CSC4200/5200 – COMPUTER NETWORKING

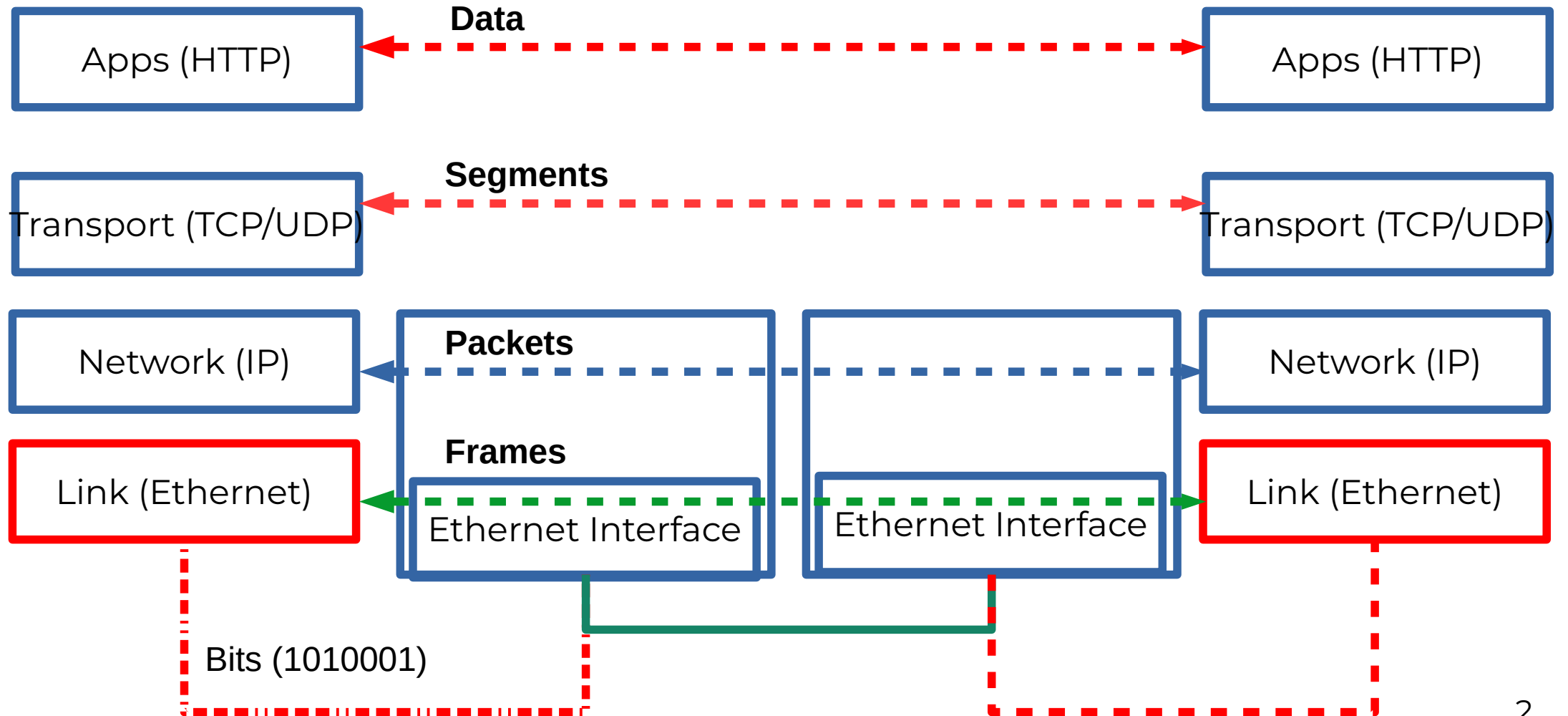## PHYSICAL AND LINK LAYER RECAP

Instructor: Susmit Shannigrahi
sshannigrahi@tntech.edu

Tennessee
TECH

Recap – All this for a cat picture!!

| Apps (HTTP) | ←---- **Data** ----→ | Apps (HTTP) |

| Transport (TCP/UDP) | ←---- **Segments** ----→ | Transport (TCP/UDP) |

| Network (IP) | ←---- **Packets** | | ----→ | Network (IP) |

Frames

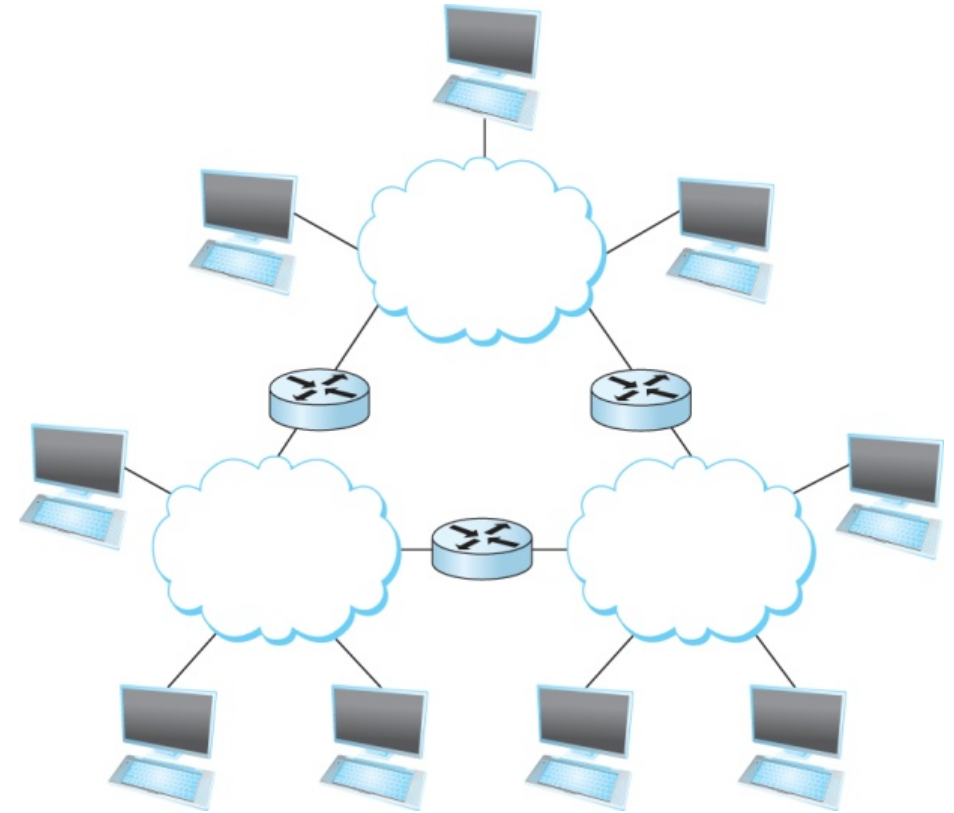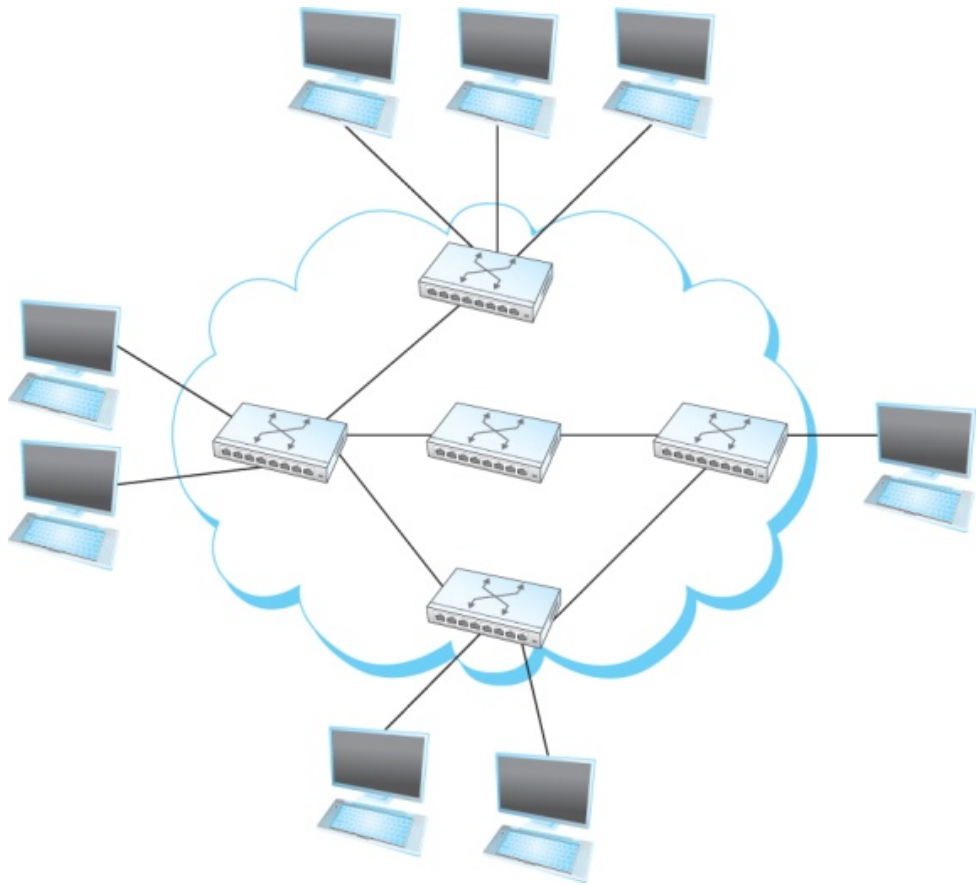| Link (Ethernet) | ←---- | Ethernet Interface | | Ethernet Interface | ----→ | Link (Ethernet) |

Bits (1010001)

2

Links, Nodes, Network, Internet

- You can view the network as a graph

- Each device (a phone, a computer) is a node

- Each connection is a link
  - Wires = real links
  - Bluetooth, Radio, Infrared = virtual links

