CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | | |

Second Internal Test

Sub:	**Data Communications**							Code:	17CS46
Date:	20/ 04 / 2019	Duration:	90 mins	Max Marks:	50	Sem:	IV	Branch:	**ISE**

Answer Any **FIVE FULL** Questions

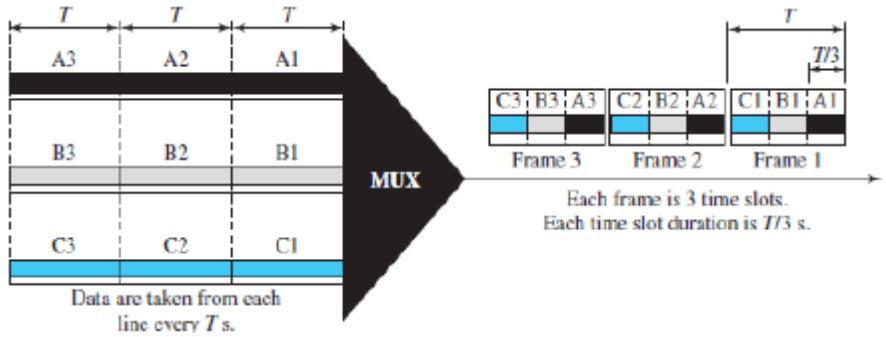
		Marks	OBE CO	OBE RBT
1	Explain synchronous TDM along with data rate management strategies.	[10]	CO2	L2
2 (a)	Four sources, each creating 250 characters per second have a character as the interleaved unit and 1 synchronizing bit is added to each frame. Find (1) data rate of each source (2) duration of each character in each source (3) frame rate (4) duration of each frame (5) no. of bits in each frame (6) data rate of the link.	[6]	CO3	L3
(b)	Two channels with bit rates of 100 kbps and 200 kbps are to be multiplexed. How can this be achieved? Calculate (1) frame rate (2) frame duration (3) bit rate of the link.	[4]	CO3	L3
3	What is spread spectrum? Explain FHSS and bandwidth sharing.	[2+8]	CO3	L2
4	Explain in detail, switching at the data link layer. Also obtain an expression for total delay.	[10]	CO3	L3
5 (a)	Explain simple parity check code with a neat diagram.	[5]	CO3	L3
(b)	Find the codeword at sender site using CRC, given data word is 101001111 and generator 10111.	[5]	CO3	L3
6 (a)	List the steps undertaken by the sender and receiver for error detection using internet checksum method.	[5]	CO3	L1
(b)	Explain the algorithms for Fletcher and Adler checksums.	[5]	CO3	L3
7	Explain stop and wait protocol with appropriate diagrams.	[10]	CO3	L3
8 (a)	Explain the frame format in HDLC protocol.	[6]	CO3	L2
(b)	Explain transition phases of Point to Point Protocol.	[4]	CO3	L2

1. Explain synchronous TDM along with data rate management strategies. (10 marks)

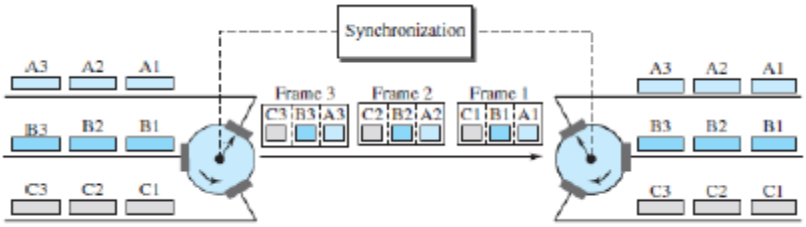
Answer:

Time-division multiplexing (TDM) is a digital process that allows several connections to share the high bandwidth of a link. Instead of sharing a portion of the bandwidth as in FDM, time is shared. Each connection occupies a portion of time in the link.

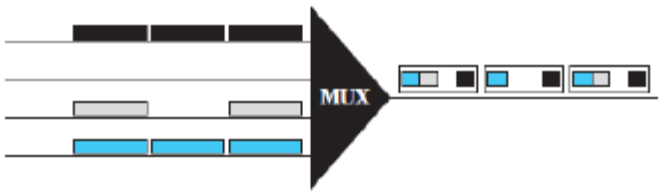
- We can divide TDM into two different schemes: synchronous and statistical
- **In synchronous TDM**, the data flow of each input connection is divided into units, where each input occupies one input time slot. A unit can be 1 bit, one character, or one block of data. Each input unit becomes one output unit and occupies one output time slot.
- The duration of an output time slot is *n* times shorter than the duration of an input time slot. If an input time slot is *T* s, the output time slot is *Tin* s where *n* is the number of connections. In other words, a unit in the output connection has a shorter duration; it travels faster.
- A round of data units from each input connection is collected into a frame . If we have n connections, frames divided into n time slots and one slot is allocated for each unit, one for each input line.
- If the duration of the input unit is *T*, the duration of each slot is *Tin* and the duration of each frame is *T*. The data rate of the output link must be *n* times the data rate of a connection to guarantee the flow of data.



- Time slots are grouped into frames. A frame consists of one complete cycle of time slots, with one slot dedicated to each sending device. In a system with n input lines, each frame has n slots, with each slot allocated to carrying data from a specific input line.
- ***Interleaving*** - TDM can be visualized as two fast-rotating switches, one on the multiplexing side and the other on the demultiplexing side. The switches are synchronized and rotate at the same speed, but in opposite directions. On the multiplexing side, as the switch opens in front of a connection, that connection has the opportunity to send a unit onto the path. This process is called interleaving. On the demultiplexing side, as the switch opens in front of a connection, that connection has the opportunity to receive a unit from the path.



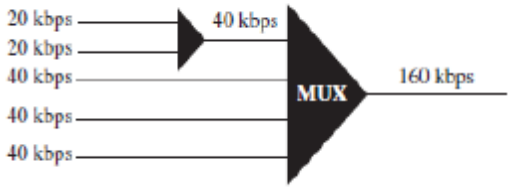
- If a source does not have data to send, the corresponding slot in the output frame is empty.



Data Rate Management

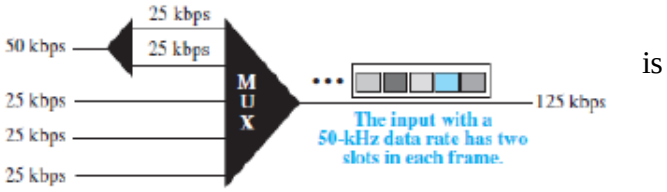
Multilevel Multiplexing

Multilevel multiplexing is a technique used when the data rate of an input line is a multiple of others. For example, in Figure 6.19, we have two inputs of 20 kbps and three inputs of 40 kbps. The first two input lines can be multiplexed together to provide a data rate equal to the last three. A second level of multiplexing can create an output of 160 kbps.



