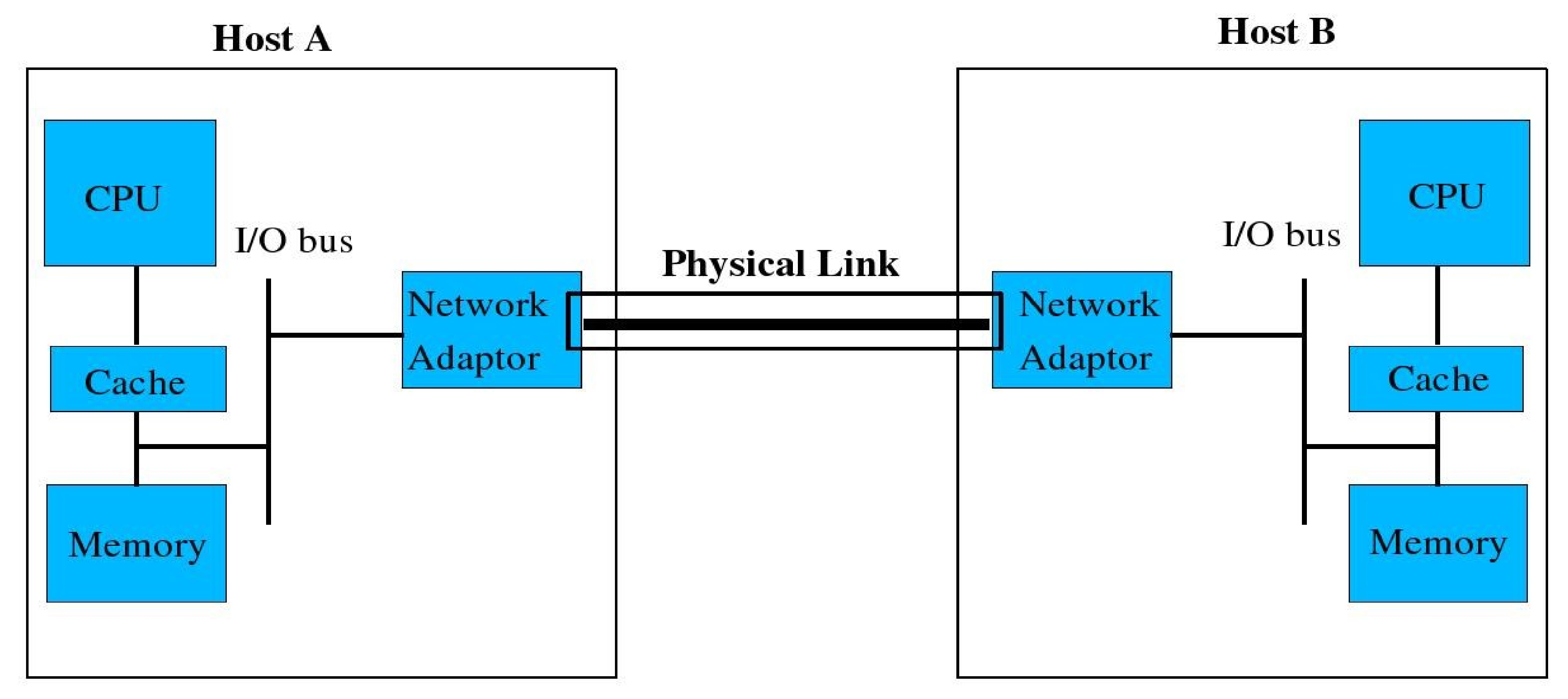
Physical and Data Link Layer

Kameswari Chebrolu
Dept. of Electrical Engineering, IIT Kanpur

Problem Statement

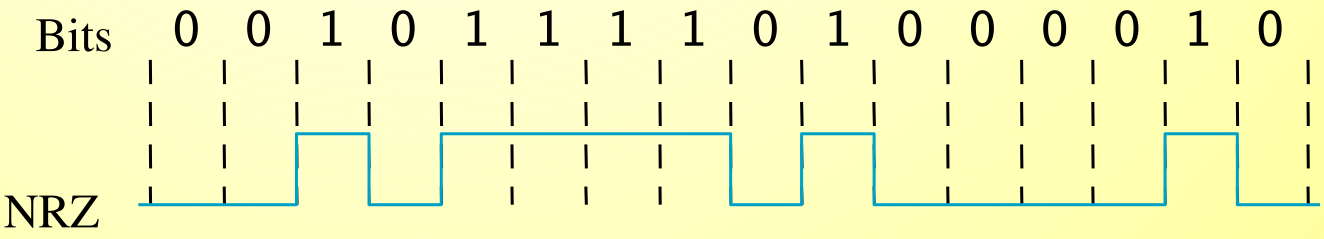
- Make two computers talk to each other



Pictures courtesy Peterson & Davie

Encoding

- Physical media transmit ***Analog*** signals
- Modulate/demodulate:
  - Encode/decode binary data into signals
  - E.g. Non-return to Zero (NRZ)
    - 0 as low signal and 1 as high signal



