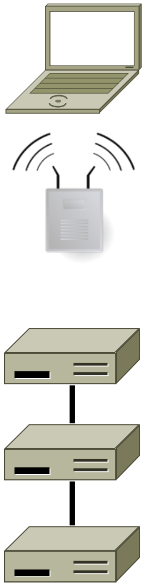
Topics (2)

3. Retransmissions
   – Handling loss
4. Multiple Access
   – Classic Ethernet, 802.11
5. Switching
   – Modern Ethernet



Topic

- Two strategies to handle errors:

1. Detect errors and retransmit frame (Automatic Repeat reQuest, ARQ)

2. Correct errors with an error correcting code

← Done this

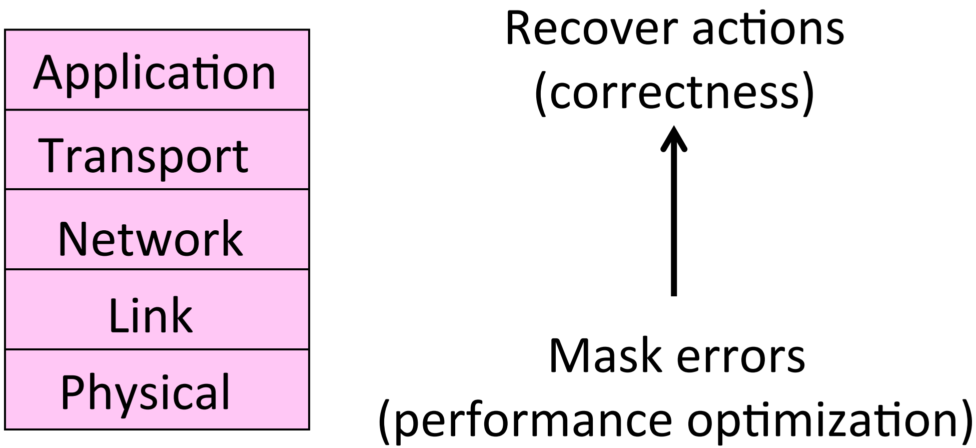
Context on Reliability

- Where in the stack should we place reliability functions?

Application
Transport
Network
Link
Physical

Context on Reliability (2)

- Everywhere! It is a key issue
  - Different layers contribute differently



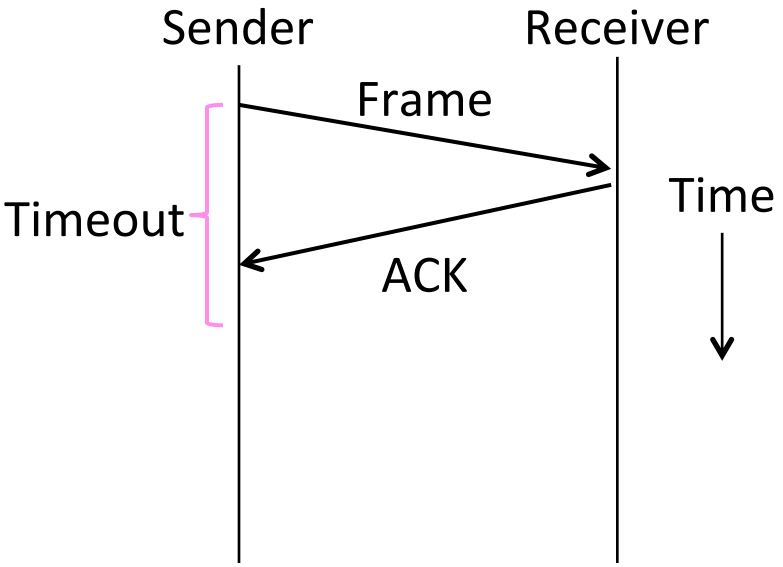
ARQ

- ARQ often used when errors are common or must be corrected
  - E.g., WiFi, and TCP (later)

- Rules at sender and receiver:
  - Receiver automatically acknowledges correct frames with an ACK
  - Sender automatically resends after a timeout, until an ACK is received

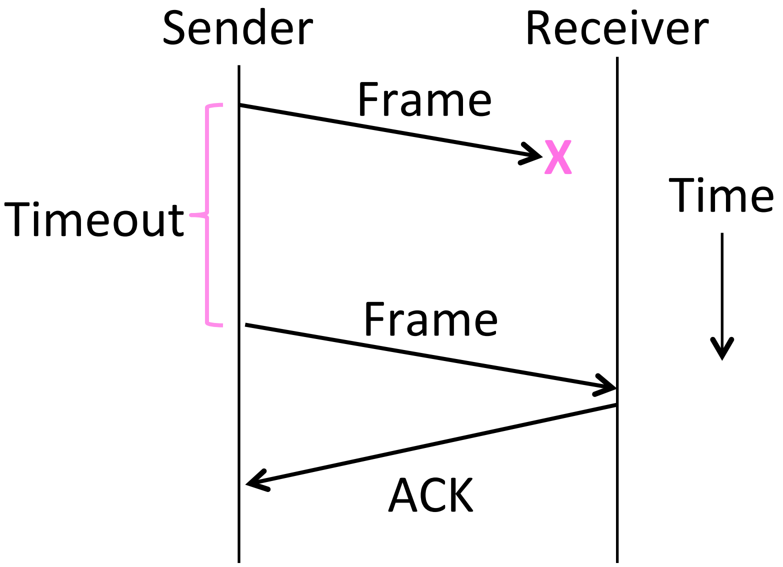
ARQ (2)

- Normal operation (no loss)



ARQ (3)

- Loss and retransmission



So What's Tricky About ARQ?

- Two non-trivial issues:
  - How long to set the timeout? »
  - How to avoid accepting duplicate frames as new frames »
- Want performance in the common case and correctness always

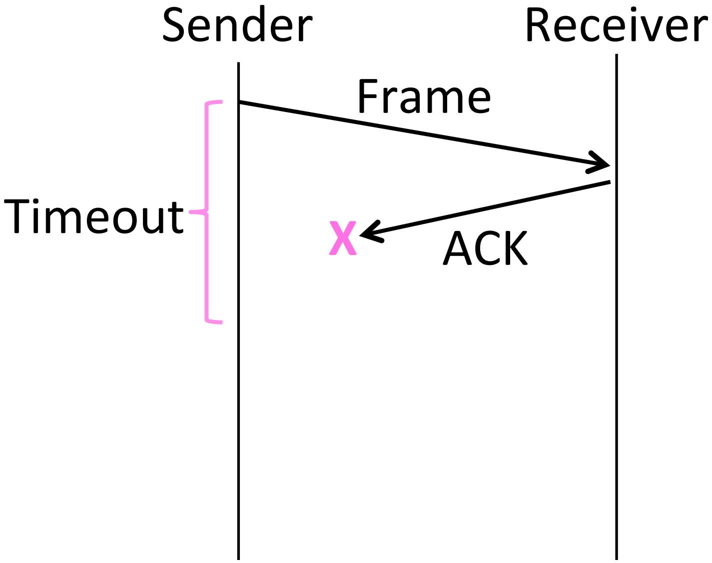
Timeouts

- Timeout should be:
  - Not too big (link goes idle)
  - Not too small (spurious resend)

- Fairly easy on a LAN
  - Clear worst case, little variation
- Fairly difficult over the Internet
  - Much variation, no obvious bound
  - We'll revisit this with TCP (later)

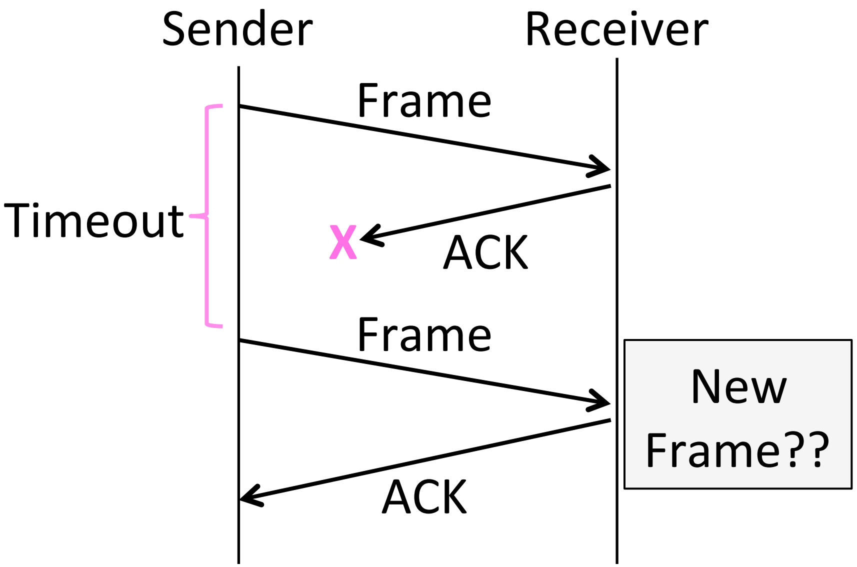
Duplicates

- What happens if an ACK is lost?



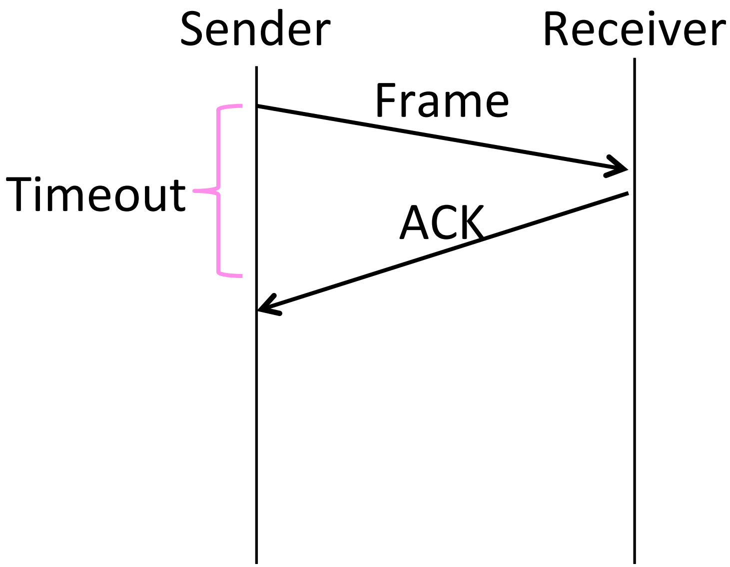
Duplicates (2)

- What happens if an ACK is lost?



