**Error Detection & Correction**

1. **Why do you need error detection? \***

As the signal is transmitted through a media, the signal gets corrupted because of noise and distortion. In other words, the media is not reliable. To achieve a reliable communication through this unreliable media, there is need for detecting the error in the signal so that suitable mechanism can be devised to take corrective actions.

**2. Explain different types of Errors?**

The errors can be divided into two types: Single-bit error and Burst error.

• Single-bit Error The term single-bit error means that only one bit of given data unit (such as a byte, character, or data unit) is changed from 1 to 0 or from 0 to 1.

• Burst Error The term burst error means that two or more bits in the data unit have changed from 0 to 1 or vice-versa. Note that burst error doesn’t necessary means that error occurs in consecutive bits.

1. **Explain the use of parity check for error detection? \***

In the Parity Check error detection scheme, a parity bit is added to the end of a block of data. The value of the bit is selected so that the character has an even number of 1s (even parity) or an odd number of 1s (odd parity). For odd parity check, the receiver examines the received character and if the total number of 1s is odd, then it assumes that no error has occurred. If any one bit (or any odd number of bits) is erroneously inverted during transmission, then the receiver will detect an error.

1. **What are the different types of errors detected by parity check?**

If one bit (or odd number of bits) gets inverted during transmission, then parity check will detect an error. In other words, only odd numbers of errors are detected by parity check. But, if two (or even number) of bits get inverted, and then the error remains undetected.

1. **What kind of error is undetectable by the checksum?**

At least three types of error cannot be detected by the current checksum calculation. First, if two data items are swapped during transmission, the sum and the checksum values will not change. Second, if the value of one data item is increased (intentionally or maliciously) and the value of another one is decreased (intention-ally or maliciously) the same amount, the sum and the checksum cannot detect these changes. Third, if one or more data items is changed in such a way that the change is a multiple of 216 − 1, the sum or the checksum cannot detect the changes.

1. **Can the value of a checksum be all 0s (in binary)? Defend your answer. Can the value be all 1s (in binary)? Defend your answer.**

The value of a checksum can be all 0s (in binary). This happens when the value of the sum (after wrapping) becomes all 1s (in binary). It is almost impossible for the value of a checksum to be all 1s. For this to happen, the value of the sum (after wrapping) must be all 0s which means all data units must be 0s.

**Flow Control & Error Control**

1. **What are the key functions of error control techniques?**

There are basically two types of errors, namely, (a) Damaged Frame (b) Lost Frame. The key functions for error control techniques are as follows:

• Error detection

• Sending of positive acknowledgement (ACK) by the receiver for no error

• Sending of negative acknowledgement (NAK) by the receiver for error

• Setting of timer for lost frame

• Numbering of frames

1. **Why is flow control needed?**

In case of data communication between a sender and a receiver, it may so happen that the rate at which data is transmitted by a fast sender is not acceptable by a slow receiver. IN such a situation, there is a need of flow control so that a fast transmitter does not overwhelm a slow receiver.

1. **Mention key advantages and disadvantages of stop-and-wait ARQ technique? \***

Advantages of stop-and-wait ARQ are: a. Simple to implement b. Frame numbering is modulo-2, i.e. only 1 bit is required. The main disadvantage of stop-and-wait ARQ is that when the propagation delay is long, it is extremely inefficient.

1. **Consider the use of 10 K-bit size frames on a 10 Mbps satellite channel with 270 ms delay. What is the link utilization for stop-and-wait ARQ technique assuming P = 10-3? \***

Link utilization = (1-P) / (1+2a)

Where a = (Propagation Time) / (Transmission Time)

Propagation time = 270 msec

Transmission time = (frame length) / (data rate) = (10 K-bit) / (10 Mbps) = 1 msec

Hence, a = 270/1 = 270

Link utilization = 0.999/(1+2\*270) ≈0.0018 =0.18%

1. **What is the channel utilization for the go-back-N protocol with window size of 7 for the problem 3?\***

Channel utilization for go-back-N = N(1 – P) / (1 + 2a)(1-P+NP)

P = probability of single frame error ≈ 10-3

Channel utilization ≈ 0.01285 = 1.285%

1. **In what way selective-repeat is better than go-back-N ARQ technique?**

In selective-repeat scheme only the frame in error is retransmitted rather than transmitting all the subsequent frames. Hence it is more efficient than go-back-N ARQ technique.

1. **In what situation Stop-and-Wait protocol works efficiently?**

In case of Stop-and-Wait protocol, the transmitter after sending a frame waits for the acknowledgement from the receiver before sending the next frame. This protocol works efficiently for long frames, where propagation time is small compared to the transmission time of the frame.

