UNIT - DATA LINK LAYER

The Services Provided by the Link Layer

- **Framing**: Almost all link-layer protocols encapsulate each network-layer datagram within a link-layer frame before transmission over the link. The structure of the frame is specified by the link-layer protocol.
- **Link access:** A medium access control (MAC) protocol specifies the rules by which a frame is transmitted onto the link. For point-to-point links that have a single sender at one end of the link and a single receiver at the other end of the link, the MAC protocol is simple the sender can send a frame whenever the link is idle. The more interesting case is when multiple nodes share a single broadcast link - the so-called multiple access problem.
- **Reliable delivery:** When a link-layer protocol provides reliable delivery service, it guarantees to move each network-layer datagram across the link without error. A link-layer reliable delivery service can be achieved with acknowledgments and retransmissions.
- **Error detection and correction:** The link-layer hardware in a receiving node can incorrectly decide that a bit in a frame is zero when it was transmitted as a one, and vice versa. Such bit errors are introduced by signal attenuation and electromagnetic noise. Because there is no need to forward a datagram that has an error, many link-layer protocols provide a mechanism to detect such bit errors. This is done by having the transmitting node include error-detection bits in the frame, and having the receiving node perform an error check.

Where is the link layer implemented?

- It is implemented in each and every host.
- The link layer is implemented in a network adapter, also sometimes known as a network interface card (NIC).

