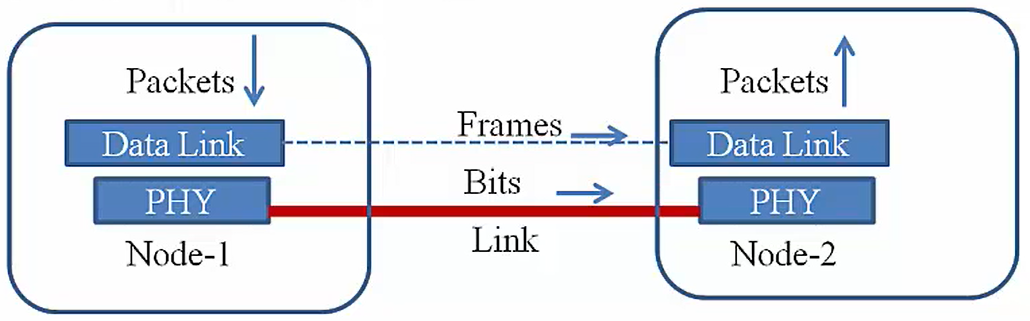
**Data Link Layer**

**Lecture Summary**

* Introduction and services of data link layer (framing, LLC (PPP, BINSINK, HDLC), MAC)
* Error detection (parity bit, crc, checksum) and correction techniques (hamming code)
* Multiple access protocols (INSIDE MAC)
* Link-Layer Addressing
* ARP
* Ethernet Frame format (MAC PROTOCOL) AND WIRELESS MAC 802.11
* GIGABIT Ethernet
* Link Layer Switching & VLANs

1. **Overview of Data Link Layer**
   1. **Introduction**

* The data link layer as well as physical layer does next-hop delivery.
* The data link layer does frame-by-frame next-hop delivery, physical layer does bit-by-bit next-hop delivery.
* Data link layer uses services of physical layer (which delivery bits) to delivery frames. Frame is a block of data exchanged at link layer.



* The data link layer delivers frames (bunch of bits) to physical layer, which converts the bits into signals and delivers to the other node. The physcial layer in the receiving node convert this signals back to bits and delivers to data link layer.
  1. **Link Layer Protocols**
* The link could be point-to-point (two nodes attached to link) or broadcast (many nodes connected to the same communication channel/link). Example-Wireless is a broadcast communication.
* The Link Layer Protocol define format of frames to be exchanged over the link as well as the actions to be taken by the nodes after receiving it.
* Examples of link layer protocols: Ethernet, Token-Ring, WiFi, PPP etc
  1. **Services at Link Layer**
* There is lots of actions at link layer. The link layer is **divided into two sub-layers.**

1. **LLC (Logical Link Control):**It acts as interface between the network layer and MAC sub-layer. Its main job is basically multiplexing. When a link layer receives a frame, it means to determine, which of the higher layer protocols it need to pass below to. This is multiplexing. Other job of LLC are error detection, error recovery (optional), flow control (optional). It is optional because not all link layer protocols employ error recovery and flow control. Example-Ethernet does not do error recovery and flow control, where as some of the wireless protocols do this.
2. **MAC (Media Access Control):** As the name suggests, it controls access to physical media. It mainly applicable to broadcast channels. If two nodes on the same channel want to communicate simultaneously, what will happen? Framing is implemented as a part of MAC.

* **Switching:** It interconnect LANs.

