Data Link Layer

• **Data Link Layer Functions**
  - deliver frames over a single link
  - framing, media access, error checking (error correction), ...
    - Cyclic Redundancy Code for error detection

• **Local Area Networks (LANs) and MAC Addresses**
  - point-to-point vs. shared access
  - MAC addresses
  - MAC addresses vs. IP addresses
  - IP Address Resolution Protocol (ARP) and IP datagram forwarding (revisited!)
Data Link Layer (cont’d)

• **Media Access Control**
  - Random Access
    • Aloha, Slotted Aloha
    • CSMA, CSMA/CD and Ethernet (802.3)
    • CSMA/CA and 802.11 Wireless LAN (covered in Chapter 6)
  - “Take Turns” Controlled Access
    • Token Passing and Token Ring (802.5)

• **Extending LANs: hubs, bridges, and switches**
  - Bridging self-learning and spanning tree algorithms

• **Other Link Layer Technology**
  - PPP
  - ATM/MPLS and Link/Network Virtualization

**Readings:** Textbook, Chapter 5
Data Link Layer: Introduction

Some terminology:

- hosts and routers are **nodes** (bridges and switches too)
- communication channels that connect adjacent nodes along communication path are **links**
  - wired links
  - wireless links
  - LANs (local area networks)
- layer 2 PDU ("packet") referred to as **frame**, which encapsulates a layer-3 packet, e.g., an IP datagram
What Does Data Link Layer Do?

Data link layer has responsibility of transferring frames from one node to adjacent node over a single link

- An IP packet from host A to host B may traverses different links using different data link protocols
  - e.g., Ethernet on first link, PPP on intermediate links, 802.11 on last link
- Each link protocol provides different services
  - e.g., may or may not provide reliable data delivery
- Different link protocols are not inter-operable!
  - IP packets are encapsulated/decapsulated with appropriate data link protocol header over each link
  - IP protocol and IP routers glue the links (“physical networks”) together and provide end-to-end data delivery!
Data Link Layer Functions

- **Framing**
  - sender (transmitter): encapsulate datagram into frame, adding header, trailer, transmit frame
  - receiver: detect beginning of frames, receive frame, decapsulate frame, stripping off header, trailer

- **Link Access (Media Access Control)**
  - determine whether it’s Okay to transmit over the link
    - particularly important when link shared by many nodes
      - also an issue over “half-duplex” point-to-point link (why?)
    - need media access control (MAC)
  - “physical addresses” identify sender/receiver on a link!
    - particularly important when link shared by many nodes, while over point-to-point link, not necessary (why?)
    - “physical addresses” often referred to as “MAC” addresses
      - different from IP addresses (which are logical & global)!
Other Data Link Layer Functions

- **Error Detection** (commonly implemented, see Section 5.2)
  - errors caused by signal attenuation, noise, etc.
  - sender computes “checksum”, attaches to frame
  - receiver detects presence of errors by verifying “checksum”
    - drops corrupted frame, may ask sender for retransmission
  - Commonly used “checksum”: cyclic redundancy code (CRC)

- **Reliable delivery between adjacent nodes** (optional)
  - using, e.g., go-back-N or selective repeat protocol
    - seldom used on low bit error link (fiber, some twisted pair)
    - wireless links: high error rates
    - Q: why both link-level and end-end reliability?

- **Error Correction** (optional)
  - receiver identifies *and corrects* bit error(s) without resorting to retransmission, using forward error correction (FEC) codes

- **Flow Control** (optional)
  - negotiating transmission rates between two nodes
Adaptors Communicating

- Link layer implemented in “adaptor” (aka NIC), with “transceiver” in it
  - Ethernet card, dial-up modem, 802.11 wireless card
- Sending side:
  - Encapsulates datagram in a frame
  - Adds error checking bits, flow control, reliable data transmission, etc.
- Receiving side
  - Looks for errors, flow control, reliable data transmission, etc.
  - Extracts datagram, passes to receiving node
- Adapter is semi-autonomous
- Data link & physical layers are closely coupled!
Error Detection

EDC = Error Detection and Correction bits (redundancy)
D  = Data protected by error checking, may include header fields

• Error detection not 100% reliable!
  • protocol may miss some errors, but rarely
  • larger EDC field yields better detection and correction
Parity Checking

**Single Bit Parity:**
Detect single bit errors

- **Parity Checking**
  - 0111000110101011 0

**Two Dimensional Bit Parity:**
Detect and correct single bit errors

- **Row Parity**
  - $d_{1,1}, \ldots, d_{1,j} \quad d_{1,j+1}$
  - $d_{2,1}, \ldots, d_{2,j} \quad d_{2,j+1}$
  - $\ldots, \ldots, \ldots, \ldots$
  - $d_{i,1}, \ldots, d_{i,j} \quad d_{i,j+1}$
  - $d_{i+1,1}, \ldots, d_{i+1,j}$
  - $d_{i+1,j+1}$

- **Column Parity**
  - $d_{1,1}$
  - $d_{2,1}$
  - $\ldots$
  - $d_{i,1}$
  - $d_{i+1,1}$

- **Examples**
  - 101011
  - 111100
  - 011101
  - 101010

  **No errors**

  - 101011
  - 101010

  **Correctable single bit error**

  - 101100
  - 011101
  - 101010
Internet Checksum

**Goal:** detect “errors” (e.g., flipped bits) in transmitted segment (note: used at transport layer only)

**Sender:**
- treat segment contents as sequence of 16-bit integers
- checksum: addition (1's complement sum) of segment contents
- sender puts checksum value into UDP checksum field