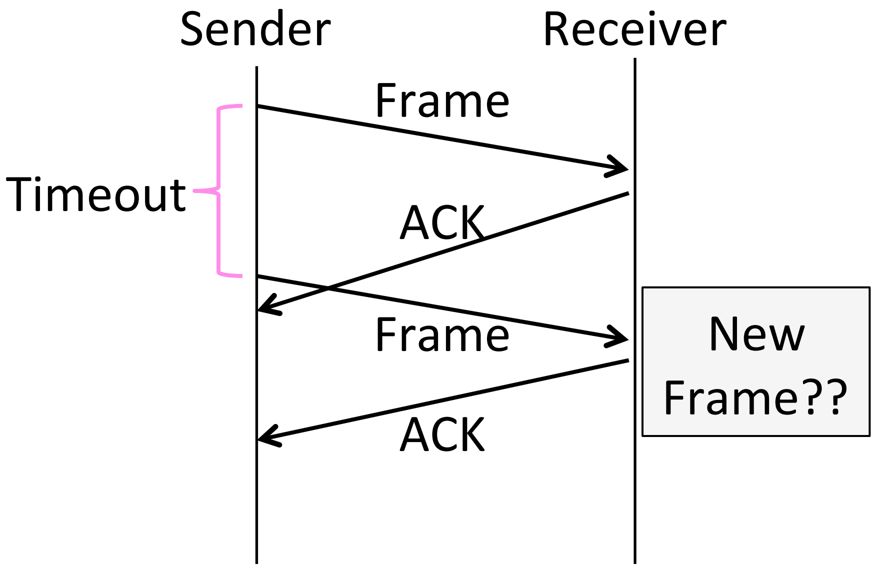
Duplicates (3)

- Or the timeout is early?



Duplicates (4)

- Or the timeout is early?

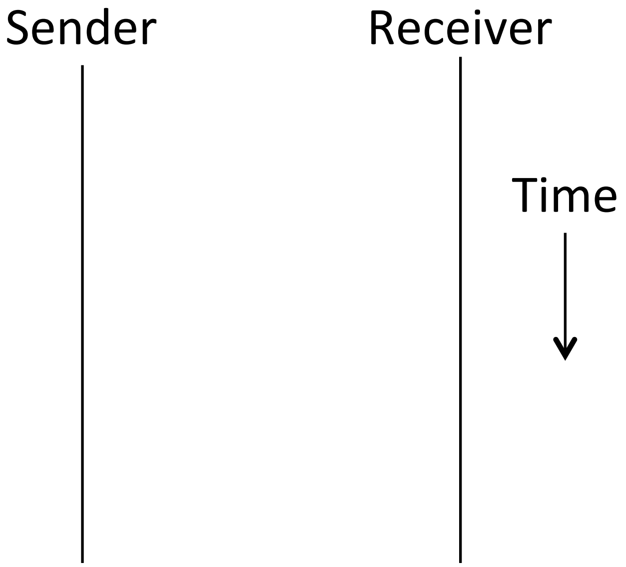


Sequence Numbers

- Frames and ACKs must both carry sequence numbers for correctness
- To distinguish the current frame from the next one, a single bit (two numbers) is sufficient
  - Called <u>Stop-and-Wait</u>

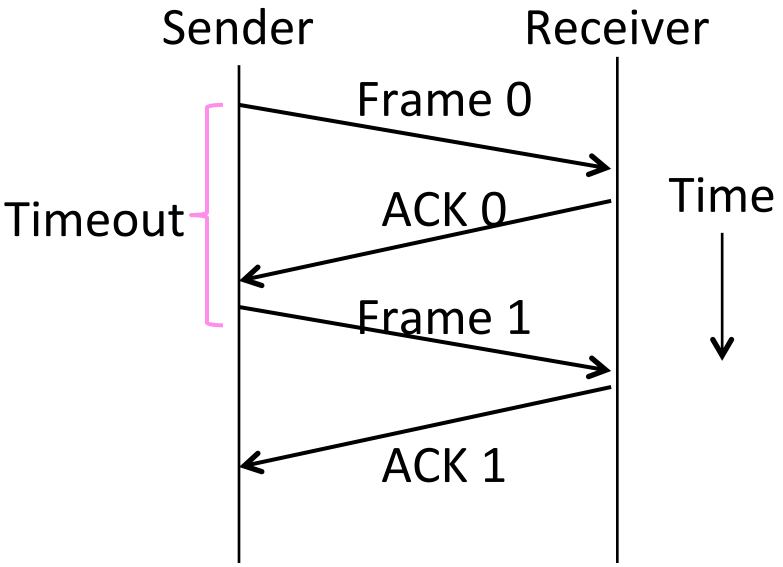
Stop-and-Wait

- In the normal case:



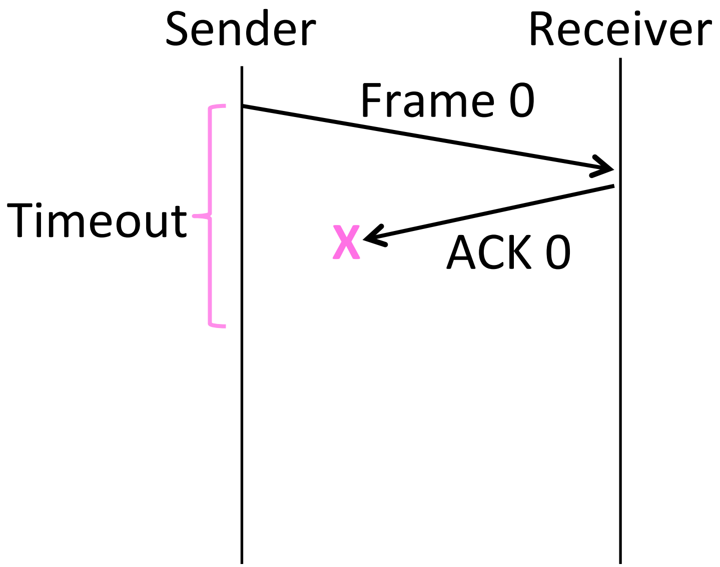
Stop-and-Wait (2)

- In the normal case:



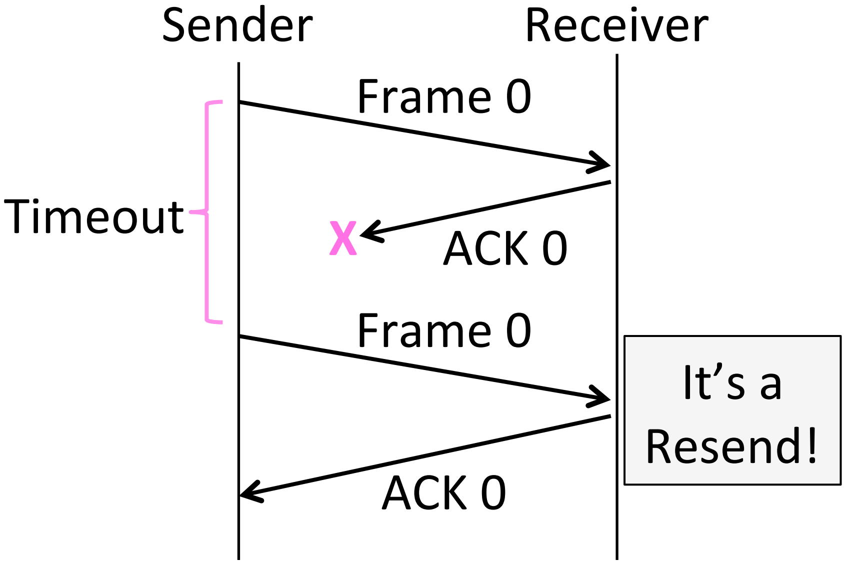
Stop-and-Wait (3)

- With ACK loss:



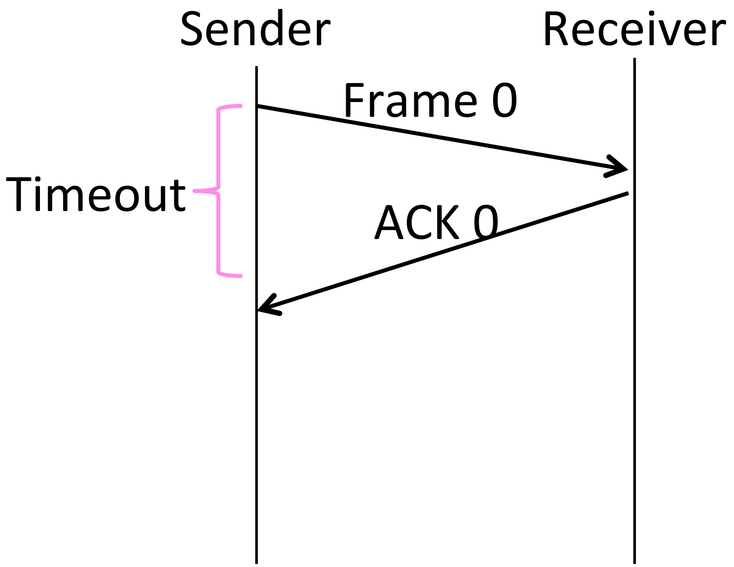
Stop-and-Wait (4)

- With ACK loss:



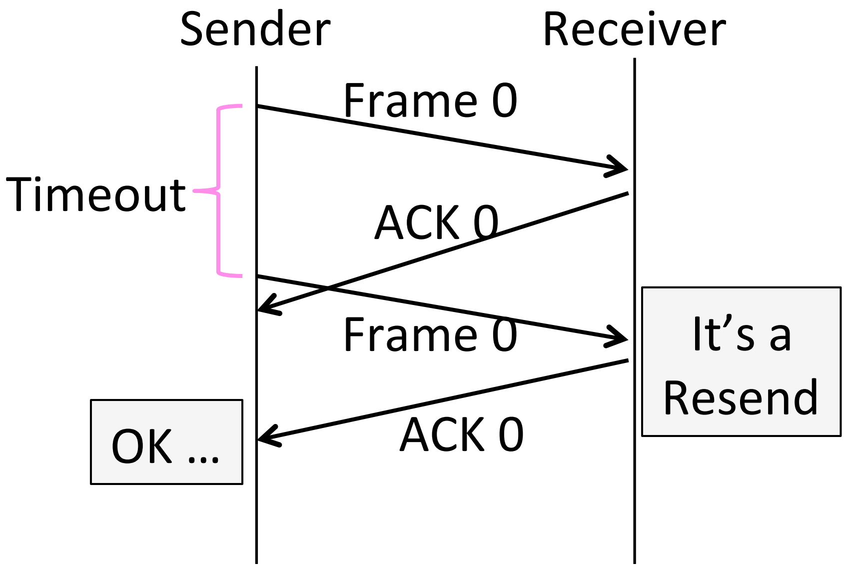
Stop-and-Wait (5)

- With early timeout:



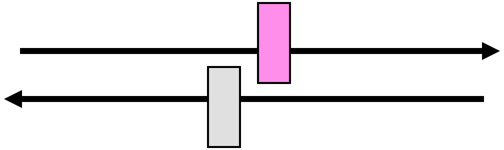
Stop-and-Wait (6)

- With early timeout:



Limitation of Stop-and-Wait

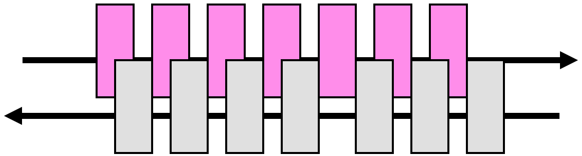
- It allows only a single frame to be outstanding from the sender:
  - Good for LAN, not efficient for high BD



- Ex: R=1 Mbps, D = 50 ms
  - How many frames/sec? If R=10 Mbps?