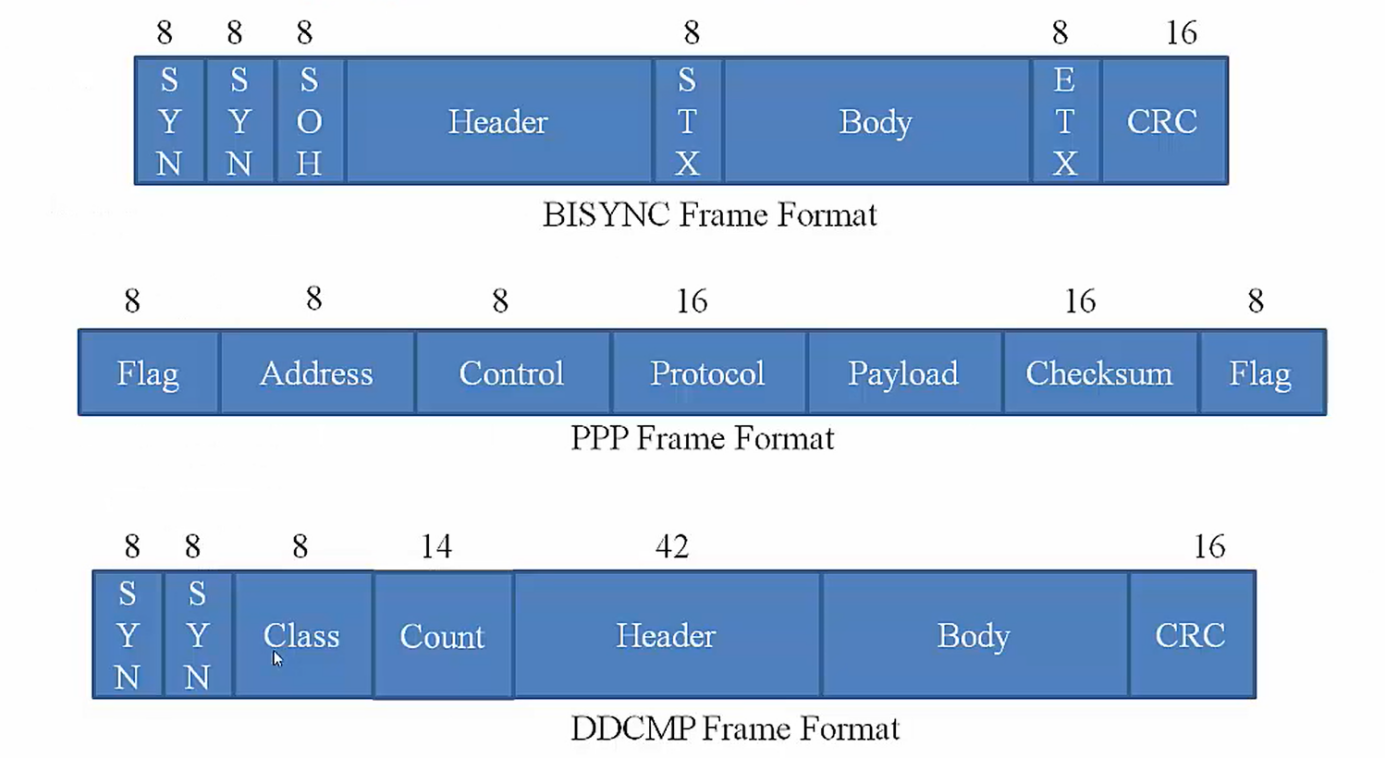
1. **Framing**
   1. **Introduction**

* Data-link layer takes packets from Network Layer and encapsulates them into Frames.Then, it sends each frame bit-by-bit on the hardware. At receiver’ end, data link layer picks up signals from hardware and assembles them into frames.
* A block of data which is termed as frame at link layer, exchanged between nodes.
* How do you determine which set of bits constitutes a frame? In writting how will you separate out words? We use spaces.
* **A possible approach.** Similar to space gap in writing is translated to time gap here (keep the link idle between the two frames) This approach can not be used because this produces a dependency on physical layer. Some encodings may use idle time to encode data. Some physical layer don’t keep link idle in order to maintain synchronisation.
* **Sentinel Approach:** Bookmark.
* A marker could be used at the beginning and end of the frames to show this is the beginning and end of the frame. This marker approach is called sentinel approach.
* This marker use a special character or bit sequence to indicate start and end of the frames.
* **Methods to Identify Frames**

Byte Count, Starting/Ending Byte (Byte Oriented), Starting/Ending Byte bit Oriented.

* 1. **Byte Oriented protocol**
* This views frame as a collection of bytes.
* In this approach, a special byte acts as sentinel.
* The example of byte oriented prtocols are

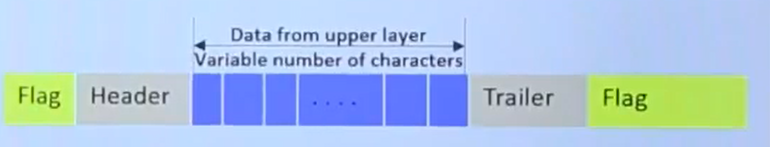
1. **BISYNC** (Binary Synchronous Communication) developed by IBM.
2. **DDCMP** (Digital Data Communication Message Protocol)
3. **PPP** (point-to-point protocol)