1. **How the inefficiency of Stop-and-Wait protocol is overcome in sliding window protocol? \***

The Stop-and-Wait protocol is inefficient when large numbers of small packets are send by the transmitter since the transmitter has to wait for the acknowledgement of each individual packet before sending the next one. This problem can be overcome by sliding window protocol. In sliding window protocol multiple frames (up to a fixed number of frames) are send before receiving an acknowledgement from the receiver.

1. **What is piggybacking? What is its advantage? \***

In practice, the link between receiver and transmitter is full duplex and usually both transmitter and receiver stations send data to each other. So, instead of sending separate acknowledgement packets, a portion (few bits) of the data frames can be used for acknowledgement. This phenomenon is known as piggybacking. The piggybacking helps in better channel utilization. Further, multi-frame acknowledgement can be done. Piggybacking is used to improve the efficiency of bidirectional transmission. When a frame is carrying data from A to B, it can also carry control information about frames from B; when a frame is carrying data from B to A, it can also carry control information about frames from A.

1. **For a k-bit numbering scheme, what is the range of sequence numbers used in sliding window protocol?\***

For k-bit numbering scheme, the total number of frames, N, in the sliding window can be given as follows (using modulo-k). N = 2k – 1, Hence the range of sequence numbers is: 0, 1, 2, and 3 … 2k – 1

1. **Briefly describe the services provided by the data link layer.**

The two main functions of the data link layer are data link control and media access control. Data link control deals with the design and procedures for communication between two adjacent nodes: node-to-node communication. Media access control deals with procedures for sharing the link.

1. **What is bit-stuffing? Why is it used? \***

Bit-stuffing: In case of synchronous transmission, flag bits (8-bit length sequence 01111110) is attached at the beginning and end of each frame. These flag bits are used for synchronization. It may so happen that the flag bit sequence may appear somewhere inside the frame and this will cause a problem for synchronization. To avoid this problem a process called bit-stuffing is used, in which extra bit is stuffed if flag bit like sequence appears inside the frame. For example, in HDLC the transmitter introduces a 0 after each occurrence of five 1’s in the data sequence by. At the receiving end these extra 0’s is removed.

1. **Explain the reason for moving from the Stop-and-Wait ARQ Protocol to the Go-Back-N ARQ Protocol.** Go-Back-N ARQ is more efficient than Stop-and-Wait ARQ. The second uses pipelining, the first does not. In the first, we need to wait for an acknowledgment for each frame before sending the next one. In the second we can send several frames before receiving an acknowledgment.
2. **Compare and contrast the Go-Back-N ARQ Protocol with Selective-Repeat ARQ.**

In the Go-Back-N ARQ Protocol, we can send several frames before receiving acknowledgments. If a frame is lost or damaged, all outstanding frames sent before that frame are resent. In the Selective- Repeat ARQ protocol we avoid unnecessary transmission by sending only the frames that are corrupted or missing. Both Go-Back-N and Selective-Repeat Protocols use sliding windows. In Go-Back-N ARQ, if m is the number of bits for the sequence number, then the size of the send window must be at most 2m−1; the size of the receiver window is always 1. In Selective-Repeat ARQ, the size of the sender and receiver window must be at most 2m−1.

1. **Compare and contrast HDLC with PPP. Which one is byte-oriented; which one is bit-oriented?**

HDLC is a bit-oriented protocol for communication over point-to-point and multi-point links. PPP is a byte-oriented protocol used for point-to-point links.

1. **A system uses the Stop-and-Wait ARQ Protocol. If each packet carries 1000 bits of data, how long does it take to send 1 million bits of data if the distance between the sender and receiver is 5000 Km and the propagation speed is 2 x 108 m? Ignore transmission, waiting, and processing delays. We assume no data or control frame is lost or damaged.**

We need to send 1000 frames. We ignore the overhead due to the header and trailer.

We ignore the overhead due to the header and trailer.

Data frame Transmission time = 1000 bits / 1,000,000 bits = 1 ms

Data frame trip time = 5000 km / 200,000 km = 25 ms

ACK transmission time = 0 (It is usually negligible)

ACK trip time = 5000 km / 200,000 km = 25 ms

Delay for 1 frame = 1 + 25 + 25 = 51 ms. Total delay = 1000 × 51 = 51 s

1. **Repeat previous question using the Go-back-N ARQ Protocol with a window size of 7. Ignore the overhead due to the header and trailer.**

In the worst case, we send the full window of size 7 and then wait for the acknowledgment of the whole window. We need to send 1000/7 ≈ 143 windows. We ignore the overhead due to the header and trailer.

Transmission time for one window = 7000 bits / 1,000,000 bits = 7 ms

Data frame trip time = 5000 km / 200,000 km = 25 ms

ACK transmission time = 0 (It is usually negligible)

ACK trip time = 5000 km / 200,000 km = 25 ms

Delay for 1 window = 7 + 25 + 25 = 57 ms.