Picture courtesy Peterson & Davie

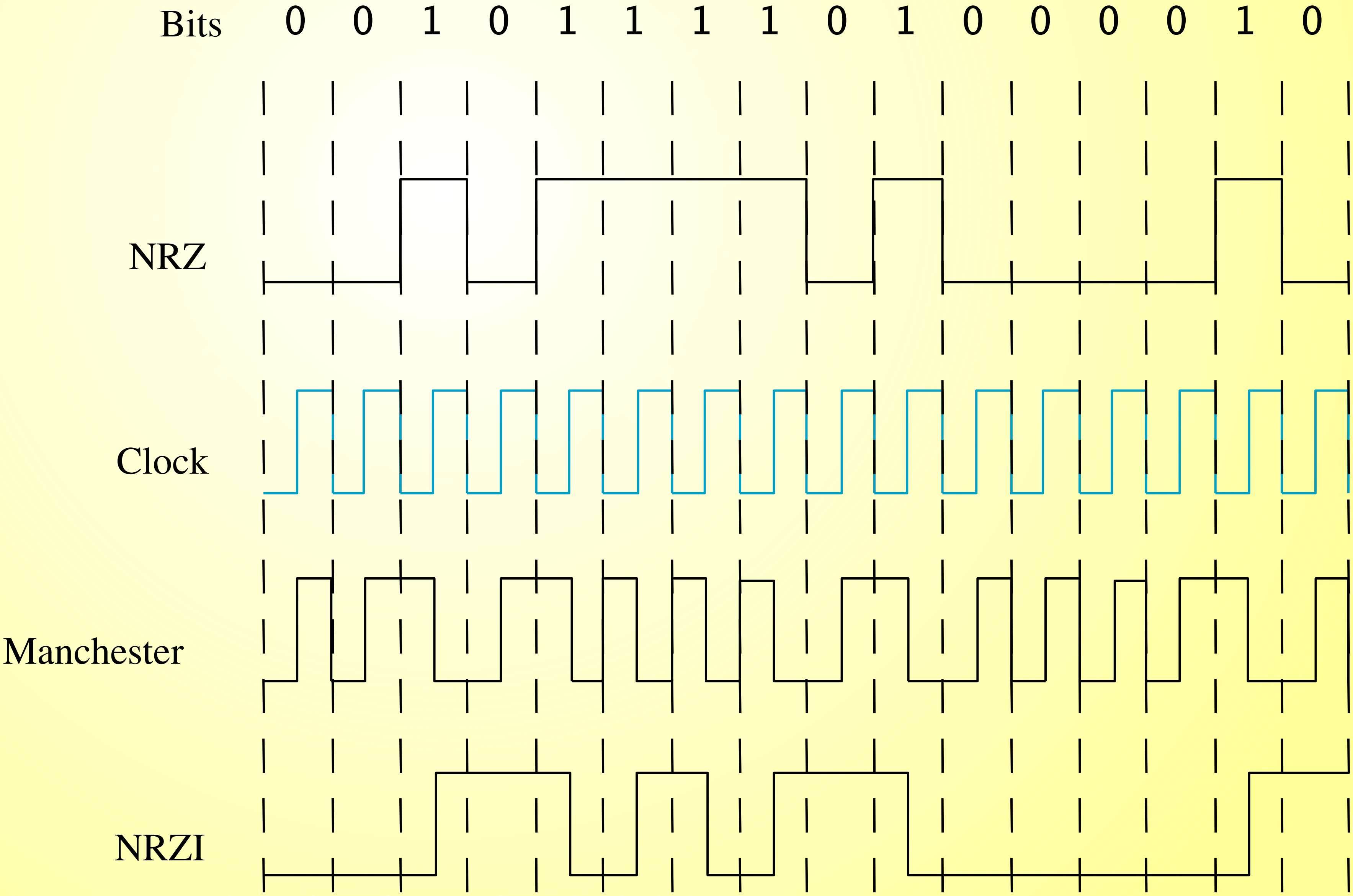
Problems with NRZ

- Consecutive 1s and 0s
  - Changes the average making it difficult to detect signals (*baseline wander*)
  - Clock Recovery
    - Sender's and receiver clocks have to be precisely synchronized
    - Receiver derives the clock from the received signal vis signal transition
    - Lesser number of transitions leads to clock drift

Alternative Encodings

- Non-return to Zero Inverted (NRZI)
  - To encode a 1, make a transition
  - To encode a 0, stay at the current signal
  - Solves problem of consecutive 1's but not 0's
- Manchester Encoding
  - Transmits XOR of the NRZ encoded data and the clock
    - 0 is encoded as low-to-high transition, 1 as high-to-low transition
  - Only 50% efficient