**Receiver:**
- compute checksum of received segment
- check if computed checksum equals checksum field value:
  - NO - error detected
  - YES - no error detected. But maybe errors nonetheless? More later ....
Checksumming: Cyclic Redundancy Check

- 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 (ATM, HDLC)

$D*: 2^r \text{ XOR } R$

$D$: data bits to be sent  $R$: CRC bits  

bit pattern

mathematical formula
CRC Example

Want:
\[ D \cdot 2^r \text{ XOR } R = nG \]
equivalently:
\[ D \cdot 2^r = nG \text{ XOR } R \]

equivalently:
if we divide \( D \cdot 2^r \) by \( G \), want remainder \( R \)

\[
R = \text{remainder} \left( \frac{D \cdot 2^r}{G} \right)
\]
Multiple Access Links and LANs

Two types of “links”:

• point-to-point, e.g.,
  - PPP for dial-up access, or over optical fibers

• broadcast (shared wire or medium), e.g.
  - traditional Ethernet
  - 802.11 wireless LAN
LAN: Issues & Technologies

• Issues:
  - addressing: physical (or MAC) addresses
  - media access control (MAC) for broadcast LANs
  - expanding LANs: connecting multiple LAN segments
    • hubs, bridges, (layer-2) switches (we’ll explain difference later)

• Various commonly used LAN technologies
  - Ethernet
  - 802.11(WiFi)
  - PPP
  - ATM
    (not really LAN)
    (example of non-broadcast multi-access network)
MAC (Physical, or LAN) Addresses

- used to get frames from one interface to another physically-connected interface (same physical network, i.e., p2p or LAN)
- 48-bit MAC address (for most LANs)
  - fixed for each adaptor, burned in the adapter ROM
  - MAC address allocation administered by IEEE
    - 1st bit: 0 unicast, 1 multicast.
    - all 1’s: broadcast
- MAC flat address -> portability
  - can move LAN card from one LAN to another
- MAC addressing operations on a LAN:
  - each adaptor on the LAN “sees” all frames
  - accept a frame if dest. MAC address matches its own MAC address
  - accept all broadcast (MAC = all 1’s) frames
  - accept all frames if set in “promiscuous” mode
  - can configure to accept certain multicast addresses (first bit = 1)
MAC vs. IP Addresses

32-bit IP address:

- *network-layer address*, logical
  - i.e., not bound to any physical device, can be re-assigned
- IP hierarchical address NOT portable
  - depends on IP network to which an interface is attached
  - when move to another IP network, IP address re-assigned
- used to get IP packets to destination IP network
  - Recall how IP datagram forwarding is performed
- IP network is “virtual,” actually packet delivery done by the underlying physical networks
  - from source host to destination host, hop-by-hop via IP routers
  - over each link, different link layer protocol used, with its own frame headers, and source and destination MAC addresses
    - Underlying physical networks do not understand IP protocol and datagram format!
Recall: IP Datagram Forwarding

Starting at A, given IP datagram addressed to B:

- look up net. address of B, find B on same net. as A
- link layer send datagram to B inside link-layer frame

frame source, dest address

B’s MAC addr  A’s MAC addr

datapgram source, dest address

A’s IP addr  B’s IP addr

IP payload

---

frame

datagram
ARP: Address Resolution Protocol

Question: how to determine MAC address of B knowing B's IP address?

- Each IP node (host, router) on LAN has ARP table
- ARP Table: IP/MAC address mappings for some LAN nodes
  \(< \text{IP address}; \text{MAC address}; \text{timer}>\)
  - timer: time after which address mapping will be forgotten (typically 20 min)

- try out “arp -a” command
ARP Protocol

- A wants to send datagram to B, and A knows B's IP address.
- A looks up B's MAC address in its ARP table.
- Suppose B's MAC address is not in A's ARP table.
- A broadcasts (why?) ARP query packet, containing B's IP address
  - all machines on LAN receive ARP query

  B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A's MAC address (unicast)

- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
  - soft state: information that times out (goes away) unless refreshed

- ARP is “plug-and-play”:
  - nodes create their ARP tables without intervention from net administrator
**ARP Messages**

<table>
<thead>
<tr>
<th>Bit Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-7</td>
<td>Hardware Address Type (e.g., Ethernet)</td>
</tr>
<tr>
<td>8-15</td>
<td>Protocol Address Type (e.g., IP)</td>
</tr>
<tr>
<td>16-23</td>
<td>Operation (ARP request or ARP response)</td>
</tr>
<tr>
<td>24-31</td>
<td>Details of the ARP message</td>
</tr>
</tbody>
</table>

- **Hardware Address Type:** e.g., Ethernet
- **Protocol Address Type:** e.g., IP
- **Operation:** ARP request or ARP response
ARP Request & Response Processing

• The requester broadcasts ARP request
• The target node unicasts (why?) ARP reply to requester
  - With its physical address
  - Adds the requester into its ARP table (why?)
• On receiving the response, requester
  - updates its table, sets timer
• Other nodes upon receiving the ARP request
  - Refresh the requester entry if already there
  - No action otherwise (why?)
• Some questions to think about:
  - Shall requester buffer IP datagram while performing ARP?
  - What shall requester do if never receive any ARP response?
ARP Operation Illustration
Forwarding to Another LAN

walkthrough: send datagram from A to B via R

assume A knows B IP address

- Two ARP tables in router R, one for each IP network (LAN1 and 2)
- In routing table at source host, find router 111.111.111.110
- In ARP table at source, find MAC address E6-E9-00-17-BB-4B, etc
• A creates datagram with source A, destination B
• A uses ARP to get R's MAC address for 111.111.111.110
• A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram
• A’s data link layer sends frame
• R's data link layer receives frame
• R removes IP datagram from Ethernet frame, sees its destined to B
• R uses ARP to get B’s physical layer address
• R creates frame containing A-to-B IP datagram sends to B
Broadcast LAN: Media Access Control

