L3: Building Direct Link Networks I

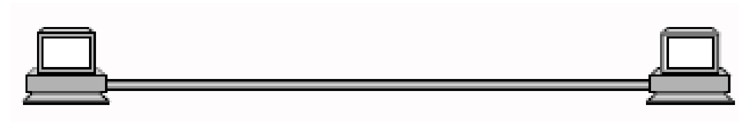
Hui Chen, Ph.D.
Dept. of Engineering & Computer Science
Virginia State University
Petersburg, VA 23806

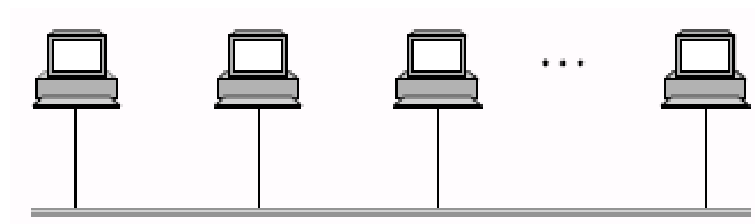
Acknowledgements

- Some pictures used in this presentation were obtained from the Internet
- The instructor used the following references
  - Larry L. Peterson and Bruce S. Davie, Computer Networks: A Systems Approach, 5th Edition, Elsevier, 2011
  - Andrew S. Tanenbaum, Computer Networks, 5th Edition, Prentice-Hall, 2010
  - James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach, 5th Ed., Addison Wesley, 2009
  - Larry L. Peterson’s (http://www.cs.princeton.edu/~llp/) Computer Networks class web site