Example

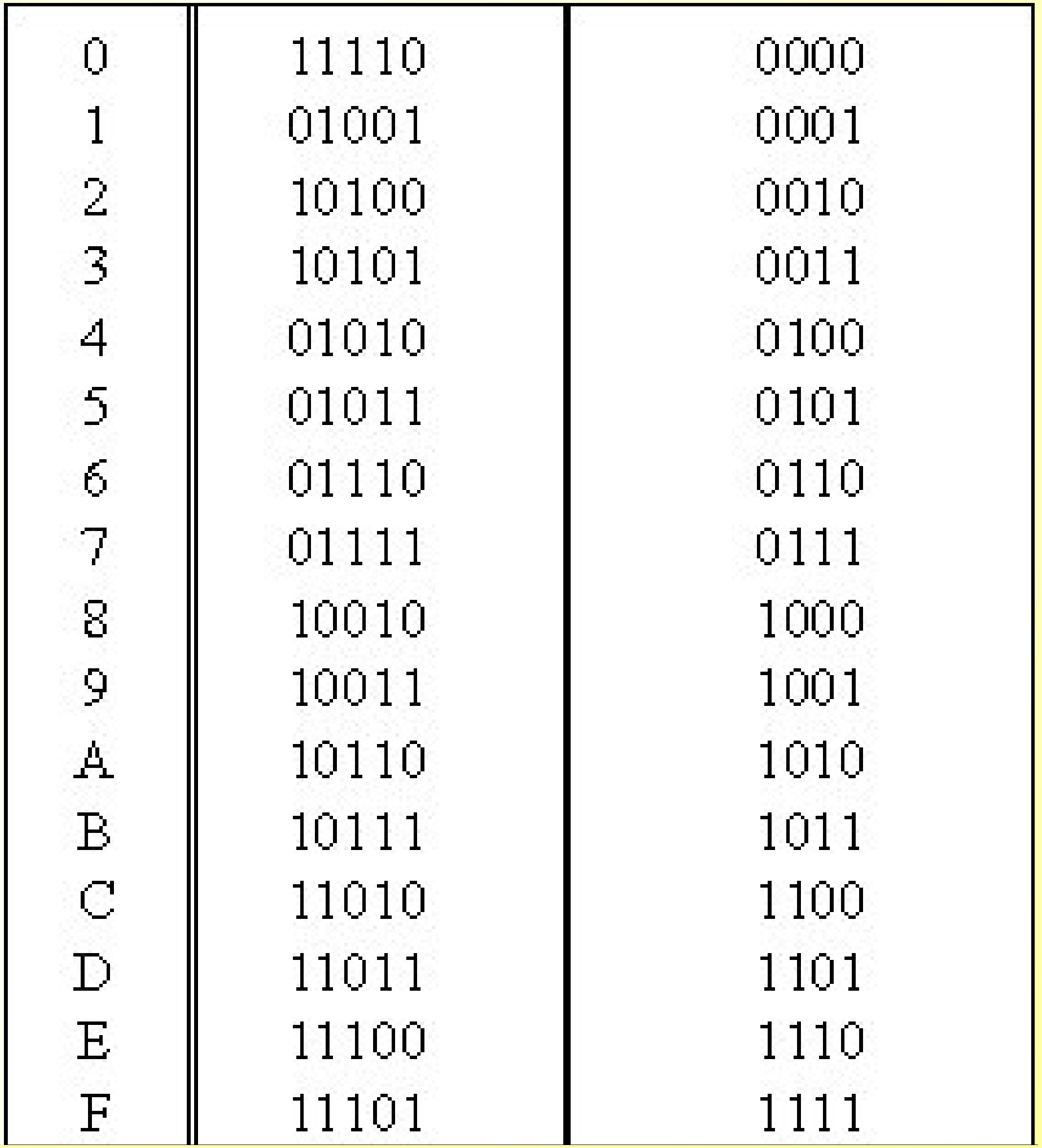


Picture courtesy Peterson & Davie

4B/5B Encoding

- Every 4 bit of actual data is encoded into a 5 bit code
- The 5 bit code words have
  - No more than one leading 0
  - No more than two trailing 0s
  - Solves consecutive zeros problem
- The 5 bit codes are sent using NRZI
- Achieves 80% efficiency

4B/5B Encoding



0	11110	0000
1	01001	0001
2	10100	0010
3	10101	0011
4	01010	0100
5	01011	0101
