ITS 323 – DIGITAL DATA COMMUNICATION TECHNIQUES EXAMPLES

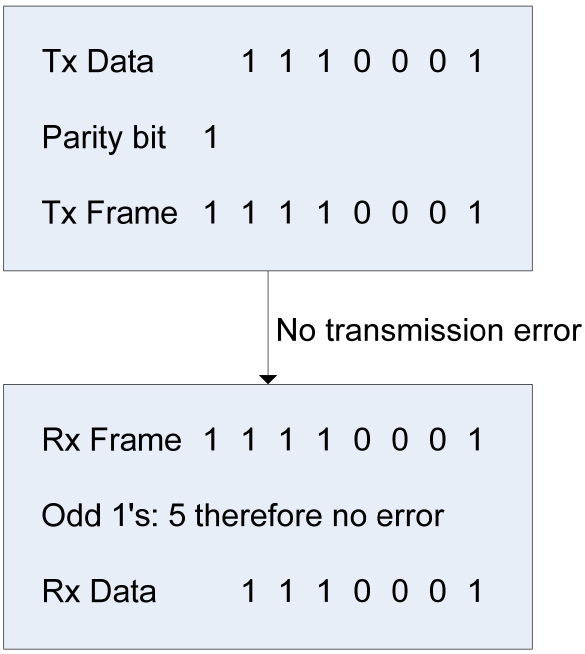
1 Error Detection

The lecture notes includes an example of using *single bit odd parity* check for error detection. Lets look at it closer.

With odd parity check:

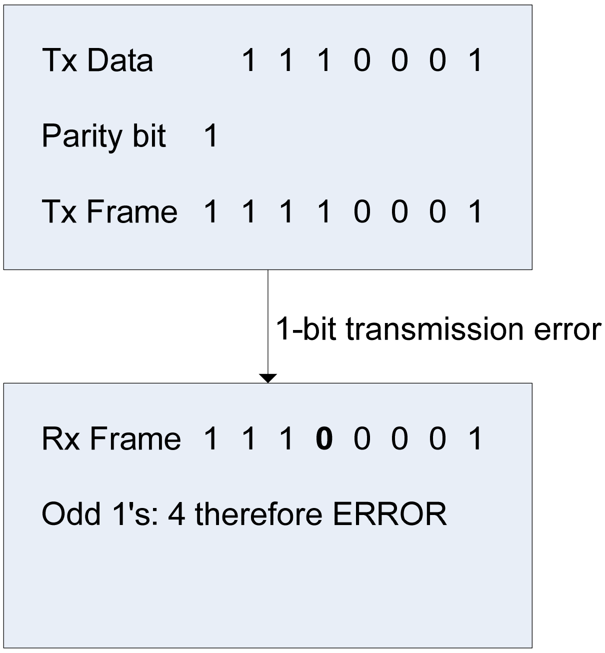
- The transmitter adds 1 bit to the data so that the total number of 1's is now odd.
- The receiver counts the number of 1's in the received frame: if it is odd, then assume no error; if it is even, then assume error.

The following diagram shows the data transmission when there are no errors.



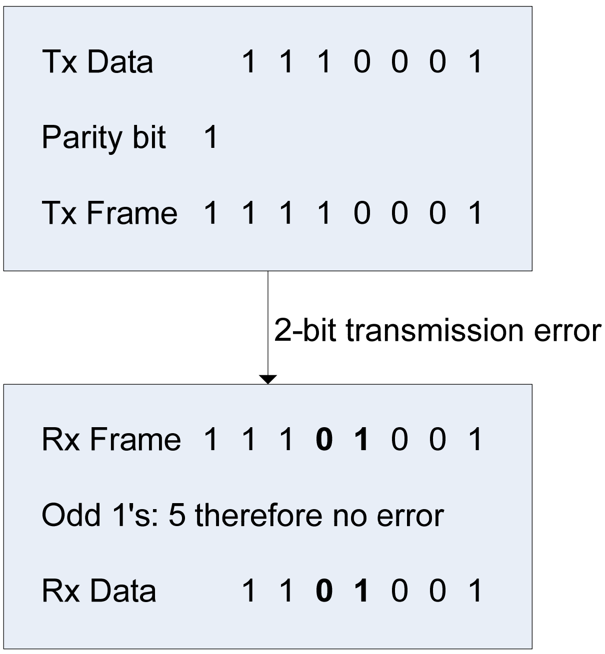
The parity check is successful: no error occurs, and no error is detected. The correct data is received.

