Link Layer

- Introduction and services
- Error detection and correction
- Link-layer Addressing
- Multiple access protocols
- Ethernet operation
- A day in the life of a web request

IP/MAC Addresses and ARP

- 32-bit IP address:
  - *network-layer address*
  - used to get datagram to destination IP subnet

- Physical address
  - Function: *get frame from one interface to another physically-connected interface (same network)*
  - On a local network, typically a MAC address
  - E.g., 48 bit MAC address (for most LANs)
    - burned in NIC ROM, also sometimes software settable
LAN Addresses and ARP

Each adapter on LAN has unique LAN address

Broadcast address = FF-FF-FF-FF-FF

LAN (wired or wireless)

MAC address allocation administered by IEEE
manufacturer buys portion of MAC address space (to assure uniqueness)

analogy:
(a) MAC address: like Social Security Number
(b) IP address: like postal address

MAC flat address $\rightarrow$ portability
can move LAN card from one LAN to another

IP hierarchical address NOT portable
address depends on IP subnet to which node is attached
**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: Same LAN (network)**

- A wants to send datagram to B, and B’s MAC address not in A’s ARP table.
- A broadcasts ARP query packet, containing B’s IP address
  - dest MAC address = FF-FF-FF-FF-FF-FF
  - 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
Addressing: 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 IP 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 NIC sends frame
- R’s NIC receives frame
- R removes IP datagram from Ethernet frame, sees its destined to B
- R uses ARP to get B’s MAC address
- R creates frame containing A-to-B IP datagram sends to B

This is a really important example – make sure you understand!
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)
  - old-fashioned Ethernet
  - 802.11 wireless LAN
  - upstream shared channels (cable, satellite)

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
  - No magic...
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, packets are transmitted first come, first served
3. simple...

MAC Protocols: a taxonomy

Three broad classes:

- **Static allocation**
  - divide channel into smaller “pieces” (time slots, frequency, code)
  - allocate piece to node for exclusive use

- **“Taking turns”**
  - nodes take turns, but nodes with more to send can take longer turns
  - E.g., token passing

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

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
Problem with static allocation

- One channel at $C$ bps vs. $N$ subchannels, each at $C/N$ bps
- $L$ bits per packet
- Aggregate arrival rate is lambda
- Delivery time?

Dynamic Channel Allocation

- bandwidth allocated on a demand basis
- must be concerned about fairness
- question: how do nodes “agree” on which node should send when
“Taking Turns”

Example: Token passing
- control token passed from one node to next sequentially.
- token message
- Concerns?

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 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) ALOHA

- no synchronization
- when frame first arrives
  - transmit immediately
- stations listen while they send, and therefore detect collisions
- when collisions occur, stations retransmit after a random period of time

Window of Vulnerability

- For a given frame, the time window when no other frame may be transmitted if a collision is to be avoided.
- Length of window?
Slotted ALOHA

**Assumptions:**
- All frames same size
- Time divided into equal size slots (time to transmit 1 frame)
- Nodes start to transmit only slot beginning
- Nodes are synchronized
- If 2 or more nodes transmit in slot, all nodes detect collision

**Operation:**
- When node obtains fresh frame, 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
- Clock synchronization
Performance Comparison

Carrier Sense Multiple Access

- What an idea! Listen before sending!
- 1-persistent CSMA
  - Station listens to channel. When the channel becomes idle, the frame is transmitted with probability 1.0.
  - If a collision occurs, the station waits random amount of time and repeats.
  - The longer the propagation delay, the worse the performance of the protocol. Why?
  - Even if the propagation delay is 0, there will still be collisions. Why?
CSMA (cont.)

- non-persistent CSMA
  - Station senses channel. If idle, send. If busy, waits a random amount of time and repeat.
  - Better channel utilization and longer delays than 1-persistent CSMA. Why?

CSMA (cont.)

- p-persistent CSMA
  - Station ready to send senses the channel.
  - If channel is idle, transmit with probability p. With probability $q = 1 - p$, wait one time unit (typically propagation delay) and repeat procedure.
  - If busy, listen until idle and repeat.
ALOHA and CSMA

Performance

![Graph showing the comparison of channel utilization versus load for various random access protocols.](image)

**Fig. 4-4.** Comparison of the channel utilization versus load for various random access protocols.

- What can we conclude?