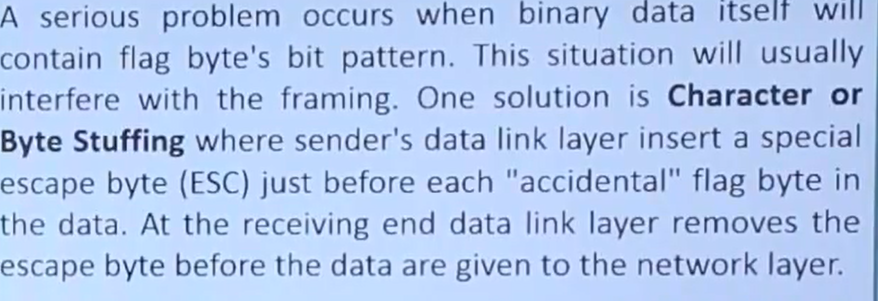


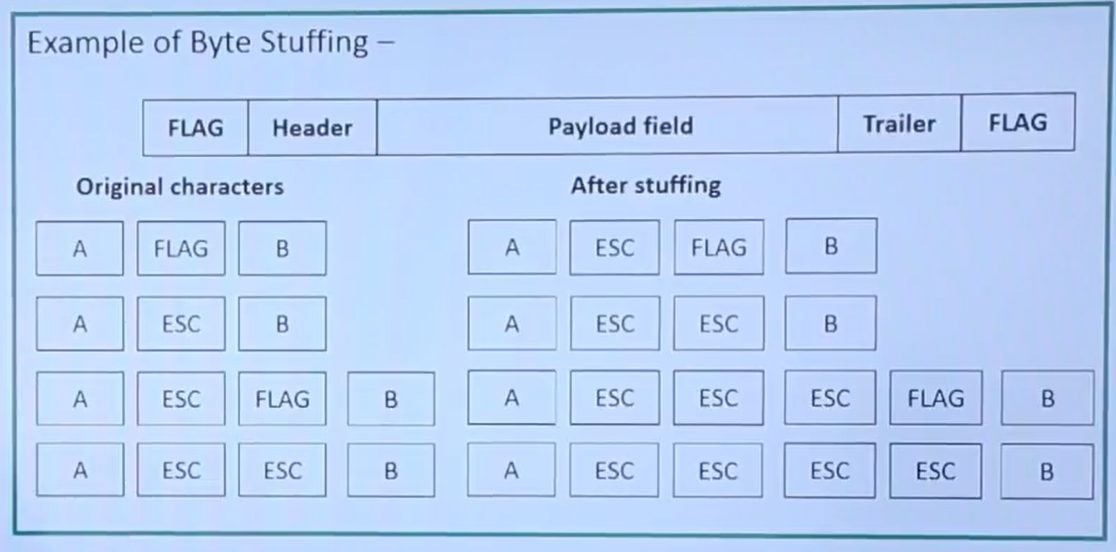
* 1. **Bit Oriented protocol**
* It views frame as a collection of bits.
* Bits could be from ASCII characters, pixel values of an image or binary file etc.
* Example is **HDLC (High level Data Link Control)** where the sequence is used as the sentinel to mark the beginning and end of the frame.

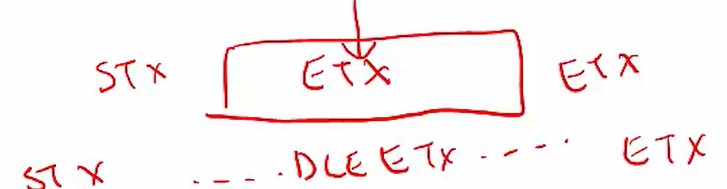


* 1. **Problems with Sentinel approach**
* There is a problem to use sentinel to mark beginning and end of the frame.
* What if the sentinel character (ETX or ending sequence) appears in the body (payload)? In this case the frame terminated permanently. This results errors.
* **Solution**
* **Byte/Character Stuffing** is a method used in byte oriented protocol. If the sentinel character is a part of the payload, then preceed it by DLE (data link escape) character.
* A problem will occurs when