- Broadcast LAN: single shared broadcast channel
  - two or more simultaneous transmissions by nodes: interference!
  - only one node can send successfully at a time!
- How to share a broadcast channel
  - Humans use multi-access protocols all the time

**multiple access protocol**
- distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
- communication about channel sharing must use channel itself!
- what to look for in multiple access protocols:
  - synchronous or asynchronous
  - information needed about other stations
  - robustness
  - performance: access delay and throughput
MAC Protocols: a Taxonomy

Three broad classes:

• **Channel Partitioning** (static controlled access)
  - divide channel into smaller “pieces” (e.g., time slots -> TDMA, frequency->FDMA, code->CDMA)
  - allocate piece to node for exclusive use

• **Random Access**
  - channel not divided, allow collisions
  - “recover” from collisions

• **“Taking turns”** (demand adaptive controlled access)
  - tightly coordinate shared access to avoid collisions
Taxonomy of MAC Protocols

Multiple Access Protocols

random access

- ALOHA
- CSMA/CD (IEEE 802.3)

controlled access

- demand adaptive
  - token passing
    - bus
    - IEEE 802.4
  - reservation
    - ring
    - IEEE 802.5
- static channel allocation
  - TDM
  - FDDI
  - FDM
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 $\rightarrow$ “collision”,

• random access MAC protocol specifies:
  - how to detect or avoid collisions
  - how to recover from collisions (e.g., via delayed retransmissions)

• Examples of random access MAC protocols:
  - ALOHA
  - slotted ALOHA
  - CSMA, CSMA/CD, CSMA/CA
Pure (unslotted) ALOHA

- unslotted Aloha: simple, no synchronization
- when frame first arrives
  - transmit immediately
- collision can happen!
  - frame sent at $t_0$ collides with other frames sent in $[t_0-1,t_0+1]$
Slotted ALOHA

Assumptions
- all frames same size
- time is divided into equal size slots, time to transmit 1 frame
- nodes start to transmit frames only at beginning of slots
- nodes are synchronized
- if 2 or more nodes transmit in slot, all nodes detect collision

Operation
- when node obtains fresh frame, it transmits in next slot
  - if no collision, node can send new frame in next slot
  - if collision, node retransmits frame in each subsequent slot with prob. 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

Success (S), Collision (C), Empty (E) slots
Efficiency is the long-run fraction of successful slots when there’s many nodes, each with many frames to send.

- Suppose N nodes with many frames to send, each transmits in slot with probability $p$.
- $\text{prob that 1st node has success in a slot} = p(1-p)^{N-1}$.
- $\text{prob that any node has a success} = Np(1-p)^{N-1}$.

- For max efficiency with N nodes, find $p^*$ that maximizes $Np(1-p)^{N-1}$.
- For many nodes, take limit of $Np^*(1-p^*)^{N-1}$ as $N$ goes to infinity, gives $1/e = 0.37$.

At best: channel used for useful transmissions 37% of time!
Pure Aloha Efficiency

\[ P(\text{success by given node}) = P(\text{node transmits}) \cdot \]
\[ P(\text{no other node transmits in } [p_0-1,p_0]) \cdot \]
\[ P(\text{no other node transmits in } [p_0-1,p_0]) = p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1} \]
\[ = p \cdot (1-p)^{2(N-1)} \]

... choosing optimum \( p \) and then letting \( n \to \infty \) ...

\[ = \frac{1}{2e} = .18 \]

Efficiency is even worse!
Performance of Aloha Protocols

Can we do better with random access?

\[ S = \text{throughput} = \text{"goodput"} \]

\[ G = \text{offered load} = Np \]
Carrier Sense Multiple Access

- Aloha is inefficient (and rude):
  - doesn’t listen before talking

- CSMA: Listen before transmit
  - Human analogy: don’t interrupt others!
  - If channel idle, transmit entire packet
  - If busy, defer transmission
    - How long should we wait?

- Persistent vs. Nonpersistent CSMA
  - Nonpersistent:
    - if idle, transmit
    - if busy, wait random amount of time
  - p-persistent
    - If idle, transmit with probability p
    - If busy, wait till it becomes idle
    - If collision, wait random amount of time

- Can carrier sense avoid collisions completely?
CSMA Collisions

Collisions can still occur:
Propagation delay means two nodes may not hear each other’s transmission.

Collision:
Entire packet transmission time wasted.

Note:
Role of distance & propagation delay in determining collision probability.
CSMA/CD (Collision Detection)

**CSMA/CD**: carrier sensing, deferral as in CSMA
- collisions detected within short time
- colliding transmissions aborted, reducing channel wastage

- human analogy: the polite conversationalist
  - Listen while talking, stop if collision detected
  - CSMA: Listen before talking

- How to detect collision?
  - easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - difficult in wireless LANs: receiver shut off while transmitting
CSMA/CD: Illustration

collision
detect/abort
time

t₀
t₁

space
Ethernet

“Dominant” LAN technology:
- cheap $20 for 100Mbs!
- first widely used LAN technology
- Simpler, cheaper than token LANs and ATM
- Kept up with speed race: 10, 100, 1000 Mbps

Metcalfe’s Ethernet sketch
Ethernet Frame Format

Sending adapter encapsulates IP datagram (or other network layer protocol packet) in **Ethernet frame**

