Error Detection Methods in Data Communication (With Sender & Receiver Side Examples)

Error Detection Methods in Data Communication

Error Detection Methods in Data Communication (With Sender & Receiver Side Examples)

1. Parity Checking

Sender Side:

1. **Even Parity**: Adds a bit to make the total number of 1s even.

   - Data: `1101011` (5 ones → odd) → Parity bit = `1` → Sent: `11010111`

2. **Odd Parity**: Adds a bit to make the total number of 1s odd.

   - Data: `1101011` (5 ones → odd) → Parity bit = `0` → Sent: `11010110`

Receiver Side:

Received Data `11010111` (Even Parity)

- Counts 1s = 5 (expected even with parity) → Error Detected (Bit flipped to `11000111`).

- No Error Case: `11010111` → 6 ones (even) → Accepted.

2. Checksum:

Sender Side:

1. Divide data into 16-bit segments.

   - Segment 1: `1100 1010 1111 0001`

   - Segment 2: `1011 0101 1000 1110`

2. Sum segments:

```
1100101011110001
+ 1011010110001110
------------------
1 0111000001111111
(Overflow discarded)
```

3. Take 1's complement → Checksum = `1000111110000000`.

Error Detection Methods in Data Communication (With Sender & Receiver Side Examples)

4. **Sent Data:** Original segments + checksum.

Receiver Side:

1. Re-calculate sum of received segments + checksum.

2. If result = `1111 1111 1111 1111` → No Error.

3. If any bit flips (e.g., `1011` → `0011`), sum ≠ `FFFF` → **Error Detected**.

---

3. Cyclic Redundancy Check (CRC)

Sender Side:

1. Data (D): `1101011011`

2. Polynomial (P): `10011` (`$x^4 + x + 1$`)

3. Append `(n-1)` zeros → `1101011011`**`0000`**.

4. Perform binary division:

```
11010110110000 ÷ 10011 → Remainder = `1110`
```

5. Sent Data: `11010110111110`.

Receiver Side:

1. Divide received data by `10011`.

2. If remainder = `0` → No Error.

3. If remainder ≠ `0` (e.g., `1101` due to bit flip) → Error Detected.

Error Detection Methods in Data Communication (With Sender & Receiver Side Examples)

4. Longitudinal Redundancy Check (LRC)

Sender Side:

1. Arrange data in a block (e.g., 4x4 bits):

```
1 0 1 1
0 1 1 0
1 1 0 1
0 0 1 0
```

2. Calculate row-wise parity (LRC): `0 0 1 1`.

3. Sent Data: Original block + LRC row.

Receiver Side:

1. Re-calculate LRC on received block.

2. If LRC matches (e.g., `0 0 1 1`) → No Error.

3. If LRC mismatches (e.g., `0 1 1 1` due to bit flip) → Error Detected.

Method	Sender Action	Receiver Verification
Parity	Adds parity bit's(even/odd)	Counts 1 + parity bit
Checksum	Computes 1s Complement of sum	Sums all segments + checksum
CRC	Appends remainder after polynomial div	Divides by polynomial check remainder
LRC	Adds block parity rows	Re-calculates block parity

Error Detection Methods in Data Communication (With Sender & Receiver Side Examples)