**Error detection and correction**

In addition to framing, the data link layer may also perform mechanisms to detect and recover from transmission errors. For a receiver to detect transmission errors, the sender must add redundant information as an [error detection](https://en.wikipedia.org/wiki/Error_detection_and_correction) code to the frame sent. When the receiver obtains a frame with an error detection code it recomputes it and verifies whether the received error detection code matches the computed error detection code.

An error detection code can be defined as a function that computes the *r* (amount of redundant bits) corresponding to each string of *N* total number of bits. The simplest error detection code is the [parity bit](https://en.wikipedia.org/wiki/Parity_bit), which allows a receiver to detect transmission errors that have affected a single bit among the transmitted *N + r* bits. If there are multiple flipped bits then the checking method might not be able to unveil this on the receiver side. More advanced methods than parity error detection do exist providing higher grades of quality and features.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **H** | **E** | **L** | **L** | **O** |
| 8 | 5 | 12 | 12 | 15 |

A simple example of how this works using [metadata](https://en.wikipedia.org/wiki/Metadata) is transmitting the word "HELLO", by encoding each letter as its position in the alphabet. Thus, the letter *A* is coded as 1, *B* as 2, and so on as shown in the table on the right. Adding up the resulting numbers yields 8 + 5 + 12 + 12 + 15 = 52, and 5 + 2 = 7 calculates the metadata. Finally, the "8 5 12 12 15 7" numbers sequence is transmitted, which the receiver will see on its end if there are no transmission errors. The receiver knows that the last number received is the error-detecting metadata and that all data before is the message, so the receiver can recalculate the above math and if the metadata matches it can be concluded that the data has been received error-free. Though, if the receiver sees something like a "7 5 12 12 15 7" sequence (first element altered by some error), it can run the check by calculating 7 + 5 + 12 + 12 + 15 = 51 and 5 + 1 = 6, and discard the received data as defective since 6 does not equal 7.

More sophisticated error detection and correction algorithms are designed to reduce the risk that multiple transmission errors in the data would cancel each other out and go undetected. An algorithm that can even detect if the correct bytes are received but out of order is the [cyclic redundancy check](https://en.wikipedia.org/wiki/Cyclic_redundancy_check) or CRC. This algorithm is often used in the data link layer.