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.
- instantiation and implementation of various link layer technologies

Layer 2: Link Layer

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet
- 5.6 Hubs and switches
- 5.7 PPP
- 5.8 Link Virtualization: ATM and MPLS
**Link Layer: Introduction**

Some terminology:
- hosts and routers are **nodes**
- communication channels that connect adjacent nodes along communication path are **links**
  - wired links
  - wireless links
  - **LANs**
- layer-2 packet is a **frame**,
  - encapsulates datagram

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

**Link layer: context**

- 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

**Transportation analogy**
- trip from Princeton to Lausanne
  - limo: Princeton to JFK
  - plane: JFK to Geneva
  - train: Geneva to Lausanne
- tourist = datagram
- transport segment = communication link
- transportation mode = link layer protocol
- travel agent = routing algorithm
Link Layer Services

- **Framing, link access:**
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - "MAC" 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 rcving node
- Adapter is semi-autonomous
- Link & physical layers

Link Layer

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet
- 5.6 Hubs and switches
- 5.7 PPP
- 5.8 Link Virtualization: ATM
### 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

![Diagram of Error Detection and Correction](image)

### Parity Checking

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

```
<table>
<thead>
<tr>
<th>d data bits</th>
<th>parity bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0111001101011011</td>
<td>0</td>
</tr>
</tbody>
</table>
```

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

<table>
<thead>
<tr>
<th>row parity</th>
<th>parity bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>d₁,₁ ...</td>
<td>d₁,j</td>
</tr>
<tr>
<td>d₂,₁ ...</td>
<td>d₂,j</td>
</tr>
<tr>
<td>... ...</td>
<td>...</td>
</tr>
<tr>
<td>dᵢ,₁ ...</td>
<td>dᵢ,j</td>
</tr>
<tr>
<td>dᵢ+1,₁ ...</td>
<td>dᵢ+1,j</td>
</tr>
</tbody>
</table>

- Forward Error Correction (FEC): The ability of the receiver to both detect and correct errors
- 2-dimensional bit parity can:
  1) detect and correct 1 bit errors in data or parity
  2) can detect but not correct 2 bit errors

**Example:**

```
<table>
<thead>
<tr>
<th>no errors</th>
<th>parity error</th>
<th>correctable single bit error</th>
</tr>
</thead>
<tbody>
<tr>
<td>10101010</td>
<td>01010101</td>
<td>01010101</td>
</tr>
<tr>
<td>111100</td>
<td>10110101</td>
<td>01101011</td>
</tr>
<tr>
<td>011110</td>
<td>011101</td>
<td>01101010</td>
</tr>
<tr>
<td>01010100</td>
<td>01010100</td>
<td>01010100</td>
</tr>
</tbody>
</table>
```
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:**
- take the 1’s complement of the sum of contents and original checksum then check if computed checksum equals all 1’s:
  - NO - error detected
  - YES - no error detected. But *not all times.*

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 (can detect errors of r continuous bits)
  - A burst of greater than r+1 bits can be detected with a probability
- widely used in practice (ATM, HDCL)

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

**bit pattern**

\[ d \text{ bits} \quad r \text{ bits} \]

**mathematical formula**
**CRC Example**

Want:

\[ D.2^r \text{ XOR } R = nG \]

equivalently:

\[ D.2^r = nG \text{ XOR } R \]

equivalently:

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

\[ R = \text{remainder}\left(\frac{D.2^r}{G}\right) \]

E.g. \( D=101110 \), \( G=1001 \), \( r=3 \)

\[
\begin{array}{c}
101110 \\
1001 \\
101110000 \\
1001 \\
1010 \\
1001 \\
10110111 \\
011
\end{array}
\]

8-, 12-, 16-, and 32-bit generators.

An 8-bit CRC to protect the 5-byte header in ATM cells

CRC-32 is used in link-level IEEE protocols

Generator = 100000100 11000001 00011101 10110111

---

**Link Layer**

- 5.1 Introduction and services
- 5.2 Error detection and correction
- 5.3 Multiple access protocols
- 5.4 Link-Layer Addressing
- 5.5 Ethernet
- 5.6 Hubs and switches
- 5.7 PPP
- 5.8 Link Virtualization: ATM

---

5: DataLink Layer 5-13
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
  - collision if node receives two or more signals at the same 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!
  - no out-of-band channel for coordination
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"
  - Nodes take turns, but nodes with more to send can take longer turns
**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

---

**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
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

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
- 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
- Clock synchronization

Slotted ALOHA efficiency

Efficiency is the long-run fraction of successful slots when there are many nodes, each with many frames to send.

- Suppose N nodes with many frames to send, each transmits in slot with probability p
- Prob that node 1 has success in a slot = p(1-p)\(^{N-1}\)
- 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 = .37

At best: channel used for useful transmissions 37% of time!
Pure (unslotted) ALOHA [Abramson’70]

☐ unslotted Aloha: simpler, no synchronization
☐ when frame first arrives
  ☐ transmit immediately
☐ collision probability increases:
  ☐ frame sent at $t_0$ collides with other frames sent in $[t_0-1,t_0+1]$

Efficiency is even worse

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 $n \to \infty$ ...

Even worse! $$= 1/(2e) = .18$$
**CSMA (Carrier Sense Multiple Access)**

**CSMA**: listen before transmit:
- If channel sensed idle: transmit entire frame
- If channel sensed busy, defer transmission
- Human analogy: don't interrupt others!

**CSMA collisions**

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

Collision:
- Entire packet transmission time wasted

Note:
- Role of distance & propagation delay in determining collision probability
**CSMA/CD [Metcalfe 1976]**

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

- **collision detection**:
  - easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - difficult in wireless LANs: receiver shut off while transmitting

- human analogy: the polite conversationalist

---

**CSMA/CD collision detection**

![Diagram](image_url)
“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!

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)
Summary of MAC protocols

- What do you do with a shared media?
  - Channel Partitioning, by time, frequency or code
    - Time 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
    - CSMA/CA used in 802.11
  - 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, switches
  - PPP
Error Detection and MAP

- Link Layer Services:
  - Framing, reliable transfer, error detection/correction
- Error Detection and Correction Techniques:
  - Parity Checks
  - Checksum Methods
  - Cyclic Redundancy Check (CRC)
- Multiple Access Protocols
  - Channel Partitioning Protocols
  - Random Access Protocols
  - Taking Turns Protocols
  - Local Area Networks