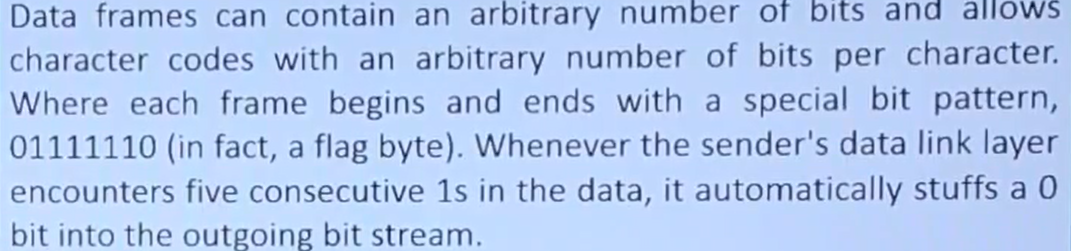


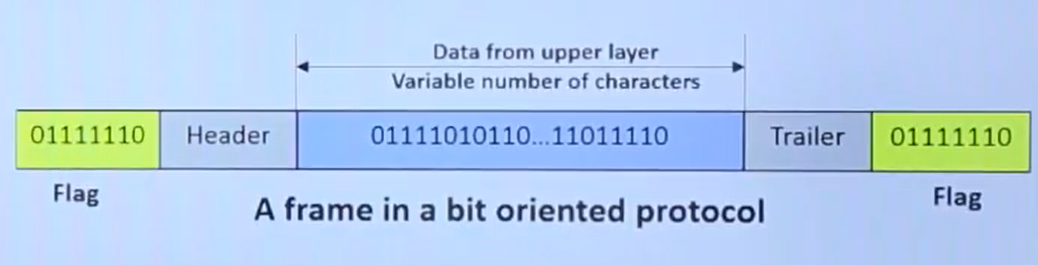


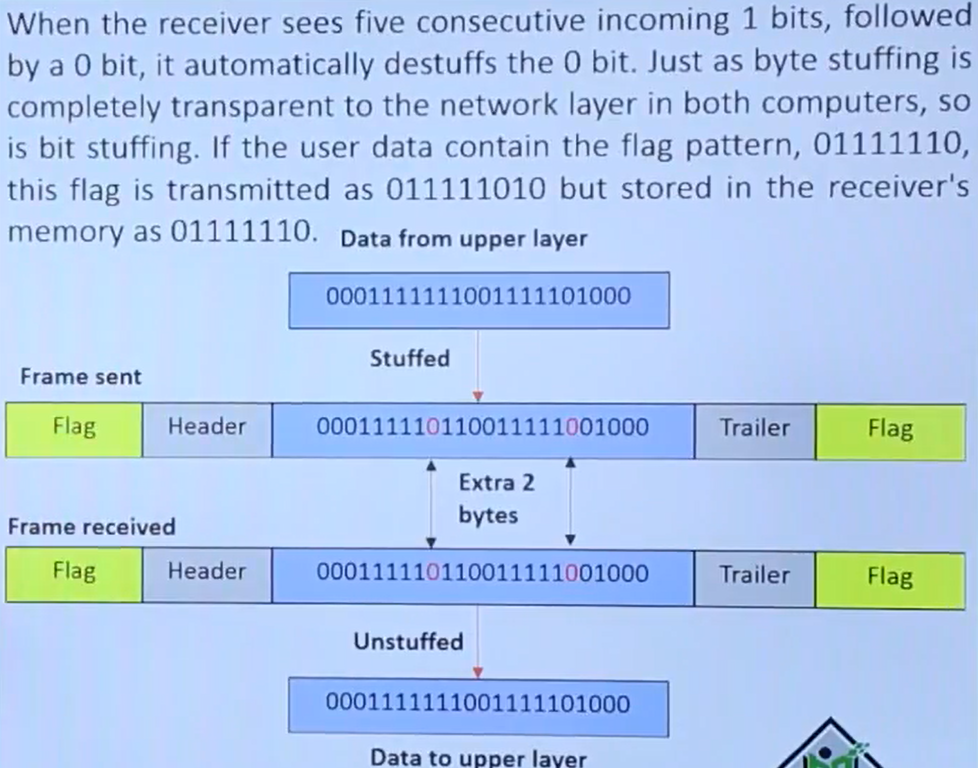


|  |  |
| --- | --- |
|  |  |

* **Bit Stuffing** is a method used in bit oriented protocol.
* If the sentinel bit sequence is a part of the payload, then what to do? Example. Let the sequence is 01111110. So in the body of the message, sender inserts a 0 after 5 consecutive 1’s. and receiver removes tha 0 that foolows 5 1’s. By doing so, 6 consecutive 1’s will never appear in the body of the message.



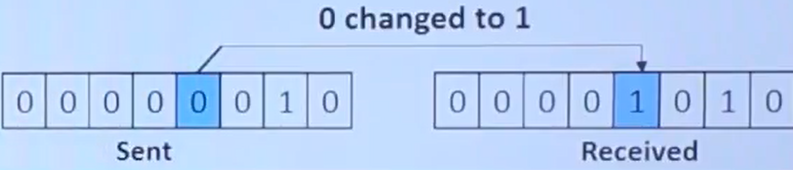




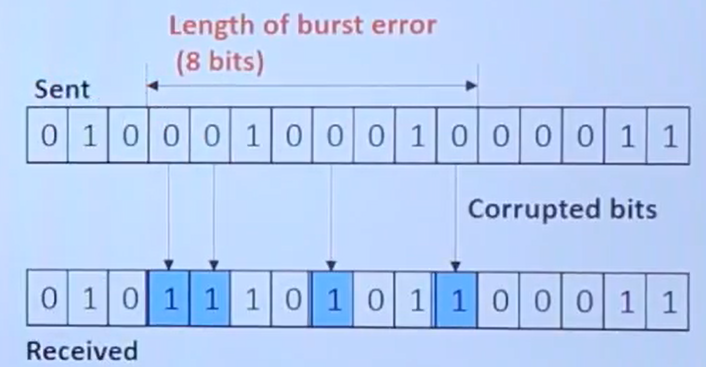
1. **Errors in Data Transmission**
   1. **Introduction**

* Data can be corrupted during transmission. In digital transmission systems, an error occurs, when a bit is altered between transmission and reception.
* Some applications requires that errors be detected and corrected.
* There are two general types of errors can occur.

1. **Single bit errors:**  Only one bit in the data unit has changed. It does not affect nearby bits.



1. **Burst errors:** Two or morebits in the data unit have changed. It affect nearby bits. A burst error of length B is a continuous sequence of B bits in which the first and last bits and any number of intermediate bits received an error.



* 1. **Error Detection**
* **What cause errors?** =>Distortions of signals due to frequency dependant attenuation, noise (physical layer)
* **Why detect errors?** => Data fidelity - we want to reach the data to other end intact, to prevent wastage of resouces.
* **What to do after detecting errors?**
* **Drop frame-**Higher layers (I.e.TCP) will recover or few losses donot hurt applications.
* **Recover frame:**

- **Error Correction:** Frame carries enough information to correct errors. It requires more redundant bits per frame than error detection. Redundant bits are sent all the time (every frame)

- **Re transmission**: receiver signals the sender on error, sender retransmits the frame. It requires another copy to be retransmitted.

