Chapter 5
Link Layer and LANs

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 "adapter" (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

Adapted by Jelena Mirkovic
All material copyright 1996-2005 J.F. Kurose and K.W. Ross, All Rights Reserved
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

Two Dimensional Bit Parity:
Detect and correct single bit errors

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...

Checksmming: Cyclic Redundancy Check

- View data bits, D, as a binary number
- Choose r CRC bits, R, such that
  - \(<D, R> \text{ exactly divisible by } G \text{ (modulus 2)}\)
  - Receiver knows G, divides \(<D, R> \text{ by } G\). If non-zero remainder: error detected!
  - Can detect all burst errors less than r+1 bits
- Widely used in practice (ATM, HDLC)

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

\[
\begin{align*}
\text{G} & \quad 101 \quad 000 \quad 1011 \quad 1011 \quad 1001 \\
\text{G} & \quad 101 \quad 000 \quad 1010 \quad 1001 \\
\text{G} & \quad 110 \quad 000 \quad 1010 \\
\text{G} & \quad 1010 \quad 1101 \\
\text{G} & \quad 111 \\
\text{R} & \quad 111 
\end{align*}
\]