Links

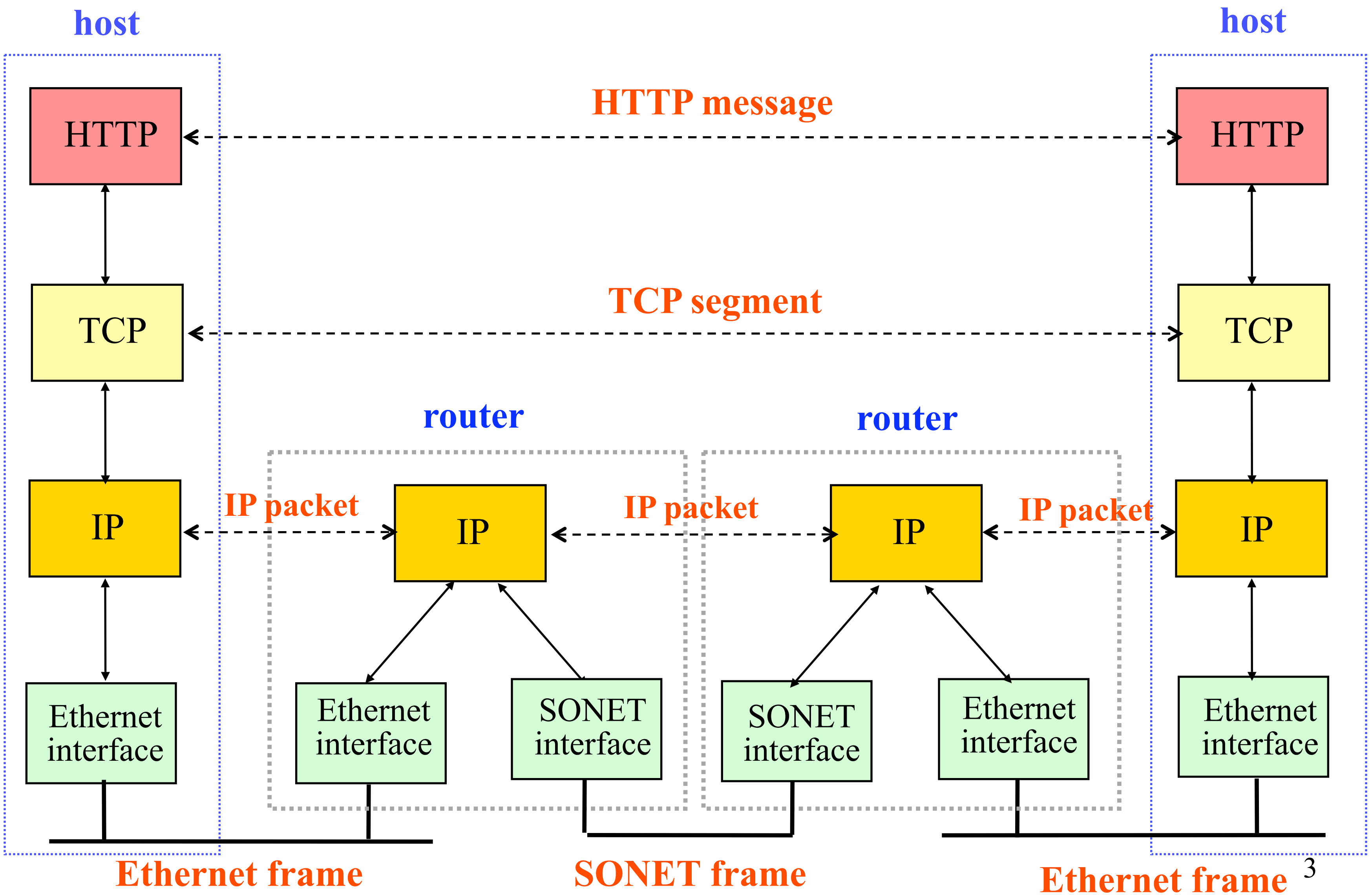
Reading: Chapter 2

Acknowledgments: Lecture slides are from Computer networks course thought by Jennifer Rexford at Princeton University. When slides are obtained from other sources, a reference will be noted on the bottom of that slide and full reference details on the last slide.

Goals of Today's Lecture

- Link-layer services
  - Encoding, framing, and error detection
  - Error correction and flow control
- Sharing a shared media
  - Channel partitioning
  - Taking turns
  - Random access
- Ethernet protocol
  - Carrier sense, collision detection, and random access
  - Frame structure
  - Hubs and switches

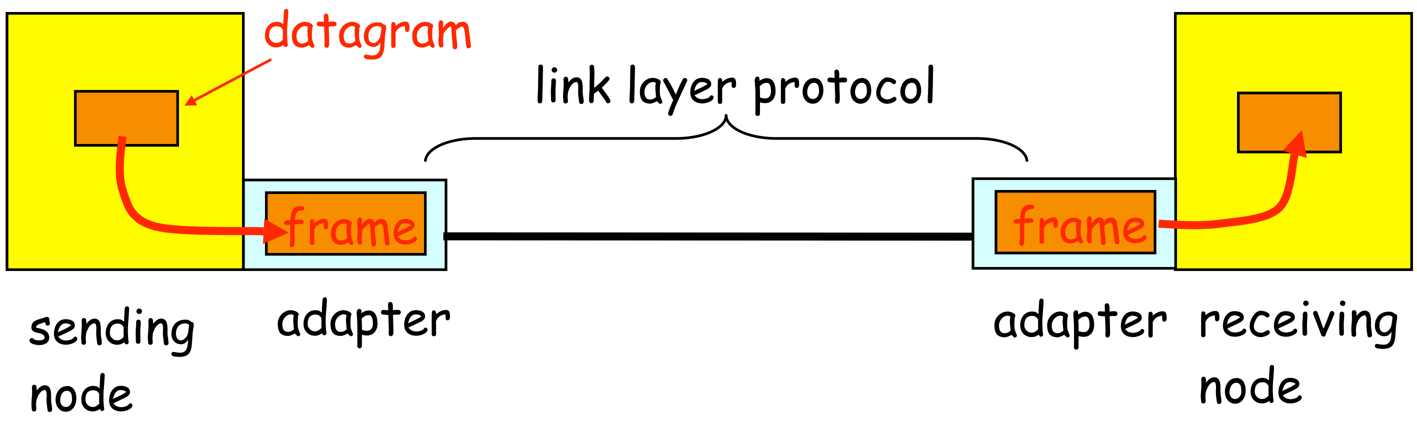
Message, Segment, Packet, and Frame



Link Layer Protocol for Each Hop

- IP packet transferred over multiple hops
  - Each hop has a link layer protocol
  - May be different on different hops
- Analogy: trip from Sharif to Lausanne
  - Taxi: Sharif to IKA
  - Plane: IKA to Geneva
  - Train: Geneva to Lausanne
- Refining the analogy
  - Tourist == packet
  - Transport segment == communication link
  - Transportation mode == link-layer protocol
  - Travel agent == routing algorithm

