

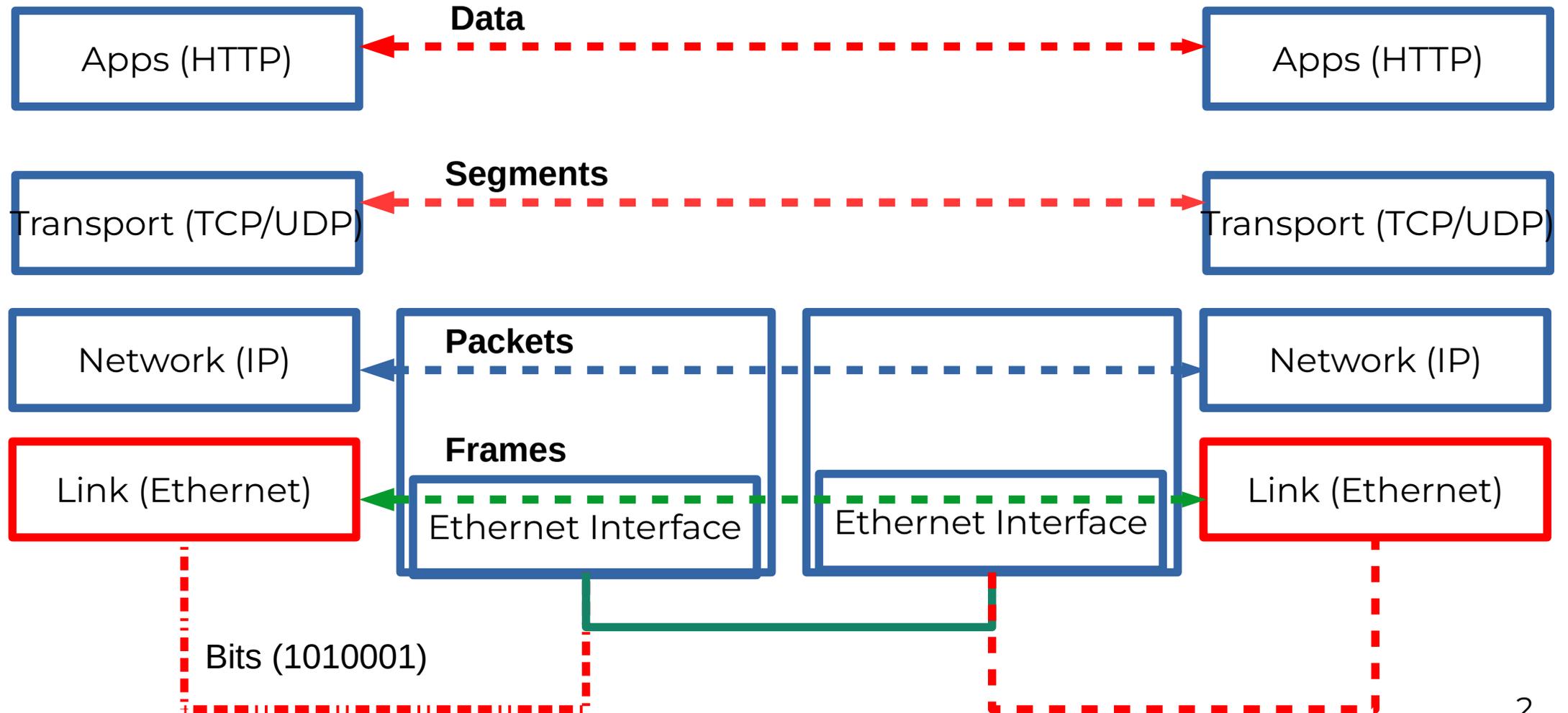
CSC4200/5200 – COMPUTER NETWORKING

PHYSICAL AND LINK LAYER RECAP

Instructor: Susmit Shannigrahi
sshannigrahi@tntech.edu



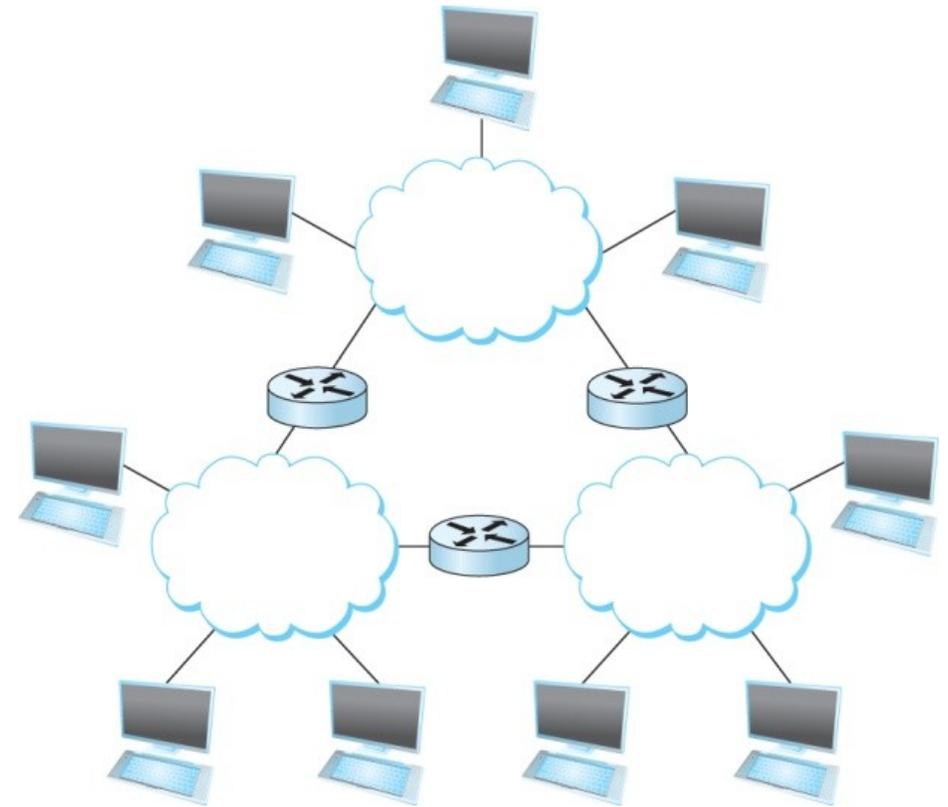
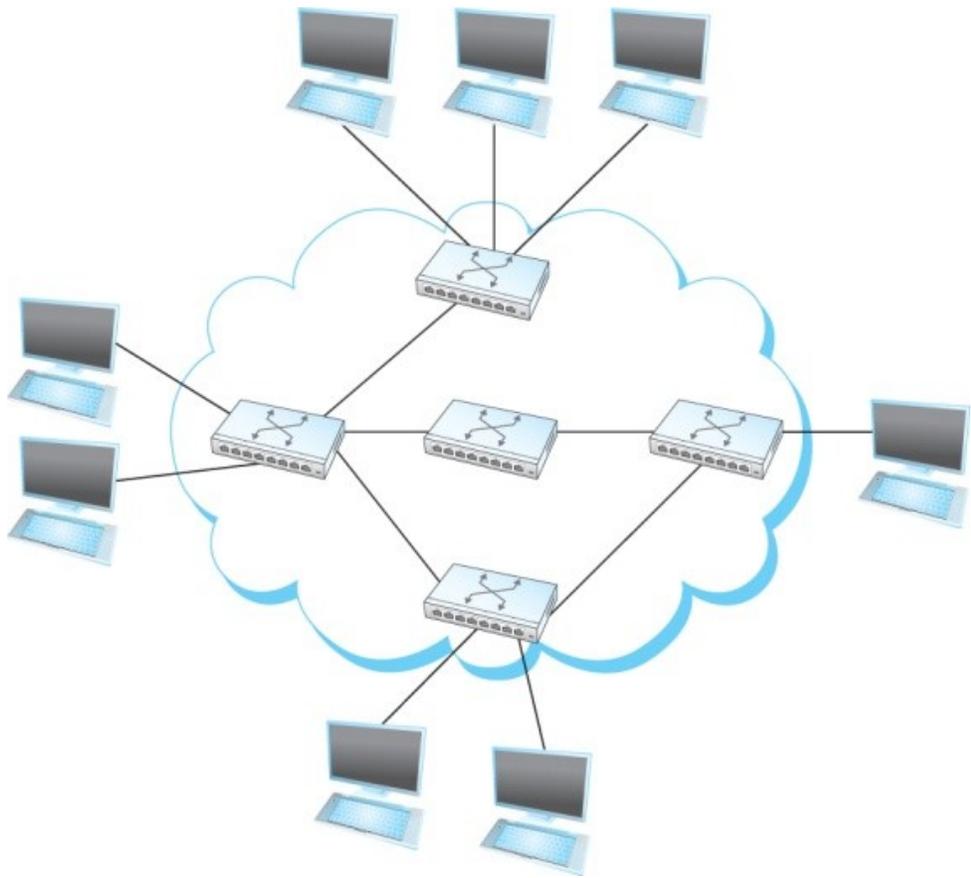
Recap – All this for a cat picture!!