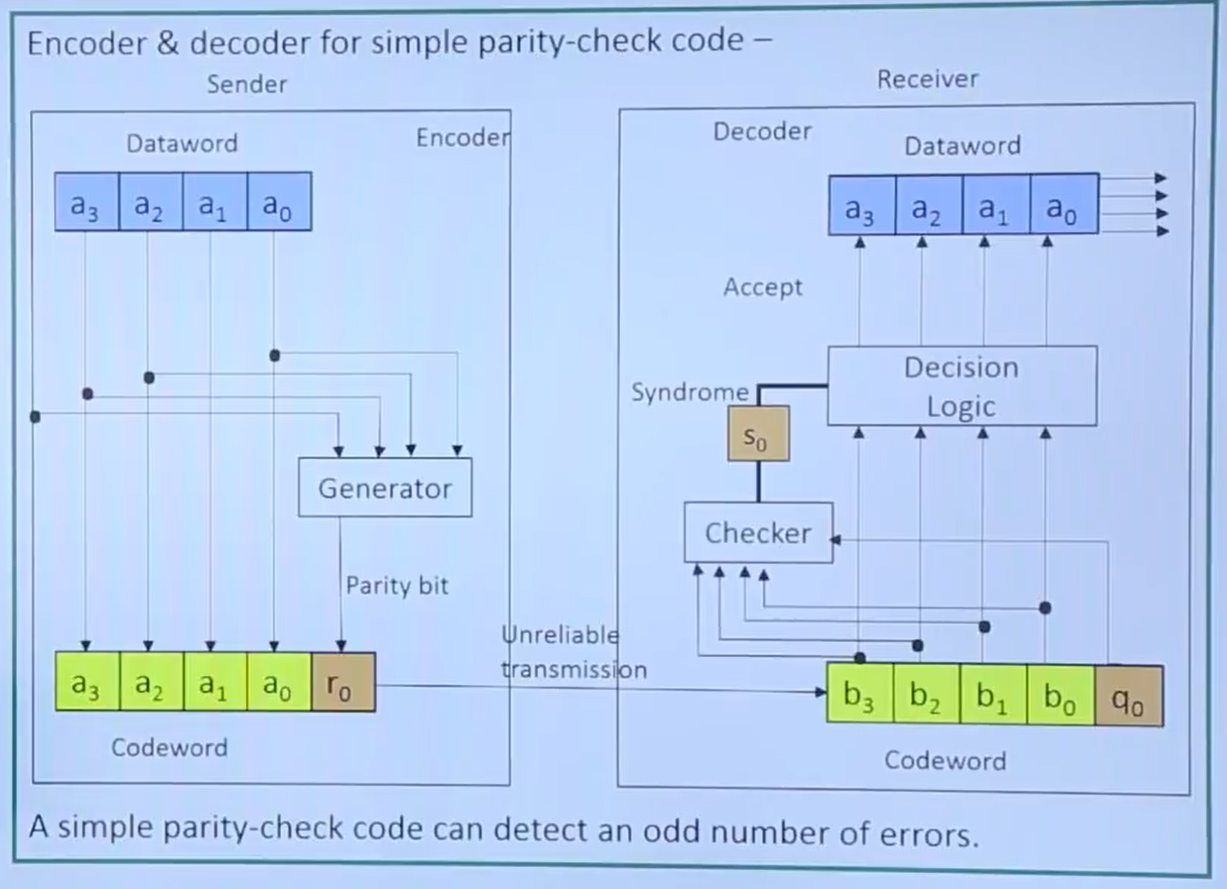
* **Error correction** is **useful** when
* Error rate is high (i.e.wireless)
* Cost (i.e.latency) of retransmission is too high.(satellite communication)
  1. **Framework for Error Detection Techniques**
* Add redundant information to a frame.
* **At Sender:** Add k bits of redundant data to a m bit message. k is derived from the original message through some algorithms. Example: k=32 and m=12000 fro Ethernet
* At Receiver: Reapply the same algorithm as sender to detect errors. Take corrective actions if necessary.
  1. **Different Types of Error Detection Techniques**

1. Simple (One Dimensional) Parity Checking
2. Two Dimensional Parity Checking
3. CRC
   1. **Simple Parity Checking**

* In this an extra bit (called parity bit) is added to each word before transmitting.
* **Even Parity:** Set the parity bit such that the total number of 1’s in the code word is **even.** Example: The word 1100 will be send with even parity as 11000, 1000 as 10001
* **Odd Parity:** Set the parity bit such that the total number of 1’s in the code word is **odd.**

Example: The word 1100 will be send with odd parity as 11001, 1000 as 10000

* **Limitations**
* It is **not suitable** for **detecting multiple errors**.
* It can **not correct errors** because it can not reveal the location of erroneous bit.