Multiple-Slot Allocation

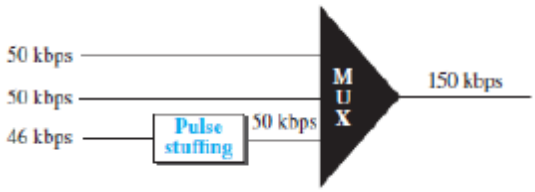
Sometimes it is more efficient to allot more than one slot in a frame to a single input line. For example, we might have an input line that has a data rate that a multiple of another input. In Figure 6.20, the input line with a 50-kbps data rate can be given two slots in the output. We insert a serial-to-parallel converter in the line to make two inputs out of one.



is

Pulse Stuffing

Sometimes the bit rates of sources are not multiple integers of each other. Therefore, neither of the above two techniques can be applied. One solution is to make the highest input data rate the dominant data rate and then add dummy bits to the input lines with lower rates. This will increase their rates. This technique is called pulse stuffing, bit padding, or bit stuffing. The idea is shown in Figure 6.21. The input with a data rate of 46 is pulse-stuffed to increase the rate to 50 kbps. Now multiplexing can take place.



2. (a) Four sources, each creating 250 characters per second have a character as the interleaved unit and 1 synchronizing bit is added to each frame. Find (1) data rate of each source (2) duration of each character in each source (3) frame rate (4) duration of each frame (5) no. of bits in each frame (6) data rate of the link. (6 marks)

Answer:

1. The data rate of each source is $250 \times 8 = 2000$ bps $= 2$ kbps.
2. Each source sends 250 characters per second; therefore, the duration of a character is 1/250 s, or 4 ms.
3. Each frame has one character from each source, which means the link needs to send 250 frames per second to keep the transmission rate of each source.
4. The duration of each frame is 1/250 s, or 4 ms. Note that the duration of each frame is the same as the duration of each character coming from each source.
5. Each frame carries 4 characters and 1 extra synchronizing bit. This means that each frame is $4 \times 8 + 1 = 33$ bits.
6. The link sends 250 frames per second, and each frame contains 33 bits. This means that the data rate of the link is 250×33, or 8250 bps. Note that the bit rate of the link is greater than the combined bit rates of the four channels. If we add the bit rates of four channels, we get 8000 bps. Because 250 frames are traveling per second and each contains 1 extra bit for synchronizing, we need to add 250 to the sum to get 8250 bps.

(b) Two channels with bit rates of 100 kbps and 200 kbps are to be multiplexed. How can this be achieved? Calculate (1) frame rate (2) frame duration (3) bit rate of the link. (4 marks)

Answer:

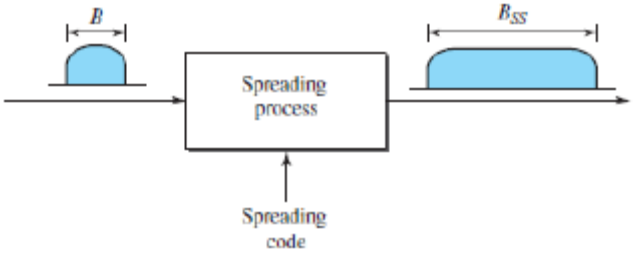
We can allocate one slot to the first channel and two slots to the second channel. Each frame carries 3 bits. The frame rate is 100,000 frames per second because it carries 1 bit from the first channel. The frame duration is 1/100,000 s, or 10 ms. The bit rate is 100,000 frames/s $\times$ 3 bits per frame, or 300 kbps. Note that because each frame carries 1 bit from the first channel, the bit rate for the first channel is preserved. The bit rate for the second channel is also preserved because each frame carries 2 bits from the second channel.

3. What is spread spectrum? Explain FHSS and bandwidth sharing. (10 marks)

Answer:

- In spread spectrum, we also combine signals from different sources to fit into a larger bandwidth, but our goals are somewhat different. Spread spectrum is designed to be used in wireless applications (LANs and WANs).
- In these types of applications, we have some concerns that outweigh bandwidth efficiency. In wireless applications, all stations use air (or a vacuum) as the medium for communication. Stations must be able to share this medium without interception by an eavesdropper and without being subject to jamming from a malicious intruder (in military operations, for example).

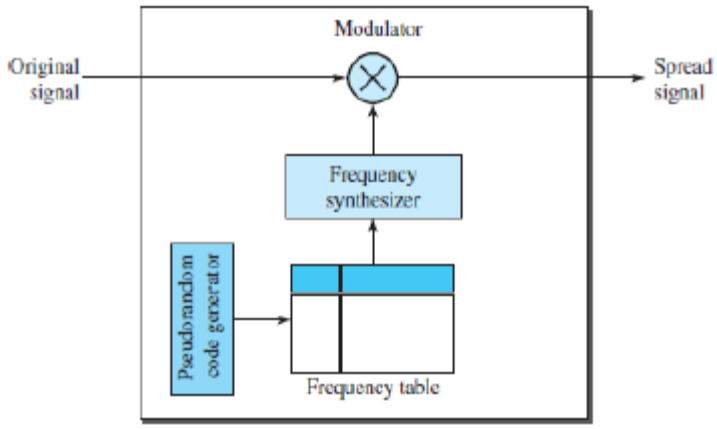
- Spread spectrum achieves its goals through two principles:
  - 1. The bandwidth allocated to each station needs to be, by far, larger than what is needed. This allows redundancy.
  - 2. The expanding of the original bandwidth *B* to the bandwidth Bss must be done by a process that is independent of the original signal. In other words, the spreading process occurs after the signal is created by the source.



- After the signal is created by the source, the spreading process uses a spreading code and spreads the bandwidth. The figure shows the original bandwidth B and the spreaded bandwidth Bss. The spreading code is a series of numbers that look random, but are actually a pattern.
- There are two techniques to spread the bandwidth: frequency hopping spread spectrum (FHSS) and direct sequence spread spectrum (DSSS).

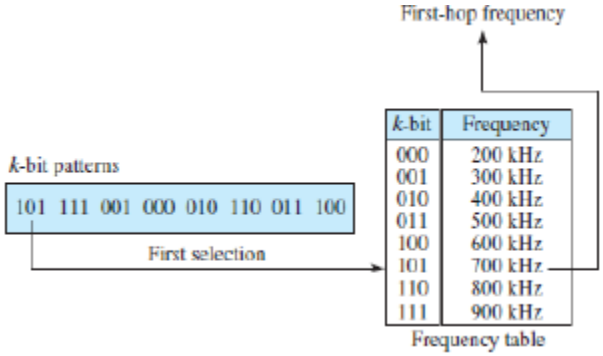
Frequency Hopping Spread Spectrum (FHSS)

- The frequency hopping spread spectrum (FHSS) technique uses *M* different carrier frequencies that are modulated by the source signal.At one moment, the signal modulates one carrier frequency; at the next moment, the signal modulates another carrier frequency. Although the modulation is done using one carrier frequency at a time, *M* frequencies are used in the long run. The bandwidth occupied by a source after spreading is *BpHSS »B.*

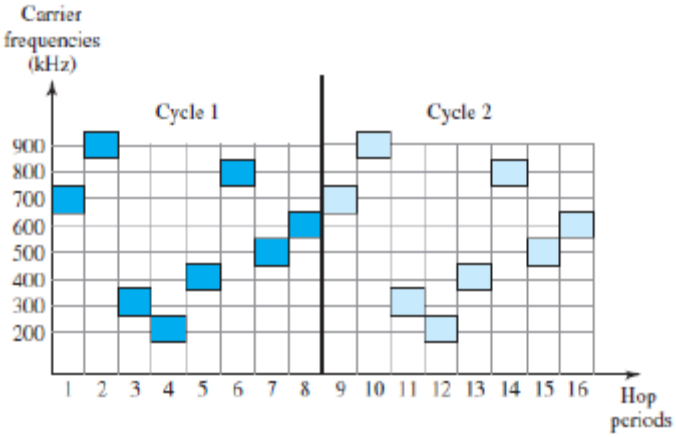


- A pseudorandom code generator, called pseudorandom noise (PN), creates a k-bit pattern for every hopping period Th• The frequency table uses the pattern to find the frequency to be used for this hopping period and passes it to the frequency synthesizer. The frequency synthesizer creates a carrier signal of that frequency, and the source signal modulates the carrier signal.