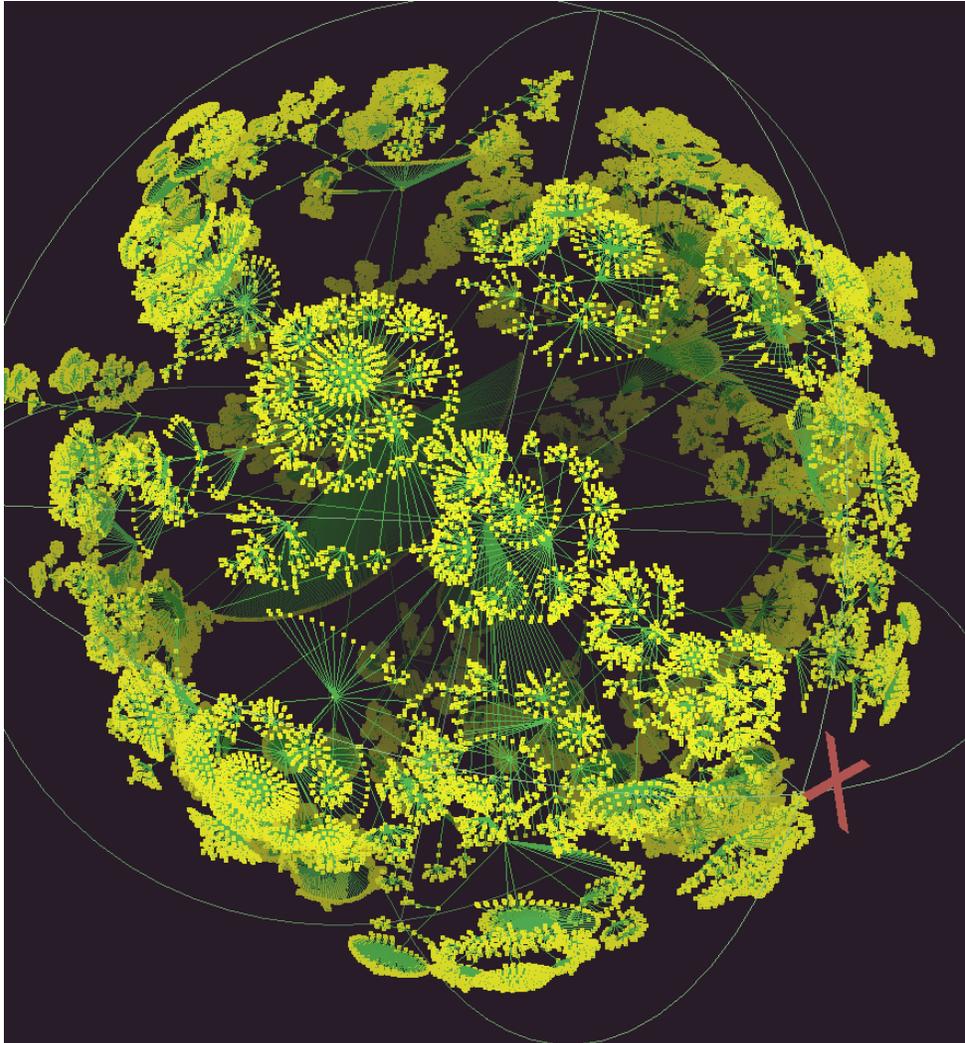
Links, Nodes, Network, Internet

- 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

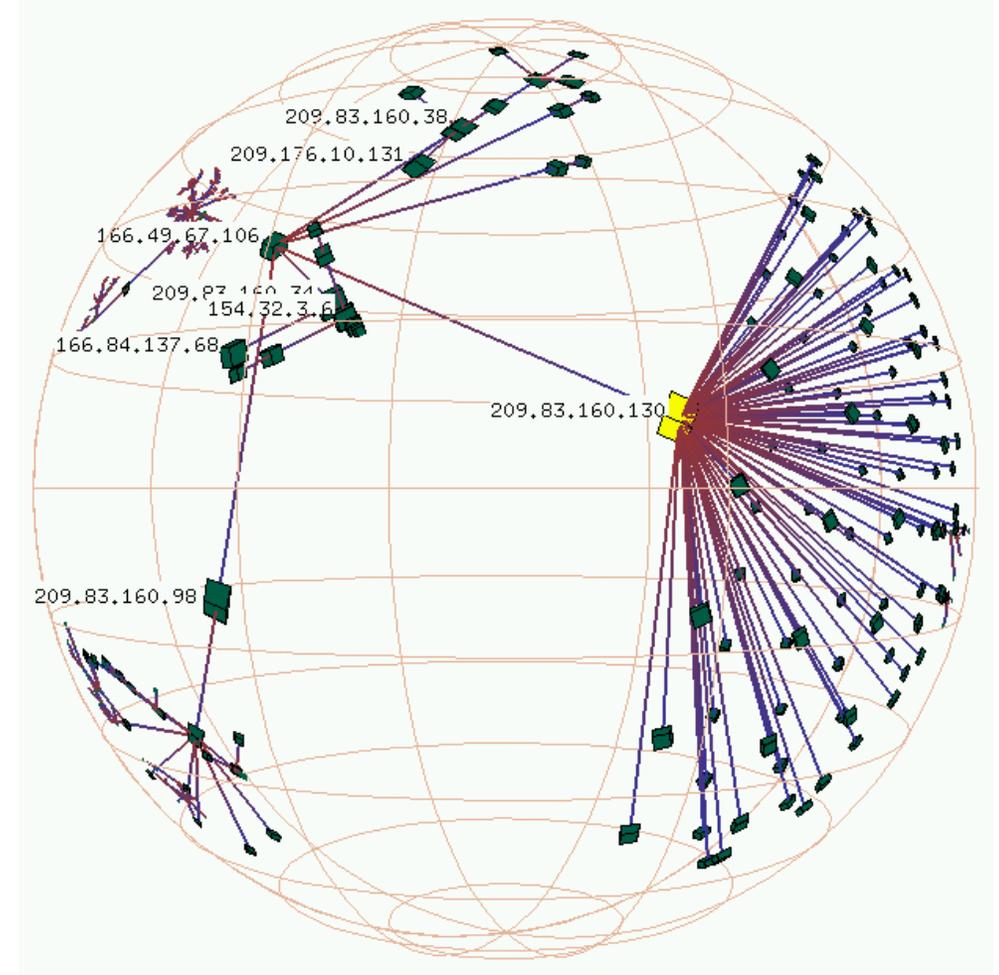
A Network and the Internet



Links, Nodes, Network, Internet

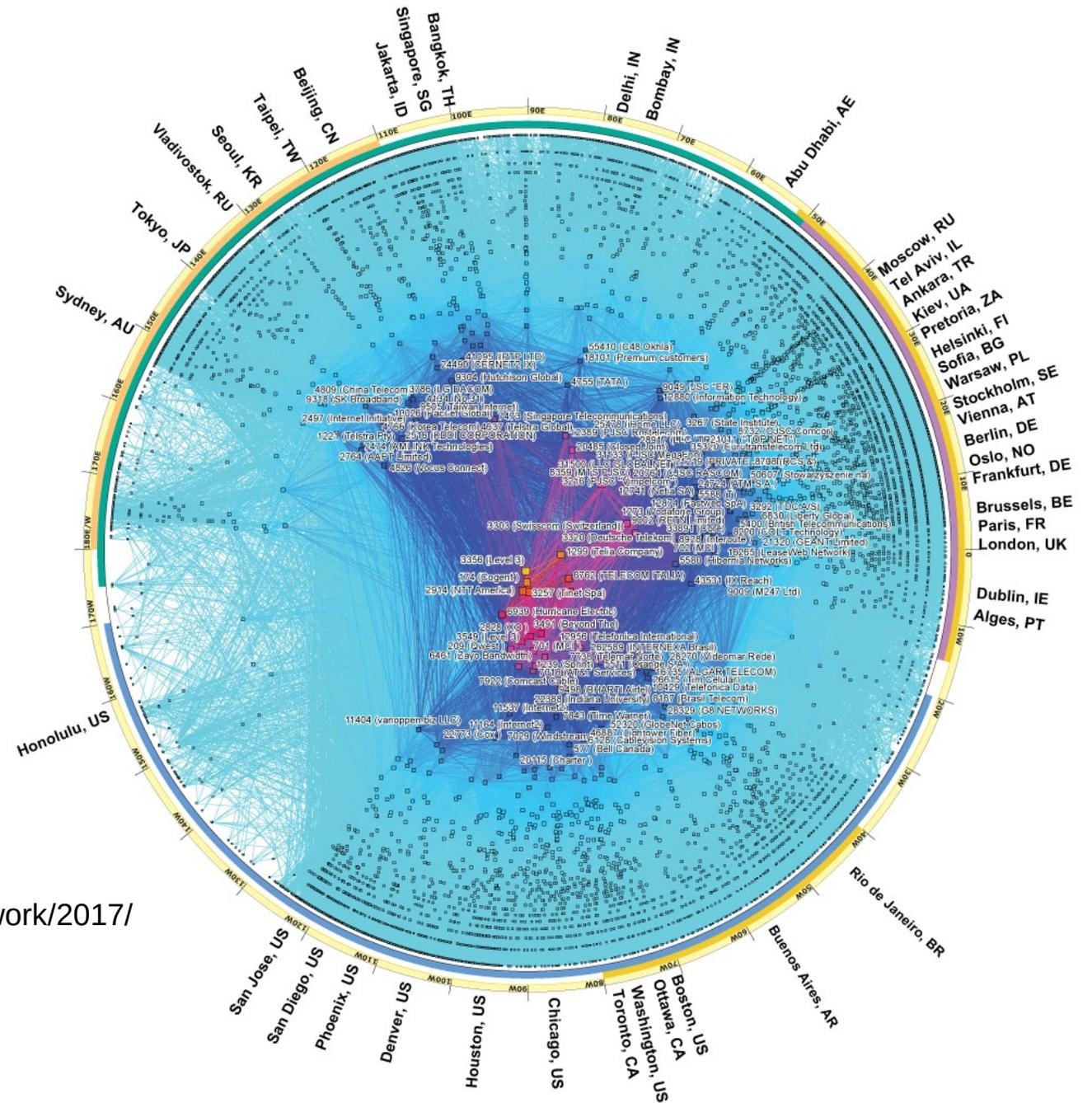


Not
Actual
data



<https://www.caida.org/tools/visualization/walrus/gallery1/lhr-old.png>

Links, Nodes, Network, Internet



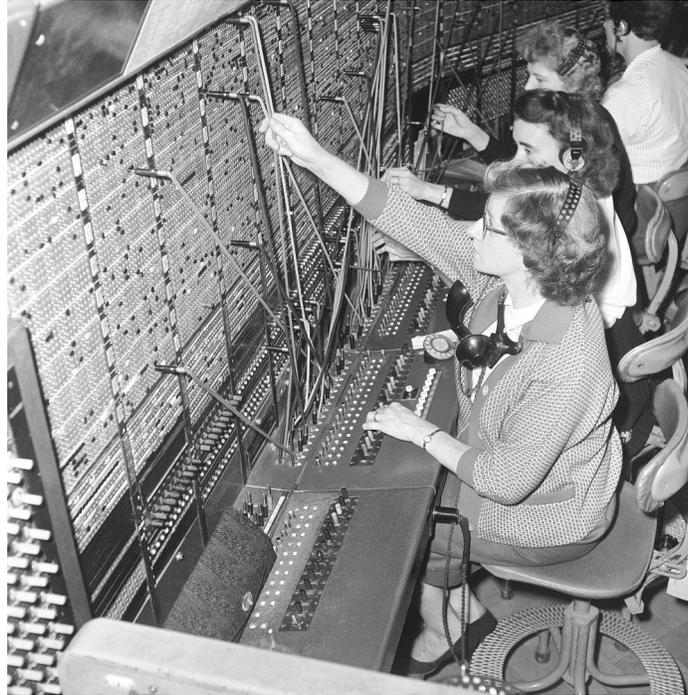
https://www.caida.org/research/topology/as_core_network/2017/

Can you create a network with both wired and wireless links?