Adaptors Communicating



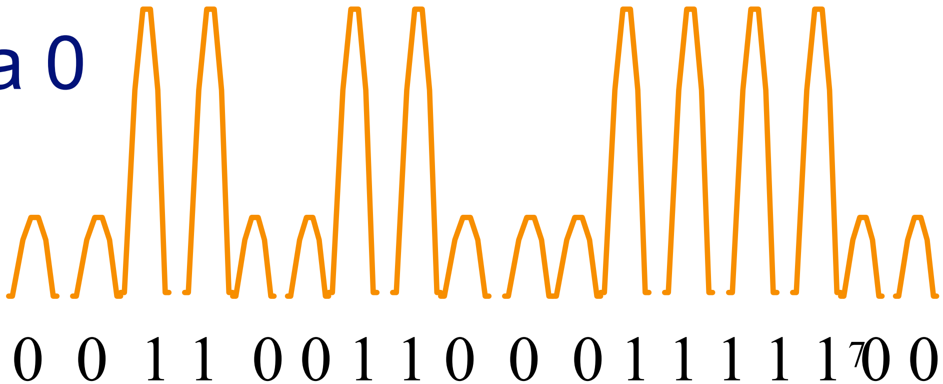
- Link layer implemented in adaptor (network interface card)
  - Ethernet card, PCMCIA card, 802.11 card
- Sending side:
  - Encapsulates datagram in a frame
  - Adds error checking bits, flow control, etc.
- Receiving side
  - Looks for errors, flow control, etc.
  - Extracts datagram and passes to receiving node

Link-Layer Services

- Encoding
  - Representing the 0s and 1s
- Framing
  - Encapsulating packet into frame, adding header, trailer
  - Using MAC addresses, rather than IP addresses
- Error detection
  - Errors caused by signal attenuation, noise.
  - Receiver detecting presence of errors
- Error correction
  - Receiver correcting errors without retransmission
- Flow control
  - Pacing between adjacent sending and receiving nodes

Encoding

- Signals propagate over physical links
  - Source node encodes the bits into a signal
  - Receiving node decodes the signal back into bits
- Simplify some electrical engineering details
  - Assume two discrete signals, high and low
  - E.g., could correspond to two different voltages
- Simple approach
  - High for a 1, low for a 0



Problem With Simple Approach

- Long strings of 0s or 1s introduce problems
  - No transitions from low-to-high, or high-to-low
- Receiver keeps average of signal it has received
  - Uses the average to distinguish between high and low
  - Long flat strings make receiver sensitive to small change
- Transitions also necessary for clock recovery
  - Receiver uses transitions to drive its own clock
  - Long flat strings do not produce any transitions
  - Can lead to clock drift at the receiver
- Alternatives (see Section 2.2)
  - Non-return to zero inverted, and Manchester encoding

Framing

- Break sequence of bits into a frame
  - Typically implemented by the network adaptor
- Sentinel-based
  - Delineate frame with special pattern (e.g., 01111110)

01111110	**Frame contents**	**01111110**

  - Problem: what if special patterns occurs within frame?
  - Solution: escaping the special characters
    - E.g., sender always inserts a 0 after five 1s
    - … and receiver always removes a 0 appearing after five 1s
  - Similar to escaping special characters in C programs

Framing (Continued)

- Counter-based
  - Include the payload length in the header
  - … instead of putting a sentinel at the end
  - Problem: what if the count field gets corrupted?
    - Causes receiver to think the frame ends at a different place
  - Solution: catch later when doing error detection
    - And wait for the next sentinel for the start of a new frame
- Clock-based
  - Make each frame a fixed size
  - No ambiguity about start and end of frame
  - But, may be wasteful

Error Detection

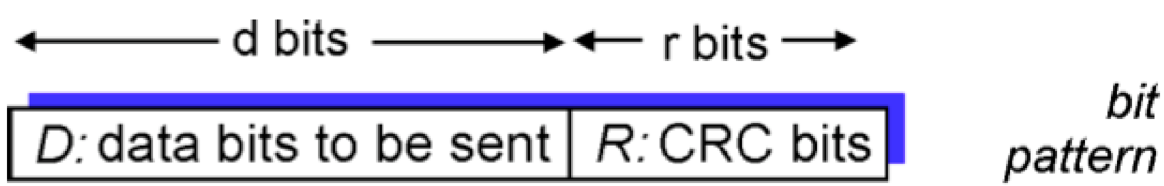
- Errors are unavoidable
  - Electrical interference, thermal noise, etc.
- Error detection
  - Transmit extra (redundant) information
  - Use redundant information to detect errors
  - Extreme case: send two copies of the data
  - Trade-off: accuracy vs. overhead
- Techniques for detecting errors
  - Parity checking
  - Checksum
  - Cyclic Redundancy Check (CRC)

Error Detection Techniques

- Parity check
  - Add an extra bit to a 7-bit code
  - Odd parity: ensure an odd number of 1s
    - E.g., 0101011 becomes 01010111
  - Even parity: ensure an even number of 1s
    - E.g., 0101011 becomes 01010110
- Checksum
  - Treat data as a sequence of 16-bit words
  - Compute a sum of all the 16-bit words, with no carries
  - Transmit the sum along with the packet
- Cyclic Redundancy Check (CRC)
  - See Section 2.4.3

Error Detection – CRC

- View data bits, D, as a binary number
- Choose r+1 bit pattern (generator), G
- Goal: choose r CRC bits, R, such that
  - <D,R> exactly divisible by G (modulo 2)
  - Receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!
  - Can detect all burst errors less than r+1 bits
- Widely used in practice (Ethernet, FDDI)



[Savage05]