- Suppose we have decided to have eight hopping frequencies. In this case, *M is* 8 and *k* is 3. The pseudorandom code generator will create eight different 3-bit patterns. These are mapped to eight different frequencies in the frequency table.



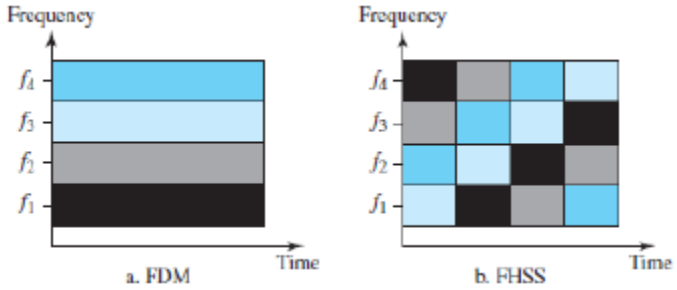
- The pattern for this station is 101, 111, 001, 000, 010, all, 100. Note that the pattern is pseudorandom it is repeated after eight hoppings. This means that at hopping period 1, the pattern is 101. The frequency selected is 700 kHz; the source signal modulates this carrier frequency. The second k-bit pattern selected is 111, which selects the 900-kHz carrier; the eighth pattern is 100, the frequency is 600 kHz. After eight hoppings, the pattern repeats, starting from 101 again.



- If there are many k-bit patterns and the hopping period is short, a sender and receiver can have **privacy**. If an intruder tries to intercept the transmitted signal, she can only access a small piece of data because she does not know the spreading sequence to quickly adapt herself to the next hop. The scheme has also an **antijamming** effect. A malicious sender may be able to send noise to jam the signal for one hopping period (randomly), but not for the whole period.

Bandwidth Sharing

- If the number of hopping frequencies is *M*, we can multiplex *M* channels into one by using the same Bss bandwidth. This is possible because a station uses just one frequency in each hopping period; *M* - 1 other frequencies can be used by other *M* - 1 stations. In other words, *M* different stations can use the same Bss if an appropriate modulation technique such as multiple FSK (MFSK) is used.



- In FDM, each station uses 1/*M* of the bandwidth, but the allocation is fixed; in FHSS, each station uses 1/*M* of the bandwidth, but the allocation changes hop to hop.

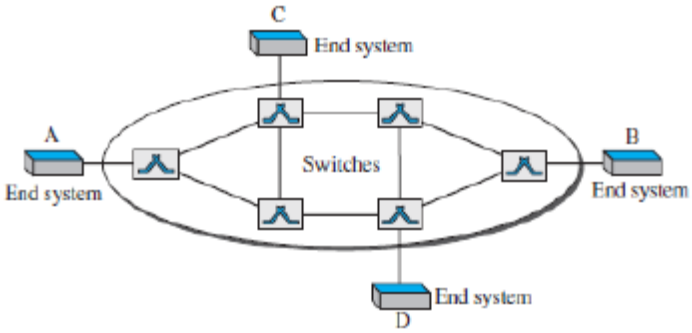
4. Explain in detail, switching at the data link layer. Also obtain an expression for total delay. (10 marks)

Answer:

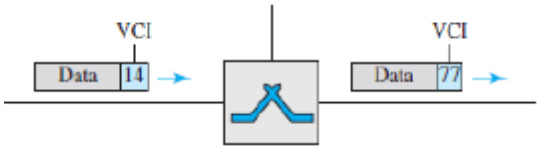
- At the data-link layer, we can have packet switching. Packet switching at the data-link layer is normally done using a virtual-circuit approach.

A **virtual-circuit network** is a cross between a circuit-switched network and a datagram network. It has some characteristics of both.

1. As in a circuit-switched network, there are setup and teardown phases in addition to the data transfer phase.
2. Resources can be allocated during the setup phase, as in a circuit-switched network, or on demand, as in a datagram network.
3. As in a datagram network, data are packetized and each packet carries an address in the header. However, the address in the header has local jurisdiction (it defines what the next switch should be and the channel on which the packet is being carried), not end-to-end jurisdiction.
4. As in a circuit-switched network, all packets follow the same path established during the connection.
5. A virtual-circuit network is normally implemented in the data-link layer, while a circuit-switched network is implemented in the physical layer and a datagram network in the network layer. But this may change in the future.



- Addressing - two types of addressing are involved: global and local (virtual-circuit identifier). A source or a destination needs to have a global address—an address that can be unique in the scope of the network or internationally if the network is part of an international network. The identifier that is actually used for data transfer is called the ***virtual-circuit identifier*(VCI)** or the ***label.*** A VCI, unlike a global address, is a small number that has only switch scope; it is used by a frame between two switches. When a frame arrives at a switch, it has a VCI; when it leaves, it has a different VCI.



- Three Phases - setup, data transfer, and teardown.

1. Data-Transfer Phase

To transfer a frame from a source to its destination, all switches need to have a table entry for this virtual circuit. The table, in its simplest form, has four columns. This means that the switch holds four pieces of information for each virtual circuit that is already set up.

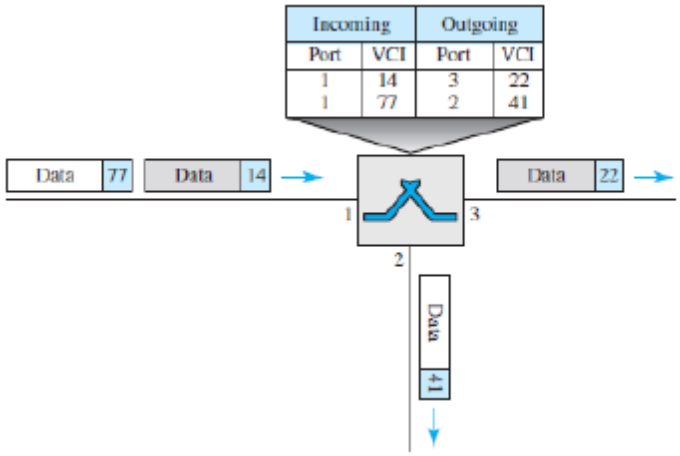
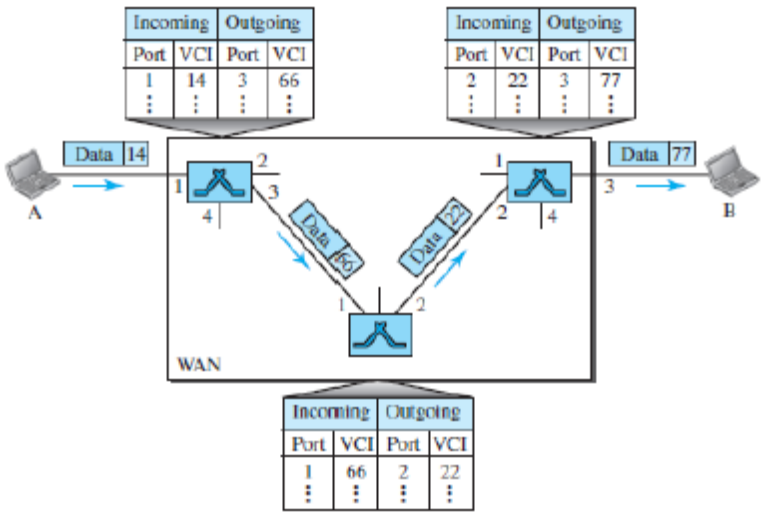