CSMA/CD (Collision Detection)

**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: received signal strength overwhelmed by local transmission strength

- How long after beginning transmission may it take to detect a collision?
**CSMA/CD operation**

- Typical behavior
  - Successful transmission
  - Followed by contention period
  - Repeat...

```
<table>
<thead>
<tr>
<th>Frame</th>
<th>Frame</th>
<th>Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmission Period</td>
<td>Contention Period</td>
<td>Idle Period</td>
</tr>
</tbody>
</table>

TIME
```

**Issue in Wireless LANs**

- CSMA-like protocols do not work well due to limited range
- Hidden terminal problem:
  - A sends to B, any transmission by C interferes
- Exposed terminal (station) problem:
  - B transmitting to A, C cannot send to D because channel appears busy

```
(a) A B C D Radio range
(b) A B C D
```

CSE 422 - McKinley
Link Layer
One Solution

- **MACA - Multiple Access Collision Avoidance**
  - Sending node sends Request-To-Send (RTS) frame
  - Receiving node sends Clear-To-Send (CTS) frame
  - Nodes within range remain quiet for period of time
  - RTS collisions can happen, but are rare and resolved by exponential backoff

Modified Solution

- **MACAW - MACA for Wireless**
- Adds link level acknowledgements
- Uses carrier sense to reduce RTS collisions
- Applies backoff algorithm on a SRC-DST basis
**Link Layer**

- Introduction and services
- Error detection and correction
- Multiple access protocols
- Link-layer Addressing
- Ethernet operation
- A day in the life of a web request

**IEEE 802.3 (Ethernet)**

- 1-persistent CSMA/CD
- Derives from original Ethernet network
  - built at Xerox, Metcalfe, Boggs et al
  - 2.94 Mbps,
  - later raised to 10 Mbps
  - 100 workstations, 1-km cable,
  - 802.3 covered a whole family of 1-persistent CSMA/CD systems, running at from 1 to 10 Mbps
Cabling Options

Frame structure

- preamble (7 bytes of 10101010)
- start-of-frame byte (10101011)
- destination address (allows multicast, broadcast)
- source address
- Type/length
- data
- pad (keeps minimum frame length at 64 bytes)
- Why?
- 32-bit checksum
802.3 Backoff Algorithm

- slot time set to 512 bit times
- after ith collision (up to 10) choose a random number of slots between 0 and $2^i - 1$ (helps minimize delay)
- failure after 16 collisions
- reason for this approach?

802.3 Performance

[Graph showing efficiency of 802.3 at 10 Mbps with 512-bit slot times]
Another study...

Channel Access Probability
## Star topology

- bus topology popular through mid 90s
  - all nodes in same collision domain (can collide with each other)
- today: star topology prevails
  - active switch in center
  - each “spoke” runs a (separate) Ethernet protocol (nodes do not collide with each other)

---

**Delay Comparison**

![Delay Comparison Chart](chart)

- ALOHA
- Slotted ALOHA
- CSMA/CD

---

*Link Layer*
**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

**Ethernet Frame Structure (more)**

- **Addresses:** 6 bytes
  - if adapter receives frame with matching destination address, or with broadcast address (e.g., ARP packet), it passes data in frame to network layer protocol
  - otherwise, adapter discards frame
- **Type:** indicates higher layer protocol (mostly IP but others possible, e.g., Novell IPX, AppleTalk)
- **CRC:** checked at receiver, if error is detected, frame is dropped
**Ethernet: Unreliable, connectionless**

- **connectionless**: No handshaking between sending and receiving NICs
- **unreliable**: receiving NIC doesn’t send acks or nacks to sending NIC
  - stream of datagrams passed to network layer can have gaps (missing datagrams)
  - gaps will be filled if app is using TCP
  - otherwise, app will see gaps
- Ethernet’s MAC protocol: unslotted **CSMA/CD**

**Ethernet CSMA/CD algorithm**

1. NIC receives datagram from network layer, creates frame
2. If NIC senses channel idle, starts frame transmission
   If NIC senses channel busy, waits until channel idle, then transmits
3. If NIC transmits entire frame without detecting another transmission, NIC is done with frame!
4. If NIC detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, NIC enters **exponential backoff**: after mth collision, NIC chooses K at random from \( \{0, 1, 2, ..., 2^m - 1\} \). NIC waits K·512 bit times, returns to Step 2
802.3 Ethernet Standards: Link & Physical Layers

- many different Ethernet standards
  - common MAC protocol and frame format
  - different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10G bps
  - different physical layer media: fiber, cable

```
application
  transport
network
  link
  physical
```