Common Generator Polynomials

CRC-8	$x^8 + x^2 + x^1 + 1$
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x^1 + 1$
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x^1 + 1$
CRC-16	$x^{16} + x^{15} + x^2 + 1$
CRC-CCITT	$x^{16} + x^{12} + x^5 + 1$
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$

[Savage05]

CRC–Example Encoding

$x^3 + x^2 + 1 = 1101$ Generator

$x^7 + x^4 + x^3 + x = 10011010$ Message

```
1101 | 10011010000   <-- Message plus k zeros
       1101
       ----
        1001
        1101
        ----
         1000
         1101
         ----
          1011
          1101
          ----
           1100
           1101
           ----
             1000
             1101
             ----
              101   <-- Remainder m mod c
```

k + 1 bit check sequence c, equivalent to a degree-k polynomial

Remainder

m mod c

Result:

Transmit message followed by remainder:

10011010101

CRC- Example Decoding - No Errors

$x^3 + x^2 + 1$	= 1101	**Generator**
$x^{10} + x^7 + x^6 + x^4 + x^2 + 1$	= 10011010101	**Received Message**

```
1101 | 10011010101   <- Received message, no errors
       1101
       ----
        1001
        1101
        ----
         1000
         1101
         ----
          1011
          1101
          ----
           1100
           1101
           ----
             1101
             1101
             ----
                0
```

k + 1 bit check sequence c, equivalent to a degree-k polynomial (→ 1101)

Remainder (→ 0)

m mod c

Result:

CRC test is passed

CRC- Example Decoding - with Errors

$x^3 + x^2 + 1$ = 1101 **Generator**

$x^{10} + x^7 + x^5 + x^4 + x^2 + 1$ = 10010110101 **Received Message**

```
1101 | 10010110101     <-- Received message
       1101
       ----
        1000
        1101
        ----
         1011
         1101
         ----
          1101
          1101
          ----
           0101
```

k + 1 bit check sequence c, equivalent to a degree-k polynomial (→ 1101)

Two bit errors (→ 01 in 10010110101)

Remainder m mod c (→ 0101)

Result:

CRC test failed

Point-to-Point vs. Broadcast Media

- Point-to-point
  - PPP for dial-up access
  - Point-to-point link between Ethernet switch and host
- Broadcast (shared wire or medium)
  - Traditional Ethernet
  - 802.11 wireless LAN

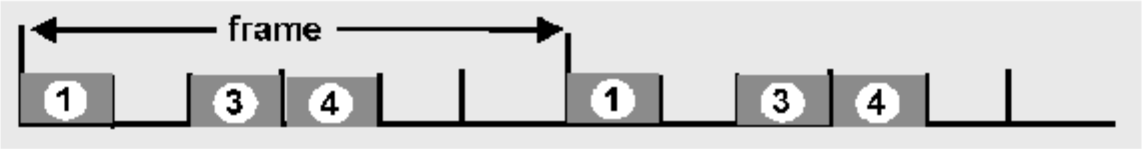
Multiple Access Protocol

- Single shared broadcast channel
  - Avoid having multiple nodes speaking at once
  - Otherwise, collisions lead to garbled data
- Multiple access protocol
  - Distributed algorithm for sharing the channel
  - Algorithm determines which node can transmit
- Classes of techniques
  - Channel partitioning: divide channel into pieces
  - Taking turns: passing a token for the right to transmit
  - Random access: allow collisions, and then recover

Channel Partitioning: TDMA

TDMA: time division multiple access

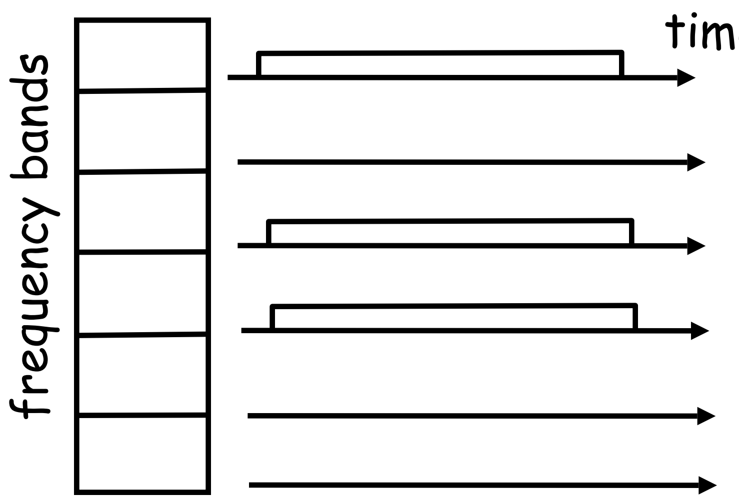
- Access to channel in "rounds"
  - Each station gets fixed length slot in each round
- Time-slot length is packet transmission time
  - Unused slots go idle
- Example: 6-station LAN with slots 1, 3, and 4



Channel Partitioning: FDMA

FDMA: frequency division multiple access

- Channel spectrum divided into frequency bands
  - Each station assigned fixed frequency band
- Unused transmission time in bands go idle
- Example: 6-station LAN with bands 1, 3, and 4



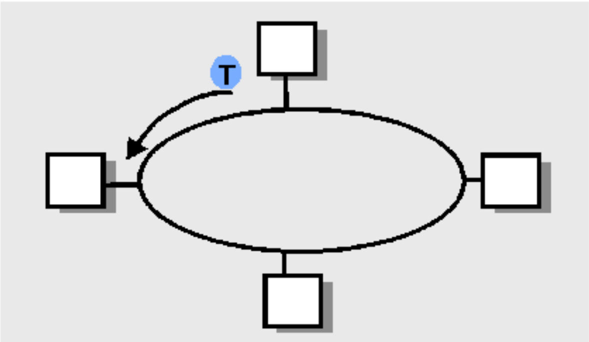
"Taking Turns" MAC protocols

Polling

- Master node "invites" slave nodes to transmit in turn
- Concerns:
  - Polling overhead
  - Latency
  - Single point of failure (master)

Token passing

- Control token passed from one node to next sequentially
- Token message
- Concerns:
  - Token overhead
  - Latency
  - Single point of failure (token)



Random Access Protocols

- When node has packet to send
  - Transmit at full channel data rate R.
  - No a priori coordination among nodes
- Two or more transmitting nodes ➜ “collision”,
- Random access MAC protocol specifies:
  - How to detect collisions
  - How to recover from collisions
- Examples
  - ALOHA and Slotted ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Key Ideas of Random Access