Total delay = 143 × 57 ms = 8.151 s.

1. **Repeat previous question using the Selective-Repeat ARQ Protocol with a window size of 4. Ignore the overhead due to the header and the trailer.**

In the worst case, we send the full window of size 4 and then wait for the acknowledgment of the whole window. We need to send 1000/4 = 250 windows. We ignore the overhead due to the header and trailer.

Transmission time for one window = 4000 bits / 1,000,000 bits = 4 ms

Data frame trip time = 5000 km / 200,000 km = 25 ms

ACK transmission time = 0 (It is usually negligible)

ACK trip time = 5000 km / 200,000 km = 25 ms

Delay for 1 window = 4 + 25 + 25 = 54 ms.

Total delay = 250 × 54 ms = 13.5 s

**19. Briefly write functionalities of different OSI layers?**

Ans: The OSI Reference Model includes seven layers. Basic functionality of each of them is as follows:

1. **Application Layer:** Provides Applications with access to network services.

6. **Presentation Layer:** Determines the format used to exchange data among networked computers.

5. **Session Layer:** Allows two applications to establish, use and disconnect a connection between them called a session. Provides for name recognition and additional functions like security, which are needed to allow applications to communicate over the network.

4**. Transport Layer:** Ensures that data is delivered error free, in sequence and with no loss, duplications or corruption. This layer also repackages data by assembling long messages into lots of smaller messages for sending, and repackaging the smaller messages into the original larger message at the receiving end.

3. **Network Layer:** This is responsible for addressing messages and data so they are sent to the correct destination, and for translating logical addresses and names (like a machine name FLAME) into physical addresses. This layer is also responsible for finding a path through the network to the destination computer.

2. **Data-Link Layer:** This layer takes the data frames or messages from the Network Layer and provides for their actual transmission. At the receiving computer, this layer receives the incoming data and sends it to the network layer for handling. The Data-Link Layer also provides error-free delivery of data between the two computers by using the physical layer. It does this by packaging the data from the Network Layer into a frame, which includes error detection information. At the receiving computer, the Data-Link Layer reads the incoming frame, and generates its own error detection information based on the received frames data. After receiving the entire frame, it then compares its error detection value with that of the incoming frames, and if they match, the frame has been received correctly.

1. **Physical Layer:** Controls the transmission of the actual data onto the network cable. It defines the electrical signals, line states and encoding of the data and the connector types used. An example is 10BaseT.

**20. What is the significance of the OSI model? (2)**

The Open System Interconnection (OSI) reference model describes how information from a software application in one computer moves through a network medium to a software application in another computer. The International Standards Organization created a model called the Open Systems Interconnection, which allows diverse systems to communicate, and it is now considered the primary architectural model for inter-computer communications. The OSI model is a layered framework for the design of network systems that allows communication between all types of computer systems. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network. The purpose of the OSI model is to show how to facilitate communication between different systems without requiring changes to the logic of the underlying hardware and software. The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and interoperable.

**21. We need to use synchronous TDM and combine 20 digital sources, each of 100 Kbps. Each output slot carries 2 bits from each digital source, but one extra bit is added to each frame for synchronization.**

**Answer the following questions: a. What is the size of an output frame in bits?**

**b. What is the output frame rate? c. What is the duration of an output frame?**

**d. What is the output data rate? e. What is the efficiency of the system (ratio of useful bits to the total bits).**

a. Each output frame carries 2 bits from each source plus one extra bit for synchronization. Frame size = 20 × 2 + 1 = 41 bits.

b. Each frame carries 2 bit from each source. Frame rate = 100,000/2 = 50,000frames/s.

c. Frame duration = 1 /(frame rate) = 1 /50,000 = 20 μs.

d. Data rate = (50,000 frames/s) × (41 bits/frame) = 2.05 Mbps.

e. In each frame 40 bits out of 41 are useful. Efficiency = 40/41= 97.5%.

**22. We have 14 sources, each creating 500 8-bit characters per second. Since only some of these sources are active at any moment, we use statistical TDM to combine these sources using character interleaving. Each frame carries 6 slots at a time, but we need to add four-bit addresses to each slot.**

**Answer the following questions: a. What is the size of an output frame in bits? b. What is the output frame rate? c. What is the duration of an output frame? d. What is the output data rate?**

a. Frame size = 6 × (8 + 4) = 72 bits.

b. We can assume that we have only 6 input lines. Each frame needs to carry one character from each of these lines. This means that the frame rate is 500frames/s.

c. Frame duration = 1 /(frame rate) = 1 /500 = 2 ms.

d. Data rate = (500 frames/s) × (72 bits/frame) = 36 kbps.