MAC protocol and frame format

<table>
<thead>
<tr>
<th>100BASE-TX</th>
<th>100BASE-T2</th>
<th>100BASE-FX</th>
</tr>
</thead>
<tbody>
<tr>
<td>100BASE-T4</td>
<td>100BASE-SX</td>
<td>100BASE-BX</td>
</tr>
</tbody>
</table>

copper (twister pair) physical layer  fiber physical layer

“Fast” Ethernet

- Fast Ethernet (802.3u) operates exactly as does regular 10Base-T Ethernet, except at a bit rate of 100 Mbps. That is, all fast Ethernet systems are based on hubs.
- Given the improvements in clocks and the short distances, Manchester encoding is not needed.
- Either hubs or switches may be used, and most handle mixes of 10/100 Mbps traffic.
Variations

- **100Base-T4**: Uses four (category 3) twisted pairs, max 100 meters, each at 25 MHz. One is dedicated in each direction, the other two “float.” Ternary signaling is used. How many signal elements? How many bits per element?

- **100Base-TX**: Uses two (one in each direction) category 5 twisted pairs, max 100 meters, operating at 125 MHz. Uses 4B5B coding, which maps 4 bits into 5 signals in patterns that guarantee some transitions.

- **100Base-FX**: Uses two multimode fibers, max 2000 meters, ON/OFF signaling.

Gigabit Ethernet (802.3z)

- **Two modes**
  - Switched full duplex: frames are buffered, no contention possible
  - Hub-based half duplex: emulates CSMA/CD
    - minimum packet length increased to 512 bytes, enabling 100 meter segments
    - frame bursting: chaining together small frames to reach 512 bytes
Gigabit Ethernet (802.3z)

- Achieving gigabit speeds on twisted pairs
  - use four twisted pairs in cable, and five voltage levels (00, 01, 10, 11 plus special control/framing signal)
  - clock at 125MHz, with 8 bits per clock
- How fast is 1Gbps? One millisecond is enough time to send 1953 512-byte packets.
- The receiver has to be ready!!

Link Layer

- Introduction and services
- Error detection and correction
- Multiple access protocols
- Link-layer Addressing
- Ethernet operation
- A day in the life of a web request
Synthesis: a day in the life of a web request

- journey down protocol stack complete!
  - application, transport, network, link
- putting-it-all-together: synthesis!
  - goal: identify, review, understand protocols (at all layers) involved in seemingly simple scenario: requesting www page
  - scenario: student attaches laptop to campus network, requests/receives www.google.com
A day in the life... connecting to the Internet

- connecting laptop needs to get its own IP address, addr of first-hop router, addr of DNS server: use **DHCP**
- DHCP request *encapsulated* in **UDP**, encapsulated in **IP**, encapsulated in **802.3** Ethernet
- Ethernet frame *broadcast* (dest: FFFFFFFF) on LAN, received at router running **DHCP** server
- Ethernet *demux’ed* to IP demux’ed, UDP demux’ed to **DHCP**

Client now has IP address, knows name & addr of DNS server, IP address of its first-hop router
A day in the life… ARP (before DNS, before HTTP)

- before sending HTTP request, need IP address of www.google.com:
  - DNS
- DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Ethernet frame. In order to send frame to router, need MAC address of router interface: ARP
- ARP query broadcast, received by router, which replies with ARP reply giving MAC address of router interface
- client now knows MAC address of first hop router, so can now send frame containing DNS query

A day in the life… using DNS

- IP datagram containing DNS query forwarded sent on LAN from client to 1st hop router
- IP datagram forwarded from campus network into comcast network, routed (tables created by RIP, OSPF, IS-IS and/or BGP routing protocols) to DNS server
- demux’ ed to DNS server
- DNS server replies to client with IP address of www.google.com
A day in the life… TCP connection carrying HTTP

- to send HTTP request, client first opens TCP socket to web server
- TCP SYN segment (step 1 in 3-way handshake) inter-domain routed to web server
- web server responds with TCP SYNACK (step 2 in 3-way handshake)
- TCP connection established!

A day in the life… HTTP request/reply

- web page finally (!!!) displayed
- HTTP request sent into TCP socket
- IP datagram containing HTTP request routed to www.google.com
- web server responds with HTTP reply (containing web page)
- IP datagram containing HTTP reply routed back to client
CSE 422 Summary

- Worked our way down the protocol stack
- Explored theoretical concepts (reliability, protocols, routing, error control, access control)
- Also saw how these are applied in the real Internet!
- Observed that rules of layering are often broken to provide good performance (sorry!)
- Learned that networks evolve!