Sliding Window

- Generalization of stop-and-wait
  - Allows W frames to be outstanding
  - Can send W frames per RTT (=2D)



  - Various options for numbering frames/ACKs and handling loss
    - Will look at along with TCP (later)

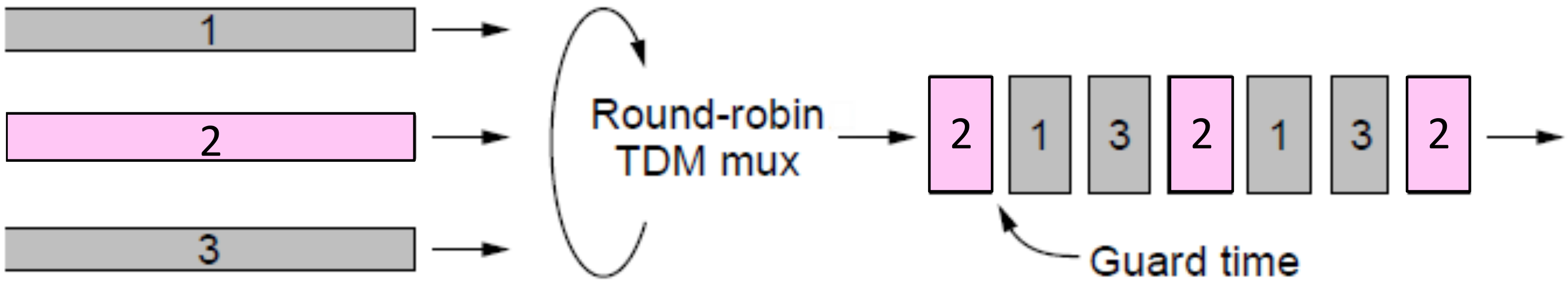
Multiple devices?

- Multiplexing is the network word for the sharing of a resource

- Classic scenario is sharing a link among different users
  - Time Division Multiplexing (TDM) »
  - Frequency Division Multiplexing (FDM) »

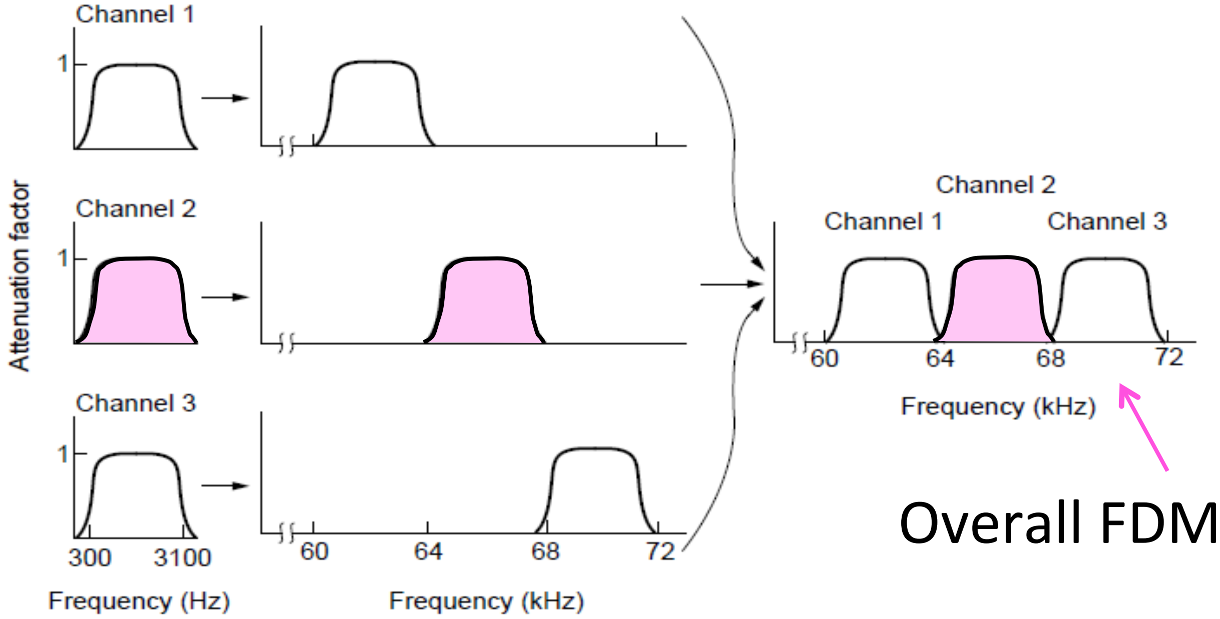
Time Division Multiplexing (TDM)

- Users take turns on a fixed schedule



Frequency Division Multiplexing (FDM)

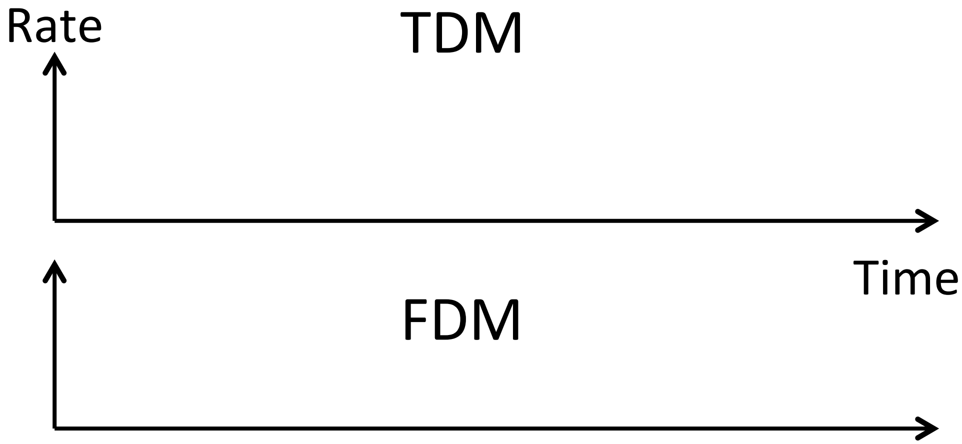
- Put different users on different frequency bands



Overall FDM channel

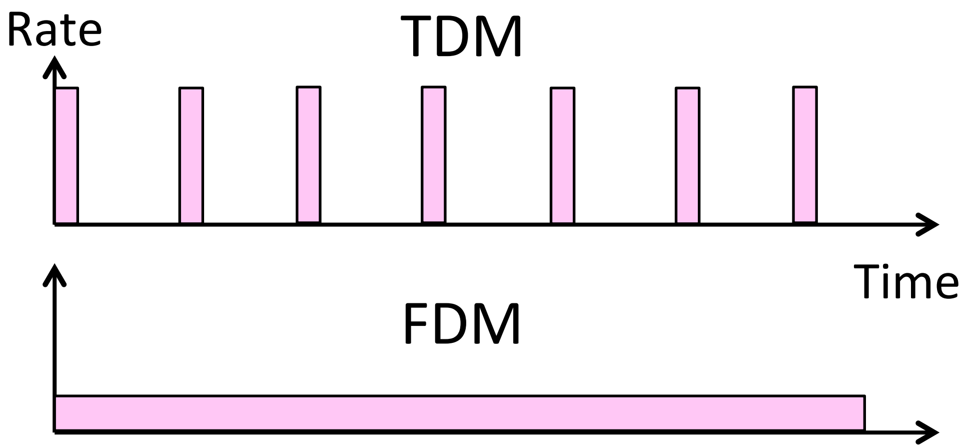
TDM versus FDM

- In TDM a user sends at a high rate a fraction of the time; in FDM, a user sends at a low rate all the time



TDM versus FDM (2)

- In TDM a user sends at a high rate a fraction of the time; in FDM, a user sends at a low rate all the time

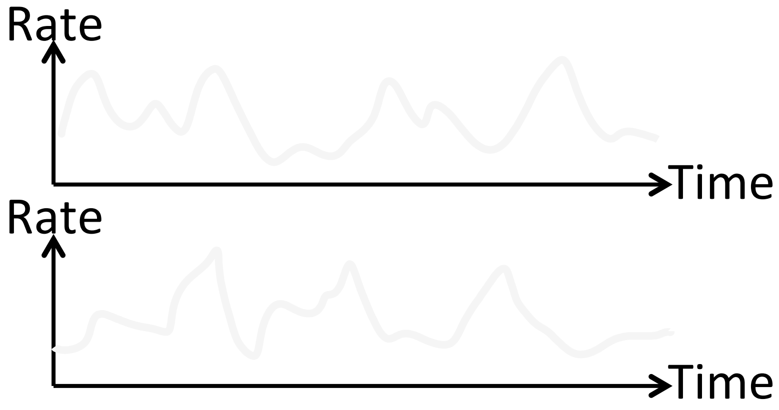


TDM/FDM Usage

- Statically divide a resource
  - Suited for continuous traffic, fixed number of users
- Widely used in telecommunications
  - TV and radio stations (FDM)
  - GSM (2G cellular) allocates calls using TDM within FDM

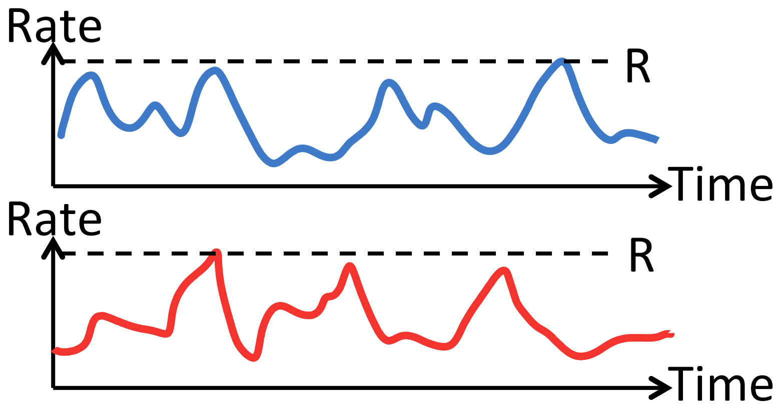
Multiplexing Network Traffic

- Network traffic is <u>bursty</u>
  - ON/OFF sources
  - Load varies greatly over time



