Chapter 5
Error Detection

Types of Errors

- A network system that cannot guarantee that the data received are completely identical to the data transmitted is essentially useless.
- Reliable systems must have a mechanism for detecting and correcting the errors.
- There are two types of errors:
  - **Single-bit error**: one bit is changed from 0 to 1 or from 1 to 0
  - **Burst error**: multiple bits are changed

Figure 5-1

Types of Errors

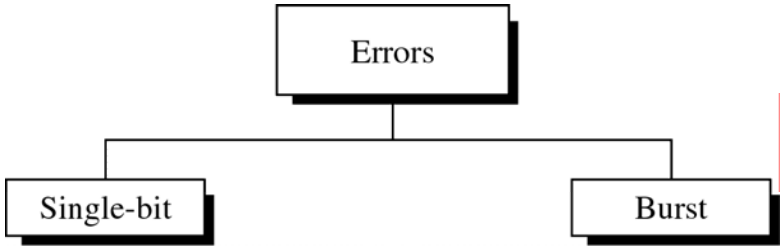
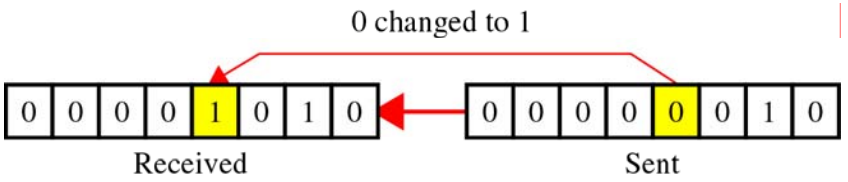


Figure 5-2

Single-Bit Error



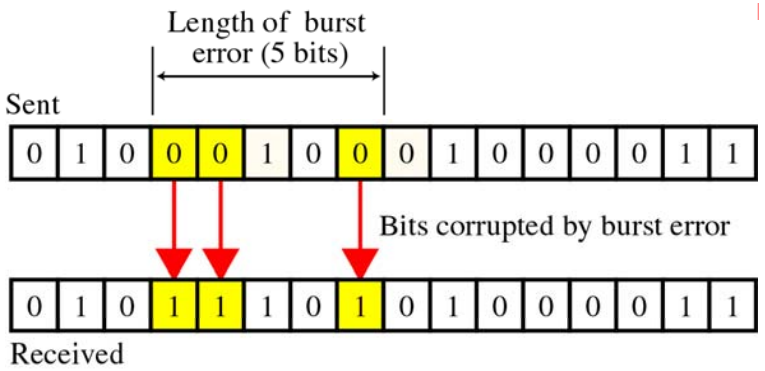
ASCII LF **ASCII STX**

Single-Bit Error

- Single-bit errors are least likely type of error in serial transmission – the duration of the noise is normally much longer than that of a bit.
- However single-bit error can happen in parallel transmission, such as data transfer inside a computer, between CPU and memory, or between a computer and a printer.
  - E.g. one noisy line out of eight in a parallel printer cable can corrupt one bit in each byte.

Figure 5-3

Burst Error of Length Five



Burst Error

- Burst errors is most likely to happen in a serial transmission.
- Burst errors does not mean errors occur in consecutive bits.
- The number of bits affected depends on the data rate and duration of noise.
- E.g. if date rate is 1 Mbps and noise duration is 1/100 s, the number of bits affected is

  1 Mbps × 1/100 s = 10,000 bits

Error Detection

- To detect an error, the **redundancy** technique is used, where a short group of bits are appended to the end of each data unit to be sent.
- The extra redundant bits carry information for error detection for the data unit.
- If the received data stream pass the checking function, the redundant bits are discarded.
- Three types of redundancy checks are used: **vertical redundancy check (VRC), longitudinal redundancy check (LRC),** and **cyclic redundancy check (CRC)**.