- Nodes + links = a network
  - Many connected networks = Internet
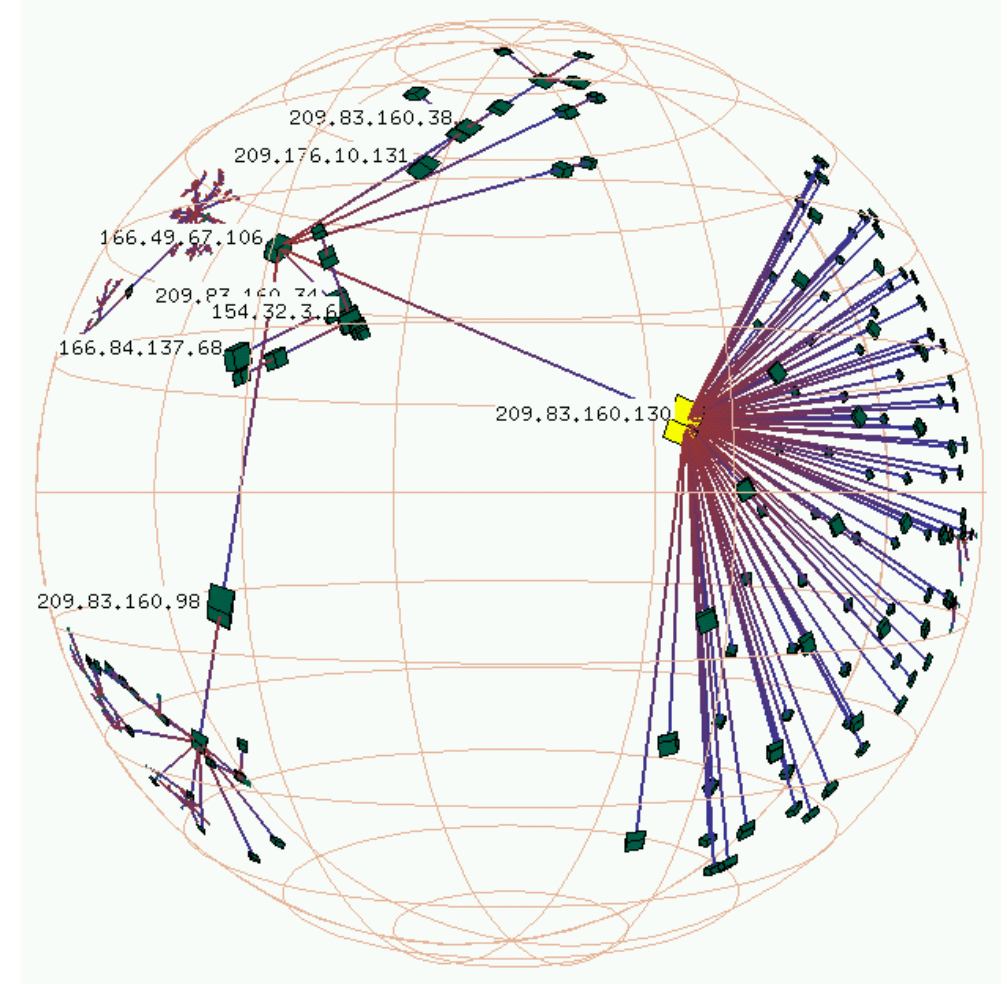
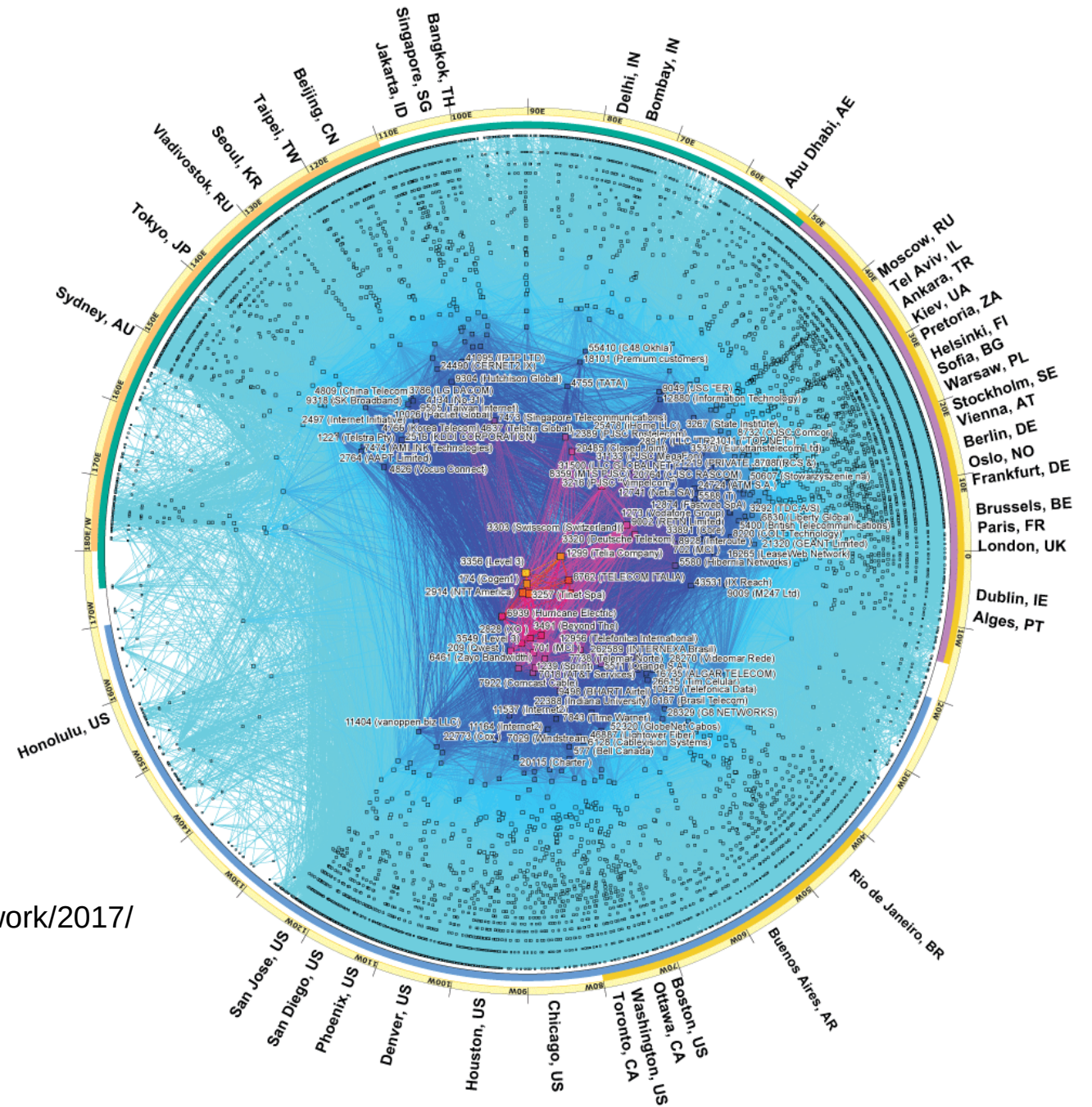# A Network and the Internet

Links, Nodes, Network, Internet

Not
Actual
data

https://www.caida.org/tools/visualization/walrus/gallery1/lhr-old.png
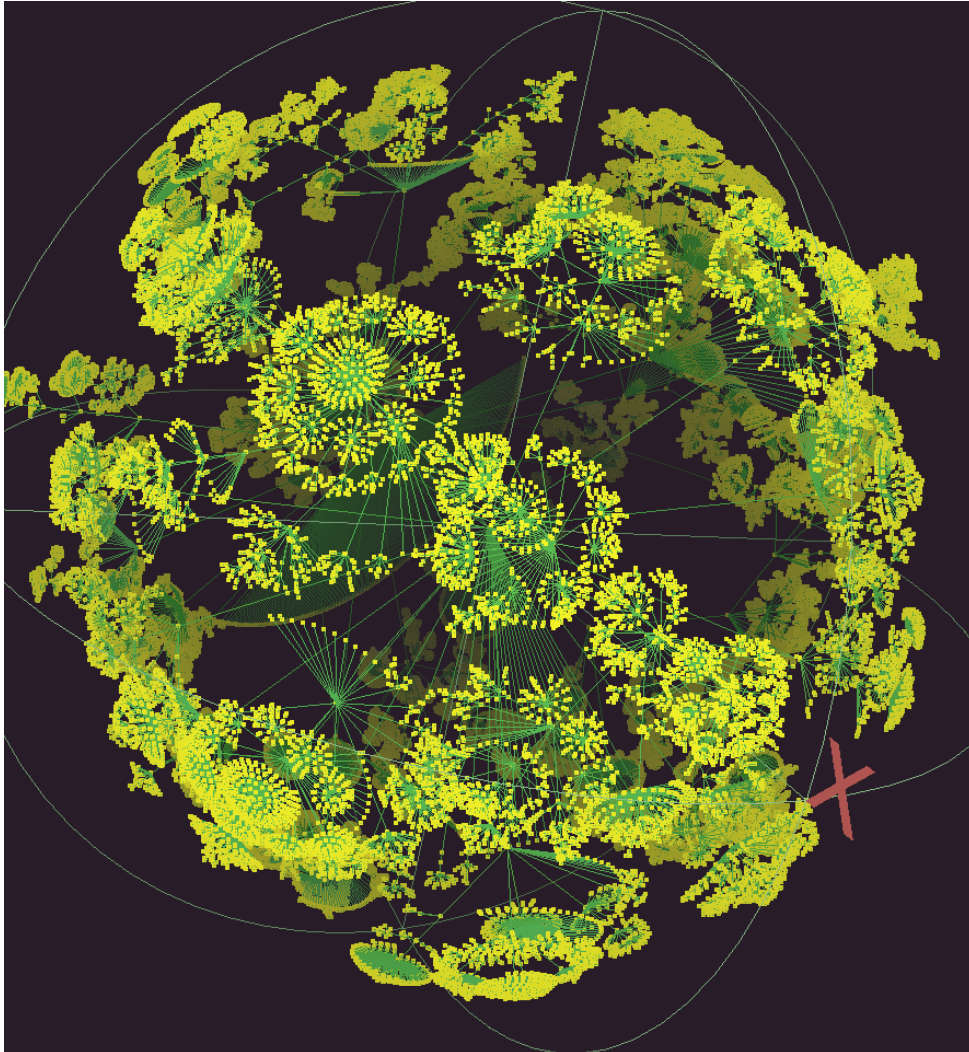
Links, Nodes, Network, Internet

https://www.caida.org/research/topology/as_core_network/2017/

Circuit Switching – Old telephone networks

Operator, get me the navy
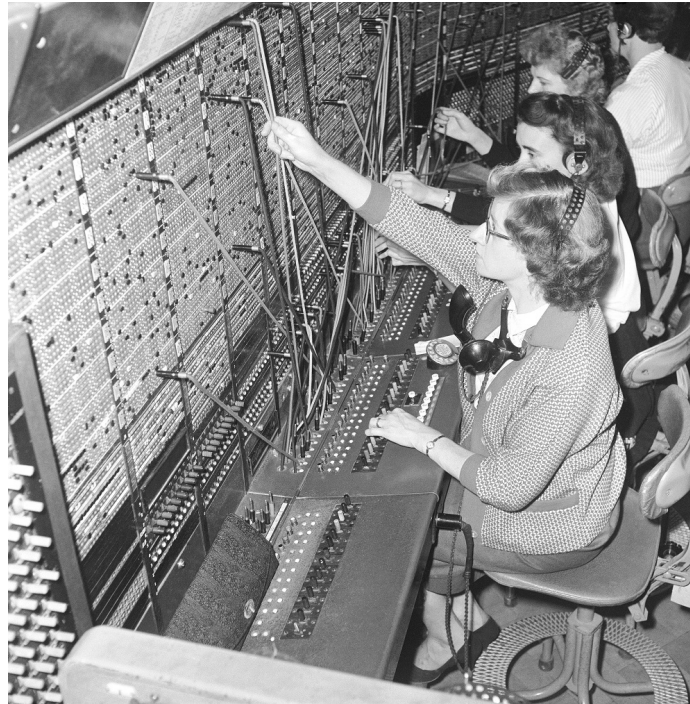
Why change?