Circuit Switching – Old telephone networks



Operator, get me
the navy

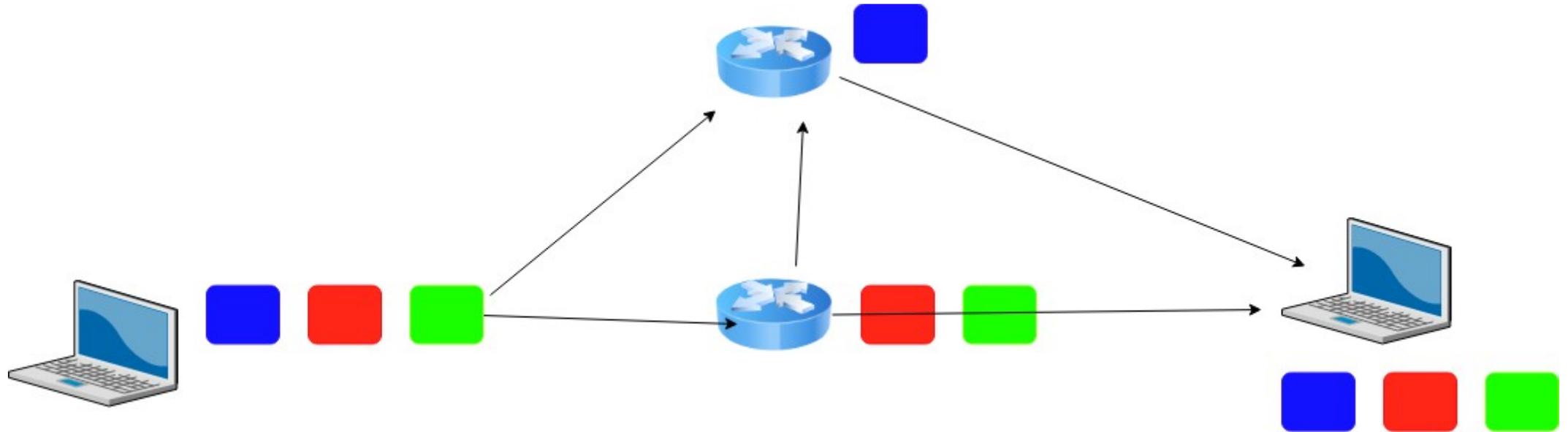


- Build physical wire:
 - Guaranteed resources
 - Great for voice

Why change?

•

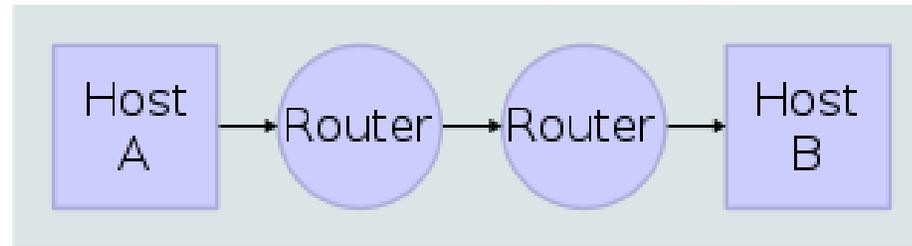
Packet Switching



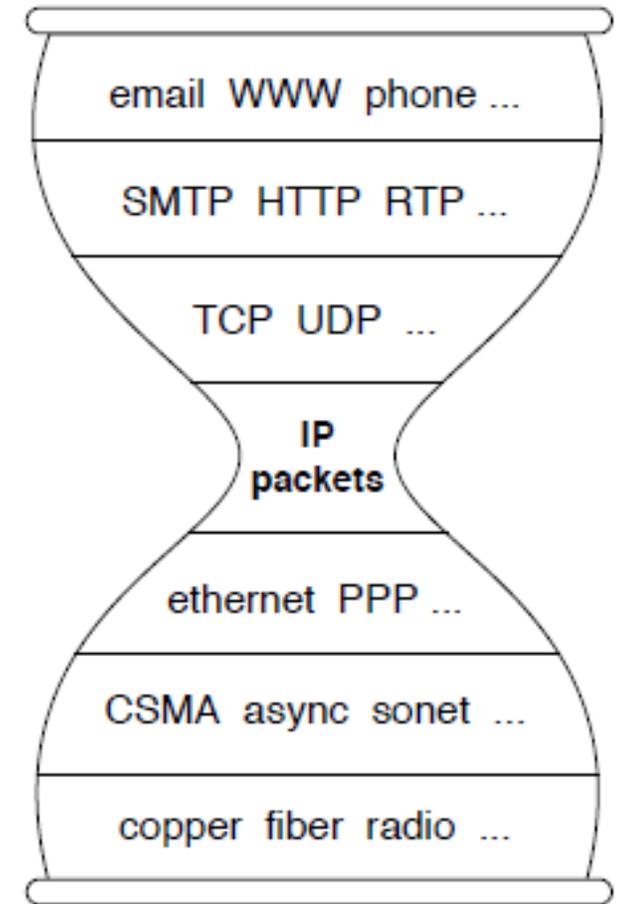
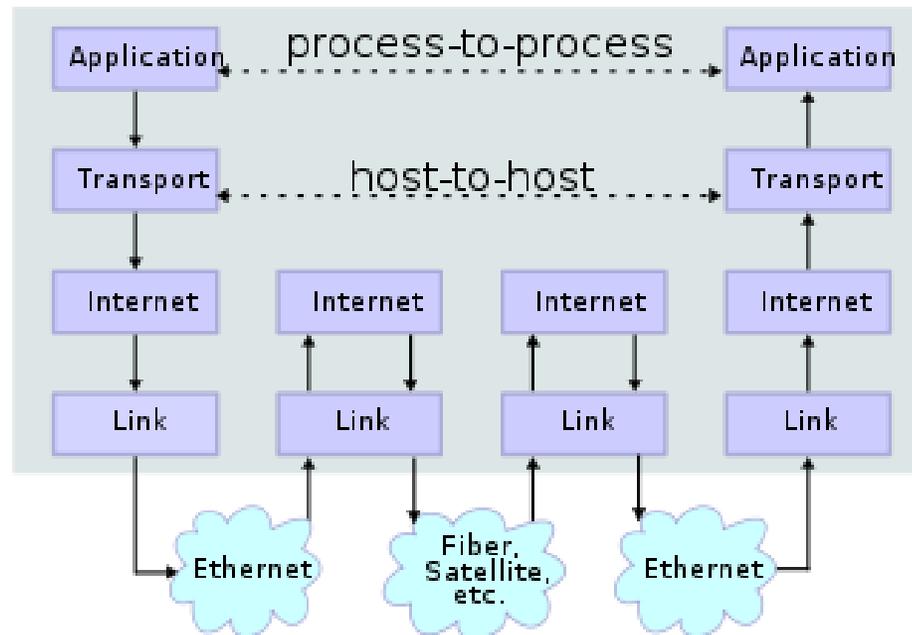
With 10000 users using a 10Mbps link, which method is more efficient?