Figure 5-4

Redundancy

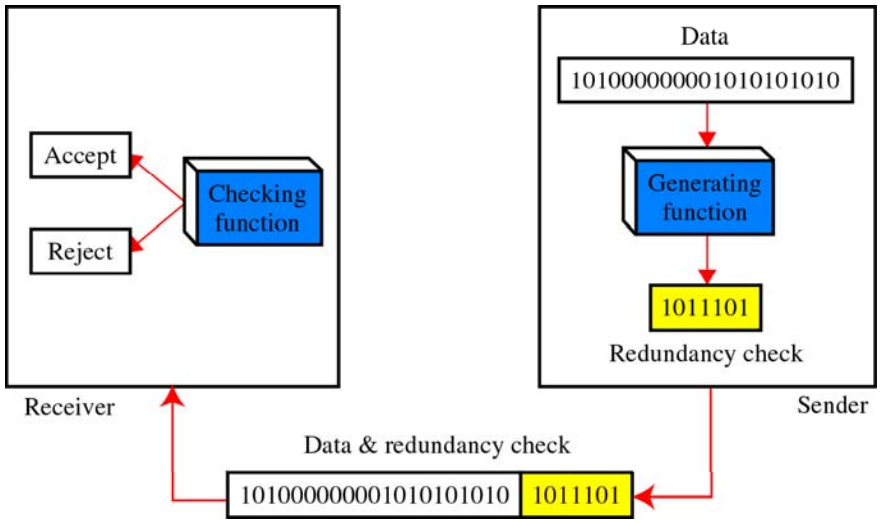
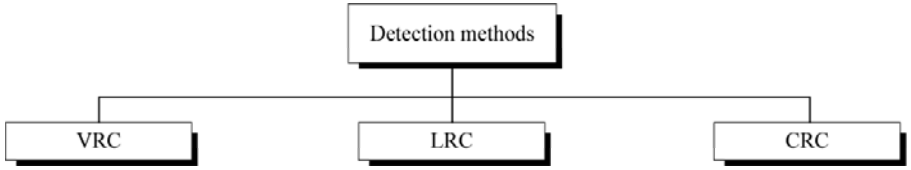


Figure 5-5

Detection Methods

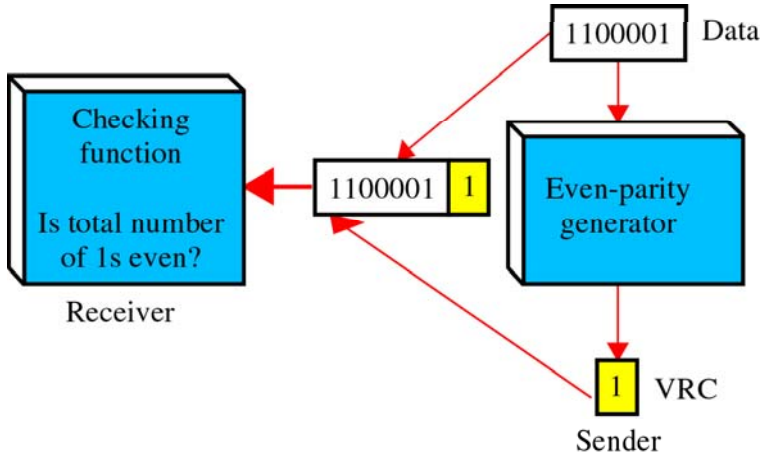


Vertical Redundancy Check (VRC)

- Most common and least expensive mechanism for error detection, often called **parity check**
- A parity bit is appended to every data unit so that the total number of 1s in the unit, including the parity bit, is even (or odd).
- Can detect all single-bit errors and burst errors that change odd (or even) number of bits.
- Cannot detect errors that change even (or odd) number of bits.

Figure 5-6

Even Parity VRC Concept

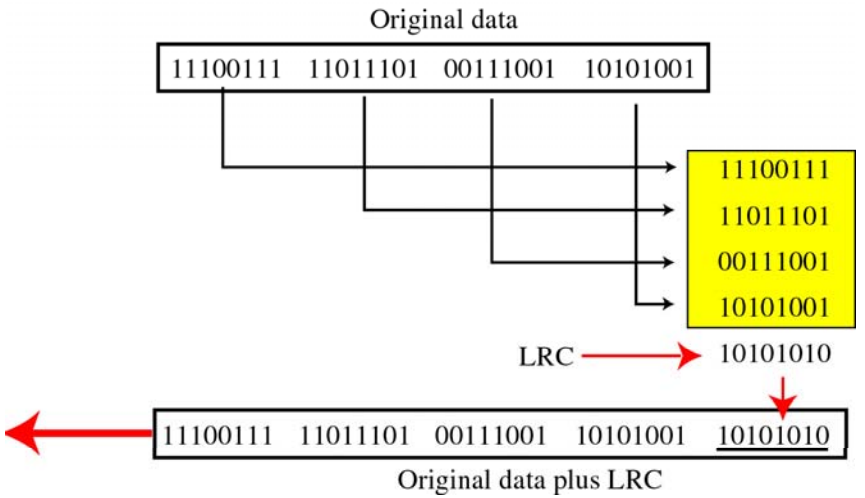


Longitudinal Redundancy Check (LRC)

- A block of data is organized in a table of rows and columns. The parity bit for each column is then calculated. All parity bits are grouped and appended to the original data to be sent (Fig. 5-7).
- LRC increases the likelihood of detecting burst errors – an LRC of n bits can detect a burst error of n bits.
- Cannot detect errors where there are even number of damaged bits that are located in the same column position of different rows.