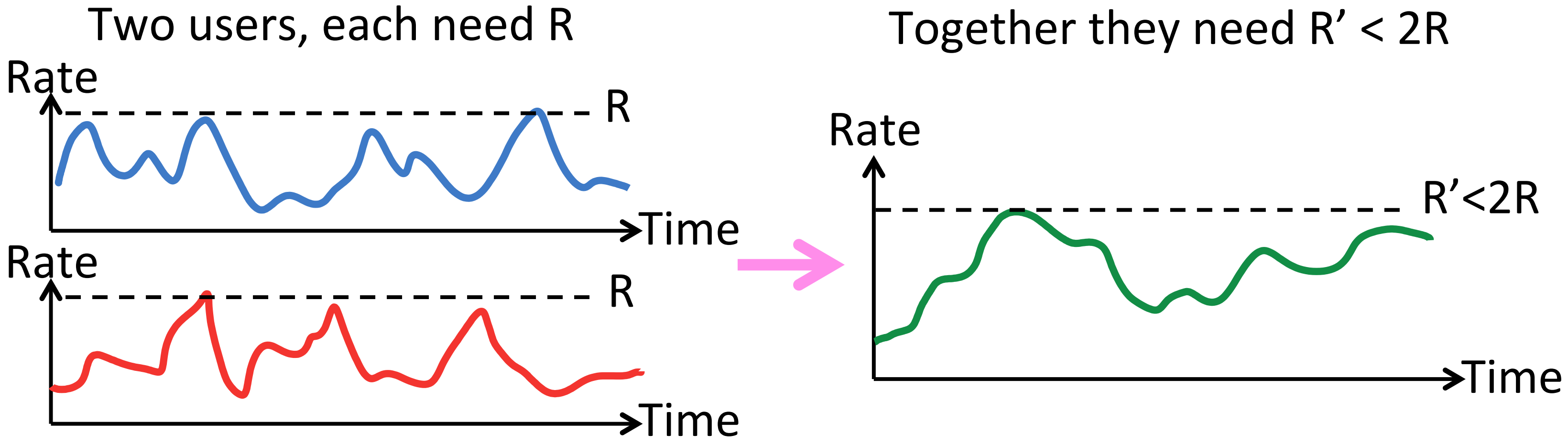
Multiplexing Network Traffic (2)

- Network traffic is <u>bursty</u>
  - Inefficient to always allocate user their ON needs with TDM/FDM



Multiplexing Network Traffic (3)

- Multiple access schemes multiplex users according to their demands – for gains of statistical multiplexing



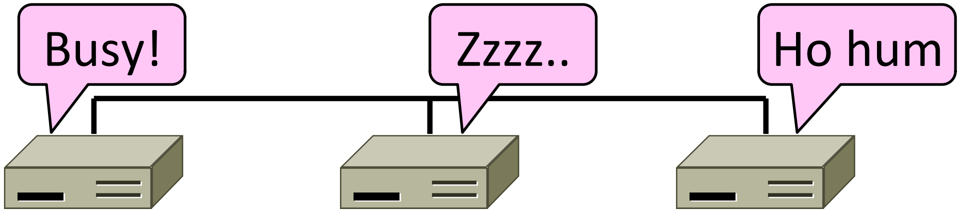
Multiple Access

- We will look at two kinds of multiple access protocols

1. Randomized. Nodes randomize their resource access attempts
   – Good for low load situations
2. Contention-free. Nodes order their resource access attempts
   – Good for high load or guaranteed quality of service situations

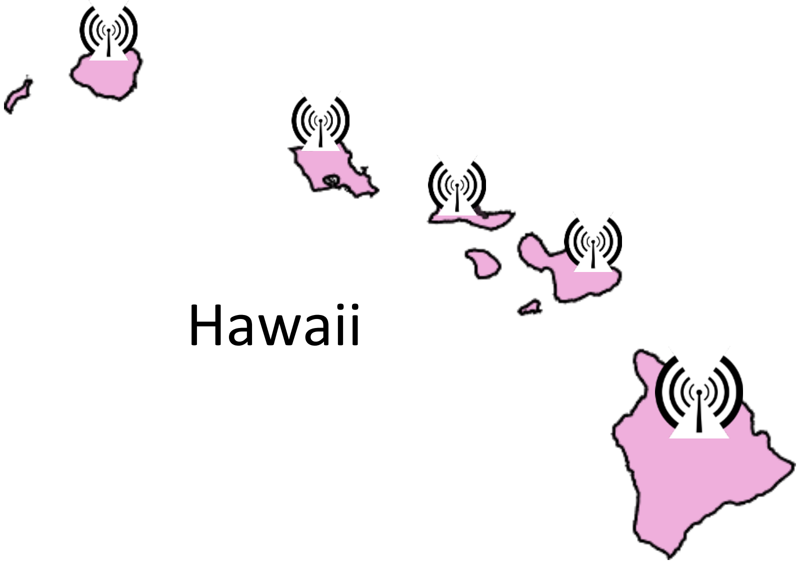
Random MAC

- We will explore random multiple access control (MAC) protocols
  - This is the basis for classic Ethernet
  - Remember: data traffic is bursty



ALOHA Network

- Seminal computer network connecting the Hawaiian islands in the late 1960s
  - When should nodes send?
  - A new protocol was devised by Norm Abramson …

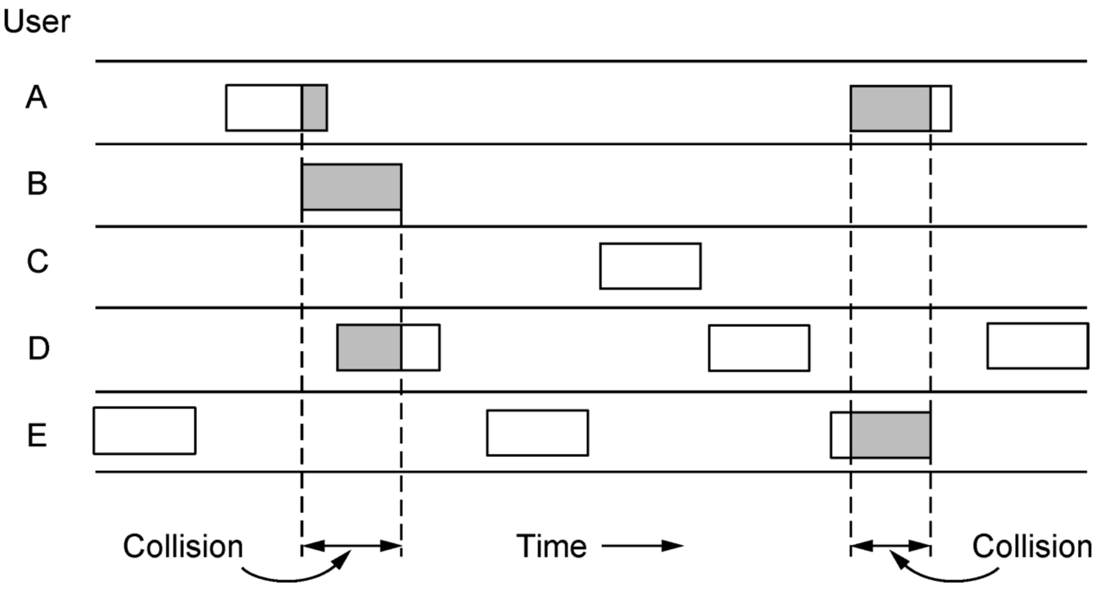


ALOHA Protocol

- Simple idea:
  - Node just sends when it has traffic.
  - If there was a collision (no ACK received) then wait a random time and resend
- That’s it!

ALOHA Protocol (2)

- Some frames will be lost, but many may get through…
- Good idea?



ALOHA Protocol (3)

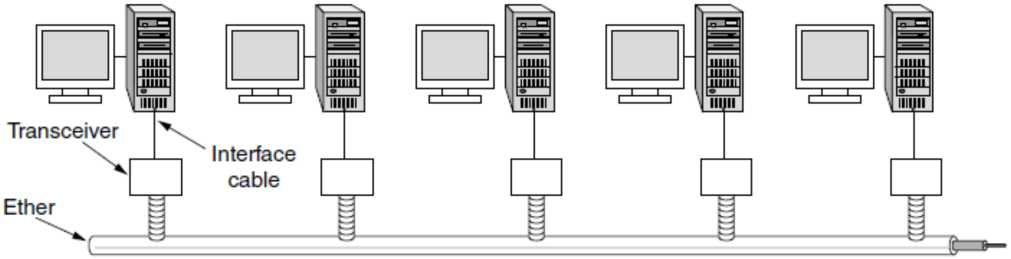
- Simple, decentralized protocol that works well under low load!

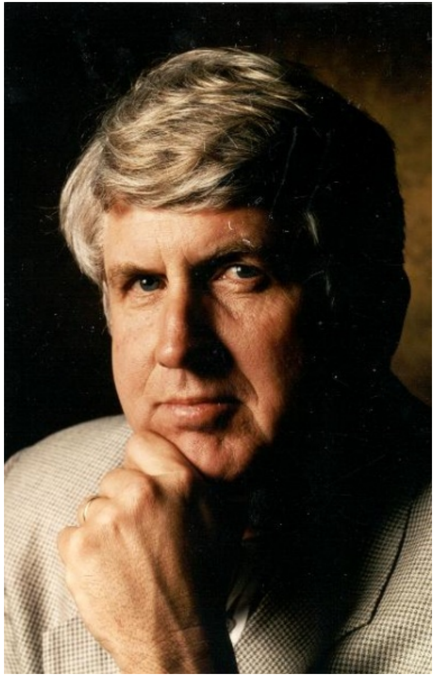
- Not efficient under high load
  - Analysis shows at most 18% efficiency
  - Improvement: divide time into slots and efficiency goes up to 36%

- We'll look at other improvements

Classic Ethernet

- ALOHA inspired Bob Metcalfe to invent Ethernet for LANs in 1973
  - Nodes share 10 Mbps coaxial cable
  - Hugely popular in 1980s, 1990s





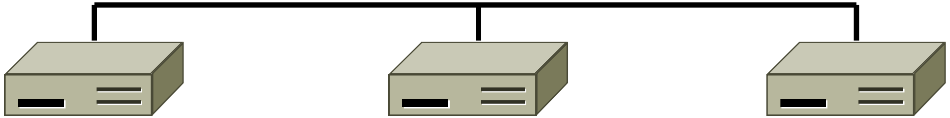
: © 2009 IEEE

CSMA (Carrier Sense Multiple Access)

- Improve ALOHA by listening for activity before we send (Doh!)
  - Can do easily with wires, not wireless
- So does this eliminate collisions?
  - Why or why not?