IP Suite

Network Topology

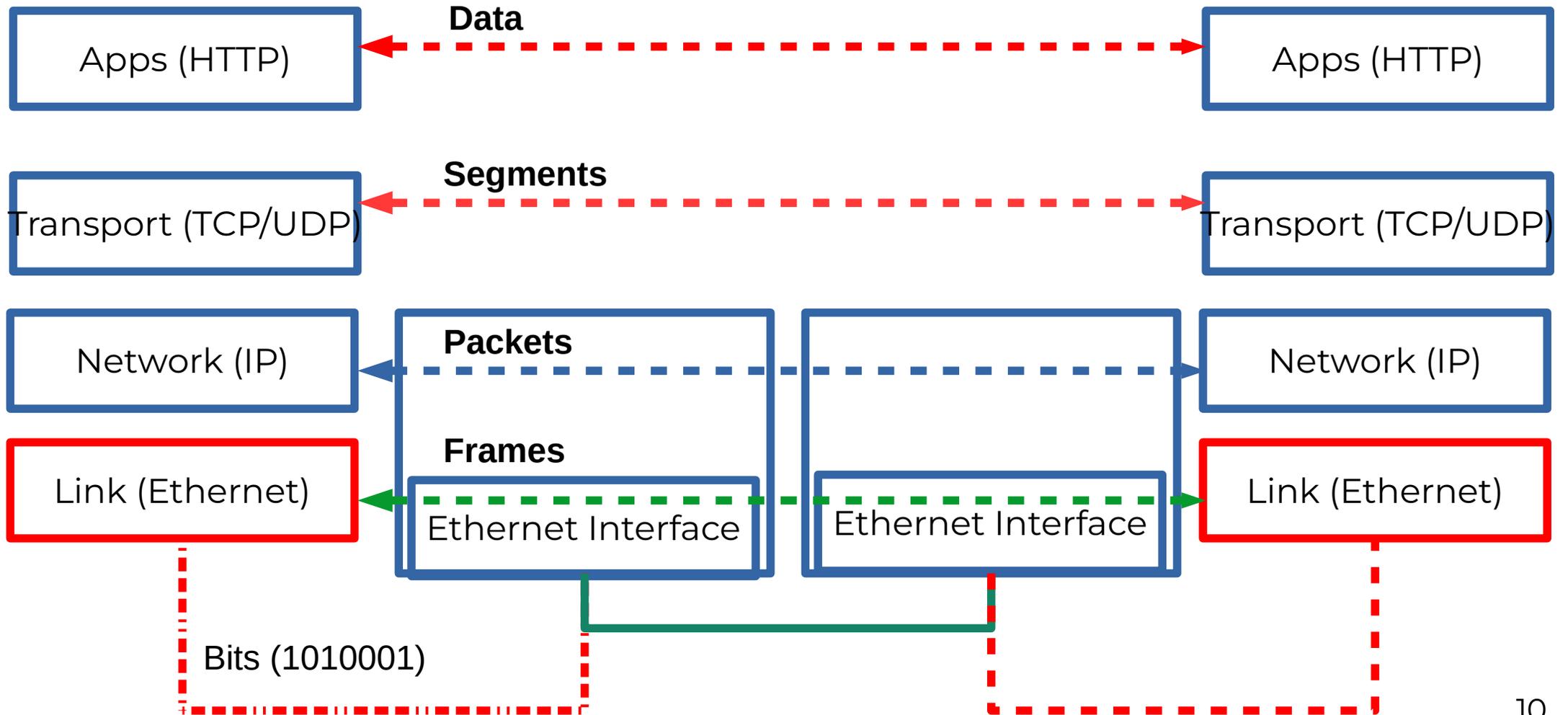


Data Flow



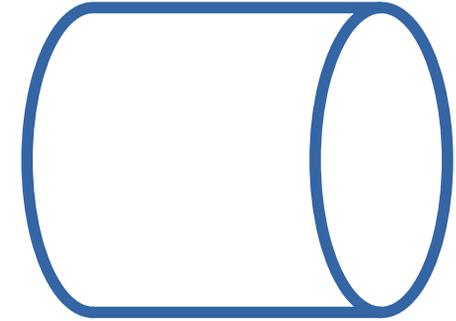
We reject kings, presidents, and voting. We believe in rough consensus and running code. (David Clark, IETF, July 1992)

wikipedia



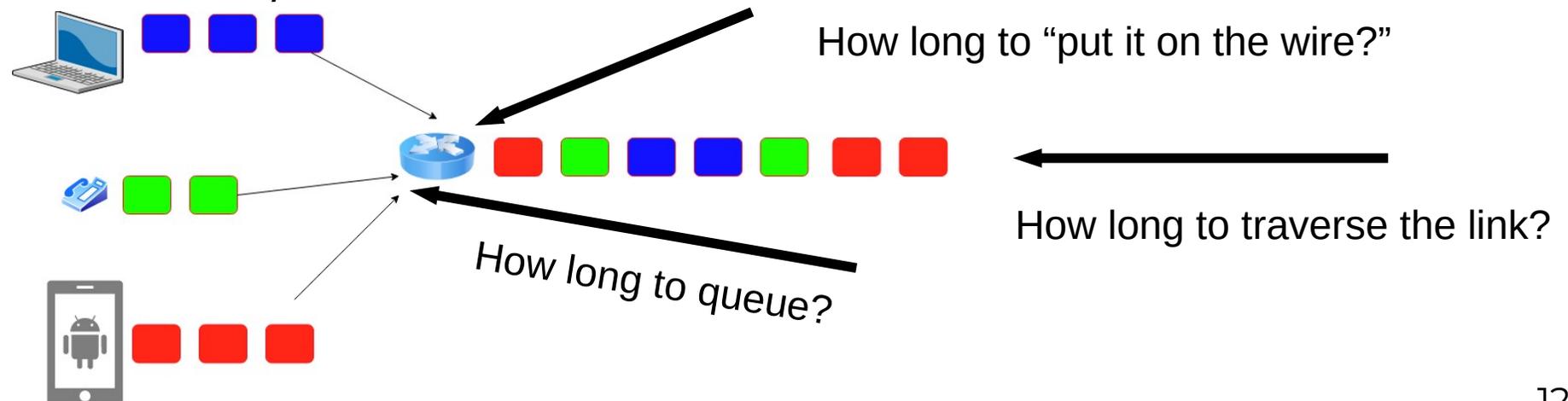
Performance - Bandwidth and Latency

- **Bandwidth = Size of the network pipe**
- **Latency = Delay in sending packets**
- **Throughput = How fast you can send data, function of both bandwidth and latency (and other things)**



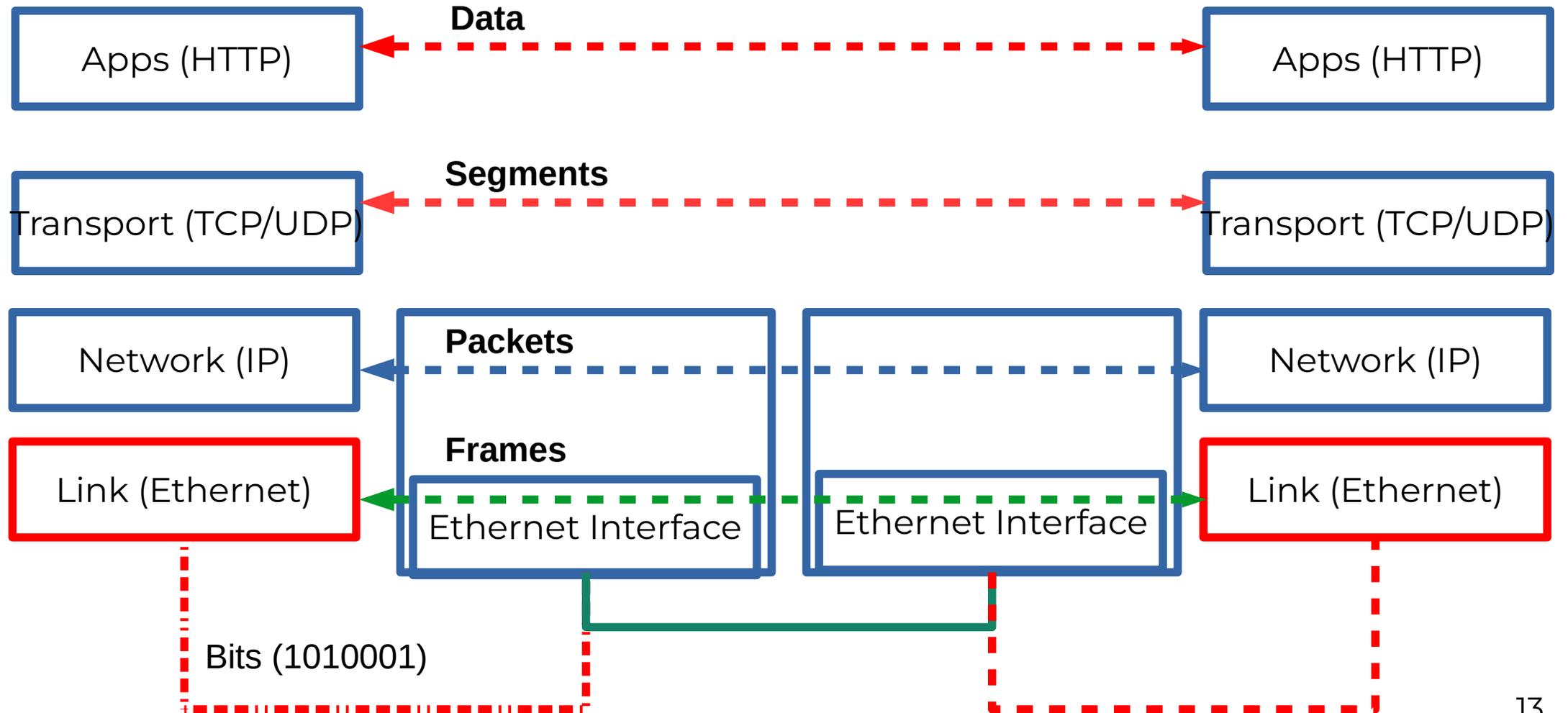
Performance - Latency

- Latency = Propagation Delay + Transmission Delay + Queuing Delay
- Propagation = Distance/Speed Of Light (in Copper or Fiber)
- Transmit = Size/Bandwidth

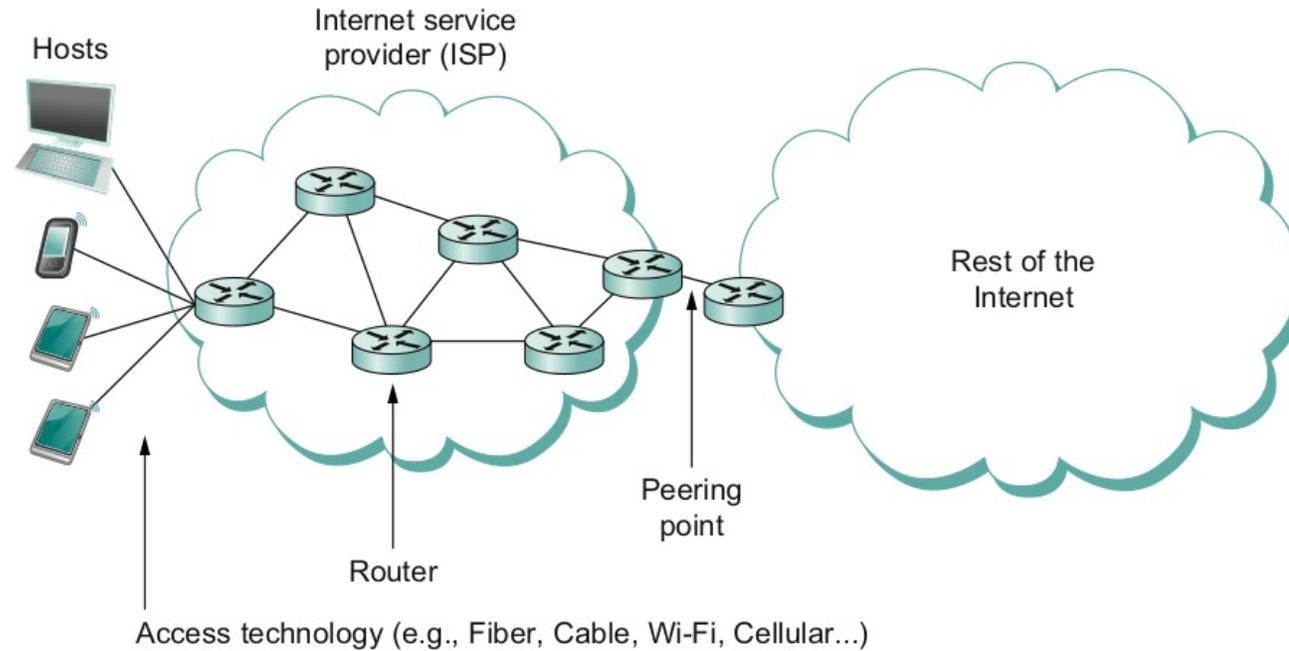




Link Layer Recap – How much work for a cat picture?