Direct Link Networks

- Types of Networks
  - Point-to-point
  - Multiple access

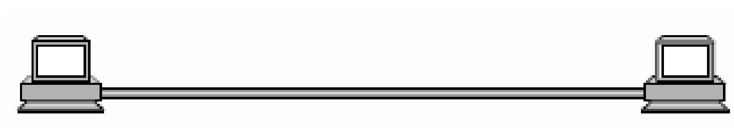


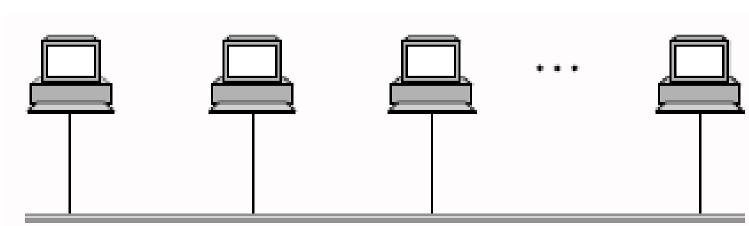


- **What problems do we need to solve to build even a direct link network?**

Direct Link Networks

- Types of Networks
  - Point-to-point
  - Multiple access





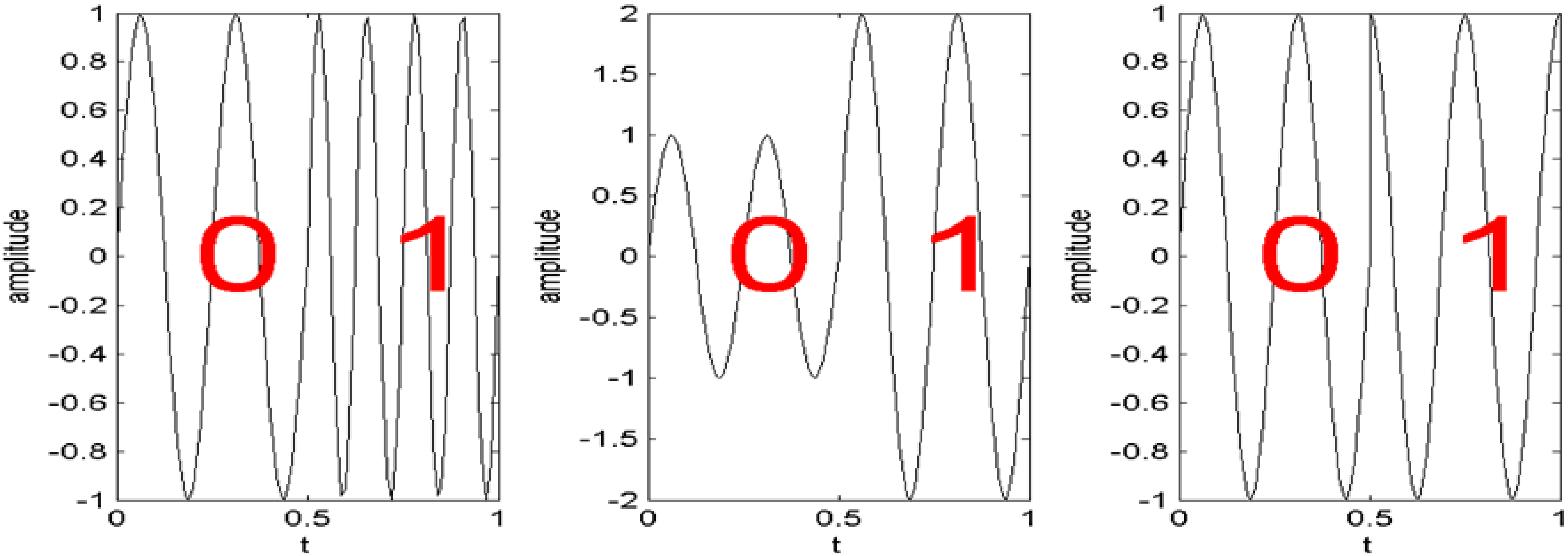
- Encoding
  - Encoding bits onto transmission medium
- Framing
  - Delineating sequence of bits into messages
- Error detection
  - Detecting errors and acting on them
- Reliable delivery
  - Making links appear reliable despite errors
- Media access control (specific to multiple access networks)
  - Mediating access to shared link

Physical Medium

- Guided media: transmission in “bounded media (wires)”
  - Examples: Twisted pair, coaxial cable, optical fiber
- Unguided media: transmission in “open space (wireless)”
  - Examples: Radio, Sonar
- Some wave oscillation propagates in the medium and carries signal
  - Frequency of carrier waves (or carrier signals)
- Problem: convert digital data to physical signal that is transmitted in a physical medium

Modulation

- Modulation
  - Varying frequency, amplitude, or phase of carrier signal with a modulating signal that carries information

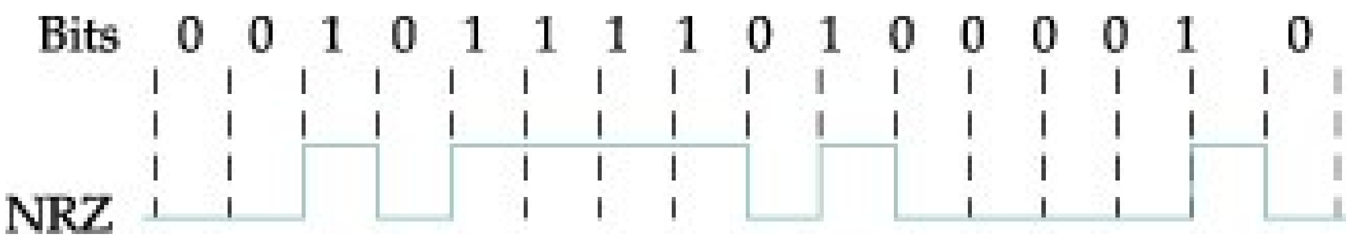


Encoding

- Encoding
  - Encode bits (binary data) into the signals
  - Modulation is not our focus
  - Assume working with two discrete signals: high and low
- Nonreturn-to-zero (NRZ)
- Nonreturn-to-zero-inverted (NRZI)
- Manchester
- 4B/5B

Non-Return-to-Zero (NRZ)

