Homework #4

• ICMP header:

```cpp
class ICMPHeader{
public:
    u_char type;   /* ICMP packet type */
    u_char code;   /* ICMP type subcode */
    u_short checksum;   /* checksum */
    u_short ID;   /* usually process ID */
    u_short seq;   /* sequence */
};
```

• Received ICMP pkts are delivered to all open ICMP sockets (since ICMP has no port numbers)
  - Routers will echo your entire IP packet in their TTL expired messages
  - Use the ID field to distinguish your pkts from junk
Homework #4

- TTL Expired pkt structure:

- Find out whether the ID field in the 4-th header matches your ID
Homework #4

• Other things to consider:
  – If your checksums are incorrect, the packet will likely be dropped and you won’t get a reply
  – If your firewall is enabled to block all incoming traffic, the kernel will not deliver ICMP packets

• In some Windows configurations, you must be admin to open ICMP sockets

• Mores caveats – read the handout!
  – UAC needs to be disabled or VS run as administrator
  – Custom in-bound firewall rules
  – Batch mode requires pinging the target before tracing
  – Hard limits on trace delay in batch mode
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
Summary
**Link Layer: Introduction**

**Terminology:**

- Hosts and routers are **nodes**
- Communication channels that connect adjacent nodes are **layer-3 links**
  - Wired or wireless
- Each link may contain multiple layer-2 devices (e.g., switches)

**Data-link layer** has responsibility of transferring IP datagram from one node to adjacent node over a single link
**Link Layer Services**

- **Framing:**
  - Add header, trailer to IP packet
  - Data-link addresses (completely independent of IP addresses) used in frame headers to identify source, dest

- **Link access:**
  - Channel access if shared medium

- **Flow control:**
  - Pacing between adjacent sending and receiving nodes

- **Error detection:**
  - Errors caused by signal attenuation, noise
  - Receiver detects presence of errors and signals data-link layer of adjacent node for retransmission or drops frame
**Link Layer Services**

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

- **Reliable delivery (rdt) between adjacent nodes**
  - Rdt 3.0 is a common technique (chapter 3)
  - Seldom used on low bit error links (fiber, twisted pair), but may be implemented in wireless networks

- **More terminology**
  - In *half-duplex* mode, nodes at both ends of link can transmit, but not at the same time
  - In *full-duplex*, bidirectional transfer happens concurrently
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
Summary
Multiple Access Links and Protocols

Two types of links:

- **Point-to-point, e.g.**:
  - PPP for dial-up and DSL access
  - Dedicated cable between Ethernet switch and host

- **Broadcast (shared wire/medium)**
  - Traditional Ethernet
  - Upstream HFC
  - 802.11 wireless LAN, satellite
Multiple Access Protocols

- Assume a single shared broadcast channel
- Two or more simultaneous transmissions by nodes is called interference or collision
  - Receiver cannot discern packets when multiple signals are jammed together

Link access protocol

- Distributed algorithm that determines how nodes share channel
- Communication about channel sharing must use the channel itself!
  - No out-of-band channel for coordination
- MAC (Media Access Control) layer = data-link layer = layer 2
Ideal Multiple Access Protocol

Desired properties

1. Single node can achieve full channel rate R (high utilization without competition)

2. When M nodes want to transmit, each can send at average rate R/M (fairness and high utilization during competition)

3. Fully decentralized:
   - No special node to coordinate transmissions
   - No synchronization of clocks

4. Simple
MAC Protocols: Taxonomy

Three broad classes:

- **Channel Partitioning**
  - Divide channel into smaller “pieces” (time slots, frequency, wavelengths)
  - 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” (time frames)
  - Each station gets fixed length slot in each round (1/N of frame time to each node), unused slots go idle

- Example: 6-station LAN

- Maximum throughput for a single user is $R / N$, which is far from ideal!
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
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 cause 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
Slotted ALOHA (1975)

Assumptions

- All packets same size
  - Time is divided into equal size slots, time to transmit 1 packet
- Nodes start transmission only at beginning of slots
  - Clocks are synchronized
- If 2 or more nodes transmit in slot, all nodes detect collision

Operation

- When node obtains fresh frame from IP, it transmits in the next time slot
- No collision, node can send new frame in next slot
- If collision, node retransmits frame in each subsequent slot with probability $p$ until success
Slotted ALOHA

Pros

- Single active node can continuously transmit at full rate of channel
- Reasonably decentralized: only slots need to be in sync
- Simple

Cons

- Collisions
- Idle/empty slots
- Full slot wasted on collision
- Accurate clock synchronization is still a headache
**Slotted Aloha Efficiency**

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

- Assume $N$ nodes with infinite data to send, each transmits in every slot with probability $p$.
- Probability that $k$ nodes transmit in slot is $\binom{N}{k} p^k (1-p)^{N-k}$.
- Prob that exactly one node transmits in a given slot (i.e., success) is $Np(1-p)^{N-1}$.

- For max efficiency with $N$ nodes, find $p$ that maximizes $Np(1-p)^{N-1}$.
- Optimal $p_0 = \frac{1}{N}$.
- For many nodes, take limit of $Np_0(1-p_0)^{N-1}$ as $N$ goes to infinity, which gives optimal efficiency $\frac{1}{e} = 0.37$.

Slotted Aloha with many users: channel utilization only 37%!
CSMA (Carrier Sense Multiple Access)

• Remove slots and allow transmission at any time

CSMA: listen before transmit

• If channel sensed idle, transmit entire frame

• If channel sensed busy, defer transmission
  – Human analogy: don’t interrupt others!

• If collision is detected at the end of transfer, wait a random period of time, then retransmit
  – Human analogy: talk until you’re done, pause, then repeat if someone else happened to start at the same time
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
- But now collisions are detected immediately
- Colliding transmissions aborted, reducing channel waste

• Human analogy: the polite conversationalist
• Collision detection:
  - Easy in wired LANs: measure signal strengths, compare transmitted, received signals
  - Difficult in wireless LANs: receiver shut off while transmitting
CSMA/CD Collision Detection
Features

TDMA/FDMA:
- 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:
- Efficient at low load: single node can fully utilize channel
- High load: potentially huge collision overhead

“Taking turns” protocols:
- Look for best of both worlds?
“Taking Turns” MAC Protocols

A) Polling:
• Coordinator “invites” other nodes to transmit in turn
• Concerns:
  – Polling overhead
  – Latency
  – Single point of failure (coordinator)

B) Token passing:
• Control token passed from one node to next sequentially
• Can send only if holding token
• Concerns:
  – Token overhead
  – Latency
  – Single point of failure (token)