DATA LINK LAYER

ECE 422-Data Communication & Computer Networks

Monday, 23 February 2026

WE ARE HERE THE SYLLABUS…

Course Content:

Introduction: Overview of Data Communications and Networking.

Physical Layer: Analog and Digital, Analog Signals, Digital Signals, Analog versus Digital, Data Rate Limits, Transmission Impairment, More about signals.

Digital Transmission: Line coding, Block coding, Sampling, Transmission mode.

Analog Transmission: Modulation of Digital Data; Telephone modems, modulation of Analog signals.

Multiplexing: FDM, WDM, TDM.

Transmission Media: Guided Media, Unguided media (wireless).

Data Link Layer: Error Detection and correction - Types of Errors, Detection, Error Correction; Data Link Control and Protocols-Flow and Error Control, Stop-and-wait ARQ. Go-Back-N ARQ, Selective Repeat ARQ, HDLC. Point-to-Point Access- Point–to-Point Protocol (PPP), PPP Stack, Multiple Access Random Access, Controlled Access, Channelization.

Network Layer: Host to Host Delivery: Internetworking, addressing and Routing Network Layer Protocols: ARP, IPV4, ICMP, IPV6 and ICMPV6

Transport Layer: Process to Process Delivery: UDP; TCP congestion control and Quality of service.

Application Layer: Client Server Model, Socket Interface, Domain Name System (DNS): Electronic Mail (SMTP) and file transfer (FTP) HTTP and WWW.

Local area Network: Ethernet - Traditional Ethernet, Fast Ethernet, Gigabit Ethernet; Token bus, token ring; Wireless LANs - IEEE 802.11, Bluetooth virtual circuits: Frame Relay and ATM.

Industrial Communication and Control Networks: Transmission methods, Network topology, Contemporary networks – Profibus, Controller Area Network (CAN), DeviceNet, CANopen, Actuator Sensor Interface (AS-1),Industrial Ethernet.

FUNCTIONS OF DATA LINK LAYER

The functions of the data link layers are:

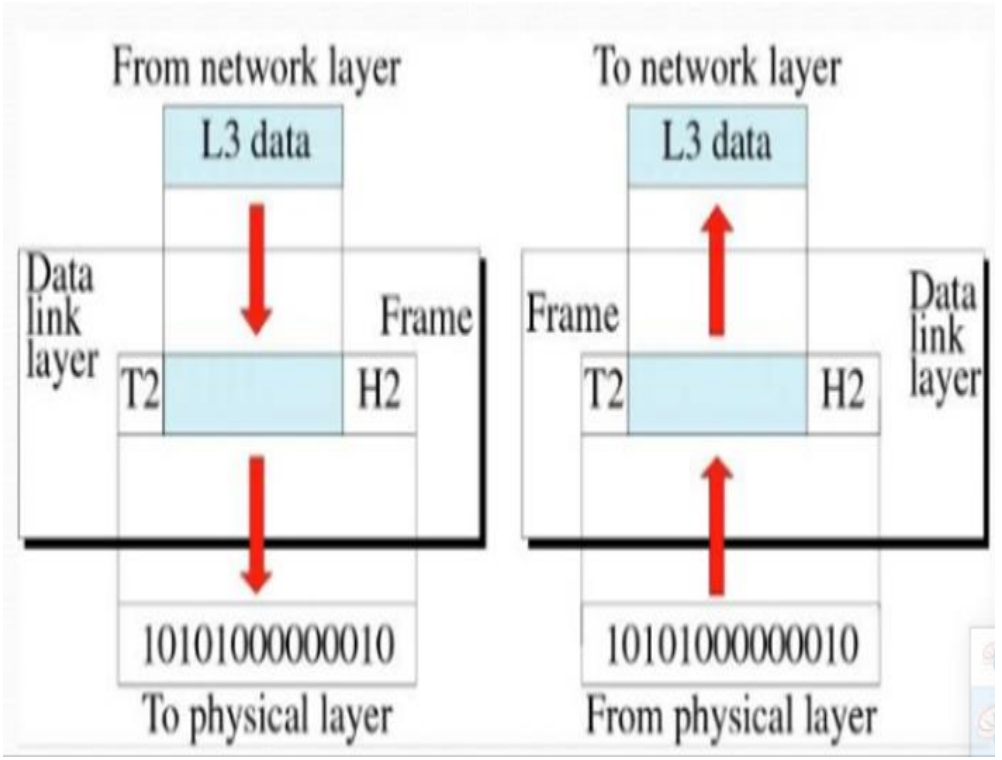
1. **Framing: T**he process of creating units of data called a frame for transmision over the data link layer. A frame consists of a header, payload data and trailer.

2. **Error control and correction:** the process of detecting or identifying and re-transmitting data frames that might be lost or corrupted during transmission.

3. **Addressing:** The process of assigning physical address, logical address, port address and application address. **Data link layer is concerned with the first two.**

4. **Flow control:** The process of managing the rate of data transmission between two nodes to prevent a fast sender from overwhelming a slow receiver.