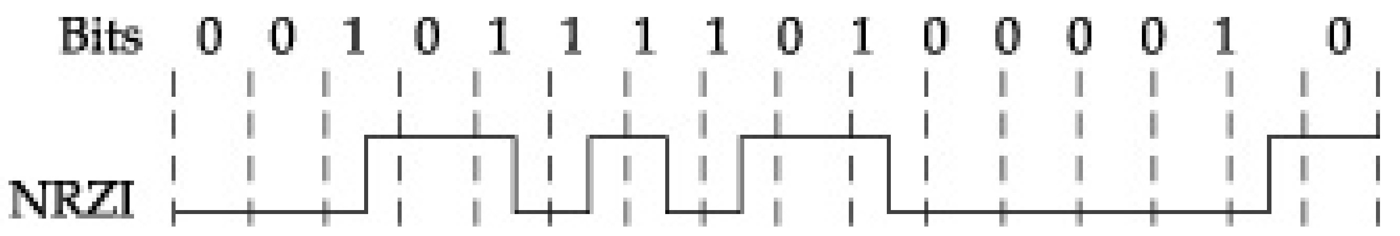
- Low → 0
- High → 1



- Long strings of 1s or 0s
  - Baseline wander
  - Difficult to recover clock

Non-Return-to-Zero-Inverted (NRZI)

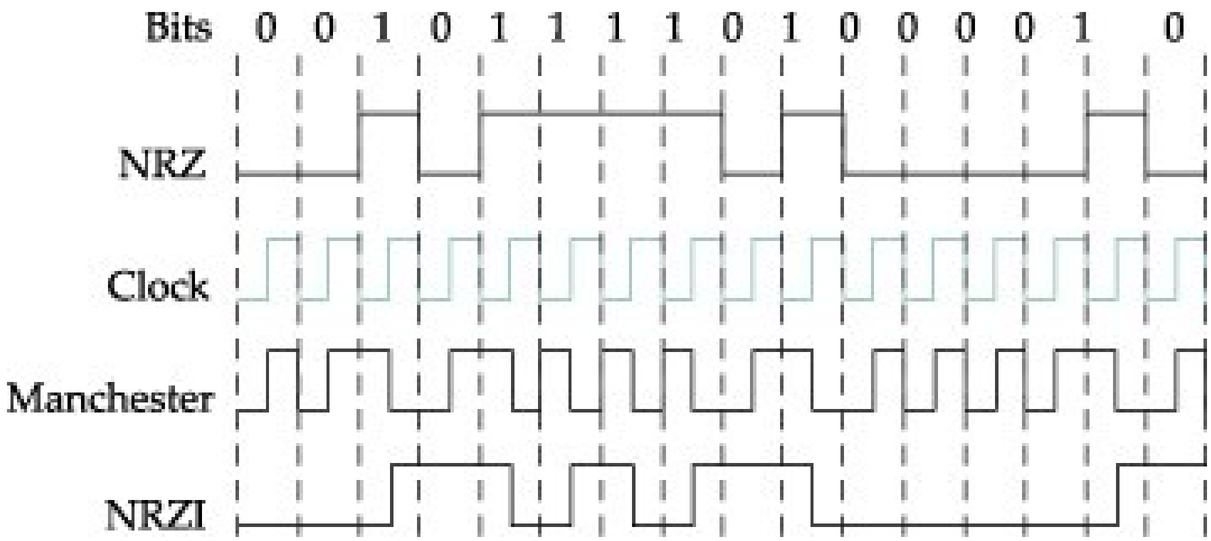
- Signal transition → 1
- No transition → 0



- Solve the problem caused by consecutive 1’s
- The problem caused by consecutive 0’s remains

Manchester

- 0 → low-to-high transition   1 → high-to-low transition
- That is, NRZ signal ⊕ Clock signal



- Solve the problems caused by both consecutive 1's and 0's
- New problem
  - 50% efficient: Two samples per clock cycle to detect a transition, bit rate is 50% of baud rate (rate of signal changes)