- Carrier Sense (CS)
  - *Listen before speaking, and don't interrupt*
  - Checking if someone else is already sending data
  - … and waiting till the other node is done
- Collision Detection (CD)
  - *If someone else starts talking at the same time, stop*
  - Realizing when two nodes are transmitting at once
  - …by detecting that the data on the wire is garbled
- Randomness
  - *Don't start talking again right away*
  - Waiting for a random time before trying again

Slotted ALOHA

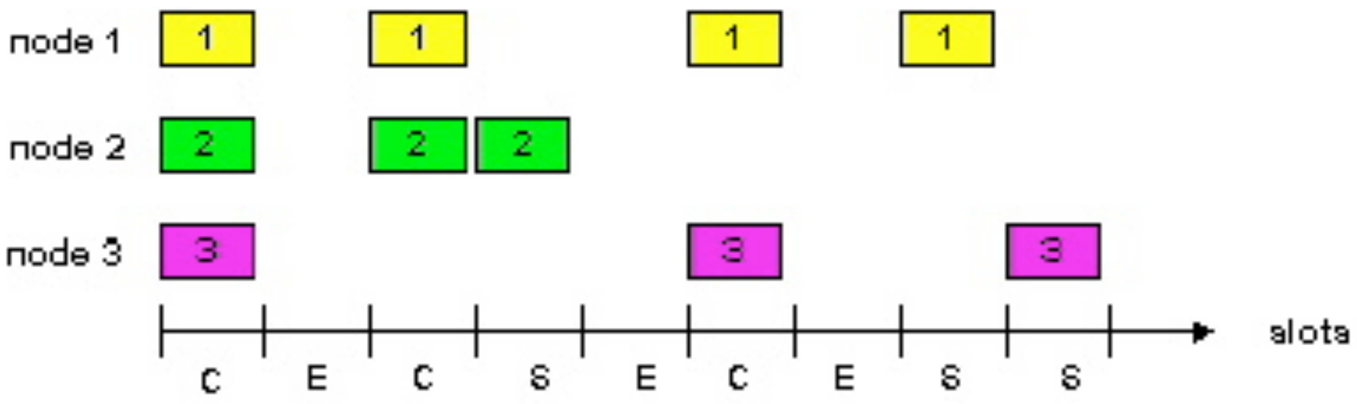
Assumptions

- All frames same size
- Time divided into equal slots (time to transmit a frame)
- Nodes start to transmit frames only at start of slots
- Nodes are synchronized
- If two or more nodes transmit, all nodes detect collision

Operation

- When node obtains fresh frame, transmits in next slot
- No collision: node can send new frame in next slot
- Collision: node retransmits frame in each subsequent slot with probability p until success

Slotted ALOHA



Pros

- Single active node can continuously transmit at full rate of channel
- Highly decentralized: only slots in nodes need to be in sync
- Simple

Cons

- Collisions, wasting slots
- Idle slots
- Nodes may be able to detect collision in less than time to transmit packet
- Clock synchronization

CSMA (Carrier Sense Multiple Access)

- Collisions hurt the efficiency of ALOHA protocol
  - At best, channel is useful 37% of the time
- CSMA: listen before transmit
  - If channel sensed idle: transmit entire frame
  - If channel sensed busy, defer transmission
- Human analogy: don’t interrupt others!

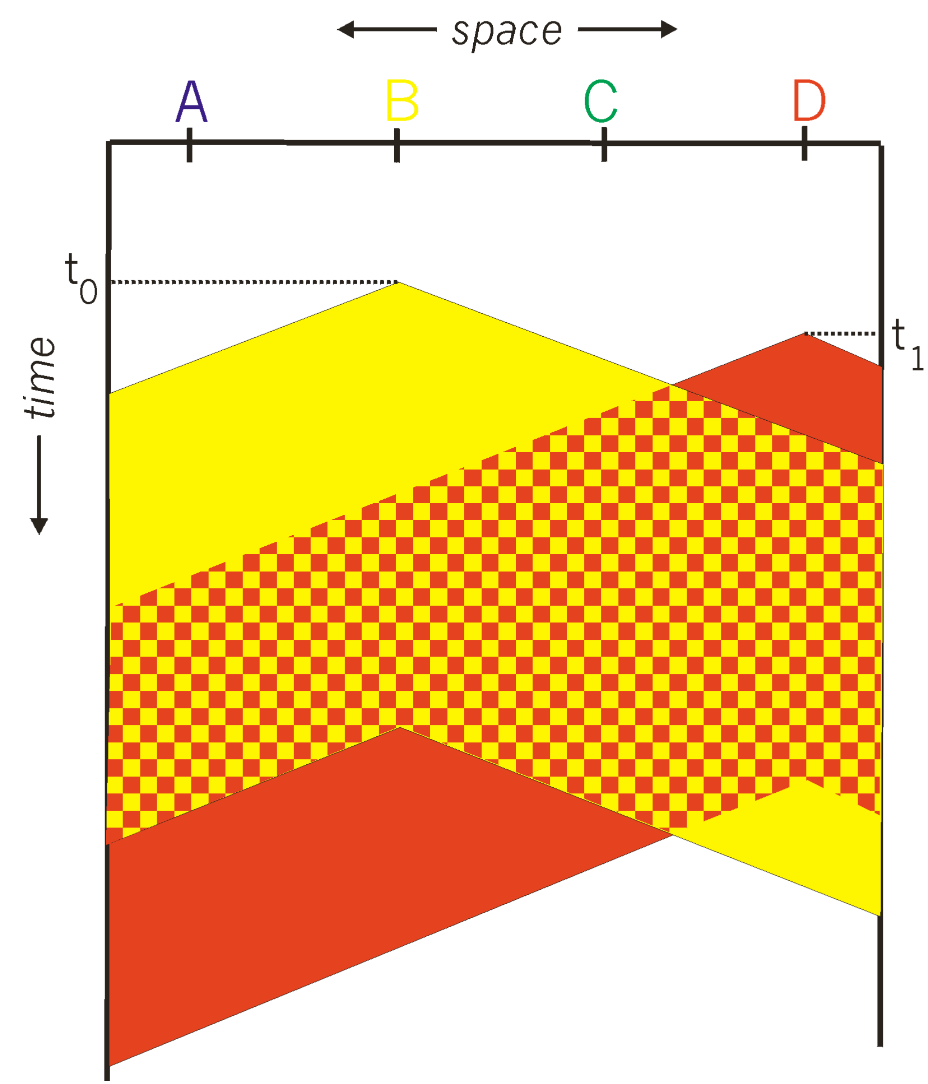
CSMA Collisions

Collisions *can* still occur:

propagation delay means two nodes may not hear each other's transmission

Collision:

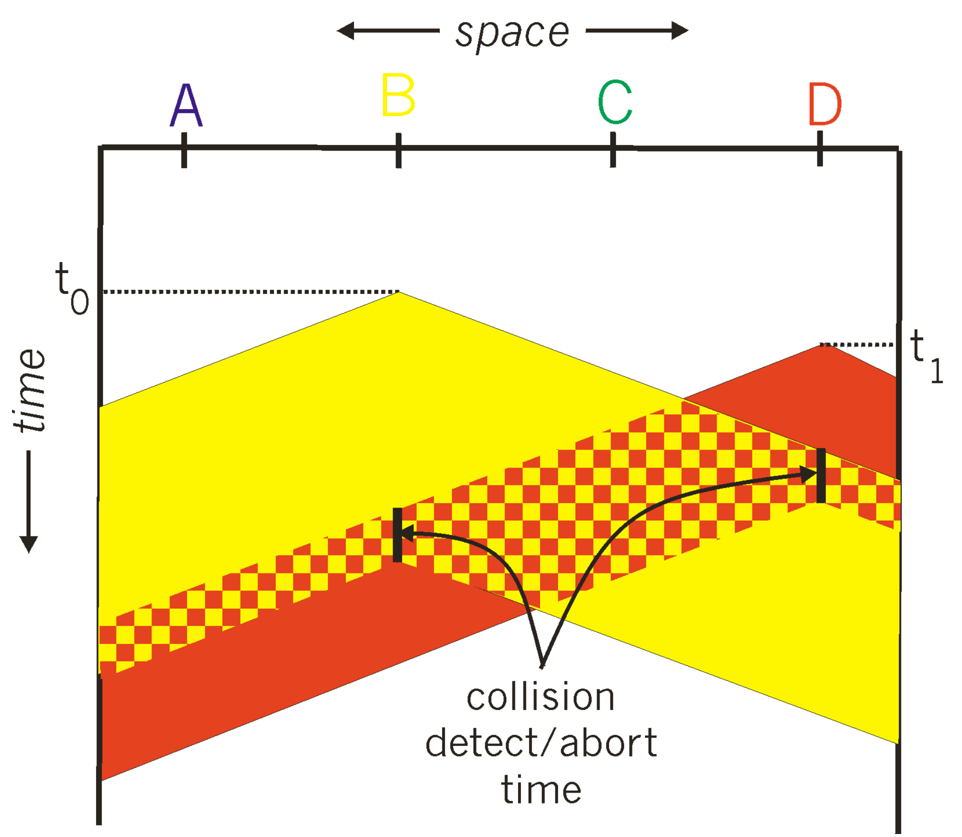
entire packet transmission time wasted



CSMA/CD (Collision Detection)

- CSMA/CD: carrier sensing, deferral as in CSMA
  - Collisions detected within short time
  - Colliding transmissions aborted, reducing wastage
- Collision detection
  - Easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - Difficult in wireless LANs: receiver shut off while transmitting
- Human analogy: the polite conversationalist

CSMA/CD Collision Detection

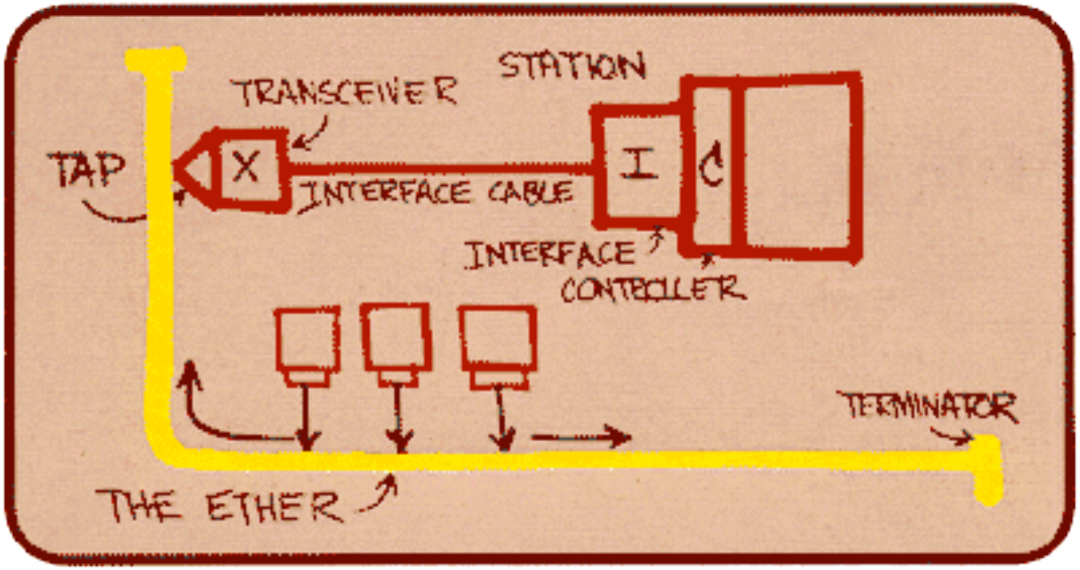


Three Ways to Share the Media

- Channel partitioning MAC protocols:
  - Share channel efficiently and fairly at high load
  - Inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!
- "Taking turns" protocols
  - Eliminates empty slots without causing collisions
  - Vulnerable to failures (e.g., failed node or lost token)
- Random access MAC protocols
  - Efficient at low load: single node can fully utilize channel
  - High load: collision overhead

Ethernet

- Dominant wired LAN technology:
- First widely used LAN technology
- Simpler, cheaper than token LANs and ATM
- Kept up with speed race: 10 Mbps – 10 Gbps

