PARSHWANATH CHARITABLE TRUST'S
A.P. SHAH INSTITUTE OF TECHNOLOGY
Department of Computer Science and Engineering
Data Science

CSE DATA SCIENCE

Semester: <u>V</u>   Subject: <u>Computer Network</u>   Academic Year: <u>2023-24</u>

Module -2

Data Link Layer: DLL Design Issues - Services, Framing

DATA LINK LAYER FUNCTIONS (SERVICES)

1. **Providing services to the network layer:**

   1 <u>Unacknowledged connectionless service</u>.

   Appropriate for low error rate and real-time traffic. Ex: Ethernet

   2. <u>Acknowledged connectionless service</u>.

   Useful in unreliable channels, WiFi. Ack/Timer/Resend

   3. <u>Acknowledged connection-oriented service</u>.

   Guarantee frames are received exactly once and in the right order. Appropriate over long, unreliable links such as a satellite channel or a long-distance telephone circuit

2. **Framing:** Frames are the streams of bits received from the network layer into manageable data units. This division of stream of bits is done by Data Link Layer.

3. **Physical Addressing**: The Data Link layer adds a header to the frame in order to define physical address of the sender or receiver of the frame, if the frames are to be distributed to different systems on the network.

4. **Flow Control:** A receiving node can receive the frames at a faster rate than it can process the frame. Without flow control, the receiver's buffer can overflow, and frames can get lost. To overcome this problem, the data link layer uses the flow control to prevent the sending node on one side of the link from overwhelming the receiving node on another side of the link. This prevents traffic jam at the receiver side.

5. **Error Control:** Error control is achieved by adding a trailer at the end of the frame. Duplication of frames are also prevented by using this mechanism. Data Link Layers adds mechanism to prevent duplication of frames.

   Error detection: Errors can be introduced by signal attenuation and noise. Data Link Layer protocol provides a mechanism to detect one or more errors. This is achieved by adding error detection bits in the frame and then receiving node can perform an error check.

   Error correction: Error correction is similar to the Error detection, except that receiving node not only detects the errors but also determine where the errors have occurred in the frame.

6. **Access Control**: Protocols of this layer determine which of the devices has control over the link at any given time, when two or more devices are connected to the same link.

7. **Reliable delivery**: Data Link Layer provides a reliable delivery service, i.e., transmits the network layer datagram without any error. A reliable delivery service is accomplished with transmissions and acknowledgements. A data link layer mainly provides the reliable delivery
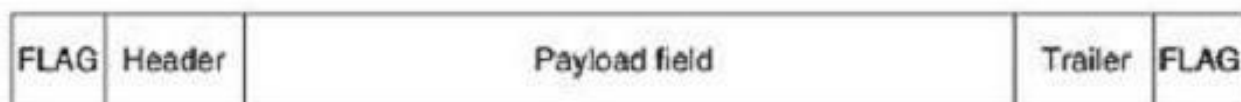
service over the links as they have higher error rates and they can be corrected locally, link at which an error occurs rather than forcing to retransmit the data.

8. **Half-Duplex & Full-Duplex**: In a Full-Duplex mode, both the nodes can transmit the data at the same time. In a Half-Duplex mode, only one node can transmit the data at the same time.

FRAMING:

To provide service to the network layer, the data link layer must use the service provided to it by the physical layer. What the physical layer does is accept a raw bit stream and attempt to deliver it to the destination. This bit stream is not guaranteed to be error free. The number of bits received may be less than, equal to, or more than the number of bits transmitted, and they may have different values. It is up to the data link layer to **detect and, if necessary, correct errors**. The usual approach is for the data link layer to break the bit stream up into discrete frames and compute the checksum for each frame **(framing)**. When a frame arrives at the destination, the checksum is recomputed. If the newly computed checksum is different from the one contained in the frame, the data link layer knows that an error has occurred and takes steps to deal with it (e.g., discarding the bad frame and possibly also sending back an error report).We will look at four framing methods:

1. Character count.
2. Flag bytes with byte stuffing.
3. Starting and ending flags, with bit stuffing.
4. Physical layer coding violations.