* **Two Dimensional Parity Checking**
* In this method, the data word is organised in a table (row & column) format.
* It catches all 1, 2, 3 bit errors and most 4 bit errors, but not all.
* **Example-1 of even parity:**

**Parity bit**

**Original Data**

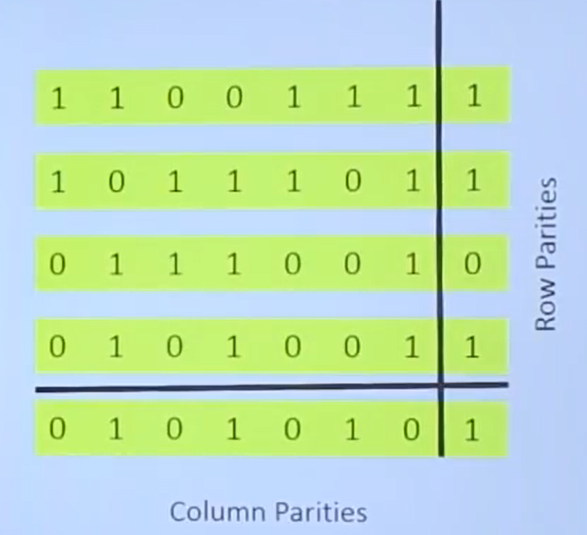
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **0** | **1** | **1** | **0** | **1** | **0** | **0** |
| **0** | **1** | **1** | **0** | **0** | **0** | **1** | **1** |
| **1** | **0** | **1** | **0** | **1** | **1** | **1** | **1**  **LRC** (longitudinal Redundant Check) or Row Parities |
| **1** | **1** | **1** | **0** | **0** | **0** | **1** | **0** |
| **1** | **0** | **0** | **1** | **0** | **0** | **1** | **1**  **Parity bit** |
| **0** | **0** | **0** | **0** | **1** | **0** | **0** | **1** |

**VRC** (Vertical Redundant Check) or Column Parities

Codeword to be sent= Original data+parity bit

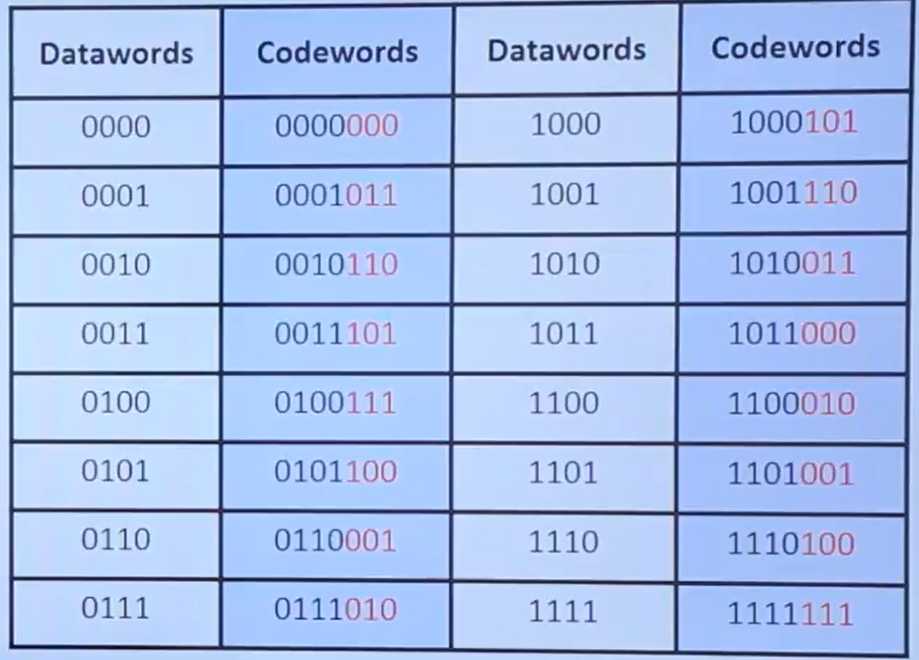
This is used by BISYNC protocol for ASCII characters.