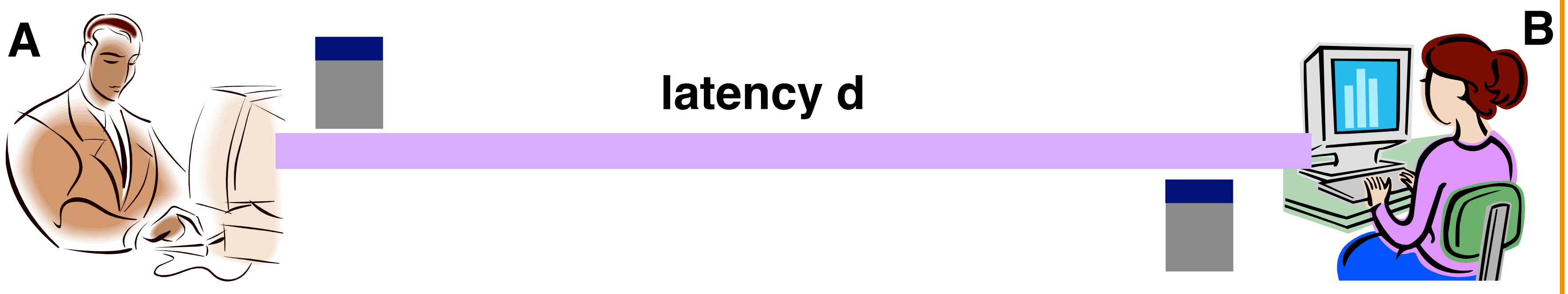


Metcalfe's Ethernet sketch

Ethernet Uses CSMA/CD

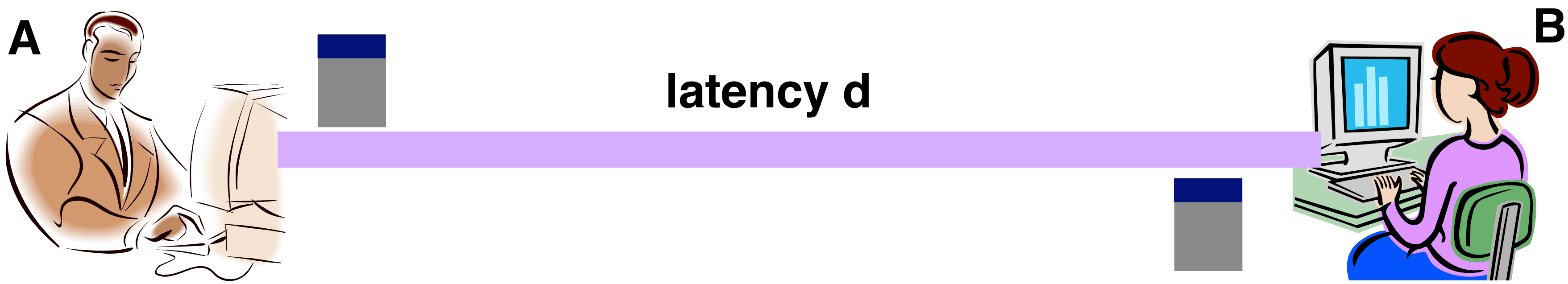
- Carrier Sense: wait for link to be idle
  - Channel idle: start transmitting
  - Channel busy: wait until idle
- Collision Detection: listen while transmitting
  - No collision: transmission is complete
  - Collision: abort transmission, and send jam signal
- Random access: exponential back-off
  - After collision, wait a random time before trying again
  - After m^{th} collision, choose K randomly from $\{0, \ldots, 2^m-1\}$
  - … and wait for K*512 bit times before trying again

Limitations on Ethernet Length



- Latency depends on physical length of link
  - Time to propagate a packet from one end to the other
- Suppose A sends a packet at time t
  - And B sees an idle line at a time just before t+d
  - … so B happily starts transmitting a packet
- B detects a collision, and sends jamming signal
  - But A doesn't see collision till t+2d

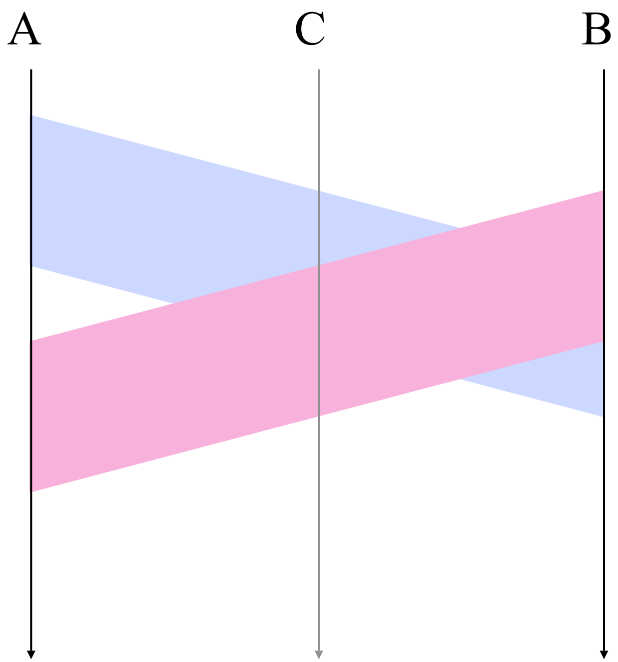
Limitations on Ethernet Length

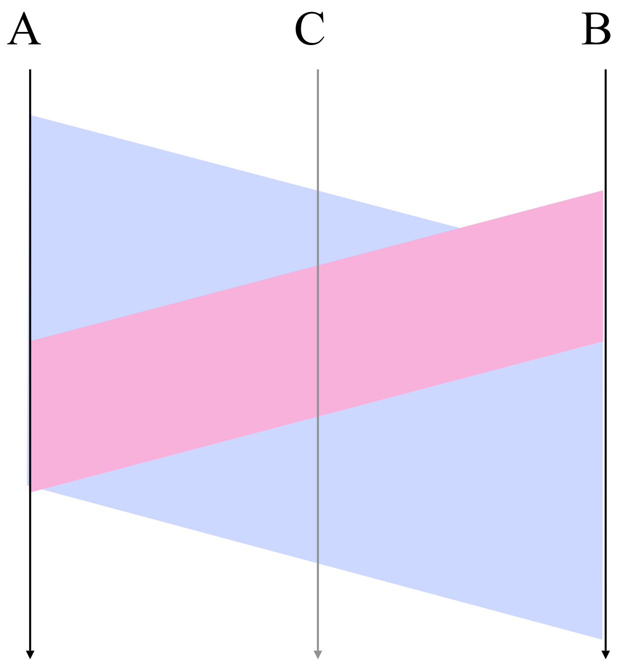


- A needs to wait for time 2d to detect collision
  - So, A should keep transmitting during this period
  - … and keep an eye out for a possible collision
- Imposes restrictions on Ethernet
  - Maximum length of the wire: 2500 meters
  - Minimum length of the packet: 512 bits (64 bytes)