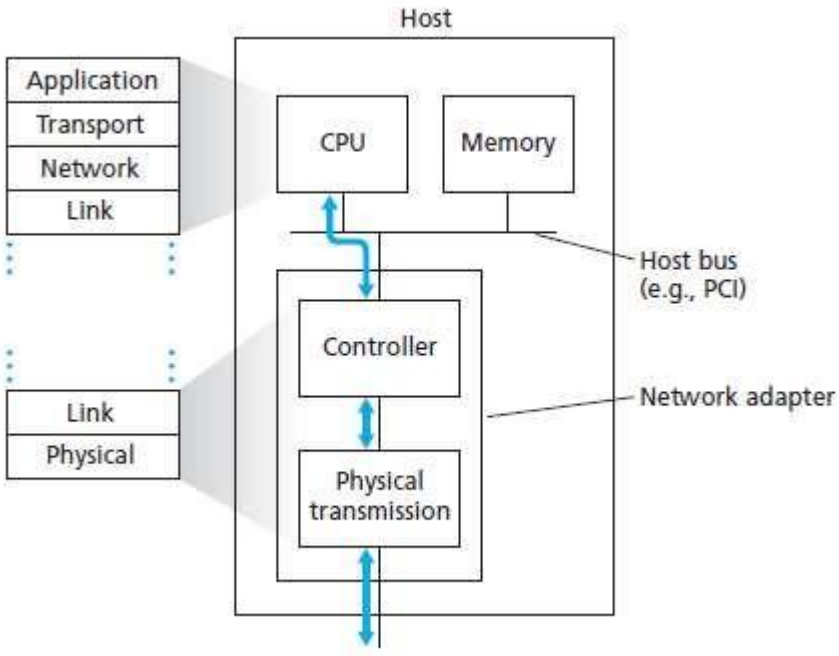


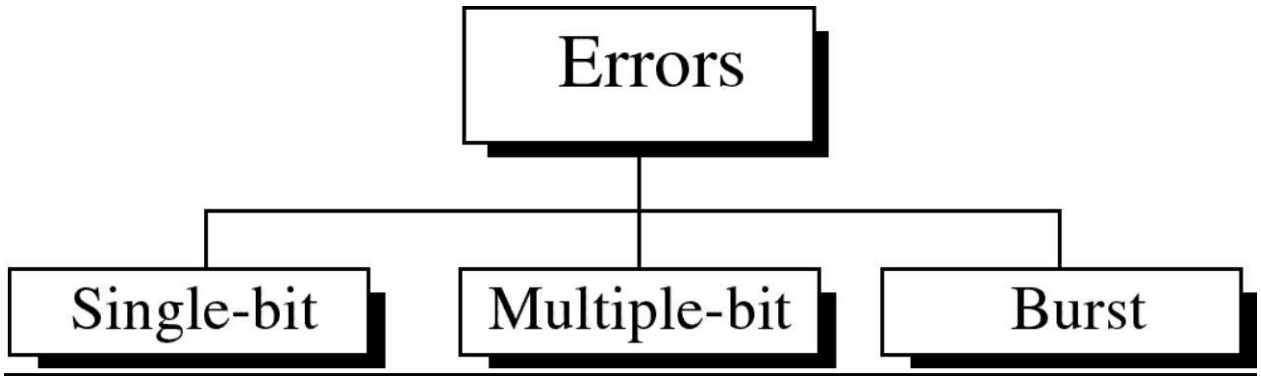
Fig. 1: Network adapter: its relationship to other host components and to protocol stack functionality

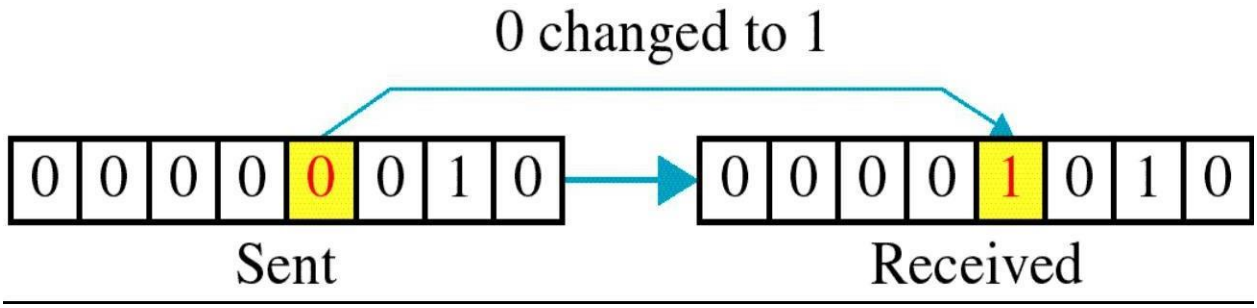
- Figure 1 shows a network adapter attaching to a host's bus (e.g., a PCI or PCI-X bus), where it looks much like any other I/O device to the other host components.

- Figure 1 also shows that while most of the link layer is implemented in hardware, part of thelink layer is implemented in software that runs on the host's CPU.
- The software components of the link layer implement higher-level link layer functionality such as assembling link-layer addressing information and activating the controller hardware.
- On the receiving side, link-layer software responds to controller interrupts (e.g., due to the receipt of one or more frames), handling error conditions and passing a datagram up to the network layer.
- Thus, the link layer is a combination of hardware and software-the place in the protocol stack where software meets hardware.

TYPES OF ERRORS:

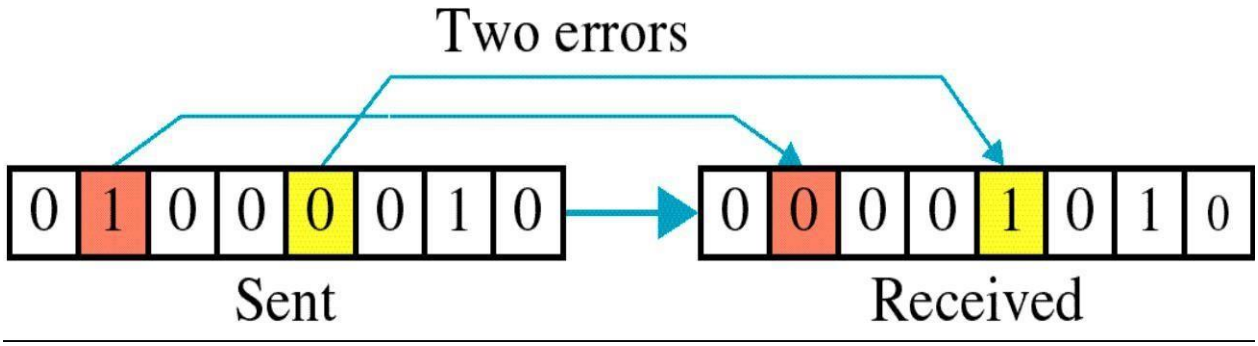
- In a single-bit error, only 1 bit in the data unit has changed.
- In a multi-bit error, more than one bit in the data unit has changed.
- A burst error means that 2 or more bits in the data unit have changed.