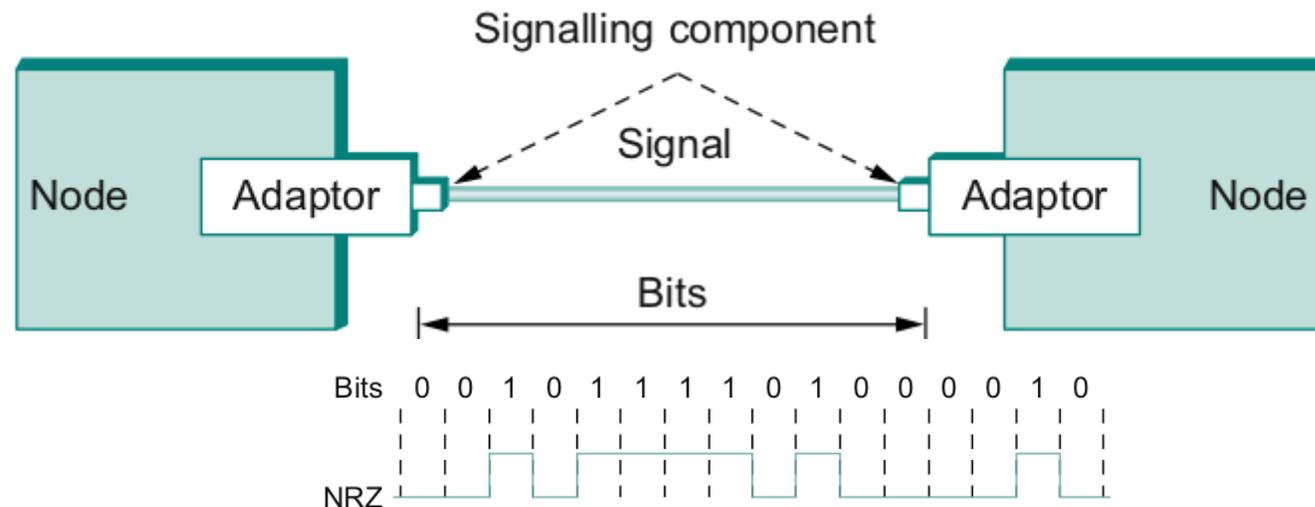
What does it take to create a link?



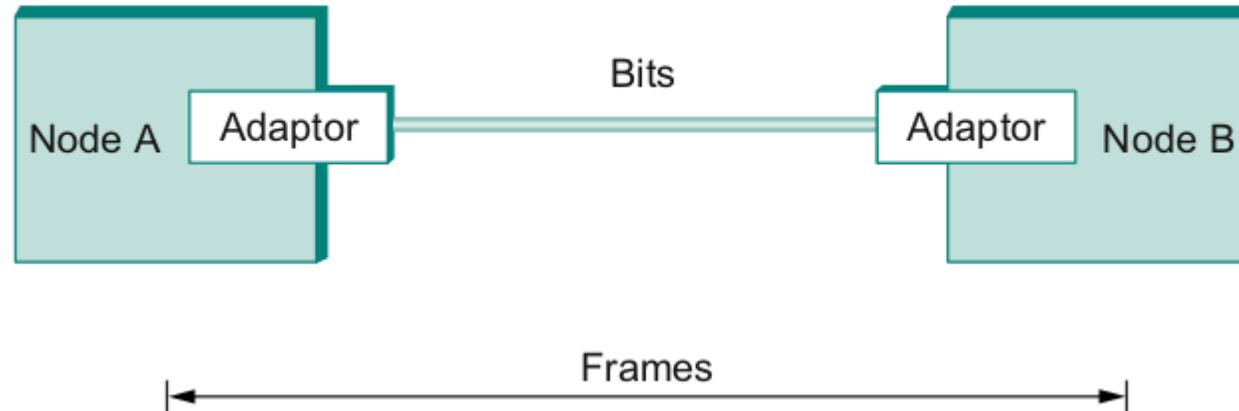
- Common abstractions
 - Why?

Packet to Low level Signals

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



Frames – bag of bits



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

Error Detection

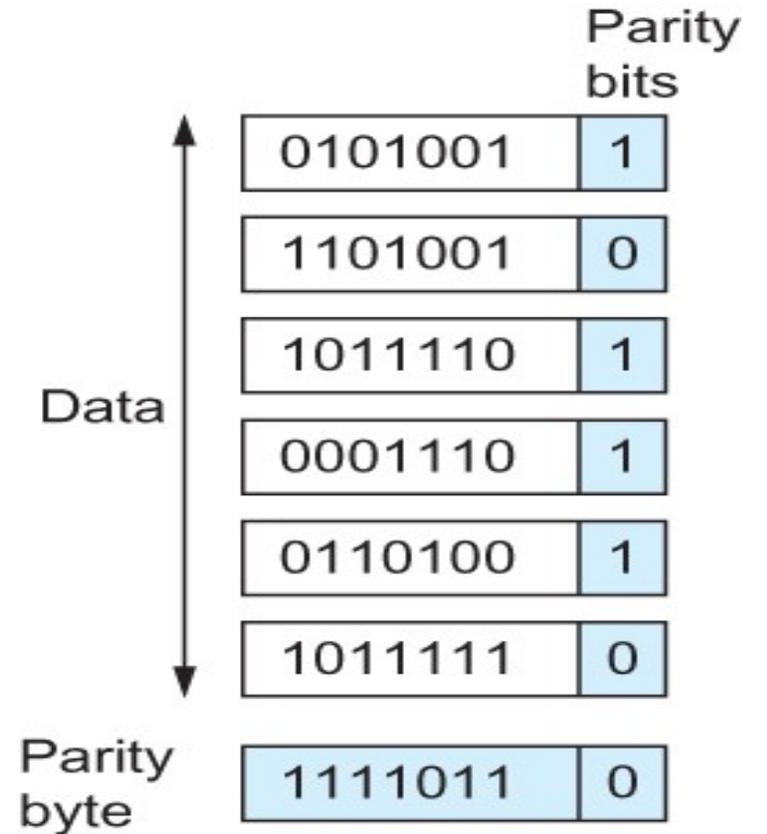
- Bit errors are introduced into frames
 - Because of electrical interference and thermal noises
- Detecting Error
- Correction Error
- Two approaches when the recipient detects an error
 - Notify the sender that the message was corrupted, so the sender can send again.
 - If the error is rare, then the 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



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

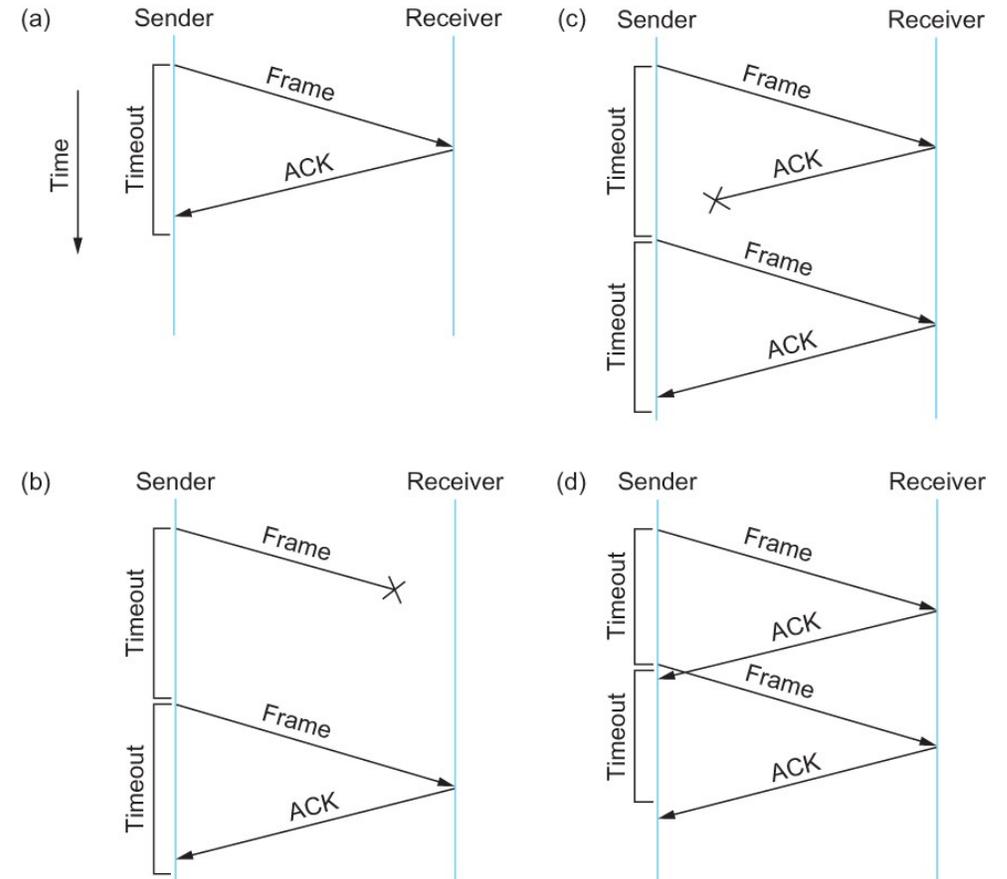
- If correct, all 1s

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?

