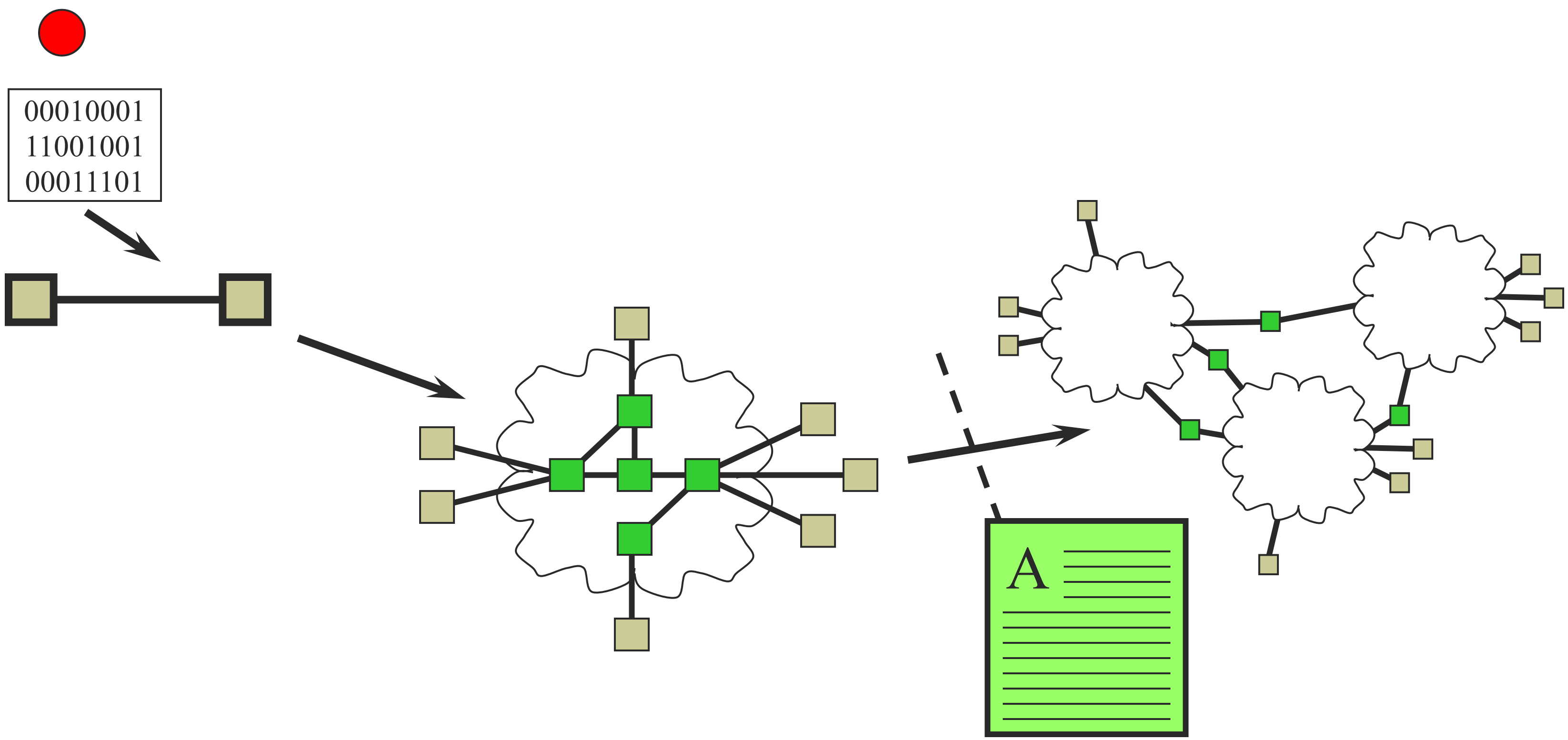
# Media Access Protocols

# Where are We?

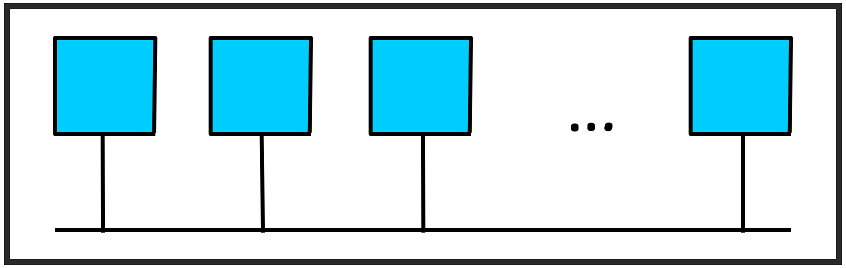
you are here




midterm is here

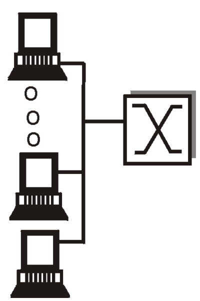
# Multiple Access Media

- Multiple senders on some media
  - Buses (Ethernet)
  - Radio, Satellite
  - Token Ring
- Need methods to mediate access
  - Fair arbitration
  - Good performance

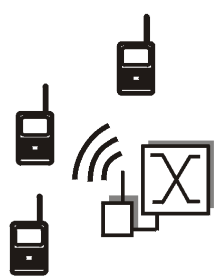


# Point-to-Point vs. Broadcast Media

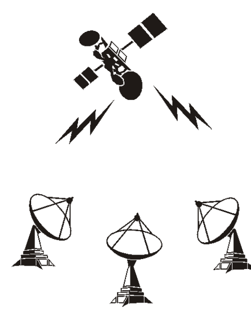
- Point-to-point: dedicated pairwise communication
  - Long-distance fiber link
  - Point-to-point link between Ethernet switch and host
- Broadcast: shared wire or medium
  - Traditional Ethernet
  - 802.11 wireless LAN



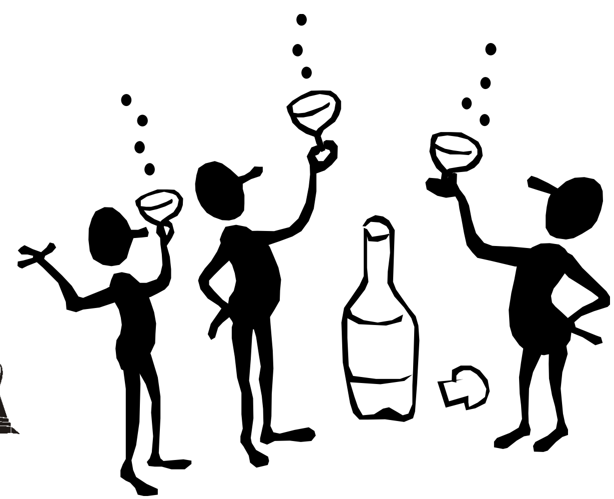
shared wire
(e.g. Ethernet)



shared wireless
(e.g. Wavelan)