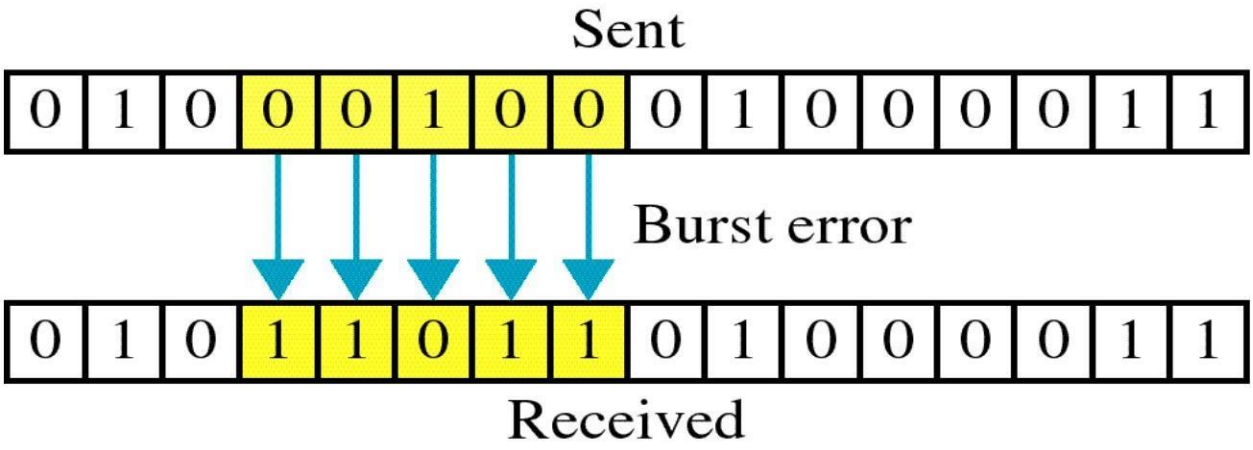




Single Bit Error:

Multiple-bit error

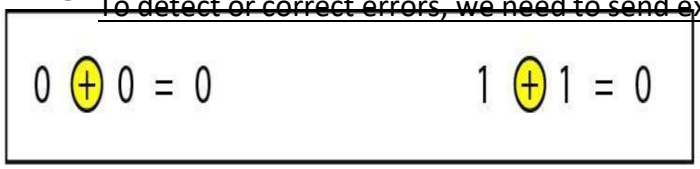




Burst error:

XORing of two single bits or two words:

- To detect or correct errors, we need to send extra (redundant) bits with data.

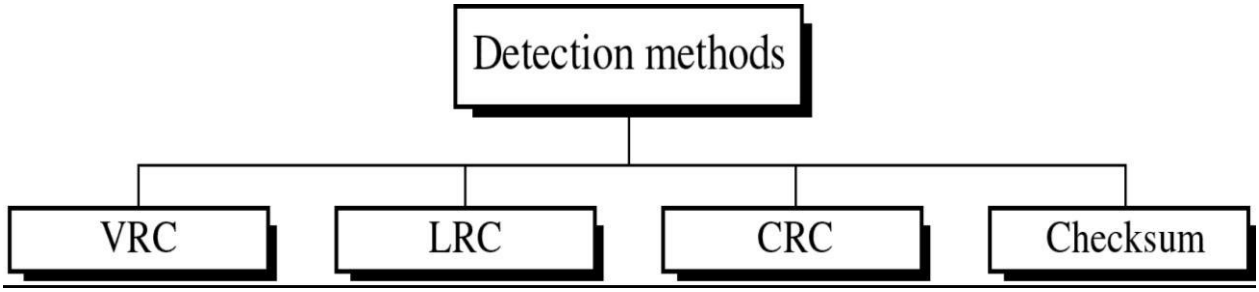


a. Two bits are the same, the result is 0.