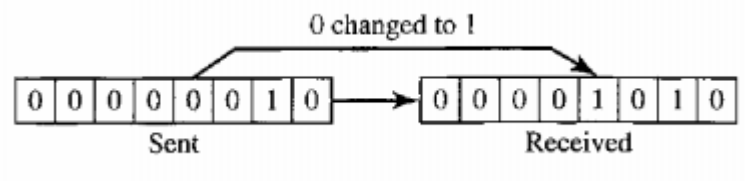
5. **Media access control**:A network data transfer policy that determines how data is transmitted between two computer terminals through the physical layer.



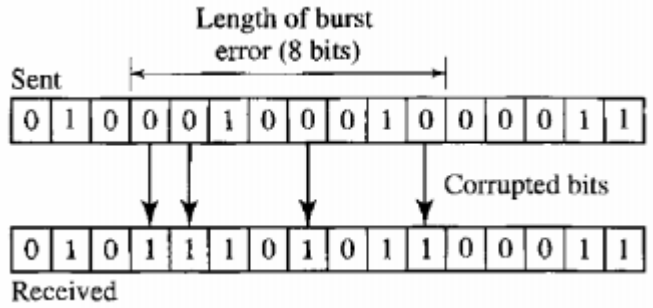
TYPES OF ERRORS

There are basically two types of errors in data communication:

1. **Single-bit errors:**
   a) Only one-bit changes
   b) Very rare type of errors since noise duration is usually longer than the duration of one bit.
2. **Burst Errors:**
   a) More than one bit changes
   b) More likely type of error since noise duration is usually longer than the duration of one bit.
   c) Number of affected bits depends on duration of noise and the bit rate of the data.



(a) Single-bit error



(b) Burst Error

REDUNDANCY

1. **Redundancy** refers to the addition of extra bits to the original data stream in order to facilitate error detection and correction.
2. Redundant information includes:
   a) Parity bits
   b) Cyclic Redundancy Check (CRC) blocks

FORWARD ERROR CORRECTION VS RETRANSMISSION

1. **Forward error correction:** The receiver tries to guess the message by using redundant bits. This is possible if the number of bits in error is small.

2. **Correction by retransmission:** The receiver detects the occurrence of an error and asks the sender to resend the message.
   - Retransmission is usually r**epeated until a message arrives that the receiver believes is error-free or timeout occurs.**

BENEFITS OF FORWARD ERROR CORRECTION (FEC)

1. Benefits or advantages of Forward Error Correction (FEC):

a) **It offers high degree of fault tolerance**. FEC decreases the Bit Error Rate (BER).

b) **It eliminates requirement of back channel.** Hence ARQ can be avoided

c) **It can be implemented using simple logic.**

d) **It is cost efficient technique** and can be implemented in software and hardware.

e) **It delivers fast results** based on its algorithm.

f) **FEC code can function in Realtime** to detect errors and correct them

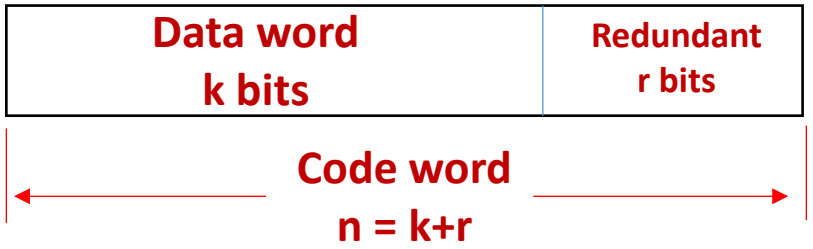
2. Drawbacks or disadvantages of Forward Error Correction (FEC):

1. It adds data redundancy or overhead to link data.
2. It is not bandwidth efficient due to overhead usage of the data.

BLOCK CODING

1. Block coding refers to the process of dividing the message into blocks of *k* bits, called **data words**.

2. *r* redundant bits are added to each block to make the length $n = k + r$. The resulting *n-bit* blocks are called **code words**.
3. The block coding process is one-to-one;.
4. This means that we have $2^n - 2^k$ codewords that are not used.
5. Unused codewords are called invalid or illegal.