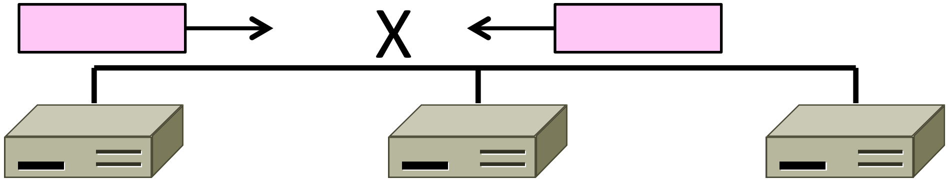
CSMA (2)

- Still possible to listen and hear nothing when another node is sending because of delay



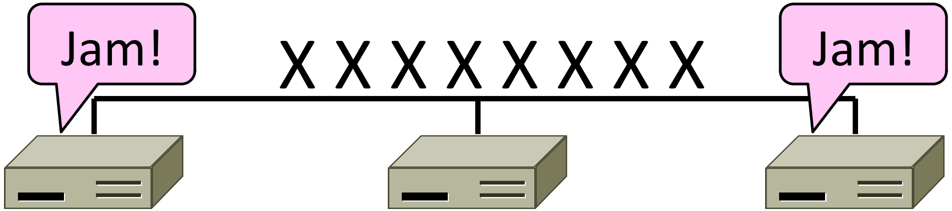
CSMA (3)

- CSMA is a good defense against collisions only when BD is small



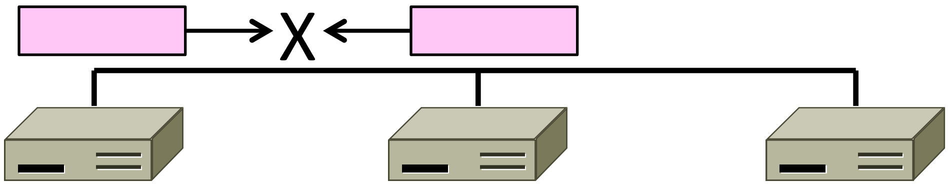
CSMA/CD (with Collision Detection)

- Can reduce the cost of collisions by detecting them and aborting (Jam) the rest of the frame time
  - Again, we can do this with wires



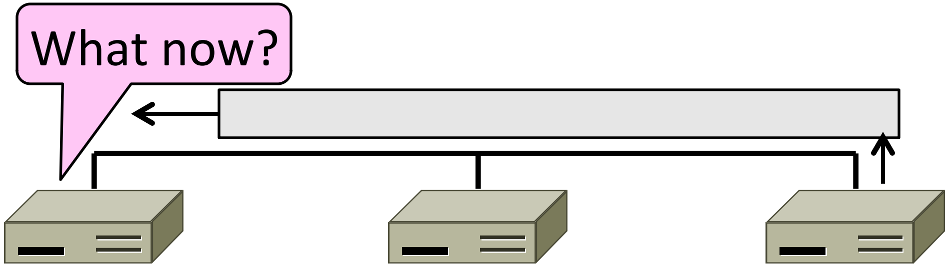
CSMA/CD Complications

- Want everyone who collides to know that it happened
  - Time window in which a node may hear of a collision is 2D seconds



CSMA "Persistence"

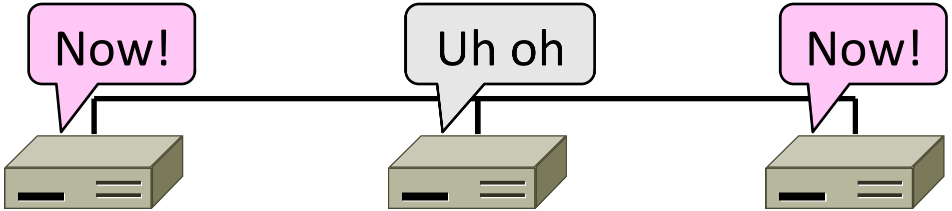
- What should a node do if another node is sending?



- Idea: Wait until it is done, and send

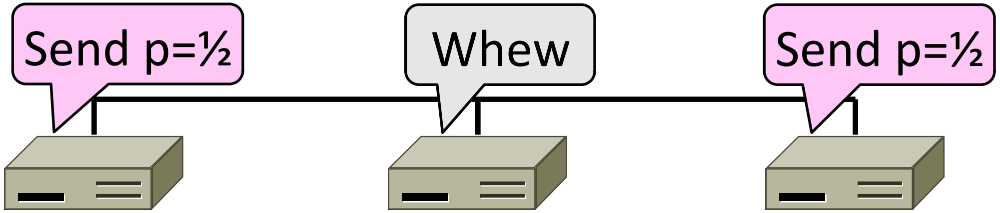
CSMA "Persistence" (2)

- Problem is that multiple waiting nodes will queue up then collide
  - More load, more of a problem



CSMA "Persistence" (3)

- Intuition for a better solution
  - If there are N queued senders, we want each to send next with probability 1/N

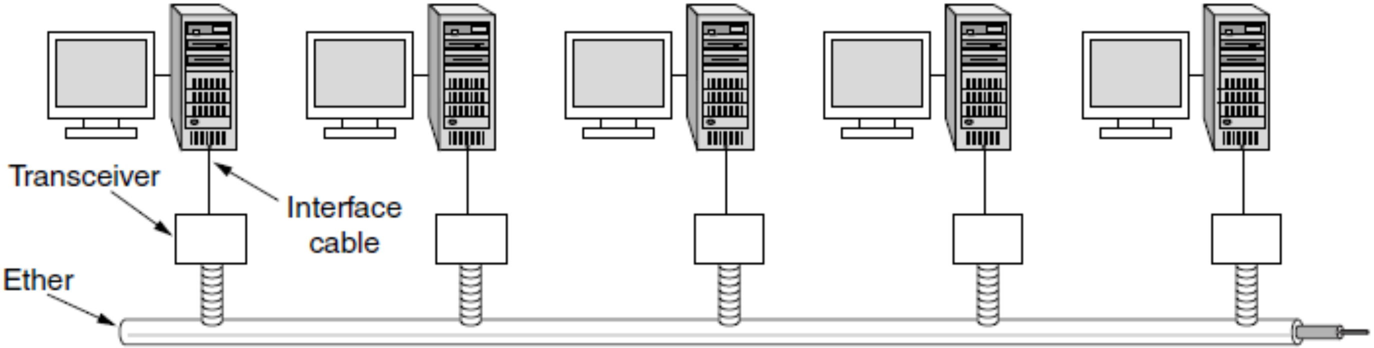


Binary Exponential Backoff (BEB)

- Cleverly estimates the probability
  - 1st collision, wait 0 or 1 frame times
  - 2nd collision, wait from 0 to 3 times
  - 3rd collision, wait from 0 to 7 times …
- BEB doubles interval for each successive collision
  - Quickly gets large enough to work
  - Very efficient in practice

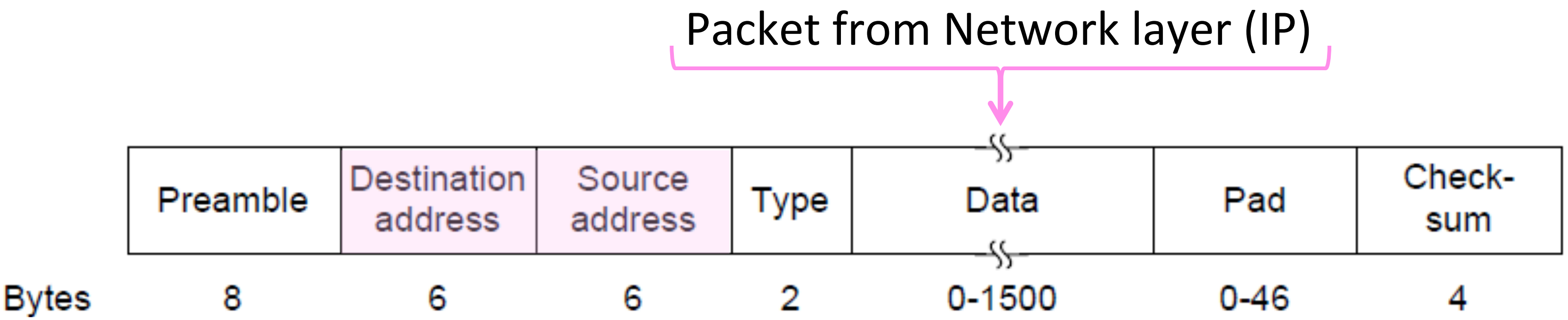
Classic Ethernet, or IEEE 802.3

- Most popular LAN of the 1980s, 1990s
  - 10 Mbps over shared coaxial cable, with baseband signals
  - Multiple access with "1-persistent CSMA/CD with BEB"



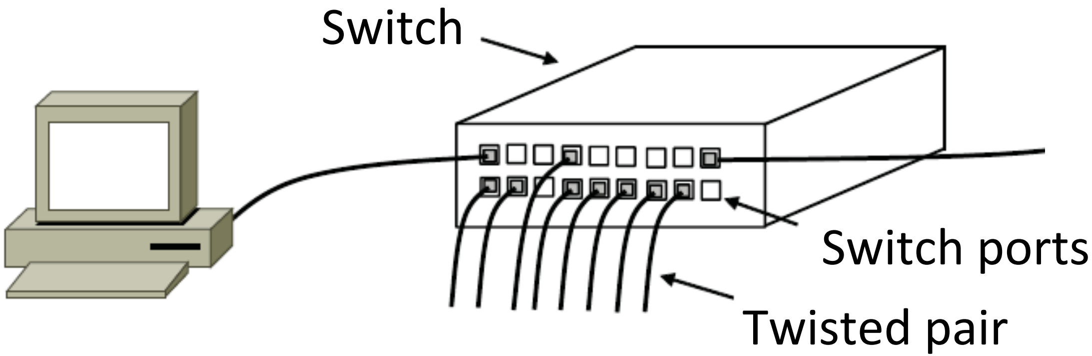
Ethernet Frame Format

- Has addresses to identify the sender and receiver
- CRC-32 for error detection; no ACKs or retransmission
- Start of frame identified with physical layer preamble



Modern Ethernet

- Based on switches, not multiple access, but still called Ethernet
  - We'll get to it in a later segment