b. Two bits are different, the result is 1.

c. Result of XORing two patterns

Q. Error Detection methods:



Error Detection and Correction Techniques

- Techniques for error detection
    1. parity checks
    2. checksum methods
    3. cyclic redundancy check

Parity checks

- In this technique, a redundant bit called parity bit, is appended to every data unit so that thenumber of 1s in the unit including the parity becomes even.
- Blocks of data from the source are subjected to a check bit or Parity bit generator form, where a parity of 1 is added to the block if it contains an odd number of 1's and 0 is added if it contains an even number of 1's.
- At the receiving end the parity bit is computed from the received data bits and compared with the received parity bit.
- This scheme makes the total number of 1's even, that is why it is called even parity checking.

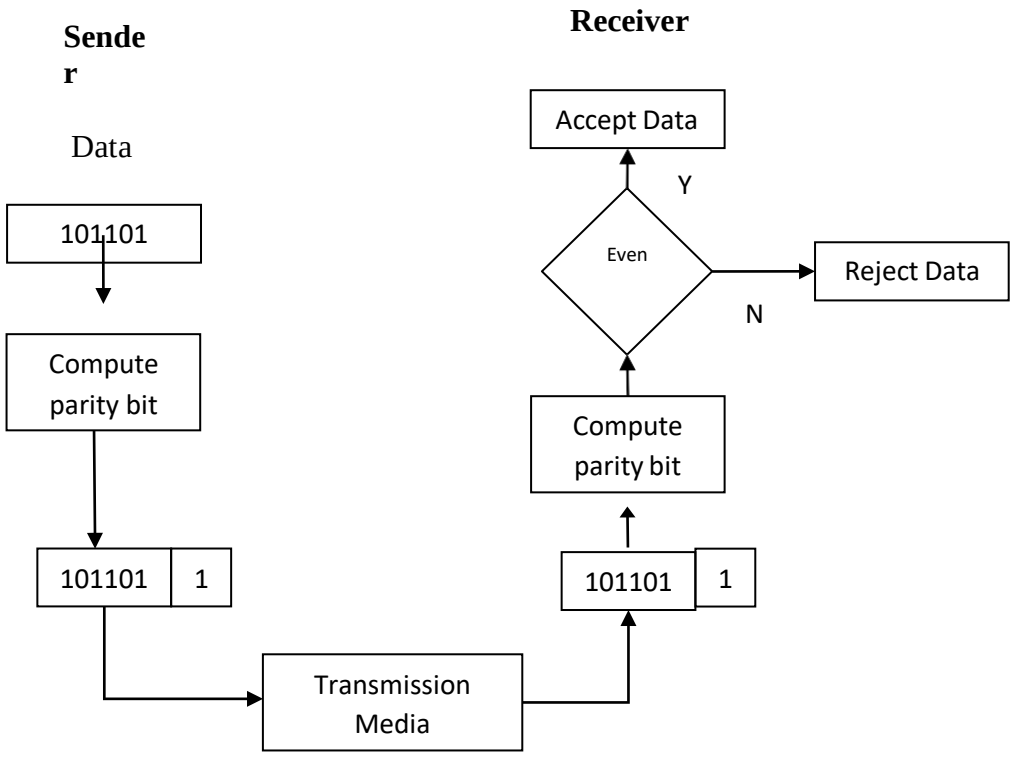


Fig. 2: Even parity checking scheme

Performance

- A receiver can detect all single bit errors in each code word.
- Errors in more than one bit cannot be detected.

Two-dimension Parity Check

- Performance can be improved by using two-dimensional parity check, which organizes theblock of bits in the form of a table.
- Parity check bits are calculated for each row, which is equivalent to a simple parity check bit.
- Parity check bits are also calculated for all columns then both are sent along with the data.
- At the receiving end these are compared with the parity bits calculated on the received data.