**DIX frame format**

<table>
<thead>
<tr>
<th>8 bytes</th>
<th>6</th>
<th>6</th>
<th>2</th>
<th>0-1500</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>Dest addr</td>
<td>Src addr</td>
<td>Type</td>
<td>Data</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**IEEE 802.3 format**

<table>
<thead>
<tr>
<th>8 bytes</th>
<th>6</th>
<th>6</th>
<th>2</th>
<th>0-1500</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>Dest addr</td>
<td>Src addr</td>
<td>Length</td>
<td>Data</td>
<td>CRC</td>
</tr>
</tbody>
</table>

- Ethernet has a maximum frame size: data portion <=1500 bytes
- It has imposed a minimum frame size: 64 bytes (excluding preamble)
  - If data portion <46 bytes, pad with “junk” to make it 46 bytes

Q: Why minimum frame size in Ethernet?
Fields in Ethernet Frame Format

- **Preamble**: 8 bytes
  - 7 bytes with pattern 10101010 followed by one byte with pattern 10101011 (SoF: start-of-frame)
  - used to synchronize receiver, sender clock rates, and identify beginning of a frame

- **Addresses**: 6 bytes (MAC address)
  - if adapter receives frame with matching destination address, or with broadcast address (eg ARP packet), it passes data in frame to net-layer protocol
  - otherwise, adapter discards frame

- **Type**: indicates the higher layer protocol, mostly IP but others may be supported such as Novell IPX and AppleTalk)
  - 802.3: Length gives data size; “protocol type” included in data

- **CRC**: checked at receiver, if error is detected, the frame is simply dropped
Ethernet and IEEE 802.3

1-persistent CSMA/CD

• **Carrier sense:** station listens to channel first
  - *Listen before talking*

• **If idle, station may initiate transmission**
  - *Talk if quiet*

• **Collision detection:** continuously monitor channel
  - *Listen while talking*

• **If collision, stop transmission**
  - *One talker at a time*
Ethernet CSMA/CD Algorithm

1. Adaptor (NIC) gets datagram from network layer and creates frame
2. If adapter senses channel idle, it starts to transmit frame. If it senses channel busy, waits until channel idle and then transmits
3. If adapter transmits entire frame without detecting another transmission, the adapter is done with frame! Signal to network layer “transmit OK”
4. If adapter detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, adapter enters exponential backoff: after the $n^{th}$ collision in a row, adapter chooses a $K$ at random from $\{0,1,2,\ldots,2^m-1\}$. $m=\text{MIN}(n, 10)$ Adapter waits $K \times 512$ bit times and returns to Step 2
6. Quit after 16 attempts, signal to network layer “transmit error”
Ethernet’s CSMA/CD (more)

Jam Signal: make sure all other transmitters are aware of collision; 48 bits;

Bit time (time to transmit a single bit): 0.1 microsec for 10 Mbps Ethernet; for K=1023, wait time is about 50 msec

Exponential Backoff:
- **Goal:** adapt retransmission attempts to estimated current load
  - heavy load: random wait will be longer
- first collision: choose K from \{0,1\}; delay is \(K \times 512\) bit transmission times
- after second collision: choose K from \{0,1,2,3\}...
- after ten collisions, choose K from \{0,1,2,3,4,\ldots,1023\}

See/interact with Java applet on AWL Web site: highly recommended!
IEEE 802.3 Parameters

• 1 bit time = time to transmit one bit
  - 10 Mbps $\Rightarrow$ 1 bit time = 0.1 $\mu$s

• Maximum cable length $\leq$ 2.5km
  - Maximum 4 repeaters

• “Collision Domain”
  - Distance within which collision can be detected
  - IEEE 802.3 specifies:
    worst case collision detection time: 51.2 $\mu$s

• Why minimum frame size?
  - 51.2 $\mu$s $\Rightarrow$ minimum # of bits can be transited at 10Mbps
    is 512 bits $\Rightarrow$ 64 bytes is required for collision detection
Worst Case Collision Detection Time

(a) Packet starts at time 0

(b) Packet almost at B at time \( t \)

(c) Collision at time \( t \)

(d) Noise burst gets back to A at time \( 2t \)
CSMA/CD Efficiency

Relevant parameters
- cable length, signal speed, frame size, bandwidth

- $T_{prop} = \text{max prop between 2 nodes in LAN}$
- $t_{trans} = \text{time to transmit max-size frame}$

$$\text{efficiency} = \frac{1}{1 + 5\frac{t_{prop}}{t_{trans}}}$$

- Efficiency goes to 1
  - as $t_{prop}$ goes to 0
  - as $t_{trans}$ goes to infinity (a huge frame)

- Much better than ALOHA, but still decentralized, simple, and cheap
Ethernet Technologies: 10Base2

- **10**: 10Mbps; **2**: under 200 meters max cable length
- thin coaxial cable in a bus topology

- repeaters used to connect up to multiple segments
- repeater repeats bits it hears on one interface to its other interfaces: physical layer device only!
- has become a legacy technology
10BaseT and 100BaseT

- 10/100 Mbps rate; latter called “fast ethernet”
- $T$ stands for Twisted Pair
- Nodes connect to a hub: “star topology”; 100 m max distance between nodes and hub

- Hubs are essentially physical-layer repeaters:
  - bits coming in one link go out all other links
  - no frame buffering
  - no CSMA/CD at hub: adapters detect collisions
  - provides net management functionality
100Base T (Fast) Ethernet: Issues

• 1 bit time = time to transmit one bit
  - 100 Mbps → 1 bit time = 0.01 μs

• If we keep the same “collision domain”, i.e., worst case collision detection time kept at 51.2 μs
  Q: What will be the minimum frame size?
  - 51.2 μs ⇒ minimum # of bits can be transited at 100Mpbs is 5120 bits ⇒ 640 bytes is required for collision detection
  - This requires change of frame format and protocol!

• Or we can keep the same minimum frame size, but reduce “collision domain” or network diameter!
  • from 51.2 μs to 5.12 μs!
  • maximum network diameter ≤ 100 m!
Gigabit Ethernet

- use standard Ethernet frame format
- allows for point-to-point links and shared broadcast channels
- in shared mode, CSMA/CD is used; short distances between nodes to be efficient
- uses hubs, called here “Buffered Distributors”
- Full-Duplex at 1 Gbps for point-to-point links
- 10 Gbps now!
Ethernet Summary

• 1-persistent CSMA/CD
• 10Base Ethernet
  - 51.2 μs to seize the channel
  - Collision not possible after 51.2 μs
  - Minimum frame size of 64 bytes
  - Binary exponential backoff
  - Works better under light load
  - Delivery time non-deterministic

• Evolution of Ethernet: Fast (100BaseT) and Gigabit Ethernet
“Taking Turns” MAC protocols

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!

Random access MAC protocols
- efficient at low load: single node can fully utilize channel
- high load: collision overhead

“taking turns” protocols
- try to look for best of both worlds (hopefully)!

Human analogy:
- traffic control with green/red light
  - fixed time vs. adaptive time vs. no lights at all
“Taking Turns” MAC Protocols

Polling:
- centralized
- master node “invites” slave nodes to transmit in turn
- concerns:
  - polling overhead
  - latency
  - single point of failure (master)

Token passing:
- distributed
- control token passed from one node to next sequentially.
- token message
- concerns:
  - token overhead
  - latency
  - single point of failure (token)
Ring Topology

(a) Unidirectional ring

(b) 1 bit delay

(c) Ring interface

To station  From station

To station  From station
Token Release

Release after Transmission

Release after Reception
Token Ring (IEEE 802.5)

• **Station**
  - Wait for token to arrive
  - Hold the token and start data transmission
    • Maximum token holding time $\Rightarrow$ max packet size
  - Strip the data frame off the ring
    • After it has gone around the ring
  - When done, release the token to next station

• **When no station has data to send**
  - Token circulates continuously
  - Ring must have sufficient delay to contain the token
Token Ring Performance

• Efficiency

\[ \approx \frac{1}{1 + a} \]

where

\[ a = \frac{PROP}{TRANS} \]
Tokens and Data Frames

![Diagram of tokens and data frames]
Token Ring Frame Fields

• **Access Control**
  - Token bit: 0 $\Rightarrow$ token  1 $\Rightarrow$ data
  - Monitor bit: used for monitoring ring
  - Priority and reservation bits: multiple priorities

• **Frame Status**
  - Set by destination, read by sender

• **Frame control**
  - Various control frames for ring maintenance
Priority and Reservation

- Token carries priority bits
  - Only stations with frames of equal or higher priority can grab the token

- A station can make reservation
  - When a data frame goes by
  - If a higher priority has not been reserved

- A station raising the priority is responsible for lowering it again
Ring Maintenance

• Each ring has a **monitor** station

• How to select a monitor?
  - Election/self-promotion: CLAIM_TOKEN

• Responsibilities
  - Insert additional delay
    • To accommodate the token
  - Check for lost token
    • Regenerate token
  - Watch for orphan frames
    • Drain them off the ring
  - Watch for garbled frames
    • Clean up the ring and regenerate token
Fault Scenarios

• What to do if ring breaks?
  - Everyone participates in detecting ring breaks
  - Send beacon frames
  - Figure out which stations are down
  - By-pass them if possible

• What happens if monitor dies?
  - Everyone gets a chance to become the new king

• What if monitor goes berserk?
HERE'S YOUR PROBLEM. THE CONNECTION TO THE NETWORK IS BROKEN.

Uh-oh. It's a "Token Ring" LAN. That means the token fell out and it's in this room someplace.

You are the wind beneath my wings.

I'll wait a week then tell him the token must be in the "Ethernet."

Copyright © 1996 United Feature Syndicate, Inc. Redistribution in whole or in part prohibited
Token Ring Summary

• Stations take **turns** to transmit
• Only the station with the token can transmit
• Sender receives its own transmission
  - Drains its frame off the ring
• Releases token after transmission/reception
• Deterministic delivery possible
• High throughput under heavy load
### Ethernet vs Token Ring

- **Non-deterministic**
- No delays at low loads
- Low throughput under heavy load
- No priorities
- No management overhead
- Large minimum size

- **Deterministic**
- Substantial delays at low loads
- High throughput under heavy load
- Multiple priorities
- Complex management
- Small frames possible
Summary of MAC Protocols

• Why media access control?
  - Shared media: only one user can send at a time
  - Media access control: determine who has access

• MAC issues:
  - distributed, using the same channel for regulating access

• What do you do with a shared media?
  - Channel Partitioning, by time, frequency or code
    • Time Division, Code Division, Frequency Division
  - Random Access (dynamic)
    • ALOHA, S-ALOHA, CSMA, CSMA/CD
    • carrier sensing easy in some technologies (wire), hard in others (wireless)
    • CSMA/CD used in Ethernet
  - Taking Turns
    • polling from a central site, token passing (Token Ring, FDDI)
Interconnecting LAN Segments

- Hubs
- Bridges
- (Layer-2) Switches
  - Remark: switches are essentially multi-port bridges.
  - What we say about bridges also holds for switches!
Hubs

Hubs are essentially physical-layer repeaters:
- bits coming from one link go out all other links
- at the same rate
- no frame buffering
- no CSMA/CD at hub: adapters detect collisions
- provides net management functionality
Interconnecting with Hubs