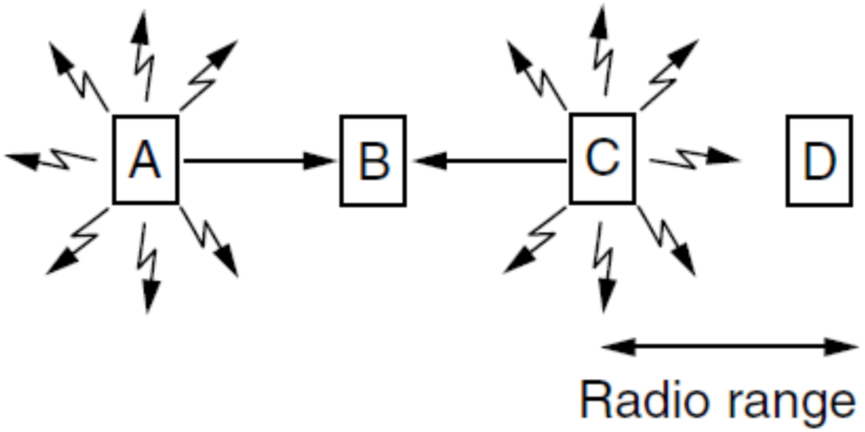
Wireless Complications

- Wireless is more complicated than the wired case (Surprise!)
  1. Nodes may have different areas of coverage – doesn't fit Carrier Sense »
  2. Nodes can't hear while sending – can't Collision Detect »

≠ CSMA/CD

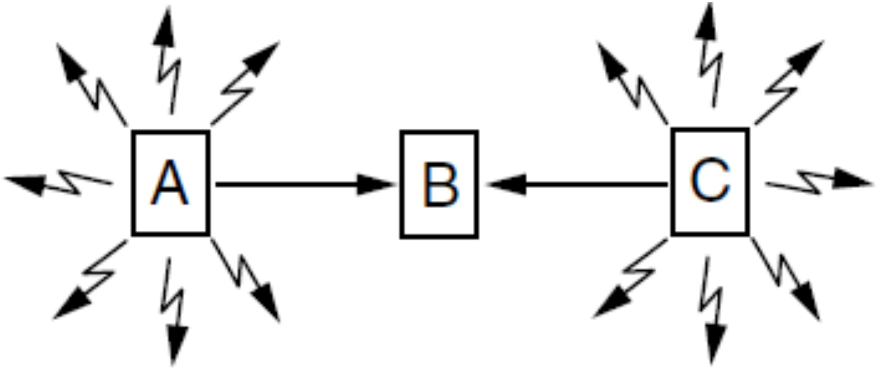
Different Coverage Areas

- Wireless signal is broadcast and received nearby, where there is sufficient SNR



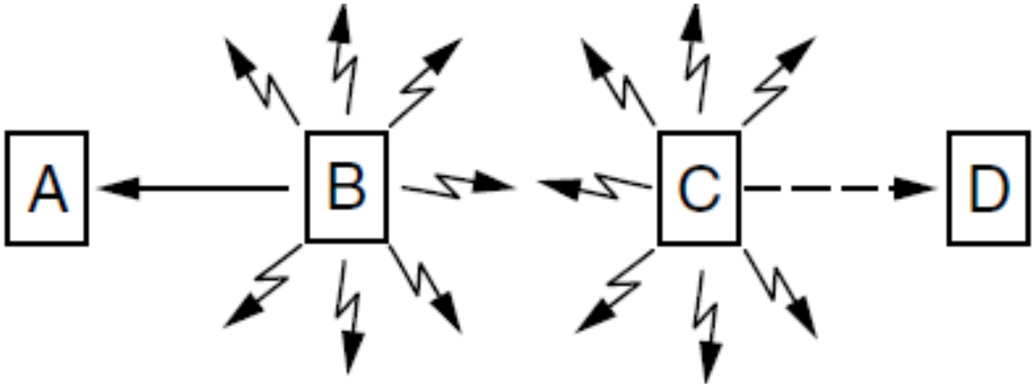
Hidden Terminals

- Nodes A and C are <u>hidden terminals</u> when sending to B
  - Can't hear each other (to coordinate) yet collide at B
  - We want to avoid the inefficiency of collisions



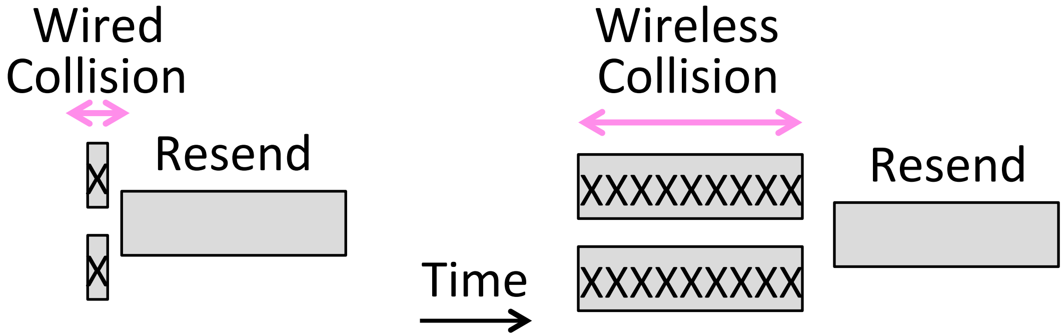
Exposed Terminals

- B and C are <u>exposed terminals</u> when sending to A and D
  - Can hear each other yet don't collide at receivers A and D
  - We want to send concurrently to increase performance



Nodes Can’t Hear While Sending

- With wires, detecting collisions (and aborting) lowers their cost
- More wasted time with wireless



Possible Solution: MACA

- MACA uses a short handshake instead of CSMA (Karn, 1990)
  - 802.11 uses a refinement of MACA (later)

- Protocol rules:
  1. A sender node transmits a RTS (Request-To-Send, with frame length)
  2. The receiver replies with a CTS (Clear-To-Send, with frame length)
  3. Sender transmits the frame while nodes hearing the CTS stay silent
  - Collisions on the RTS/CTS are still possible, but less likely

MACA – Hidden Terminals

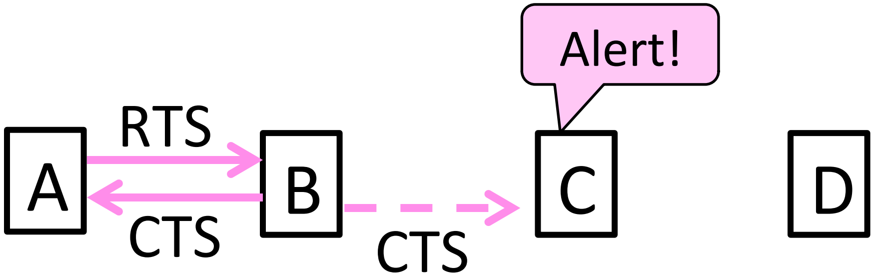
- A→B with hidden terminal C
  1. A sends RTS, to B

MACA – Hidden Terminals (2)

- A→B with hidden terminal C
  2. B sends CTS, to A, and C too

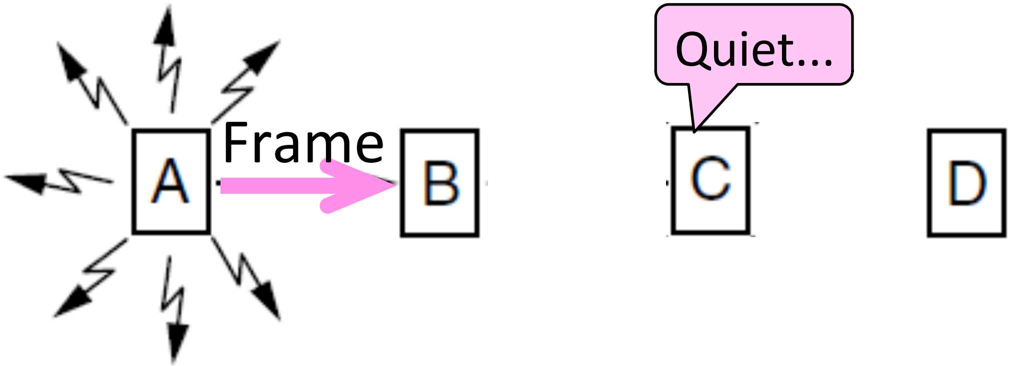
MACA – Hidden Terminals (3)

- A→B with hidden terminal C
  2. B sends CTS, to A, and C too



MACA – Hidden Terminals (4)

- A→B with hidden terminal C
  3. A sends frame while C defers

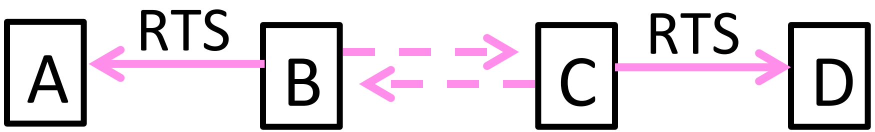


MACA – Exposed Terminals

- B→A, C→D as exposed terminals
  - B and C send RTS to A and D

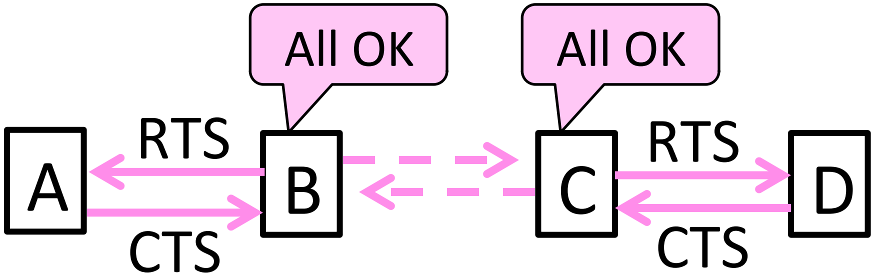
MACA – Exposed Terminals (2)

- B→A, C→D as exposed terminals
  - A and D send CTS to B and C



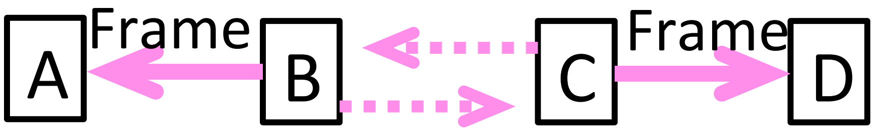
MACA – Exposed Terminals (3)

- B→A, C→D as exposed terminals
  - A and D send CTS to B and C



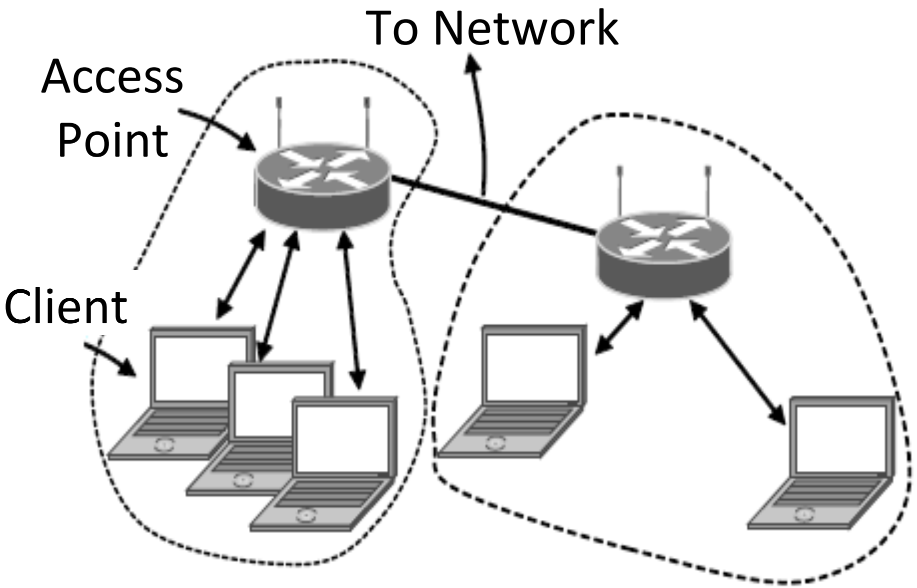
MACA – Exposed Terminals (4)