* **Example-2 of even parity:**

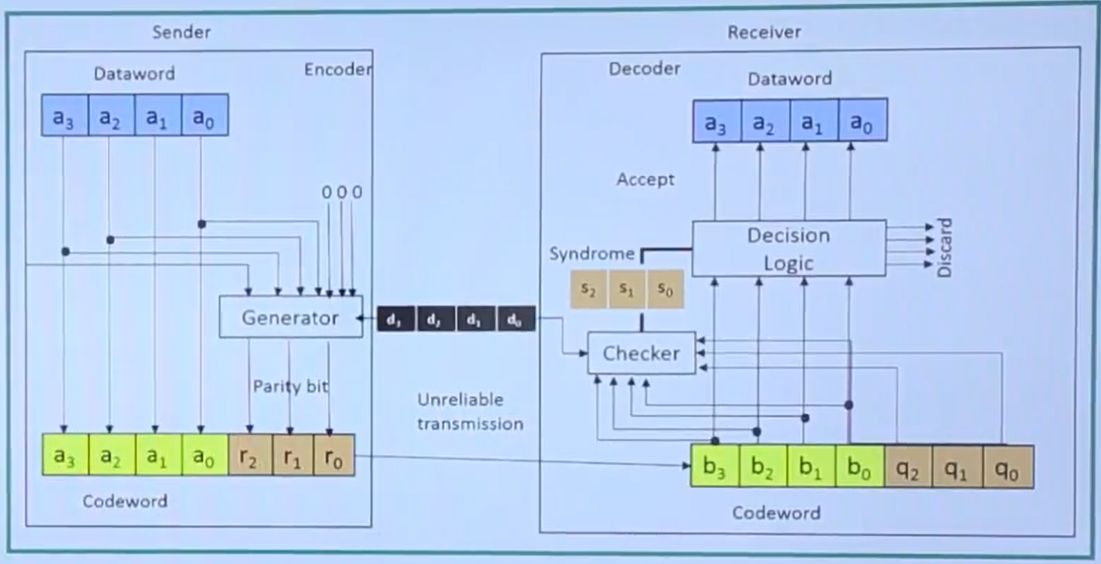


(Deriving Row and Column Parities)

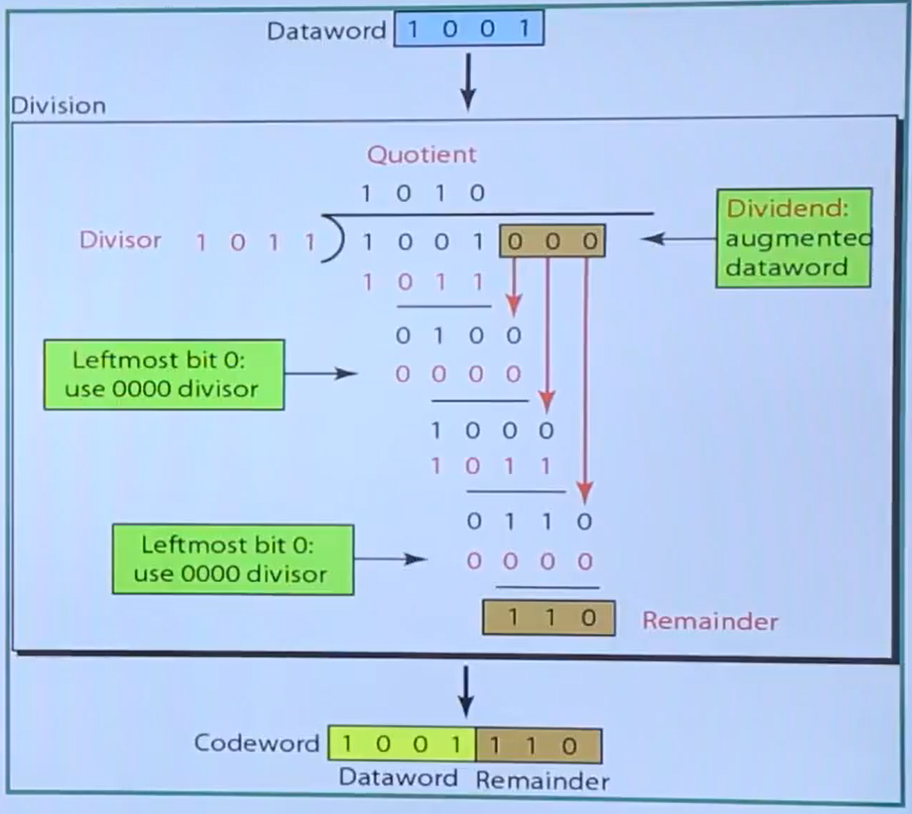
* 1. **Cyclic Redundancy Check (CRC)**
* **Cyclic Codes** are special linear block codes with one extra property. In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword. As for example, if 1100110 is a codeword and we cyclically left-shift, then 1001101 is also a codeword.
* A CRC Code with C(7, 4) means where dataword length is 4 and codeword length is 7 that includes 3 redundant bits.



