# **Error Detection and Correction**

1. Why do errors occur in data transmission?

| **Source of error** | **Cause** |
| --- | --- |
| Line outages | Storms, accidents and animals! |
| Noise | Random electron movement. |
| Crosstalk | Using adjacent wireless channels. Cables not adequately shielded. |
| Echo | Poor connections. |
| Attenuation | Gradual decrease in signal over distance. |
| Intermodulation | Signals from several circuits combine. |
| Jitter | Sender and receiver clocks are not perfectly synchronised. |

2. Parity is one method of error detection. Show how a parity bit is added to the data stream using an odd parity protocol.

| **Data to send** | **Parity bit** | **What happens when a device receives the data with a parity bit:** |
| --- | --- | --- |
| 01101010 | 1 | The number of “on” bits (bits set to 1) are counted, this includes the parity bit.  If we are using odd parity, then the total number of “on” bits should be odd.  If it is even, then we assume and error has occurred during transmission and we ask for the data to be sent again.  If two bits change then parity doesn’t work. |
| 11001101 | 0 |
| 10101100 | 1 |

3. Parity is used to detect an error, but it can also be used to correct errors too.  
Assuming odd parity, show what happens if the data stream received is: 0011 0100 1011 0111  
Draw a circle around the bit that is incorrect.

| 0 | 0 | 1 | 1 |
| --- | --- | --- | --- |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |

One obvious disadvantage is that the data stream now contains more data to be transmitted. Increasing the chance of error and decreasing the data transfer rate.

| ~~0~~1 | 0 | ~~1~~0 | 1 |  | 4. Show why the data stream 0111 0100 1011 0111 can be received as  1101 0100 0001 0111 and still not be detected as corrupt. In what unique situation is parity not self-correcting? |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 0 | 0 |  |
| ~~1~~0 | 0 | ~~1~~0 | 1 |  | If two or more bits change, we know that an error has occurred, but not where. In this case we may need to resend the data. If the 4 bits in each corner change then an error will not be detected. This is a serious problem for the self-correcting algorithm but happens too infrequently to be a concern. Probably fine for non-critical systems. |
| 0 | 1 | 1 | 1 |  |
|  |  |  |  |  |

5. Find out how alternative algorithms, echo back and checksums work.

Echo back is a primitive error control method in which the receiving device echoes the received data back to the transmitting device, character by character.

Echo back is too much of a strain on bandwidth and doesn’t self-correct. It has potential to generate more errors as the data transfer is doubled!

A checksum is an error-detection method where we compute a numerical value according to the number of set or unset bits in a message and then send this value along with the message.

At the receiver end, the same checksum algorithm is applied to the message to retrieve the numerical value.

If the received checksum value matches the sent value, the transmission is considered to be successful and error-free.

Cyclic redundancy checks (self-correcting checksums) are used by devices communicating today. As they use more than one bit for error checking they are more reliable than parity.