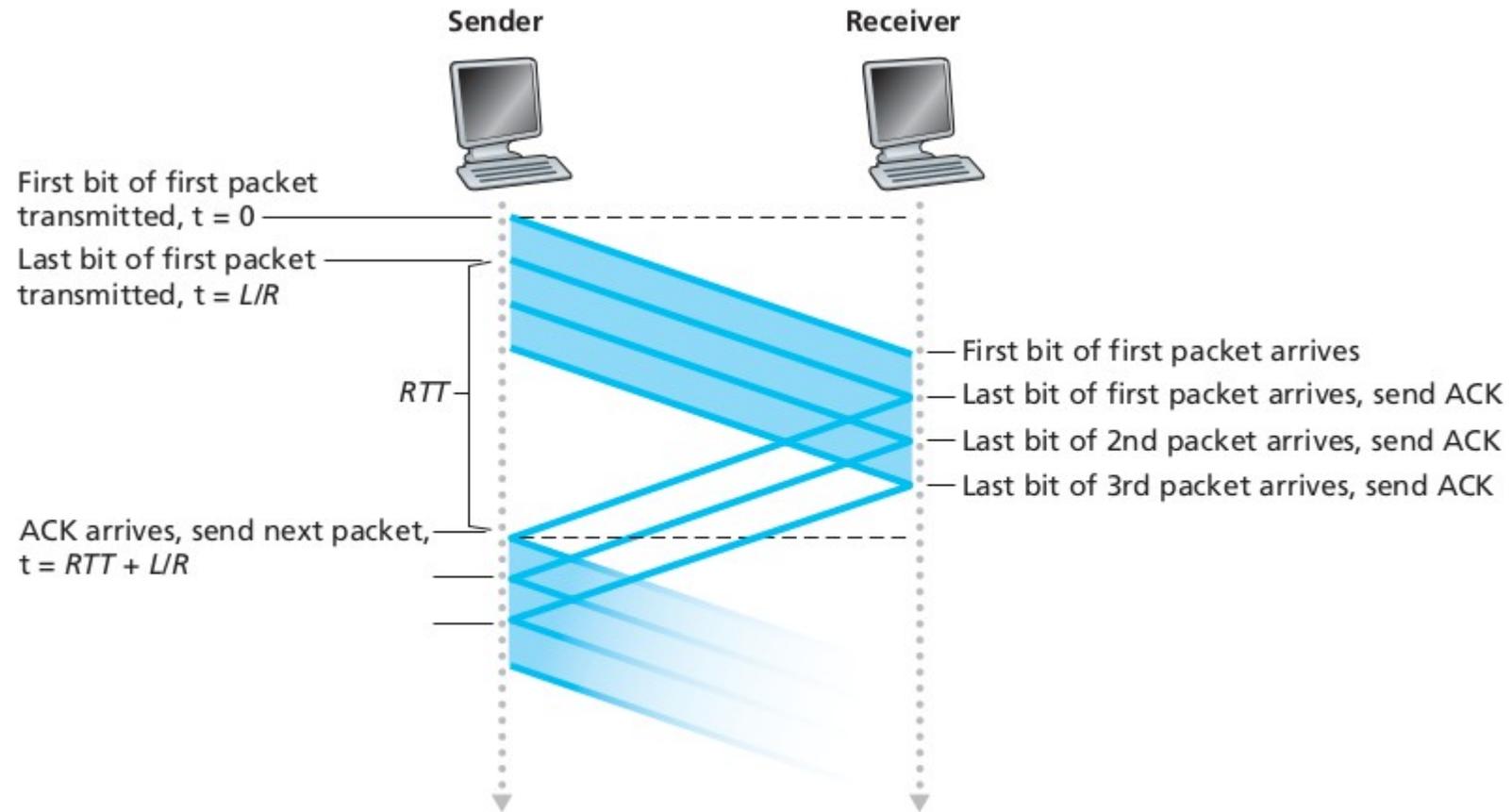
Stop and Wait

- 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**

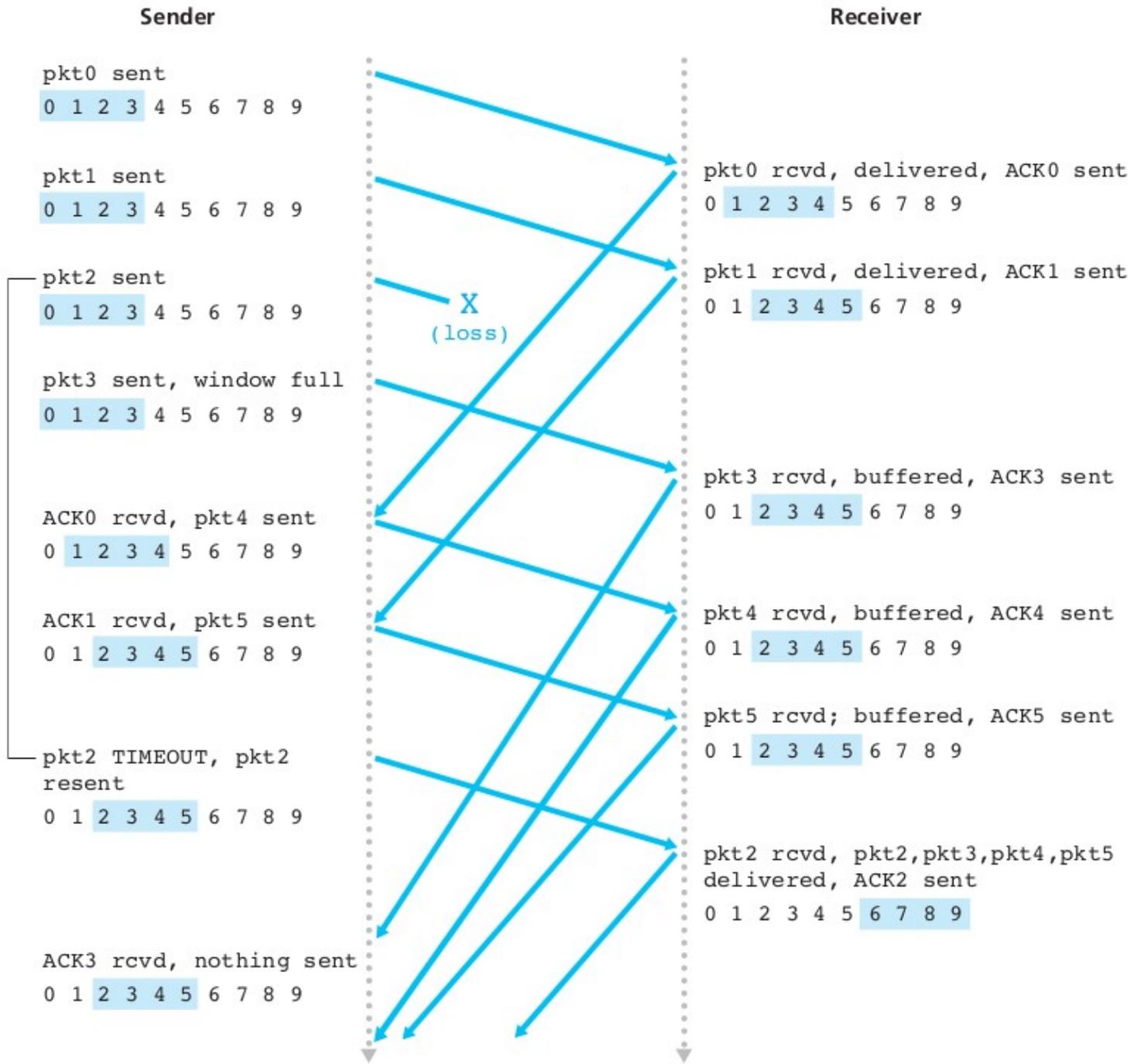


Sliding window to the rescue!