Exercise L3-1

- Encode bit sequence 01101 using NRZ, NRZI, Manchester encoding by indicating bits and by drawing clock, NRZ, NRZI, and Manchester signals as in [page 80 of the textbook]

4B/5B

- Motivations
  - To cope with baseline wander
  - To ease clock recovery
  - To increase encoding efficiency (bit rate/baud rate)

4B/5B

- Break long strings of repeated 0's and 1's
  - before transmission: 4 bits → 5 bits
    - Every 4 bits of actual data are encoded in a 5-bit code
    - 4 bits data symbols: $2^4 = 16$
    - 5 bits codes: $2^5 = 32$
    - No more than 1 leading 0's in codes and no more than 2 trailing 0's
    - No pair of 5-bit codes results in more than 3 consecutive 0's
    - 11111: line is idle   01101: control symbols
  - Transmit codes using NRZI: 80% efficiency

4B/5B Encoding

4-Bit Data Symbol	5-Bit Code
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111

4-Bit Data Symbol	5-Bit Code
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

Exercise L3-2

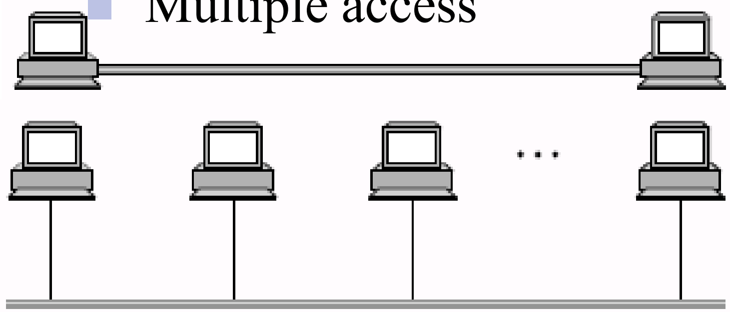
- Encode bit sequence 01011000 using 4B/5B encoding by showing the conversion between 4-bit sequences of data and their 5-bit sequences of codes and drawing clock and 4B/5B signals

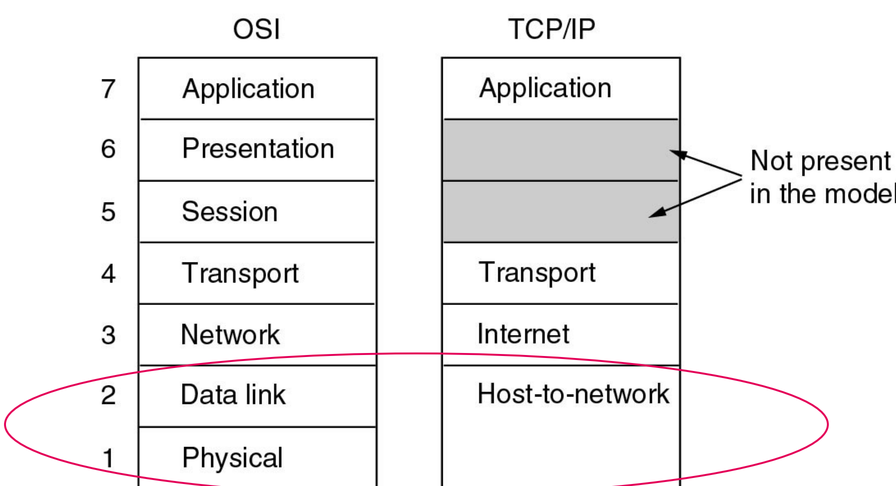
4-Bit Data Symbol	5-Bit Code
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111

4-Bit Data Symbol	5-Bit Code
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

Direct Link Networks

- Types of Networks
  - Point-to-point
  - Multiple access





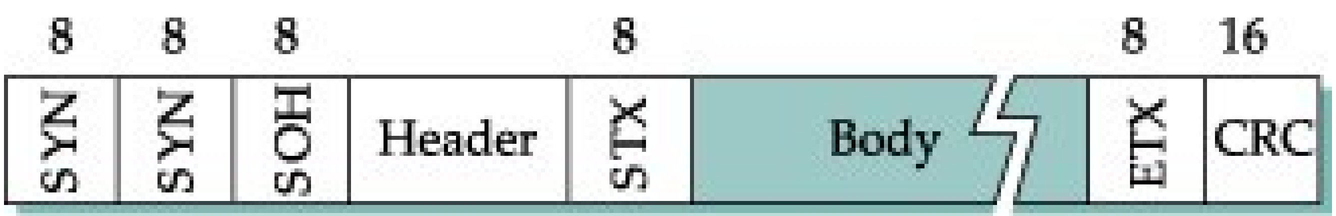
- Encoding
  - Encoding bits onto transmission medium
- **Framing**
  - **Delineating sequence of bits into messages**
- Error detection
  - Detecting errors and acting on them
- Reliable delivery
  - Making links appear reliable despite errors
- Media access control
  - Mediating access to shared link

Framing

- Packet-switched networks → block of data are exchanged between nodes
  - Breaking bits into frames
- Key issue: identity where a frame begins and the ends
- To discuss in the context of point-to-point links
  - Byte-oriented framing: frame as a collection of bytes
  - Bit-oriented framing: frame as a collection of bits
  - Clock-based framing (vs. sentinel-based approaches)
- Framing is a fundamental problem that must be addressed in multiple-access networks as well.

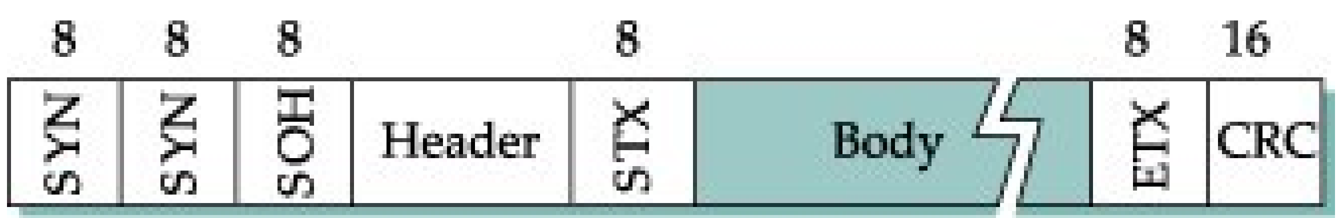
Byte-Oriented Framing

- Example: Binary Synchronous Communication (BISYNC)
- A frame is illustrated as a sequence of labeled fields



BISYNC

- Uses special characters known as *sentinel characters* to indicate whether frames begins and ends
- The beginning of a frame: SYN
- The data portion of the frame is contained between STX and ETX
- The start of header: SOH



BISYNC

- Problem: ETX may appear in the data
- Solution: escaping
  - ETX → DLE ETX
  - DLE → DLE DLE

Exercise L3-3

- DATA = 1A E2 02 2A 16 10 20. Use the sentinels in ASCII table, what would be the bytes in the *body* of the frame using BISYNC?
- In the *body* of a frame using BISYNC, is it possible to see the following byte sequence and why?

  4A 10 51 6B

Bit-Oriented Framing

- View a frame as a collection of bits
- Example: High-level Data Link Control (HDLC)

HDLC

- Beginning and ending sequence: 01111110
- 01111110 may appear in the body of the frame → bit stuffing
  - 11111 → 111110 → no 6 1's in a row in the body

Clock-based Framing

- Versus sentinel-based approaches
- Example: SONET
  - Popular in network backbone
  - Has a close tie with telephony systems
  - Specification takes entire book
  - Frame is of fixed length
    - 125 μs for STS-1 (51.84 Mbps)
      - Q: How big is a Mega (sidebar in page 45)
      - 125 μs = ? Bytes
  - Detect the frame header at each interval of frame size

Summary

- Bits → signal
  - Encoding
- Bits → frames
  - Framing
- Q: What if the link is not error free? In other words, what if a frame is corrupted?
  - Error detection
  - Reliable transmission