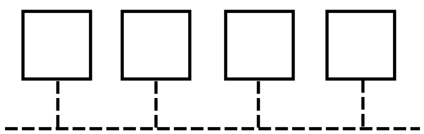
MEDIUM ACCESS CONTROL (MAC)

Direct Links –

- point-to-point link
- **multiple access (broadcast, random access)** channel



Channel Allocation Problem – how to allocate a single broadcast channel among competing users

Difficulty – only a *single* channel is available

Solution – Multiple Access (or MAC) Protocols

- A set of rules employed *independently* by each multi-access user to gain access to the channel

$\Rightarrow$ MAC Sub-Layer $\in$ Data Link Layer

Types of networks – LAN and Wireless

Desirable characteristics

- When only one node has data to send, that node has a throughput of R bps
- When M nodes have data to send, each node has an average throughput of R/M bps
- The protocol is decentralized (no single point of failure)
- The protocol is simple, so that it is inexpensive to implement

Types of MAC Protocols

- Fixed Assignment (Channel Partitioning) Protocols
  - Time Division Multiple Access (TDMA)
  - Frequency Division Multiple Access (FDMA)
  - Code Division Multiple Access (CDMA)

  Bad if user # is large/dynamic or traffic is bursty
- Random Assignment (Random Access) Protocols
  - ALOHA
  - Carrier Sense Multiple Access (CSMA)
  - CSMA with Collision Detection (CSMA/CD)
- Demand Assignment (Taking-Turns) Protocols
  - Polling
  - Token Passing

Standards – IEEE 802

Random Access Protocols

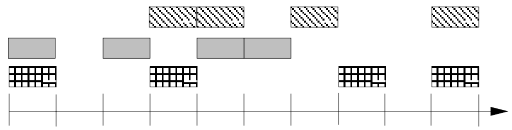
Principles

- A transmitting node always transmits at the full rate of the channel
- When there is a collision, each node involved *repeatedly* retransmit its frame until the frame gets through without a collision
- It waits a *random* delay before retransmitting the frame
- Each node involved in a collision chooses *independent* random delays

Slotted ALOHA – ground-based broadcast radio

- Feedback property of broadcasting – listen to detect damaged farmes
- Assumptions
  - All frames consist of exactly L bits
  - **Time is divided into slots of size L/R seconds**
  - Nodes start to transmit frames only at the *beginnings* of slots
  - The nodes are synchronized so that each node knows when the slots begin
  - If two or more frames collide in a slot, then all the nodes detect the collision before the slot ends

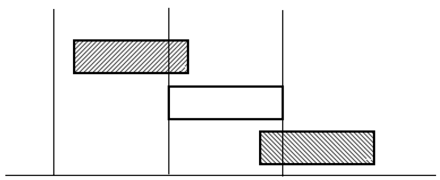
- Protocol
  - When a node has a fresh frame to send, it waits until the beginning of the next slot and transmits the entire frame in the slot
  - If there is a collision, the node detects the collision before the end of the slot. The node retransmits its frame in each subsequent slot with probability p until the frame is transmitted without a collision



- Efficiency
  - Definition – the long-term fraction of successful slots when there are a large number of active nodes, with each node having a large number of frames to send
  - Assume each node always has a frame to send and transmits a frame in each slot with probability p
  - Suppose there are N nodes
  - The probability a given node has a success is $p(1-p)^{N-1}$
  - The probability that an arbitrary node has a success is $Np(1-p)^{N-1}$
  - The probability is maximized when $p = 1/N$
    $\lim_{N\to\infty} Np(1-p)^{N-1} = 1/e \approx 0.368$

Pure ALOHA – unslotted and fully decentralized

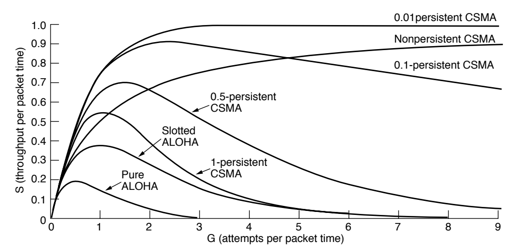
- Protocol
  - A frame is immediately tx'ed whenever it is available
  - If collision, after completely transmitting its collided frame, the node immediately retx the frame with probability p.
  - Otherwise, the node waits for a frame tx time. After this wait, it retx the frame with p, or waits for another frame time with $1-p$



- Efficiency
  - At any given time, a node tx a frame with prob. p
  - Suppose this frame begins transmission at time T_0
  - For this frame to be successfully tx'ed, no other nodes can begin their tx in interval $[T_0 - 1, T_0] \Rightarrow$ prob. $(1-p)^{N-1}$
  - No other node can begin tx while the node is tx'ing $\Rightarrow$ probability $(1-p)^{N-1}$
  - The prob. that a given node has a successful tx is $p(1-p)^{2(N-1)}$
  - Maximum efficiency $\lim_{N\to\infty} Np(1-p)^{2(N-1)} = 1/(2e)$

Carrier Sense Multiple Access (CSMA)

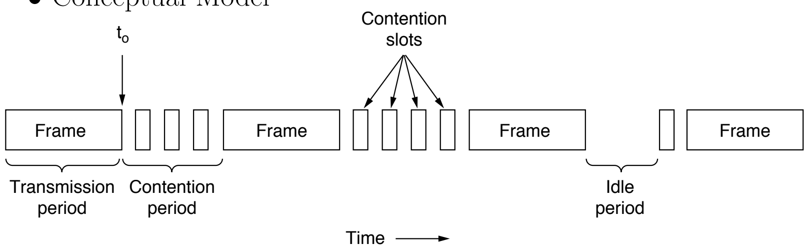
- **Carrier Sense – listen before transmit**
  vs. ALOHA's *tx at will*
- 1-Persistent CSMA
  - tx with probability 1 whenever it finds the channel idle
  - if collision, wait a random time and start all over again
  - **propagation delay** is the bad guy...
- Non-Persistent CSMA
  - sense the channel $\rightarrow$ if idle, transmit
  - if busy, do not continually sense channel for idleness
    $\Rightarrow$ simply wait a random time and repeat



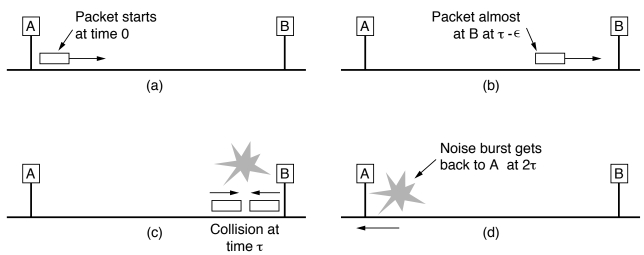
- p-Persistent CSMA
  - if idle, tx with probability p, or defer to the next slot with probability $1-p$ and try again

CSMA with Collision Detection (CD)

- **Collision Detection** – abort a tx as soon as a collision is detected, rather than finish tx'ing garbled frames
- Conceptual Model



- Key question – **how long will it take to realize that there has been a collision?**



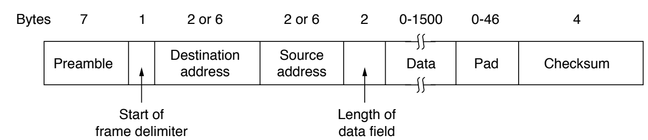
Answer – a station cannot be sure that it has seized the channel until it has tx'ed for 2τ without hearing a collision

IEEE 802.3 and Ethernet

- **1-persistent CSMA/CD**
- Cabling

Name	Cable	Max. Seg.	Nodes/seg.
10Base5	thick coax	500m	100
10Base2	thin coax	200m	30
10Base-T	twisted pair	100m	1024
10Base-F	optical fiber	2000m	1024

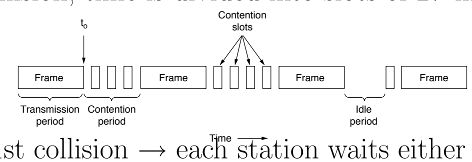
- **Repeater** – **physical layer device** to receive, amplify, and retransmit signals in both directions
- For 10Base5 – **max. # of repeaters: 4** $\Longrightarrow$ **max. length: 2500m**
- **Frame Format – minimum 64 bytes in length**



- Addresses:
  - **Unique**, 48-bit unicast address assigned to each adaptor
  - Example – `8:0:2b:e4:b1:2`
  - Multicast: first bit is `1`
  - Broadcast: all `1`s

Collision ⇒ Binary Exponential Backoff Algorithm

After a collision, time is divided into slots of 2τ in length



- after the 1st collision → each station waits either 0 or 1 slot times before trying again
- after the 2nd collision → each station waits either 0, 1, 2, or 3 slot times before trying again
- after the ith collision → each station waits for a slot time between 0 and $2^i - 1$ before trying again
- after the 10th collision → each station waits for a slot time between 0 and 1023 ($2^{10} - 1$) before trying again
- after the 16th collision → give up and report to higher layer

Key − *dynamically* **adapt to # of competing stations**

- if fixed at [0, 1023], chance of collision again is low, but average wait (delay) is long
- if fixed at [0, 1], chance of collision again is high
- randomization interval grows *exponentially* as more and more consecutive collisions occur

  ⇒ low delay for a few competing stations

  ⇒ reasonable delay when many stations collide

802.3 Efficiency for heavy and constant load

$\Longrightarrow$ k stations always ready to transmit

- p = probability of tx during a contention slot
- A = prob. that some station (out of k) acquires the channel in a contention slot $= kp(1-p)^{k-1}$
- **A is maximized to $1/e$ when $p = 1/k$ and $k \to \infty$**
- Prob. that the contention interval has exactly j slots is $A(1-A)^{j-1}$
- Mean # of slots per contention is

$$\sum_{j=0}^{\infty} jA(1-A)^{j-1} = \frac{1}{A}$$

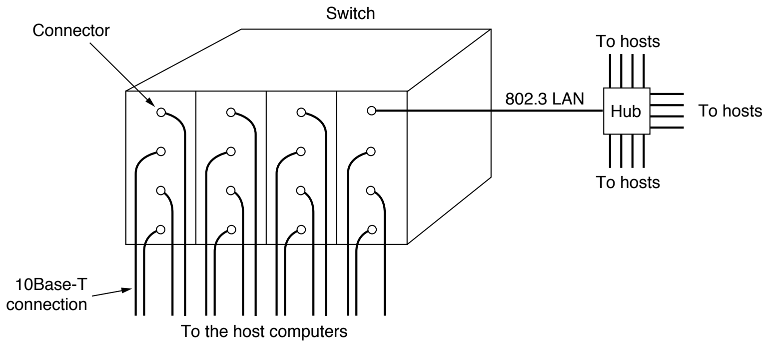
- Duration of a contention slot $= 2\tau$ (51.2 μsec)
  ⇒ **mean contention interval** $= 2\tau/A$
- B(bandwidth) F(frame length) c(speed of signal) L(cable length)
- If the mean farme takes P sec to tx, **channel efficiency** =

$$\frac{P}{P + 2\tau/A} \Longrightarrow \frac{1}{1 + 2BLe/cF}$$

- **Observations – Delay(L)×Bandwidth(B) product**
  - the longer the L, the lower the channel efficiency
  - the higher the B, the lower the channel efficiency

Switched 802.3 LANs

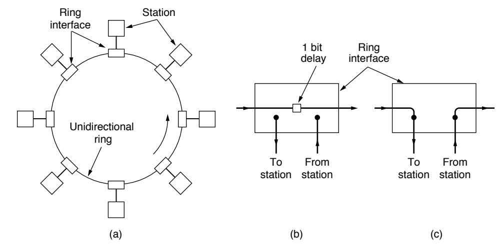
- More stations on 802.3 LAN $\Longrightarrow$ saturate!
- Alternatives –
  - Higher speed adaptor cards $\Rightarrow$ too expensive
  - Switch with **high-speed backplane** – switched LAN



$\Rightarrow$ each card forms a **collision domain**

IEEE 802.5 and Token Ring

- 802.3 – arbitrary (unbounded) delay and no priority $\Longrightarrow$ not suitable for real-time systems
- **Ring – fair and known upper-bound on channel access**

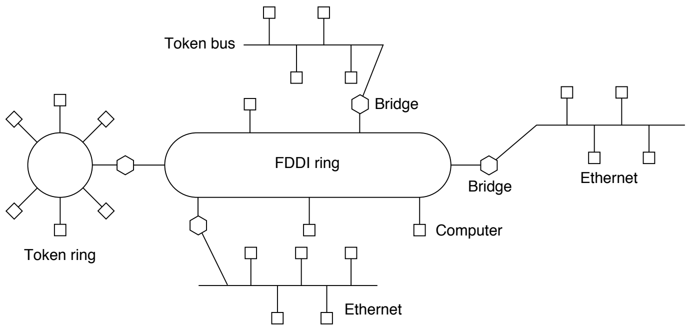


- **Token** – special bit pattern circulate around the ring whenever all stations are idle
- To tx data, **seize the token** and convert it into **start of frame**
- **Operating modes of ring interface**
  - Listen – copy input bits to output
  - Transmit – enter data into the ring
- Frames are removed by the sender
- **Regenerate the token after finish transmitting the frame**
- **Permission to send rotates smoothly aound the ring in a round-robin fashion**

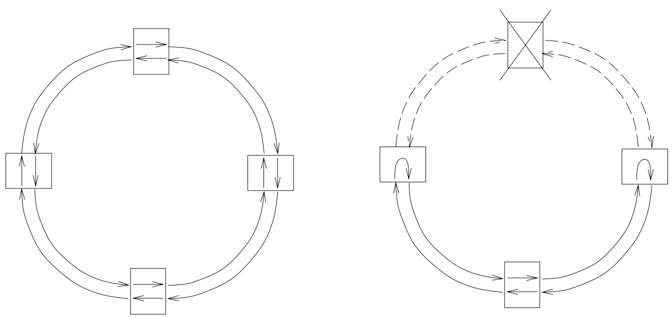
High-Speed LAN – FDDI

Fiber Distributed Data Interface

- Fiber optic token ring running at 100Mbps

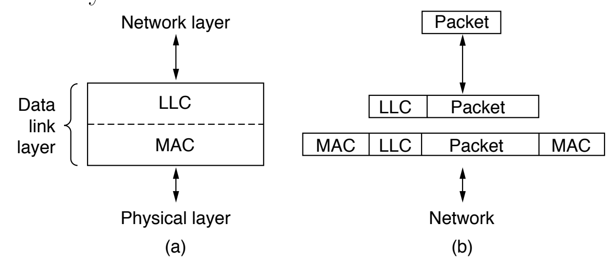


- **Dual ring** configuration – **self-healing**



IEEE 802.2 – Logical Link Control

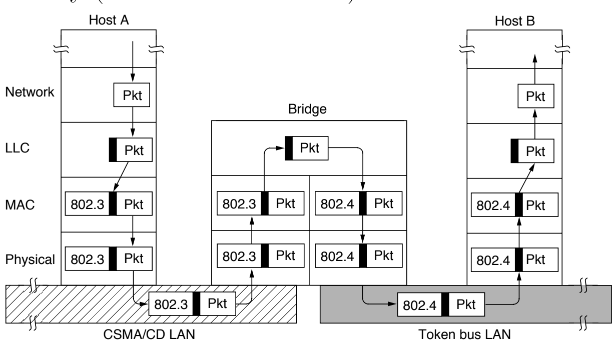
- Functionality of Data Link Layer – provide reliable communication over unreliable links using DLL protocols
  - error control – ACK
  - flow control – sliding window
- IEEE 802 offers **best-efforts datagram service**
- Logical Link Control (LLC) – run on top of all 802 protocols by providing a single format, interface, and protocol to the network layer



- LLC header – sequence # and ACK
- Services
  - unreliable datagram (no sequence # and no ACK)
  - ACKed datagram
  - reliable connection-oriented

Bridges

- LANs have physical limitations $\Longrightarrow$ multiple LANs
- Connect two or more LANs with a **bridge** in the Data Link Layer layer (does not examine Network Layer header)
- Why we have multiple LANs
  - Autonomy (of each organization)
  - Cost (cheaper than having one coax cable)
  - Locality (to accommodate the load)
  - Distance (too far apart)
  - Reliability (isolation like a fire door)
  - Security (isolation like firewall)



- Operations of a bridge (how it works)
- multiple LANs + bridges $\equiv$ Extended LAN

A bridge connecting k different LANs will have k different MAC sublayers

Bridges from 802.x to 802.y

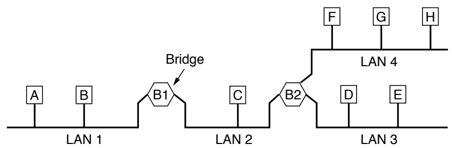
- Common problems
  - Each of the LANs uses a different frame format
    - ∗ Reformat $\Rightarrow$ calculate new checksum
  - Bridged LANs may not run at the same data rate
    - ∗ Faster LANs may swamp slower LANs
    - ∗ Bottleneck bridges may cause timeout in higher layers
  - LANs have different maximum frame lengths
    - ∗ A frame is a frame which cannot be further splitted
- Specific problems for (802.x $\rightarrow$ 802.y)

Two approaches to bridge design

- Transparent (spanning tree) bridge
- Source routing bridge

Transparent (Spanning Tree) Bridges

- Goal – **transparency** $\Longrightarrow$ **plug-and-play**
- Operation – operate in **promiscuous mode** and **accept** every frame tx'ed on all the LANs to which it's attached

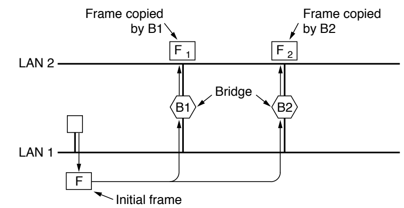


  - **discard** frames when S and D are on the same LAN
  - **forward** frames when S and D are on different LANs $\Longrightarrow$ **forwarding table**

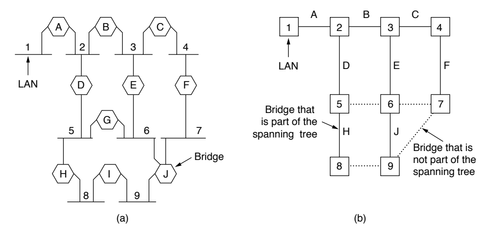
- Forwarding Table
  - Empty when bridge first powered up
  - **Flooding** with $\boxed{\textbf{backward learning}}$
    - ∗ Frame for an unknown destination is output on all LANs attached except the one it arrived on
    - ∗ By looking at **source** address, bridge knows which host is accessible on which LAN
  - Dynamic topology changes
    - ∗ Timestamp each entry
    - ∗ Purge 'old' entries periodically

Parallel Transparent Bridges – for reliability

- Problem – **loops** $\Longrightarrow$ endless cycle for unknown D



- Solution – **spanning tree** bridge
  - Overlay the actual topology with a spanning tree $\Longrightarrow$ loop-free
  - Exactly one path from every LAN to every other LAN
  - All forwarding between LANs follows the spanning tree



- Spanning Tree Algorithm
  - Choose a **root** first
    - ∗ each B broadcasts its serial #
    - ∗ the B with the lowest serial # becomes the root
  - Each B construct a **shortest paths** to the root

Source Routing Bridges

Transparent Bridge

- Good – plug-and-play
- Bad – not optimal use of bandwidth (spanning tree is a *sub-graph* of the original topology)

Source Routing Bridge

- Include in the frame header the exact **path** that the frame will follow
- **Path** – a sequence of LAN, B, LAN, B, LAN, $\cdots$ #
- Routing algorithm
  - B scans the route looking for the # of the LAN on which the frame arrived
  - If this LAN # is followed by B's own B #, forward the frame to the LAN whose # follows the B # on the route
- Requirement – every H knows the best path to every other H
  - If a D is not known, the S broadcasts a **discovery frame** asking where it is
  - When the reply comes back, the B record their # in it
  - S can see the exact route taken and choose the best one
- Suffer from frame explosion $\Longrightarrow$ congestion
  vs. flooding along the spanning tree for Transparent B

Limitations of Bridges

- **Do not scale** – up to tens of LANs
  - spanning tree algorithm provides no hierarchy
  - bridge forwards all broadcast frames
- **Do not accommodate heterogeneity**
  - not able to connect any 802.x with any 802.y
- **Transparency** – *dangerous*
  - no special protocol needed for end hosts – good
  - bridges may be congested to drop frames
  - latency becomes larger and highly variable
  - frames may be reorded in an extended LAN
    vs. frame order is never shuffled on a single Ethernet