Week 3 – Data Link Layer

COMP nologies

Lecturer: Ling Luo

Semester 2, 2021

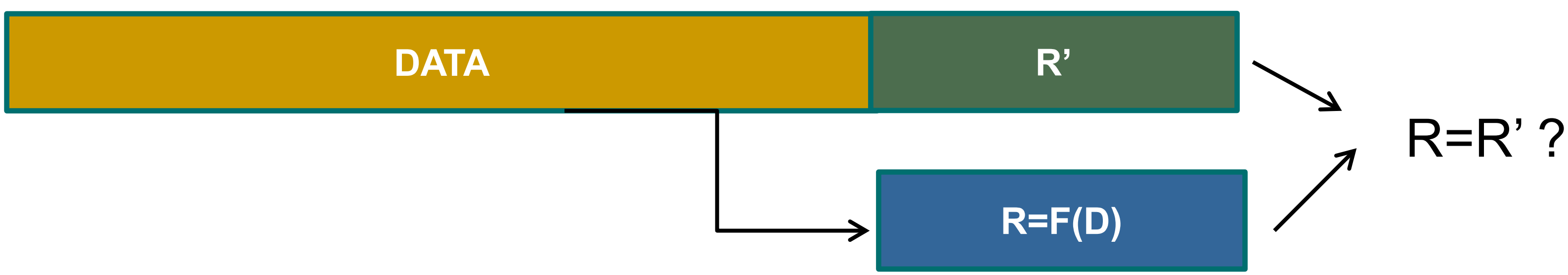
Error Detection Codes

- **Parity Bit** (1 bit): (Hamming distance=2)
- **Internet Checksum** (16 bits): (Hamming distance=2)
- **Cyclic Redundancy Check** (CRC) (Standard 32-bit CRC:)

How it works?

- Sender: calculates R check bits using a function of data bits:

- Receiver: rec and calculates the same function on the match the results with received chec



Parity Bit

Given data 10001110, count the number of 1s

Sender: Add parity bit → 10001110**0** (for even parity)

100011101 (for odd parity)

Receiver: Check t

Hamming distance is 2 for Parity Bit

2 – 1 = **1 error bit can be detected** and

(2 – 1) / 2 = ½ not even 1 bit error can be corrected

Internet Checksum

- There are different variations of checksum
- Internet Checksum (16-bit word):

  Sum modulo 2^{16} and add any overflow of high order b der bits

Example of Checksum

Calculate checksum (5-bit word) for data

00110 10001 11001 01011

1
00110
+10001 = $\boxed{10111}$

2
11001
+10111 = 10000 +1 = $\boxed{10001}$

3
+10001 = $\boxed{11100}$

4
The checksum is *one's complement* of 11100 which is $\boxed{00011}$

Data sent: 00110 10001 11001 01011 **00011**

Cyclic Redundancy Check

- Based on a generator polynomial G(x)
  - e.g. $G(x) = x^4+x+1$ (10011)
  - Steps:
    - Let *r* be the **d *r* zero bits to the low-order end** ins *m* + *r* bits and corresponds to the polynomi
    - **Divide the bit string corres** **(x)** into the bit string corresponding to $x^rM(x)$, using modulo 2 division.
    - **Subtract the remainder** (which is always *r* or fewer bits) from the bit string corresponding to $x^rM(x)$ using modulo 2 subtraction.
    - The result is the checksummed frame to be transmitted. Call its polynomial $T(x)$.

Example

Data: **1101001** and G(x) = x^4+x+1 (**10011**)

5 bits polynomial add **4** bits as the checksum – so add 0000

```
10011 | 11010010000
        10011
        01001
         1001
         00001100
            10011
            010110
             10011
             00101
```


Data sent:
11010010101

Error Correction: Hamming Code

- $n=2^k-k-1$ (n: number of data, k: check bits)

  Example: Data: 0101 - > requires 3 check bits

  $$4 = (2^3) - 3 - 1$$

- Put check bits _____________________ **<u>power of 2</u>**, starting with position 1

- Check bit in **<u>position *p* is parity of positions with a *p* term in their value</u>**

Example

Put check bits in positions p that are power of 2, starting with position 1

- Data: 0101 → requires 3 check bits

111

Position	P1	P2	P3	P4	P5	P6	P7
Data	?	?	0	?	1	0	1

1. Calculate the parity bits for P1, P2, P4 (rule: even parity)

P1 + P3 + P … P1 = 0

P2 + P3 + P … 2 = 1

P4 + P5 + P6 + P7 = ?+1+0+1 = 0

Data sent: 0100101

error | error

Example 1: At the receiver: 0100100

P1 + P3 + P5 + P7 = 0+0+1+0= 1 ×

P2 + P3 + P6 + P7 = 1+0+0+0= 1 ×

P4 + P5 + P6 + P7 = 0+1+0+0= 1 ×

Error bit: P1, P2, P4 → P(1+2+4)=P7

Example 2: At the receiver: 0000101

P1 + P3 + P5 + P7 = 0+0+1+1= 0

P2 + P3 + P6 + P7 = 0+0+0+1= 1 ×

P4 + P5 + P6 + P7 = 0+1+0+1= 0

Error bit: P2

Error Control Discussion

- Error Correction: More efficient in noisy transmission media e.g., wireless
- Error Detection: More efficient in the transmission media where low error rates occur, e.g. quality copper

- The error can
- Require assum

ber of errors occurring in transmission