A Minimum Frame Size is Necessary to Guarantee Collision Detection

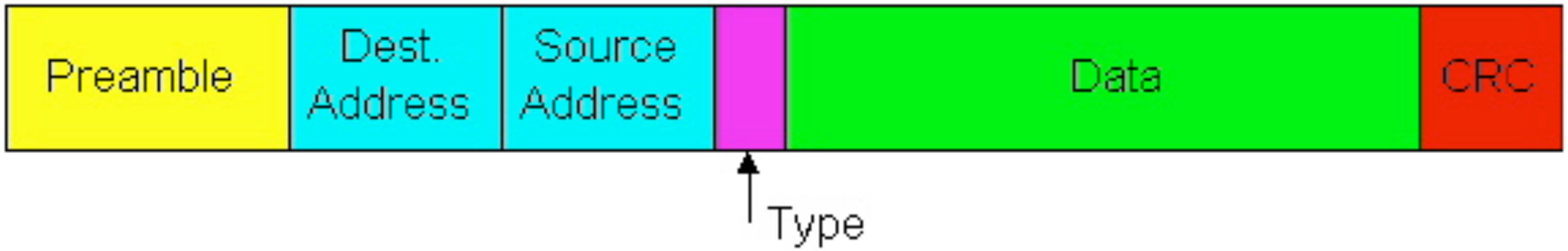
- Frame sent by A is too short: collision is not visible at A but is visible at C
- Frame sent by A is large enough collision is visible at A





Ethernet Frame Structure

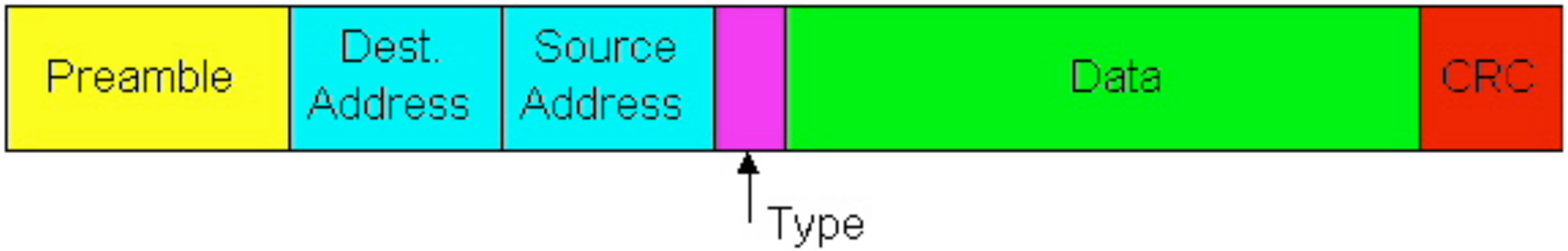
- Sending adapter encapsulates packet in frame



- Preamble: synchronization
  - Seven bytes with pattern 10101010, followed by one byte with pattern 10101011
  - Used to synchronize receiver, sender clock rates

Ethernet Frame Structure (Continued)

- Addresses: source and destination MAC addresses
  - Adaptor passes frame to network-level protocol
    - If destination address matches the adaptor
    - Or the destination address is the broadcast address
  - Otherwise, adapter discards frame
- Type: indicates the higher layer protocol
  - Usually IP
  - But also Novell IPX, AppleTalk, …
- CRC: cyclic redundancy check
  - Checked at receiver
  - If error is detected, the frame is simply dropped

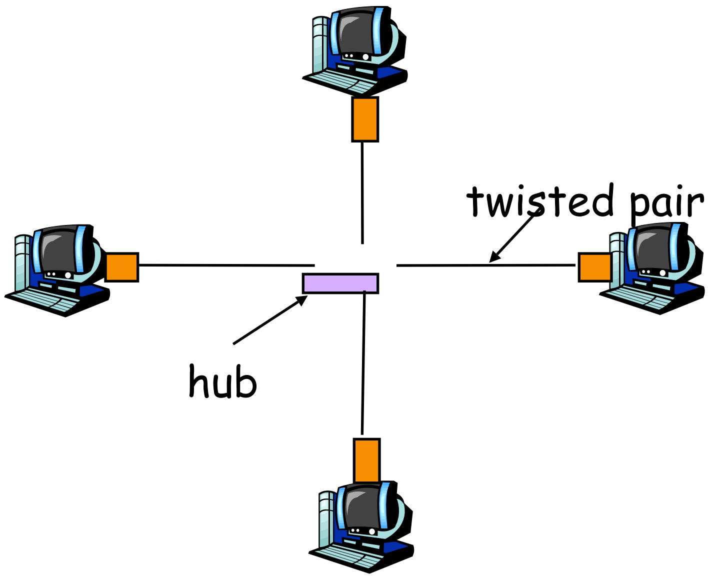


Unreliable, Connectionless Service

- Connectionless
  - No handshaking between sending and receiving adapter.
- Unreliable
  - Receiving adapter doesn't send ACKs or NACKs
  - Packets passed to network layer can have gaps
  - Gaps will be filled if application is using TCP
  - Otherwise, the application will see the gaps