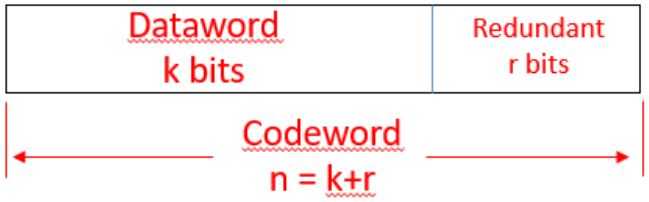


EXAMPLE OF BLOCK CODING – 4B/5B

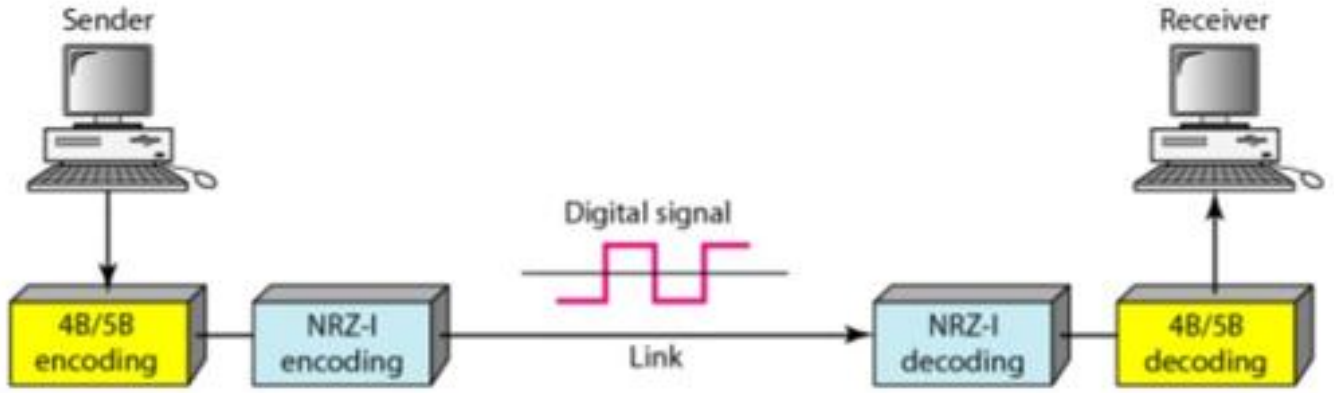
1. 4B/5B maps groups of four bits onto groups of 5 bits, with a minimum density of 1 bit in the output.
2. When NRZI-encoded, the 1 bits provide necessary clock transitions for the receiver.
3. Example:
   a) a run of 4 bits such as 0000 contains no transitions and that causes clocking problems for the receiver.
   b) 4B/5B solves this problem by assigning each block of 4 consecutive bits an equivalent word of 5 bits.
   c) The 5 bit words are chosen to ensure that there will be at least two transitions per block of bits.
   d) In this coding scheme, k =4 and n =5.
   e) There are 2^k =16 datawords and 2^n =32 codewords.
   f) As a result only 16 out of 32 codewords are used for message transfer and the rest are redundant (invalid/illegal).

Data Sequence	*Encoded Sequence*
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11 010
1101	11011
1110	11100
1111	11101

Example of 4B/5B Coding



USING BLOCK CODING 4B/5B WITH NRZ-I LINE CODING SCHEME



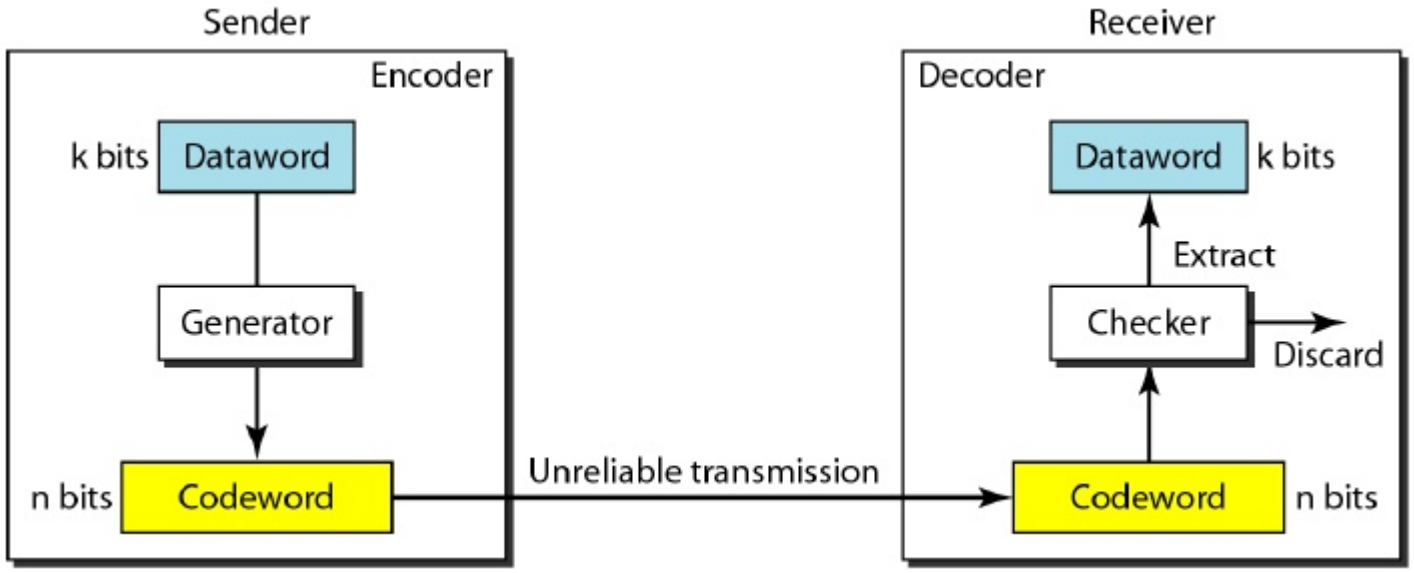
Data Sequence	*Encoded Sequence*
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11 010
1101	11011
1110	11100
1111	11101