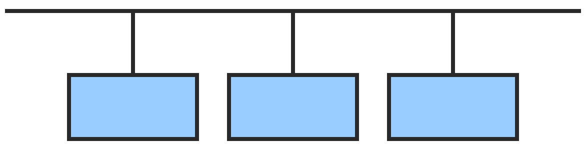


satellite



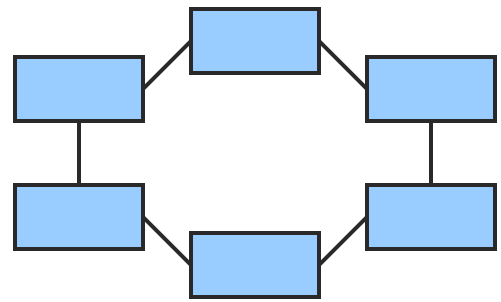
cocktail party

# Types of Shared Link Networks



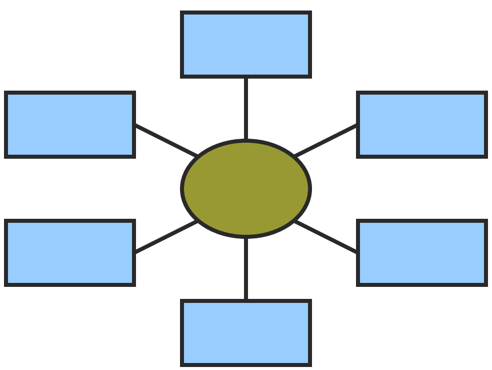
Bus Topology: Shared

Ethernet, Token Bus



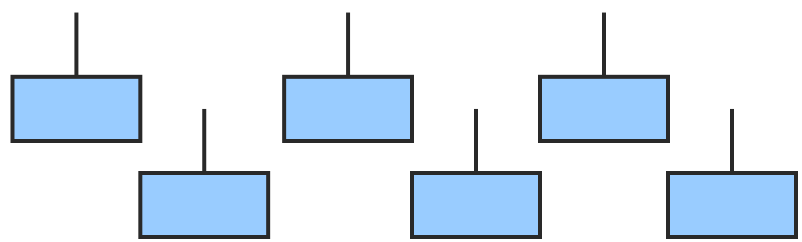
Ring Topology: Multihop

FDDI, IEEE 802.5



Star Topology: Active or Passive Hub

ATM



Wireless: Shared

IEEE 802.11

# Multiple Access Algorithm

- Single shared broadcast channel
  - Must avoid having multiple nodes speaking at once
  - Otherwise, collisions lead to garbled data
  - Need distributed algorithm for sharing the channel
  - Algorithm determines which node can transmit

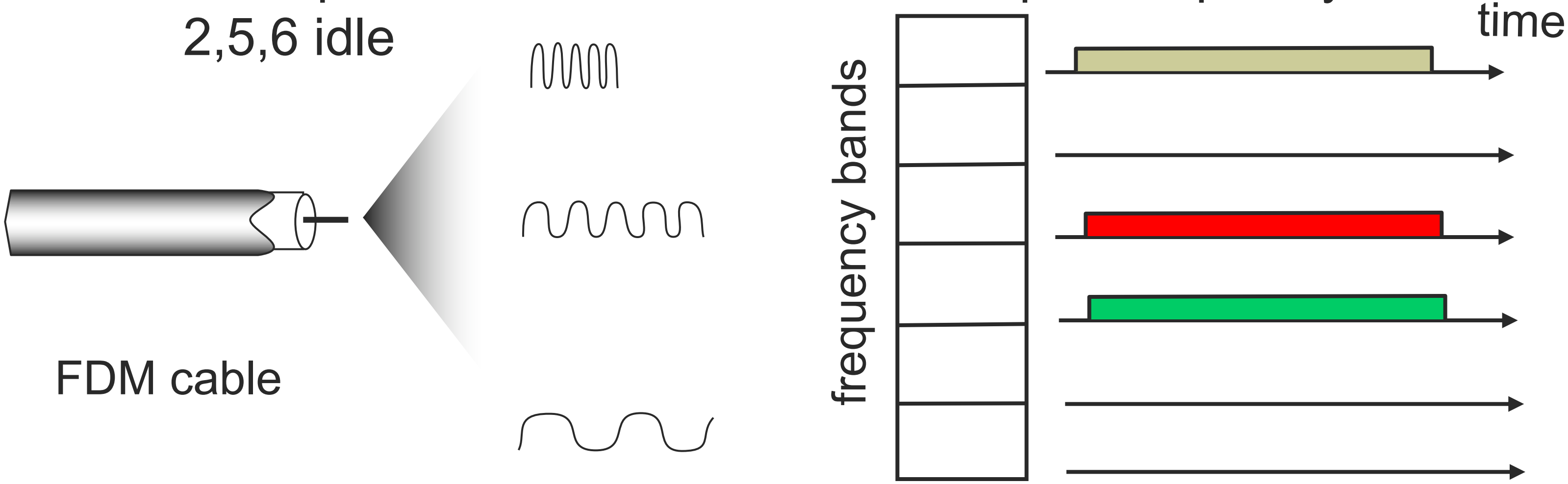
- Typical assumptions
  - Communication needs vary
    - Over time
    - Between hosts
  - Network is not fully utilized

# Multiple Access Media

- Which kind of multiplexing is best?
  - Channel partitioning: divide channel into pieces
    - Frequency-division multiplexing (FDM, separate bands)
  - Taking turns: scheme for trading off who gets to transmit
    - Time-division multiplexing (TDM, synchronous time slots)
    - Statistical time-division multiplexing (STDM, time slots on demand)
  - Random access: allow collisions, and then recover