The following diagram shows the data transmission when there is a single bit in error on the transmission.



The parity check is successful: an error occurs, and the error is detected. The receiver does not receive the data (it will have to do something else in order to get the data, e.g. ask for a retransmission).

The following diagram shows the data transmission when there are 2 bits in error on the transmission.



The parity check us **unsuccessful!** Errors occur, but the check does not detect these errors, and as a result the receiver receives incorrect data.

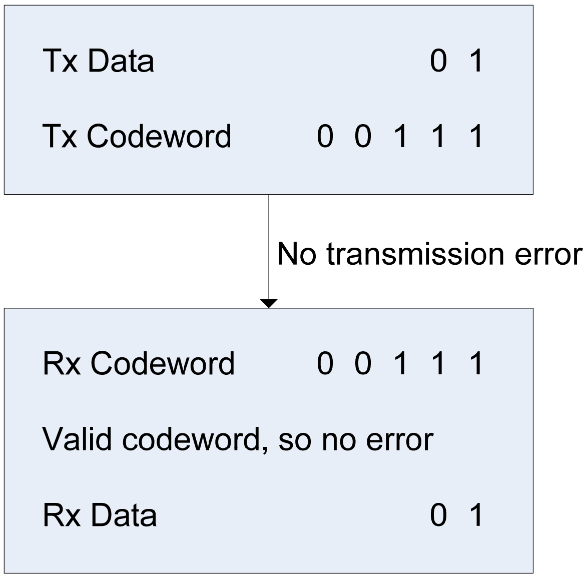
2 Forward Error Correction

The example used in the lecture was that of a Error Correcting Code that maps two bits of data into a 5-bit codeword as per the following:

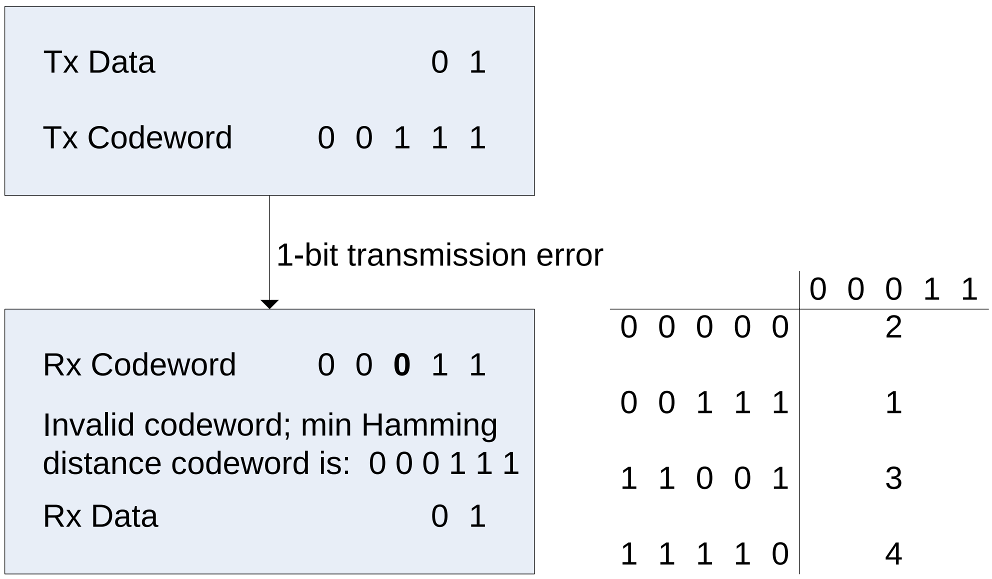
- 00 => 00000
- 01 => 00111
- 10 => 11001
- 11 => 11110

If the received codeword is invalid (that is, not in the above list), then the ECC chooses the codeword with the minimum Hamming distance. There must be a code with a unique minimum Hamming Distance, otherwise the ECC at the receiver cannot choose a codeword (and the error is uncorrected, but detected).