ERROR DETECTION IN BLOCK CODING

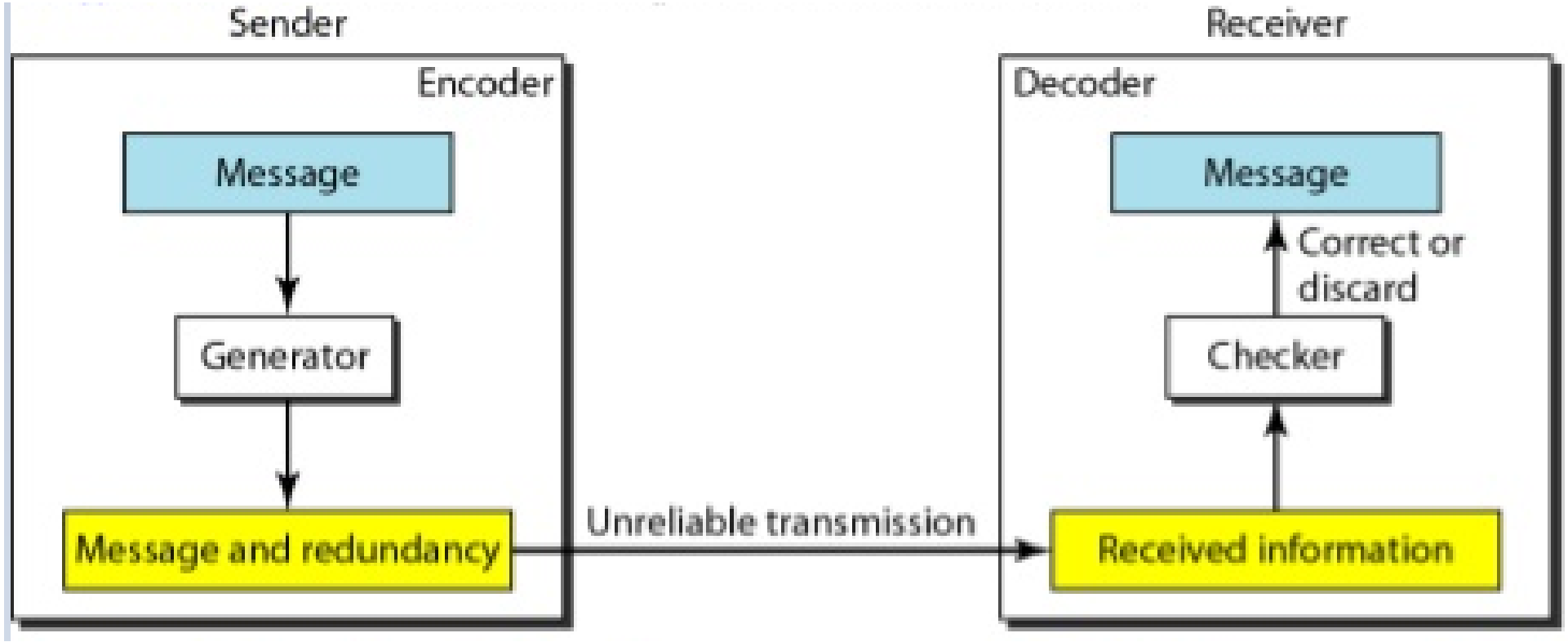
The receiver can detect a change in the original codeword if the following conditions are met:

1. The receiver has a list of valid codewords.

2. The original codeword has changed to an invalid one.

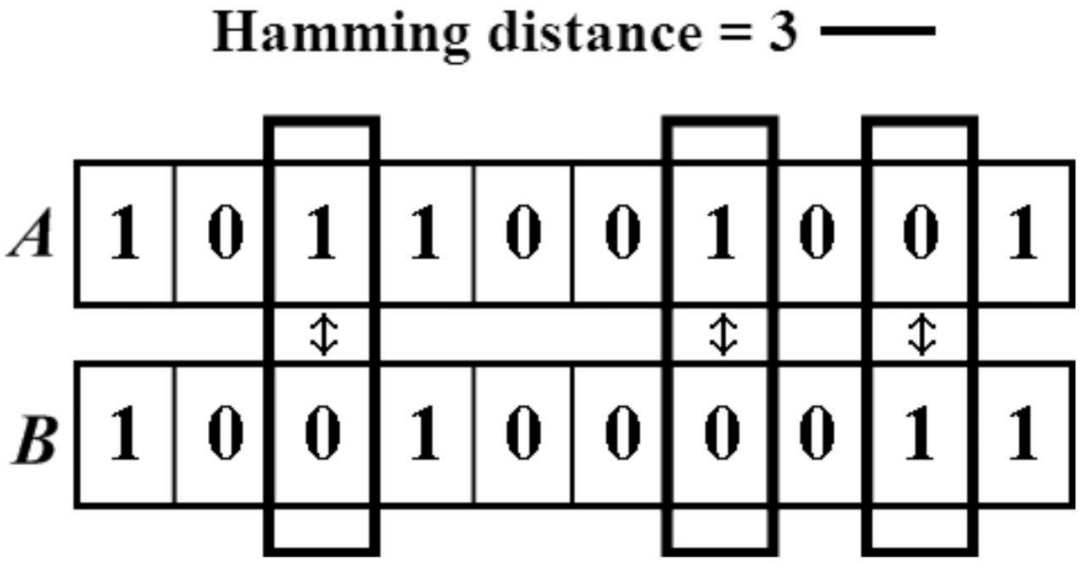


USING BLOCK CODING FOR ERROR CORRECTION



HAMMING DISTANCE

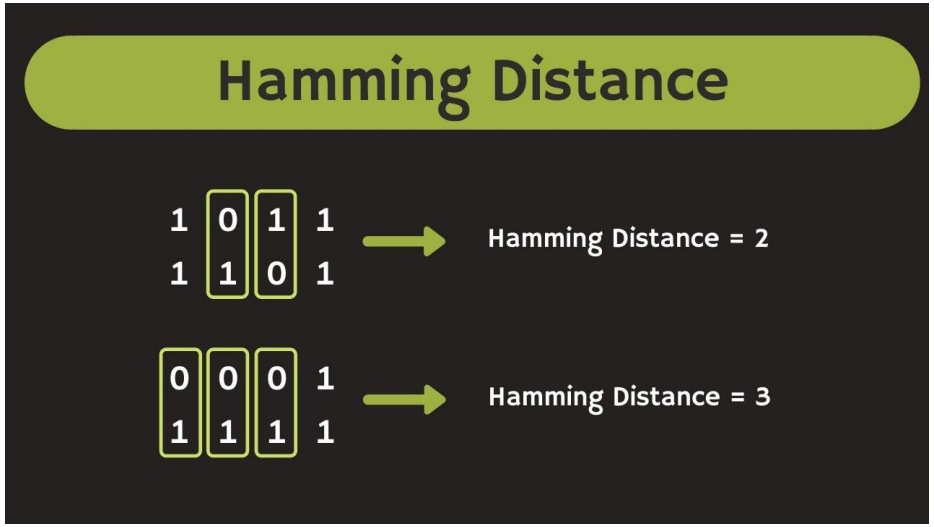
1. **Hamming distance** between two words (of the same size) is the number of differences between the corresponding bits.
2. Hamming distance between two words *x* and *y* is denoted as *d(x, y).*
3. The Hamming distance can easily be found by applying the XOR operation on the two words and counting the number of Is in the result.
4. **Hamming Distance** measures the minimum number of *substitutions* required to change one codeword into the other, or the minimum number of *errors* that could have transformed one codeword into another valid code word.



HAMMING DISTANCE: ORIGIN & APPLICATION

1. **Hamming distance** is named after Richard Hamming, who introduced Hamming codes Error detecting and error correcting codes in 1950.

2. **Hamming distance** is used in data communication to estimate error by counting the number of flipped bits in a fixed-length binary word.



C FUNCTION FOR CALCULATING HAMMING DISTANCE

```
int hamming_distance(unsigned x, unsigned y)
{
    int      dist;
    unsigned  val;
    dist = 0;
    val = x ^ y;    // XOR
    // Count the number of bits set
    while (val != 0)
    {
        // A bit is set, so increment the count and clear the bit
        dist++;
        val &= val - 1;
    }
    // Return the number of differing bits
    return dist;
}
```

MINIMUM HAMMING DISTANCE

The minimum Hamming distance is the smallest Hamming distance between all possible pairs in a set of code words.

HAMMING DISTANCE - EXAMPLE

Find the Minimum Hamming distance for the code below.

Datawords	*Codewords*
00	000
01	011
10	101
11	110

SOLUTION

The Hamming distances are:

d(000,011) = 2 d(000,101) = 2 d(000,110) = 2

d(011,101) = 2 d(011,110) = 2

d(101,110) = 2

The minimum Hamming distance is therefore, $d_{min} = 2$

CODE DESCRIPTION-PARAMETERS

1. Any coding scheme needs to have at least three parameters:
    a) the codeword size *n*,
    b) the dataword size *k*, and
    c) the minimum Hamming distance d_{min}

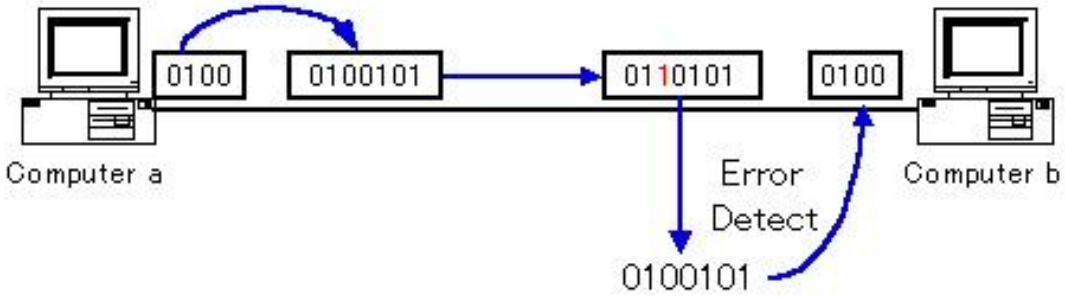
2. A coding scheme C is therefore written as *C(n, k)*, d_{min}

Datawords	*Codewords*
00	000
01	011
10	101
11	110

Code descriptor: C(3,2), 2

HAMMING DISTANCE & ERROR DETECTION

1. When a **codeword is corrupted** during transmission, the Hamming distance between the sent and received codewords is the number of bits affected by the error.
2. **The number of bits that are corrupted during transmission** is therefore equal to the Hamming distance between the received codeword and the sent codeword.



MINIMUM DISTANCE FOR ERROR DETECTION

1. **If s bit errors** occur during transmission, the **Hamming distance** between the sent codeword and received codeword is **s**.