Figure shows a frame arriving at port 1 with a VCI of 14. When the frame arrives, the switch looks in its table to find port 1 and a VCI of 14. When it is found, the switch knows to change the VCI to 22 and send out the frame from port 3.

The data-transfer phase is active until the source sends all its frames to the destination. The procedure at the switch is the same for each frame of a message.



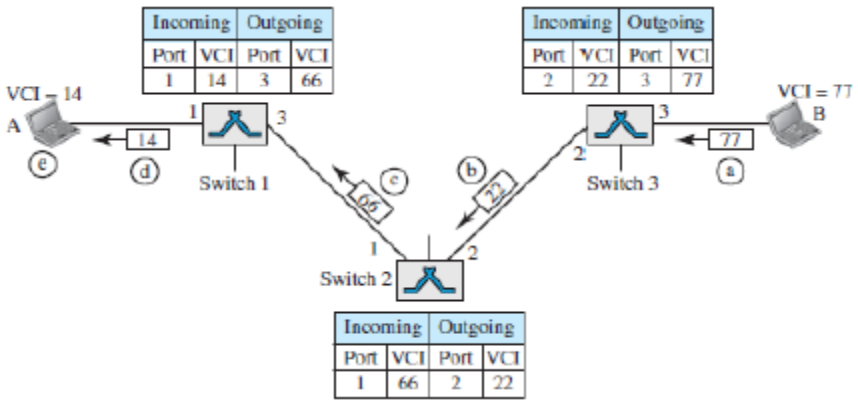
2. Setup Phase

In the setup phase, a switch creates an entry for a virtual circuit. For example, suppose source A needs to create a virtual circuit to B. Two steps are required: the setup request and the acknowledgment.

- A setup request frame is sent from the source to the destination. Figure 8.14 shows the process.

a. Source A sends a setup frame to switch 1.
b. Switch 1 receives the setup request frame. It knows that a frame going from A to B goes out through port 3. The switch, in the setup phase, acts as a packet switch; it has a routing table which is different from the switching table. For the moment, assume that it knows the output port. The switch creates an entry in its table for this virtual circuit, but it is only able to fill three of the four columns. The switch assigns the incoming port (1) and chooses an available incoming VCI (14) and the outgoing port (3). It does not yet know the outgoing VCI, which will be found during the acknowledgment step. The switch then forwards the frame through port 3 to switch 2.
c. Switch 2 receives the setup request frame. The same events happen here as at switch 1; three columns of the table are completed: in this case, incoming port (1),incoming VCI (66), and outgoing port (2).
d. Switch 3 receives the setup request frame. Again, three columns are completed: incoming port (2), incoming VCI (22), and outgoing port (3).
e. Destination B receives the setup frame, and if it is ready to receive frames from A,it assigns a VCI to the incoming frames that come from A, in this case 77. This VCI lets the destination know that the frames come from A, and not other sources.

- A special frame, called the *acknowledgment frame*, completes the entries in the switching tables.



a. The destination sends an acknowledgment to switch 3. The acknowledgment carries the global source and destination addresses so the switch knows which entry in the table is to be completed. The frame also carries VCI 77, chosen by the destination as the incoming VCI for frames from A. Switch 3 uses this VCI to complete the outgoing VCI column for this entry. Note that 77 is the incoming VCI for destination B, but the outgoing VCI for switch 3.
b. Switch 3 sends an acknowledgment to switch 2 that contains its incoming VCI in the table, chosen in the previous step. Switch 2 uses this as the outgoing VCI in the table.

c. Switch 2 sends an acknowledgment to switch 1 that contains its incoming VCI in the table, chosen in the previous step. Switch 1 uses this as the outgoing VCI in the table.
d. Finally switch 1 sends an acknowledgment to source A that contains its incoming VCI in the table, chosen in the previous step.
e. The source uses this as the outgoing VCI for the data frames to be sent to destination B

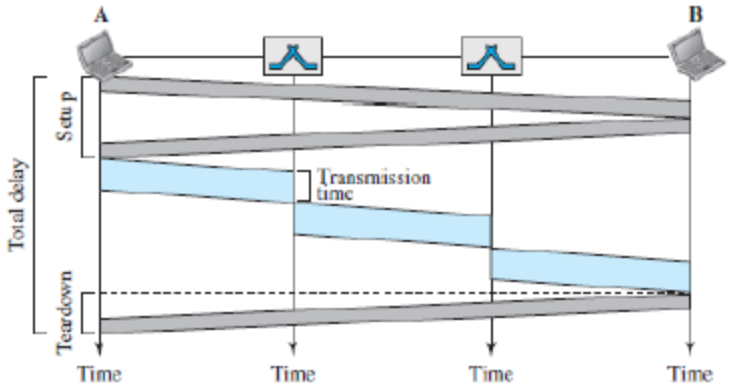
3. ***Teardown Phase***

In this phase, source A, after sending all frames to B, sends a special frame called a *teardown request*. Destination B responds with a teardown confirmation frame. All switches delete the corresponding entry from their tables.

Delay in Virtual-Circuit Networks

In a virtual-circuit network, there is a one-time delay for setup and a one-time delay for teardown. If resources are allocated during the setup phase, there is no wait time for individual packets. Figure 8.16 shows the delay for a packet traveling through two switches in a virtual-circuit network.

