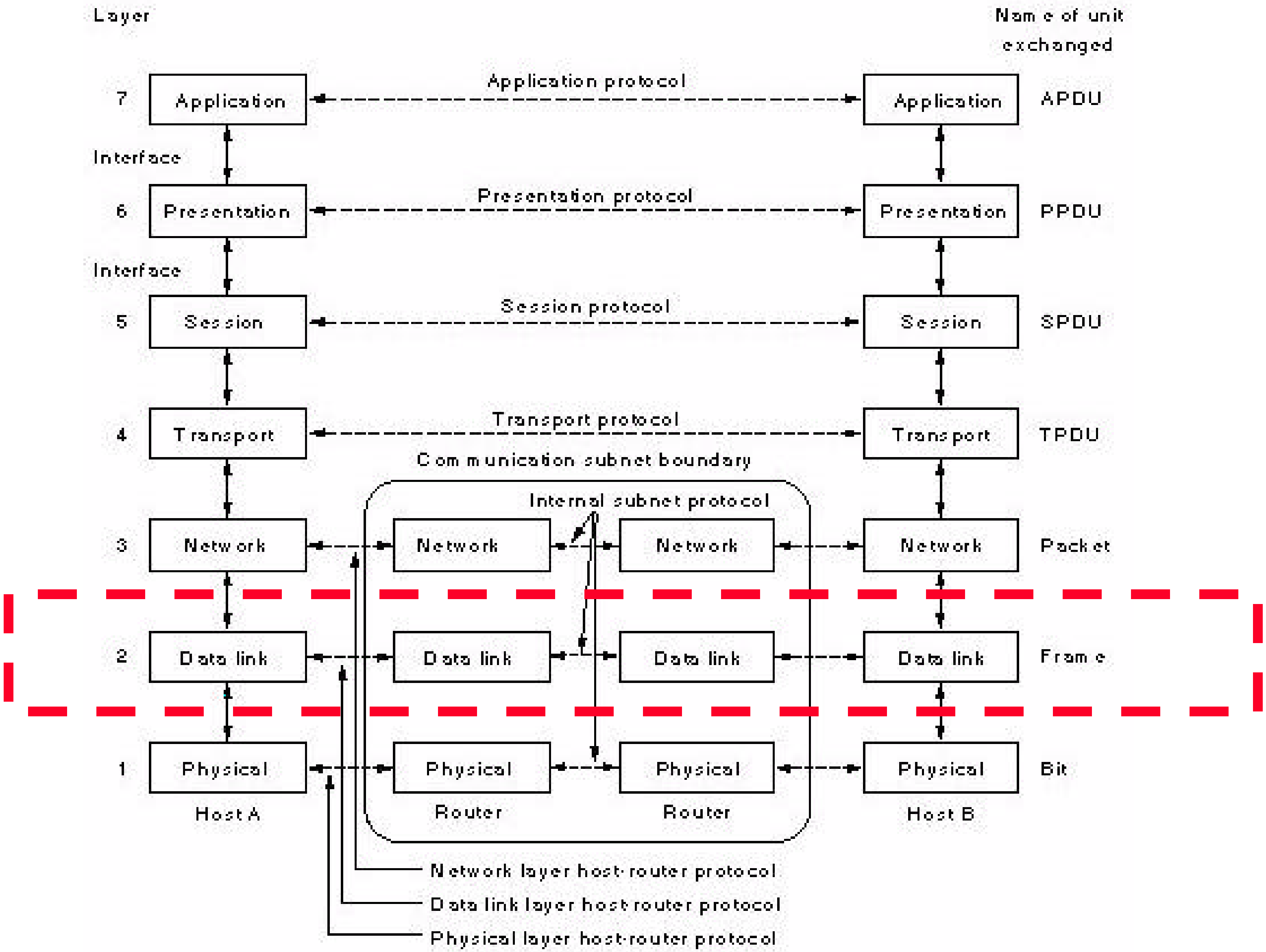