2. Therefore,if our code is to detect up to s errors, the **minimum hamming distance** between the valid codes must be **s + 1** to ensure that no erroneous codeword matches a valid codeword.

WORKED EXAMPLE

- How many errors are guaranteed to be corrected by the code below?

Dataword	*Codeword*
00	00000
01	01011
10	10101
11	11110

SOLUTION

D(00000,01011) = 3; d(00000,10101)=3; d(00000,11110)=4

D(01011,10101)=4; d(01011,11110)=3

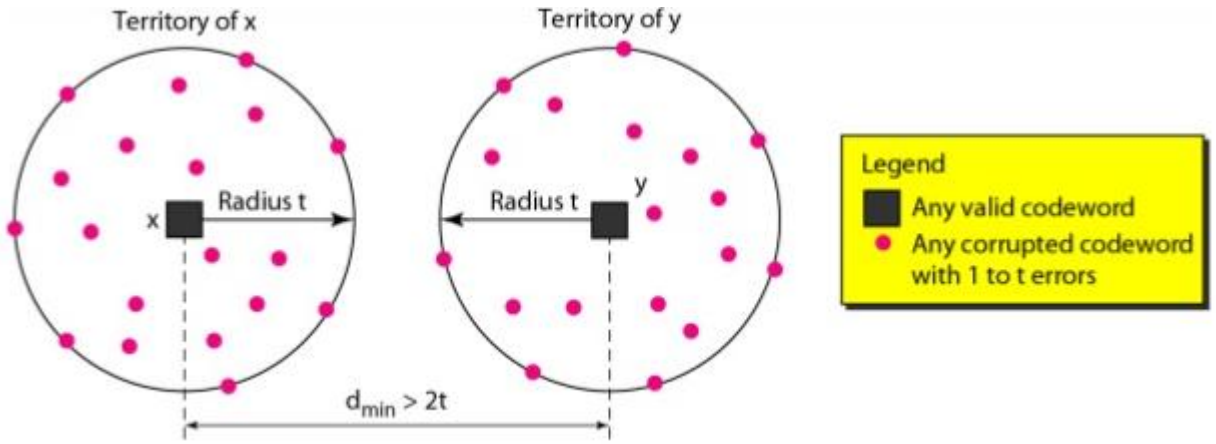
D(10101,11110) = 3

Therefore $d_{min} = 3$

The code can therefore correct up to $d_{min} - 1$ or 3-1 = 2 errors.

MINIMUM DISTANCE FOR ERROR CORRECTION

1. When a receiver gets an invalid codeword, the receiver needs to decide which valid codeword was actually sent.
2. The decision is based on the concept of territory, an exclusive area surrounding the code word.
3. Each valid code word has its own territory as shown.



To guarantee correction of up to *t* errors in all cases, the minimum Hamming distance in a block code must be $d_{min} = 2t + 1$.

EXAMPLE

Q1. A code scheme has a Hamming distance $d_{min} = 6$.

What is the error detection and correction capability of this scheme?

SOLUTION

The code scheme can detect up to $d_{min} - 1 = 5$ errors

Since $d_{min} = 2t + 1$, the scheme can correct $t = \frac{d_{min}-1}{2} = 2$ errors.

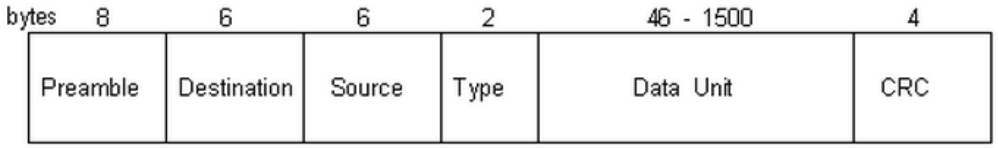
DATA LINK LAYER
FRAMING

ECE 422 – DATA COMMUNICATION & COMPUTER NETWORKS

Monday, 23 February 2026

PURPOSE OF FRAMING

1. **Framing in the data link layer** separates a message from one source to a destination from other messages to other destinations, by adding a sender address and a destination address.
2. Although the whole message (from network layer) could be packed in one frame, that is not normally done.
3. The reason is that when a frame is very large, flow and error control are very inefficient.
   a) Errors in transmission would require the retransmission of the whole message.
   b) smaller frames ensure that a few bit errors affects one frame.