6	01110	0110
7	01111	0111
8	10010	1000
9	10011	1001
A	10110	1010
B	10111	1011
C	11010	1100
D	11011	1101
E	11100	1110
F	11101	1111

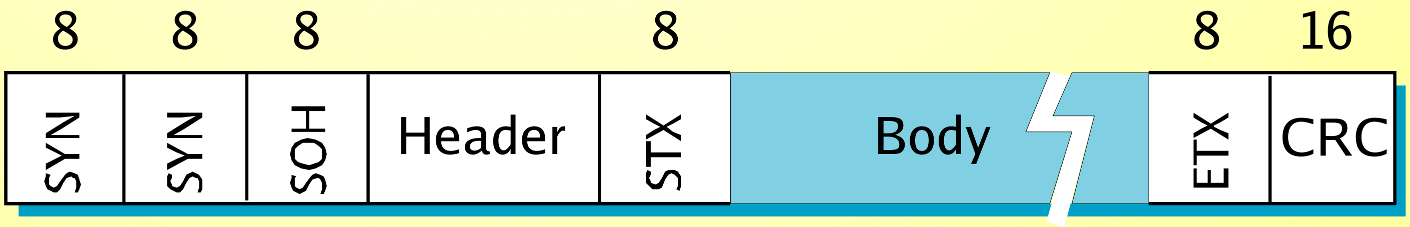
Picture courtesy Google

Framing

- Blocks of data exchanged between nodes
- We know how to transmit sequence of bits over a link
- Challenge: What sets of bits constitute a **frame**
  - Where is the beginning and the end of frame?
- Framing Protocols
  - Examples: PPP, HDLC, DDCMP

Byte Oriented protocols

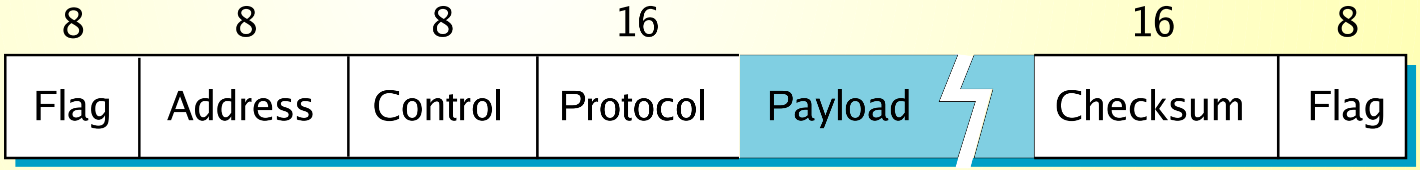
- Frame is a collection of bytes and not bit stream
- Sentinel approach
  - Use a special byte to mark beginning and end of frame
- Example: BISYNC (binary synchronous communication) protocol (developed by IBM)
  - Character Stuffing: Escape ETX with DLE (data-link-escape)
  - Escape DLE with another DLE



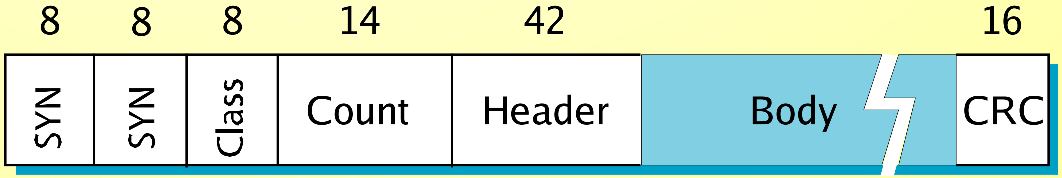
Picture courtesy Peterson & Davie

Cont...

- Example: Point to Point protocol (PPP)
  - Flag is 01111110



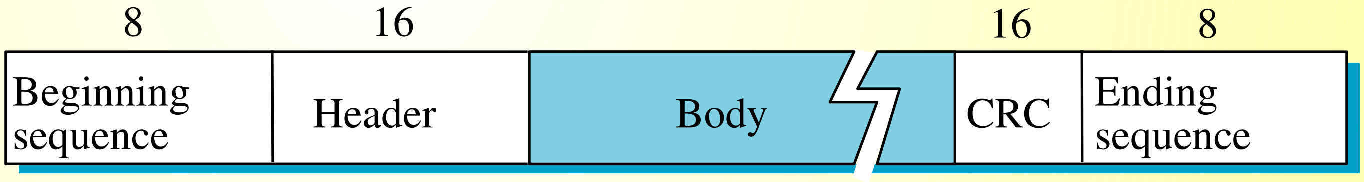
- Byte Counting approach
  - Include number of bytes contained in the frame in the header
  - Example: DDCMP



Picture courtesy Peterson & Davie

Bit-Oriented Protocols

- Not concerned with byte boundaries
- Example: High-Level Data Link Control (HDLC)



  - Sequence is 01111110
  - Special pattern may appear in payload
  - Solution: Bit Stuffing
    - Sender inserts a 0 after 5 consecutive 1's
    - Receiver removes the 0 that follows 5 1's

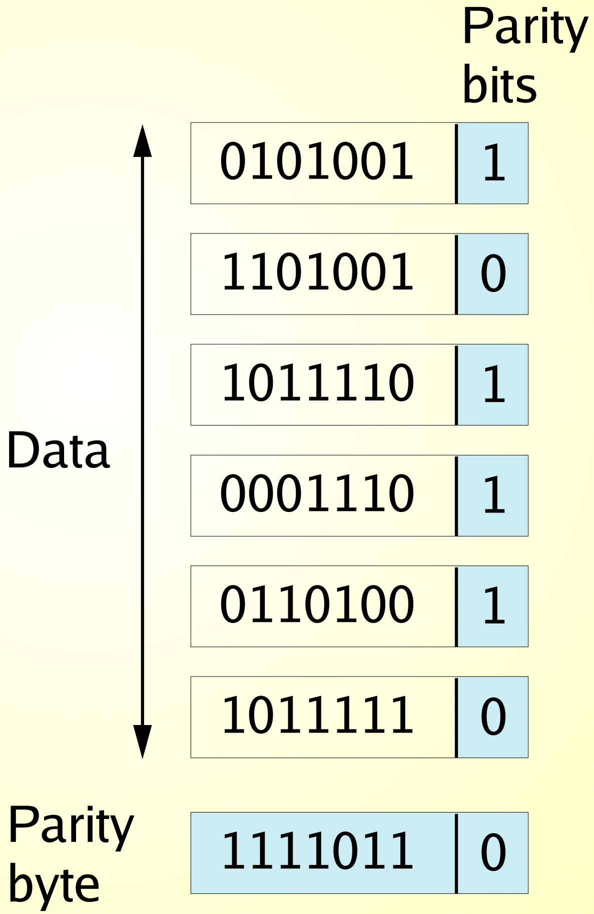
Picture courtesy Peterson & Davie

Error Detection

- Links suffer from errors
- Basic Idea: Add redundant information to a frame
  - At Sender
    - Add k bits of redundant data to a n bit message
    - k << n; k = 32; n = 12,000 for Ethernet
    - k derived from original message through some algorithm
  - At Receiver
    - Reapply same algorithm as sender
    - Redundant bits must match; take corrective action otherwise

Two Dimensional Parity

- Example:



		Parity bits
Data	0101001	1
	1101001	0
	1011110	1
	0001110	1
	0110100	1
	1011111	0
Parity byte	1111011	0

- 14 bits of redundancy for a 42 bit message
- Catches all 1,2,3 bit errors and most 4 bit errors
- Used by BISYNC protocol for ASCII characters

Picture courtesy Peterson & Davie

Internet Checksum

- View data in a frame to be transmitted as a sequence of 16-bit integers.
- Add the integers using 16 bit one's complement arithmetic.
- Take the one's complement of the result – this result is the checksum
- Not very strong in detecting errors

Cyclic Redundancy Check (CRC)

- Uses powerful math based on finite fields
- Represent a n+1 bit message with a polynomial of degree n, Message polynomial M(x)
  - 11000101 = x7 + x6 +x2 + 1
- Sender and receiver agree on a divisor polynomial C(x)
  - C(x) = x3+x2+1 (degree k = 3)
  - Choice of C(x) significantly effects error detection
    - Ethernet uses CRC of 32 bits, HDLC, DDCMP use 16 bits
    - x32+x26+x23+x22+x16+x12+x11+x10+x8+x7+x5+x4+x2+x+1

Cont.....

- Sender sends n+1+k bits => Transmitted message P(x)
- We contrive to make P(x) exactly divisibly by C(x)
- Received message R(x)
  - No errors: R(x) = P(x), exactly divisible by C(x)
  - Errors: R(x) ~= P(x); likely not divisible by C(x)
- Polynomial Algebra Rules:

Generate P(x)

- Multiply M(x) by x^k to get T(x)
  - Add k zeros at the end of the message
- Divide T(x) by C(x) to get remainder
- Subtract remainder from T(x) to get P(x)
- P(x) is now exactly divisible by C(x)
- Example:

Error Correction

- More complex math
- Needs more redundancy
- Tradeoff between error detection and correction
  - Error detection needs another copy to be transmitted if error is detected
    - Wastes bandwidth and introduces latency
  - Error detection requires more bits to be sent when errors occur
  - Error Correction requires more bits to be sent all the time