- B→A, C→D as exposed terminals
  - A and D send CTS to B and C



802.11, or WiFi

- Very popular wireless LAN started in the 1990s
- Clients get connectivity from a (wired) AP (Access Point)
- It's a multi-access problem ☺
- Various flavors have been developed over time
  - Faster, more features

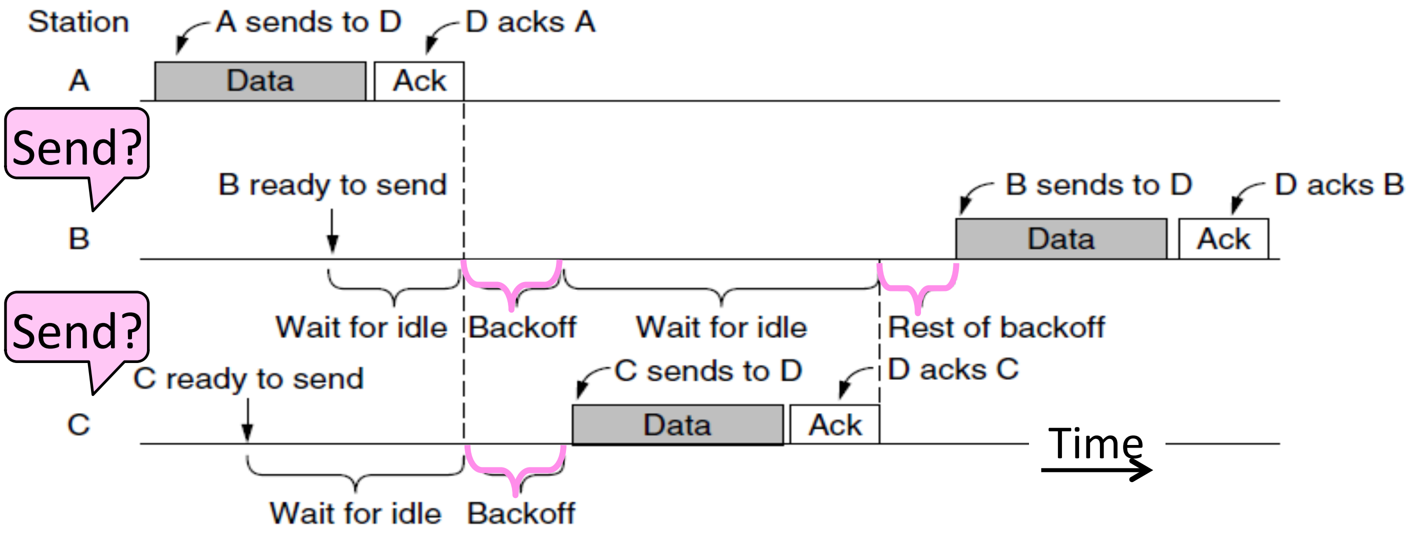


802.11 Physical Layer

- Uses 20/40 MHz channels on ISM bands
  - 802.11b/g/n on 2.4 GHz
  - 802.11 a/n on 5 GHz
- OFDM modulation (except legacy 802.11b)
  - Different amplitudes/phases for varying SNRs
  - Rates from 6 to 54 Mbps plus error correction
  - 802.11n uses multiple antennas; see "802.11 with Multiple Antennas for Dummies"

802.11 CSMA/CA for Multiple Access

- Sender avoids collisions by inserting small random gaps
  - E.g., when both B and C send, C picks a smaller gap, goes first



The Future of 802.11 (Guess)

- Likely ubiquitous for Internet connectivity
  - Greater diversity, from low- to high-end devices
- Innovation in physical layer drives speed
  - And power-efficient operation too
- More seamless integration of connectivity
  - Too manual now, and limited (e.g., device-to-device)

Issues with Random Multiple Access

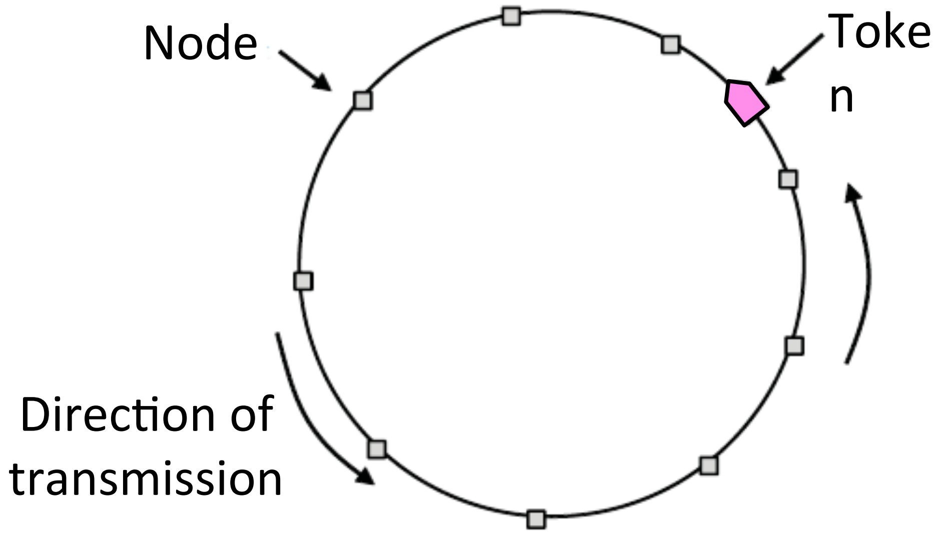
- CSMA is good under low load:
  - Grants immediate access
  - Little overhead (few collisions)
- But not so good under high load:
  - High overhead (expect collisions)
  - Access time varies (lucky/unlucky)
- We want to do better under load!

Turn-Taking Multiple Access Protocols

- They define an order in which nodes get a chance to send
  - Or pass, if no traffic at present
- We just need some ordering …
  - E.g., Token Ring »
  - E.g., node addresses

Token Ring

- Arrange nodes in a ring; token rotates "permission to send" to each node in turn



Turn-Taking Advantages

- Fixed overhead with no collisions
  - More efficient under load
- Regular chance to send with no unlucky nodes
  - Predictable service, easily extended to guaranteed quality of service

Turn-Taking Disadvantages

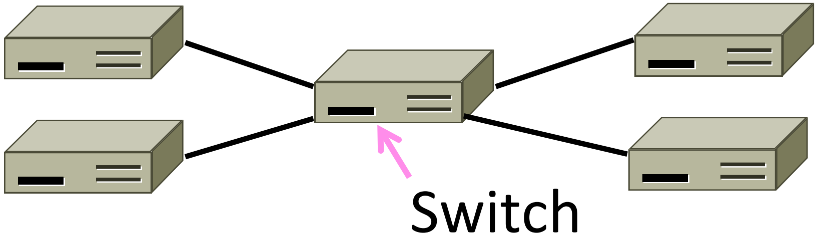
- Complexity
  - More things that can go wrong than random access protocols!
    - E.g., what if the token is lost?
  - Higher overhead at low load

Turn-Taking in Practice

- Regularly tried as an improvement offering better service
  - E.g., qualities of service
- But random multiple access is hard to beat
  - Simple, and usually good enough
  - Scales from few to many nodes

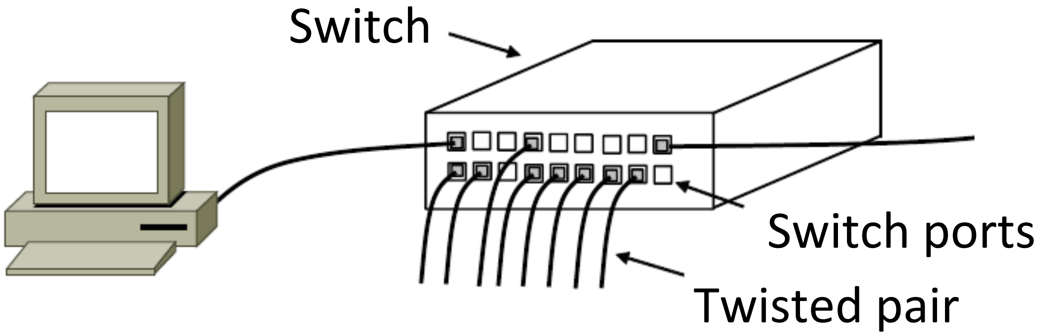
Topic

- How do we connect nodes with a switch instead of multiple access
  - Uses multiple links/wires
  - Basis of modern (switched) Ethernet



Switched Ethernet

- Hosts are wired to Ethernet switches with twisted pair
  - Switch serves to connect the hosts
  - Wires usually run to a closet



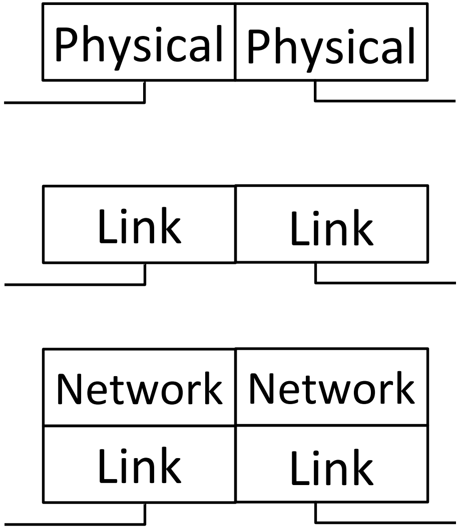
What's in the box?