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



b. Pipelined operation

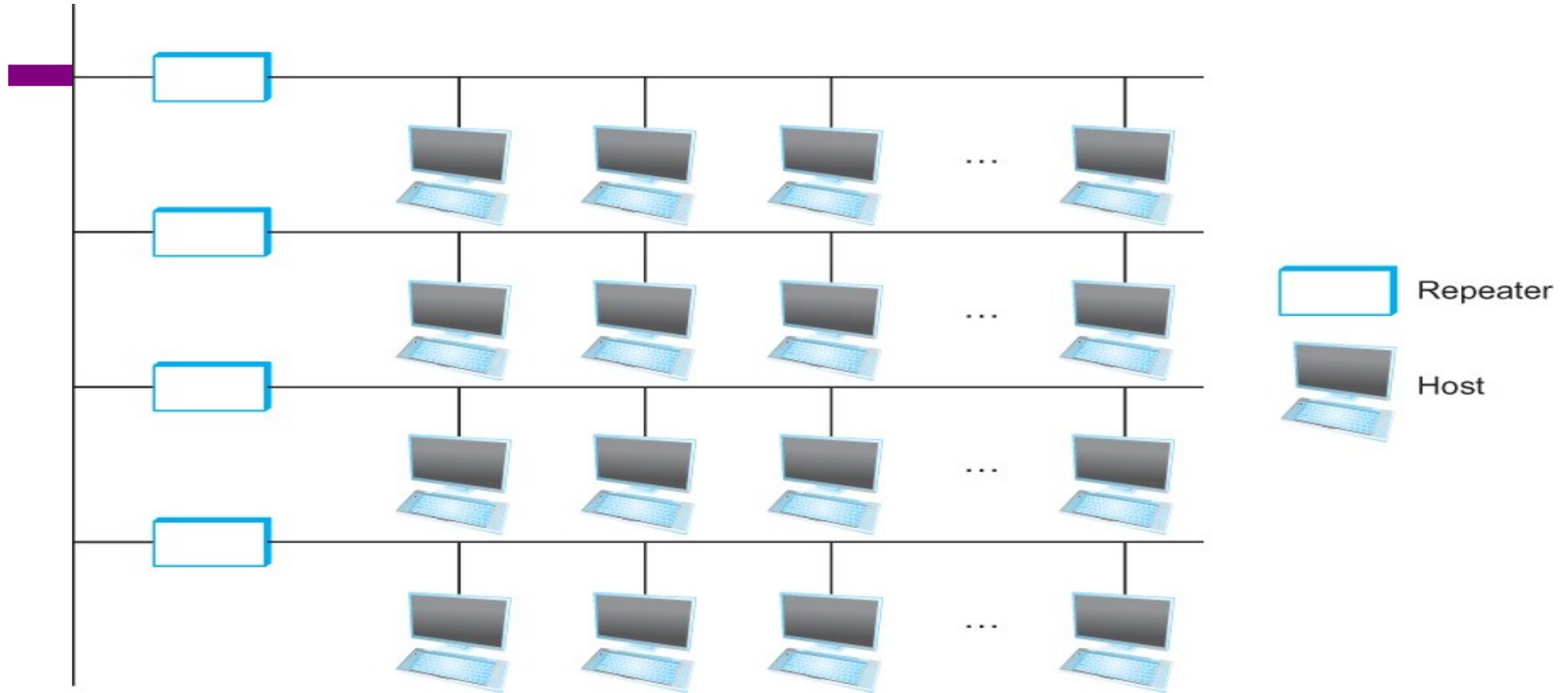


So far we connected two machines – how about more than two?



- 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?

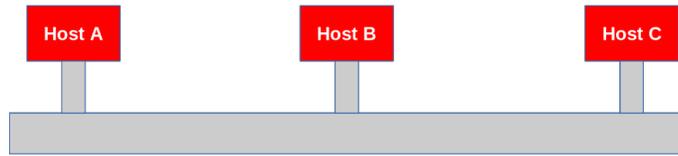
Ethernet



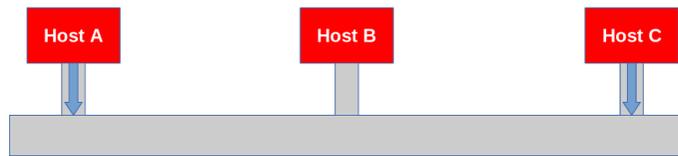
Ethernet repeater

CSMA/CD – Ethernet

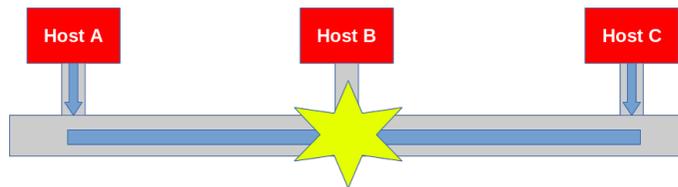
1) Carrier Sense



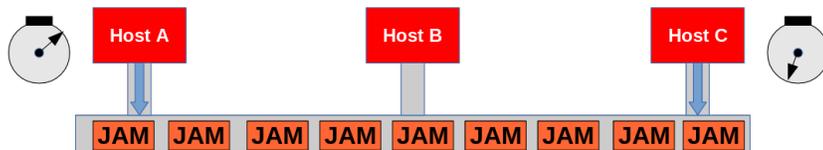
2) Multiple Access



3) Collision

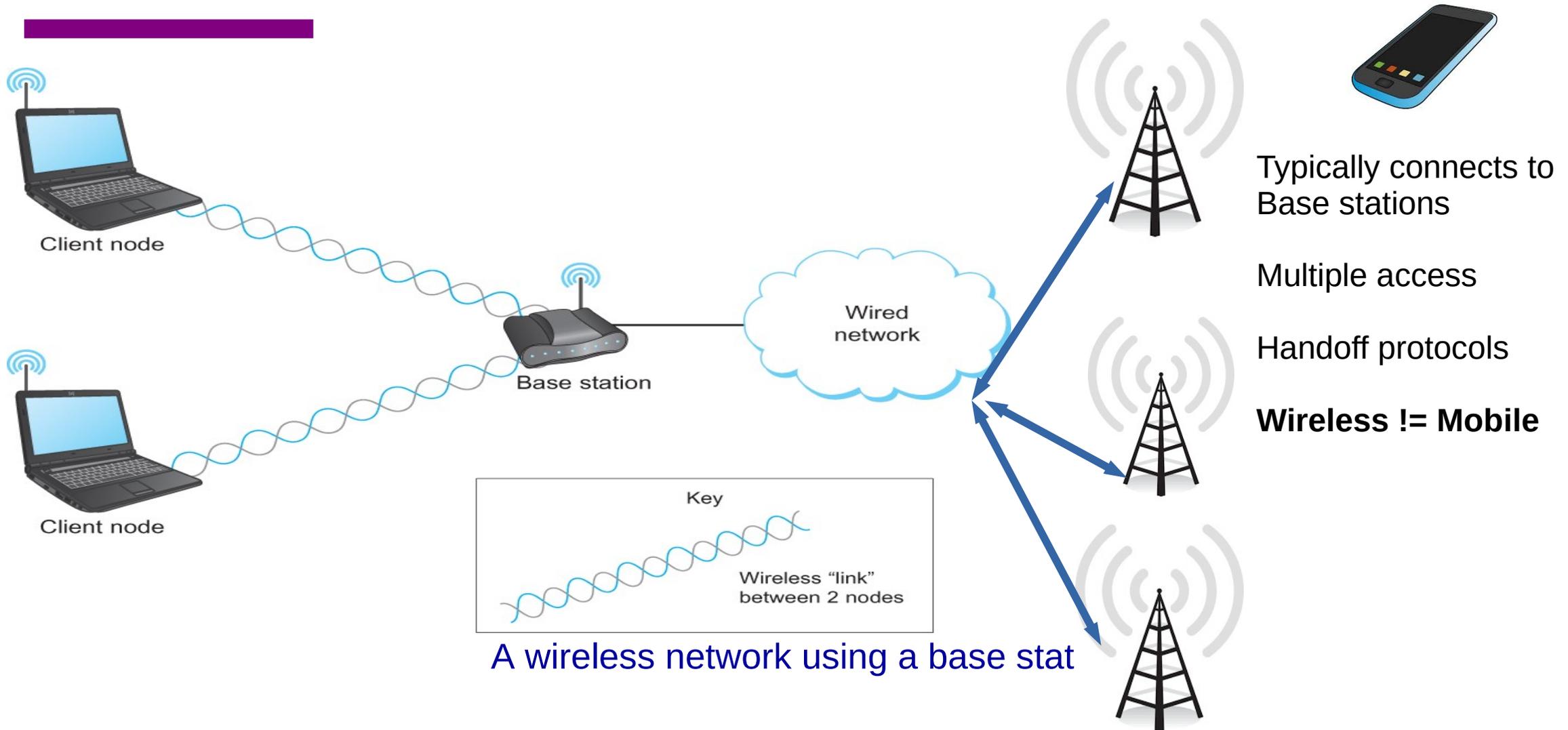


4) Collision Detection (Back off Algorithmus)



- CS – wait until idle
 - Channel idle – transmit
 - Channel busy – wait
- CD – listen while transmitting
 - No collision: transmission successful
 - Collision: abort, send jam signal (32bit special sequence)
- Wait random time
 - Try again
 - After m^{th} collision, $t = \text{random}(0, 2^m - 1)$,
 - Wait $t * 512$ bit times before retry