Character count method uses a field in the header to specify the number of characters in the frame. When the data link layer at the destination sees the character count, it knows how many characters follow and hence where the end of the frame is. This technique is shown in Fig. (a) For four frames of sizes 5, 5, 8, and 8 characters, respectively.

Fig: A character stream. (a) Without errors. (b) With one error

PARSHWANATH CHARITABLE TRUST'S
A.P. SHAH INSTITUTE OF TECHNOLOGY
Department of Computer Science and Engineering
Data Science

CSE DATA SCIENCE

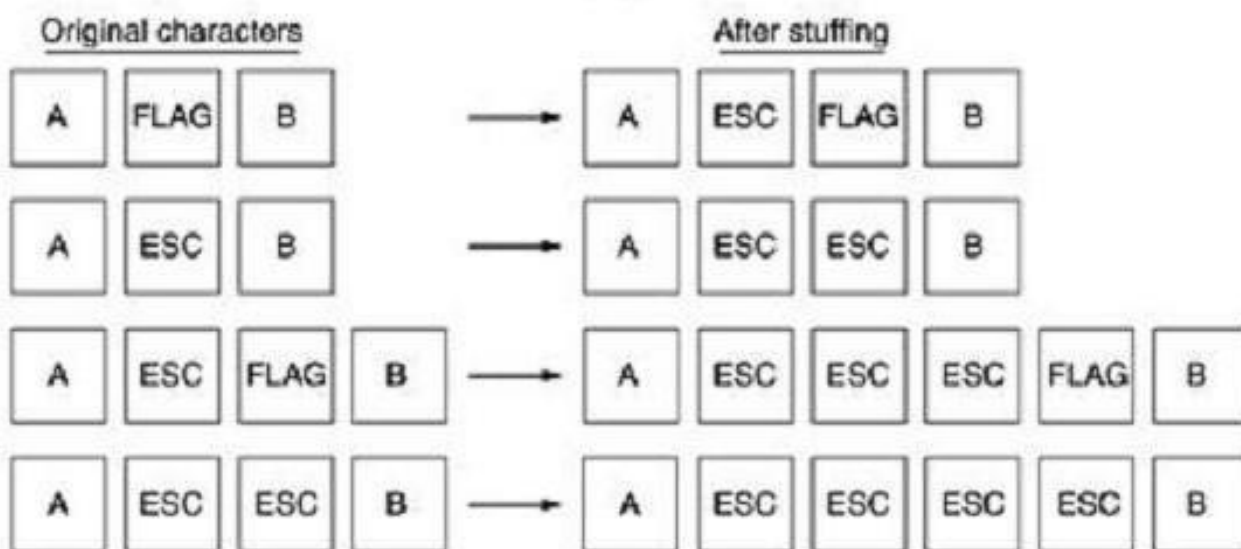**Semester: V**   **Subject: Computer Network**   **Academic Year: 2023-24**

The trouble with this algorithm is that the count can be garbled by a transmission error. For example, if the character count of 5 in the second frame of Fig. (b) becomes a 7, the destination will get out of synchronization and will be unable to locate the start of the next frame. Even if the checksum is incorrect so the destination knows that the frame is bad, it still has no way of telling where the next frame starts. Sending a frame back to the source asking for a retransmission does not help either, since the destination does not know how many characters to skip over to get to the start of the retransmission. For this reason, the character count method is rarely used anymore.

Flag bytes with byte stuffing method gets around the problem of resynchronization after an error by having each frame start and end with special bytes. In the past, the starting and ending bytes were different, but in recent years most protocols have used the same byte, called a flag byte, as both the starting and ending delimiter, as shown in Fig. (a) as FLAG. In this way, if the receiver ever loses synchronization, it can just search for the flag byte to find the end of the current frame. Two consecutive flag bytes indicate the end of one frame and start of the next one.

**Fig:** (a) A frame delimited by flag bytes (b) Four examples of byte sequences before and after byte stuffing

PARSHWANATH CHARITABLE TRUST'S

A.P. SHAH INSTITUTE OF TECHNOLOGY
Department of Computer Science and Engineering
Data Science

CSE DATA SCIENCE

| Semester: <u>V</u> | Subject: <u>Computer Network</u> | Academic Year: <u>2023-24</u> |

It may easily happen that the flag byte's bit pattern occurs in the data. This situation will usually interfere with the framing. One way to solve this problem is to have the sender's data link layer insert a special escape byte (ESC) just before each "accidental" flag byte in the data. The data link layer on the receiving end removes the escape byte before the data are given to the network layer. This technique is called byte stuffing or character stuffing.