- Build physical wire:
  - Guaranteed resources
  - Great for voice

Packet Switching
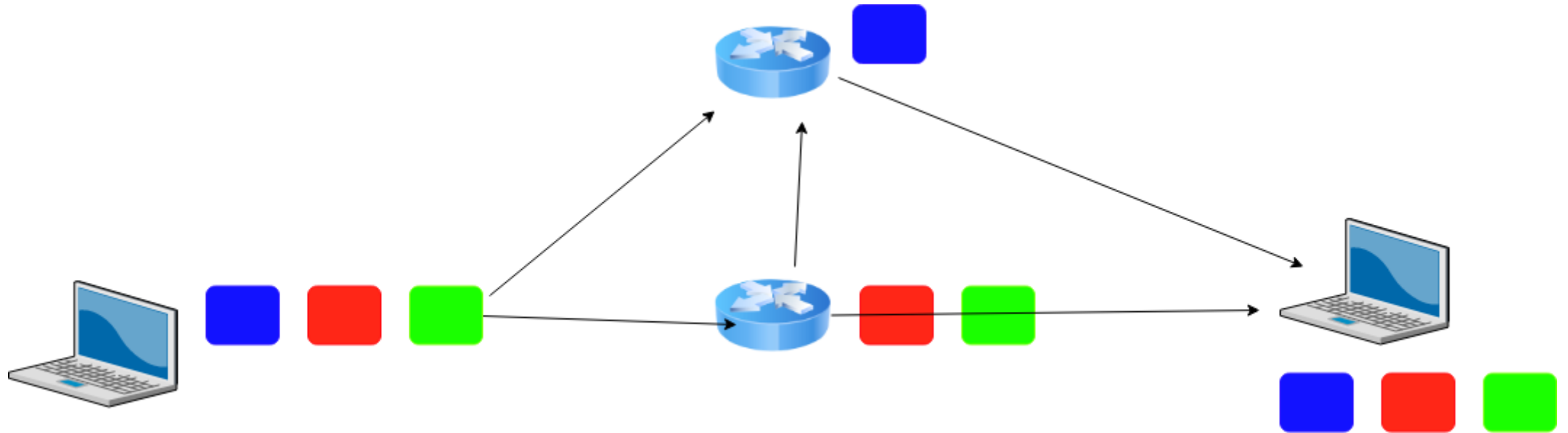
IP Suite

Network Topology
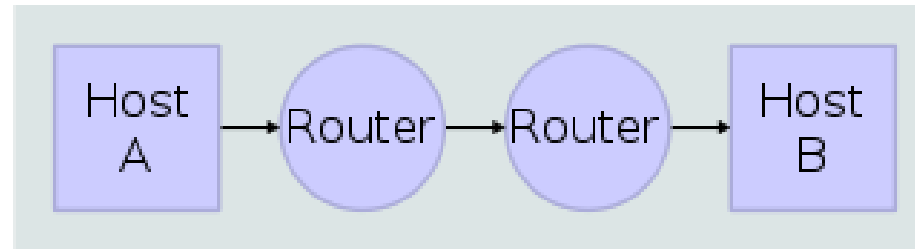
Data Flow
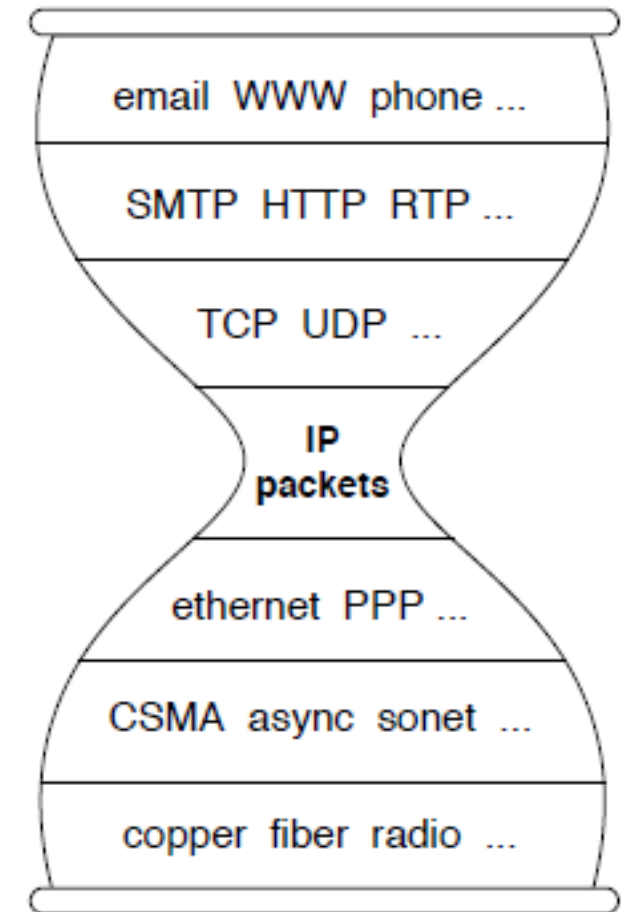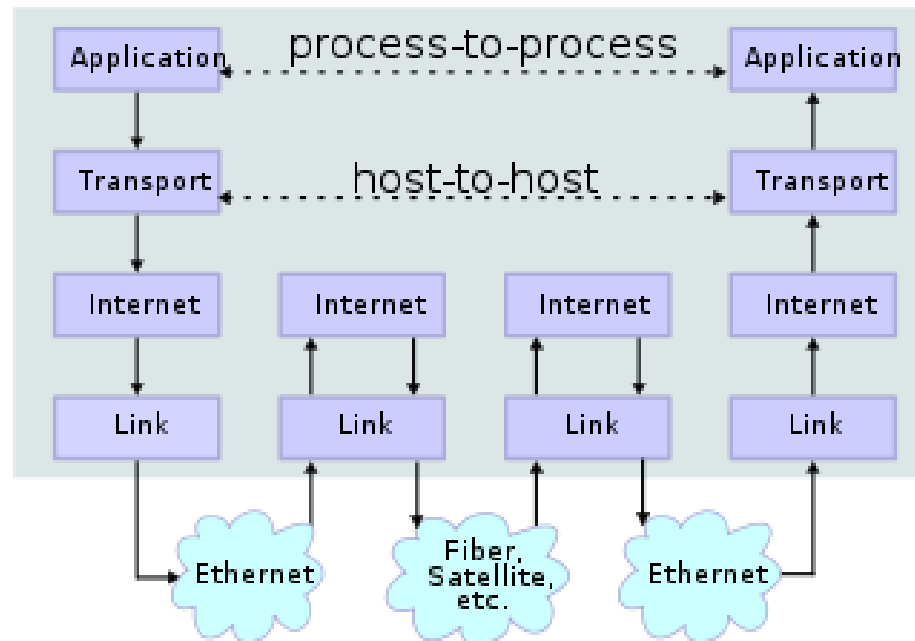
wikipedia

We reject kings, presidents, and voting. We believe in rough consensus and running code. (David Clark, IETF, July 1992)

9

Apps (HTTP) — **Data** → Apps (HTTP)

Transport (TCP/UDP) — **Segments** → Transport (TCP/UDP)

Network (IP) — **Packets** → Network (IP)

Frames

Link (Ethernet) — Ethernet Interface — Ethernet Interface — Link (Ethernet)

Bits (1010001)
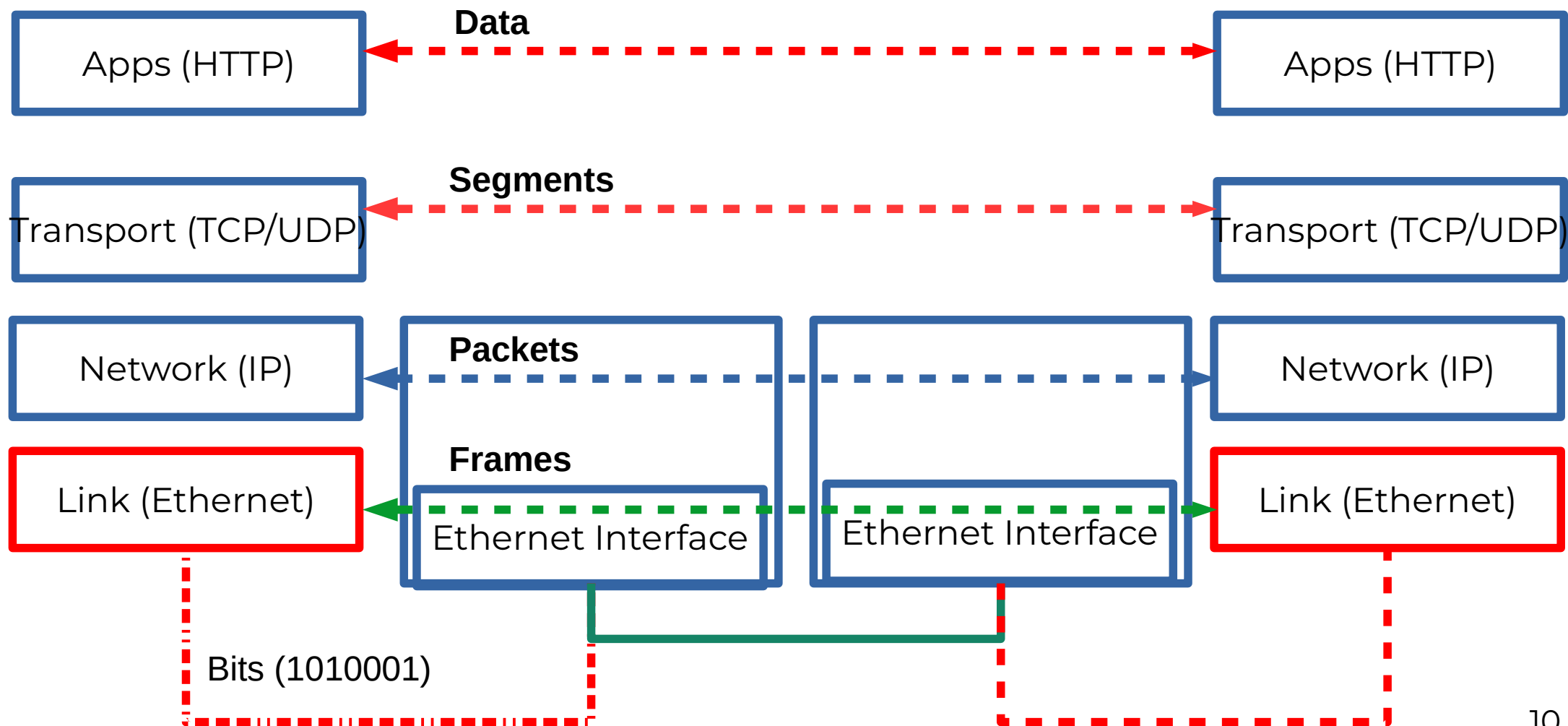
Performance - Bandwidth and Latency

