Error Detection

- Assume that a DLC module receives a frame
- The DLC must determine if the packet (data) in the frame is error-free
- Since any bit combination in the packet may be valid, extra bits must be added (in the header or trailer) to allow error detection
- Standard schemes:
  - Single parity checks
  - Horizontal and vertical parity checks
  - Parity check codes
  - Cyclic redundancy check (CRC) codes

Single Parity Checks

- Inadequate for reliable error detection
- Burst errors
  - Cause errors in a sub-string of arbitrary length
  - A burst error is as likely to cause an even number of errors (undetectable) as an odd number of errors (detectable)

Horizontal and Vertical Parity Checks

- Arrange a string of bits as a two-dimensional array and compute parity over each row and each column of the array
- Can detect
  - Any number of errors in a single row (detect even number of errors with column parity)
  - Any number of errors in a single column (detect even number of errors with row parity)

Parity Check Codes

- Parity check codes may be used to more efficiently detect multiple errors or for *error correction*
- Hamming codes, named in honor of Richard Hamming

Cyclic Redundancy Check (CRC) codes

- Cyclic redundancy check (CRC) codes are the most widely used in DLCs
- A polynomial is generated and represented as an error code (CRC code); receiver calculates polynomial from received data that will be divisible by the error code polynomial if there is no error
- Burst errors are hard to model -- three parameters are typically used to measure the effectiveness of a code for error detection
- 1. Minimum distance of the code
  2. Burst-detecting capability
  3. Probability that a random string is accepted as being error-free