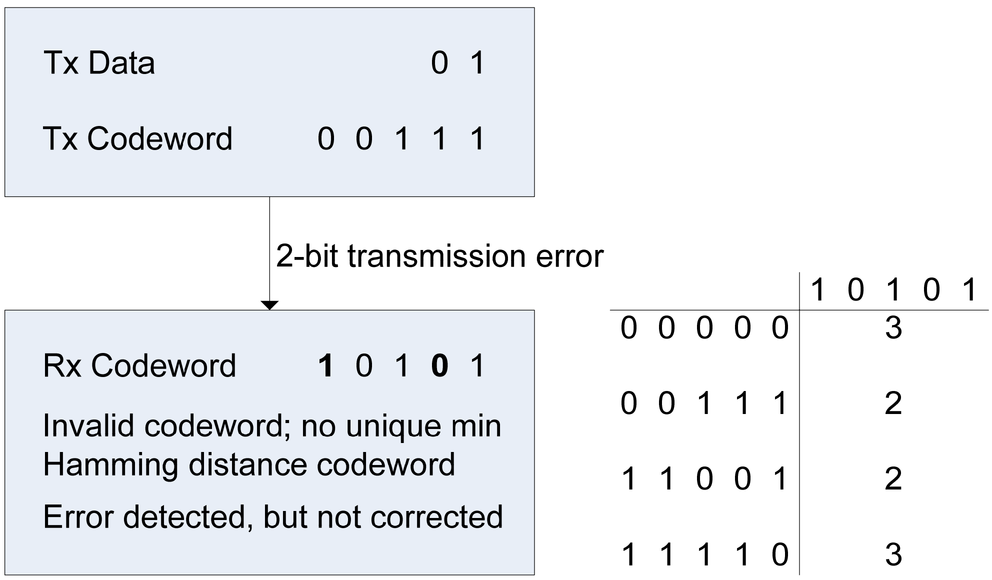
The following diagram shows a case of no transmission error, and therefore the ECC at the receiver correctly decoding the original data.



The following diagram shows a case of a 1-bit transmission error, and the ECC at the receiver correctly detecting and correcting the error, to recover the original data.



The following diagram shows a case of a 2-bit transmission error, and the ECC at the receiver detecting the error, but not able to correct (since there are two possible valid codewords with Hamming distance 2 – the receiver does not know which codeword is the transmitted codeword).



Of course there could be a fourth outcome: bit errors occur that the ECC at the receiver cannot correct or detect!

3 Errors

NOTE: This section wasn't covered in lectures. It is just included as extra detail for those people interested.

Example 6.3 (p171) from Stallings

An impulse noise event or a fading event of 1us occurs. At a data rate of 10Mb/s, there is a resulting error burst of 10 bits. At a data rate of 100 Mb/s, there is an error burst of 100bits.

Probability of Frame Errors

Assume data are transmitted as one or more contiguous sequences of bits, called frames.

- P_b: Probability that a bit is received in error; also known as the bit error rate (BER)
- P_1: Probability that a frame arrives with no errors
- P_2: Probability that, with an error-detecting algorithm in use, a frame arrives with one or more undetected errors
- P_3: Probability that, with an error-detecting algorithm in use, a frame arrives with one or more detected bit errors, but no undetected bit errors

If we first assume there is no error-detection performed, then $P_3 = 0$.

$$P_1 = (1-P_b)^F$$

$$P_2 = 1 - P_1$$

where F is the number of bits in frame.

Example 6.4 (p172) from Stallings

Objective of ISDN system: BER is less than 10^{-6} on a 64kb/s channel on at least 90% of observed 1-minute intervals

User requirement: one frame with undetected error can occur per day (on continuously transmitting channel)

Assume frame length is 1000 bits

Number of frames that can be transmitted in a day is 5.529×10^6

Desired frame error rate, $P_2 = 1/(5.529 \times 10^6) = 0.16 \times 10^{-6}$

If we assume P_b is 10^{-6}, then $P_1 = (0.999999)^{1000} = 0.999$ and therefore $P_2 = 10^{-3}$

Therefore P_2 is too large to meet our user requirement.