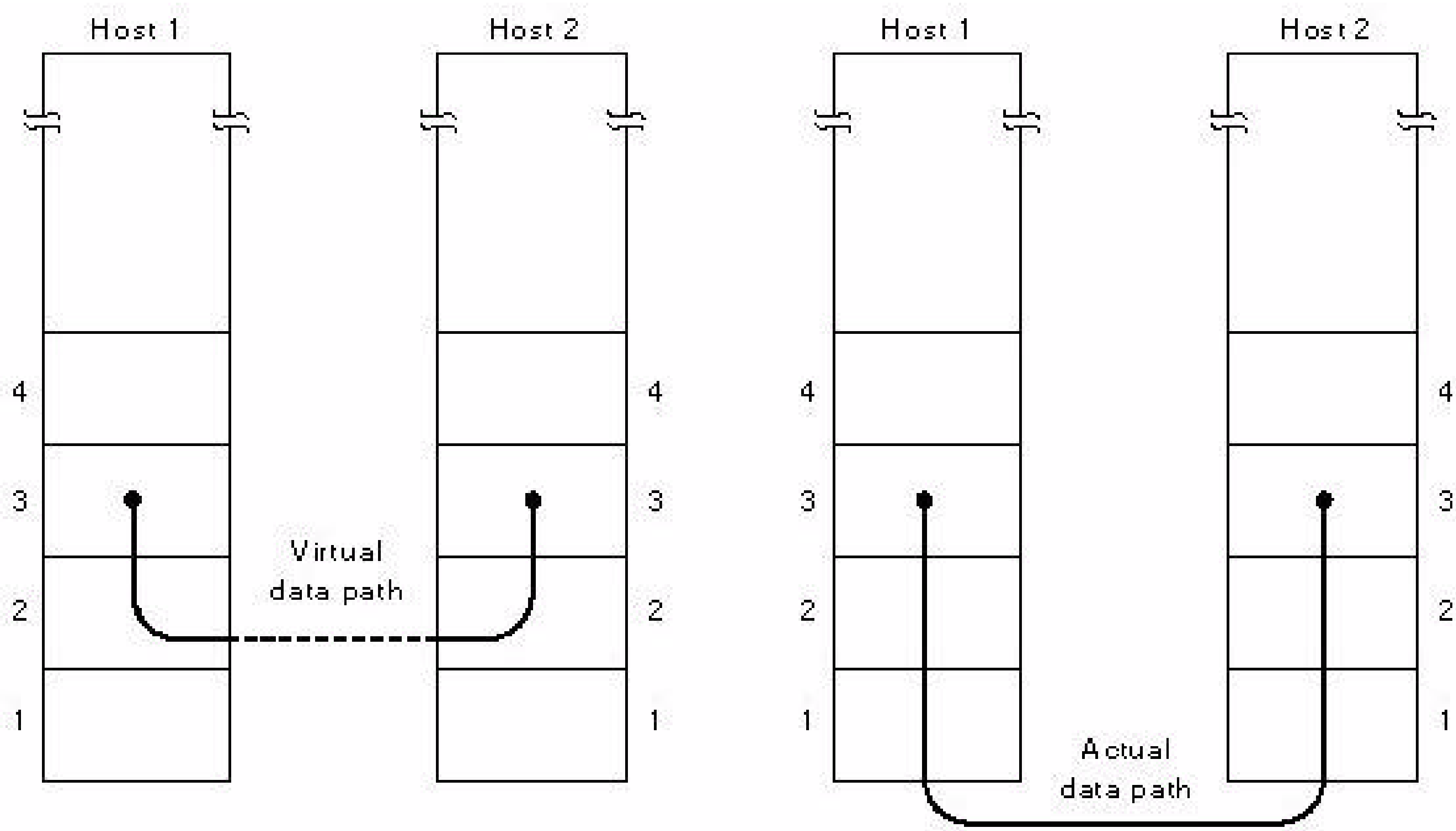
The OSI Reference Model