Figure 8.16 *Delay in a virtual-circuit network*

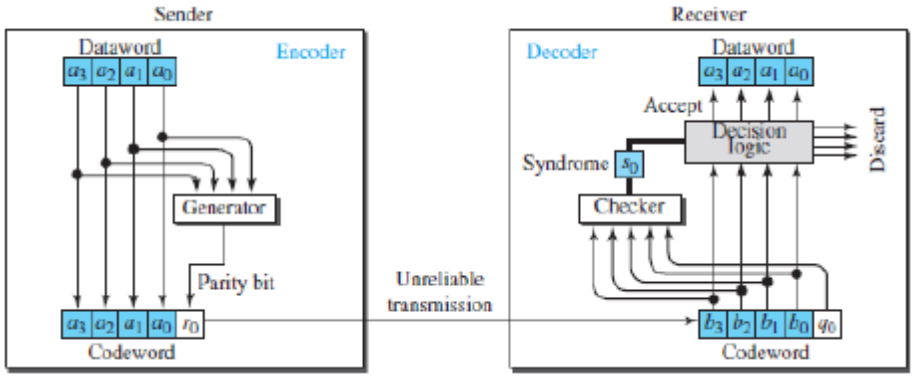


The packet is traveling through two switches (routers). There are three transmission times ($3T$), three propagation times (3τ), data transfer depicted by the sloping lines, a setup delay (which includes transmission and propagation in two directions), and a teardown delay (which includes transmission and propagation in one direction). The total delay time is

$$\text{Total delay} = 3T + 3\tau + \text{setup delay} + \text{teardown delay}$$

5. (a) Explain simple parity check code with a neat diagram. (5 marks)
Answer:

In this code, a *k-bit* data word is changed to an n-bit codeword where $n = k + 1$. The extra bit, called the parity bit, is selected to make the total number of 1s in the codeword even. The minimum Hamming distance for this category is dmin =2, which means that the code is a single-bit error-detecting code.



The calculation is done in modular arithmetic. The encoder uses a generator that takes a copy of a 4-bit dataword (a0, a1, a2, and a3) and generates a parity bit r0. The dataword bits and the parity bit create the 5-bit codeword. The parity bit that is added makes the number of 1s in the codeword even. This is normally done by adding the 4 bits of the dataword (modulo-2); the result is the parity bit. In other words,

$$r_0 = a_3 + a_2 + a_1 + a_0 \quad \text{(modulo-2)}$$

If the number of 1s is even, the result is 0; if the number of 1s is odd, the result is 1. In both cases, the total number of 1s in the codeword is even. The sender sends the codeword which may be corrupted during transmission. The receiver receives a 5-bit word. The checker at the receiver does the same thing as the generator in the sender with one exception: The addition is done over all 5 bits. The result, which is called the syndrome, is just 1 bit. The syndrome is 0 when the number of 1s in the received codeword is even; otherwise, it is 1.

$$s_0 = b_3 + b_2 + b_1 + b_0 + q_0 \quad \text{(modulo-2)}$$

The syndrome is passed to the decision logic analyzer. If the syndrome is 0, there is no error in the received codeword; the data portion of the received codeword is accepted as the data word; if the syndrome is 1, the data portion of the received codeword is discarded. The data word is not created.

(b) Find the codeword at sender site using CRC, given data word is 101001111 and generator 10111. (5 marks)
Answer:

Dataword : 101001111 ⇒ k = 9 bits → No: of bits in dataword.

Generator (divisor) : 10111 ⇒ n-k+1 = 5 bits

⇒ n-k = 4

n-9 = 4

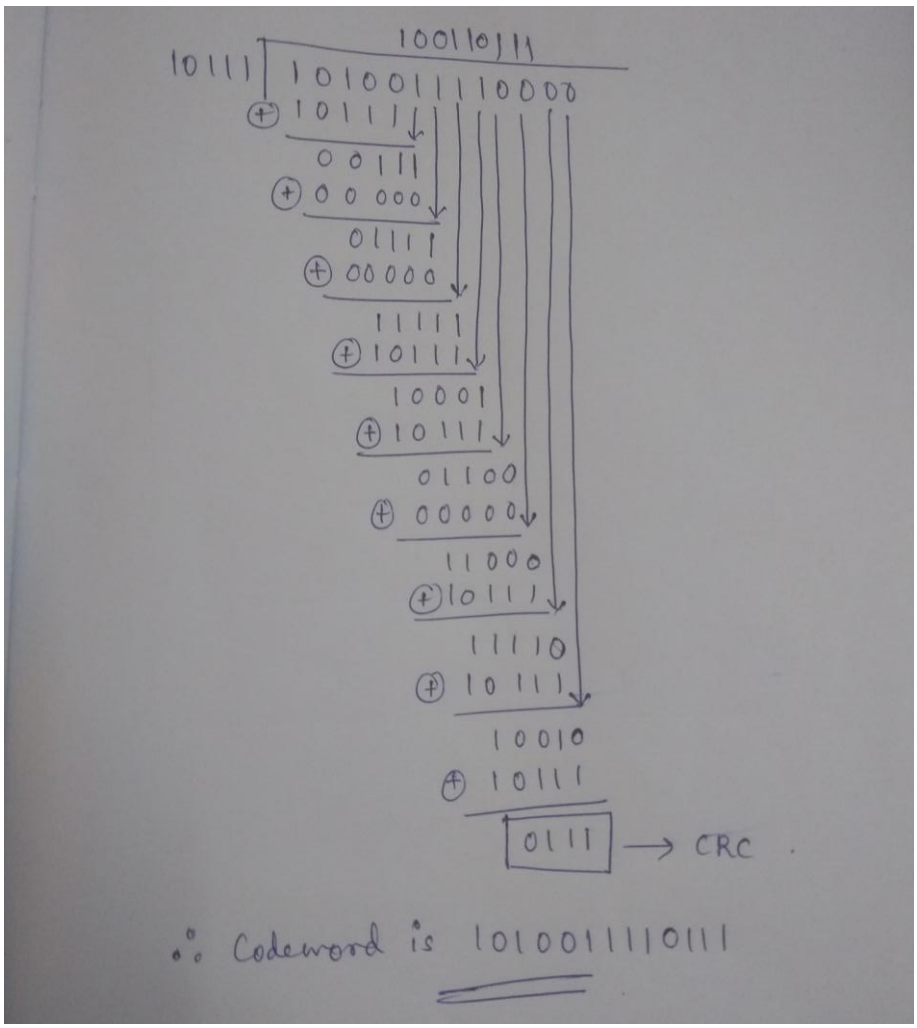
⇒ n = 13 bits ⇒ No: of bits in codeword.

∴ Append n-k = 4 bits (0s) to the dataword to form augmented dataword

⇒ Augmented dataword : 1010011110000

CRC will be the remainder when augmented dataword is divided by the generator 10111.

No: of bits in CRC = n-k = 4.



∴ Codeword is 1010011110111

6. (a) List the steps undertaken by the sender and receiver for error detection using internet checksum method. (5 marks)

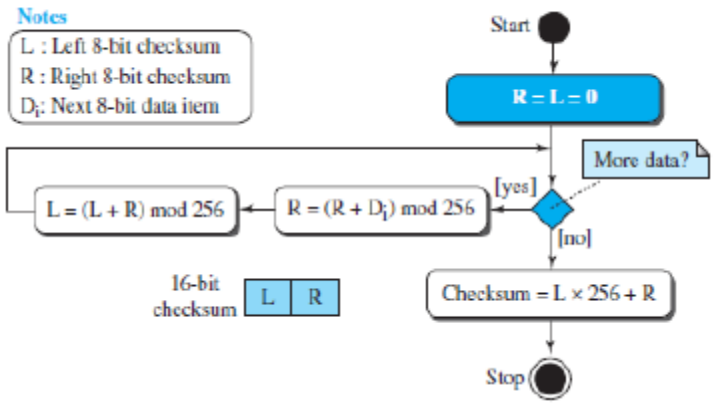
Answer:

Sender	*Receiver*
1. The message is divided into 16-bit words. 2. The value of the checksum word is initially set to zero. 3. All words including the checksum are added using one's complement addition. 4. The sum is complemented and becomes the checksum. 5. The checksum is sent with the data.	1. The message and the checksum are received. 2. The message is divided into 16-bit words. 3. All words are added using one's complement addition. 4. The sum is complemented and becomes the new checksum. 5. If the value of the checksum is 0, the message is accepted; otherwise, it is rejected.

(b) Explain the algorithms for Fletcher and Adler checksums. (5 marks)

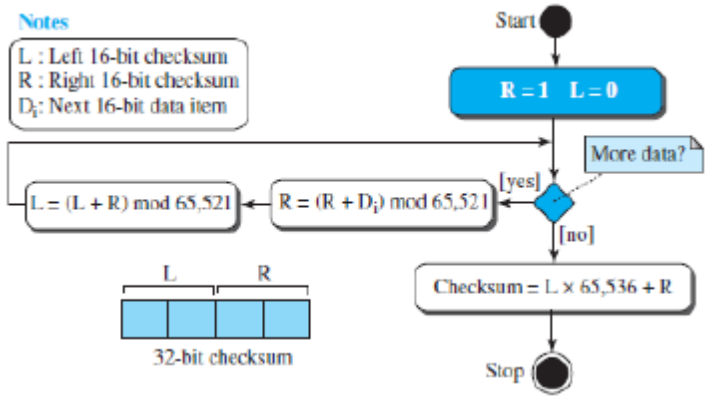
Answer:

Fletcher Checksum:



The Fletcher checksum was devised to weight each data item according to its position. Fletcher has proposed two algorithms: 8-bit and 16-bit. The first, 8-bit Fletcher, calculates on 8-bit data items and creates a 16-bit checksum. The second, 16-bit Fletcher, calculates on 16-bit data items and creates a 32-bit checksum. The 8-bit Fletcher is calculated over data octets (bytes) and creates a 16-bit check-sum. The calculation is done modulo 256 (2^8), which means the intermediate results are divided by 256 and the remainder is kept. The algorithm uses two accumulators, L and R. The first simply adds data items together; the second adds a weight to the calculation. There are many variations of the 8-bit Fletcher algorithm; we show a simple one in Figure 10.18. The 16-bit Fletcher checksum is similar to the 8-bit Fletcher checksum, but it is calculated over 16-bit data items and creates a 32-bit checksum. The calculation is done modulo 65,536.

Adler Checksum:

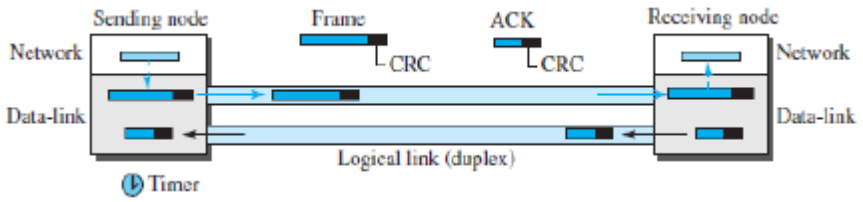


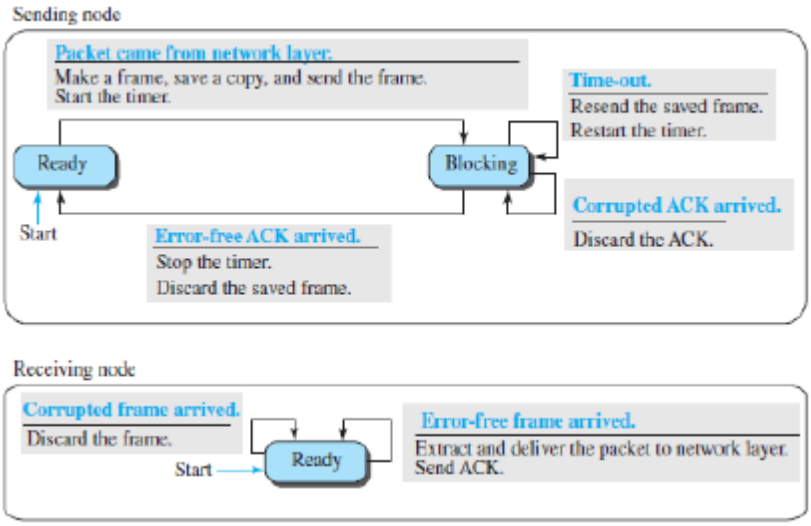
The Adler checksum is a 32-bit checksum. Figure 10.19 shows a simple algorithm in flowchart form. It is similar to the 16-bit Fletcher with three differences. First, calculation is done on single bytes instead of 2 bytes at a time. Second, the modulus is a prime number (65,521) instead of 65,536. Third, L is initialized to 1 instead of 0. It has been proved that a prime modulo has a better detecting capability in some combinations of data.

7. Explain stop and wait protocol with appropriate diagrams. (10 marks)
Answer:

- In this protocol, the sender sends one frame at a time and waits for an acknowledgment before sending the next one.
- To detect corrupted frames, we need to add a CRC to each data frame.

- When a frame arrives at the receiver site, it is checked.
- If its CRC is incorrect, the frame is corrupted and silently discarded.
- The silence of the receiver is a signal for the sender that a frame was either corrupted or lost.
- Every time the sender sends a frame, it starts a timer.
- If an acknowledgment arrives before the timer expires, the timer is stopped and the sender sends the next frame (if it has one to send).
- If the timer expires, the sender resends the previous frame, assuming that the frame was either lost or corrupted.
- This means that the sender needs to keep a copy of the frame until its acknowledgment arrives.
- When the corresponding acknowledgment arrives, the sender discards the copy and sends the next frame if it is ready.
- Figure shows the outline for the Stop-and-Wait protocol.
- Note that only one frame and one acknowledgment can be in the channels at any time.





Sender States

The sender is initially in the ready state, but it can move between the ready and blocking state.

1. **Ready State**

- When the sender is in this state, it is only waiting for a packet from the network layer.
- If a packet comes from the network layer, the sender creates a frame, saves a copy of the frame, starts the only timer and sends the frame.
- The sender then moves to the blocking state.

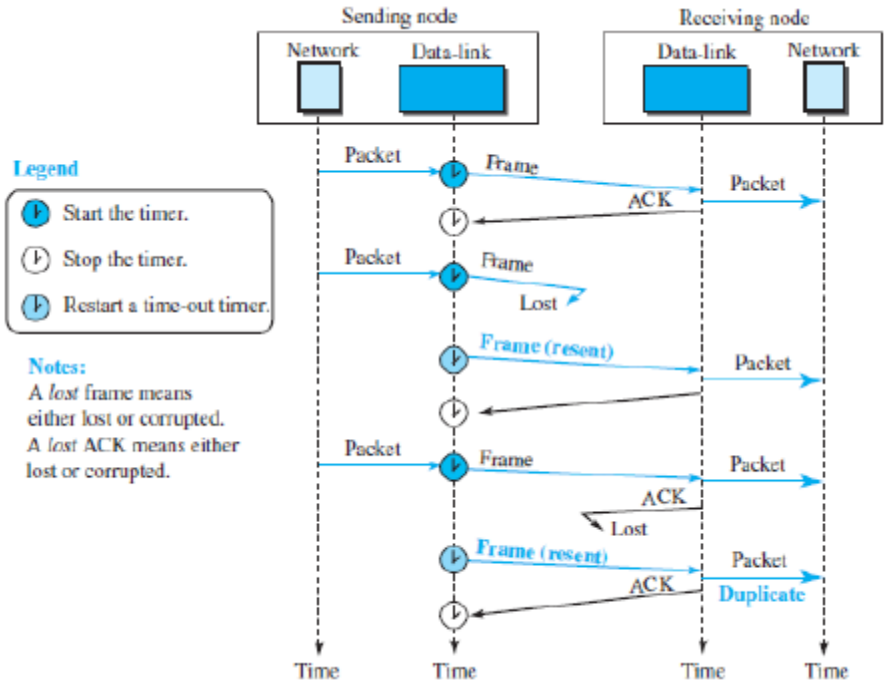
2. **Blocking State: When the sender is in this state, three events can occur**

- If a time-out occurs, the sender resends the saved copy of the frame and restarts the timer.
- If a corrupted ACK arrives, it is discarded.
- If an error-free ACK arrives, the sender stops the timer and discards the saved copy of the frame. It then moves to the ready state.

Receiver

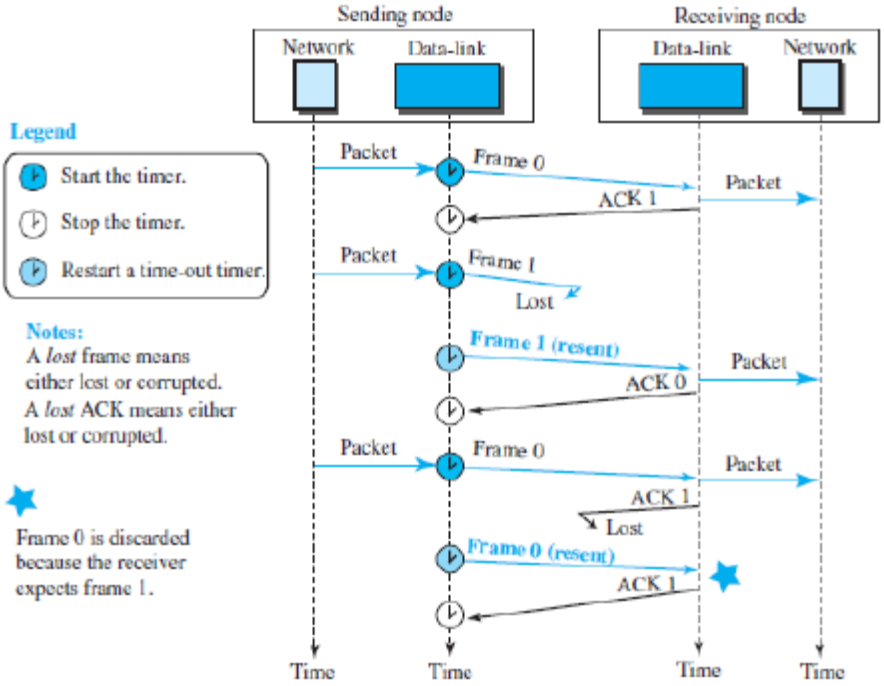
The receiver is always in the ready state. Two events may occur:
a. If an error-free frame arrives, the message in the frame is delivered to the network layer and an ACK is sent.
b. If a corrupted frame arrives, the frame is discarded.

The next figure shows an example of this protocol. The first frame is sent and acknowledged. The second frame is sent, but lost. After time-out, it is resent. The third frame is sent and acknowledged, but the acknowledgment is lost. The frame is resent. However, there is a problem with this scheme. The network layer at the receiver site receives two copies of the third packet, which is not right. This can be corrected using sequence numbers and acknowledgment numbers.



Sequence and Acknowledgment Numbers

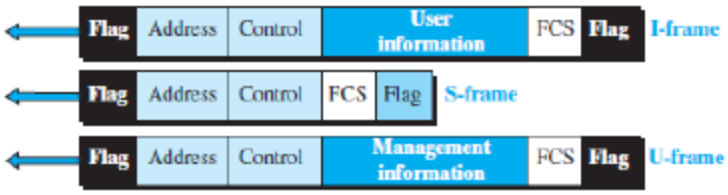
Duplicate packets, as much as corrupted packets, need to be avoided. To correct this, we need to add sequence numbers to the data frames and acknowledgment numbers to the ACK frames. However, numbering in this case is very simple. Sequence numbers are 0, 1, 0, 1, 0, 1, . . . ; the acknowledgment numbers can also be 1, 0, 1, 0, 1, 0, … In other words, the sequence numbers start with 0, the acknowledgment numbers start with 1. An acknowledgment number always defines the sequence number of the next frame to receive.



8. (a) Explain the frame format in HDLC protocol. (5 marks)

Answer:

- To provide the flexibility necessary to support all the options possible in the modes and configurations just described, HDLC defines three types of frames:
- Information frames (I-frames) , supervisory frames (S-frames), and unnumbered frames (U frames).
- Each type of frame serves as an envelope for the transmission of a different type of message.
- I-frames are used to data-link user data and control information relating to user data (piggy-backing).
- S-frames are used only to transport control information.
- U frames are reserved for system management.
- Information carried by U-frames is intended for managing the link itself.
- Each frame in HDLC may contain up to six fields, as shown in Figure 11.16 - a beginning flag field, an address field, a control field, an information field, a frame check sequence (FCS) field, and an ending flag field.
- In multiple-frame transmissions, the ending flag of one frame can serve as the beginning flag of the next frame.



Flag field

This field contains synchronization pattern 01111110, which identifies both the beginning and the end of a frame.

Address field

This field contains the address of the secondary station. If a primary station created the frame, it contains a to address. If a secondary station creates the frame, it contains a from address. The address field can be one byte or several bytes long, depending on the needs of the network.

Control field

The control field is one or two bytes used for flow and error control.

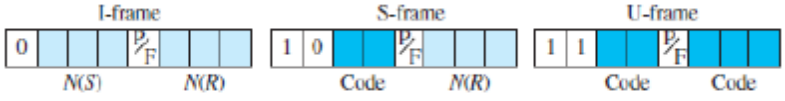
Information field

The information field contains the user's data from the network layer or management information. Its length can vary from one network to another.

FCS field

The frame check sequence (FCS) is the HDLC error detection field. It can contain either a 2- or 4-byte CRC.

Control Field Format:



Control Field for I-Frames

- I-frames are designed to carry user data from the network layer.
- In addition, they can include flow- and error-control information (piggybacking).
- The subfields in the control field are used to define these functions.

- The first bit defines the type.
- If the first bit of the control field is 0, this means the frame is an I-frame.
- The next 3 bits, called N(S), define the sequence number of the frame.
- Note that with 3 bits, we can define a sequence number between 0 and 7.

- The last 3 bits, called N(R), correspond to the acknowledgment number when piggybacking is used.
- The single bit between N(S) and N(R) is called the P/F bit.
- The P/F field is a single bit with a dual purpose.
- It has meaning only when it is set (bit =1) and can mean poll or final.
- It means poll when the frame is sent by a primary station to a secondary (when the address field contains the address of the receiver).
- It means final when the frame is sent by a secondary to a primary (when the address field contains the address of the sender).

Control Field for S-Frames

- Supervisory frames are used for flow and error control whenever piggybacking is either impossible or inappropriate.
- S-frames do not have information fields.
- If the first 2 bits of the control field are 10, this means the frame is an S-frame.
- The last 3 bits, called N(R), correspond to the acknowledgment number (ACK) or negative acknowledgment number (NAK), depending on the type of S-frame.
- The 2 bits called code are used to define the type of S-frame itself.
- With 2 bits, we can have four types of S-frames, as described below:

1. Receive ready (RR)

- If the value of the code subfield is 00, it is an RR S-frame.
- This kind of frame acknowledges the receipt of a safe and sound frame or group of frames.
- In this case, the value of the N(R) field defines the acknowledgment number.

2. Receive not ready (RNR)

- If the value of the code subfield is 10, it is an RNR S-frame.
- This kind of frame is an RR frame with additional functions.
- It acknowledges the receipt of a frame or group of frames, and it announces that the receiver is busy and cannot receive more frames. It acts as a kind of congestion control mechanism by asking the sender to slow down. The value of N(R) is the acknowledgment number.

3. Reject (REJ)

- If the value of the code subfield is 01, it is an REJ S-frame.
- This is a NAK frame, but not like the one used for Selective Repeat ARQ.

- It is a NAK that can be used in Go-Back-N ARQ to improve the efficiency of the process by informing the sender, before the sender timer expires, that the last frame is lost or damaged.
- The value of N(R) is the negative acknowledgment number.

4. Selective reject (SREJ)

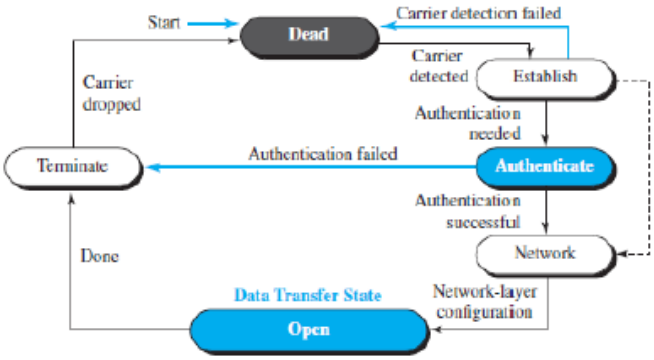
- If the value of the code subfield is 11, it is an SREJ S-frame.
- This is a NAK frame used in Selective Repeat ARQ.
- Note that the HDLC Protocol uses the term selective reject instead of selective repeat.
- The value of N(R) is the negative acknowledgment number.

Control Field for U-Frames

- Unnumbered frames are used to exchange session management and control information between connected devices.
- U-frames contain an information field, but one used for system management information, not user data.
- U-frame codes are divided in to two sections: a 2-bit prefix before the P/F bit and a 3 bit suffix after the P/F bit. Together, these two segments (5 bits) can be used to create up to 32 different types of U-frames.

(b). Explain transition phases of Point to Point Protocol. (5 marks)

Answer:



- A PPP connection goes through phases which can be shown in a transition phase diagram.
- The transition diagram, which is an FSM, starts with the **dead state**.
- In **dead state**, there is no active carrier (at the physical layer) and the line is quiet. When one of the two nodes starts the communication, the connection goes into the **establish state**.
- In **establish state**, options are negotiated between the two parties.
- If the two parties agree that they need authentication (for example, if they do not know each other), then the system needs to do authentication (an extra step); otherwise, the parties can simply start communication.
- Data transfer takes place in the **open state**.
- When a connection reaches **open state**, the exchange of data packets can be started.
- The connection remains in **open state** until one of the end points wants to terminate the connection.
- In this case, the system goes to the **terminate state**. The system remains in this state until the carrier (physical-layer signal) is dropped, which moves the system to the **dead state** again.

Scheme Of Evaluation
Internal Assessment Test 2 – April 2019

Sub:	Data Communication							Code:	17CS46
Date:	20/04/2019	Duration:	90mins	Max Marks:	50	Sem:	IV	Branch:	ISE

Note: Answer Any Five Questions

Question #		Description	Marks Distribution		Max Marks
1		**Explain synchronous TDM along with data rate management strategies.** • Diagram • Time slots, empty slots & frames + Interleaving + Framing bits • 3 strategies	2 M 5 M 3 M		10 M
2	a)	**Four sources, each creating 250 characters per second have a character as the interleaved unit and 1 synchronizing bit is added to each frame. Find (1) data rate of each source (2) duration of each character in each source (3) frame rate (4) duration of each frame (5) no. of bits in each frame (6) data rate of the link.** • 2 kbps • 4 ms • 250 frames per second • 4 ms • 33 • 8250 bps	1 M * 6	6 M	10 M
	b)	**Two channels with bit rates of 100 kbps and 200 kbps are to be multiplexed. How can this be achieved? Calculate (1) frame rate (2) frame duration (3) bit rate of the link.** • Allocate 1 slot to 1st channel and 2 slots to 2nd channel • 100000 frames per second • 10 ms	1 M * 5	4 M	

		• 300 kbps			
3		**What is spread spectrum? Explain FHSS and bandwidth sharing.** • Why is SS needed + 2 principles • FHSS diagram • Explanation • Bandwidth sharing	2 M 2 M 4 M 2 M	2 M + 8 M	10 M
4		**Explain in detail, switching at the data link layer. Also obtain an expression for total delay.** • Virtual circuit switching – description • 3 phases with necessary diagrams • Delay diagram + calculation	2M 2M+2M+2M 2 M	10 M	10 M
5	a)	**Explain simple parity check code with a neat diagram.** • Block diagram • Explanation	2.5 M 2.5 M	5 M	10 M
	b)	**Find the codeword at sender site using CRC, given data word is 101001111 and generator 10111.** • Finding no. of 0s to be appended to data word • Finding the augmented data word • Division • Getting the CRC • Obtaining the code word	1 M * 5	5 M	
6	a)	**List the steps undertaken by the sender and receiver for error detection using internet checksum method.** • 5 steps each at sender and receiver	1 M * 5	5 M	10 M
	b)	**Explain the algorithms for Fletcher and Adler checksums.** • Fletcher checksum algorithm + explanation • Adler checksum algorithm + explanation	2.5 M 2.5 M	5 M	

7		**Explain stop and wait protocol with appropriate diagrams.** • Diagram + explanation • FSM for sender and receiver nodes • Description of sender and receiver states • Duplicate packets • Sequence and acknowledgement number + flow diagrams	2 M 2 M 2 M 1 M 3 M	10 M	10 M
8	a)	**Explain the frame format in HDLC protocol.** • 3 types of frames with structure • Control field format	3 M 2 M	5 M	10 M
	b)	**Explain transition phases of Point to Point Protocol.** • Transition phase diagram • Explanation	3 M 2 M	5 M	