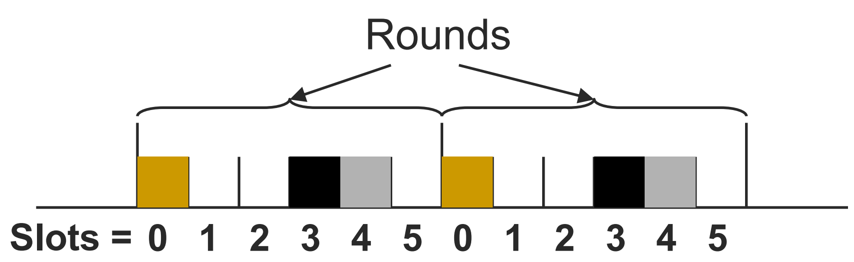
# Channel Partitioning: FDMA

- FDMA: Frequency Division Multiple Access
  - Channel spectrum divided into frequency bands
  - Each station assigned fixed frequency band
  - Unused transmission time in frequency bands go idle
  - Example: 6-station LAN, 1,3,4 have pkt, frequency bands 2,5,6 idle



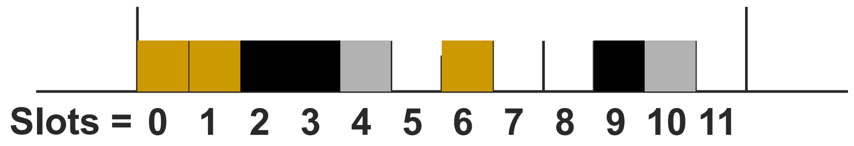
# Channel Partitioning: TDMA

- 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 0, 3, and 4



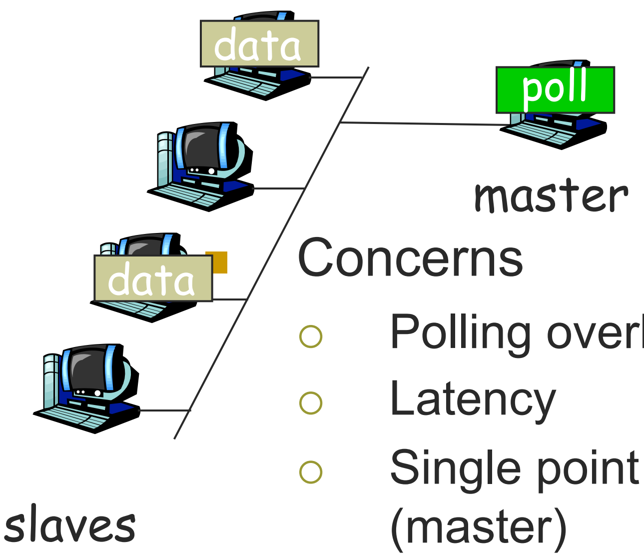
# Channel Partitioning: STDMA

- STDMA: Statistical Time Division Multiple Access
  - Access to channel as needed
    - Each station gets fixed length on transmission
  - Time-slot length is packet transmission time
    - Unused slots go idle only if no station has data to send
  - Example



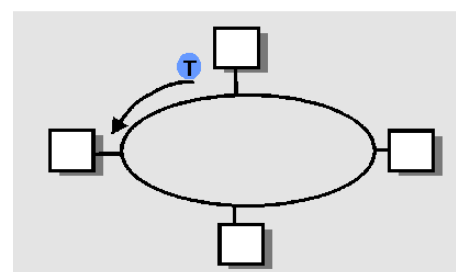
# Channel Partitioning: Taking Turns

- 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
- Node must have token to send
- Concerns
  - Token overhead
  - Latency
  - At mercy of any node



# Multiple Access Media: Random Access

- Random access
  - Optimize for the common case (no collision)
  - Don't avoid collisions, just recover from them….
- When node has packet to send
  - Transmit at full channel data rate
  - No a priori coordination among nodes
- Two or more transmitting nodes $\Rightarrow$ collision
  - Data lost
- Random access MAC protocol specifies
  - How to detect collisions
  - How to recover from collisions

# Multiple Access Media to Discuss

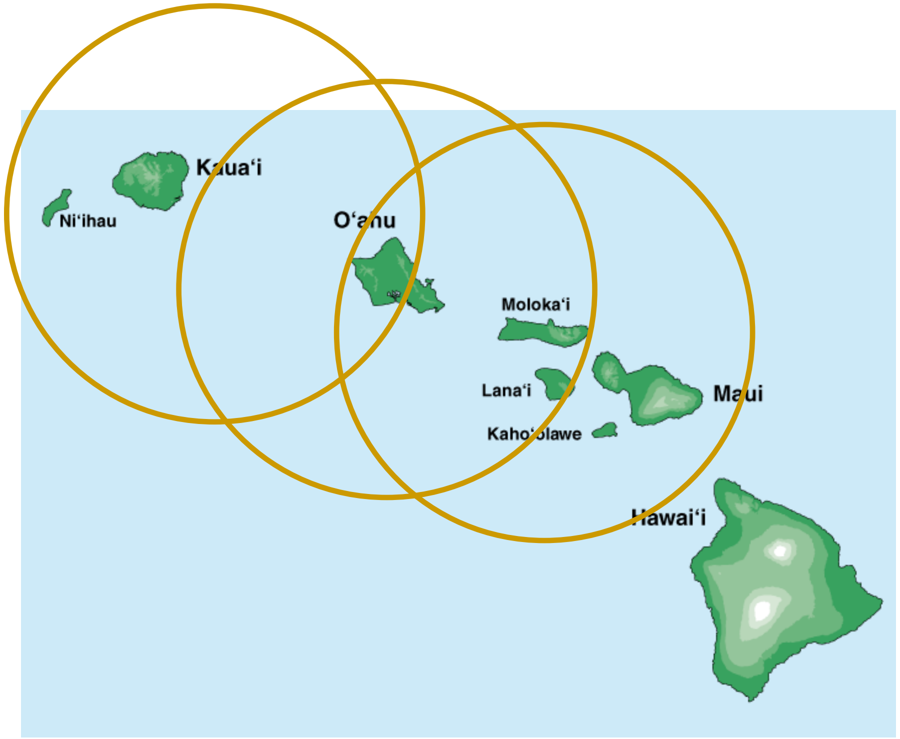
- Two solutions (of many)
  - Carrier sense multiple access with collision detection (CSMA/CD)
    - Send only if medium is idle
    - Stop sending immediately if collision detected
  - Carrier sense multiple access with collision Avoidance (CSMA/CA)
    - Send only if medium is idle
    - Design send algorithm to avoid collisions
- Tanenbaum Sec. 4.2 covers many others

# Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

- Used by Ethernet
  - Xerox and IEEE 802.3 (10Mbps standards)
  - IEEE 802.3u (Fast Ethernet, 100Mbps standard)
  - IEEE 802.3z,ab (1Gbps Ethernet)
  - IEEE 802.3-2005/8 (10 Gbps Ethernet, no shared bus)
- Outline
  - Historical development
  - Topologies and components
  - MAC algorithm
  - Collision detection limitations
  - Lessons learned

# Aloha, Networking!

- Aloha Packet Radio Network
  - Norm Abramson left Stanford to surf
  - Set up first data communication system for Hawaiian islands
  - Hub at U. Hawaii, Oahu
  - Two radio channels:
    - Random access: for sites sending data
    - Broadcast for hub rebroadcasting data



# Pure ALOHA

- Keep it simple
  - User transmits at will
  - If two or more messages overlap in time → collision
    - Receiver cannot decode packets
  - Wait roundtrip time plus a fixed increment → collision
    - Lack of ACK
  - After a collision
    - Colliding stations retransmit
    - Stagger attempts randomly to reduce repeat collisions
  - After several attempts, senders give up
- Simple but wasteful
  - Max efficiency of at most 1/(2e) = 18%!

# From Aloha comes Ethernet

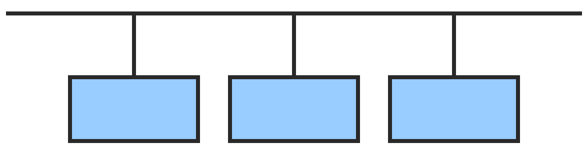
- Ethernet
  - Developed by Xerox PARC,  1974
  - Standardized by Xerox, DEC and Intel in 1978
  - Later, IEEE 802.3 standard
  - Fast Ethernet (100 Mbps)  - IEEE 802.3u standard
  - Switched Ethernet now popular
- Numerous standards with increasing bandwidth over the years
  - 10 Mbps – 100 Mbps – 1 Gbps – 10 Gbps

Xerox Alto, first machine networked with ethernet

# Ethernet - CSMA/CD

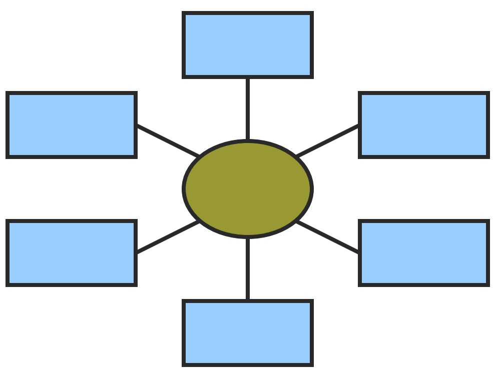
- CS – Carrier Sense
  - Nodes can distinguish between an idle and a busy link
- MA - Multiple Access
  - A set of nodes send and receive frames over a shared link
- CD – Collision Detection
  - Nodes listen during transmission to determine if there has been interference

# Ethernet Topologies



Bus Topology: Shared

All nodes connected to a wire



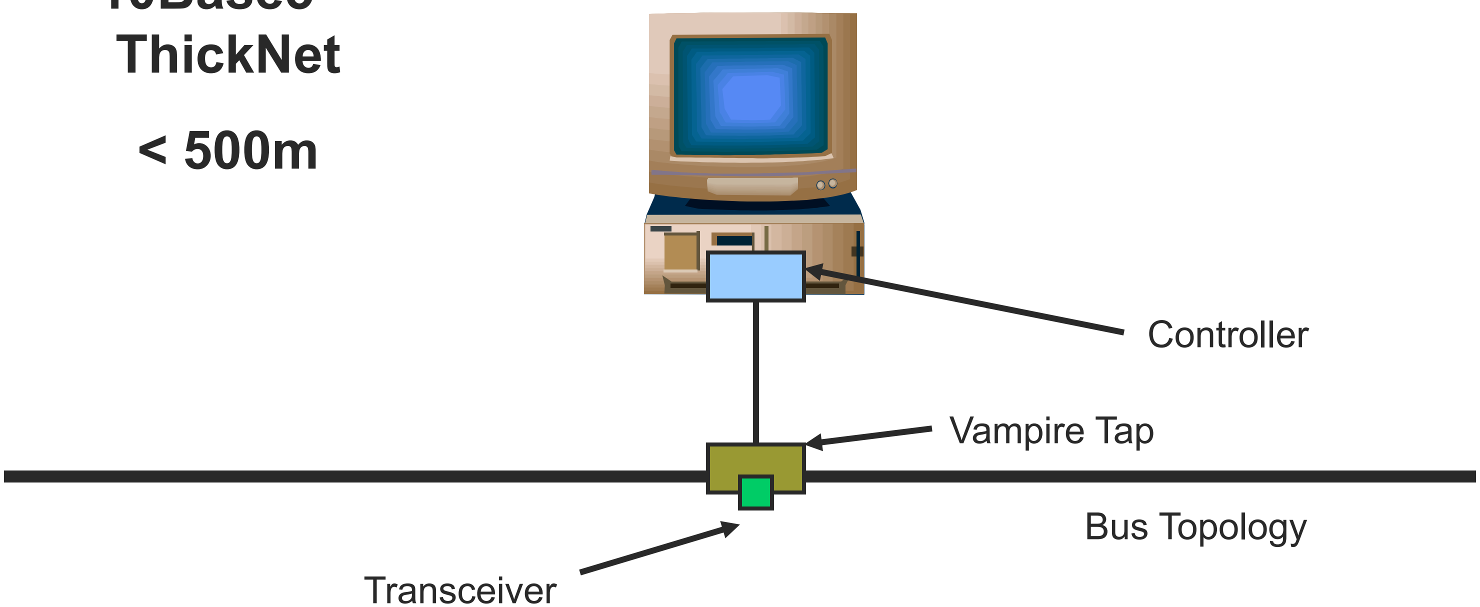
Star Topology:

All nodes connected to a central repeater

# Ethernet Connectivity

**10Base5 – ThickNet**

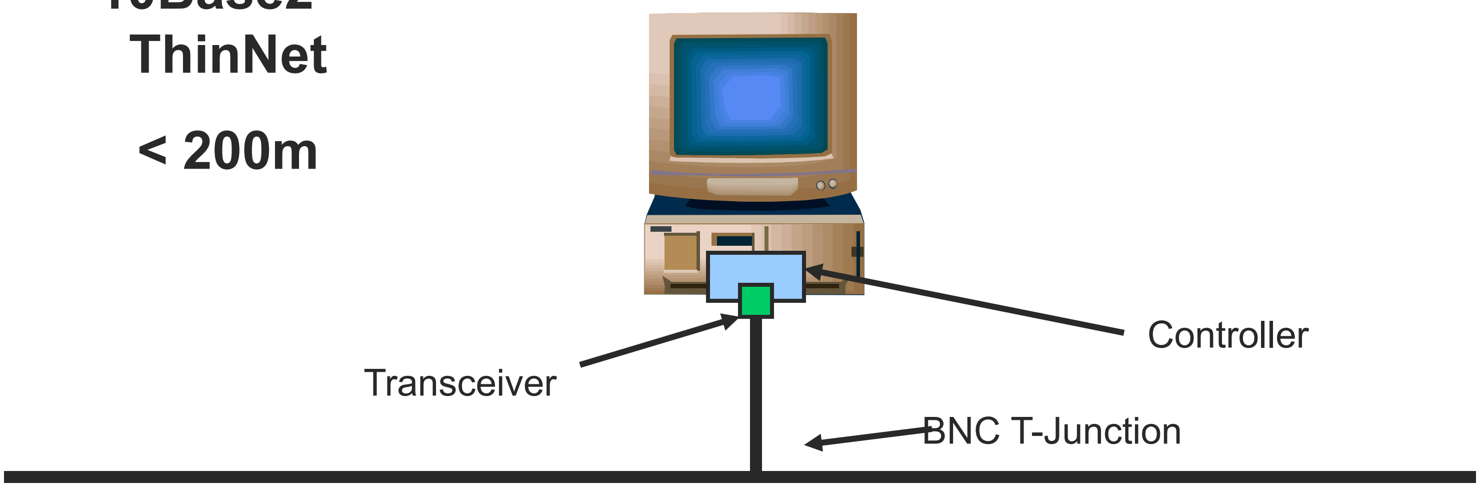
**< 500m**



# Ethernet Connectivity

**10Base2 – ThinNet**

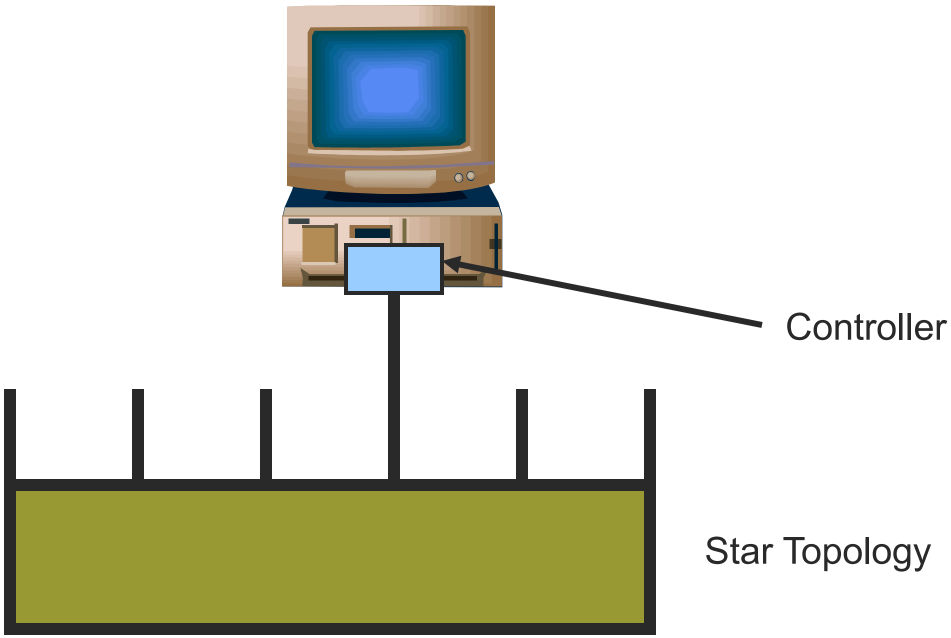
**< 200m**



# Ethernet Connectivity

**10BaseT**

**< 100m**



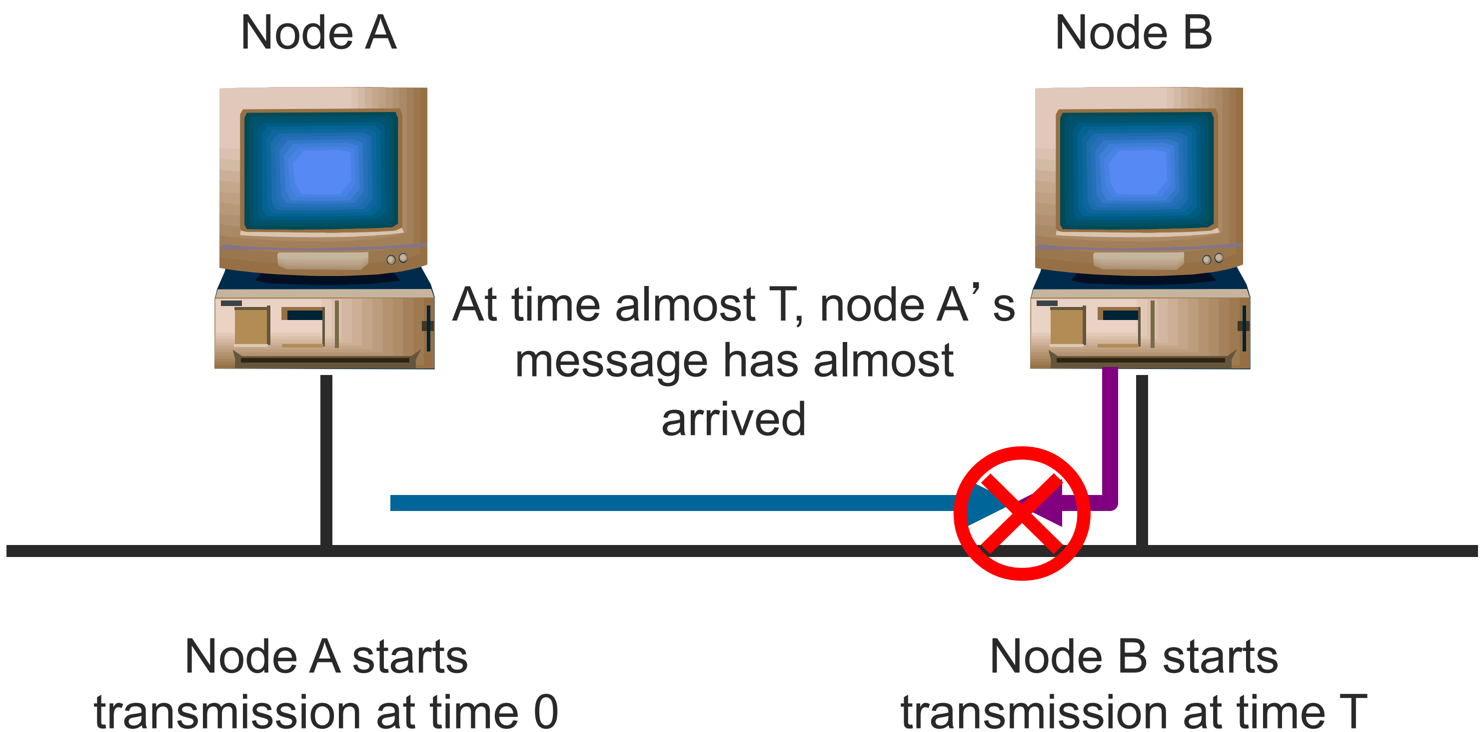
# Ethernet Specifications

- Coaxial Cable
  - up to 500m
- Taps
  - > 2.5m apart
- Transceiver
  - Idle detection
  - Sends/Receives signal
- Repeater
  - Joins multiple Ethernet segments
  - < 5 repeaters between any two hosts
- < 1024 hosts

# 10Mb Ethernet Specifications

- Broadcast
- Encoding
  - Manchester
  - 10 Mbps ⇒ Transmission at 20Mhz
  - Faster Ethernet standards use more efficient encodings
- Framing
  - Preamble marks beginning, sentinel marks end of frame
  - Bit oriented (similar to HDLC)
  - Data-dependent length
- Error Detection
  - 32-bit CRC

# Ethernet MAC Algorithm

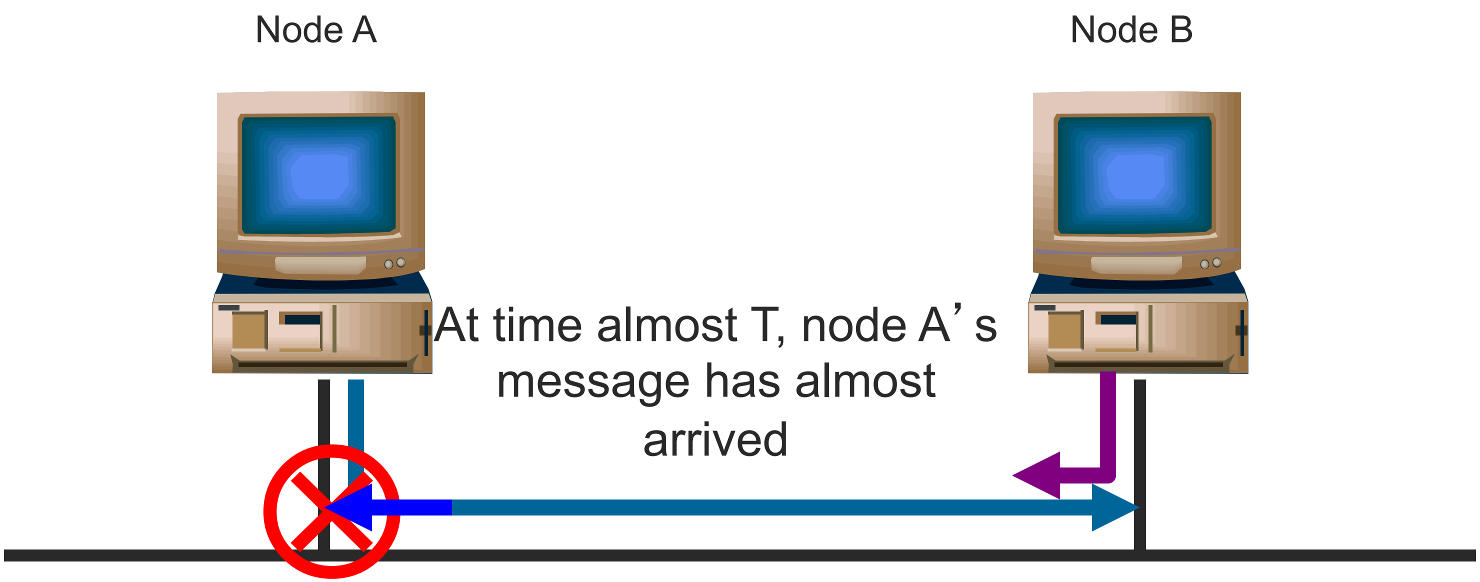



How can we ensure that A knows about the collision?

# Collision Detection

- Problem
  - How can A detect a collision?
- Solution
  - A must still be transmitting when it receives B's transmission!
- Example
  - Node A's message reaches node B at time T
  - Node B's message reaches node A at time 2T
  - For node A to detect a collision, node A must still be transmitting at time 2T

# Ethernet MAC Algorithm




Node A starts transmission at time 0

Node B starts transmission at time T

At time 2T, A is still transmitting and notices a collision

# Collision Detection

- IEEE 802.3
  - 2T is bounded to 51.2μs
  - At 10Mbps 51.2μs = 512b or 64 = 512b or 64B
  - Packet length ≥ 64B
- Jam after collision
  - Ensures that all hosts notice the collision

# Ethernet MAC Algorithm

- Sender/Transmitter
  - If line is idle (carrier sensed)
    - Send immediately
    - Send maximum of 1500B data (1527B total)
    - Wait 9.6 μs before sending again
  - If line is busy (no carrier sense)
    - Wait until line becomes idle
    - Send immediately (1-persistent)
  - If collision detected ← How do we do this?
    - Stop sending and jam signal
    - Try again later ← How do we do this?

Why have a max size?

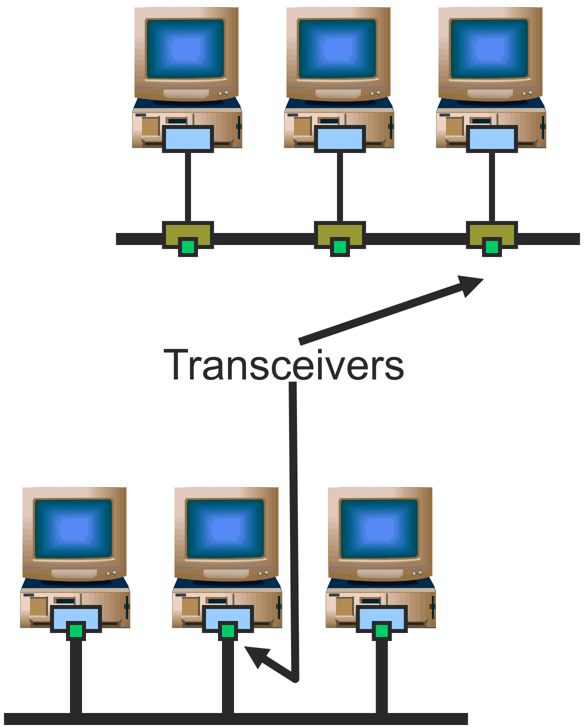
Want to prevent one node from taking over completely

Why 9.6 μs?

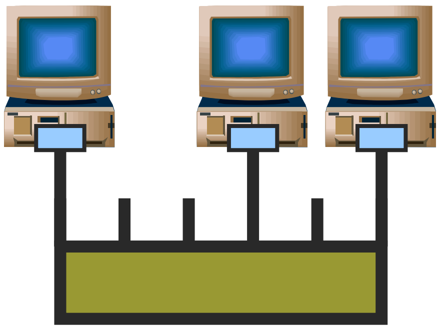
Too long: wastes time
Too short: doesn't allow other nodes to transmit (fairness)

# Collision Detection Techniques: Bus Topology

- Transceiver handles
  - Carrier detection
  - Collision detection
  - Jamming after collision
- Transceiver sees sum of voltages
  - Outgoing signal
  - Incoming signal
- Transceiver looks for
  - Voltages impossible for only outgoing



# Collision Detection Techniques: Hub Topology



- Controller/Card handles
  - Carrier detection
- Hub handles
  - Collision detection
  - Jamming after collision
- Need to detect activity on all lines
  - If more than one line is active
    - Assert collision to all lines
    - Continue until no lines are active

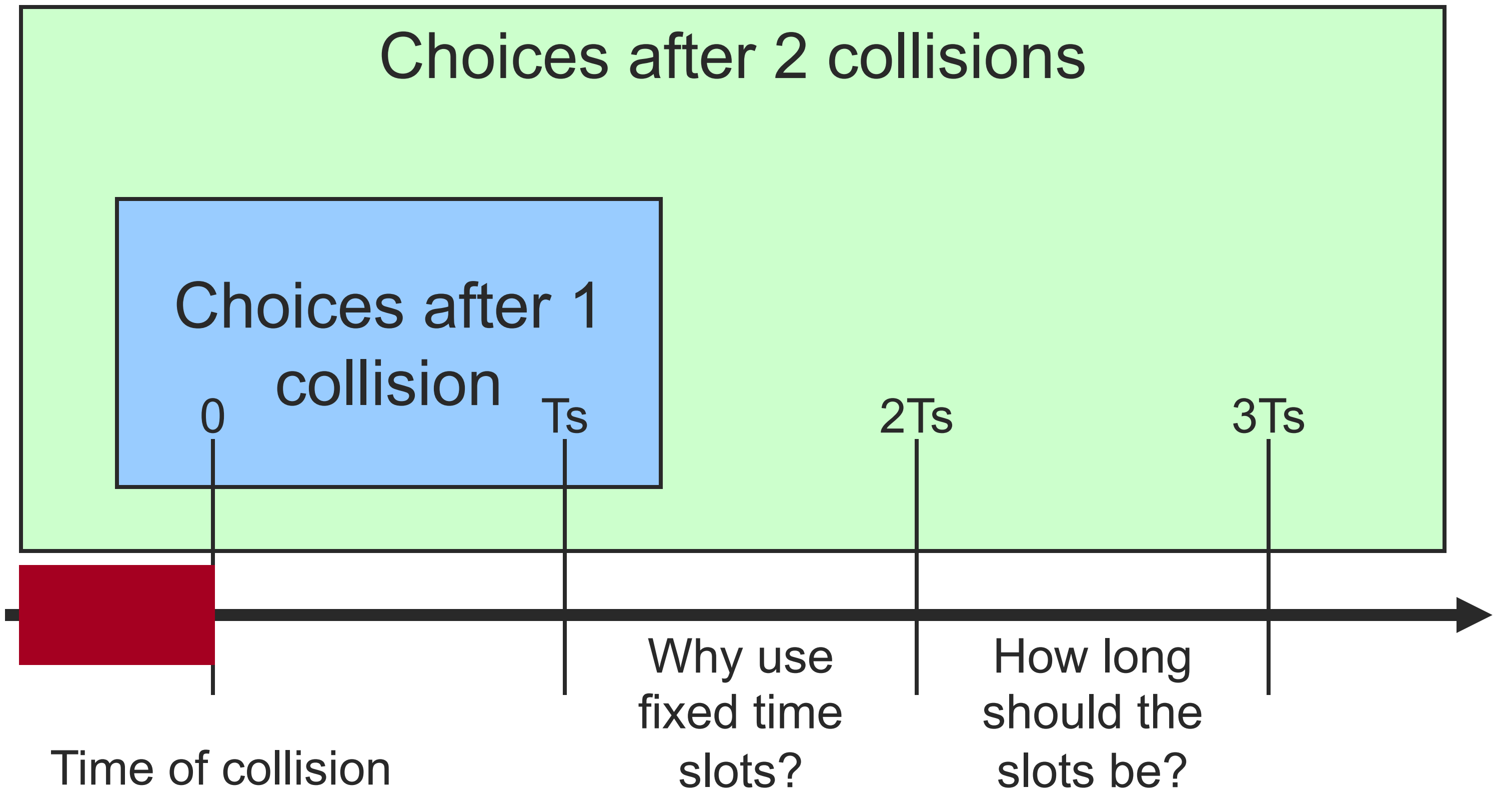
# Frame Reception

- Sender handles all access control
- Receiver simply pulls the frame from the network
- Ethernet controller/card
  - Sees all frames
  - Selectively passes frames to host processor
- Acceptable frames
  - Addressed to host
  - Addressed to broadcast
  - Addressed to multicast address to which host belongs
  - Anything (if in promiscuous mode)
    - Need this for packet sniffers/TCPDump

# Retransmission

- How long should a host wait to retry after a collision?
- What happens if the host waits too long?
  - Wasted bandwidth
- What happens if the host doesn't wait long enough?
  - More collisions
- Ethernet Solution
  - Binary exponential backoff
    - Maximum backoff doubles with each failure
    - After N failures, pick an N-bit number
    - $2^N$ discrete possibilities from 0 to maximum