**23. Ten sources, six with a bit rate of 200 kbps and four with a bit rate of 400 kbps are to be combined using multilevel TDM with no synchronizing bits.**

**Answer the following questions about the final stage of the multiplexing: a. What is the size of a frame in bits? b. What is the frame rate? c. What is the duration of a frame? d. What is the data rate?**

We combine six 200-kbps sources into three 400-kbps. Now we have seven 400-

kbps channel.

a. Each output frame carries 1 bit from each of the seven 400-kbps line. Frame

size = 7 × 1 = 7 bits.

b. Each frame carries 1 bit from each 400-kbps source. Frame rate = 400,000

frames/s.

c. Frame duration = 1 /(frame rate) = 1 /400,000 = 2.5 μs.

d. Output data rate = (400,000 frames/s) × (7 bits/frame) = 2.8 Mbps. We can also

calculate the output data rate as the sum of input data rate because there is no

synchronizing bits. Output data rate = 6 × 200 + 4 × 400 = 2.8 Mbps.

We combine six 200-kbps sources into three 400-kbps. Now we have seven 400-

kbps channel.

a. Each output frame carries 1 bit from each of the seven 400-kbps line. Frame

size = 7 × 1 = 7 bits.

b. Each frame carries 1 bit from each 400-kbps source. Frame rate = 400,000

frames/s.

c. Frame duration = 1 /(frame rate) = 1 /400,000 = 2.5 μs.

d. Output data rate = (400,000 frames/s) × (7 bits/frame) = 2.8 Mbps. We can also

calculate the output data rate as the sum of input data rate because there is no

synchronizing bits. Output data rate = 6 × 200 + 4 × 400 = 2.8 Mbps.

We combine six 200-kbps sources into three 400-kbps. Now we have seven 400-kbps channel.

a. Each output frame carries 1 bit from each of the seven 400-kbps line. Frame size = 7 × 1 = 7 bits.

b. Each frame carries 1 bit from each 400-kbps source. Frame rate = 400,000frames/s.

c. Frame duration = 1 /(frame rate) = 1 /400,000 = 2.5 μs.

d. Output data rate = (400,000 frames/s)×(7 bits/frame) = 2.8 Mbps. We can also calculate the output data rate as the sum of input data rate because there are no synchronizing bits. Output data rate = 6 × 200 + 4 × 400 = 2.8 Mbps.

**24. Four channels, two with a bit rate of 200 kbps and two with a bit rate of 150 kbps, are to be multiplexed using multiple slot TDM with no synchronization bits.**

**Answer the following questions: a. What is the size of a frame in bits? b. What is the frame rate?**

**c. What is the duration of a frame? d. What is the data rate?**

a. The frame carries 4 bits from each of the first two sources and 3 bits from each

of the second two sources. Frame size = 4 × 2 + 3 × 2 = 14 bits.

b. Each frame carries 4 bit from each 200-kbps source or 3 bits from each 150

kbps. Frame rate = 200,000 / 4 = 150,000 /3 = 50,000 frames/s.

c. Frame duration = 1 /(frame rate) = 1 /50,000 = 20 μs.

d. Output data rate = (50,000 frames/s) × (14 bits/frame) = 700 kbps. We can also

calculate the output data rate as the sum of input data rates because there are no

synchronization bits. Output data rate = 2 × 200 + 2 × 150 = 700 kbps.

a. The frame carries 4 bits from each of the first two sources and 3 bits from each of the second two sources. Frame size = 4 × 2 + 3 × 2 = 14 bits.

b. Each frame carries 4 bits from each 200-kbps source or 3 bits from each 150kbps. Frame rate = 200,000 / 4 = 150,000 /3 = 50,000 frames/s.

c. Frame duration = 1 /(frame rate) = 1 /50,000 = 20 μs.

d. Output data rate = (50,000 frames/s)×(14 bits/frame) = 700 kbps. We can also calculate the output data rate as the sum of input data rates because there are no synchronization bits. Output data rate = 2 × 200 + 2 × 150 = 700 kbps.

**25. Two channels, one with a bit rate of 190 kbps and another with a bit rate of 180 kbps, are to be multiplexed using pulse stuffing TDM with no synchronization bits.**

**Answer the following questions: a. What is the size of a frame in bits? b. What is the frame rate? c. What is the duration of a frame? d. What is the data rate?**

We need to add extra bits to the second source to make both rates = 190 kbps. Now we have two sources, each of 190 Kbps.

a. The frame carries 1 bit from each source. Frame size = 1 + 1 = 2 bits.

b. Each frame carries 1 bit from each 190-kbps source. Frame rate = 190,000frames/s.

c. Frame duration = 1 / (frame rate) = 1 /190,000 = 5.3 μs.

d. Output data rate = (190,000 frames/s) × (2 bits/frame) = 380 kbps. Here the output bit rate is greater than the sum of the input rates (370 kbps) because of extra bits added to the second source.