ECE453 – Introduction to Computer Networks

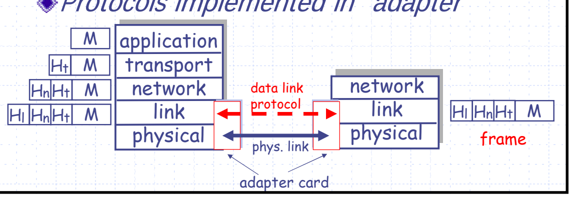
Lecture 4 – Data Link Layer (I)

Design Issues

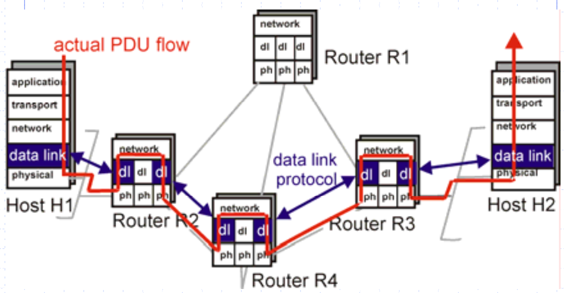
- Provide a well-defined service interface
- Group bits (PHY) into frames (DL)
- Deal with transmission errors
- Regulate the flow of frames

Link Layer: Setting the Context

- two *physically connected* devices:
  - host-router, router-router, host-host
- unit of data: *frame*
- *Protocols implemented in "adapter"*

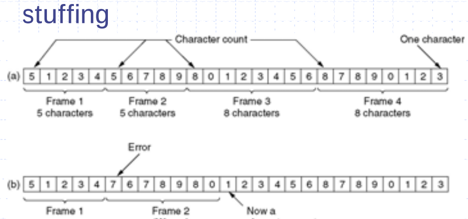


Link Layer – The Logical & Physical Flows

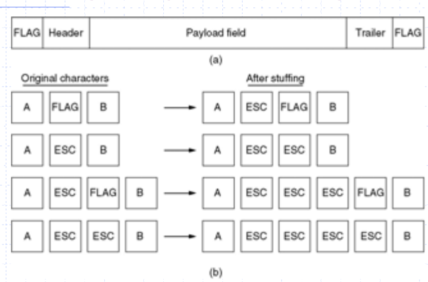


Framing

- Character count
- Flag bytes with byte stuffing
- Starting and ending flags with bit stuffing



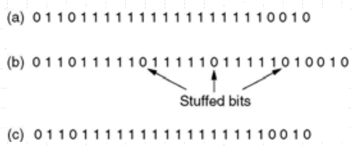
Framing – Byte Stuffing



Tightly tied to the used of 8-bit characters

Framing – Bit Stuffing

- Flag pattern: 01111110
- Sender stuff: add 0 after 11111
- Receiver destuff: delete 0 after 11111



Error Detection and Correction1

- Use redundancy
  - Error detection and retransmission
    - Over low error rate media
  - Error correction
    - Over high error rate media

The Hamming Distance

- Codewords (n bits): message (m bits) + check bits (r bits)
- Complete list of codewords?
- Hamming distance: the number of positions in which two codewords differ
- Hamming distance of the complete code: two codewords with the minimum Hamming distance
- To detect *d* errors, Hamming distance of the code: (d+1)
- To correct *d* errors, Hamming distance of the code: (2d+1)

Parity Bit

- Even parity
- Odd parity

- A code with a single parity bit has a Hamming distance of ???
- It can be used to detect ??? errors

Example

0000000000
0000011111
1111100000
1111111111

The code has a Hamming distance of ???
It can detect ??? errors
It can correct ??? errors

Arrival	Original
0000000111	0000011111
0000000111	0000000000

The Hamming Code

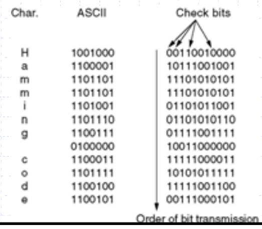
- Theorem: Given a code with *m* message bits and *r* check bits ($n=m+r$) which allows all single errors to be corrected, the lower limit on *r* is $(m+r+1)<=2^r$
  - **Hamming code (the bits that are power of 2 are check bits, others are message bits, each check bit forces the parity of some collection of bits, including itself)**
  - **Can only correct single bit error**

Example - Hamming Code

10010000

How to Correct Burst Errors

- Uses *kr* check bits to make blocks of *km* data bits immune to a single burst error of length *k* or less



Error Detecting Code - I

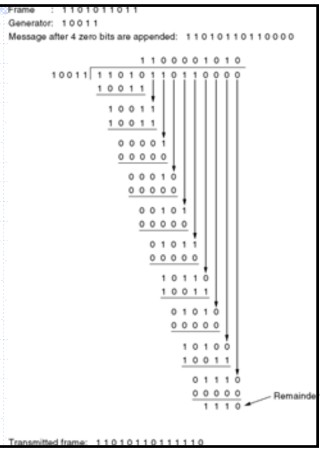
```
x x x x x x x | o
x x x x x x x | o
x x x x x x x | o
x x x x x x x | o
x x x x x x x | o
x x x x x x x | o
x x x x x x x | o
--------------+--
o o o o o o o | o
```

Error-Detecting Codes

- Polynomial code (CRC – Cyclic Redundancy Check)
  - Generator polynomial G(x)
  - Message polynomial M(x)
  - Method: append a checksum of *r* bits to the end of M(x) such that the appended polynomial T(x) is divisible by G(x)

Q(x) ... R(x) = $x^rM(x)/G(x)$

T(x) = $x^rM(x)$ + R(x)



More on Polynomial Code

- Single error detection?
- Double error detection?
- No polynomial with an odd number of terms is divisible by x+1
- A polynomial code with r check bits will detect all burst errors of length <=r
  - Burst error: at least the first and the last bits of a bit stream are wrong
- Hardware implementation: shifted register circuit
- International standard of G(x)
  - $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x^{1}+1$