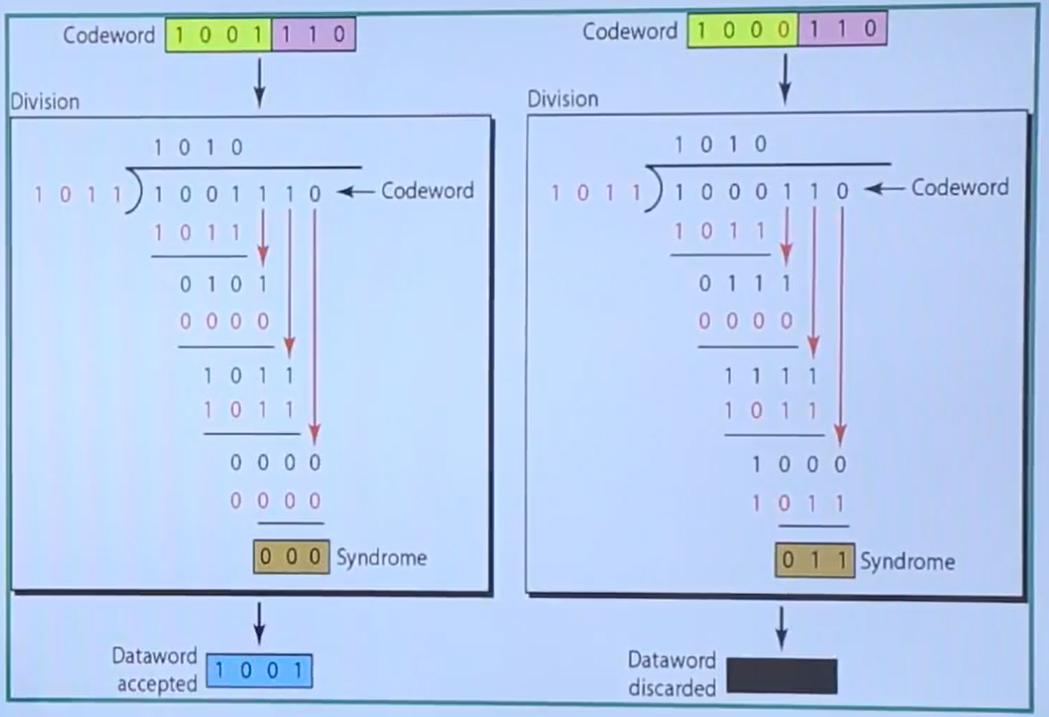
* **CRC Encoder and Decoder**



* **Division in CRC encoder**



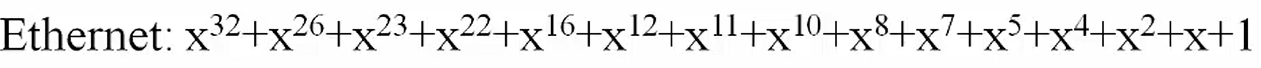
* **Use:**This is used by link level protocols: HDLC, DDCMP, Ethernet, Token Ring. It uses powerful math based on finite fields.
* **Division in CRC decoder for two cases**



* **Polynomial Arithmetic**
* Represents a m bit message with a polynomial of degree m-1. As for example, 11000101 = x7 + x6 + x2 + 1
* Arithmatic over the field of integers modulo 2 (coefficients 1 or 0)
* Addition and Subtractions are identical: XOR operation
* Polynomial division is very similar to integer division

X/Y is X=q\*Y+r where 0≤r<Y

* For polynomials: degree of r (remainder of polynomial) is less than divisor polynomial.
* **CRC Polynomial**
* Message Polynomial M(x): m bit message represented with a polynomial of degree m-1. Example: 11000101 = x7 + x6 + x2 + 1
* Sender and receiver agree on a divisor polynomial C(x) of degree k. k is the number of redundant bit. Example: C(x) = x3 + x2 + 1 (degree=3)
* Choice of C(x) significantly effects error detection and is derived carefully based on observed error patterns
* Ethernet uses CRC of 32 bits, HDLC, DDCMP uses 16 bits.

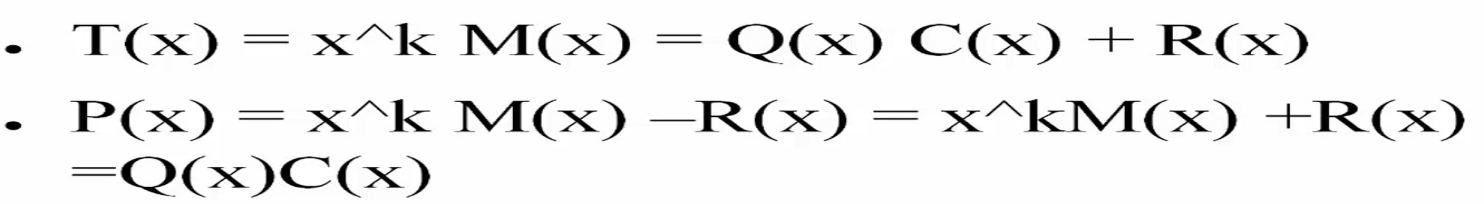


* **Idea:** Sender sends m+k bits. => Transmitted message P(x).
* Contrive P(x) is exactly divisible by C(x).
* Received message Q(x)

**No Error: Q**(x)=P(x) => Q(x) exactly divisible by C(x)

**Error: Q**(x)≠P(x) => Q(x) is likely not divisible by C(x)

* **Generate P(x):** You have M(x) and C(x). Multiply M(x) by xk to get T(x) (Add k zeroes at the end of the M(x)). Divide T(x) by C(x) to get reaminder R(x). Subtract R(x) from T(x) to get P(x). P(x) is now divisible by C(x).
* **Details**



* Where, Coefficients of R(x) are redundant bits. Transmitted bits: Message m bits followed by k redundant bits.
* **Example of CRC**
* Message M=11001011
* Divisor C=1101
* T=11001011000
* Remainder R=101
* Transmitted Message P=M followed by R=11001011101
  1. **Checksum for Error Detection**
* In this each word is added to the previous word and total sum (called checksum) is calculated. Then the checksum is calculated along with the data.
* Example

Let

1. **Framing**
   1. **Introduction**

* The data link layer as well as physical layer does next-hop del

1. **Framing**
   1. **Introduction**

* The data link layer as well as physical layer does next-hop delivery.