Figure 5-7

LRC



An LRC Example

- Suppose the following block is sent:

  <=== 10101001 00111001 11011101 11100111 10101010 (LRC)

- However it is hit by a burst noise of length eight

  <=== 10100011 10001001 11011101 11100111 10101010 (LRC)

- The receiver checks the LRC and finds some bits do not follow even-parity rule, so the whole block is discarded.

  <=== 10100011 10001001 11011101 11100111 10101010 (LRC)

Cyclic Redundancy Check (CRC)

- The most powerful redundancy checking technique
- A sequence of redundant bits, called CRC, is appended to the end of a data unit so that the resulting data unit is exactly divisible by a predetermined binary number.
- At the destination, the data unit is divided by the same binary number. If remainder is 0, the data is assumed to be intact and accepted. If remainder is not 0, the data is considered damaged and rejected.

CRC Generation

- CRC bits are derived by dividing the data unit by the predetermined divisor – the remainder is the CRC.
- CRC is always one bit less than the divisor.
- CRC is found as follows:
  1) First a string of *n* 0s is appended to the data unit (assume divisor has *n*+1 bits)
  2) Second the new data unit is divided by the divisor using binary division. The remainder is the CRC.
  3) Third the *n* CRC bits replace the appended 0s.

Figure 5-8

CRC Generator and Checker

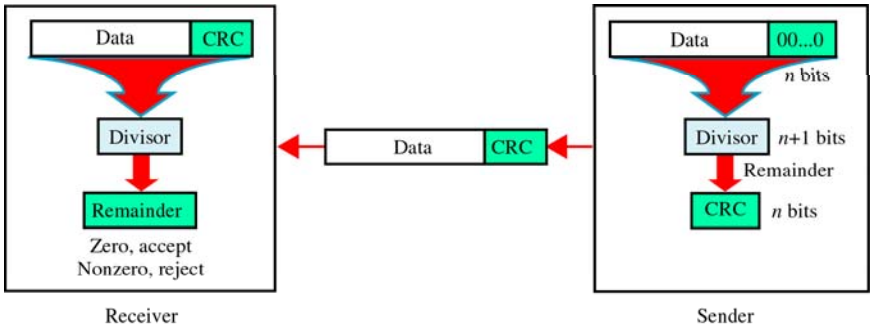


Figure 5-9

Binary Division in a CRC Generator

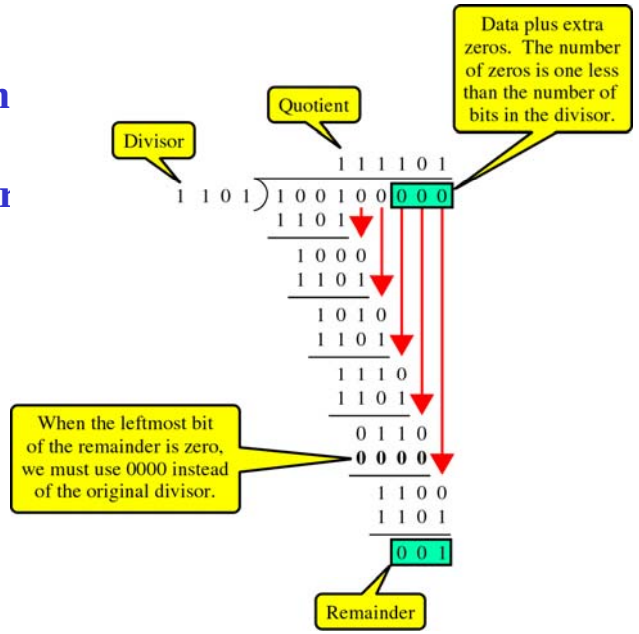
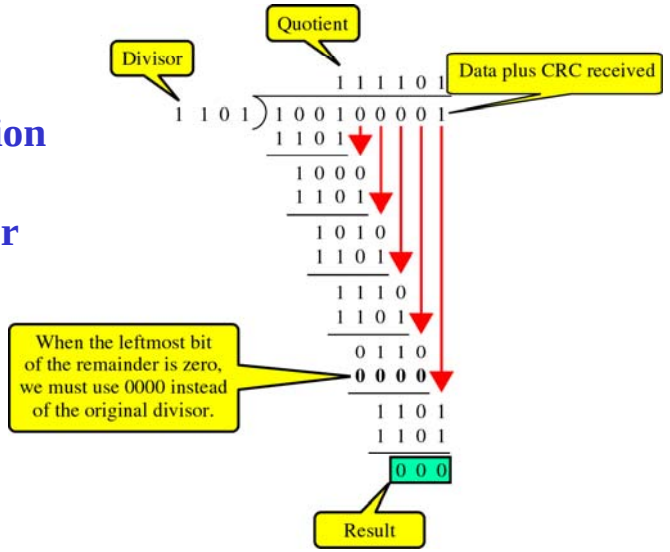


Figure 5-10

Binary Division in a CRC Checker



Binary Division in a CRC Generator

- **Suppose we want to transmit the information string: 1111101.**
- **The receiver and sender decide to use the (arbitrary) polynomial pattern, 1101.**
- **The information string is shifted left by one position less than the number of positions in the divisor.**
- **The remainder is found through modulo 2 division (at right) and added to the information string: 1111101000 + 111 = 1111101111.**

```
          1011011
    ------------
1101)1111101000
     1101
     ----
     001010
        1101
        ----
        01111
         1101
         ----
         001000
            1101
            ----
            01010
             1101
             ----
             0111
```


Binary Division in a CRC Checker

If no bits are lost or corrupted, dividing the received information string by the agreed upon pattern will give a remainder of zero.

We see this is so in the calculation at the right.

Real applications use longer binary numbers to cover larger information strings.

```
          1011011
    ------------
1101)1111101111
     1101
     ----
     001010
        1101
        ----
        01111
         1101
         ----
         001011
            1101
            ----
            01101
             1101
             ----
             0000
```


Polynomials

- The CRC divisor is most often represented as an algebraic **polynomial**, rather than a binary number, for purpose of mathematical proof.
- The degree of a polynomial is its highest power.
- Any polynomial selected must have the following properties:
  - It should not be divisible by x
  - It should be divisible by x+1
- Several common polynomials are used (see Fig. 5-13)

Figure 5-11

A Polynomial of Degree Seven

$$x^7 + x^6 + x^4 + x^3 + x + 1$$

Figure 5-12

A Polynomial Representing a Divisor

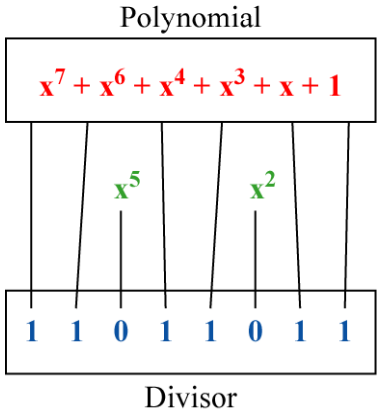


Figure 5-13

Standard Polynomials

CRC-12

$x^{12} + x^{11} + x^3 + x^2 + x + 1$

CRC-16

$x^{16} + x^{15} + x^2 + 1$

CRC- ITU-T

$x^{16} + x^{12} + x^5 + 1$

CRC-32

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

CRC-32 is used in both Ethernet and Token Ring network.

Performance

- CRC can detect all burst errors affecting an odd number of bits.
- CRC can detect all burst errors of length less than or equal to the degree of the polynomial
- It can detect with a very high probability burst errors of length greater than the degree of the polynomial.
- E.g. CRC-12 will detect all burst errors affecting an odd number of bits, will detect all burst errors with length ≤ 12, and will detect 99.97% of the burst errors with a length ≥ 12.

Error Correction

- Error correction can be handled in two ways:
  - When an error is detected, the sender is notified and the entire data unit is retransmitted
  - The receiver uses an error-correcting code to correct the errors automatically
- Error correction by retransmission is practical and common method in networking and communication.
- Error correcting codes are more complicated than error detecting codes. Hamming code is a popular single-bit error correction method.

Summary

- Errors can be categorized as a single bit or burst.
- Three types of redundancy checks are used in LANs: VRC, LRC and CRC.
- In VRC, a parity bit is added to the data unit.
- In LRC, data unit is organized into a table, and a parity bit is calculated from each column.
- CRC, the most powerful of the three, is based on binary division.
- In CRC, the generator, using a specific divisor, creates redundant bits that are appended to the data. The checker uses the same divisor to verify the data.