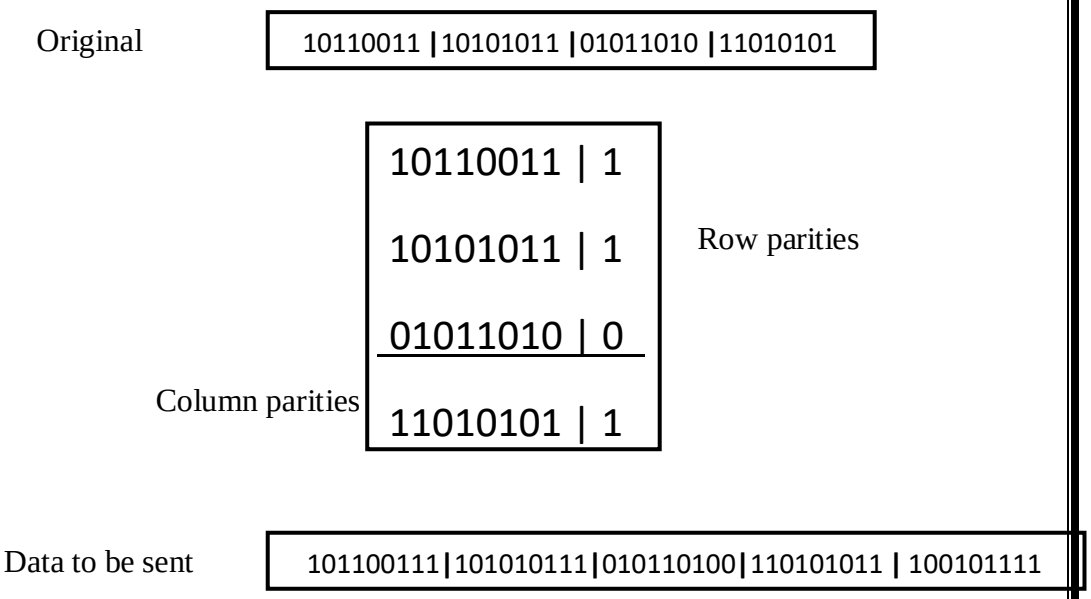


Fig. 3: Two-dimensional parity check

Performance

- Two- Dimension Parity Checking increases the likelihood of detecting burst errors.
- 2-D Parity check of n bits can detect a burst error of n bits.
- A burst error of more than n bits is also detected by 2-D Parity check with a high-probability.
- If two bits in one data unit are damaged and two bits in exactly same position in anotherdata unit are also damaged, the 2-D Parity check checker will not detect an error.

Checksum

- Here, the data is divided into k segments each of m bits.
- In the sender's end the segments are added using 1's complement arithmetic to get the sum.
- The sum is complemented to get the checksum.
- The checksum segment is sent along with the data segments.
- At the receiver's end, all received segments are added using 1's complement arithmetic toget the sum. The sum is complemented.
- If the result is zero, the received data is accepted; otherwise discarded.