- Remember from protocol layers:

Hub, or repeater

Switch

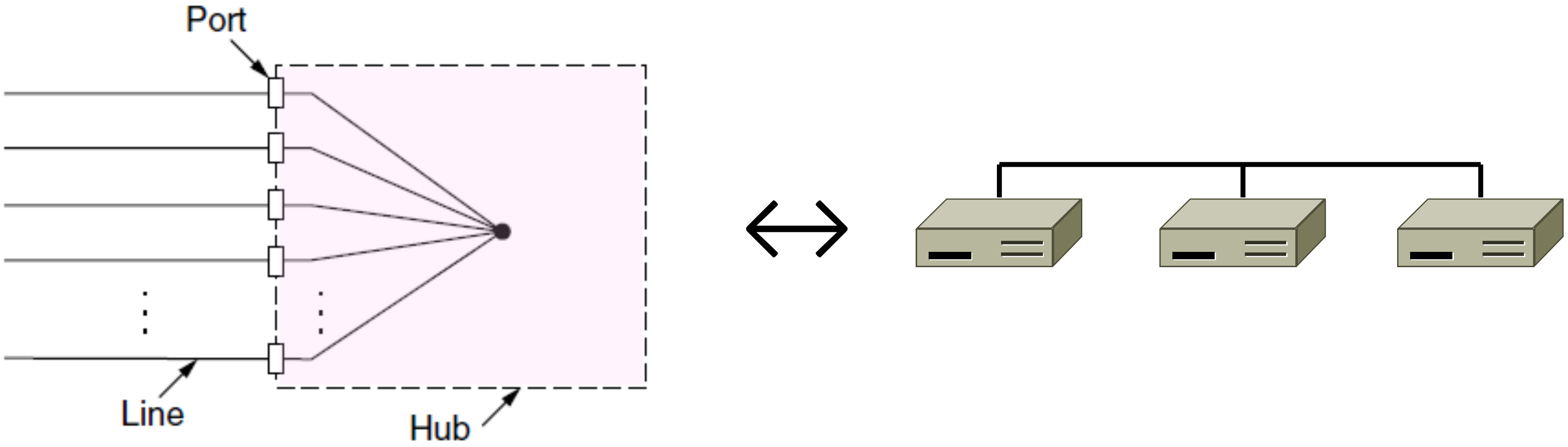
Router



All look like this:

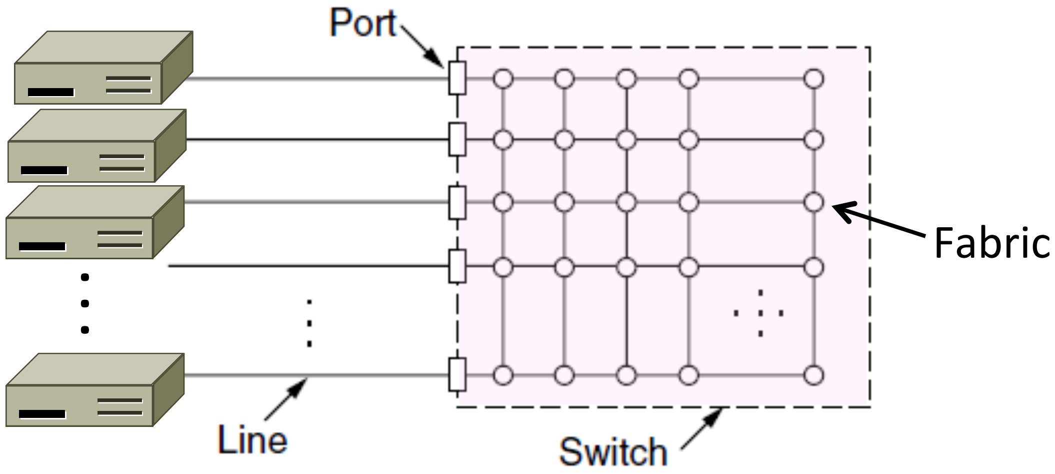
Inside a Hub

- All ports are wired together; more convenient and reliable than a single shared wire



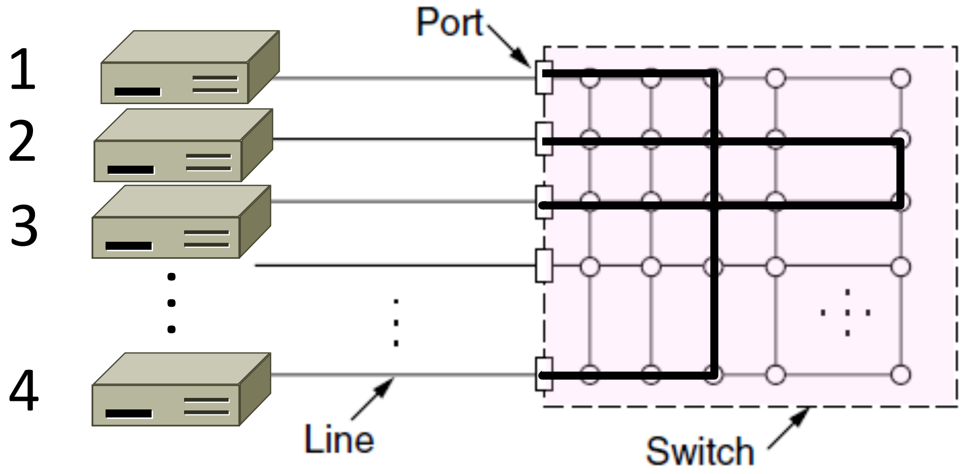
Inside a Switch

- Uses frame addresses to connect input port to the right output port; multiple frames may be switched in parallel



Inside a Switch (2)

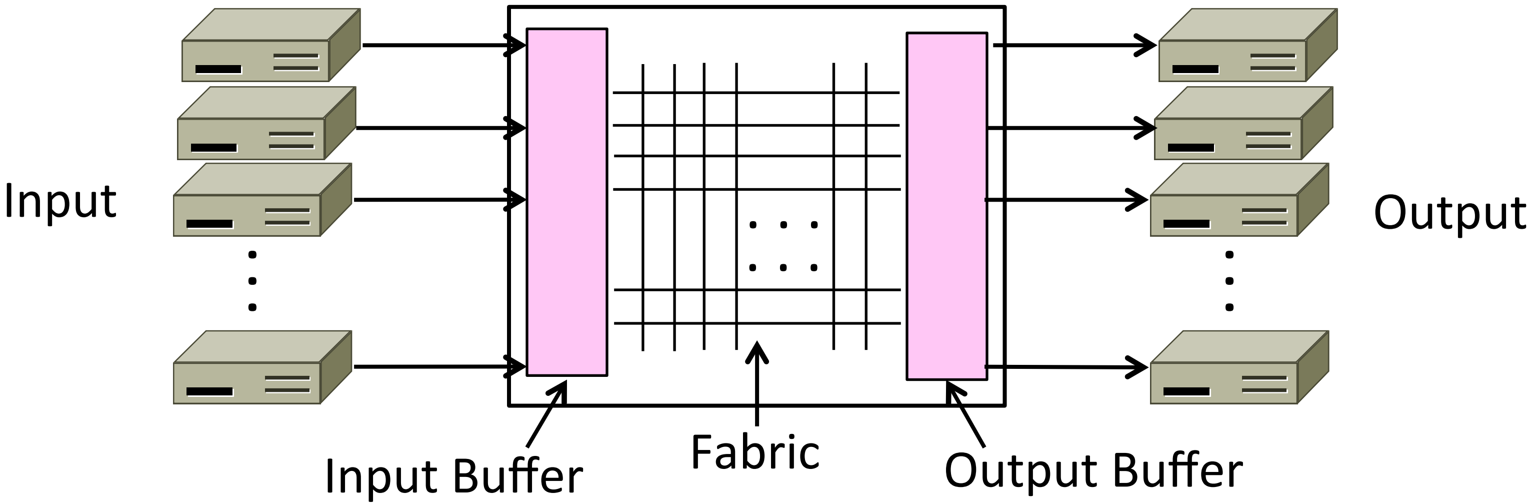
- Port may be used for both input and output (full-duplex)
  - Just send, no multiple access protocol



1 → 4
and
2 → 3

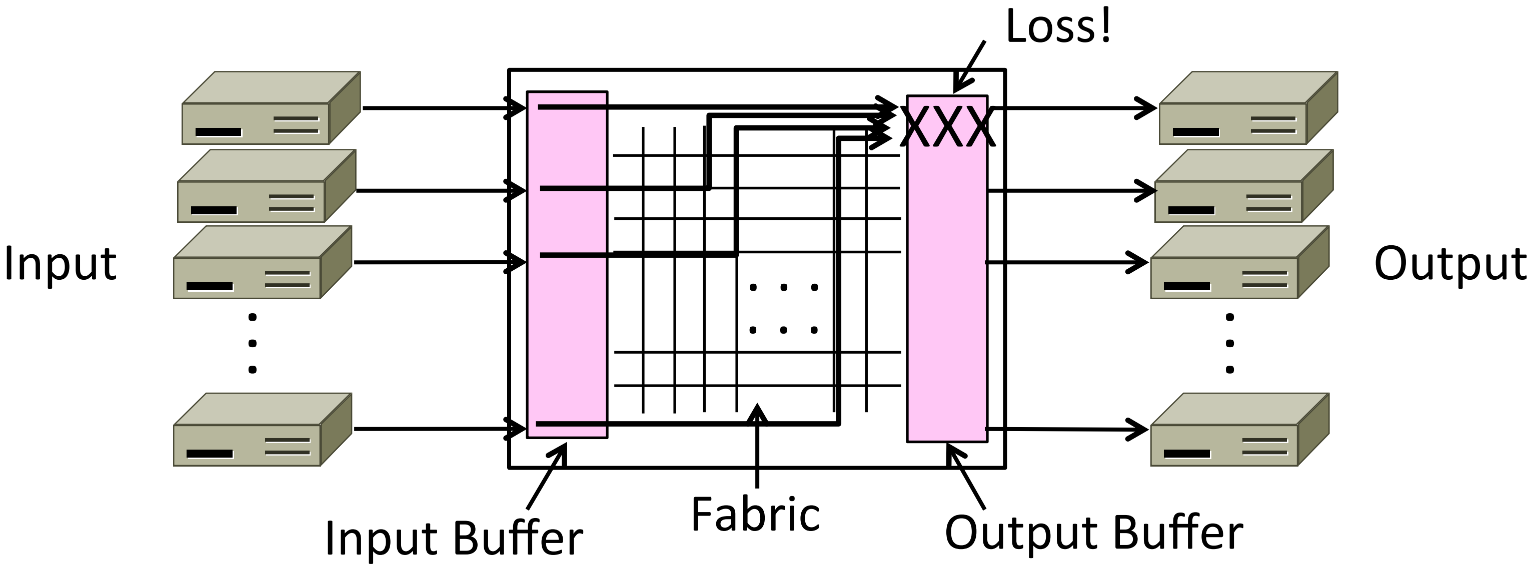
Inside a Switch (3)

- Need buffers for multiple inputs to send to one output



Inside a Switch (4)

- Sustained overload will fill buffer and lead to frame loss



Advantages of Switches

- Switches and hubs have replaced the shared cable of classic Ethernet
  - Convenient to run wires to one location
  - More reliable; wire cut is not a single point of failure that is hard to find
- Switches offer scalable performance
  - E.g., 100 Mbps per port instead of 100 Mbps for all nodes of shared cable / hub