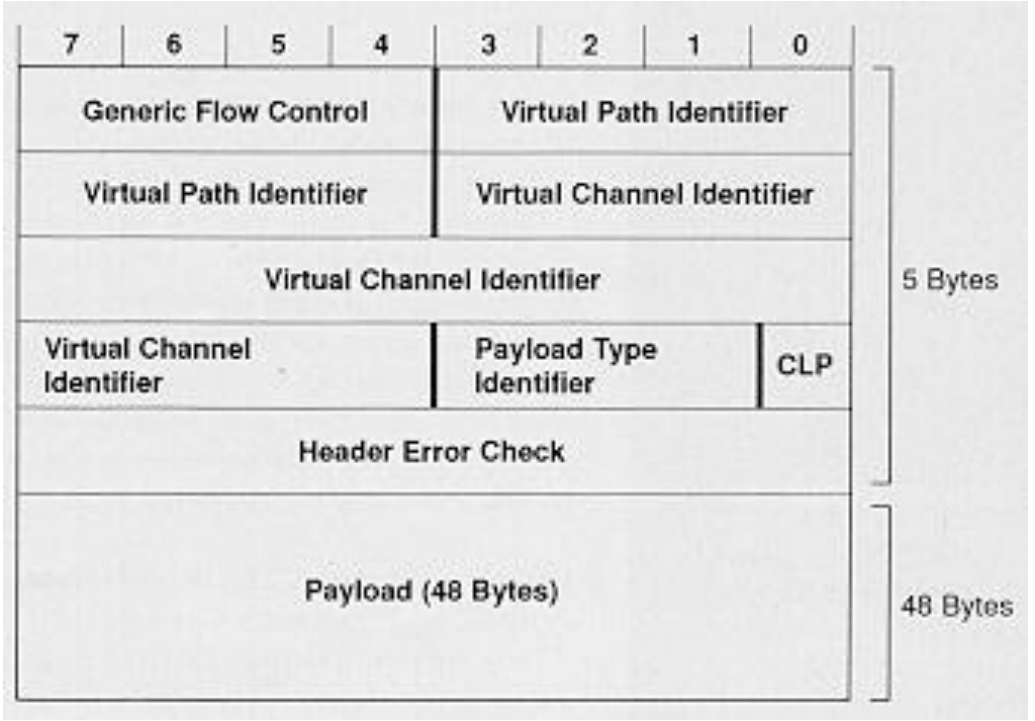


Original/DIX Ethernet Format

FIXED-SIZE FRAMING

1. In **fixed-size framing**, the size itself is a delimiter and there is no need for defining the boundaries of the frames;.

2. An example of fixed framing is the Asynchronous Transfer Mode (ATM) wide-area network, which uses frames of fixed size called **cells**.



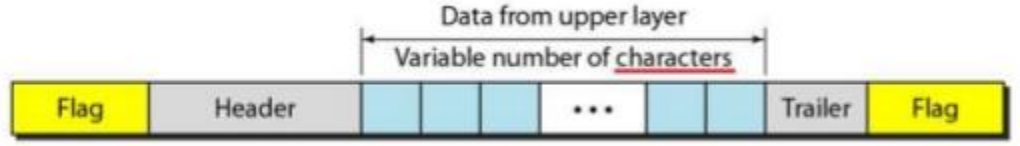
ATM Frame

VARIABLE-SIZE FRAMING

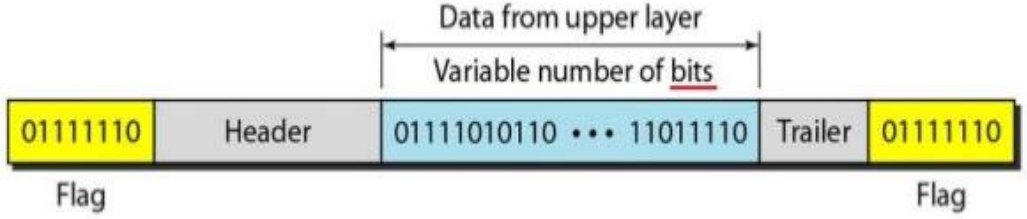
1. In variable-size framing, we need a way to define the beginning and end of each frame.

2. Historically, two approaches were used for this purpose:
   a) character-oriented approach, and
   b) bit-oriented approach.

3. Variable-Size Framing is prevalent in local area networks.



(a) Character-oriented framing



(b) Bit-oriented framing

CHARACTER-ORIENTED

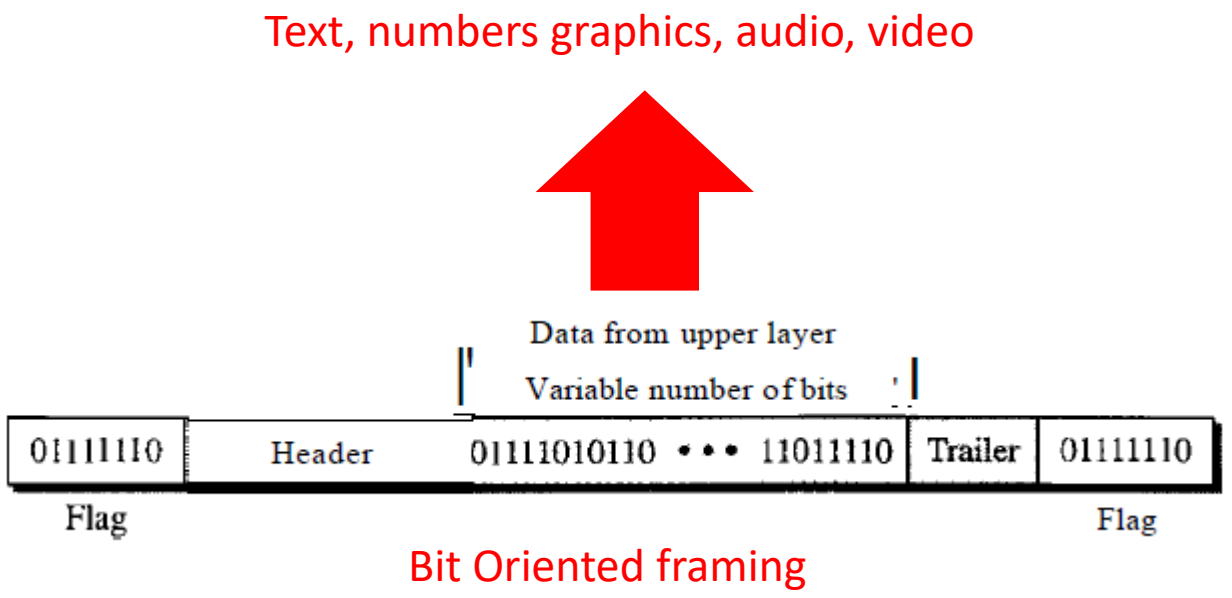
1. **Character-oriented framing** was popular when only text was exchanged by the data link layer.
2. The flag could be selected to be any character not used for text communication.
3. However, contemporary computer systems send other types of information such as graphs, audio, and video hence the preference for bit-oriented framing.

