Error Detection and Correction in the Data Link Layer

Department of Computer Science and Engineering

Contents

1 Introduction

Data transmission over physical communication channels is inherently unreliable due to noise, interference, attenuation, and hardware imperfections. As a result, bits may flip during transmission, leading to corrupted frames.

The **Data Link Layer (DLL)** is responsible for detecting and, in some cases, correcting such errors to ensure reliable node-to-node communication. This tutorial presents a comprehensive study of classical and modern error detection and correction techniques used at the data link layer.

2 Types of Errors in Data Transmission

2.1 Single-Bit Error

A single-bit error occurs when exactly one bit in a data unit changes value during transmission.

$$101101 \rightarrow 101001$$

Single-bit errors are common in low-noise wired channels.

2.2 Multiple-Bit Error

Multiple bits may flip independently within a frame.

$$101101 \rightarrow 100001$$

2.3 Burst Error

A burst error affects multiple consecutive bits.

$$110011001100 \rightarrow 110000111100$$

Burst errors are dominant in wireless and optical links.

3 Single Parity Check

3.1 Concept

A single parity bit is appended to the data to ensure that the total number of 1s is either even (even parity) or odd (odd parity).

3.2 Numerical Example

Data:

$$1010110$$

Number of 1s = 4 (even) Parity bit (even parity) = 0
Transmitted frame:

$$10101100$$

3.3 Detection Capability

- Detects all odd-numbered bit errors - Fails for even-numbered bit errors

3.4 Limitations

- Cannot detect burst errors reliably
- No error correction capability

3.5 Industry Usage

Rarely used today; mainly educational or legacy hardware.

4 Two-Dimensional (2D) Parity

4.1 Concept

Data is arranged in a matrix. Parity bits are added:

- Horizontally (row parity)
- Vertically (column parity)

4.2 Numerical Example

Assume even parity.

$$\begin{matrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{matrix}$$

Row parity added:

$$\begin{matrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{matrix}$$

Column parity row:

$$0 \quad 0 \quad 0 \quad 0 \quad 0$$

4.3 Capabilities

- Detects all single-bit errors
- Corrects single-bit errors
- Detects many multi-bit and burst errors

4.4 Limitations

Fails when:

- Two-bit errors occur in same row or column

4.5 Industry Usage

Used in:

- Early memory systems
- RAID parity concepts

5 Checksum

5.1 Concept

Checksum is based on arithmetic addition of data words using one's complement arithmetic.

5.2 Numerical Example

Assume 8-bit words:

$$01010101, \quad 00110011, \quad 00001111$$

Sum:

$$10010111$$

Checksum:

$$01101000$$

Receiver adds all words including checksum; result must be all 1s.

5.3 Detection Capability

- Detects all single-bit errors
- Detects many multi-bit errors
- Weaker than CRC

5.4 Limitations

- Fails for some structured error patterns
- Poor burst error detection

5.5 Industry Usage

- TCP
- UDP
- IP header

Chosen for software efficiency.

6 Cyclic Redundancy Check (CRC)

6.1 Concept

CRC treats data as a polynomial over GF(2). The sender appends remainder obtained by polynomial division.

6.2 Numerical Example

Data: 110101 Generator: 1011

Append zeros:

$$110101000$$

Perform modulo-2 division $\rightarrow$ remainder:

$$011$$

Transmitted frame:

$$110101011$$

6.3 Detection Capability

CRC detects:

- All single-bit errors
- All double-bit errors
- All odd number of errors (if generator has factor $x + 1$)
- All burst errors shorter than generator degree

6.4 Why CRC is Strong

CRC creates a large minimum Hamming distance implicitly.

6.5 Industry Usage

- Ethernet (CRC-32)
- Wi-Fi
- USB
- Storage systems

CRC is the **de facto standard** in the data link layer.

7 Hamming Code

7.1 Concept

Hamming codes add parity bits at positions that are powers of two.

7.2 Hamming Distance

$$d_{\min} = 3$$

7.3 Capabilities

- Corrects 1-bit error
- Detects 2-bit errors

7.4 Numerical Example

For data 1011, parity bits are computed and inserted to form a (7,4) code.

7.5 Limitations

- Inefficient for large blocks
- Not suitable for burst errors

7.6 Industry Usage

- ECC memory
- Space communication

Rarely used in data link layer.

8 Hamming Distance

8.1 Definition

Hamming distance is the number of bit positions in which two codewords differ.

8.2 Error Detection Rule

$$\text{Detectable errors} = d_{\min} - 1$$

8.3 Error Correction Rule

$$\text{Correctable errors} = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor$$

8.4 Key Insight

Hamming distance is a **design-time property**, not transmitted at runtime.

9 Comparative Summary

10 Industry Perspective

Modern networks favor:

- **CRC at Data Link Layer**

Technique	Detect	Correct	Burst Errors	Used in DLL
Single Parity	Weak	No	No	No
2D Parity	Medium	1-bit	Some	Rare
Checksum	Medium	No	Weak	No
CRC	Very Strong	No	Excellent	Yes
Hamming Code	Strong	1-bit	Weak	Rare

- **Retransmission (ARQ)** instead of correction

Error correction is used when:

- Retransmission is expensive (space, storage)

11 Conclusion

Error detection and correction techniques represent a tradeoff between redundancy, complexity, and reliability. While simple parity and checksum illustrate foundational concepts, CRC dominates modern data link layer implementations due to its superior detection capability and hardware efficiency. Hamming distance provides the theoretical framework that unifies all error control schemes.

Final Insight: *Error control is not about eliminating errors, but about managing uncertainty efficiently.*