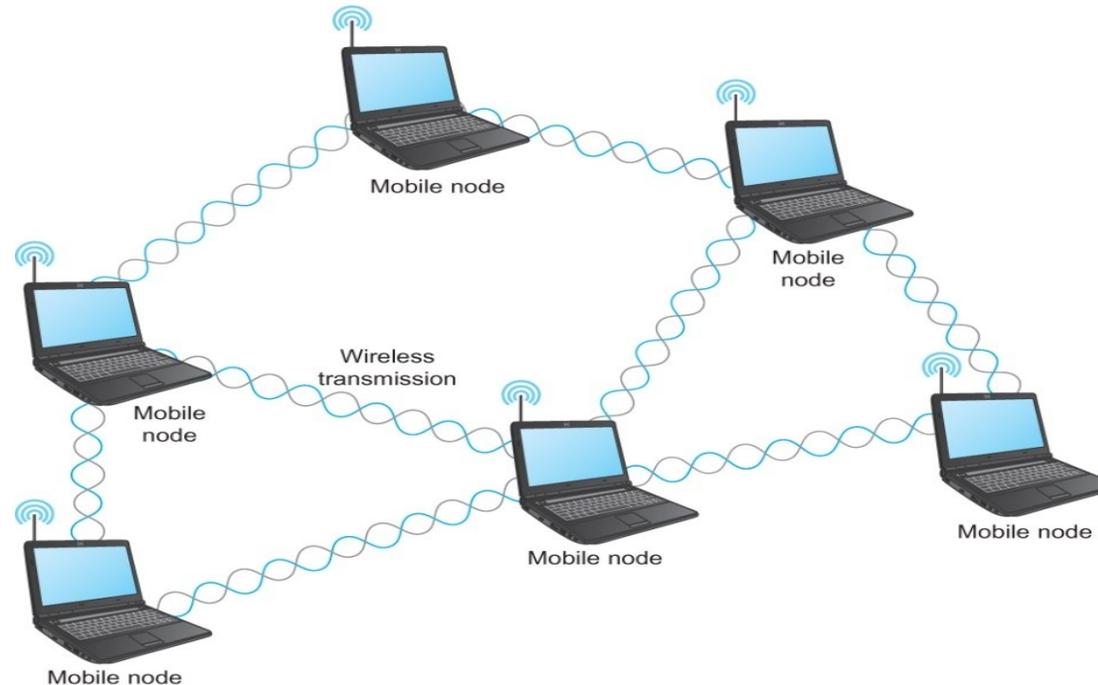
Wireless Links - Infrastructure



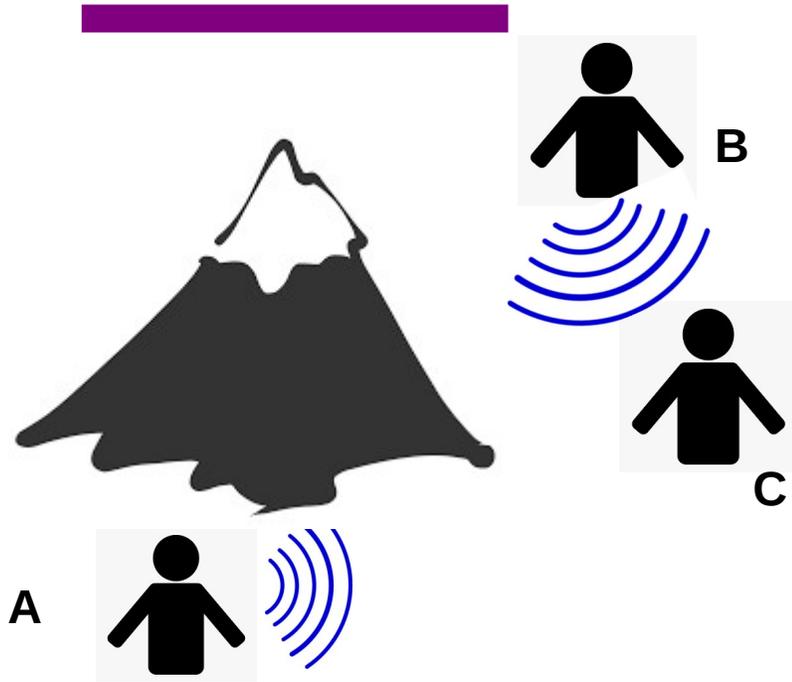
A wireless network using a base station

Wireless Links – Ad hoc

- Mesh or Ad-hoc network
 - Nodes are peers
 - Messages may be forwarded via a chain of peer nodes

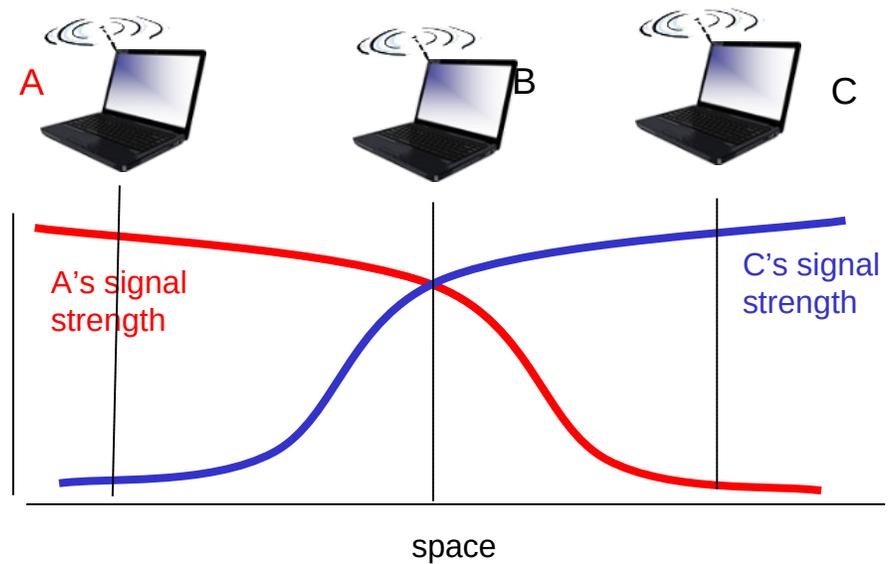


Wireless Links – problems



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

Hidden terminal Signal Fading



IEEE 802.11 MAC Protocol: CSMA/CA

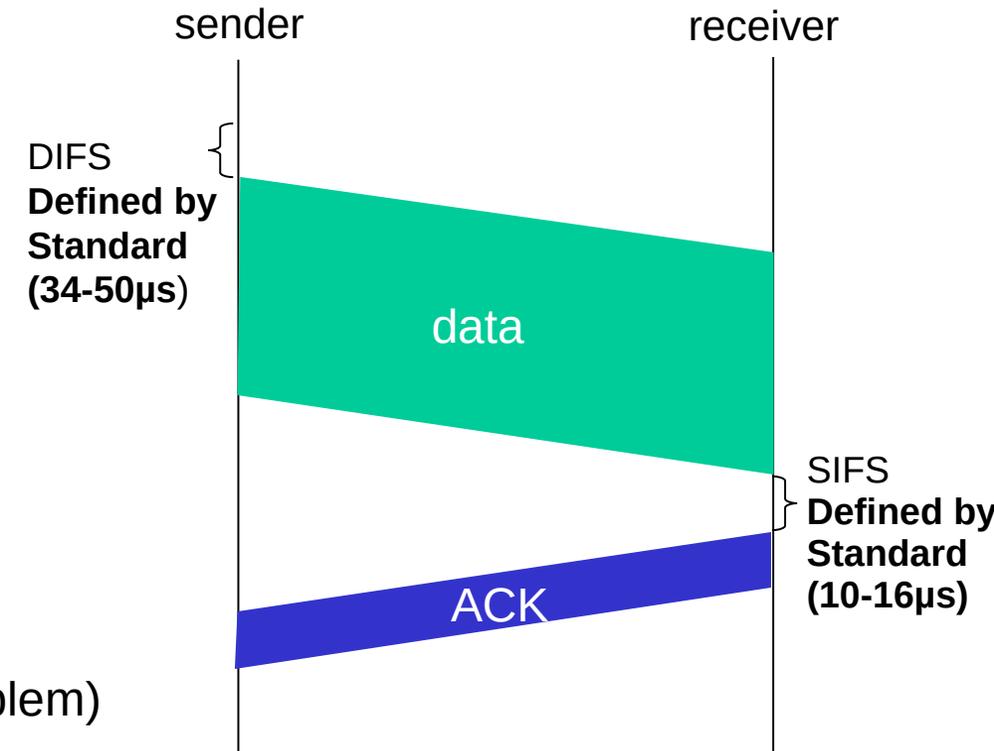
802.11 sender

- 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

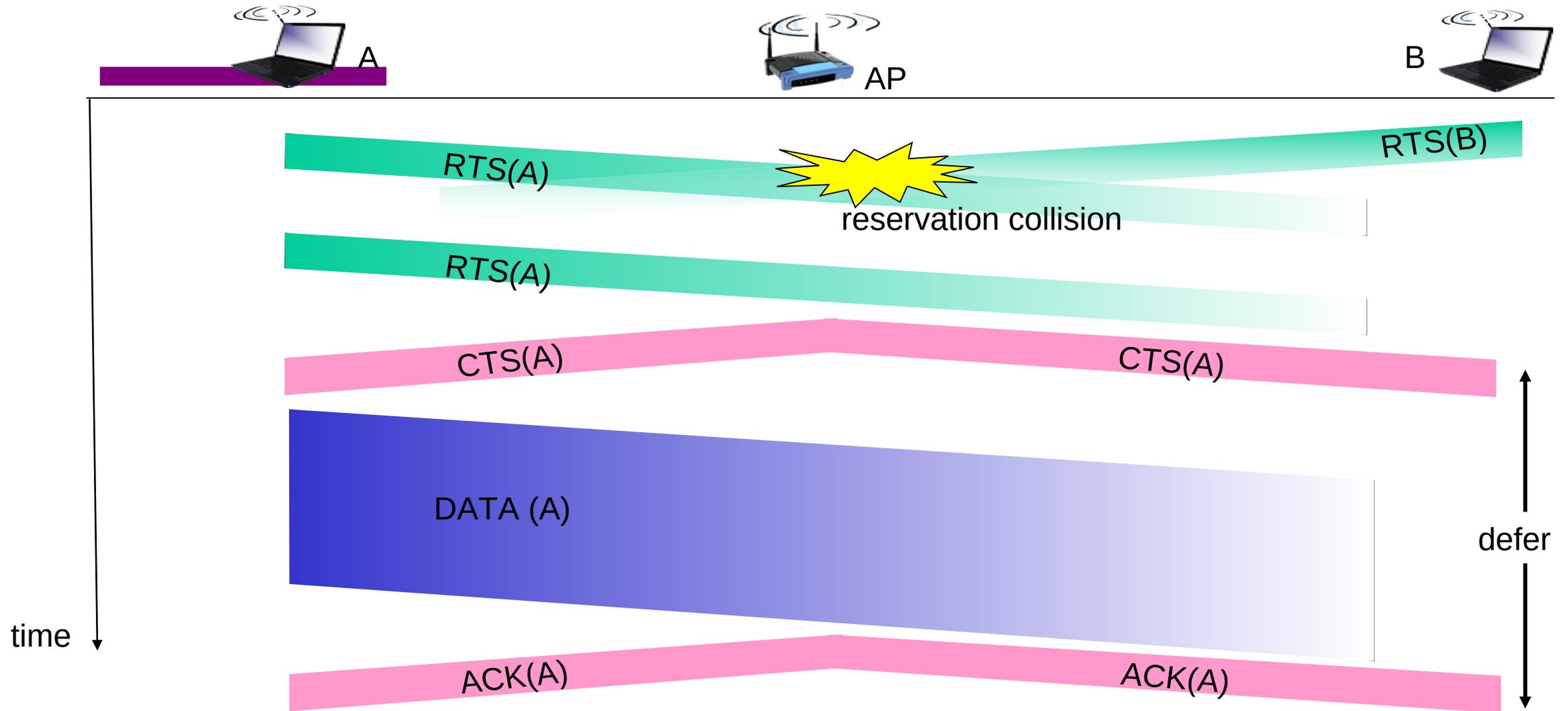
802.11 receiver

- if frame received OK
return ACK after **SIFS** (ACK needed due to hidden terminal problem)

$$\text{DIFS} = \text{SIFS} + (2 * \text{Slot time})$$



Collision Avoidance: RTS-CTS exchange





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