BIT-ORIENTED PROTOCOL

1. **Bit-oriented protocol**, the data section of a frame is a sequence of bits to be interpreted by the upper layers as text, graphic, audio, video,etc.

2. In addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other.

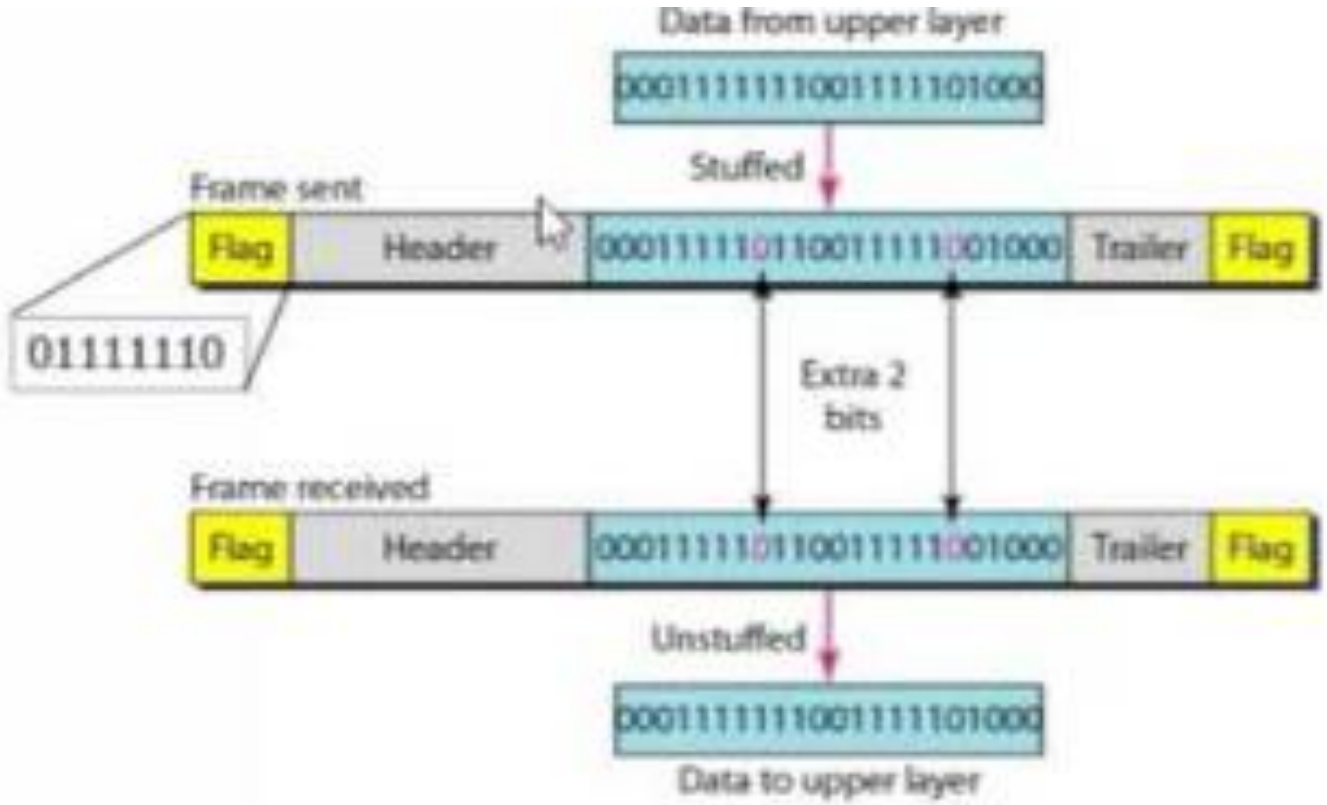
3. Most protocols use a special 8-bit pattern flag 01111110 (7E Hex) as the delimiter to define the beginning and the end of the frame.



Bit Oriented framing

BIT STUFFING DURING FRAMING

- **Bit stuffing during framing** is the process of adding one extra 0 whenever five consecutive 1's follow a 0 in the data, so that the receiver does not mistake the pattern **0111110** for a flag.



DIFFERENCE BETWEEN FLOW CONTROL & ERROR CONTROL

1. **Flow control** refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.

2. **Error control** allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender.