1 parity bit

0101001 1 1101001 0 1011110 1 0001110 1 0110100 1

10

- Detects 1-bit errors and some 2-bit errors
- Not reliable against bursty errors

Two Dimensional Parity



- Can detect all 1-, 2-, and 3-bit errors, some 4-bit errors
- 14% overhead

Two Dimensional Parity Examples

0101001	1
1111000	0
1011110	1
0011111	1
0110100	1
1011111	0
1111011	0

Odd number of 1s

Odd number of 1s

Odd Number of 1s

Odd number of 1s

Checksums

- Idea:
  - Add up the bytes in the data
  - Include the sum in the frame

- Use ones-complement arithmetic
- Lower overhead than parity: 16 bits per frame
- But, not resilient to errors
  - Why?
- Used in UDP, TCP, and IP

1 101001 + 0 101001= 10010010

Cyclic Redundancy Check (CRC)

- Uses field theory to compute a semi-unique value for a given message
- Much better performance than previous approaches
  - Fixed size overhead per frame (usually 32-bits)
  - Quick to implement in hardware
  - Only 1 in 2^{32} chance of missing an error with 32-bit CRC
- Details are in the book/on Wikipedia

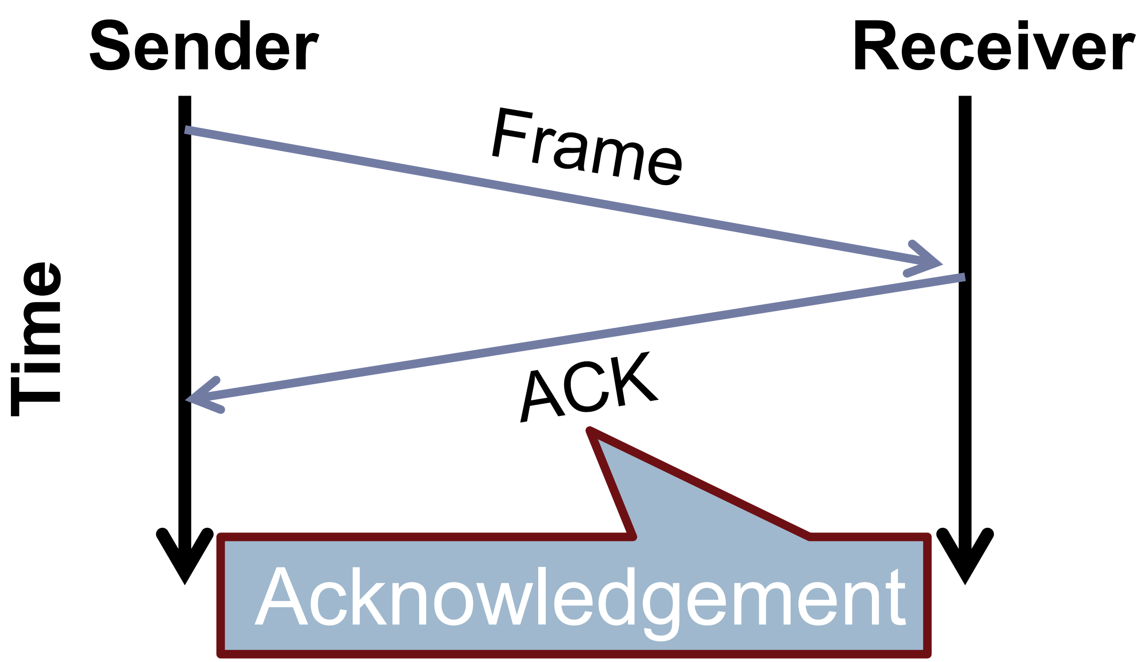
Should We Error Check in the Data Link?

- Recall the End-to-End Argument
- Cons:
  - Error free transmission cannot be guaranteed
  - Not all applications want this functionality
  - Error checking adds CPU and packet size overhead
  - Error recovery requires buffering
- Pros:
  - Potentially better performance than app-level error checking
- Data link error checking in practice
  - Most useful over lossy links
  - Wifi, cellular, satellite

3: Reliability

What About Reliability?

- How does a sender know that a frame was received?
  - What if it has errors?
  - What if it never arrives at all?

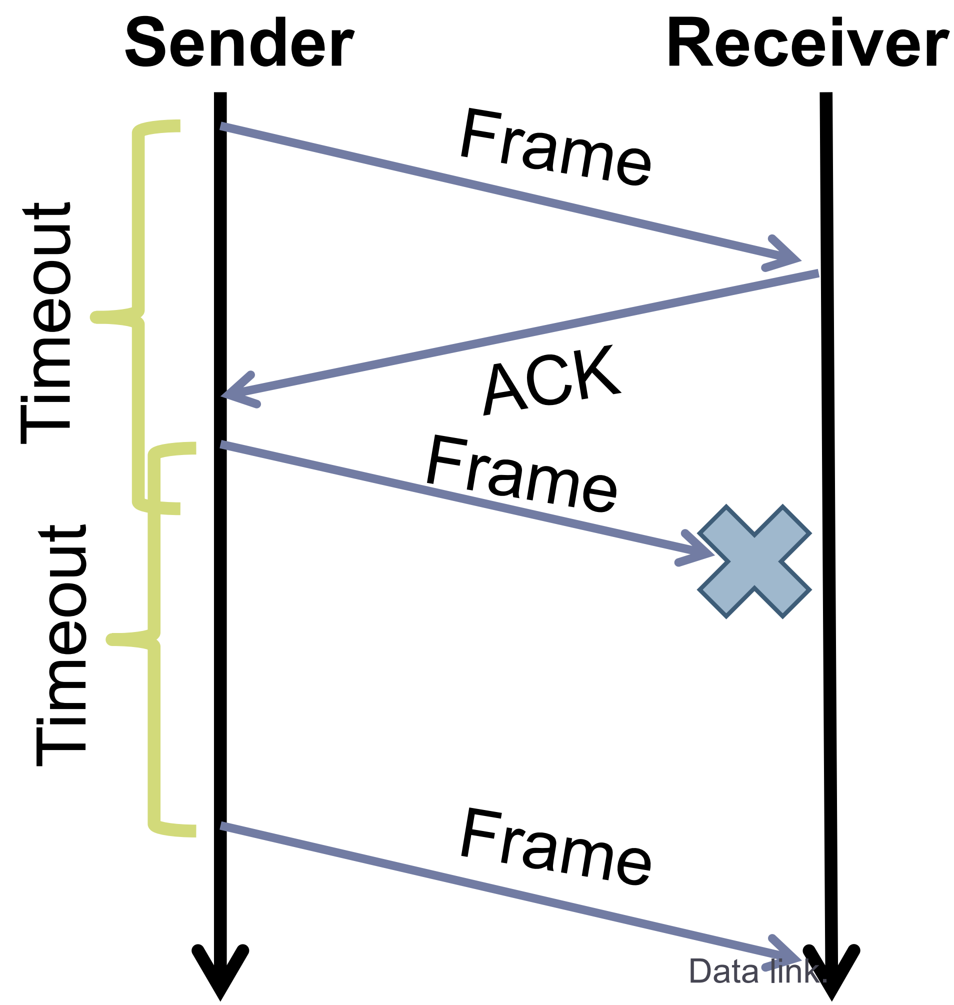


Stop and Wait

- Simplest form of reliability
- Example: Bluetooth
- Problems?
  - Utilization
  - Can only have one frame in flight at any time
- 10Gbps link and 10ms delay
  - Need 100 Mbit to fill the pipe
  - Assume packets are 1500B

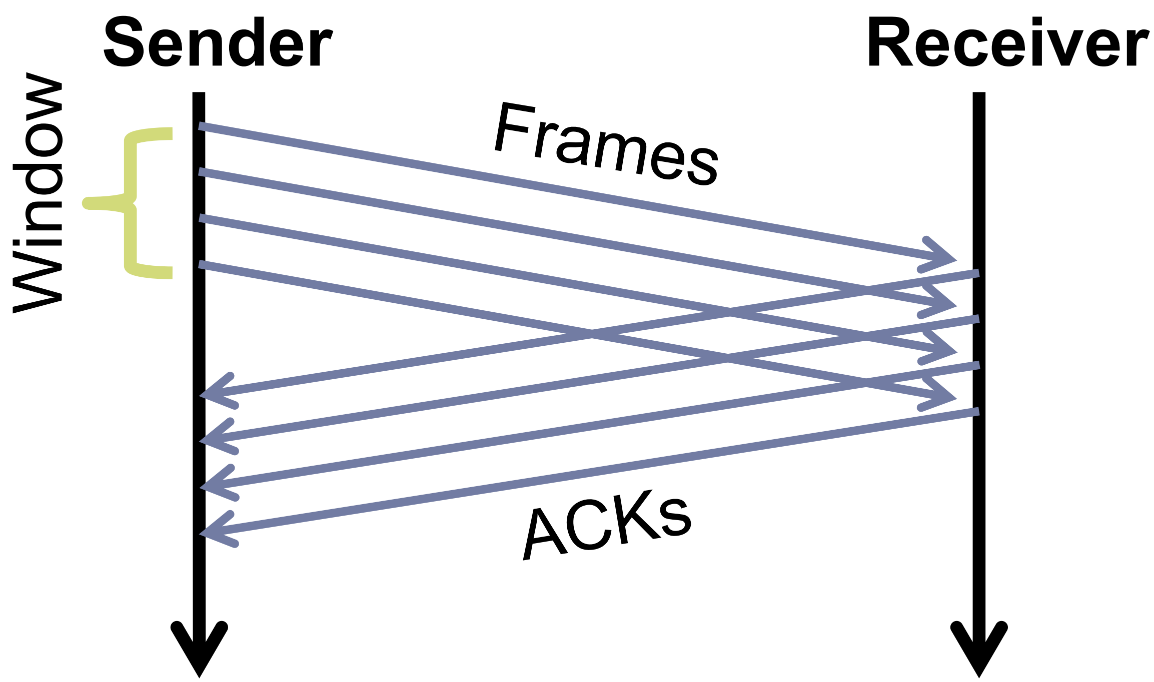
1500B*8bit/(2*10ms) = 600Kbps

Utilization is 0.006%



Sliding Window

- Allow multiple outstanding, un-ACKed frames
- Number of un-ACKed frames is called the window



- Made famous by TCP
  - We'll look at this in more detail later

2: Media access

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
- 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
- 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
- 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

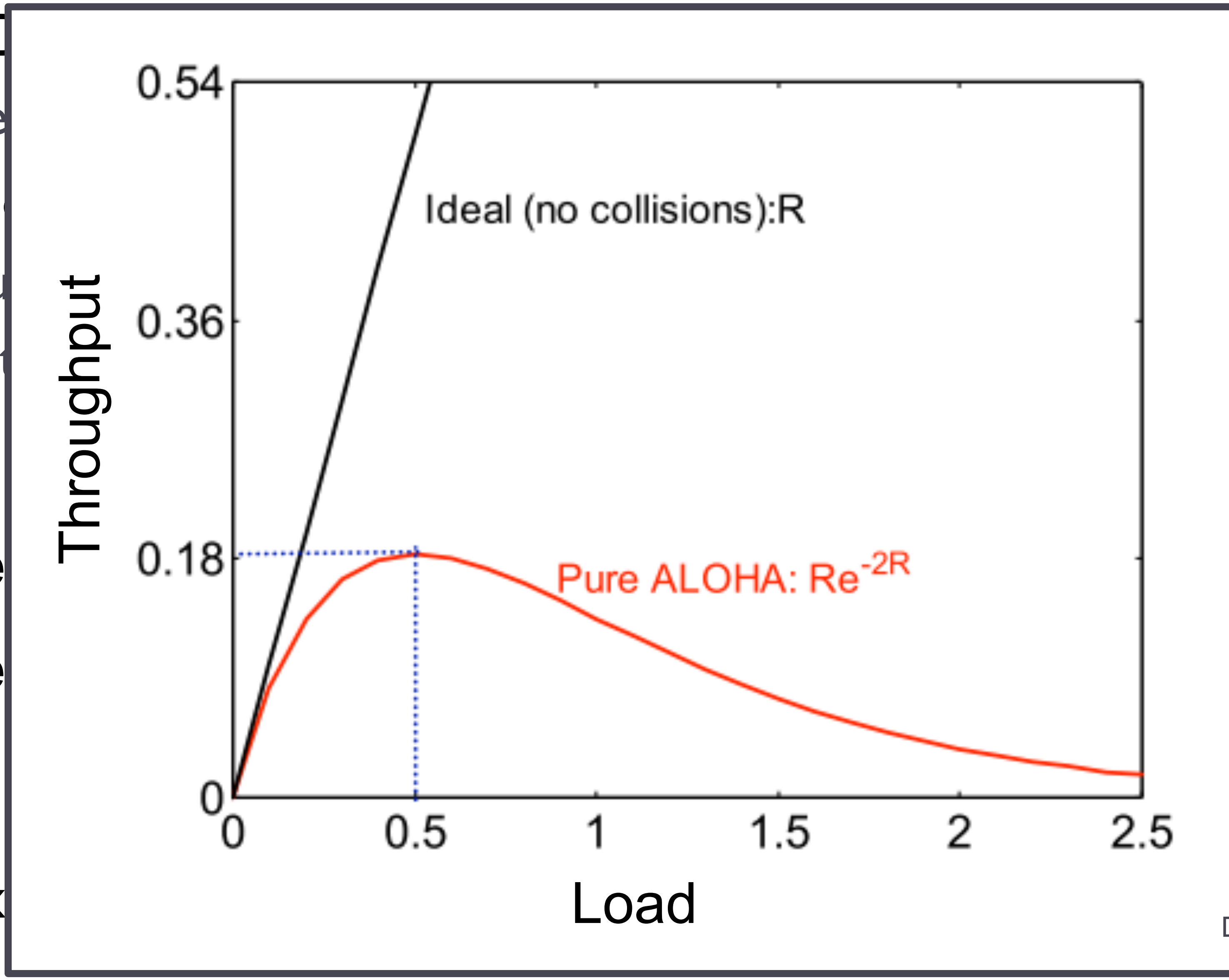
ALOHA

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

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

Tradeoffs vs. TDMA

- In TD
  - De
- In Al
  - Mu
  - But



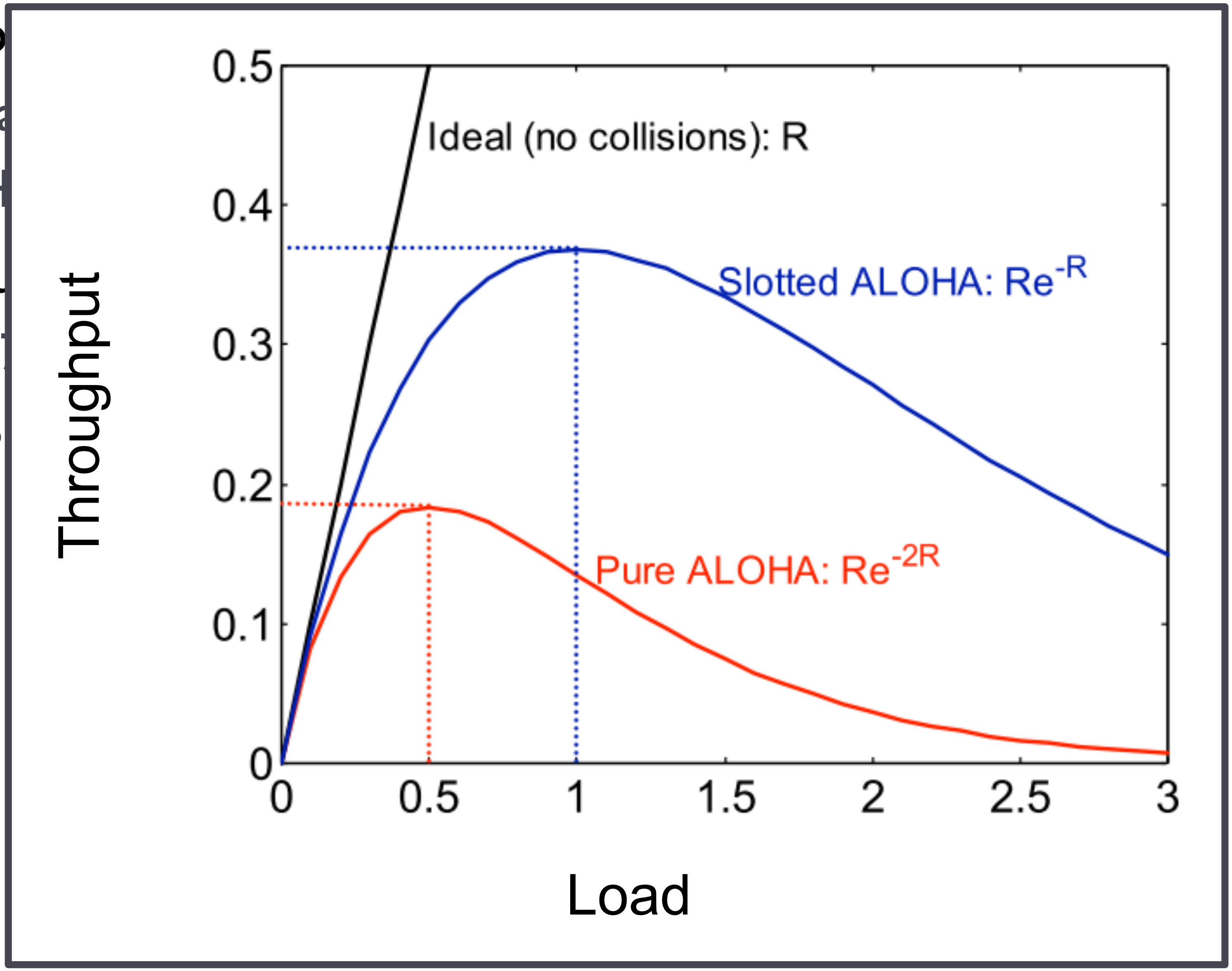
Sende

Sende

- Max

Slotted ALOHA

- Pro
  - Sa
  - H
- Thu
  - 37
  - B



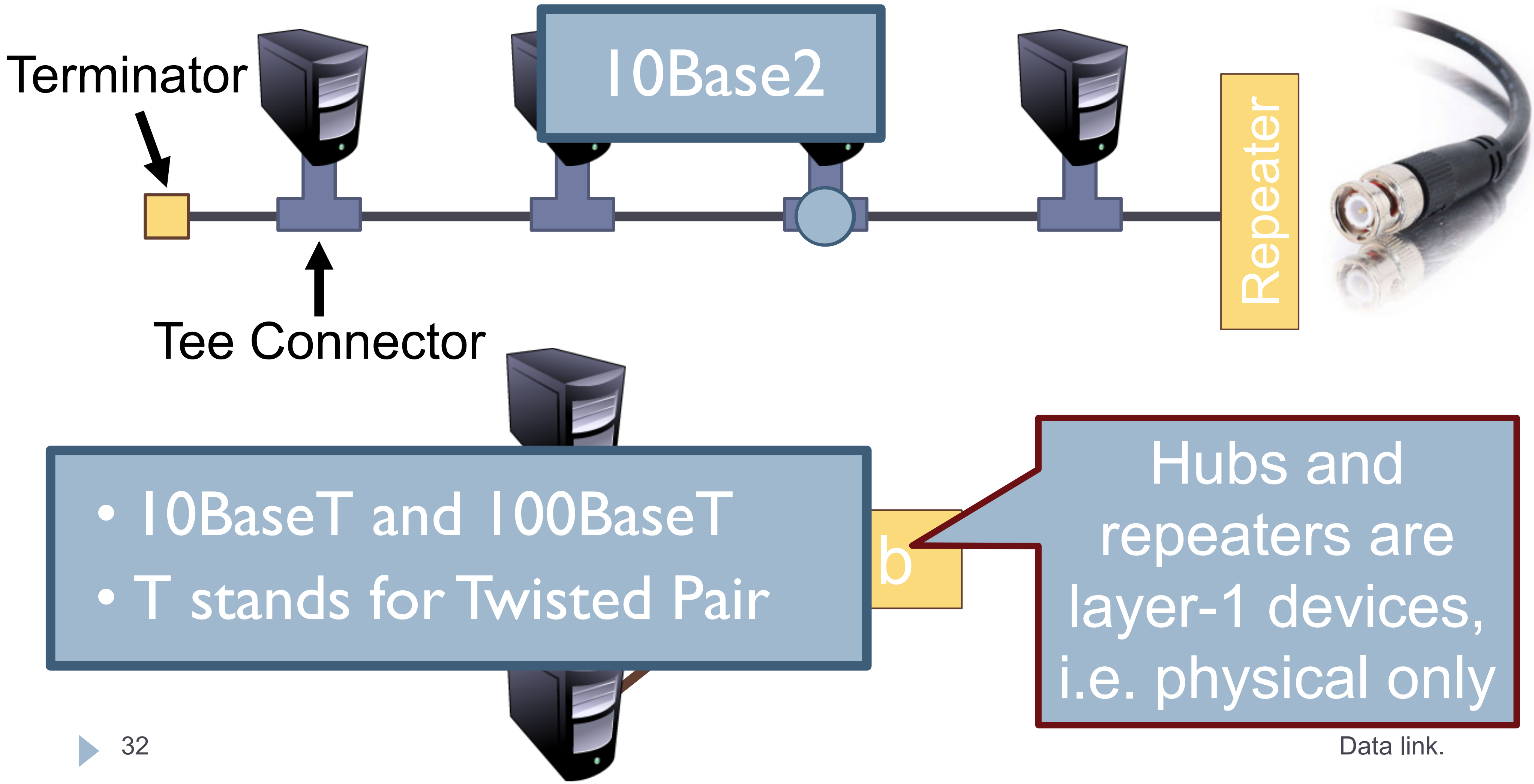
802.3 Ethernet

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.
- 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

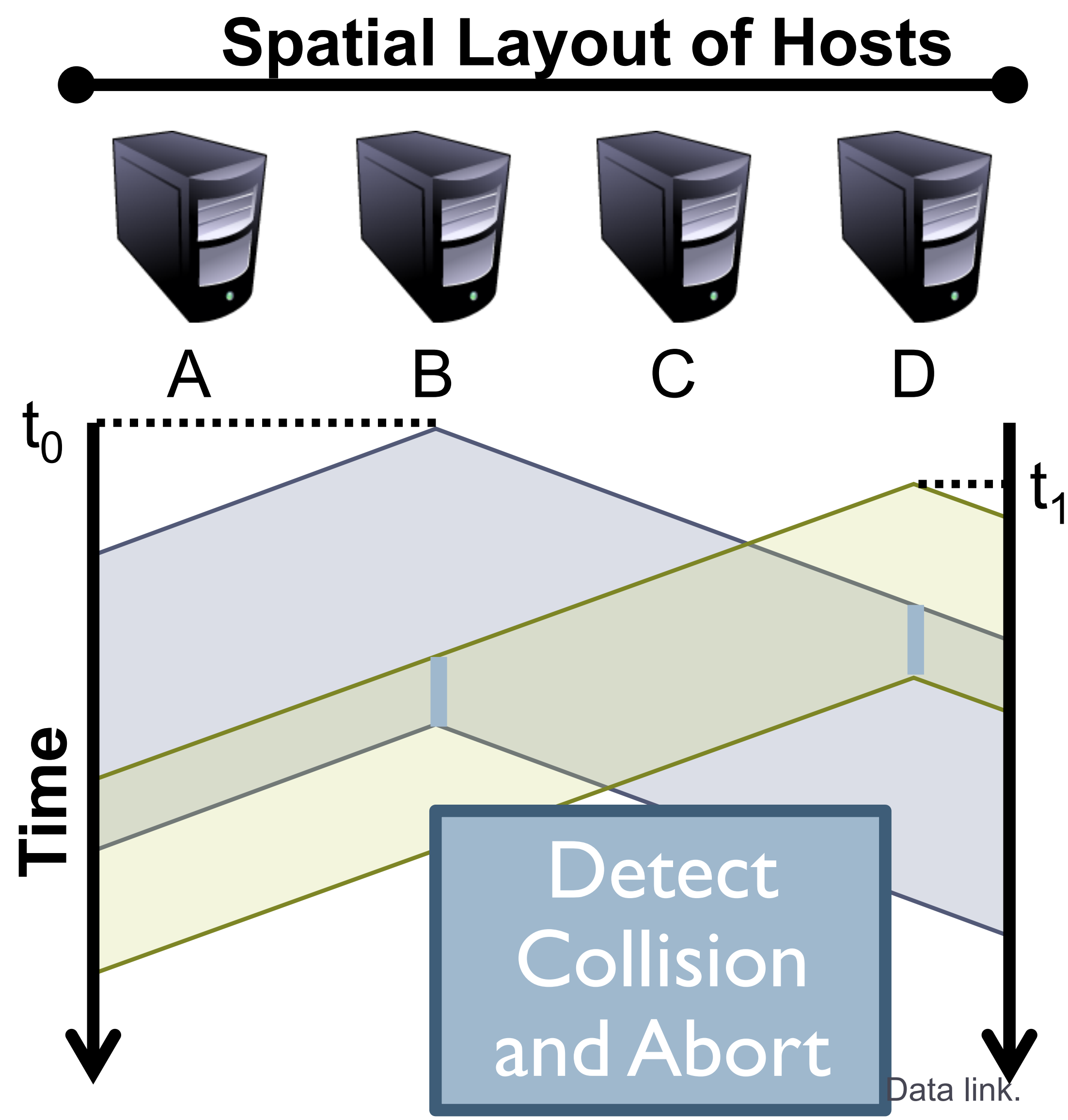


CSMA/CD

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



Exponential Backoff

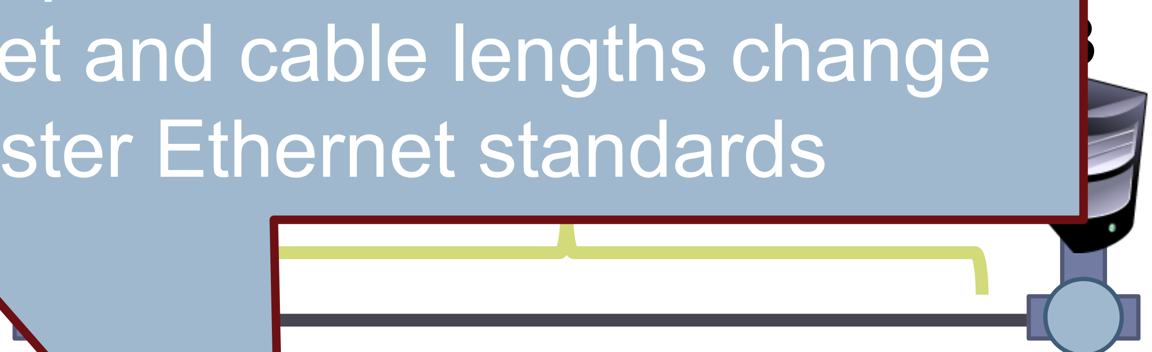
- When a sender detects a collision, send "jam signal"
  - Make sure all hosts are aware of collision
  - Jam signal is 32 bits long (plus header overhead)
- Exponential backoff operates in multiples of 512 bits
  - Select $k \in [0, 2^n - 1]$, where n = number of collisions
  - Wait $k * 51.2\mu s$ 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

- 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 s transmitting
2. Time *t* + *d*: Host starts transmitting
3. Time *t* + 2**d*: collision detected

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

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

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
- 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

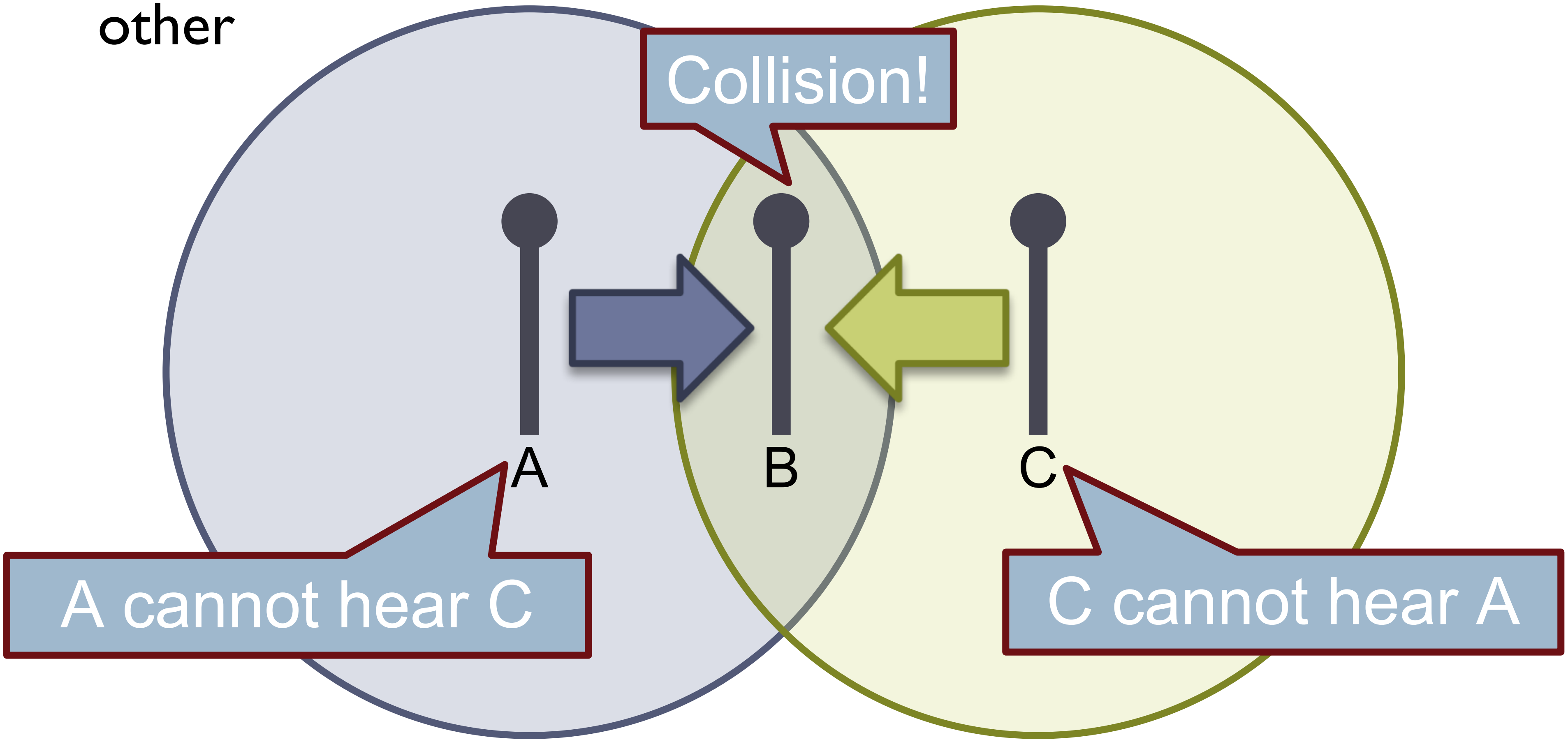
- Today's Ethernet is switched
  - More on this later
- 1Gbit and 10Gbit Ethernet now common
  - 100Gbit on the way
  - Uses same old packet header
  - Full duplex (send and receive at the same time)
  - Auto negotiating (backwards compatibility)
  - Can also carry power

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

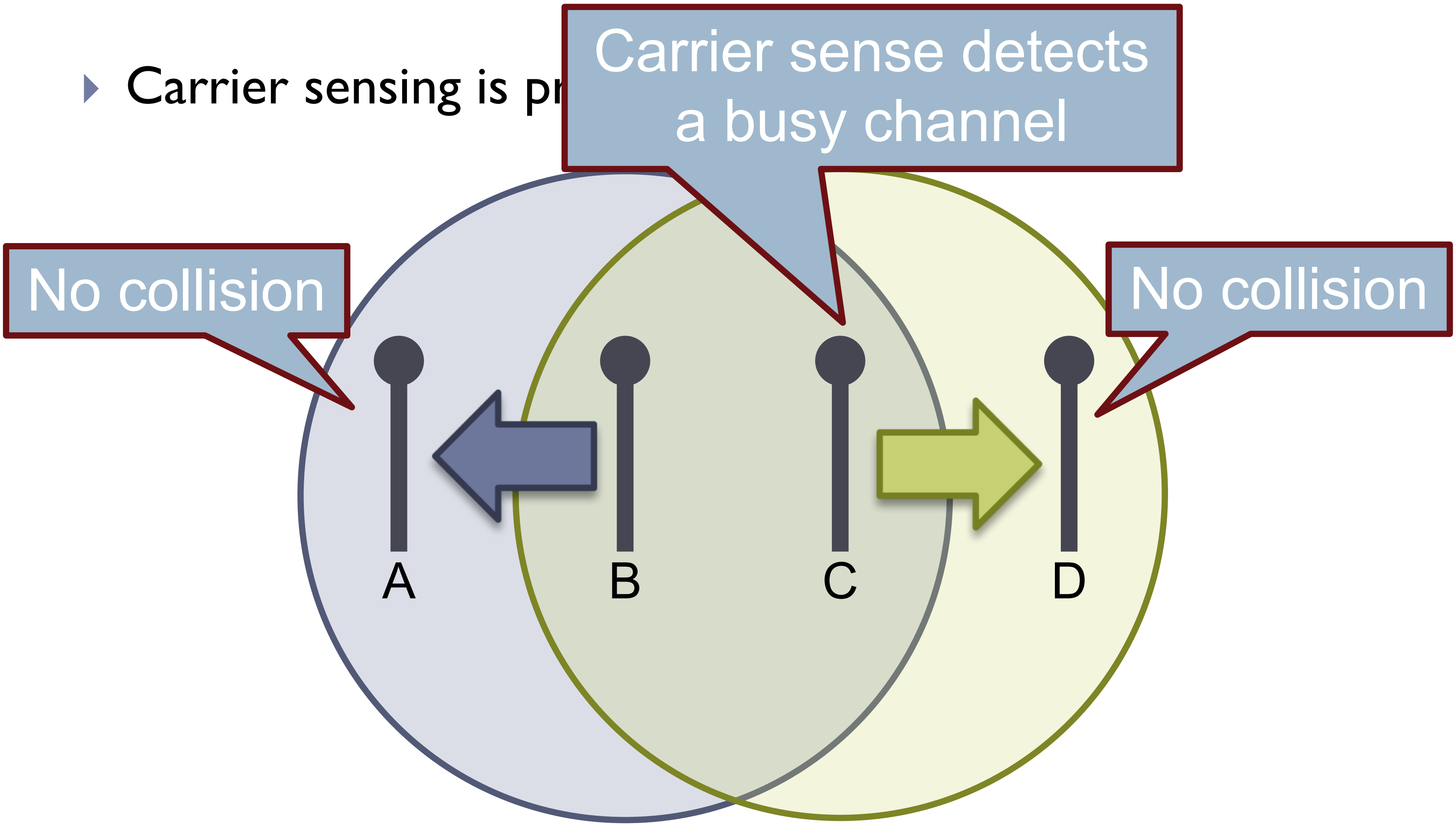
- Radios on the same network cannot always hear each other



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

Exposed Terminal Problem

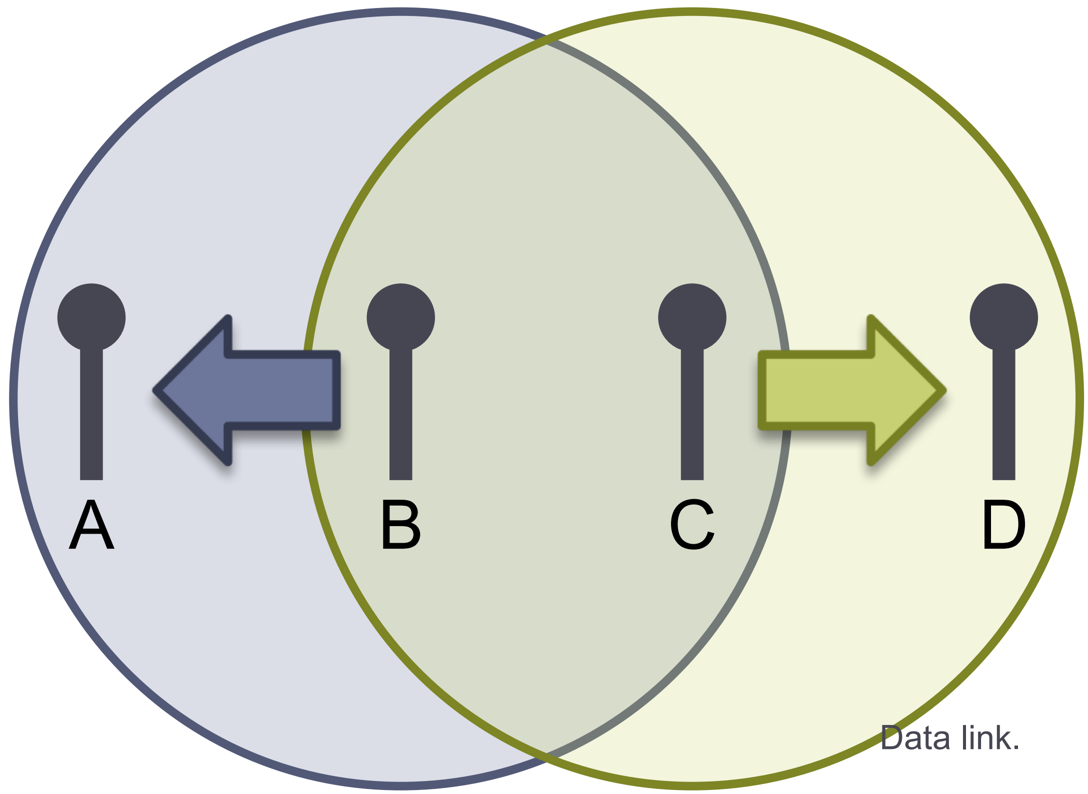
- Carrier sensing is pr



- Carrier sense can erroneously reduce utilization

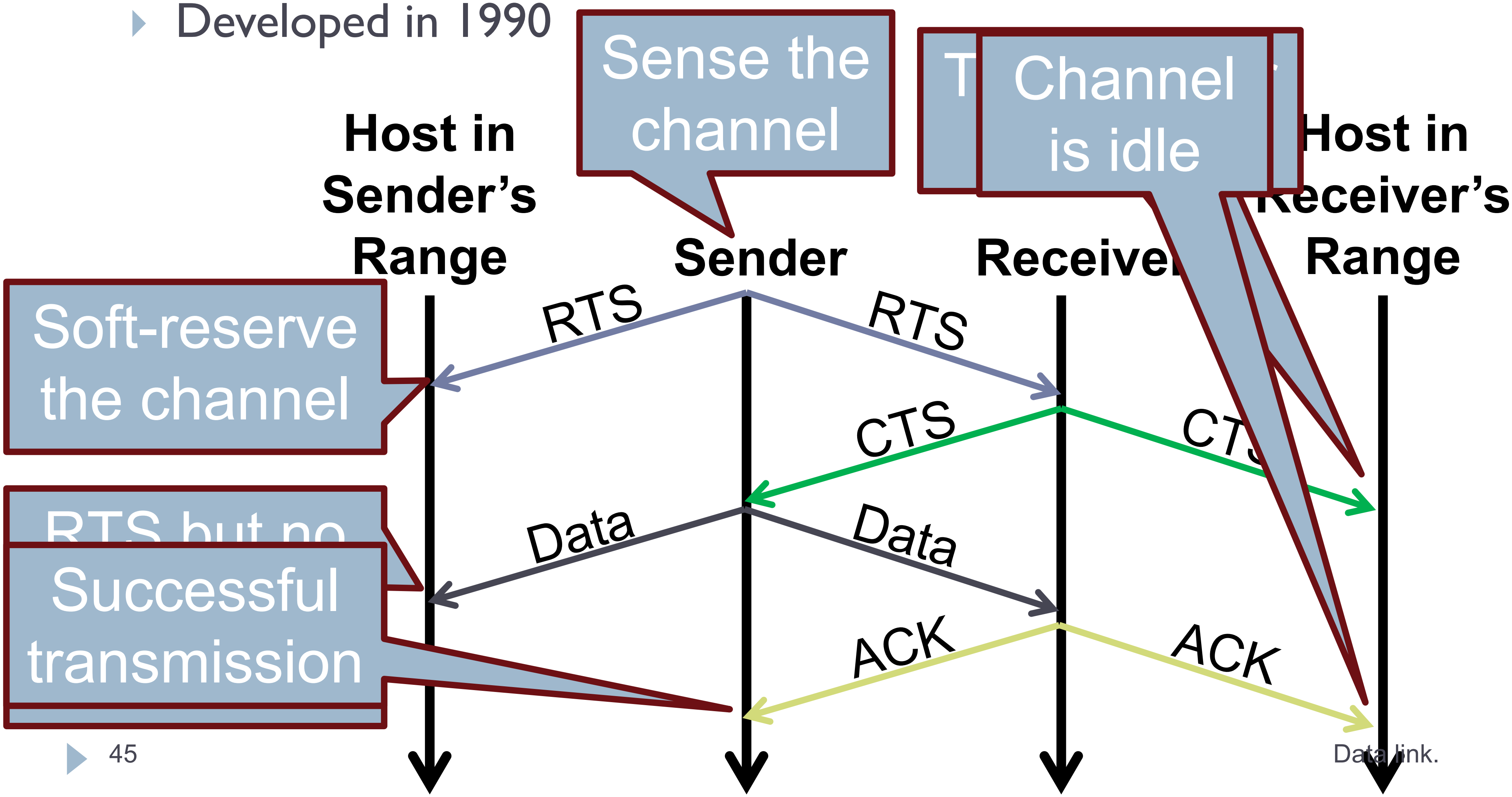
Reachability in Wireless

- 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

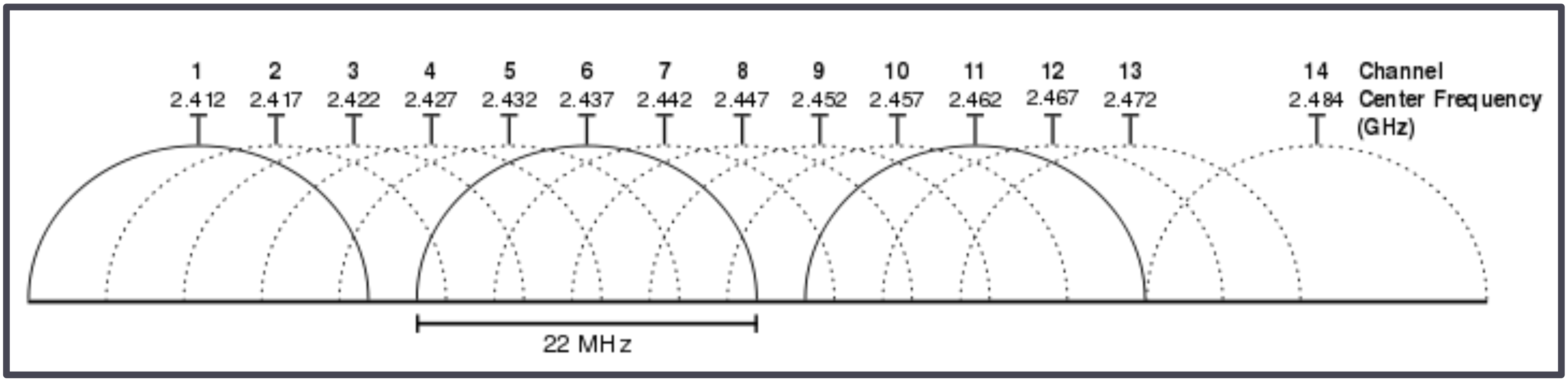


Collisions in MACA

- 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



802.11a/g

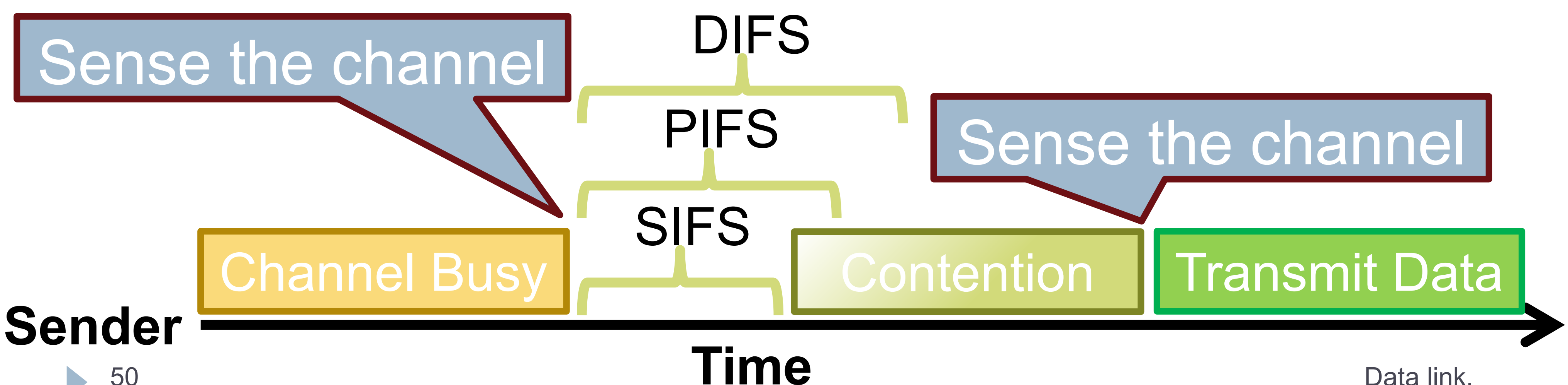
- 802.11a
  - Uses the 5 Ghz band
  - 6, 9, 12, 18, 24, 36, 48, 54 Mbps
  - Switches from CCK to Orthogonal Frequency Division Multiplexing (OFDM)
    - Each frequency is orthogonal
- 802.11g
  - Introduced in 2003
  - Uses OFDM to improve performance (54 Mbps)
  - Backwards compatible with 802.11b
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

802.11n/ac

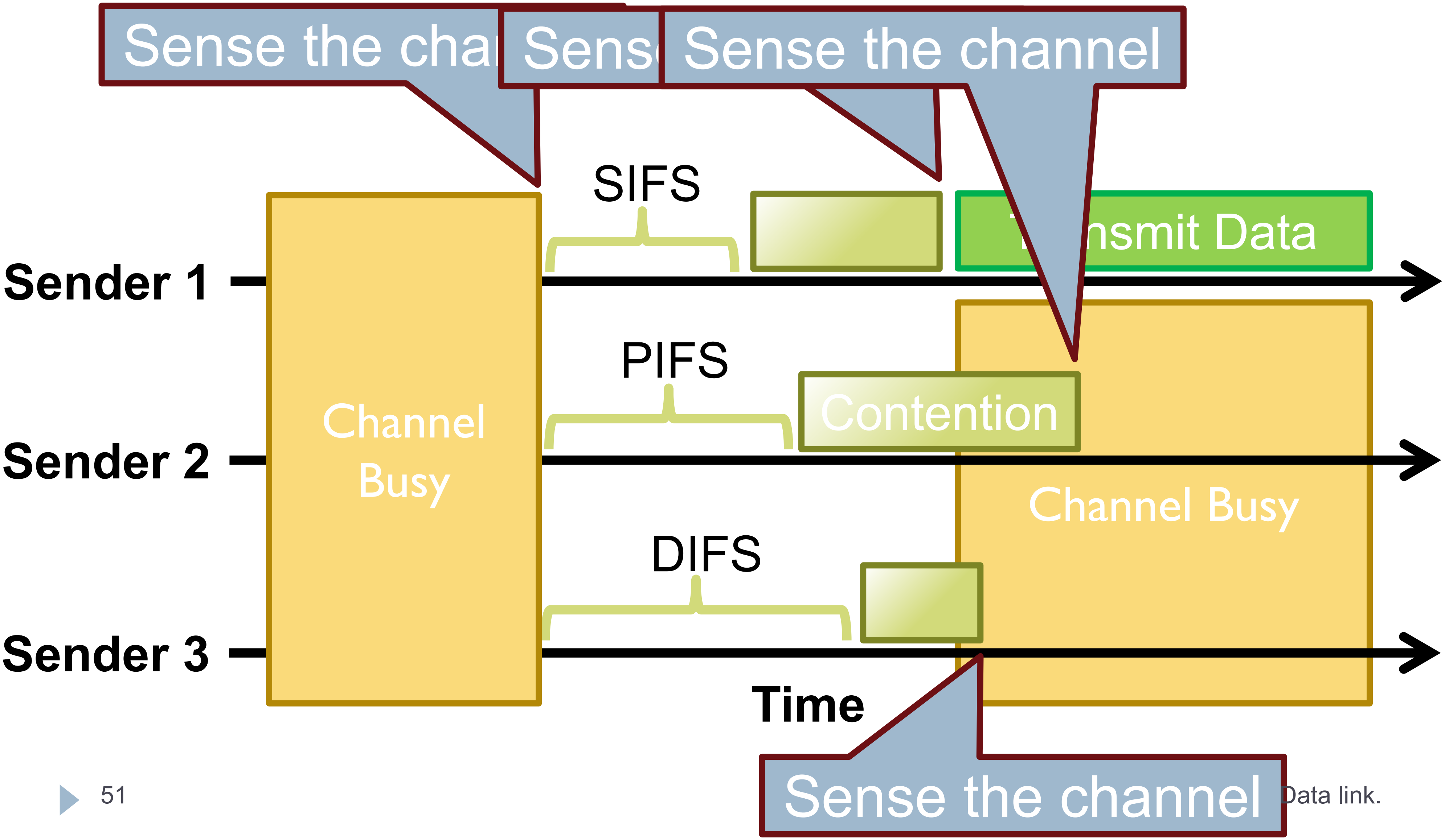
- 802.11n
  - Introduced in 2009
  - Multiple Input Multiple Output (MIMO)
    - Multiple send and receive antennas per devices (up to four)
    - Data stream is multiplexed across all antennas
  - Maximum 600 Mbps transfer rate (in a 4x4 configuration)
  - 300 Mbps is more common (2x2 configuration)
- 802.11ac (January 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 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