Thus, a framing flag byte can be distinguished from one in the data by the absence or presence of an escape byte before it.

What happens if an escape byte occurs in the middle of the data? The answer is that, it too is stuffed with an escape byte. Thus, any single escape byte is part of an escape sequence, whereas a doubled one indicates that a single escape occurred naturally in the data. Some examples are shown in Fig. (b). In all cases, the byte sequence delivered after de stuffing is exactly the same as the original byte sequence.

A major disadvantage of using this framing method is that it is closely tied to the use of 8-bit characters. Not all character codes use 8-bit characters. For example UNICODE uses 16-bit characters, so a new technique had to be developed to allow arbitrary sized characters

Starting and ending flags, with bit stuffing allows data frames to contain an arbitrary number of bits and allows character codes with an arbitrary number of bits per character. It works like this. Each frame begins and ends with a special bit pattern, 01111110 (in fact, a flag byte). Whenever the sender's data link layer encounters five consecutive 1s in the data, it automatically stuffs a 0 bit into the outgoing bit stream. This bit stuffing is analogous to byte stuffing, in which an escape byte is stuffed into the outgoing character stream before a flag byte in the data.

When the receiver sees five consecutive incoming 1 bits, followed by a 0 bit, it automatically de- stuffs (i.e., deletes) the 0 bit. Just as byte stuffing is completely transparent to the network layer in both computers, so is bit stuffing. If the user data contain the flag pattern, 01111110, this flag is transmitted as 011111010 but stored in the receiver's memory as 01111110.

(a) 0110111111111111111110010

(b) 0110111110111110111110010010

Stuffed bits

(c) 0110111111111111111110010

Fig: Bit stuffing. (a) The original data. (b) The data as they appear on the line. (c) The data as they are stored in the receiver's memory after destuffing

Semester: V   **Subject: Computer Network**   **Academic Year: 2023-24**

Physical layer coding violations method of framing is only applicable to networks in which the encoding on the physical medium contains some redundancy. For example, some LANs encode 1 bit of data by using 2 physical bits. Normally, a 1 bit is a high-low pair and a 0 bit is a low-high pair. The scheme means that every data bit has a transition in the middle, making it easy for the receiver to locate the bit boundaries. The combinations high-high and low-low are not used for data but are used for delimiting frames in some protocols.
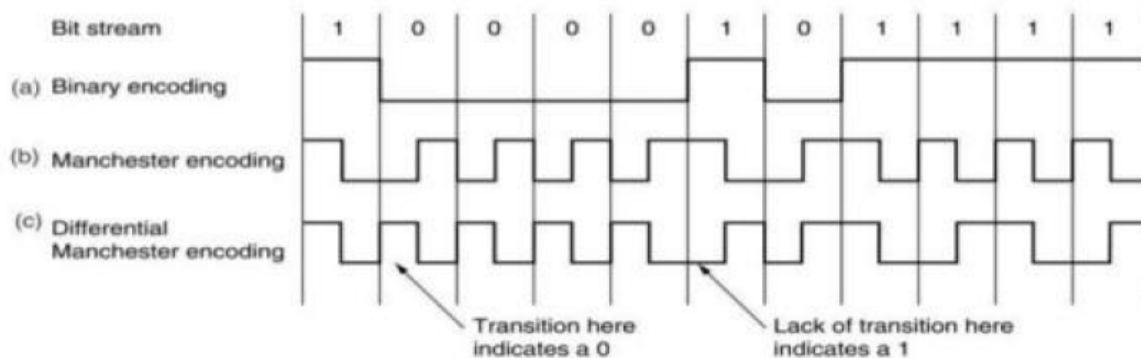
As a final note on framing, many data link protocols use combination of a character count with one of the other methods for extra safety. When a frame arrives, the count field is used to locate the end of the frame. Only if the appropriate delimiter is present at that position and the checksum is correct is the frame accepted as valid. Otherwise, the input stream is scanned for the next delimiter