# Binary Exponential Backoff



# Binary Exponential Backoff

- For IEEE 802.3, T = 51.2 μs
- Consider the following
  - *k* hosts collide
  - Each picks a random number from *0* to $2^{(N-1)}$
  - If the minimum value is unique
    - All other hosts see a busy line
    - Note: Ethernet RTT < 51.2 μs
  - If the minimum value is not unique
    - Hosts with minimum value slot collide again!
    - Next slot is idle
    - Consider the next smallest backoff value

# Binary Exponential backoff algorithm

- When collision first occurs
  - Send a jamming signal to prevent further data being sent
- Resend a frame
  - After either $0$ or $T$ seconds, chosen at random
- If resend fails, resend the frame again
  - After either $0$, $T$, $2T$, or $3T$ seconds.
  - In other words, send after $kT$ seconds, where k is a random integer with $0 \leq k < 2^2$
- If that still doesn't work, resend the frame again
  - After $kT$, where k is a random number with $0 \leq k < 2^3$
- In general, after the $n^{th}$ failed attempt, resend the frame after $kT$, where $k$ is a random number and $0 \leq k < 2^n$

# 10 Mbps Ethernet Example

- Two nodes are ready to send a packet at the same time a third ends transmission
- $i^{th}$ round
  - Each nodes wait [0, 1, …, $2^{(i-1)}$ – 1] slots until next attempt
    - $1^{st}$ round choices: 0
    - $2^{nd}$ round choices: 0, 1
    - $3^{rd}$ round choices: 0, 1, 2, 3
  - All $2^{i-1}$ choices have equal probability
- $q_i$ = P[collision in the $i^{th}$ round]
  - Assuming collisions in all the previous $i$ – 1 rounds

# 10 Mbps Ethernet Example

- Find $q_i$ as a function of $i$ for all $i \geq 1$
  - There are $2^{(i-1)}$ slots to chose from
  - Station A selects a slot with probability $1/2^{(i-1)}$
  - Station B selects a slot with probability $1/2^{(i-1)}$
  - And the same slot is chosen with probability $1/2^{(i-1)}$
  - This probability doesn't depend on the slot the first station selected, so the unconditional probability is

    $q_i = 2^{-(i-1)}$

# 10 Mbps Ethernet Example

- Find the probability $p_i$ that exactly $i$ rounds are needed for the first success
  - $p_i = q_1 q_2 q_3 \ldots q_{i-1} (1 - q_i)$
- Compute $p_1$, $p_2$, $p_3$, $p_4$ and $p_5$
  - $p_1 = 1 - q_1 = 0$
  - $p_2 = q_1 \times (1 - q_2) = 1 \times (1 - ½)$
  - $p_3 = q_1 \times q_2 \times (1 - q_3) = 1 \times ½ \times (1 - ¼)$
  - $p_4 = q_1 \times q_2 \times q_3 \times (1 - q_4) = 1 \times ½ \times ¼ \times (1 - 1/8)$
  - $p_5 = q_1 \times q_2 \times q_3 \times q_4 \times (1 - q_5) = 1 \times ½ \times ¼ \times 1/8 \times (1 - 1/16)$

# 10 Mbps Ethernet Example

- Give an *upper bound* on the probability it takes more that 20 *ms* until the first success

- Slot duration is 51.2$\mu s$
  - 20 ms = 390 slots
- Maximum probability → smallest number of collisions
  - Both stations must wait maximum amount of time
  - Delay for eight collisions =
    1 + 2+ 4 + 8 + 16 + 32 + 64 + 128 = 255
  - Delay for nine collisions =
    1 + 2+ 4 + 8 + 16 + 32 + 64 + 128 + 256 = 511

# 10 Mbps Ethernet Example

- Give an *upper bound* on the probability it takes more that 20 *ms* until the first success

- At least 8 collisions → delay >= 20 ms
- P(delay >= 20 ms) =
  $q_1 q_2 q_3 q_4 q_5 q_6 q_7 q_8 = 2^{-(0+1+2+3+4+5+6+7)}$

# 10Mbps Ethernet Media

| Name | Cable | Advantages | Max. Segment Length | Max Nodes on Segment |
|---|---|---|---|---|
| **10Base5** | Thick Coaxial (10mm) | Good for backbones | 500m | 100 |
| **10Base2** | Thin Coaxial (5mm) | Cheapest system | 200m | 30 |
| **10BaseT** | Twisted Pair (0.5mm) | Easy Maintenance | 100m | 1 (to hub) |
| **10BaseFP** | Fiber (0.1mm) | Best between buildings | 500m | 33 |
| Extended segments may have up to 4 repeaters (total of 2.5km) | | | | |

# 10Mbps Ethernet Media

| Name | Cable | Advantages | Max. Segment Length | Max Nodes on Segment |
|---|---|---|---|---|
| **10Base5** | Thick Coaxial (10mm) | Good for backbones | 500m | 100 |
| **10Base2** | | | 200m | 30 |
| **10BaseT** | (0.5mm) | Maintenance | 100m | 1 (to hub) |
| **10BaseFP** | Fiber (0.1mm) | Best between buildings | 500m | 33 |
| Extended segments may have up to 4 repeaters (total of 2.5km) | | | | |

The fixed T defines the maximum segment length

# 100Mbps Ethernet Media

| Name | Cable | Max. Segment Length | Advantages |
|---|---|---|---|
| **100BaseT4** | 4 Twisted Pair | 100m | Cat 3, 4 or 5 UTP |
| **100BaseTX** | Twisted Pair | 100m | Full duplex on Cat 5 UTP |
| **100BaseFX** | Fiber Pair | 100m | Full duplex, long runs |
| All hub based. Other types not allowed. Hubs can be shared or switched | | | |

# 100Mbps Ethernet Media

| Name | Cable | Max. Segment Length | Advantages |
|---|---|---|---|
| **100BaseT4** | 4 Twisted Pair | 100m | Cat 3, 4 or 5 UTP |
| **100Bas** | | 100m | Full duplex on Cat 5 UTP |
| **100BaseFX** | Fiber Pair | 100m | Full duplex, long runs |
| All hub based. Other types not allowed. Hubs can be shared or switched | | | |

Shorter distances, same protocol!

# ..and beyond

- Gigabit ethernet common in PCs
- 100 GB ethernet standard ratified in June 2010

# Ethernet in Practice

- Number of hosts
  - Limited to 200 in practice, standard allows 1024
- Range
  - Typically much shorter than 2.5km limit in standard
- Round Trip Time
  - Typically 5 or 10 $\mu$s, not 50
- Flow Control
  - Higher level flow control limits load (e.g. TCP)
- Topology
  - Star easier to administer than bus
  - Even better: exclusive access rather than shared!