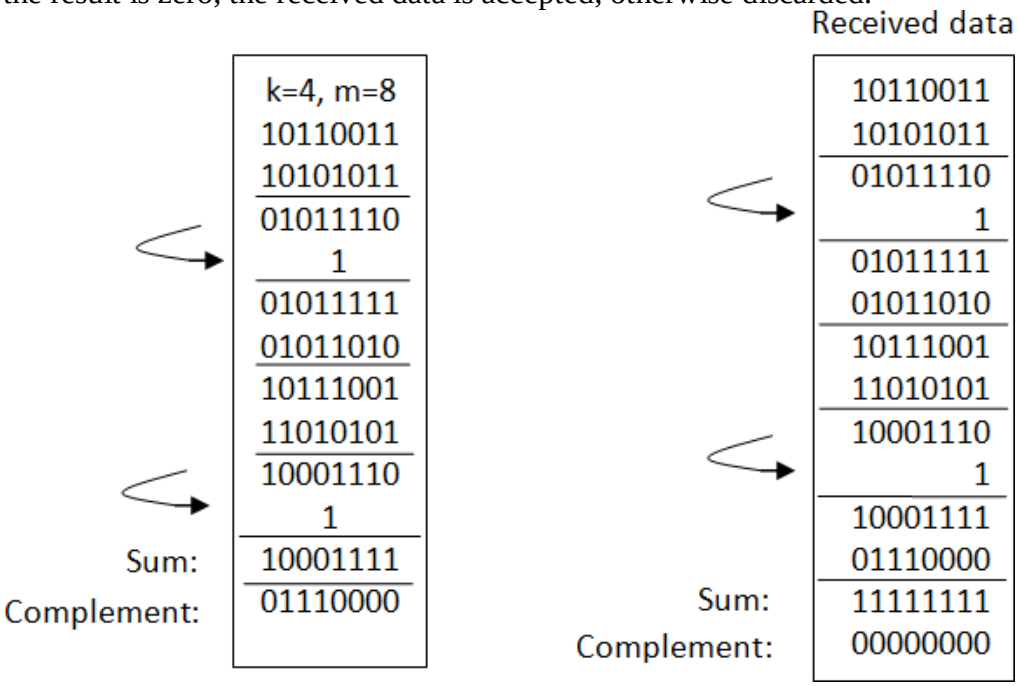


Fig. 4: Checksum

Performance

- The checksum detects all errors involving an odd number of bits.
- It also detects most errors involving even number of bits.

Cyclic Redundancy Checks (CRC)

- CRC is the most powerful and easy to implement technique.
- CRC is based on *binary division*.
- In CRC, a sequence of redundant bits, are appended to the end of data unit so that theresulting data unit becomes exactly divisible by a second, predetermined binary number.
- At the destination, the incoming data unit is divided by the same number.
- If at this step there is no remainder, the data unit is assumed to be correct and is thereforeaccepted.
- A remainder indicates that the data unit has been damaged in transit and therefore must berejected.
- The binary number, which is (r+1) bit in length, can also be considered as the coefficients ofa polynomial, called *Generator Polynomial*.

Performance

- CRC is a very effective error detection technique.
- If the divisor is chosen according to the previously mentioned rules, its performance can besummarized as follows
- CRC can detect all single-bit errors
- CRC can detect all double-bit errors (three 1's)
- CRC can detect any odd number of errors (X+1)
- CRC can detect all burst errors of less than the degree of the polynomial.

