Chapter 5: The Data Link Layer

Our goals:
- understand principles behind data link layer services:
  - error detection, correction
  - sharing a broadcast channel: multiple access
  - link layer addressing
  - reliable data transfer, flow control: done!
- instantiation and implementation of various link layer technologies

Overview:
- link layer services
- error detection, correction
- multiple access protocols and LANs
- link layer addressing, ARP
- specific link layer technologies:
  - Ethernet
  - hubs, bridges, switches
  - IEEE 802.11 LANs and wireless
  - PPP
  - ATM

Link Layer: Introduction

data-link layer has responsibility of transferring datagram from one node to adjacent node over a link

Datagram transferred by different link protocols over different links:
- e.g., Ethernet on first link, frame relay on intermediate links, 802.11 on last link

Each link protocol provides different services
- e.g., may or may not provide rdt over link
Link Layer: setting the context

- two physically connected devices:
  - host-router, router-router, host-host
- unit of data: frame
**Link Layer Services**

- **Framing, link access:**
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - 'physical addresses' used in frame headers to identify source, dest
    - different from IP address!

- **Reliable delivery between adjacent nodes**
  - we learned how to do this already (chapter 3)!
  - 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?

---

**Link Layer Services (more)**

- **Flow Control:**
  - pacing between adjacent sending and receiving nodes

- **Error Detection:**
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame

- **Error Correction:**
  - receiver identifies and corrects bit error(s) without resorting to retransmission

- **Half-duplex and full-duplex**
  - with half duplex, nodes at both ends of link can transmit, but not at same time
Adaptors Communicating

- Link layer implemented in "adaptor" (aka NIC)
  - Ethernet card, PCMCI card, 802.11 card
- Sending side:
  - Encapsulates datagram in a frame
  - Adds error checking bits, rdt, flow control, etc.

- Receiving side:
  - Looks for errors, rdt, flow control, etc
  - Extracts datagram, passes to receiving node
- Adapter is semi-autonomous
- Link & physical layers

Error Detection

EDC = Error Detection and Correction bits (redundancy)

- 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

- d data bits
- parity bit

0111000110101011 0

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

\[
\begin{array}{cccc}
\text{d}_{1,1} & \ldots & \text{d}_{1,j} & \text{d}_{1,j+1} \\
\text{d}_{2,1} & \ldots & \text{d}_{2,j} & \text{d}_{2,j+1} \\
\vdots & \ldots & \vdots & \vdots \\
\text{d}_{i,1} & \ldots & \text{d}_{i,j} & \text{d}_{i,j+1} \\
\text{d}_{i+1,1} & \ldots & \text{d}_{i+1,j} & \text{d}_{i+1,j+1} \\
\end{array}
\]

*row parity*

*column parity*

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 ....

5: DataLink Layer 5a-10
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)

\[ D \cdot 2^r \text{ XOR } R \]

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 Protocols

Two types of “links“:
- **point-to-point**
  - PPP for dial-up access
  - point-to-point link between Ethernet switch and host
- **broadcast** (shared wire or medium)
  - traditional Ethernet
  - upstream HFC
  - 802.11 wireless LAN

Multiple Access protocols

- single shared broadcast channel
- two or more simultaneous transmissions by nodes: interference
  - only one node can send *successfully* at a 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!
- claim: humans use multiple access protocols all the time
Ideal Multiple Access Protocol

Broadcast channel of rate R bps
1. When one node wants to transmit, it can send at rate R.
2. When M nodes want to transmit, each can send at average rate R/M
3. Fully decentralized:
   - no special node to coordinate transmissions
   - no synchronization of clocks, slots
4. Simple

MAC Protocols: a taxonomy

Three broad classes:
- Channel Partitioning
  - divide channel into smaller "pieces" (time slots, frequency, code)
  - allocate piece to node for exclusive use
- Random Access
  - channel not divided, allow collisions
  - "recover" from collisions
- "Taking turns"
  - tightly coordinate shared access to avoid collisions
Channel Partitioning MAC protocols: TDMA

**TDMA: time division multiple access**
- access to channel in "rounds"
- each station gets fixed length slot (length = pkt trans time) in each round
- unused slots go idle
- example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle

![Diagram showing TDMA access]

Channel Partitioning MAC protocols: 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

![Diagram showing FDMA access]
Channel Partitioning (CDMA)

CDMA (Code Division Multiple Access)
- unique “code” assigned to each user; i.e., code set partitioning
- used mostly in wireless broadcast channels (cellular, satellite, etc)
- all users share same frequency, but each user has own “chipping” sequence (i.e., code) to encode data
- encoded signal = (original data) X (chipping sequence)
- decoding: inner-product of encoded signal and chipping sequence
- allows multiple users to “coexist” and transmit simultaneously with minimal interference (if codes are “orthogonal”)

CDMA Encode/Decode

sender

channel output Z_{i,m}

M

slot 1 channel output

M

slot 0 channel output

receiver

M

slot 1 received input

M

slot 0 received input

sender

data bits

code

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

sender

data bits

code

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

channel output Z_{i,m}

M

slot 1 channel output

M

slot 0 channel output

receiver

M

slot 1 received input

M

slot 0 received input

CDMA Encode/Decode

sender

data bits

code

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

g_{i+1}

channel output Z_{i,m}

M

slot 1 channel output

M

slot 0 channel output

receiver

M

slot 1 received input

M

slot 0 received input

CDMA Encode/Decode
**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 (e.g., via delayed retransmissions)
- Examples of random access MAC protocols:
  - slotted ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA
Slotted Aloha