- Backbone hub interconnects LAN segments
- Extends max distance between nodes
- But individual segment collision domains become one large collision domain
- Can’t interconnect 10BaseT & 100BaseT
Bridge and Layer-2 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
Bridge/Switch: Traffic Isolation

- Bridge/switch installation breaks subnet into LAN segments
- Bridge/switch filters packets:
  - same-LAN-segment frames not usually forwarded onto other LAN segments
  - segments become separate collision domains

![Diagram showing traffic isolation with switches and collision domains]
• How do determine onto which LAN segment to forward frame?
• Looks like a routing problem...
Self Learning

- A bridge/switch has a **forwarding (or switch) table**
- entry in forwarding table:
  - (MAC Address, Interface, Time Stamp)
  - stale entries in table dropped (TTL can be 60 min)
- Bridge/switch *learns* which hosts can be reached through which interfaces
  - when frame received, switch "learns" location of sender: incoming LAN segment
  - records sender/location pair in forwarding table
Filtering/Forwarding

When bridge/switch receives a frame:

index switch table using MAC dest address
if entry found for destination
  then{
    if dest on segment from which frame arrived
      then drop the frame
    else forward the frame on interface indicated
  }
else flood
forward on all but the interface on which the frame arrived
Bridge/Switch Example

Suppose C sends frame to D

- Switch receives frame from C
  - notes in bridge table that C is on interface 1
  - because D is not in table, switch forwards frame into interfaces 2 and 3

- Frame received by D
Bridge/Switch Example

Suppose D replies back with frame to C.

- Switch receives frame from from D
  - notes in bridge table that D is on interface 2
  - because C is in table, switch forwards frame only to interface 1

- frame received by C
Switches: Dedicated Access

- Switch with many interfaces
- Hosts have direct connection to switch
- No collisions; full duplex

**Switching:** A-to-A’ and B-to-B’ simultaneously, no collisions
More on Switches

• cut-through switching: frame forwarded from input to output port without first collecting entire frame
  - slight reduction in latency

• combinations of shared/dedicated, 10/100/1000 Mbps interfaces
Interconnection without Backbone

- Not recommended for two reasons:
  - single point of failure at Computer Science hub
  - all traffic between EE and SE must path over CS segment
Backbone Configuration

Recommended!
Bridges Spanning Tree

- for increased reliability, desirable to have redundant, alternative paths from source to dest
- with multiple paths, cycles result - bridges may multiply and forward frame forever
- solution: organize bridges in a spanning tree by disabling subset of interfaces
Bridge Spanning Tree Algorithm: Algorhyme

I think that I shall never see
A graph more lovely than a tree.
A tree whose crucial property
Is loop-free connectivity.
A tree that must be sure to span
So packets can reach every LAN.
First, the root must be selected.
By ID, it is elected.
Least cost paths from root are traced.
In the tree, these paths are placed.
A mesh is made by folks like me,
Then bridges find a spanning tree

-- Radia Perlman
Some Bridge/Switch Features

• Isolates collision domains resulting in higher total max throughput
• limitless number of nodes and geographical coverage
• Can connect different Ethernet types
• Transparent (“plug-and-play”): no configuration necessary
Institutional Network

to external network

router

mail server

web server

IP subnet

hub

hub

hub
Not an atypical LAN (IP network)
Bridges/Switches vs. Routers

- both store-and-forward devices
  - routers: network layer devices (examine network layer headers)
  - bridges are link layer devices
- routers maintain routing tables, implement routing algorithms
- bridges maintain bridge forwarding tables, implement filtering, learning and spanning tree algorithms
Routers vs. Bridges/Switches

Bridges + and -

+ Bridge operation is simpler requiring less packet processing
+ Bridge tables are self learning
- All traffic confined to spanning tree, even when alternative bandwidth is available
- Bridges do not offer protection from broadcast storms
Routers vs. Bridges/Switches ...

Routers + and -

+ arbitrary topologies can be supported, cycling is limited by TTL counters (and good routing protocols)
+ provide protection against broadcast storms
- require IP address configuration (not plug and play)
- require higher packet processing

• bridges do well in small (few hundred hosts) while routers used in large networks (thousands of hosts)
## Summary Comparison

<table>
<thead>
<tr>
<th></th>
<th>hubs</th>
<th>bridges</th>
<th>routers</th>
<th>(Layer-2) switches</th>
</tr>
</thead>
<tbody>
<tr>
<td>traffic isolation</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>plug &amp; play</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>optimal routing</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>cut through</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>
Point to Point Data Link Control

• one sender, one receiver, one link: easier than broadcast link:
  - no Media Access Control
  - no need for explicit MAC addressing
  - e.g., dialup link, ISDN line

• popular point-to-point DLC protocols:
  - PPP (point-to-point protocol)
  - HDLC: High level data link control
    • data link layer used to be considered “high layer” in protocol stack!
PPP Design Requirements [RFC 1557]

- **packet framing**: encapsulation of network-layer datagram in data link frame
  - carry network layer data of any network layer protocol (not just IP) at same time
  - ability to demultiplex upwards
- **bit transparency**: must carry any bit pattern in the data field
- **error detection** (no correction)
- **connection liveness**: detect, signal link failure to network layer
- **network layer address negotiation**: endpoint can learn/configure each other’s network address
PPP Non-Requirements

• no error correction/recovery
• no flow control
• out of order delivery OK
• no need to support multipoint links (e.g., polling)

Error recovery, flow control, data re-ordering all relegated to higher layers!
PPP Data Frame

- **Flag**: delimiter (framing)
- **Address**: does nothing (only one option)
- **Control**: does nothing; in the future possible multiple control fields
- **Protocol**: upper layer protocol to which frame delivered (e.g., PPP-LCP, IP, IPCP, etc)

```
 01111110 11111111 00000011 protocol info check 01111110
   flag    control address
```
PPP Data Frame

- **info**: upper layer data being carried
- **check**: cyclic redundancy check for error detection

![PPP Data Frame Diagram]
Byte Stuffing

• “data transparency” requirement: data field must be allowed to include flag pattern \(<01111110>\)
  - \textcolor{red}{Q}: is received \(<01111110>\) data or flag?

