ECE 463
Introduction to Computer Networks

Lecture: Error Detection

Sanjay Rao

Error Detection and Correction

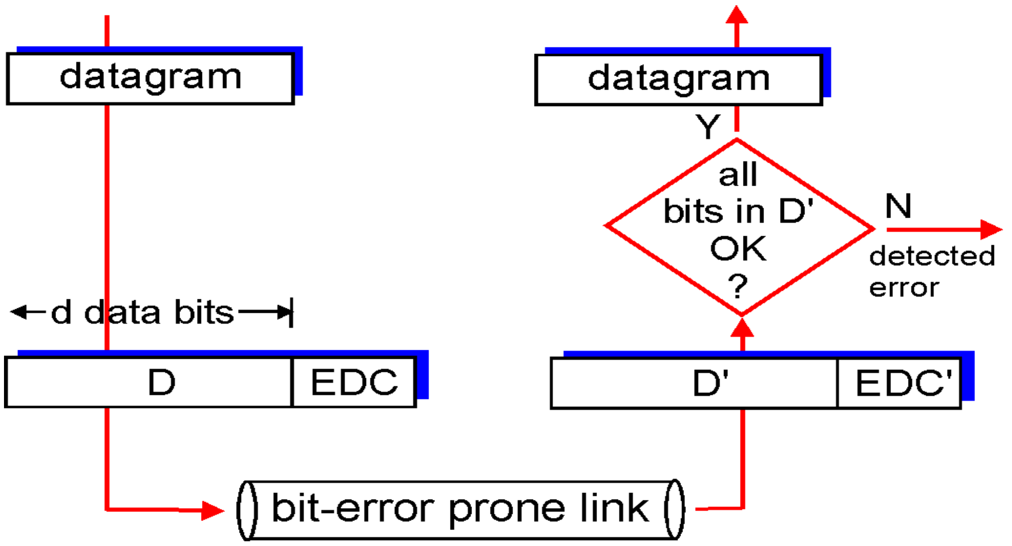
- Errors occur due to noise or interference on a communication channel
- Bit Error Rate (BER):
  – e.g. 10^{-12} is one error in 10^{12} bits (on average)
- Error detection and correction
  – How do we detect errors?
  – Can we correct the errors we have detected?
  – What is the cost of error detection/correction?

Redundancy

- To detect/correct errors, we need redundancy
- Issues: How much redundancy ? What is the cost?
- Correction Vs. Detection
  - Correction involves additional redundancy
  - may be useful for real-time communication or for long, fat pipes or for very noisy channels.

Error Detection

- EDC = Error Detection and Correction bits (redundancy)
- D = Data protected by error checking, may include header fields
- Error detection not 100% reliable!
  - protocol may miss some errors, but rarely
  - larger EDC field yields better detection and correction: trade-off



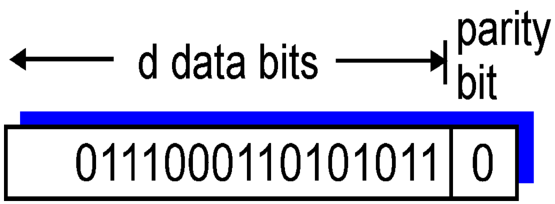
Common Approaches

- Parity bits
- Cyclic redundancy check (CRC)
- Internet checksum

Parity Checking

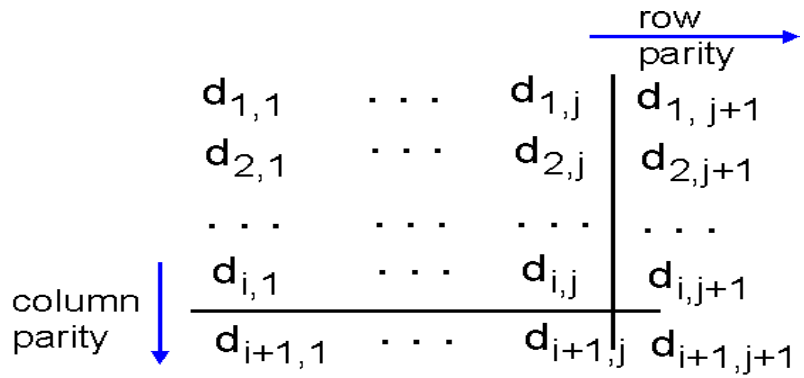
Single Bit Parity:

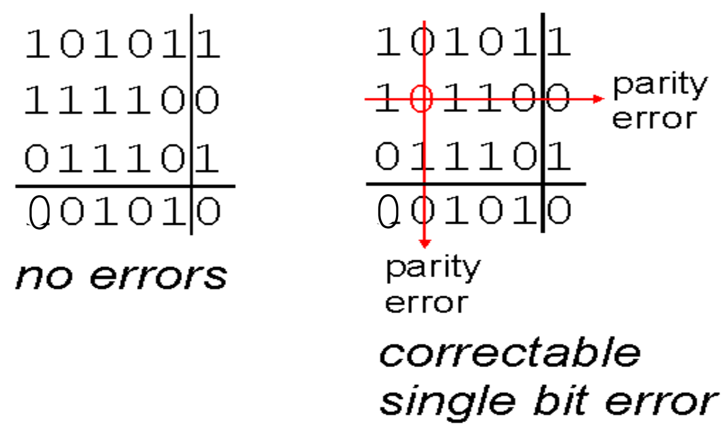
Detect single bit errors



Two Dimensional Bit Parity:

Detect *and correct* single bit errors, *detect also 2 and 3 errors*





Still too limited!
Especially since errors come in burst.

Exercise

- Can a 2-D parity scheme be used to correct 2-bit errors? Why/why not?

- Can a 2-D parity scheme be used to detect all 4-bit errors? Why/why not?

Polynomial codes

- Polynomial codes: Cyclic Redundancy Check (CRC)
- Principle: Consider n bit message as corresponding to an (n-1) degree polynomial with the message bits as coefficients
- Example:
  - m = 10011010

$$M(x) = x^7 + x^4 + x^3 + x^1$$

Polynomial codes: the principle

- Both ends agree on a **generator polynomial (*G*) of degree g.**
- The sender creates the polynomial $x^g M(x)$ by appending g zeros at the right of M.
- The sender computes the remainder $R(x)$ of the division of $x^g M(x)$ by $G(x)$.
- The sender sends $S(x) = x^g M(x) - R(x)$ which is divisible by $G(x)$.
- The receiver receives $S_r(x)$. It divides it by $G(x)$. It the remainder is zero, it believes there is no error

Polynomial codes: arithmetic

- Any polynomial B(x) can be divided by a divisor polynomial C(x) if B(x) is of higher degree than C(x), or same degree as C(x)
- Like normal division, except, all operations are done modulo 2. A subtraction is equivalent to an XOR.
- Example:

```
   1100  (-)
   1010
-----------
    110
-----------
```

Polynomial codes

M	11011100	message, degree 7, 8 bits
G	1100	generator polynomial, degree 3, 4 bits, $x^3 + x^2$
x^3M	11011100**000**	extended message, degree 10, 11 bits

Sender: divide x^3M by G:

```
         10010111
    ______________
1100|11011100000
     1100
     ----
        1110
        1100
        ----
          1000
          1100
          ----
           1000
           1100
           ----
            1000
            1100
            ----
             100
```

R **100** degree 2, 3 bits

x^3M	11011100**000**	
R	**100**	–
S	11011100**100**	

Receiver: divide S_r by G:

```
         10010111
    ______________
1100|11011100100
     1100
     ----
        1110
        1100
        ----
          1001
          1100
          ----
           1010
           1100
           ----
            1100
            1100
            ----
               0
```

Choosing G(x)

- G(x) is standardized to be small but typically produce remainders. Detects:
  - all single bit errors
  - all double-bit errors if G(x) has a factor with at least 3 terms
  - any odd number of errors, if (x+1) divides G(x)
  - any burst error of length < length of CRC
  - most large burst errors

Lots of other properties possible …

CRC-16 $G(x) = x16 + x15 + x2 + 1$
CRC-CCITT $G(x) = x16 + x12 + x5 + 1$

Both give 16-bit checksums which will detect:

- all 1 and 2 bit errors
- all error bursts of up to 16 bits in length
- all bursts affecting an odd number of bits
- 99.997% of 17 bit error bursts
- 99.998% of 18 and longer bursts

Standard CRC Polynomials

CRC-8: 100000111

CRC-10: 11000110011

CRC-12: 1100000001111

CRC-16: 11000000000000101

CRC-CCITT: 10001000000100001

CRC-32: 100000100110000010001110110110111

The Internet Checksum

- Used in IP, ICMP, TCP, UDP, … (hence at higher layers)
- Use a much simpler technique based on binary sums
- Easy to compute and check in software
- Not as strong as CRC

Exercise

- Consider the message 10011010, with the generator polynomial being 1101. What message must be transmitted with CRC?