Data Link Layer: Virtual Vs. Actual Communication



Virtual Communication **Actual Communication**

The Data Link Layer Functions

Concerned with <u>reliable</u>, error-free and efficient communication between *adjacent* machines in the network through the following functions:

1. **Data Framing:**
   The term "frame" refers to a small block of data used in a specific network. The data link layer groups raw data bits to/from the physical layer into discrete frames with error detection/correction code bits added. Framing methods:
   - **Character count.**
   - **Starting and ending characters, with character stuffing.**
   - **Starting and ending flags with bit stuffing.**
   - **Physical layer coding violations.**
2. **Error Detection/Correction:**
   - **Error Detection:**
     - **Include enough redundant information in each frame to allow the receiver to deduce that an error has occurred, but not which error and to request a retransmission.**
     - **Uses error-detecting codes.**
   - **Error Correction:**
     - **Include redundant information in the transmitted frame to enable the receiver not only to deduce that an error has occurred but also correct the error.**
     - **Uses error-correcting codes.**

The Data Link Layer Functions

3 **Services to the network layer:**

- **Unacknowledged connectionless service:**
  - **Independent frames sent without having the destination acknowledge them.**
  - **Suitable for real-time data such as speech and video where transmission speed is more important than absolute reliability.**
  - **Utilized in most LANS.**
- **Acknowledged connectionless service:**
  - **Each frame sent is acknowledged by the receiver.**
  - **Acknowledgment at the layer level is not essential but provides more efficiency than acknowledgment at higher layers (transport) which is done only for the whole message.**
  - **A lost acknowledgment may cause a frame to be sent and received several times.**

The Data Link Layer Functions

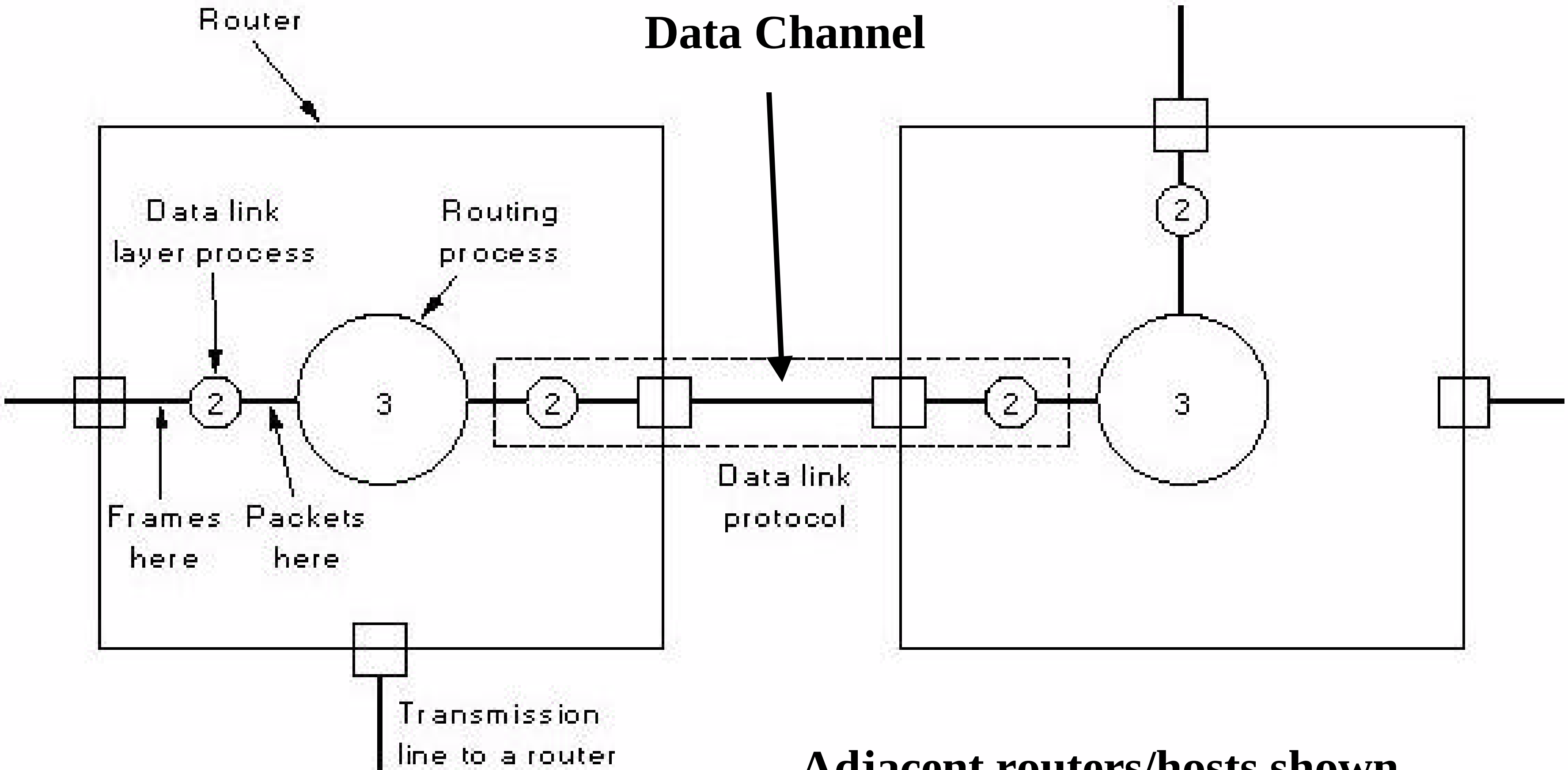
- Acknowledged connection-oriented service:
  - The sender and receiver establish a connection before any data transmission.
  - The message is broken into numbered frames.
  - The data link guarantees that each frame sent is received exactly once and in the right order.