• \textcolor{blue}{Sender}: adds (“stuffs”) extra \(<01111110>\) byte after each \(<01111110>\) \textit{data} byte

• \textcolor{blue}{Receiver}:
  - two \(01111110\) bytes in a row: discard first byte, continue data reception
  - single \(01111110\): flag byte
**Byte Stuffing**

Flag byte pattern in data to send:

- Flag byte pattern: 01111110
- Stuffed byte: 01111110

Flag byte pattern plus stuffed byte in transmitted data:

- Original data: b5 b4 01111110
- Transmitted data: 011111101 b2 b1
PPP Link/Network Control Protocols

Before exchanging network-layer data, data link peers must:

- **configure PPP link** (max. frame length, authentication)
- **learn/configure network layer information**
  - for IP: carry IP Control Protocol (IPCP) msgs (protocol field: 8021) to configure/learn IP address
Virtualization of Networks
(optional material)

Virtualization of resources: a powerful abstraction in systems engineering:

• computing examples: virtual memory, virtual devices
  - Virtual machines: e.g., java
  - IBM VM os from 1960’s/70’s

• layering of abstractions: don’t sweat the details of the lower layer, only deal with lower layers abstractly
The Internet: Virtualizing Networks

1974: multiple unconnected nets
- ARPAnet
- data-over-cable networks
- packet satellite network (Aloha)
- packet radio network

... differing in:
- addressing conventions
- packet formats
- error recovery
- routing

The Internet: Virtualizing Networks

Internetwork layer (IP):
- addressing: internetwork appears as a single, uniform entity, despite underlying local network heterogeneity
- network of networks

Gateway:
- “embed internetwork packets in local packet format or extract them”
- route (at internetwork level) to next gateway

ARPAnet

satellite net
Cerf & Kahn’s Internetwork Architecture

What is virtualized?

• two layers of addressing: internetwork and local network
• new layer (IP) makes everything homogeneous at internetwork layer
• underlying local network technology
  - cable
  - satellite
  - 56K telephone modem
  - today: ATM, MPLS

... “invisible” at internetwork layer. Looks like a link layer technology to IP!
ATM and MPLS

- ATM, MPLS separate networks in their own right
  - different service models, addressing, routing from Internet
- viewed by Internet as logical link connecting IP routers
  - just like dialup link is really part of separate network (telephone network)
- ATM, MPSL: of technical interest in their own right
Asynchronous Transfer Mode: ATM

• 1990’s/00 standard for high-speed (155Mbps to 622 Mbps and higher) Broadband Integrated Service Digital Network architecture

• **Goal**: integrated, end-end transport of carry voice, video, data
  - meeting timing/QoS requirements of voice, video (versus Internet best-effort model)
  - “next generation” telephony: technical roots in telephone world
  - packet-switching (fixed length packets, called “cells”) using virtual circuits
ATM Architecture

- **adaptation layer**: only at edge of ATM network
  - data segmentation/reassembly
  - roughly analogous to Internet transport layer
- **ATM layer**: “network” layer
  - cell switching, routing
- **physical layer**
**Vision:** end-to-end
transport: “ATM from desktop to desktop”
- ATM is a network technology

**Reality:** used to connect
IP backbone routers
- “IP over ATM”
- ATM as switched link layer, connecting IP routers
ATM Adaptation Layer (AAL)

- **ATM Adaptation Layer (AAL):** “adapts” upper layers (IP or native ATM applications) to ATM layer below
- **AAL present only in end systems, not in switches**
- **AAL layer segment (header/trailer fields, data) fragmented across multiple ATM cells**
  - analogy: TCP segment in many IP packets
ATM Adaptation Layer (AAL) [more]

Different versions of AAL layers, depending on ATM service class:

- **AAL1**: for CBR (Constant Bit Rate) services, e.g., circuit emulation
- **AAL2**: for VBR (Variable Bit Rate) services, e.g., MPEG video
- **AAL5**: for data (e.g., IP datagrams)
### ATM Layer

**Service:** transport cells across ATM network
- analogous to IP network layer
- very different services than IP network layer

<table>
<thead>
<tr>
<th>Network Architecture</th>
<th>Service Model</th>
<th>Bandwidth</th>
<th>Loss</th>
<th>Order</th>
<th>Timing</th>
<th>Guaranettes ?</th>
<th>Congestion feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet</td>
<td>best effort</td>
<td>none</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no (inferred via loss)</td>
</tr>
<tr>
<td>ATM</td>
<td>CBR</td>
<td>constant rate</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>ATM</td>
<td>VBR</td>
<td>guaranteed rate</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>ATM</td>
<td>ABR</td>
<td>guaranteed minimum</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>ATM</td>
<td>UBR</td>
<td>none</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>
ATM Layer: Virtual Circuits

• **VC transport:** cells carried on VC from source to dest
  - call setup, teardown for each call before data can flow
  - each packet carries VC identifier (not destination ID)
  - every switch on source-dest path maintain “state” for each passing connection
  - link, switch resources (bandwidth, buffers) may be allocated to VC: to get circuit-like perf.

• **Permanent VCs (PVCs)**
  - long lasting connections
  - typically: “permanent” route between to IP routers

• **Switched VCs (SVC):**
  - dynamically set up on per-call basis
ATM VCs

- Advantages of ATM VC approach:
  - QoS performance guarantee for connection mapped to VC (bandwidth, delay, delay jitter)

- Drawbacks of ATM VC approach:
  - Inefficient support of datagram traffic
  - one PVC between each source/dest pair) does not scale (N*2 connections needed)
  - SVC introduces call setup latency, processing overhead for short lived connections
ATM Layer: ATM Cell

- 5-byte ATM cell header
- 48-byte payload
  - Why?: small payload -> short cell-creation delay for digitized voice
  - halfway between 32 and 64 (compromise!)
ATM Cell Header

- **VCI**: virtual channel ID
  - will *change* from link to link thru net
- **PT**: Payload type (e.g. RM cell versus data cell)
- **CLP**: Cell Loss Priority bit
  - CLP = 1 implies low priority cell, can be discarded if congestion
- **HEC**: Header Error Checksum
  - cyclic redundancy check
ATM Physical Layer (cont’d)

Two pieces (sublayers) of physical layer:
• Transmission Convergence Sublayer (TCS): adapts ATM layer above to PMD sublayer below
• Physical Medium Dependent: depends on physical medium being used

TCS Functions:
  - Header checksum generation: 8 bits CRC
  - Cell delineation
  - With “unstructured” PMD sublayer, transmission of idle cells when no data cells to send
ATM Physical Layer

Physical Medium Dependent (PMD) sublayer

- **SONET/SDH**: transmission frame structure (like a container carrying bits);
  - bit synchronization;
  - bandwidth partitions (TDM);
  - several speeds: OC3 = 155.52 Mbps; OC12 = 622.08 Mbps; OC48 = 2.45 Gbps, OC192 = 9.6 Gbps

- **TI/T3**: transmission frame structure (old telephone hierarchy): 1.5 Mbps/45 Mbps

- **unstructured**: just cells (busy/idle)
IP-Over-ATM

Classic IP only

- 3 “networks” (e.g., LAN segments)
- MAC (802.3) and IP addresses

IP over ATM

- replace “network” (e.g., LAN segment) with ATM network
- ATM addresses, IP addresses
Datagram Journey in IP-over-ATM Network

- **at Source Host:**
  - IP layer maps between IP, ATM dest address (using ARP)
  - passes datagram to AAL5
  - AAL5 encapsulates data, segments cells, passes to ATM layer
- **ATM network:** moves cell along VC to destination
- **at Destination Host:**
  - AAL5 reassembles cells into original datagram
  - if CRC OK, datagram is passed to IP
IP-Over-ATM

Issues:
- IP datagrams into ATM AAL5 PDUs
- from IP addresses to ATM addresses
  - just like IP addresses to 802.3 MAC addresses!
Multiprotocol Label Switching (MPLS)

- initial goal: speed up IP forwarding by using fixed length label (instead of IP address) to do forwarding
  - borrowing ideas from Virtual Circuit (VC) approach
  - but IP datagram still keeps IP address!

<table>
<thead>
<tr>
<th>PPP or Ethernet header</th>
<th>MPLS header</th>
<th>IP header</th>
<th>remainder of link-layer frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>Exp</td>
<td>S</td>
<td>TTL</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>1</td>
<td>8</td>
</tr>
</tbody>
</table>
MPLS Capable Routers

- a.k.a. label-switched router
- forwards packets to outgoing interface based only on label value (don’t inspect IP address)
  - MPLS forwarding table distinct from IP forwarding tables
- signaling protocol needed to set up forwarding
  - RSVP-TE or LDP
  - forwarding possible along paths that IP alone would not allow (e.g., source-specific routing)!!
  - use MPLS for traffic engineering
- must co-exist with IP-only routers
MPLS Forwarding Tables

<table>
<thead>
<tr>
<th>in label</th>
<th>out label</th>
<th>dest</th>
<th>out interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>A</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>D</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>A</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>in label</th>
<th>out label</th>
<th>dest</th>
<th>out interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>D</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>in label</th>
<th>out label</th>
<th>dest</th>
<th>out interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>6</td>
<td>A</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>in label</th>
<th>out label</th>
<th>dest</th>
<th>out interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>-</td>
<td>A</td>
<td>0</td>
</tr>
</tbody>
</table>
Data Link Layer Summary

• Data Link Layer Functions
  - deliver frames over a single link
  - framing, media access, error checking (error correction), …

• Principles behind data link layer services:
  - sharing a broadcast channel: multiple access
  - link layer addressing, ARP

• Local Area Networks (LANs) and MAC Addresses
  - point-to-point vs. shared access
  - MAC addresses
  - MAC addresses vs. IP addresses
  - IP Address Resolution Protocol (ARP) and IP datagram forwarding revisited
Data Link Layer Summary (cont’d)

- Media Access Control and Link Layer Technologies
  - Why media access control, issues
  - Taxonomy of MAC protocols
  - Random access protocols:
    - Aloha, slotted Aloha,
    - CSMA, CSMA/CD and Ethernet
    - CSMA/CA and 802.11 Wireless LAN (Chapter 6)
  - “Take Turns” protocols
    - polling, token passing and Token Ring
- Extending and segmenting LANs
  - hubs, bridges, switches
- Point-to-Point Link and Link/Network Virtualization
  - PPP, ATM, MPLS
- journey down the protocol stack now (nearly) OVER!
  - Next: wireless networks and Mobility