- time is divided into equal size slots (= pkt trans. time)
- node with new arriving pkt: transmit at beginning of next slot
- if collision: retransmit pkt in future slots with probability p, until successful.

```
node 1 1 1 1 1
node 2 2 2 2 2
node 3 3 3 3 3
C E C E C E
```

Success (S), Collision (C), Empty (E) slots

5: DataLink Layer 5a-23

Slotted Aloha efficiency

Q: what is max fraction slots successful?
A: Suppose N stations have packets to send
  - each transmits in slot with probability p
  - prob. successful transmission S is:

  by single node: $S = p (1-p)^{(N-1)}$

  by any of N nodes
  $S = \text{Prob (only one transmits)}$
  $= N \cdot p (1-p)^{(N-1)}$
  ... choosing optimum p as $n \to \infty$ ...
  $= \frac{1}{e} = .37$ as $N \to \infty$

At best: channel use for useful transmissions 37% of time!

5: DataLink Layer 5a-24
Pure (unslotted) ALOHA

- unslotted Aloha: simpler, no synchronization
- pkt needs transmission:
  - send without awaiting for beginning of slot
- collision probability increases:
  - pkt sent at \( t_0 \) collide with other pkts sent in \([t_0-1, t_0+1]\)

Pure Aloha (cont.)

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

\[
P(\text{no other node transmits in } [p_0-1,p_0]) = p \cdot (1-p) \cdot (1-p)
\]

\[
P(\text{success by any of } N \text{ nodes}) = N \cdot p \cdot (1-p) \cdot (1-p)
\]

... choosing optimum \( p \) as \( n \to \infty \) ...

\[
= 1/(2e) = .18
\]

\[ S \] = throughput = "goodput"

\[ G = \text{offered load} = Np \]

protocol constrains effective channel throughput!
**CSMA: Carrier Sense Multiple Access**

**CSMA:** listen before transmit:
- If channel sensed idle: transmit entire pkt
- If channel sensed busy, defer transmission
  - Persistent CSMA: retry immediately with probability p when channel becomes idle (may cause instability)
  - Non-persistent CSMA: retry after random interval
- Human analogy: don’t interrupt others!

**CSMA collisions**

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

Collision:
- Entire packet transmission time wasted

Note:
- Role of distance and propagation delay in determining collision prob.
CSMA/CD (Collision Detection)

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

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
“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
look for best of both worlds!

---

“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)
Reservation-based protocols

Distributed Polling:
- time divided into slots
- begins with N short reservation slots
  - reservation slot time equal to channel end-end propagation delay
  - station with message to send posts reservation
  - reservation seen by all stations
- after reservation slots, message transmissions ordered by known priority

Summary of MAC protocols

- What do you do with a shared media?
  - Channel Partitioning, by time, frequency or code
    - Time Division, Code Division, Frequency Division
  - Random partitioning (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
LAN technologies

Data link layer so far:
- services, error detection/correction, multiple access

Next: LAN technologies
- addressing
- Ethernet
- hubs, bridges, switches
- 802.11
- PPP
- ATM

LAN Addresses and ARP

32-bit IP address:
- network-layer address
- used to get datagram to destination IP network (recall IP network definition)

LAN (or MAC or physical or Ethernet) address:
- used to get datagram from one interface to another physically-connected interface (same network)
- 48 bit MAC address (for most LANs)
  burned in the adapter ROM
LAN Addresses and ARP

Each adapter on LAN has unique LAN address

- MAC address allocation administered by IEEE
- Manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy:
  - MAC address: like Social Security Number
  - IP address: like postal address
- MAC flat address => portability
  - Can move LAN card from one LAN to another
- IP hierarchical address NOT portable
  - Depends on IP network to which node is attached
Recall earlier routing discussion

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

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
  - IP address, MAC address, TTL
- TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)
ARP protocol

- A wants to send datagram to B, and A knows B's IP address.
- Suppose B's MAC address is not in A's ARP table.
- A broadcasts 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.

Routing to another LAN

Walkthrough: send datagram from A to B via R.
Assume A knows B's IP address.

- Two ARP tables in router R, one for each IP network (LAN).
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

---

**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 Structure

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

Preamble:
- 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
- used to synchronize receiver, sender clock rates

Addresses: 6 bytes
- 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

CRC: 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 to sending adapter
  - Stream of datagrams passed to network layer can have gaps
  - Gaps will be filled if app is using TCP
  - Otherwise, app will see the gaps

Ethernet uses CSMA/CD

- No slots
- Adapter doesn’t transmit if it senses that some other adapter is transmitting, that is, carrier sense
- Transmitting adapter aborts when it senses that another adapter is transmitting, that is, collision detection
- Before attempting a retransmission, adapter waits a random time, that is, random access
Ethernet: uses CSMA/CD

A: sense channel, if idle
   then {
      transmit and monitor the channel;
      If detect another transmission
      then {
         abort and send jam signal;
         update # collisions;
         delay as required by exponential backoff algorithm;
         goto A
      }
      else {done with the frame; set collisions to zero}
   }
else {wait until ongoing transmission is over and goto A}

Ethernet's CSMA/CD (more)

Jam Signal: make sure all other transmitters are aware of collision; 48 bits;
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 x 512 bit transmission times
- after second collision: choose K from {0,1,2,3}...
- after ten or more collisions, choose K from {0,1,2,3,4,...,1023}
**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
Manchester encoding

- Used in 10BaseT, 10Base2
- Each bit has a transition
- Allows clocks in sending and receiving nodes to synchronize to each other
  - no need for a centralized, global clock among nodes!
- Hey, this is physical-layer stuff!

Gbit 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!