CPE348: Introduction to Computer Networks

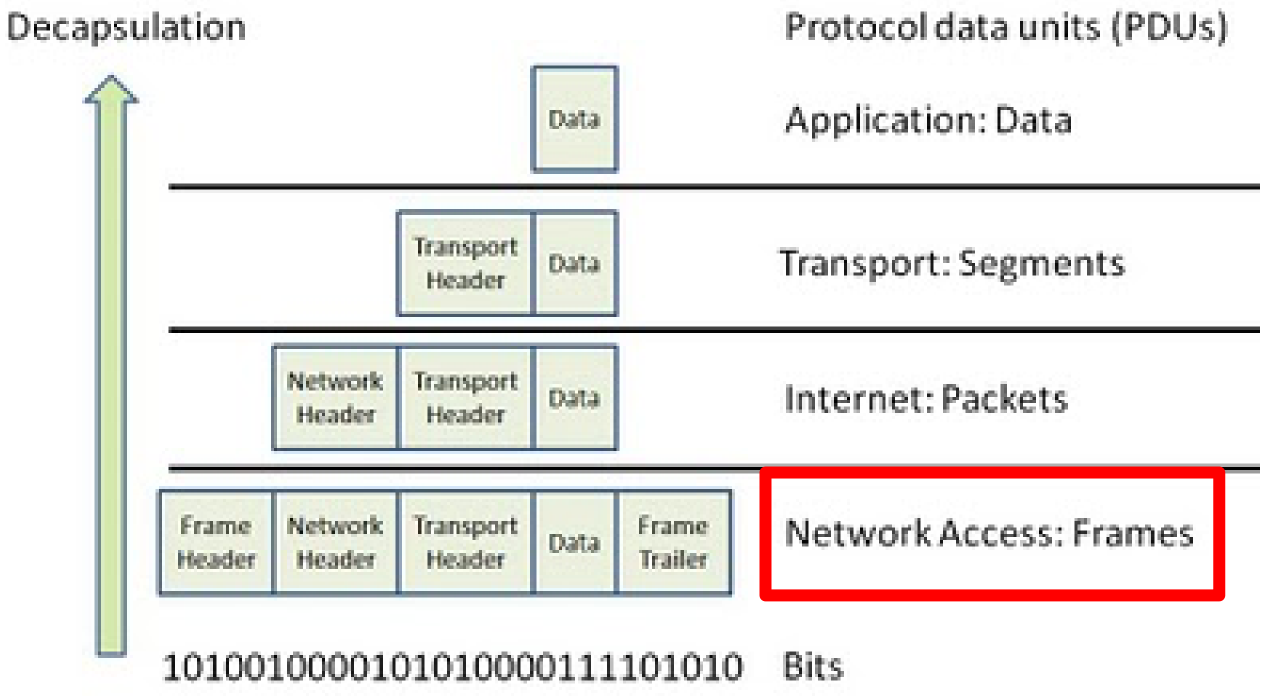
Lecture #5: Chapter 2.2

Jianqing Liu
Assistant Professor of Electrical and Computer Engineering, University of Alabama in Huntsville

jianqing.liu@uah.edu
http://jianqingliu.net

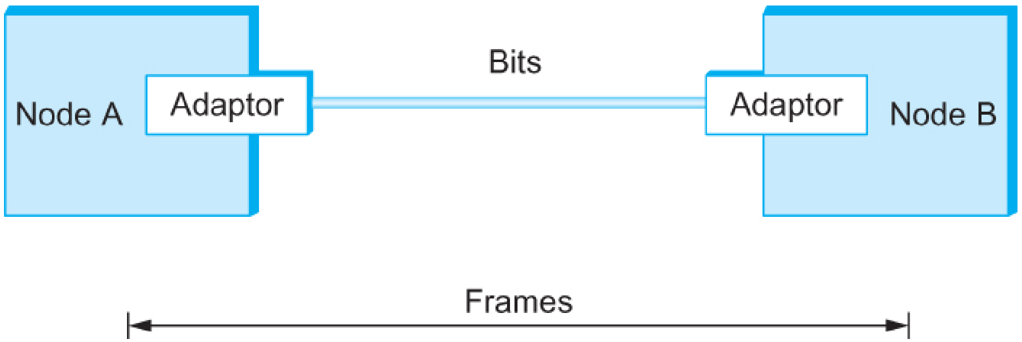
Framing

- What is the name of a data unit?



Framing

- Blocks of data (i.e., frames), not bit streams, are exchanged between nodes.
- It is the network adaptor that enables the nodes to exchange frames.



Bits flow between adaptors, frames between hosts

Framing

- What is important?
  - determining where the frame begins and ends
  - Distinguishing data payload and header

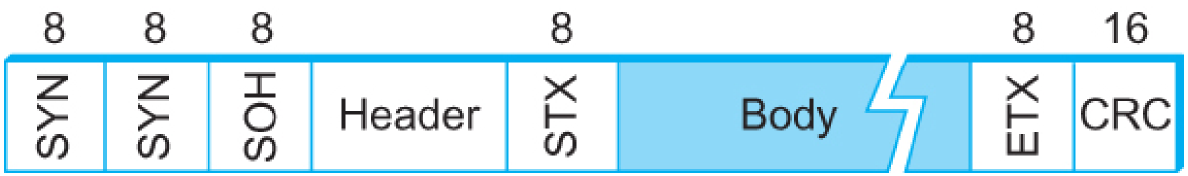
Framing

- Byte-oriented Protocols
  - To view each frame as a collection (i.e., multiplier) of bytes (characters) rather than bits.
  - Three examples
    - BISYNC (Binary Synchronous Communication) Protocol
    - PPP (Point-to-Point Protocol)
    - DDCMP (Digital Data Communication Protocol)

Framing

- BISYNC – sentinel approach
  - Frames transmitted beginning with leftmost field
  - Begin with a SYN (synchronize) character
  - Data portion is between special sentinel character STX (start of text) and ETX (end of text)
  - SOH : Start of Header
  - DLE : Data Link Escape
    - Used to precede a data portion that is equivalent to ETX
    - Used to precede a data portion that is equivalent to DLE
  - CRC: Cyclic Redundancy Check

Framing



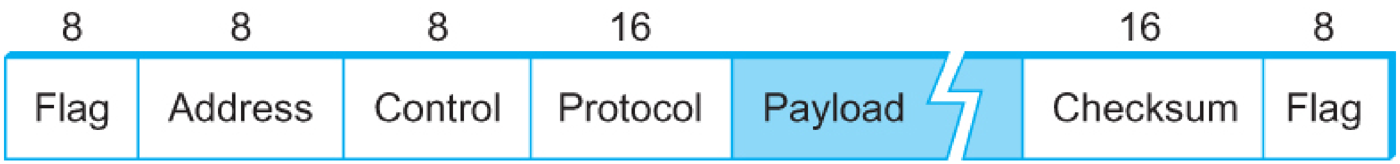
BISYNC Frame Format

Framing

- Point-to-point protocol (PPP)
  - Special start of text character denoted as Flag

    0 1 1 1 1 1 1 0
  - Address, control : default numbers
  - Protocol for demux : IP / IPX
  - Payload : negotiated (default:1500 bytes)
  - Checksum : for error detection
  - Several field sizes negotiated – Link control protocol (LCP)

Framing

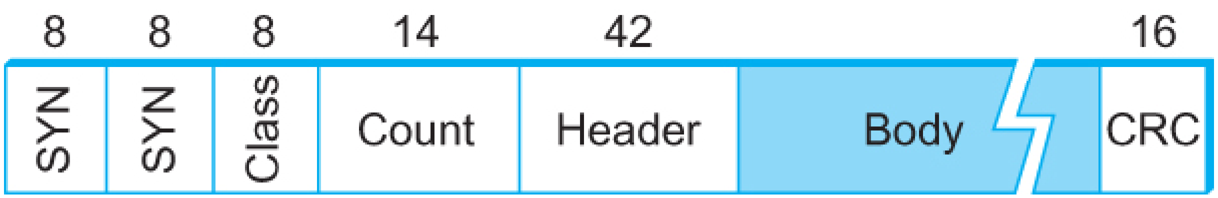


PPP Frame Format

Framing

- DDCMP (Digital Data Communication Message Protocol)
  - *count* : how many bytes are contained in the frame body
  - If *count* is corrupted
    - Framing error
    - Usually results in back-to-back frames received in error

Framing

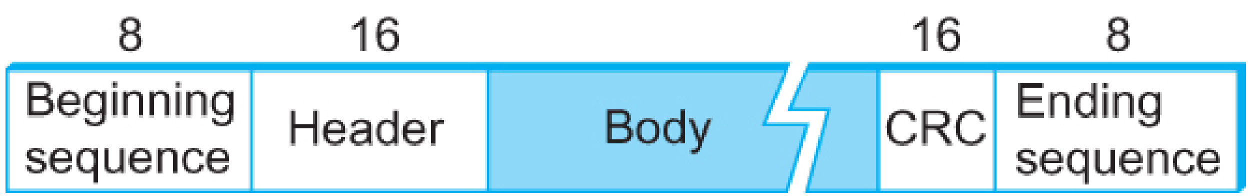


DDCMP Frame Format

Framing

- Bit-oriented Protocol
  - HDLC : High Level Data Link Control
    - Beginning and Ending Sequences

      0 1 1 1 1 1 1 0
    - Send 0 1 1 1 1 1 1 0 when idle



HDLC Frame Format

Framing

- HDLC Protocol – Sending Side
  - any time five consecutive 1's transmitted from the body of the message
  - The sender inserts 0 before transmitting the next bit (bit stuffing)

Framing

- HDLC Protocol - On the receiving side
  - Receive 5 consecutive 1's
    - Next bit 0 : Stuffed, so discard it, continue with data
    - Next bit 1 : Either End of the frame marker,
      Or Error introduced in the bitstream

      Look at the next bit after the 6^{th} 1

      If 0 (01111110) → End of the frame marker

      If 1 (01111111) → Error, discard the whole frame

      The receiver needs to wait for next 01111110 before it can start receiving again

Error Detection

- Bit errors are introduced into frames
  - Because of electrical interference
  - Because of thermal noises
- Need a mechanism to
  - Detect any errors
  - Notify of or correct errors

Error Detection

- Two approaches when the recipient detects an error
  - Notify the sender that the message was corrupted, so the sender can send again.
    - Repeat re-transmission till correctly receiving it or time expiration
  - Using some error correct detection and correction algorithm, the receiver reconstructs the message

Error Detection

- Common techniques for detecting transmission error
  - CRC (Cyclic Redundancy Check)
    - Used in HDLC, DDCMP, CSMA/CD, Token Ring
  - Two Dimensional Parity (BISYNC)
  - Checksum (IP)

Error Detection is implemented in different OSI layers!

Why?

Error Detection

- Basic Idea of Error Detection
  - Add redundant information to a frame that can be used to determine if errors have been introduced
    - Extreme Case: Transmitting two complete copies of data
      - Identical → No error
      - Differ → Error
      - Poor Scheme? – not very efficient
        - n bit message, n bit redundant information
        - Error can go undetected
    - In general, we can provide strong error detection technique
      - k redundant bits, n bits message, k << n
      - In Ethernet, a frame carrying up to 12,000 bits of data requires only 32-bit CRC

Error Detection

- Extra bits are redundant
  - Derived from the original message using some algorithms
  - Both the sender and receiver know the algorithm

Sender	
m	**r**

Receiver	
m'	**r'**

Receiver receives *m'* and *r'* computes *r* using *m'*

If the computed *r* matches the received *r'*, then message is assumed to be error free

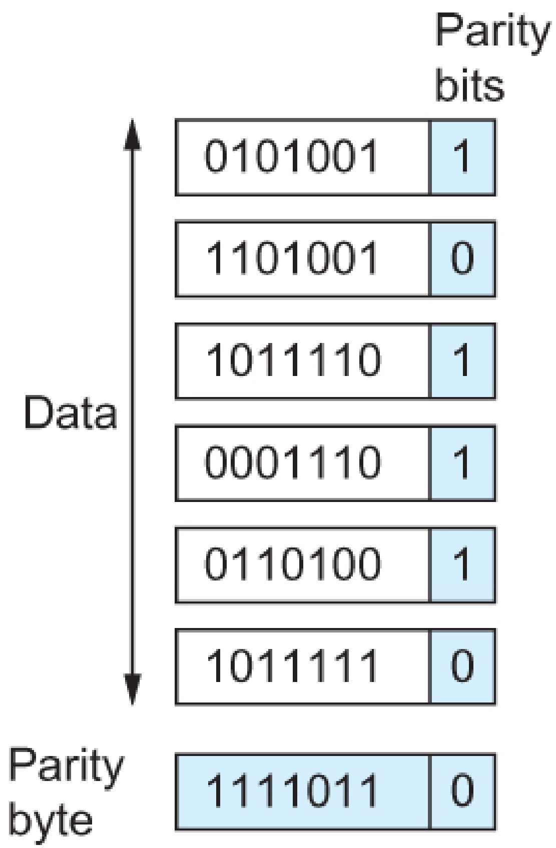
Two-dimensional parity

- Two-dimensional parity is exactly what the name suggests.
- One-dimensional parity is,
  - adding one extra bit to a 7-bit code
  - Odd parity sets the eighth bit to 1 if needed to give an odd number of 1s in the byte, and
  - Even parity sets the eighth bit to 1 if needed to give an even number of 1s in the byte

Two-dimensional parity

- Two-dimensional parity catches all 1-, 2-, and 3-bit errors and most 4-bit errors
- Or, how many bit errors does the 2-D parity code guarantee to detect?
- In the following example, 42 bits of data and 14 bits of parity (redundant bits)

Two-dimensional parity



Two Dimensional Parity (using even parity)

Internet Checksum Algorithm

- Used at the network level
- Add up all the words that are transmitted and then transmit the result of that sum
  - The result is called the checksum
- The receiver performs the same calculation on the received data and compares the result with the received checksum

Internet Checksum Algorithm

- Consider the data being checksummed as a sequence of 16-bit integers.
- Add them together using 16-bit ones complement and then take the ones complement of the result.
- Not a good detector of bit errors – however it is easy to implement in software

Why not use the same error detection technique in Layer 2?

Internet Checksum Algorithm

- In ones complement arithmetic, a negative integer −x is represented as the complement of x;
  - Each bit of x is inverted.
- When adding numbers in ones complement arithmetic, a carryout from the most significant bit needs to be added to the result.

Internet Checksum Algorithm

- Consider, for example, the addition of −5 and −3 in ones complement arithmetic on 4-bit integers
  - +5 is 0101, so −5 is 1010; +3 is 0011, so −3 is 1100
- If we add 1010 and 1100 ignoring the carry, we get 0110
- In ones complement arithmetic, the fact that this operation caused a carry from the most significant bit causes us to increment the result, giving 0111, which is the ones complement representation of −8 (obtained by inverting the bits in 1000), as we would expect

Cyclic Redundancy Check (CRC)

- Reduce the number of extra bits and maximize protection
- Given a bit string 110001 we can associate a polynomial on a single variable *x* for it.

  $1*x^5+1*x^4+0*x^3+0*x^2+0*x^1+1*x^0 = x^5+x^4+1$ and the degree is 5.

  A *k*-bit frame has a maximum degree of *k*-1
- Let M(x) be a message polynomial and C(x) be a generator polynomial – pulled from a table.

Cyclic Redundancy Check (CRC)

- Let M(x)/C(x) leave a remainder of 0.
- When M(x) is sent and M'(x) is received we have M'(x) = M(x)+E(x)
- The receiver computes M'(x)/C(x) and if the remainder is nonzero, then an error has occurred.
- The only thing the sender and the receiver should know is C(x).

Cyclic Redundancy Check (CRC)

- Let M(x) be a frame with *m* bits and let the generator polynomial have less than *m* bits say equal to *k*.
- Let *r* be the degree of C(x) where r = k-1
- Append *r* zero bits to the low-order end of the frame, so it now contains *m*+*r* bits and corresponds to the polynomial $x^rM(x)$.