4 Flow control:

Protocols to control the rate the sender transmits frames at a rate acceptable to the receiver, and the ability to retransmit lost or damaged frames. This insures that slow receivers are not swamped by fast senders and further aids error detection/correction.

- Several flow control protocols exist, but all essentially require a form of feedback to make the sender aware of whether the receiver can keep up.
  - Stop-and-wait Protocols:
    - A positive acknowledgment frame is send by the receiver to indicate that the frame has been received and to indicate being ready for the next frame.
    - Positive Acknowledgment with Retransmission (PAR); uses timeouts
  - Sliding Window Protocols:
    - Data frames and acknowledgement frames are mixed in both directions.
    - Frames sent contain sequence numbers
    - Timeouts used to initiate retransmission of lost frames.

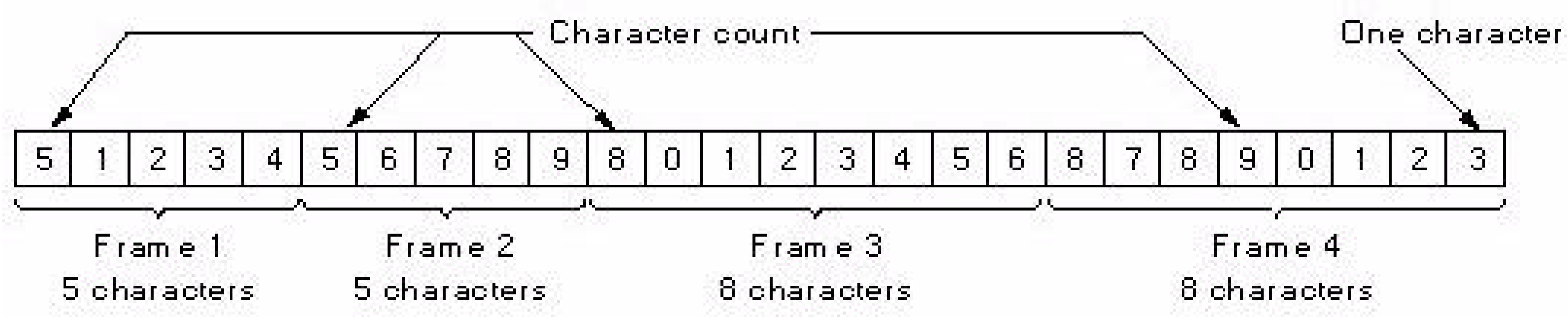
Placement of The Data Link Protocol



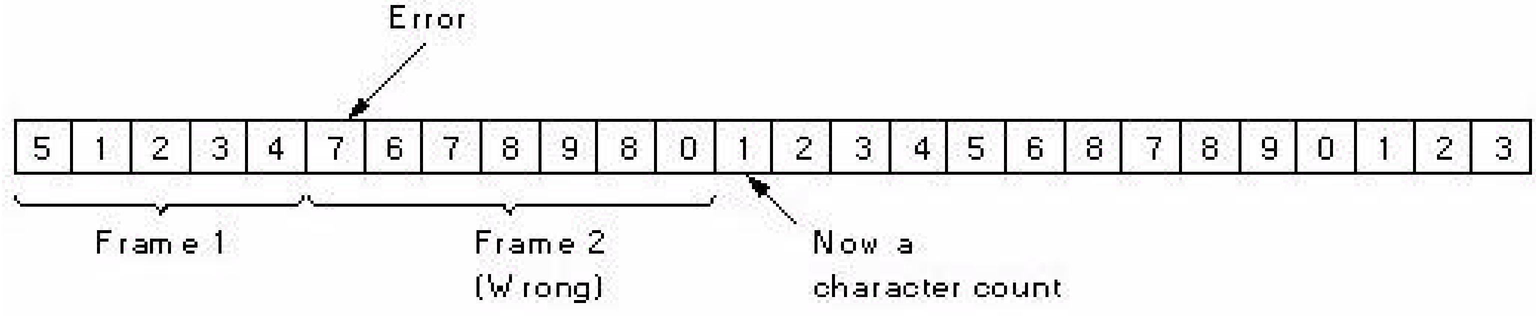
Adjacent routers/hosts shown

Data Link Layer: Framing

- **The character count method:**
  - The frame header includes the count of characters in the frame
  - A transmission error can cause an incorrect count causing the source and destination to get out of synchronization
  - Rarely used in actual data link protocols