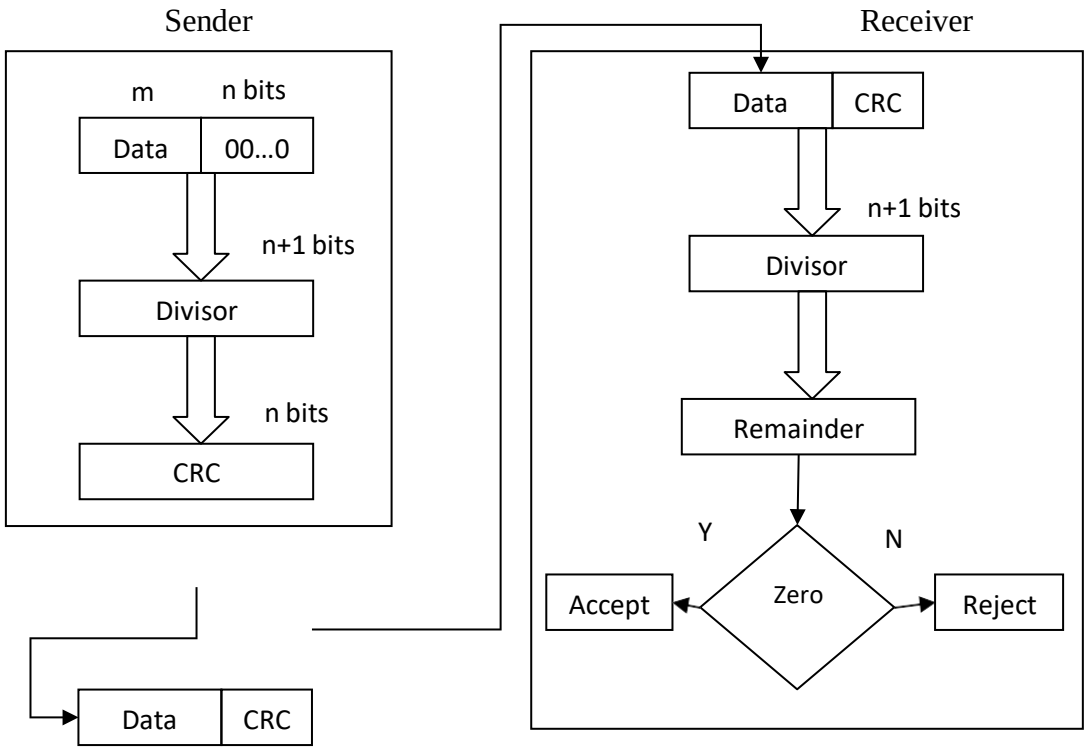


Fig. 5: Basic scheme for Cyclic Redundancy Check

Frame : 1 1 0 1 0 1 1 0 1 1

Generator: 1 0 0 1 1

Message after appending 4 zero bits: 1 1 0 1 0 1 1 0 0 0 0

```
                1 1 0 0 0 0 1 0 1 0
10011 | 1 1 0 1 0 1 1 0 1 1 0 0 0 0
        1 0 0 1 1
          1 0 0 1 1
          1 0 0 1 1
              0 0 0 0 1
              0 0 0 0 0
```

Frame : 1 1 0 1 0 1 1 0 1 1
Generator: 1 0 0 1 1
Message after appending 4 zero bits: 1 1 0 1 0 1 1 0 0 0 0

```
                1 1 0 0 0 0 1 0 1 0
          ┌─────────────────────────
10011     │ 1 1 0 1 0 1 1 0 1 1 0 0 0 0
            1 0 1 0 0
            1 0 0 1 1
            ---------
              0 1 1 1 0
              0 0 0 0 0
              ---------
                1 1 1 0   <- Remainder
```

Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 1 0

Q. Error Correction: Hamming Code:

- Redundancy is needed for error handling
- m data bits and r redundant bits
- for m + r bits only one correct value of r for a given m
- one correct bit pattern requires m + r incorrect patterns
- $\mathbf{m + r + 1 < 2^n}$
- 7 bit data 4 bit redundant bits makes it 11

0	0	1	1	0	0	1	0	0	0	0
1	2	3	4	5	6	7	8	9	10	11

R1	R2	M1	R3	M2	M3	M4	R4	M5	M6	M7
1	2	3	4	5	6	7	8	9	10	11

(a)

R1	R2	1	R3	0	0	1	R4	0	0	0
1	2	3	4	5	6	7	8	9	10	11

(b)

R1	R2	M1	R3	M2	M3	M4	R4	M5	M6	M7
1	2	2+1	4	4+1	4+2	4+2+1	8	8+1	8+2	8+2+1

(c)

R1 to R4 = Redundant bits
M1 to M7 = Data bits

- R1 represents the parity of M1, M2, M4, M5, 1 0 0 1 0 0 = 0
- *R2 represents the parity of M1, M3, M4, M6, and M7 = 1 1 0 0 =0*
- R3 represents the parity of M2, M3, and M4=1
- R4 represents the parity of M5, M6, and M7=0
- Method can be handled in two ways:
- when an error is discovered, the receiver can have the sender retransmit the entire data unit.
- a receiver can use an error-correcting code, which automatically corrects certain errors.
- Hamming Code ~ developed by R.W. Hamming
- positions of redundancy bits in Hamming code
    - Note: Do it the way I taught you in lecture. Follow lecture notes.