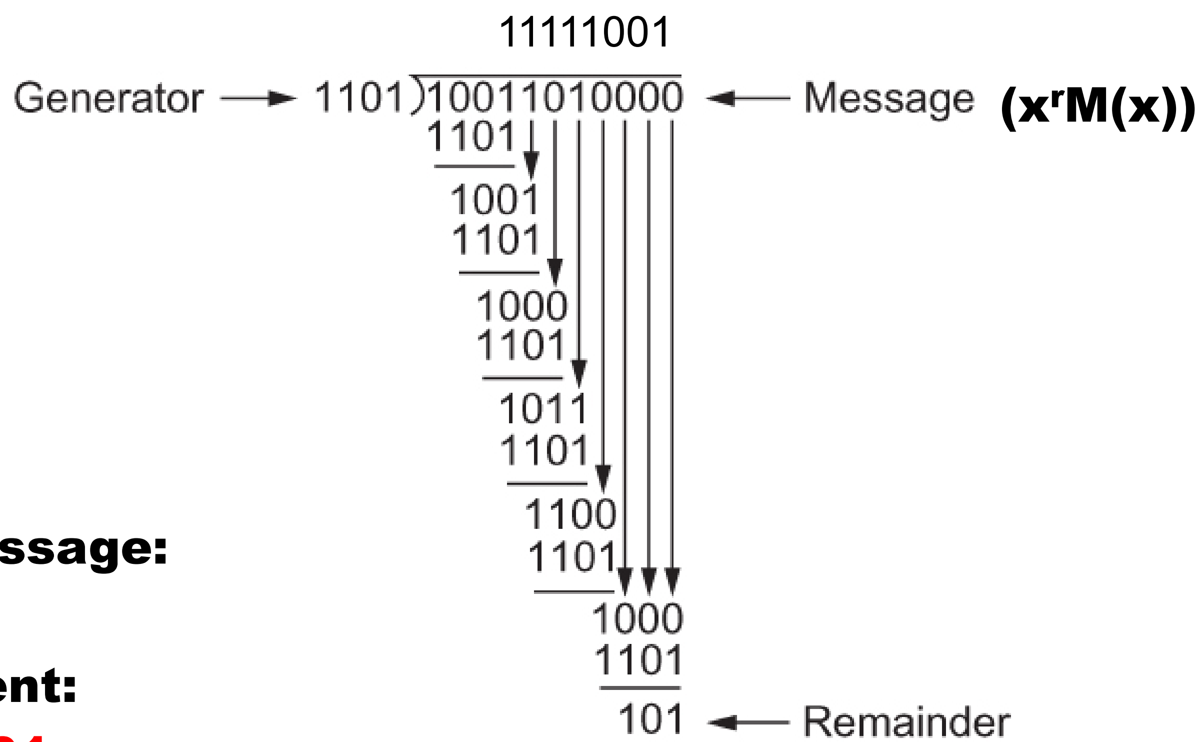
If $M(x) = x^5+x^4+1$ and $C(x) = x^3+x^2+1$ ➔ r=3

Then $x^rM(x) = x^8+x^7+x^3$

Cyclic Redundancy Check (CRC)

- Divide the bit string corresponding to $x^rM(x)$ by the bit string corresponding to $C(x)$ using modulo 2 division.
- Subtract the remainder (which is always *r* or fewer bits) from the string corresponding to $x^rM(x)$ using modulo 2 subtraction (addition and subtraction are the same in modulo 2). XOR
- The result is to be transmitted. Call it polynomial M'(x).

Cyclic Redundancy Check (CRC)



Original Message:

10011010

Message sent:

10011010101

CRC Calculation using Polynomial Long Division

Cyclic Redundancy Check (CRC)

- Properties of Generator Polynomial
  - It is possible to detect the following types of errors by a C(x) with degree r
    - All single-bit errors, as long as the x^r and x^0 terms have nonzero coefficients.
    - All double-bit errors, as long as C(x) has a factor with at least three terms.
    - Any odd number of errors, as long as C(x) contains the factor (x+1).
    - Any “burst” error (i.e., sequence of consecutive error bits) for which the length of the burst is less than *r* bits. (Most burst errors of larger than *r* bits can also be detected.)

Cyclic Redundancy Check (CRC)

- Six generator polynomials that have become international standards are:
  - CRC-8 = x^8+x^2+x+1
  - CRC-10 = $x^{10}+x^9+x^5+x^4+x+1$
  - CRC-12 = $x^{12}+x^{11}+x^3+x^2+x+1$
  - CRC-16 = $x^{16}+x^{15}+x^2+1$
  - CRC-CCITT = $x^{16}+x^{12}+x^5+1$
  - CRC-32 = $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$