A character stream with no errors



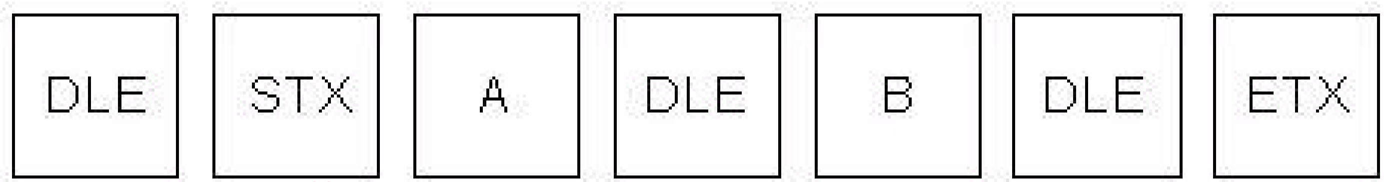
A character stream with one error

Data Link Layer: Framing

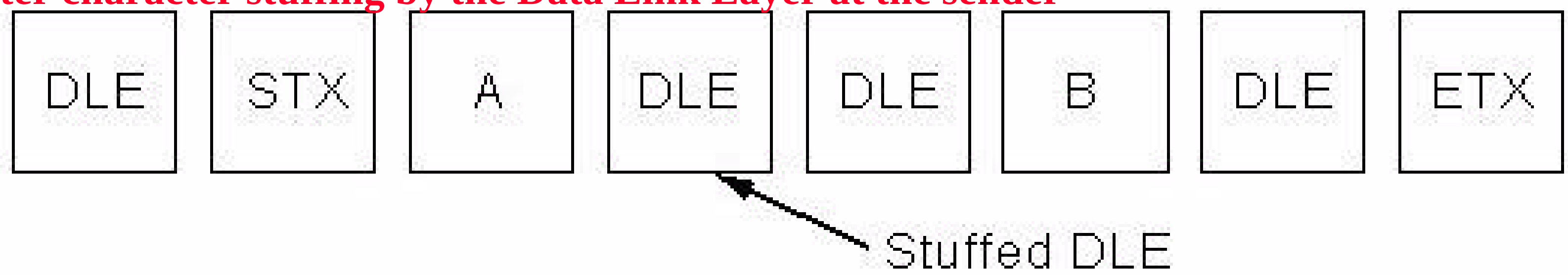
Using Starting and ending characters, with character stuffing

- **Each frame starts with the ASCII character sequence DLE (Data Link Escape) and STX (Start of TeXt) and ends with DLE ETX (End of TeXt)**
- **When binary data is transmitted where (DLE STX or DLE ETX) can occur in data, character stuffing is used (additional DLE is inserted in the data).**
- **Limited to 8-bit characters and ASCII.**

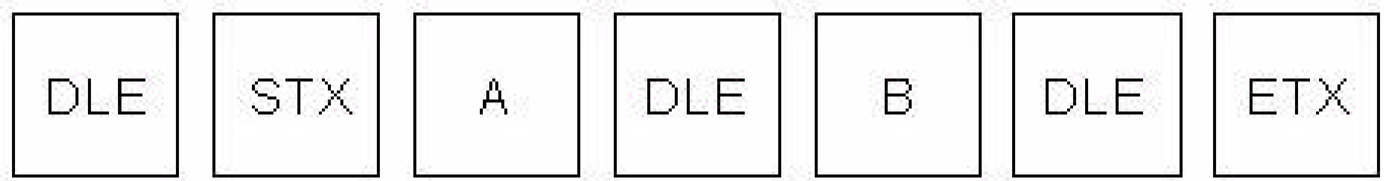
Network Layer Data at the sender



Data after character stuffing by the Data Link Layer at the sender



Network Layer Data at the Receiver



Data Link Layer: Framing

- **Bit-Oriented Using Start/End Flags:**
  - Each frame begins and ends with 01111110
  - Bit stuffing: After each five consecutive ones in a data a zero is stuffed
  - Stuffed zero bits are removed by the data link layer at receiving end.