- **Bandwidth = Size of the network pipe**

- **Latency = Delay in sending packets**

- **Throughput = How fast your can send data, function of both bandwidth and latency (and other things)**
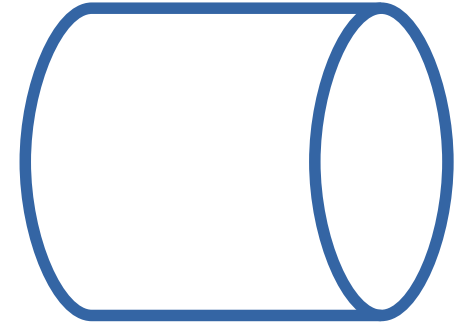
Performance - Latency

- Latency = Propagation Delay + Transmission Delay + Queuing Delay

- Propagation = Distance/Speed Of Light (in Copper or Fiber)

- Transmit = Size/Bandwidth

How long to "put it on the wire?"

How long to traverse the link?

How long to queue?
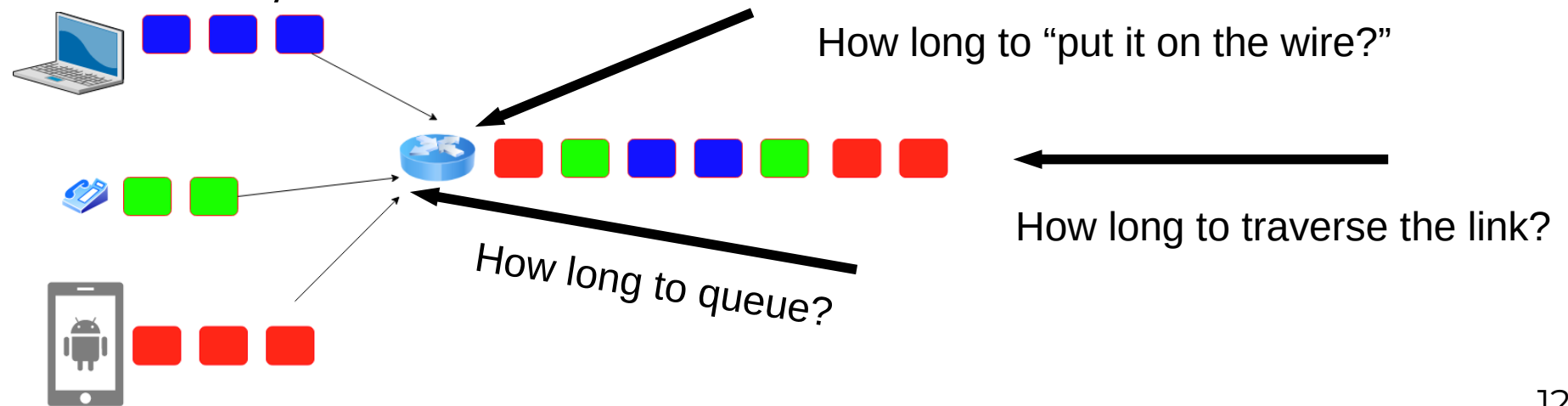
Link Layer Recap – How much work for a cat picture?

Apps (HTTP)	← ← **Data** → → **Apps (HTTP)**
Transport (TCP/UDP)	← ← **Segments** → → **Transport (TCP/UDP)**
Network (IP)	← **Packets** → **Network (IP)**
Link (Ethernet)	← **Frames** → **Link (Ethernet)**

Ethernet Interface   Ethernet Interface

Bits (1010001)

13
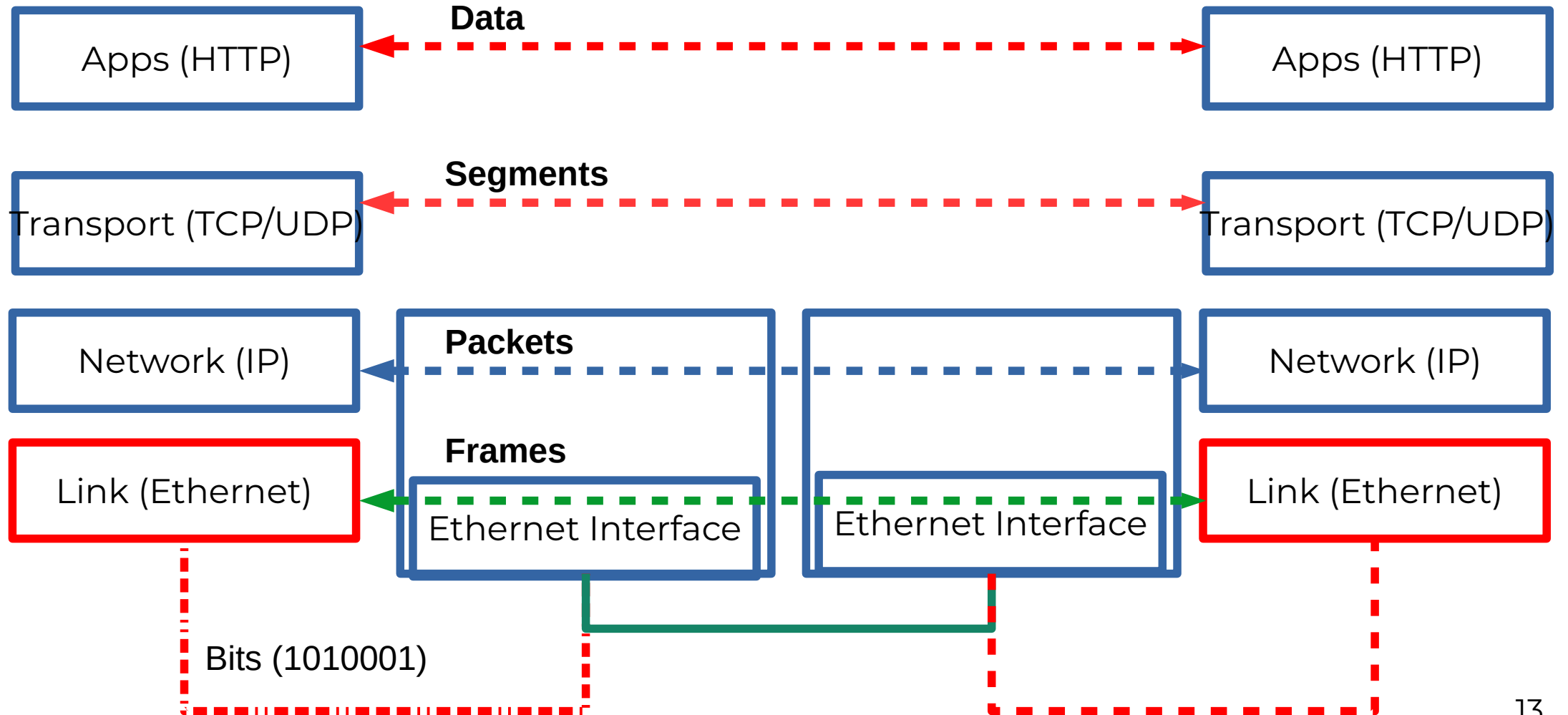
What does it take to create a link?
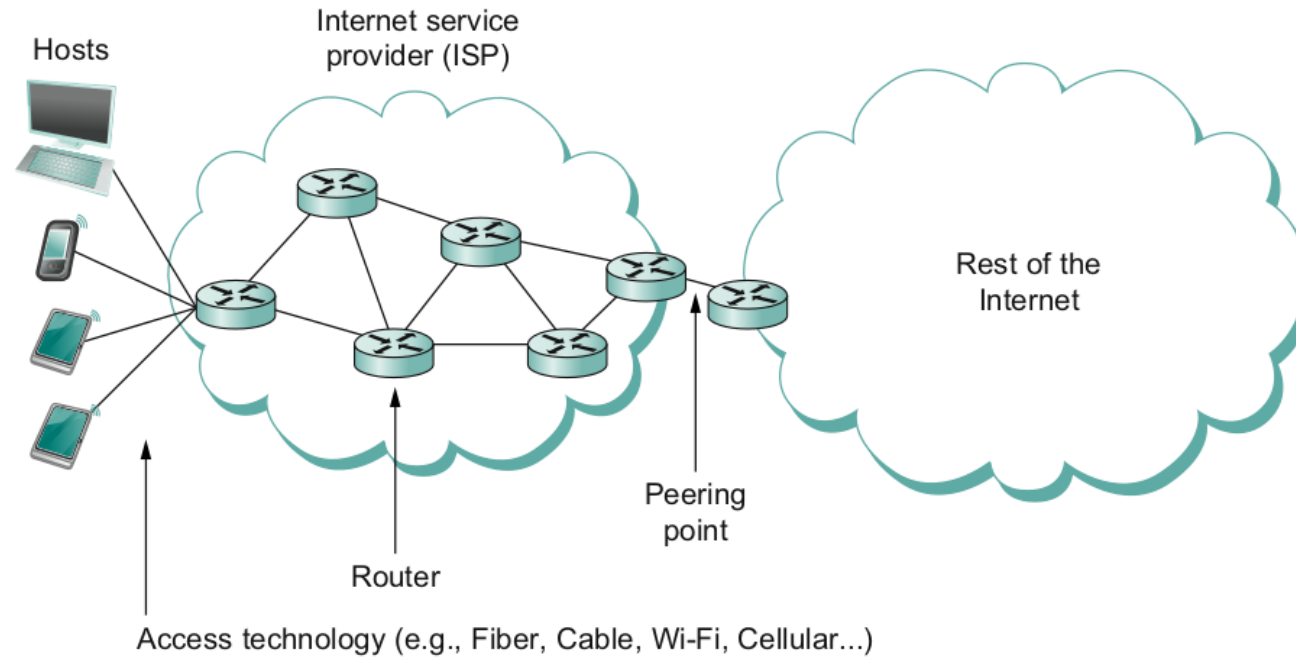
- Common abstractions
  - Why?

Packet to Low level Signals

- Bit pattern - 0101001
  - Must encode it into electrical signals and then decode it on the other end!

Signalling component

Signal

Node | Adaptor | | Adaptor | Node

Bits

Bits 0 0 1 0 1 1 1 1 0 1 0 0 0 0 1 0

NRZ
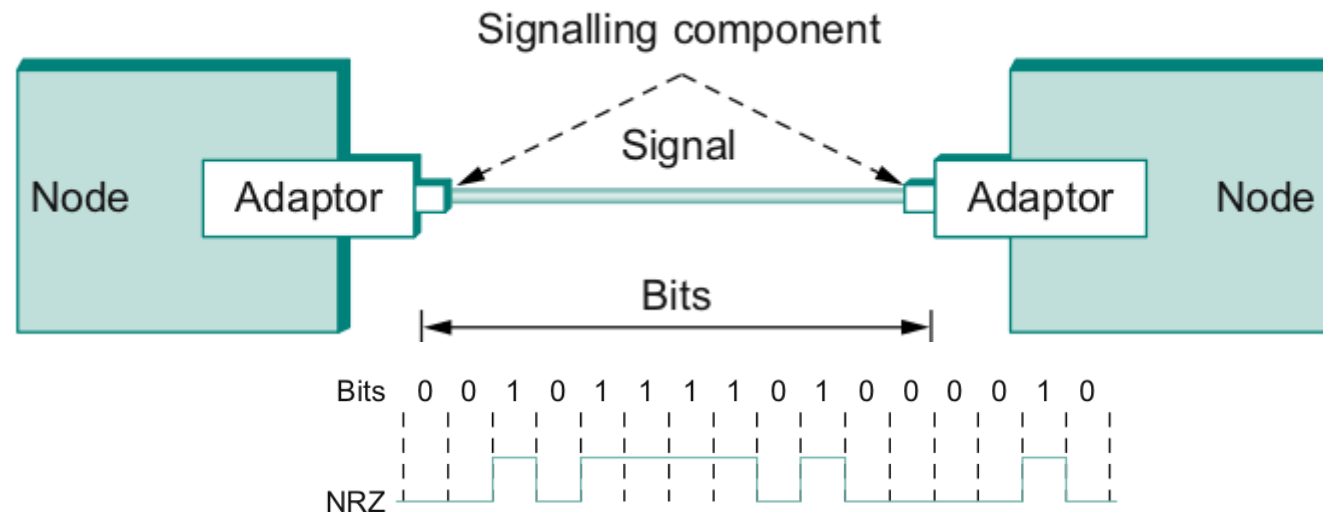
Frames – bag of bits

- Sending side – encapsulation, add error check bits, flow control
- Receiving side – extract frames, check for error, flow control

Error Detection

- Bit errors are introduced into frames
  - Because of electrical interference and thermal noises
- Detecting Error
- Correction Error
- Two approaches when the recipient detects an error
  - Notify the sender that the message was corrupted, so the sender can send again.
    - If the error is rare, then the retransmitted message will be error-free
  - Using some error correct detection and correction algorithm, the receiver reconstructs the message

One an Two-dimensional parity

0	1	0	1	0	**0**

0	1	0	1	1	**1**

Number of 1s
- Odd 1s = Parity bit 0
- Even 1s = Parity bit 1

Parity bits

Data	0101001	1
	1101001	0
	1011110	1
	0001110	1
	0110100	1
	1011111	0
Parity byte	1111011	0

Two Dimensional Parity

Internet Checksum Algorithm (RFC 1071)

-

- A = 0110011001100110
- B = 0101010101010101

  A+B = 1011101110111011
- C =   0000111100001111

        1100101011001010 (sum of all segments)

        0011010100110101 (1's complement, 1→0, 0→ 1) <= this is the checksum

  At receiver:

  Add sum of all segments and checksum

  − If correct, all 1s

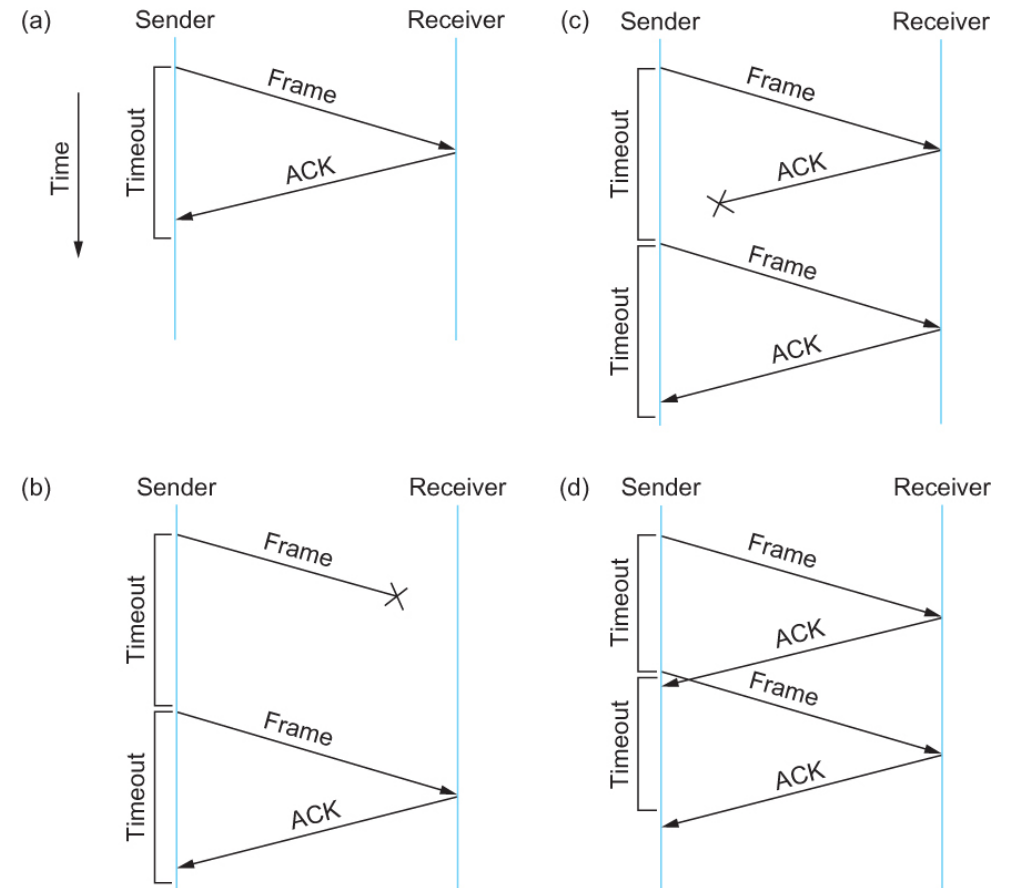# Reliable Delivery – Correct FRAMEs!!!

- Frames might get lost
  - Too many bits lost
  - Clock did not sync properly
  - Error detected but the report got lost

- Can we build links that does not have errors?
  - Not possible

- How about all those error correction stuff we learned?
  - Can we add them to frames?
  - We could, but think of the overhead
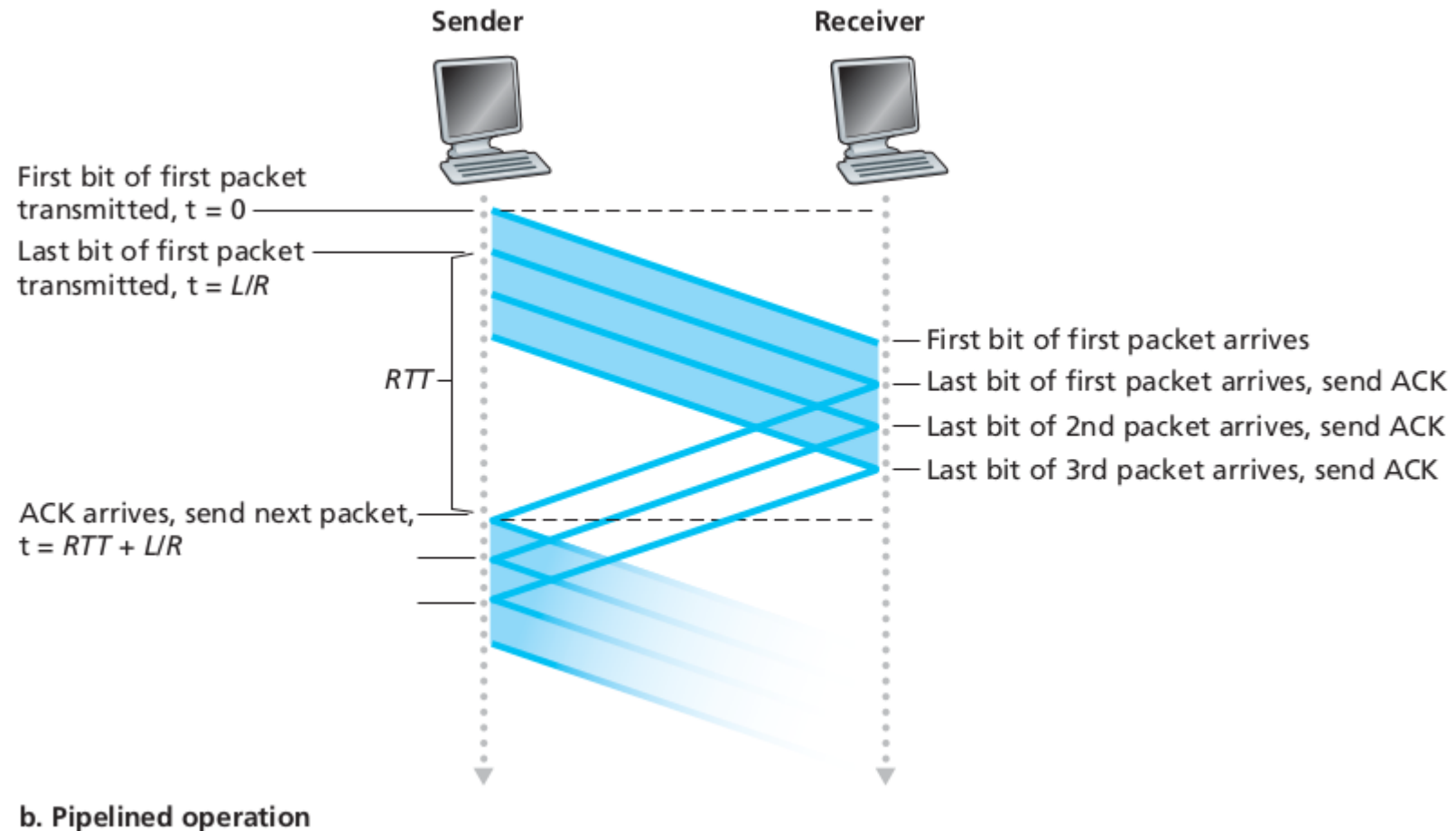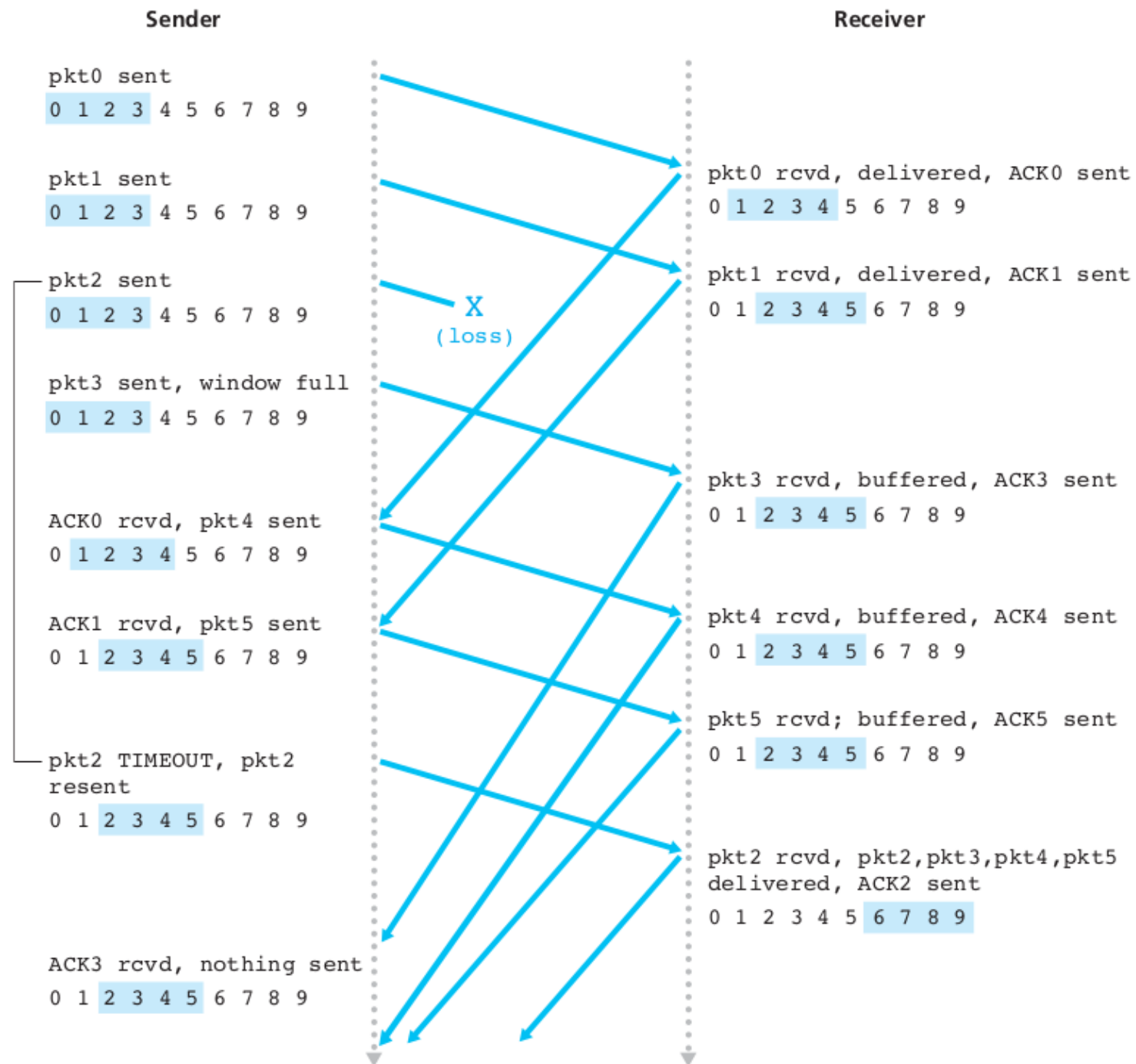 - What happens when the entire frame is lost?

Stop and Wait

- Sender sends a frame, sets a timeout (e.g., 1 sec)

- Receiver receives the frame, sends an ACK

- Sender
  - sends the next frame on ACK
  - retransmits the same frame if timeout happens

- **Spot the bugs in the protocol**
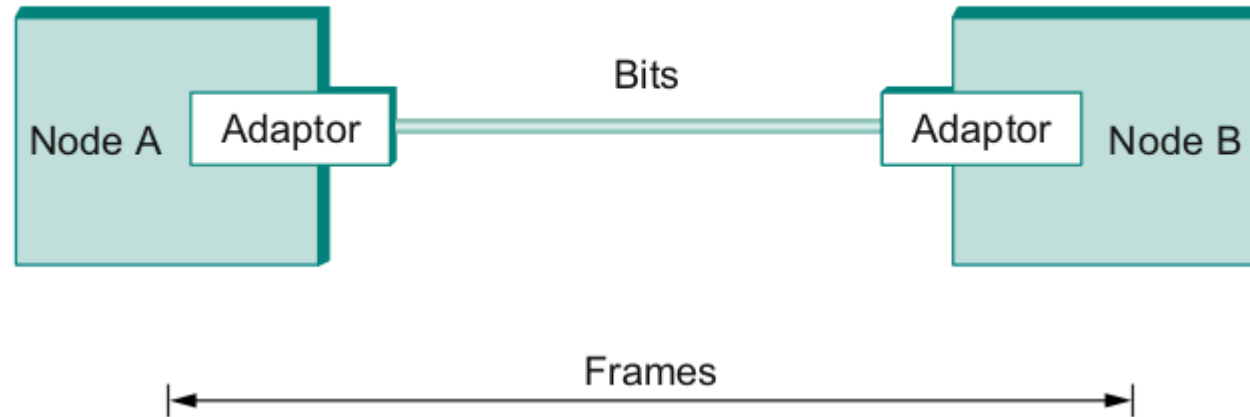
Sliding window to the rescue!

Utilization = 0.008*3/30.008 = 0.00079 (3 times increase)

b. Pipelined operation

Sender   **Receiver**

pkt0 sent
0 1 2 3 4 5 6 7 8 9

pkt1 sent   pkt0 rcvd, delivered, ACK0 sent
0 1 2 3 4 5 6 7 8 9   0 1 2 3 4 5 6 7 8 9

pkt2 sent   pkt1 rcvd, delivered, ACK1 sent
0 1 2 3 4 5 6 7 8 9   0 1 2 3 4 5 6 7 8 9

                          X
                        (loss)

pkt3 sent, window full
0 1 2 3 4 5 6 7 8 9   pkt3 rcvd, buffered, ACK3 sent
                                                     0 1 2 3 4 5 6 7 8 9
ACK0 rcvd, pkt4 sent
0 1 2 3 4 5 6 7 8 9

ACK1 rcvd, pkt5 sent   pkt4 rcvd, buffered, ACK4 sent
0 1 2 3 4 5 6 7 8 9   0 1 2 3 4 5 6 7 8 9

                                                     pkt5 rcvd; buffered, ACK5 sent
                                                     0 1 2 3 4 5 6 7 8 9
pkt2 TIMEOUT, pkt2
resent
0 1 2 3 4 5 6 7 8 9   pkt2 rcvd, pkt2,pkt3,pkt4,pkt5
                                                     delivered, ACK2 sent
                                                     0 1 2 3 4 5 6 7 8 9

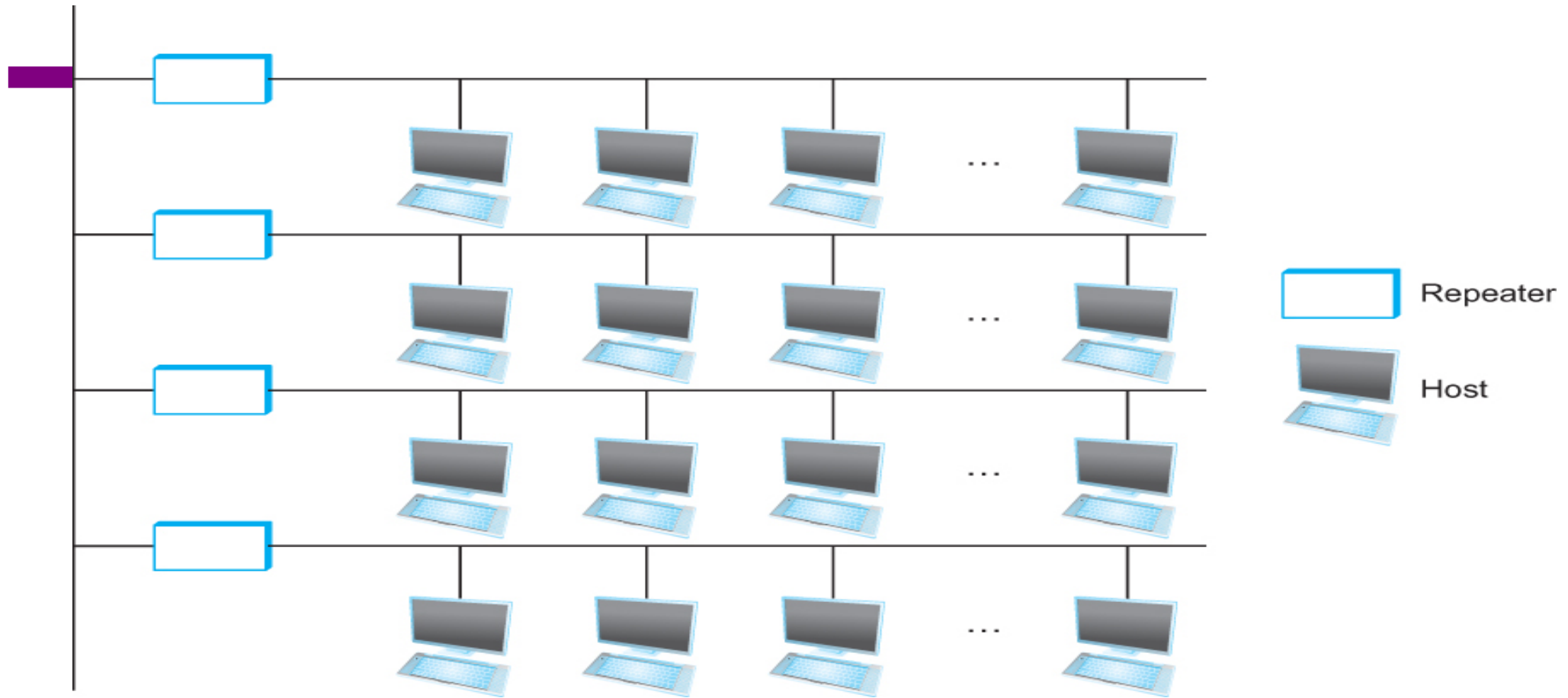ACK3 rcvd, nothing sent
0 1 2 3 4 5 6 7 8 9

23

So far we connected two machines – how about more than two?

- We have connected two machines using point to point wires
  - Encoded bits
  - Sent bits as Frames
  - Caught and corrected errors
  - Tuned efficiency and reliability using sliding window

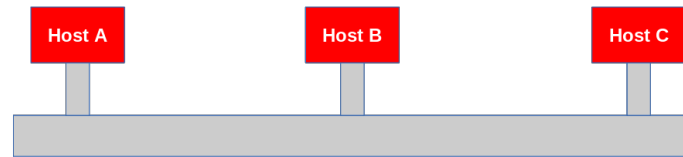- What happens when there are more than two machines?
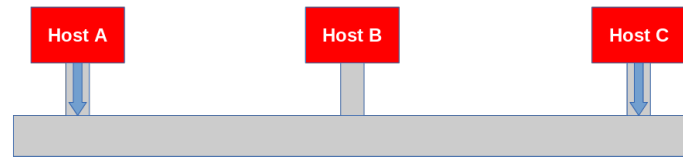
Ethernet

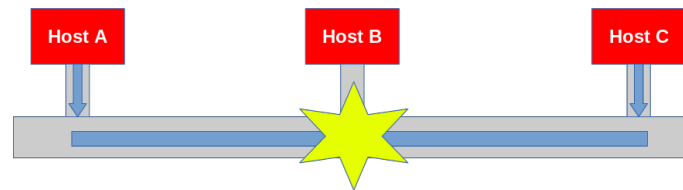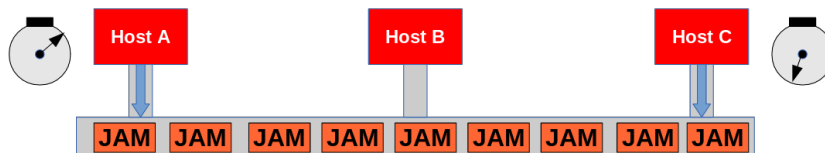Ethernet repeater

25

CSMA/CD – Ethernet

1) Carrier Sense
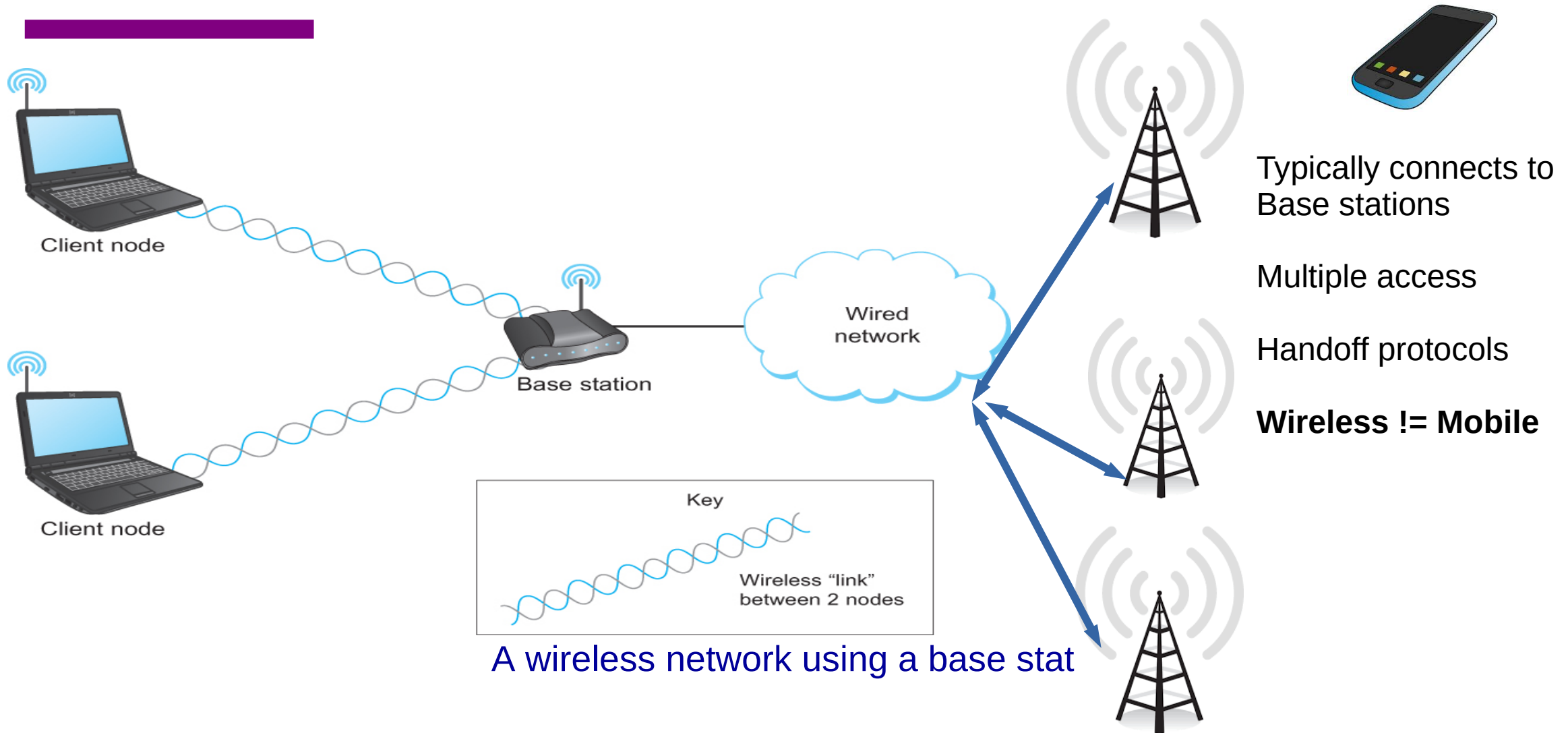

2) Multiple Access

3) Collision

**4) Collision Detection (Back off Algorithmus)**

- CS – wait until idle
  - Channel idle – trasmit
  - Channel busy – wait

- CD – listen while transmitting
  - No collision: transmission successful
  - Collission: abort, send jam signal (32bit special sequence)

- Wait random time
  - Try again
  - After m^{th} collision, $t = random(0, 2^{m-1})$,
  - Wait t*512 bit times before retry