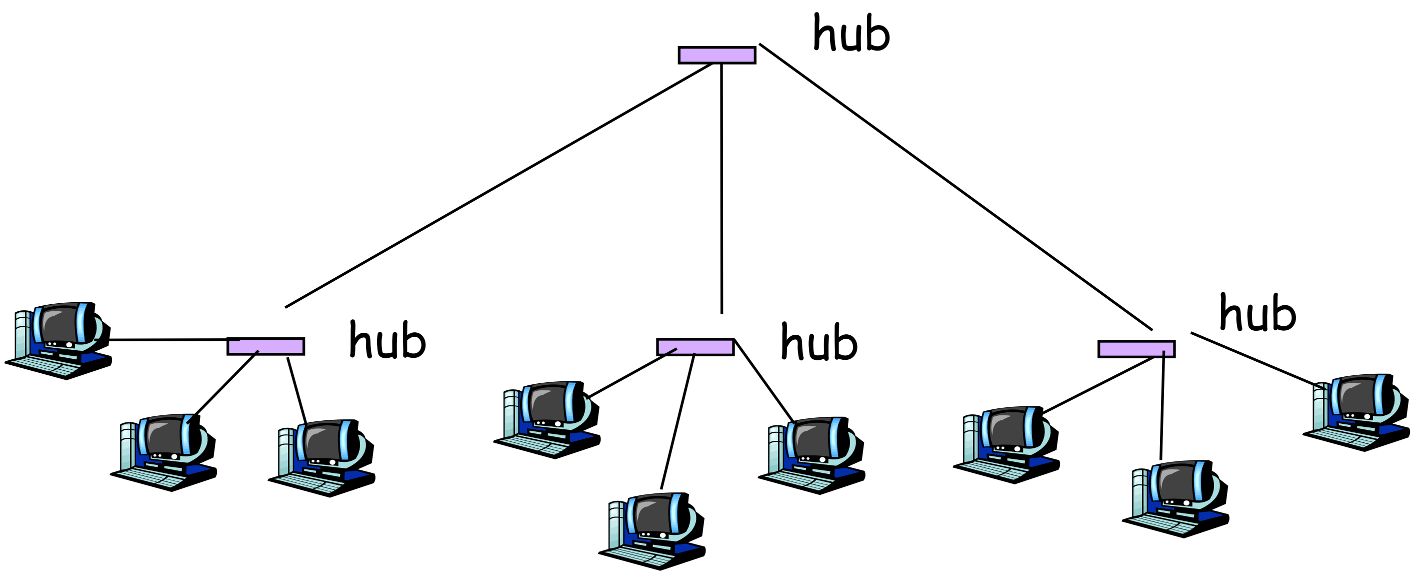
Hubs: Physical-Layer Repeaters

- Hubs are physical-layer repeaters
  - Bits coming from one link go out all other links
  - At the same rate, with no frame buffering
  - No CSMA/CD at hub: adapters detect collisions



Interconnecting with Hubs

- Backbone hub interconnects LAN segments
- All packets seen everywhere, forming one large collision domain
- Can't interconnect Ethernets of different speeds

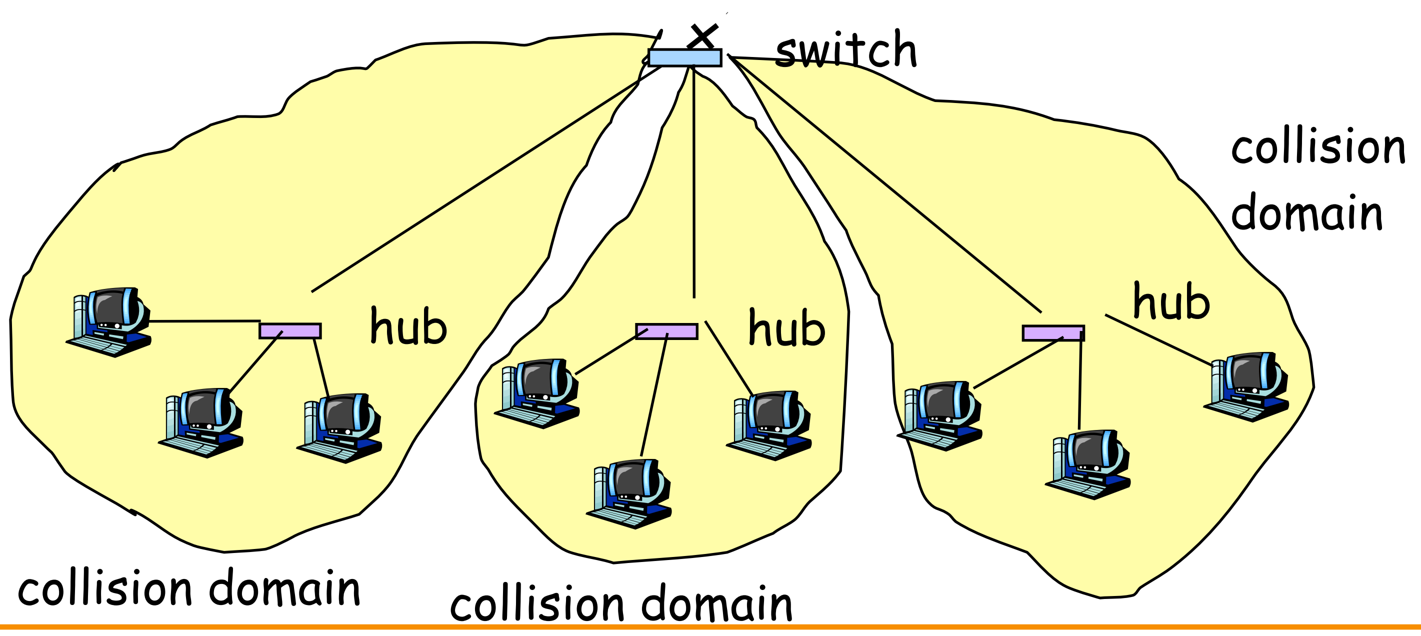


Switch

- Link layer device
  - Stores and forwards Ethernet frames
  - Examines frame header and selectively forwards frame based on MAC dest address
  - When frame is to be forwarded on segment, uses CSMA/CD to access segment
- Transparent
  - Hosts are unaware of presence of switches
- Plug-and-play, self-learning
  - Switches do not need to be configured

Switch: Traffic Isolation

- Switch breaks subnet into LAN segments
- Switch filters packets
  - Same-LAN-segment frames not usually forwarded onto other LAN segments
  - Segments become separate collision domains



Benefits of Ethernet

- Easy to administer and maintain
- Inexpensive
- Increasingly higher speed
- Moved from shared media to switches
  - Change everything except the frame format
  - A good general lesson for evolving the Internet

Conclusions

- IP runs on a variety of link layer technologies
  - Point-to-point links vs. shared media
  - Wide varieties within each class
- Link layer performs key services
  - Encoding, framing, and error detection
  - Optionally error correction and flow control
- Shared media introduce interesting challenges
  - Decentralized control over resource sharing
  - Partitioned channel, taking turns, and random access
  - Ethernet as a wildly popular example

Future Classes

- Link-State Routing
- Distance-Vector and Path-Vector Routing
- Policy-Based Path-Vector Routing
- Overlay Networks
- Peer-to-Peer
- Circuit Switching
- Wireless and Mobile Networks

Acknowledgments/References

- [Savage05] CSE 123A, Computer Networks, Stefan Savage, Fall 2005.UCSD.