The Original Data

0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Data appearing on the line after bit stuffing

0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

Data received after destuffing

0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

Data Link Layer: Error Detection/Correction

- **Simplest error detection : Parity bits and checksum (sum of 1's in data).**
- **Error-detecting and -correcting codes:**
  - ***m* data bits + *r* redundant bits added.**
  - **$n = m + r$ transmitted in frame.**
  - **Only 2^m code words out of possible 2^{m+r} words are legal.**
  - **The Hamming distance --minimum number of positions any two legal code words differ-- of a code defines its error detection/correction ability.**
  - **To detect *d* errors code Hamming distance = $d + 1$**
  - **To correct *d* errors code Hamming distance = $2d + 1$**
  - **Some codes are more suitable to correct burst errors rather than isolated errors.**
  - **Polynomial codes: Cyclic Redundancy Check (CRC) Codes, are characterized by a generating polynomial G(X)**

Cyclic Redundancy Check (CRC)

- Based on polynomial arithmetic over finite field.
- View m-bit string $a_{m-1}a_{m-2} \ldots a_0$ as a polynomial of degree m-1:

$$M(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots + a_0$$

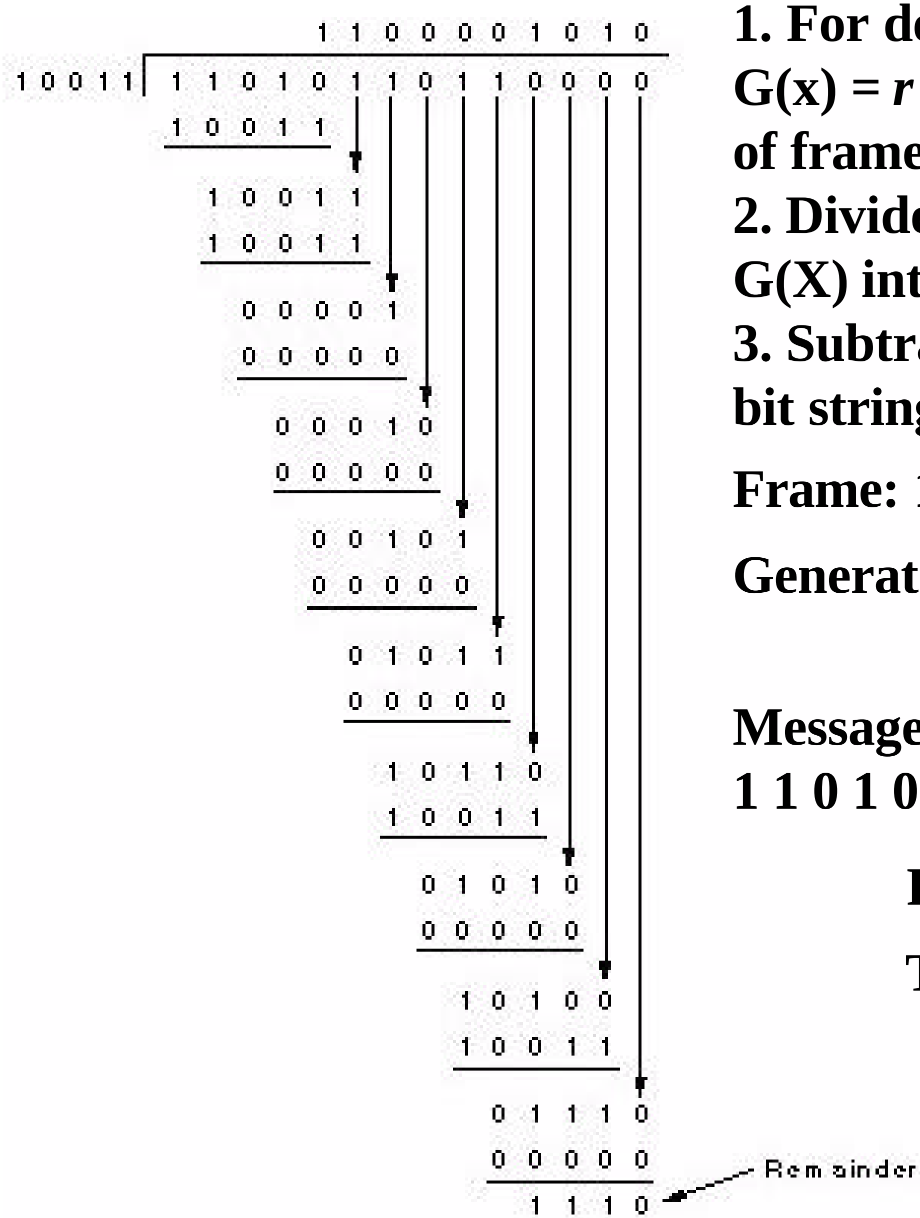
- Select a generating polynomial G(x) of degree r.
- Let R(x) be the remainder of $x^r M(x) / G(x)$
- The code word T(x) of length m + r bit generated is then given by:

$$T(x) = x^r M(x) - R(x)$$

- Assume code word T(x) is transmitted, but T(x) + E(x) arrives at the receiver:
  - If E(x) = 0 then no transmission errors and T(x)/G(x) = 0
  - If $E(x) \neq 0$ then transmission error(s) occurred and:

$$[T(x) + E(x)] / G(x) \neq 0$$

Calculation of Polynomial Code (CRC) Checksum



1. For degree of generating polynomial $G(x) = r$, append r zero bits to low-order of frame. The frame now has $m+r$ bits.

2. Divide the bit string corresponding to G(X) into the bit string $x^r M(x)$ mod(2)

3. Subtract the remainder R(x) from the bit string $x^r M(x)$ mod(2)

Frame: 1 1 0 1 0 1 1 0 1 1

Generator: 1 0 0 1 1

$G(X) = X^4 + X + 1$

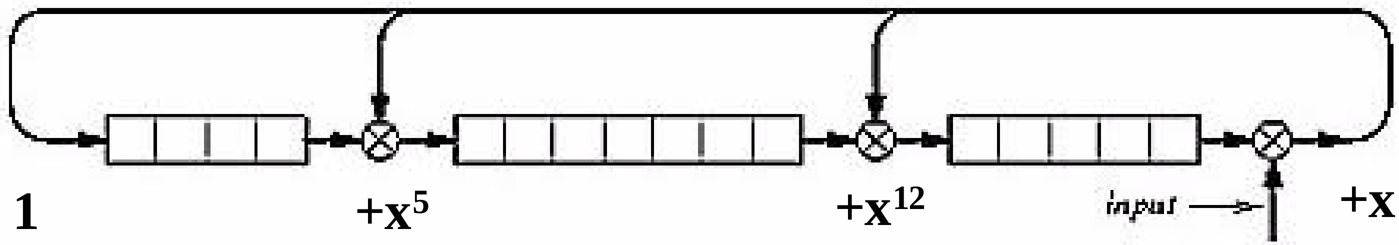
Message after appending four 0's:
1 1 0 1 0 1 1 0 1 1 0 0 0 0

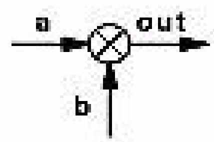
Remainder: 1110

Transmitted Frame:
1 1 0 1 0 1 1 0 1 1 1 1 1 0

Hardware Computation of CRC

For $G(x) = x^{16} + x^{12} + x^5 + 1$





a	b	out
0	0	0
0	1	1
1	0	1
1	1	0

An Example Frame Format with CRC bits

Common CRC Generator Polynomials

- **CRC-32:** $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$

  $\rightarrow$ **Used in FDDI, Ethernet.**

- **CRC-CCITT:** $x^{16} + X^{12} + x^{5} + 1$

  $\rightarrow$ **Used in HDLC.**

- **CRC-8:** $x^{8} + x^{2} + x + 1$

  $\rightarrow$ **Used in ATM.**

Use of A Hamming Code to Correct Burst Errors

Char.	ASCII	Check bits
H	1001000	00110010000
a	1100001	10111001001
m	1101101	11101010101
m	1101101	11101010101
i	1101001	01101011001
n	1101110	01101010110
g	1100111	11111001111
	0100000	10011000000
c	1100011	11111000011
o	1101111	00101011111
d	1100100	11111001100
e	1100101	00111000101

Order of bit transmission