26

Wireless Links - Infrastructure

Client node

Client node

Base station

Wired network

Key

Wireless "link" between 2 nodes

A wireless network using a base stat

Typically connects to Base stations

Multiple access

Handoff protocols
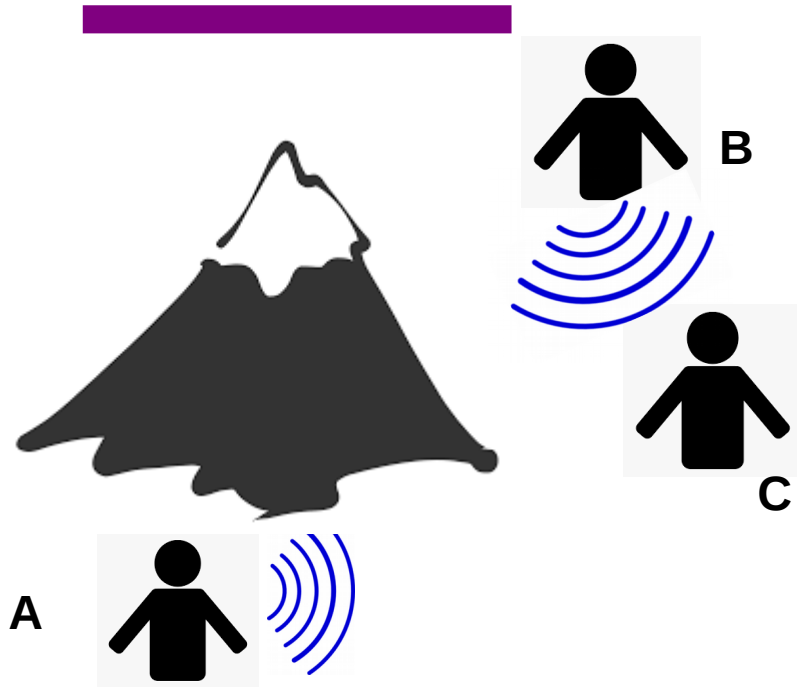
Wireless != Mobile

Wireless Links – Ad hoc

- Mesh or Ad-hoc network
  - Nodes are peers
  - Messages may be forwarded via a chain of peer nodes
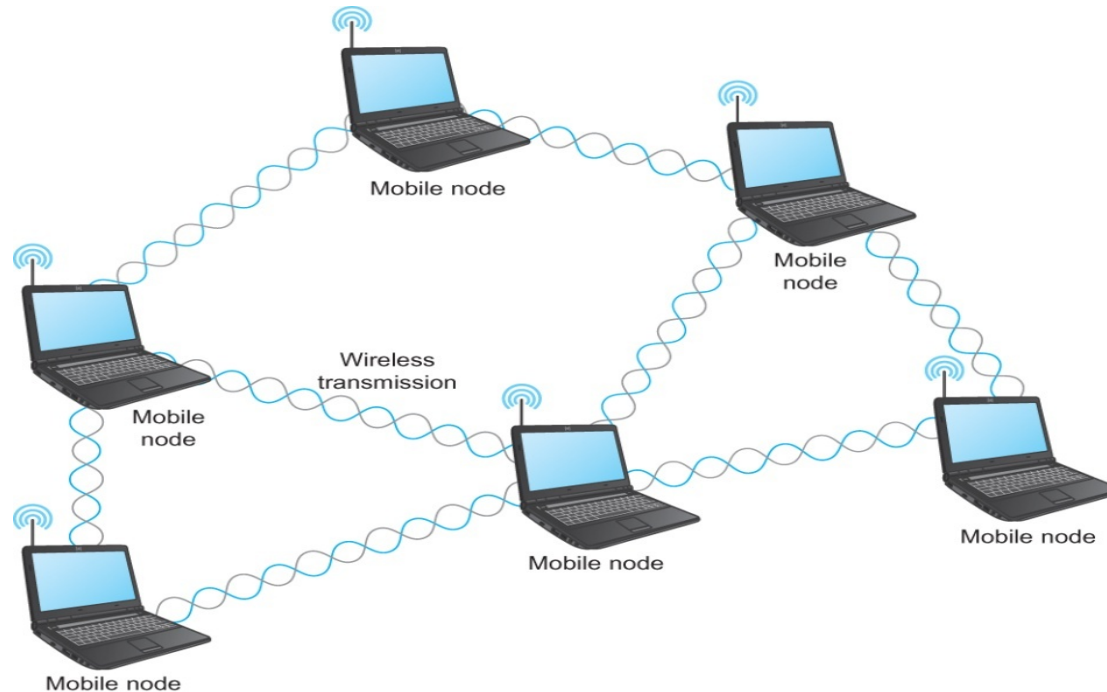
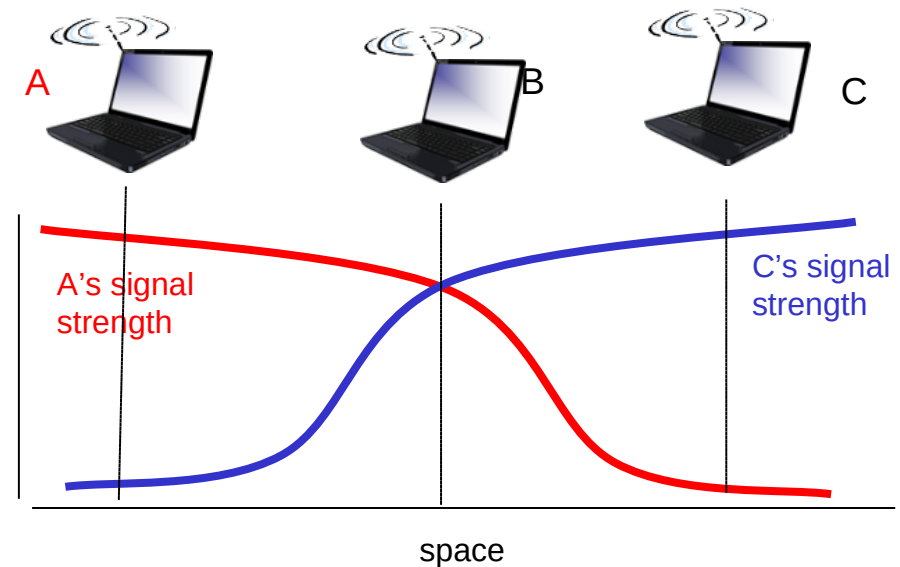# Wireless Links – problems

B

C

A

A and C can talk
B and C can talk
A and B can not!!!
Interference at B

Hidden terminal

Signal Fading

A

B

C

A's signal strength

C's signal strength
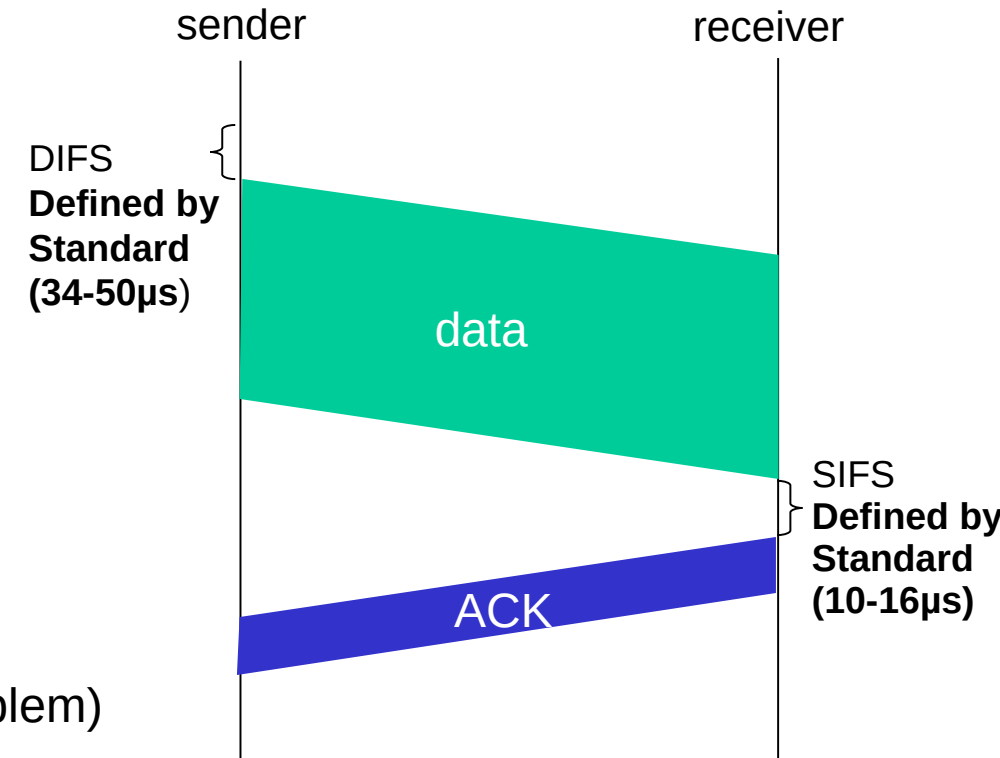
space

IEEE 802.11 MAC Protocol: CSMA/CA

802.11 sender

1 if sense channel idle for **DIFS** then

   transmit entire frame (no CD)

2 if sense channel busy then

   start random backoff time

   timer counts down while channel idle

   transmit when timer expires
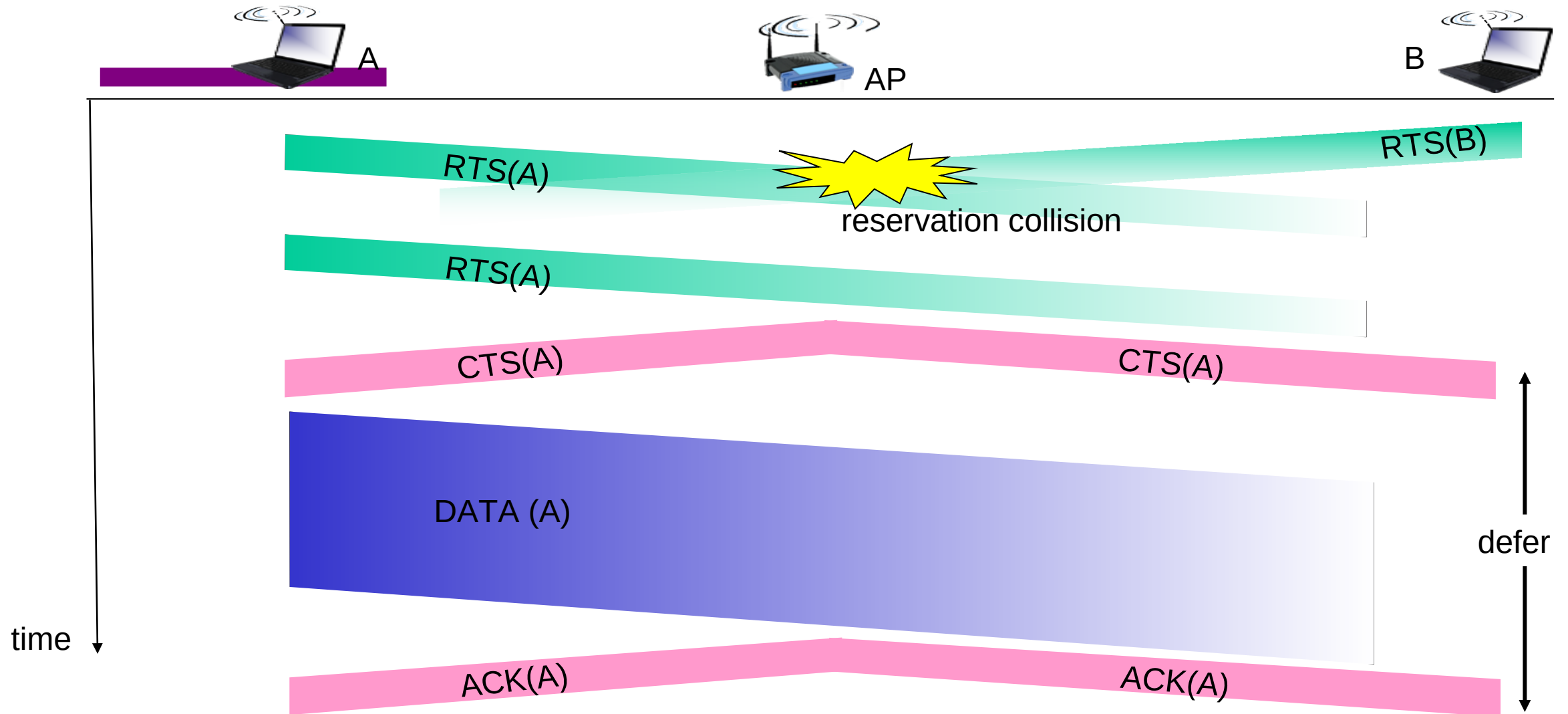
   if no ACK, increase random backoff interval, repeat 2

802.11 receiver

- if frame received OK

   return ACK after **SIFS** (ACK needed due to hidden terminal problem)

DIFS = SIFS + (2 * Slot time)

sender

receiver

DIFS
Defined by Standard (34-50µs)

data

SIFS
Defined by Standard (10-16µs)

ACK

Collision Avoidance: RTS-CTS exchange

A

AP

B

RTS(B)

RTS(A)

reservation collision

RTS(A)

CTS(A)

CTS(A)

DATA (A)

defer

time

ACK(A)

ACK(A)

Next Step – Cat in bits
To Cat in packets!!!!

Data

Apps (HTTP)   Apps (HTTP)

Segments

Transport (TCP/UDP)   Transport (TCP/UDP)

Packets

Network (IP)   Network (IP)

Frames

Link (Ethernet)   Ethernet Interface   Ethernet Interface   Link (Ethernet)

Bits (1010001)