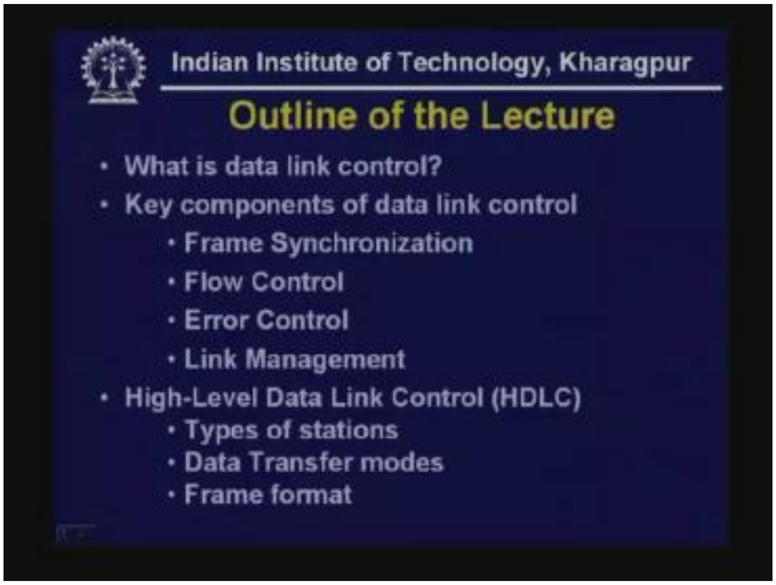
Data Communications
Prof. A. Pal
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur
Lecture 17
Data Link Control

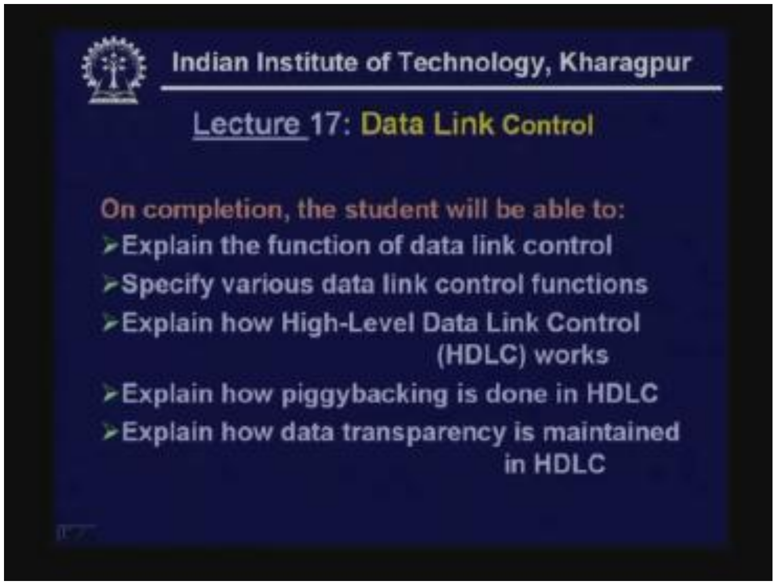
Hello viewers welcome to today's lecture on Data Link Control. In the last lecture we have discussed about flow and error control through important mechanism used for efficient and reliable data communication. In this lecture we shall discuss about other components that is required for data communication between two machines or two systems. Actually a higher level of logic has to be added above the physical layer to facilitate efficient and reliable data communication. That is the topic of today's lecture.

(Refer Slide Time: 01:56)



Here is the outline of the lecture. First we shall discuss about Data Link Control then we shall discuss the key components of Data Link Control such as frame synchronization, flow control, error control and link management. We have already discussed about flow control and error control in detail so we shall not discuss about these two important components in this lecture. However, we shall see how they are used here in some standard protocol. Then we shall consider about link management and we shall discuss about an important standard that is known as High-Level Data Link Control or HDLC in short. Particularly it is characterized by three important parameters like types of stations, data transfer modes, frame format etc which we shall discuss in detail.

(Refer Slide Time: 02:58)



On completion of this lecture the students will be able to explain the functions of Data Link Control, they will be able to specify various Data Link Control functions, they will be able to explain how High-Level Data Link Control works that is HDLC which is the most popular Data Link Control protocol that is in used today, explain how piggybacking is done is HDLC and also explain how data transparency is maintained in HDLC.

So, before we discuss about the definition of Data Link Control let us look at the key features of the link that is being used for communication between two machines. First feature is we shall assume that two machines are directly connected. That means they are not connected through some intermediate node or other machine. We shall assume that two machines are directly connected.

Second assumption is there will be occasional errors in the communication circuit so there will be possibility of some errors in the communication circuit as data communication takes place there will be some errors but obviously the probability of error is quite small and as a result it will occur occasionally not frequently.

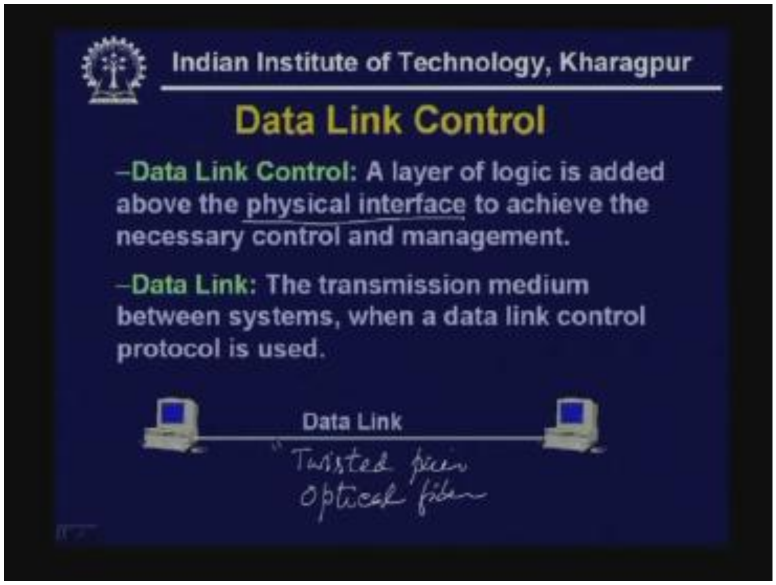
Third important assumption is there is non zero propagation delay. As I mentioned in the last lecture there will be some propagation delay for data communication it will take some time and obviously this delay will be dependant on the transmission media used and also it will be dependant on the distance between the two machines. That means the transmission media and also the distance the between the two will decide the propagation delay. Then the communication link and the machines have finite data rate. So communication will take place with some finite data because the machines will have limited processing capability and based on the bandwidth of the medium the data rate will be limited. So it is limited both by machines and also by the communication link.

(Refer Slide Time: 03:38)

Obviously in this scenario our objective is to devise suitable mechanism for efficient and reliable communication by using the unreliable transmission link. This will require great deal of coordination between the two machines and this coordination is represented by this Data Link Control protocol.

So, a Data Link Control is nothing but a layer to logic added above the physical interface to achieve necessary control and management. As it is shown here you have got two computers they are connected by communication link or Data Link, it can be twisted-pair, it can be optical fiber, it can be coaxial cable, it can be wireless or whatever it may be they are directly linked and the transmission media between the system when a Data Link Control protocol is used is known as Data Link so this is the Data Link (Refer Slide Time: 6:50) so you can use varieties of medium but the protocol that we are discussing is really independent of the medium that you are using. So this is irrespective of the medium that we are using for linking two devices. We are mainly concerned with the layer of logic or the protocol that has to be used for efficient and reliable communication.

(Refer Slide Time: 06:02)



What are the key components of Data Link Control? These are listed here.

(Refer Slide Time: 07:20)

First one is frame synchronization. We have already discussed about some kind of synchronization that is necessary for data communication between two links particularly in the context of asynchronous and synchronous serial communication. Now synchronization can be done at three different levels; bit level, word level and frame level. Here we are concerned with frame communication.

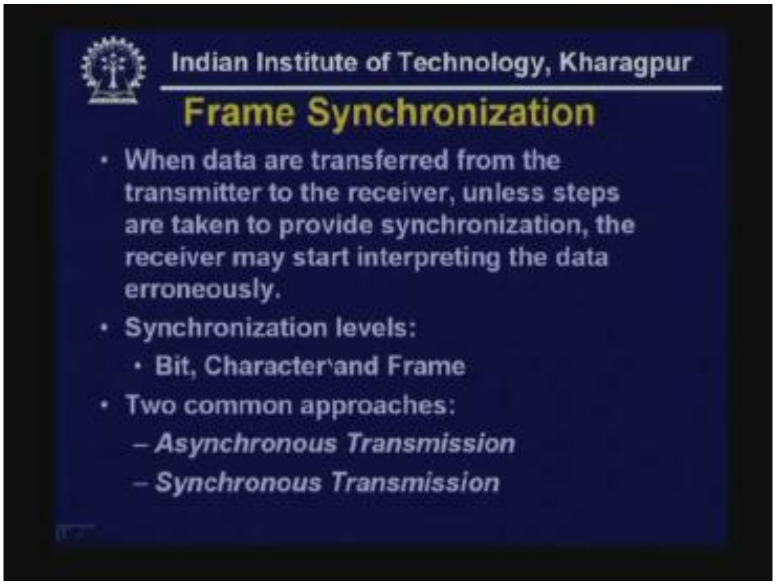
We are interested in sending a frame of information a sequence of bits or a sequence of characters and obviously the beginning and end of a data block called a frame should be distinguished so that the receiving end can identify where a particular frame is beginning and when it is ending known as frame synchronization. So this is in addition to bit and word synchronization. This will work at a little higher level.

Secondly, the flow control which we have already discussed in detail. The basic objective is the sender should not send frames at a faster rate such that the receiver is overwhelmed. We have already various flow control mechanisms such as stop-and-wait flow control and also the sliding-window flow control. And in most of the applications here we shall see that the sliding-window flow control is used for its efficiency.

We have already discussed error control techniques such as stop-and-wait ARQ go-back-N ARQ and the select-repeat ARQ these are used in this protocol. These are included in this Data Link Control protocols. Error control technique is necessary because any bit errors introduced by the transmission system should be corrected. Then we have the control and data on same link. You have to send data and also you have to perform the management. The receiver must be able to distinguish between control information and data that is being sent using the same medium. So you are sending that control and data through the same link and unless there is some kind of some kind of protocol that means agreed upon rules and conventions then it is not possible to identify what is data and what is control information. This has to be used to show that the receiver can identify what is control information and what is data being transmitted. Then comes the link management. This is necessary so that the procedures for the management of initiation, maintenance and termination of a sustained data exchange. that means when two machines are connected then you have to first set up the connection then you have to maintain the connection throughout the period for data communication and also when data transfer is over then it has to be terminated. Therefore all these things are a part of the link management. These are the key components of the Data Link Control. We shall discuss how actually they work.

First let us focus on frame synchronization.

(Refer Slide Time: 11:05)



When data is transferred from the transmitter to the receiver unless steps are taken to provide synchronization the receiver may start interpreting the data erroneously so some kind of agreed up on convention has to be used. And as I mentioned synchronization levels are done at three different levels; bit, character and frame. And the two more common approaches are asynchronous transmission and synchronous transmission. So let us consider these two one after the other.

First let us consider asynchronous transmission. So, in asynchronous transmission the synchronization is done at the character level so data transmits one character at a time. As we know the number of bits can vary from 5 to 8 and timing or synchronization must only be maintained within each character. We have already discussed about this.

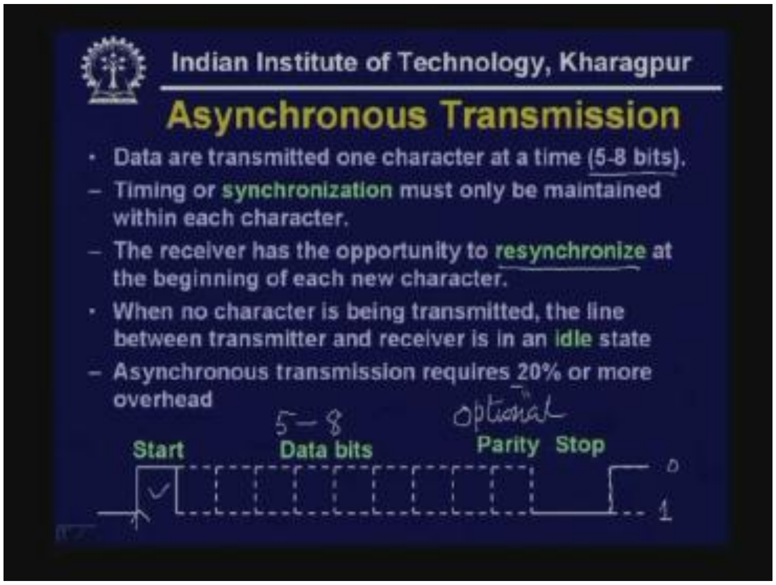
With the help of a start character the synchronization is achieved and this signifies the beginning of a character and after that there can be data bits which can vary from 5 to 8, here this is an optional parity bit (Refer Slide time: 12:35), if error detection is required parity bit is used then it is followed by stopped bit and so the synchronization is restricted within this character which is done with the help of start and stop bits. And then immediately after the stop bit either another start bit can start so in that case it will become 1 or it can remain 0. actually this is one this is zero (Refer Slide time: 13:03) it will remain 1 if no new character is sent or it will become 0 that means a start character will start if another character is being sent. So in this way character by character synchronization is done in asynchronous transmission.

And here as you can see the receiver has the opportunity to resynchronize at beginning of each new character. So because synchronization is done at the beginning of each character the clock frequency between transmitter and receiver need not be exactly same there can be a difference of about 5%. So, that kind of flexibility exists here. So when no character is being transmitted the line between the transmitter and receiver is in idle state.

So this is the idle state one state (Refer Slide Time: 14:05) so a character can be sent, the line can remain idle then another character can be sent which is synchronized.

We have already seen that asynchronous transmission require 20% or more overhead. So asynchronous transmission is although very simple and some kind of self synchronizing it is not very efficient because of high overhead.

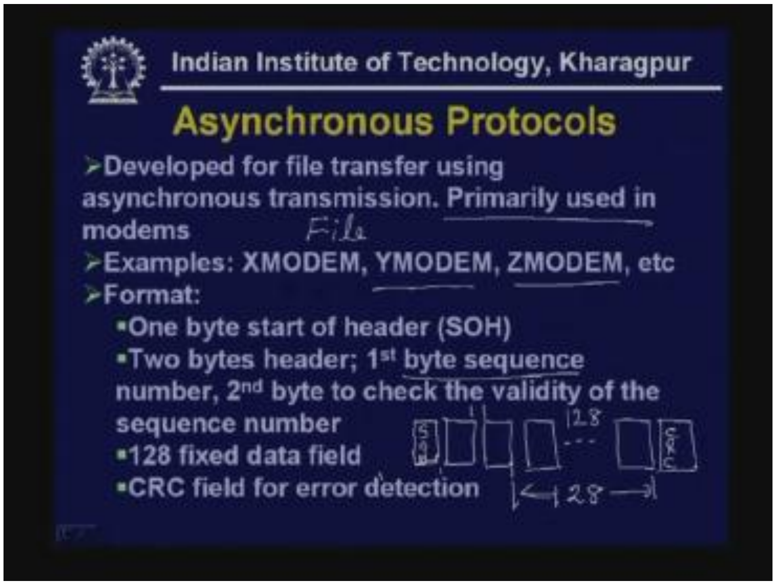
(Refer Slide Time: 11:47)



Now asynchronous protocols have to be developed on top of the character oriented transmission. This has been primarily used in modems. Particularly whenever you want to send a file from one machine to another this type of asynchronous protocol have to be used. Some of the example protocols are known as X-MODEM, Y-MODEM, Z-MODEM and so on. So the character format is briefly specified here. It has got one byte start of header so you have got a character which is start of header SOH and it is followed by two bytes of header so this is two bytes of header however one byte gives you the sequence number. So one byte is a sequence number and the other byte actually the complement of that is being sent here. Therefore using these two bytes the sequence number and its complement is received so if there is an error that can be detected with the help of these two bits. So, second byte is essentially to check the validity of the sequence number. so whatever character is sent the complement of the bit is sent.

Since we are using one byte of character the number of stations it can have is 128 and after that there is 128 fixed data field so it is restricted to 128 and after that 128 data field there is a checksum or CRC that is only of one character that means 8 bit. Hence this is the frame format and here you have got 128 data and finally the character. This is how you can send one file or one frame.

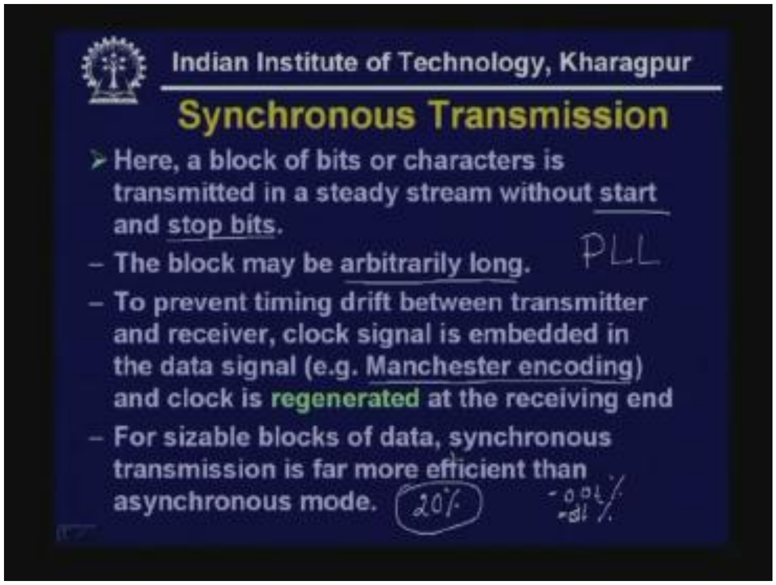
(Refer Slide Time: 14:35)



As you can see there is a character sent which is start of header followed by 2 bytes and then 128 byte fixed characters and CRC and in between one character there can be a gap so as we have seen this gap (Refer Slide time: 17:05) can be variable between two characters so the variable gap is provided. So this is how the XMODEM works and YMODEM, ZMODEM are essentially some extensions over XMODEM. There are several such protocols.

However, this asynchronous protocol is not very widely used because of the inefficiency or high overhead that is why the synchronous transmission is commonly used. In synchronous transmission a block of bits or characters are transmitted in a steady stream without start and stop bits. So, without start and stop bits both the stream of characters or bits can be sent one after the other without start bit and overhead and the block may arbitrarily long. So although the block may be arbitrarily long to prevent timing drift between transmitter and receiver, clock signal is embedded in the data signal. As we know if a frame is arbitrarily long the probability of error increases that is why the length is not really very long it is restricted to half the limit.

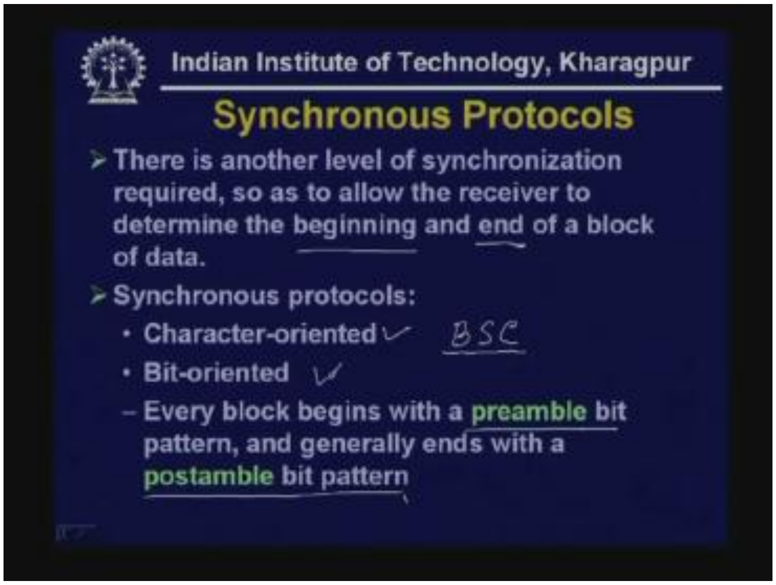
(Refer Slide Time: 18:22)



Therefore to prevent timing drift between transmitter and receiver clock signal is embedded in the data signal for example by using Manchester encoding and clock is regenerated at the receiving end. As we have mentioned earlier the clock cannot be separately sent whenever two machines are located wide apart but for synchronous transmission it is necessary that these two clocks should be identical, how that can be achieved? One way of doing that is we embed the clock as part of the data signal by suitable encoding that is the Manchester encoding technique where we have seen that for each bit one transition is included which helps in regenerating the clock at the receiving end with the help of some special hardware such as Phase Lock Loop or PLL so Phase Lock Loop is used to regenerate the clock at the receiving end so for that purpose some encoding is used. Other encoding techniques can also be used but Manchester encoding is one of the popular techniques used.

For sizable blocks of data synchronous transmission is far more efficient than asynchronous mode. As I have already mentioned asynchronous transmission requires 20% or more overhead and obviously this is not acceptable whenever we are sending over a long distance or you want higher efficiency. In such a case synchronous transmission gives you very high efficiency and that efficiency can be even 001% or may be .01% where the overhead is very small .01% or .001% so with such a low overhead the synchronous transmission gives you much better efficiency.

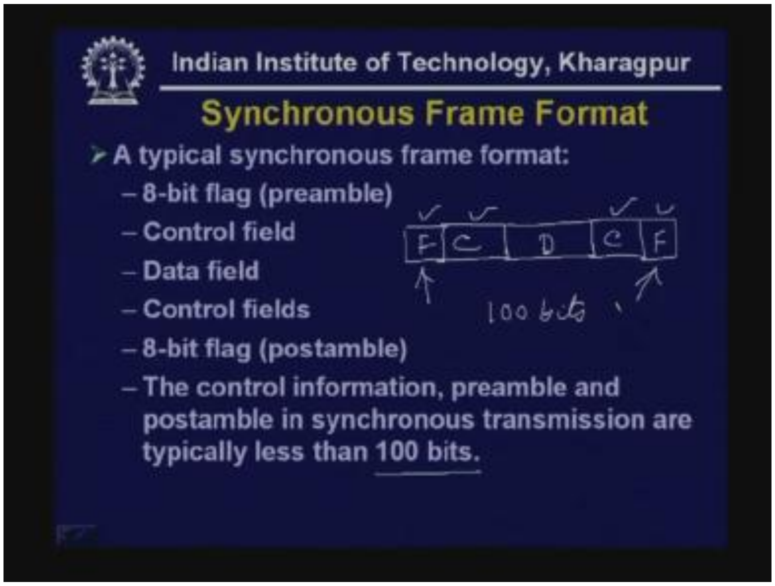
(Refer Slide Time: 20:32)



Now let us look at the protocols that are being used in synchronous transmission. There is another level of synchronization required so as to allow the receiver to determine the beginning and end of a block of data. That means apart from the synchronization of the clock there is a need for synchronization for identifying the beginning and end of a beginning and end of a block of data and there are several protocols which can be broadly divided into two types namely character oriented and bit oriented. The example of a character oriented protocol is Binary Synchronous Communication BSC which is based on characters. So a sequence of characters sent one after the other. Obviously there will be a frame involving a string character then a sequence of characters then CRC and other things. So the BSC is one such example but the character oriented protocols are not very popular because we cannot really pack more information in them.

On the other hand, the bit oriented protocol allows you to pack more information and as a result bit oriented protocols are very popular. In both cases every block begins with a preamble, bit pattern and generally ends with a postamble bit pattern. So this is the typical framing that is being done. a typical synchronous frame format is given here where it will have a some kind of an 8-bit flag followed by some control bit then there can data bits then the control field so again you have the data then some control fields for error detection and again the flag. This (Refer Slide Time: 22:43) is the typical format that is being used in synchronous frame format. As we can see the beginning and end is identified with the help of a special flag it can be 8-bit flag. So the control information, preamble and postamble in synchronous transmission are typically less than 100 bits.

(Refer Slide Time: 22:18)



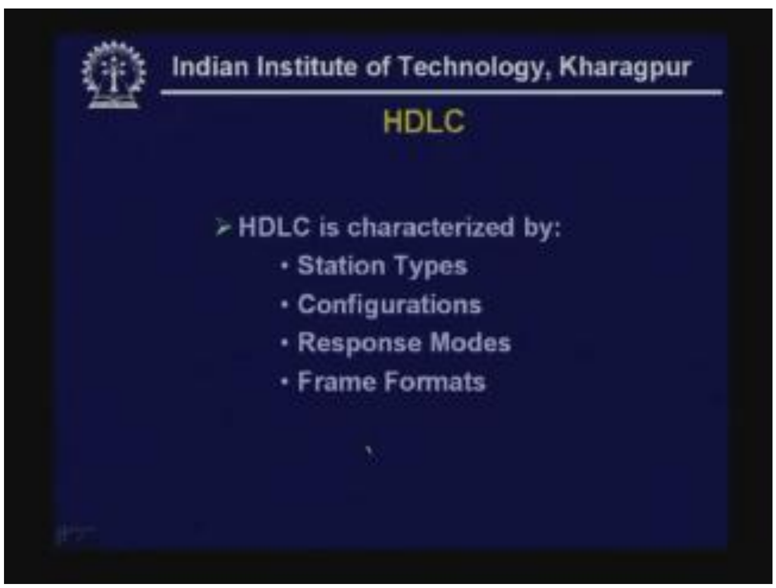
So these are essentially the overhead, this is restricted to less than 100 bits and this can be thousands of bits. Since this is thousands of bits and this is restricted to only 100 bits obviously the overhead is insignificant portion of the entire frame that is being sent so as a result that it gives you higher efficiency. Now let us focus on the most popular protocol that is being used for Data Link Control and this is known as High-Level Data Link Control or HDLC in short. It is one of the most important protocols as I mentioned and it is most widely used bit oriented protocol. There is a possibility of having two types of protocols; character oriented and bit oriented. But character oriented protocols are not popular so bit oriented protocols are used and HLDLC is one of them.

(Refer Slide Time: 24:36)

It is also the basis for many of the important Data Link protocols. Although HDLC has not been fully accepted in many protocols but some subset of it or a variation of it has been included in many protocols that's why it is the basis for many other important Data Link Control protocol. This particular protocol HDLC was adopted by ISO in International Standards Organization committee and also embraced by ITUT. So, as a consequence two important standards provider accepted this protocol and obviously this has to very widely accepted.

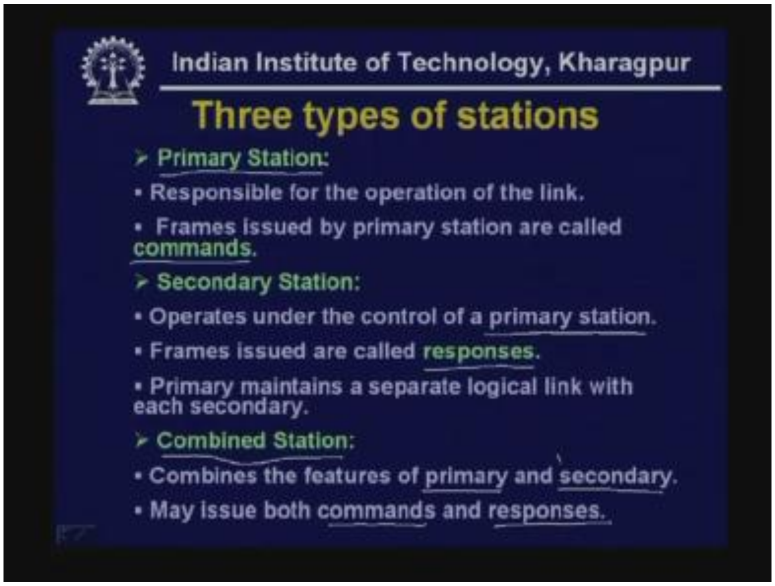
Some of the important characteristics are it supports both full-duplex as well as half-duplex communication also it can work in point-to-point and multipoint configurations. Point-to-point means it can be connected between two machines or you can have a number of machines connected at one point and it can be connected to other machines also that is one to many so multipoint so it can work in both the cases as we shall see.

(Refer Slide Time: 25:38)



This HDLC is characterized by four important parameters namely station types, configurations, response modes and frame formats. So we shall discuss each of them one after the other in little more detail. First let us focus on station types. There are three types of stations.

(Refer Slide Time: 26:11)

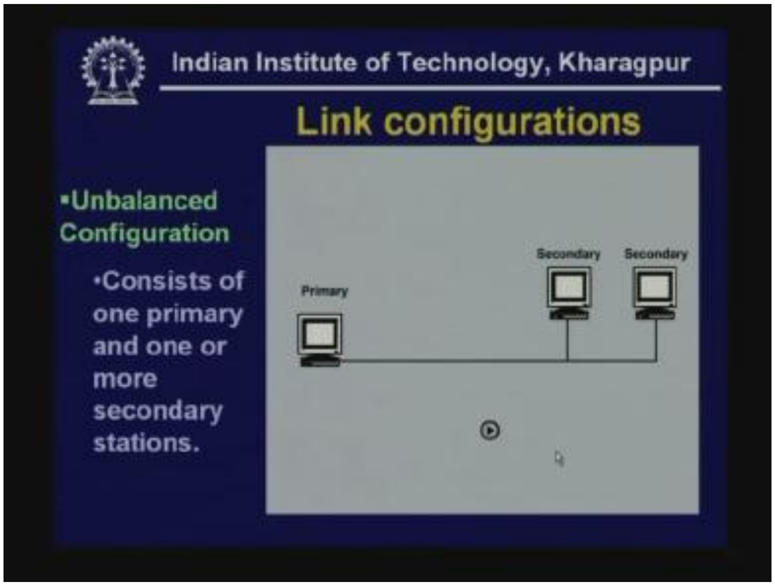


First one is known as primary station. Primary station is responsible for the operation of the link so it acts as some kind of master and frames issued by primary stations are called commands. That means the primary station acts as a master when data communication takes place between two machines and whatever is issued by the primary station is called commands so commands are issued by primary station.

On the other hand, secondary station operates under the control of primary station and frames issued by secondary stations are known as responses. So commands are issued by primary stations and responses are issued by secondary stations. And primary stations primary maintains a separate logical link with each secondary. That means primary maintains a separate logical link with each secondary so essentially link is established between the primary and secondary.

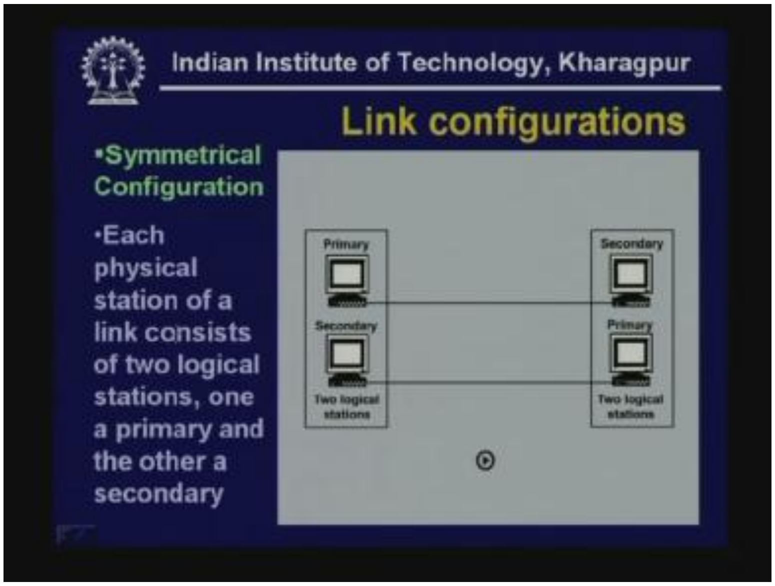
Now there is a there is a third type of station or machine which are known as combined station. So the combined stations combine the features of primary and secondary. We have seen that primary stations can issue commands and secondary can issue responses. But whenever it is a combined station it may issue both commands and responses. So, both can be issued by combined station. We have seen the three types of stations that can be used in HDLC.

(Refer Slide Time: 27:44)



Now let us consider the link configurations. There are several link configurations like unbalanced configuration which is the first one. So, in the case of unbalanced configuration as you can see it consists of one primary and one or more secondary stations. So whenever you have got one primary and one secondary it can be point-to-point or whenever you have got one primary and several secondaries then it can be multipoint. So let us see how communication takes place. As we have already mentioned primary issues a command and in the multipoint configuration the command can go to both of them. Obviously one of them will respond so the response is coming from the intended secondary which goes to the primary. This is how the unbalanced configuration works.

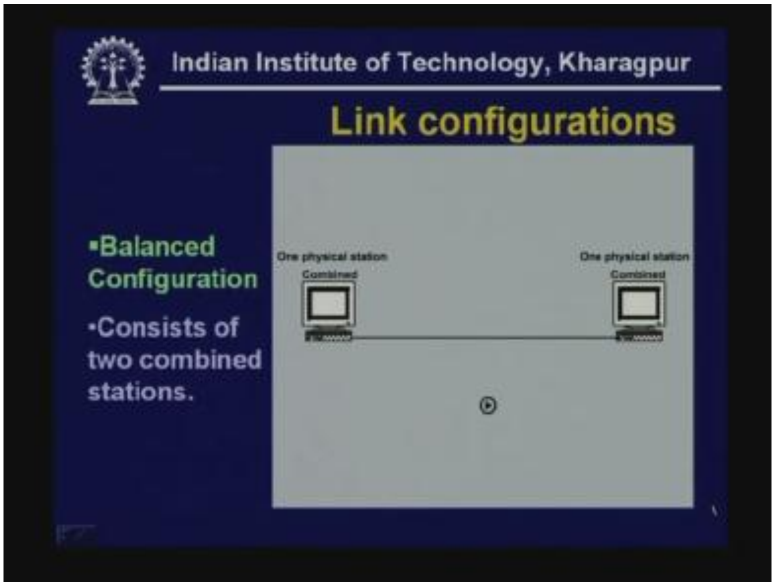
(Refer Slide Time: 28:42)



The second link configuration is known as symmetrical configuration. In symmetrical configuration each physical station of a link consists of two logical stations one primary and one secondary. So you see here you have got a single physical station however functionally it has got two logical stations. That means one is primary and another is secondary and as a result here you have got a symmetrical configuration, the secondary of this side. So these two logical stations communicate with another station which is also having two logical stations so you require two different links and here the primary can issue commands and the other side the secondary will issue response. Similarly the command can be issued by the primary of the other side and obviously the secondary of this side can issue responses. This is how the communication takes place in symmetrical configuration.

So, in symmetrical configuration as we can see you have got two logical stations within a single physical station and the communication takes place in this manner. Coming to the third configuration which is known as balanced configuration consists of two combined stations. So here you have got one physical station and one physical station and both are of combined type.

(Refer Slide Time: 30:05)



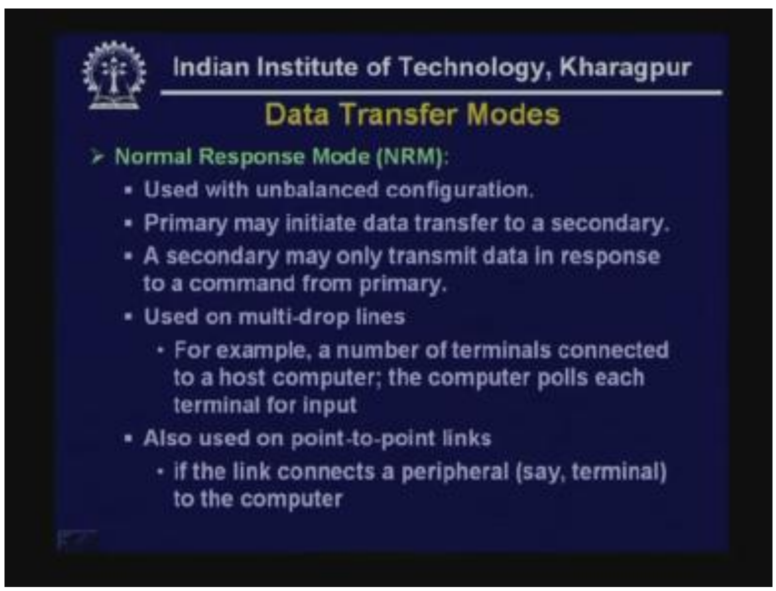
Since both of them are combined type, both have the capability of issuing commands as well as responses. Now a command has been issued by this station (Refer Slide Time: 30:37) which goes to the other side and the other side will issue response for the corresponding command. Similarly this side can also issue a command and this side will in turn generate a response. So a single combined station can issue both commands and responses and that's how the exchange of information takes place in balanced configuration. This is how the balance configuration works using two combined stations and here it is always point-to-point.

Coming to the third important parameter that is the data transfer modes there are three data transfer modes such as Normal Response Mode NRM, Asynchronous Balanced Mode ABM and Synchronous Response Mode ARM. These modes work in different ways as they are given here.

(Refer Slide Time: 31:11)

First let us consider the Normal Response Mode NRM.

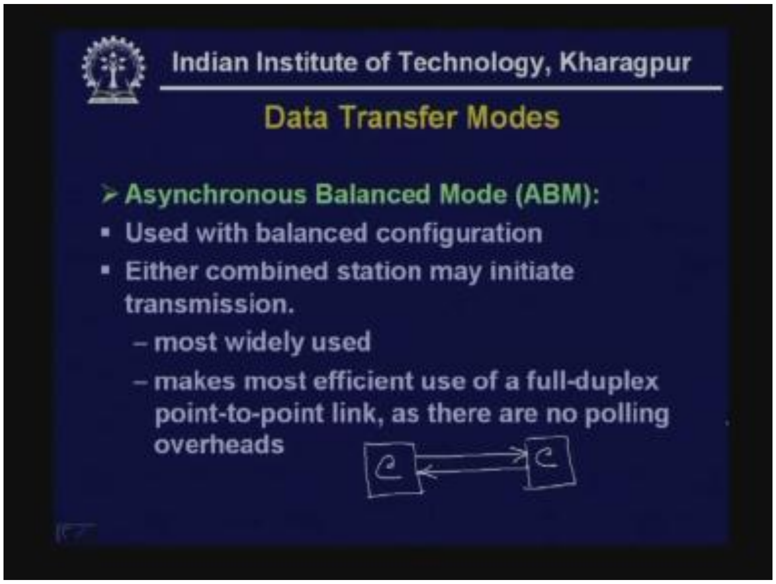
(Refer Slide Time: 31:52)



This is always used in unbalanced configuration. That means there will be one primary and you can have one or more secondary. So primary may initiate a data transfer to a secondary, a secondary may only transmit data in response to a command from the primary. So this secondary will be able to transmit the data when it is pulled or when it is asked to give some response. However, it can be used in multi-drop lines also.

As I mentioned there can be one primary and several secondaries. So in this case a number of terminals are connected to a host computer this is an example of multi-drop lines. So you can have a computer this can be your server and then you can have a number of terminals so these are terminals (Refer Slide Time: 32:55) so this is the situation for this Normal Response Mode. What I have shown here is a logical diagram but you can use a …….. sort of thing. So the computer polls each terminal for input and we have already discussed how the polling can be done. It is also used for point-to-point links. This can be used for both point-to-point as well as multipoint as I mentioned. This is the point-to-point where a computer is connected to peripheral. This is a computer and this is a printer. so this is essentially the master and this is the primary and this is the secondary in the HDLC terminology and the communication is in the unbalanced mode.

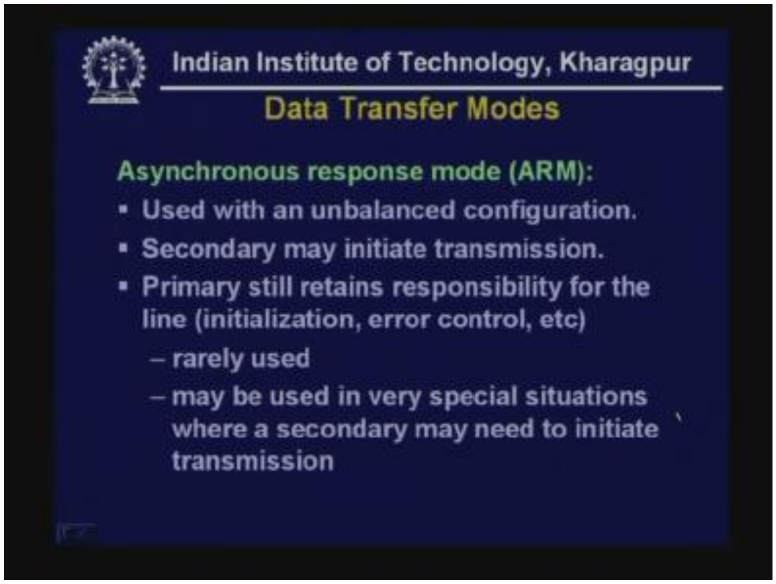
(Refer Slide Time: 33:53)



Coming to the second data transfer modes asynchronous balanced mode this is used in balanced configuration. This asynchronous balanced mode can be used as its name implies it can be used only with balanced configuration so either combined stations may initiate transmission. So here you have got two machines and both of them are combined type and obviously in this case data transmission can be initiated by either of them and this is possibly the most widely used technique and it makes most efficient use of a full-duplex point-to-point link as there are no polling overheads.

As we have seen in case of unbalanced mode if it is a multipoint then polling overhead is there. But in this particular case you can have two machines connected by full-duplex link so data can flow in this direction so there is no need for any polling and both of them can be combined machines. This is how the asynchronous balanced mode works.

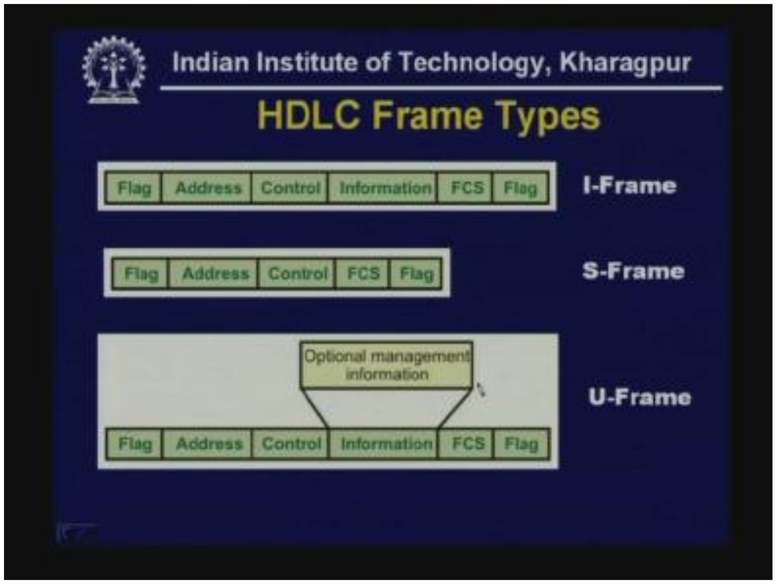
(Refer Slide Time: 35:04)



The third type is Asynchronous Response Mode or ARM. This is used with an unbalanced configuration, secondary may initiate transmission, primary still retains responsibility of the line for example initialization, error control these are all done by the primary however in this secondary may initiate transmission in the Asynchronous Response Mode which is not possible in the first mode.

However, this is rarely used and may be used in very special situations where a secondary may need to initiate transmission. Normally all the data transfer takes place with the initiation of the primary but in this Asynchronous Response Mode the secondary may also initiate data transmission so whenever this kind of situation is required in that case Asynchronous Response Mode can be used. So, coming to the fourth important parameter that is your frame types we have got three different frames namely I-frames which is the Information frame, S-frame or U frame

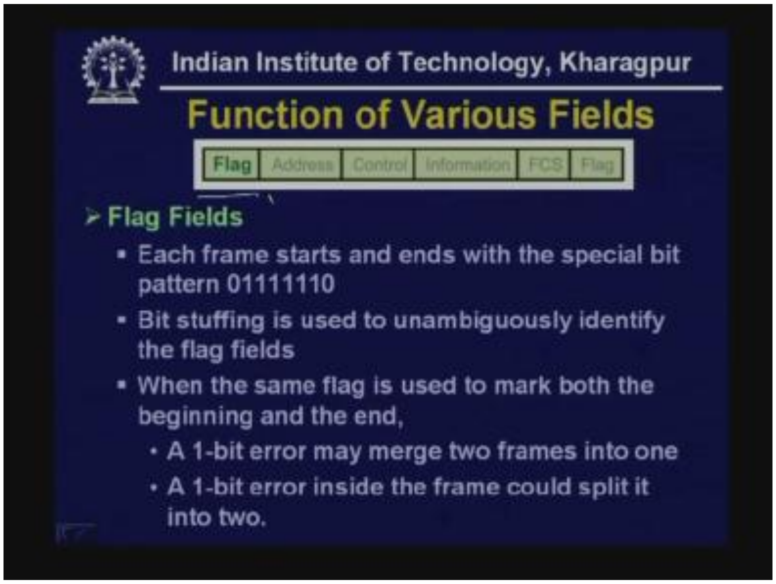
(Refer Slide Time: 36:10)



So there is I-frame, S-frame and U-frame, so this is your unnamed frame (Refer Slide Time: 36:35), S stands for secondary frame and I stands for information frame. As you can see here you have got flag, address, control, information and Frame Check Sequence and flag. This is the typical format of the I-frame. However, S-frame is not having information part but it is having all the other fields the flag, address, control, information, Frame Check Sequence and flag and in case of unnamed frame or the U-frame you will find there is a flag, address, control and information part is optional and this information is provided for management and control so this is not really the user data. In the case of I-frame this is user data but here it is used for management and control. Whenever some information has to be sent for management and control it is provided here. So you have got flag, address, control, this optional information, Frame Check Sequence and flag. So these are three different types of frames possible. Let us consider each of these fields separately one after the other.

First let us consider the flag field.

(Refer Slide Time: 37:55)



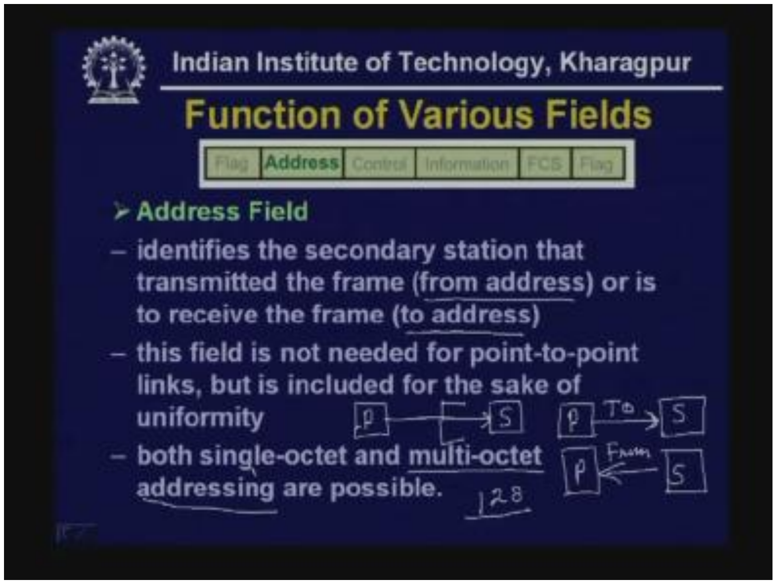
Each frame starts and ends with the separate bit pattern that is your 01111110 so you have got six ones and two zeros planked by two zeros on both sides so six ones planked by two zeros on both sides acts as a flag which is actually the starting and ending frame and which signifies the start and end of a particular frame. Now this bit pattern may appear as part of the information or it may appear as part of control so in such a case what has to be used.

Whenever we are sending the bit pattern 0 then six ones and then 0 which appears as part of data then there is a possibility that if it appears in the middle of information bit that can be considered as the ending flag. So as a consequence this will divide the single frame into two frames so this problem has to be avoided and that is being avoided by bit stuffing so that there is data transparency to unambiguously identify the flag fields.

So the flag fields should not be present in the information bits. So, to unambiguously identify the flag fields bit stuffing is used. As you can see here (Refer Slide Time: 39:40) as you are using the same flag when the flag is used to mark both beginning and the end. There are two situations; A 1-bit error may merge two frames into 1 and a 1-bit error inside the frame could split it into two, this can happen. However, when no error occurs there is no problem as you are using Frame Check Sequence for detecting errors.

Now let us consider the address field.

(Refer Slide Time: 40:08)



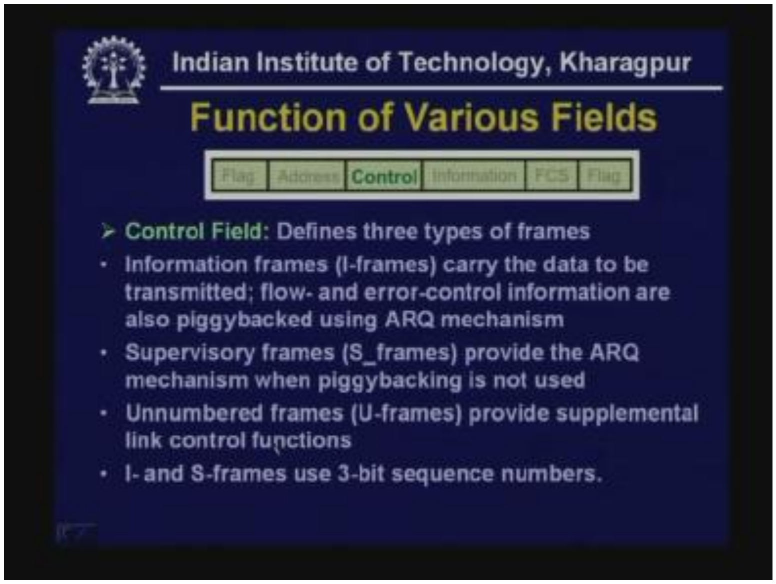
This address field identifies the secondary station that transmitted the frame. That means whenever a secondary station sends a frame it gives its own address known as (from address). That means whenever it is going from a secondary to a primary then it is called (from address). That means where from it is coming is provided as part of this address. On the other hand, there is another situation, whenever it is going from a primary to a secondary, so this is your primary and this is your secondary (Refer Slide Time: 40:48) so when it is going from here to here then the address is essentially (to address). So here it is (to) and here it is (from) so in the same address field you can have either two types of addresses (to) address and (from) address depending on information transfer that is taking place.

Obviously there is only one primary in the system so there is no need for its address to be part of the frame so only the address of secondary is needed that's why this (from) address and (to) address approach is used. So this field is not required for point-to-point links but is included for the sake of uniformity.

Whenever it is point-to-point you have got only two stations one primary and the other secondary so there is no need for any address because there is only one. But when it is multipoint there are several secondaries in such a case address has to be used and since it is of 1 byte you can have at most 128 different types of secondaries. So this address whenever it is single octet or 1 byte then you can have 128 addresses but there is a provision for multi-octet addressing in such cases obviously the number of machines you can have in a multipoint communication can exceed 128 so there is provision for multi-octet addressing but single octet is the most commonly used one.

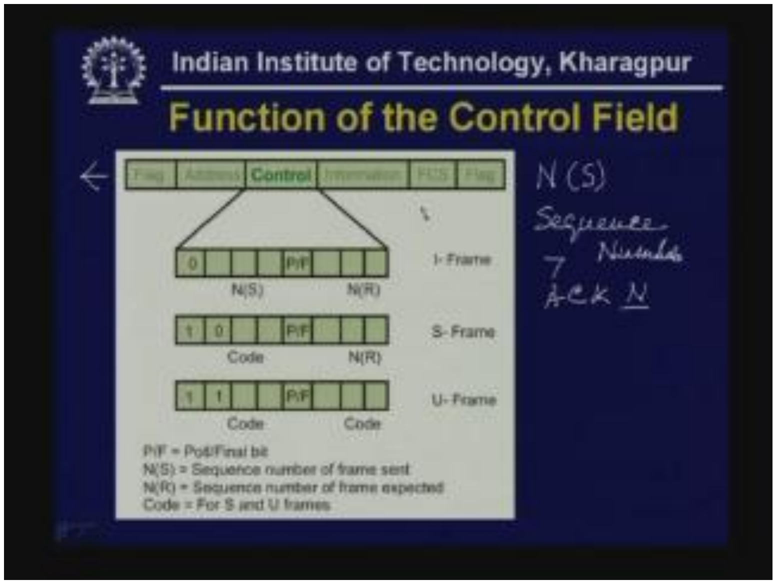
Now let us focus on the third field that is the control field. this control field defines three types of frames namely information frames which can carry the data to be transmitted, flow and error control information are also piggybacked using ARQ mechanism then supervisory frames provide the ARQ mechanism when piggybacking is not used, unnumbered frames provide supplemental link control functions and in case of I and S frames they use 3-bit sequence number as we shall see so the flow and data control used here will be explained how it is being done.

(Refer Slide Time: 42:34)



Let us look at the control frame format. First of all control frame decides what type of frame it is. For example, if it is going in this direction it is the first bit of the control field if it is 0 then it is I-frame or information frame. And as you can see information has got two 3-it numbers NS and NR. NS is actually the sequence number of the station which is sending the information. So it is the sequence number of the station which is sending the information so it is 3-bit.

(Refer Slide Time: 44:18)



Obviously if we use go-back-N flow control and ARQ then the total number will be 7. That means 7 possible sequence numbers can be used. And NR is used for piggybacking. NR essentially specifies the frame it is expecting. That means this NR tells ACK that N and that N value is provided here. So how piggybacking is done is whenever both sides are connected by using full-duplex communication and two stations are connected by using full-duplex communication then piggybacking can be used and acknowledgement information can be provided as part of the information frame. This is how these three bits are used for that purpose. So you can provide ACK as well as reject other frames as part of this.

However, when there is no data to be sent by the station that means whenever there is not data to be sent by the secondary then it can send one S-frame. Hence in case of S-frame there is a code and it is decided by one 0 so in that S-frame there is no data field. However, it has got the two bit code then this bit stands for poll and final.

Whenever you are using a secondary it is used in this case information frame also and whenever you are sending you have got a primary and a secondary and whenever primary sends it actually this bit is 1 so this signifies the polling. That means when it is going from primary to the secondary. On the other hand, when it is coming from the secondary to primary and if it is 0 then data is coming from secondary to primary and whenever it is 1 that signifies the final frame that is coming. So in this way the sequence of frames can come from secondary to primary so polling is performed with the help of these bits.

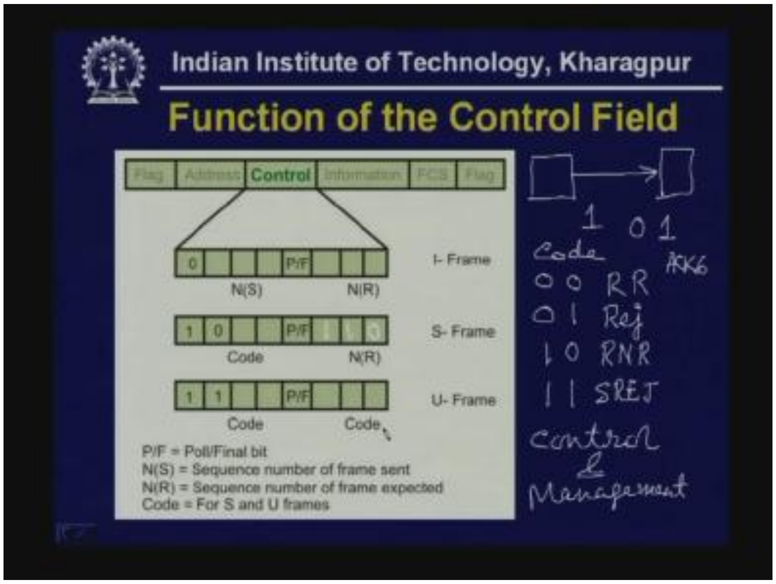
And the secondary can signify (Refer Slide time: 46:59) with the help of this code the different types of information that is being provided as part of that NR. For example, when the code is 0 0 this stands for RR that means Receive Ready. That means Receive Ready is essentially that whenever the receiver is sending an acknowledgement which is provided as part of NR. So NR is providing that acknowledgement then it is RR Receive Ready for receiving that particular frame.

For example, ACK6 means 6 is provided here so here you have got 1 1 and 0 so 6 is written here and then Receive Ready so an acknowledgement frame is being sent. Or it can be say 0 1 then it is reject. that means whenever negative acknowledgement is being sent a particular frame has to be rejected so NAK for example it is reject then it is 1 1 and 0 once again. That means here in case of go-back-N ARQ the frame has to be repeated from the frame number 6 so this signifies the frame which is in error which has to be retransmitted that is the case whenever it is 0 1 that is reject. So it specifies the function and the frame number is provided here. That acknowledgement number is provided here. Then if it is 1 0 then it is Receive Not Ready RNR. That means whenever the other side is not ready then the data cannot be sent. So you see here you can perform flow control. If the secondary is not ready in such a situation RNR frame can be sent that supervisory frame can be sent by the secondary to the primary.

Finally whenever it is 1 1 then it can be S reject SREJ. That means whenever the selective-repeat ARQ can be used in such a case a particular frame number which is mentioned here has to be selectively repeated that is being specified with the help of this bit. So this is how the S-frame with the help of this code can mention about different types of functions of that supervisory frame which is coming from the receiving side to the transmission side. So whenever a secondary has no data to send piggybacking cannot be done so in such a case this negative acknowledgement or acknowledgement can be sent in this manner.

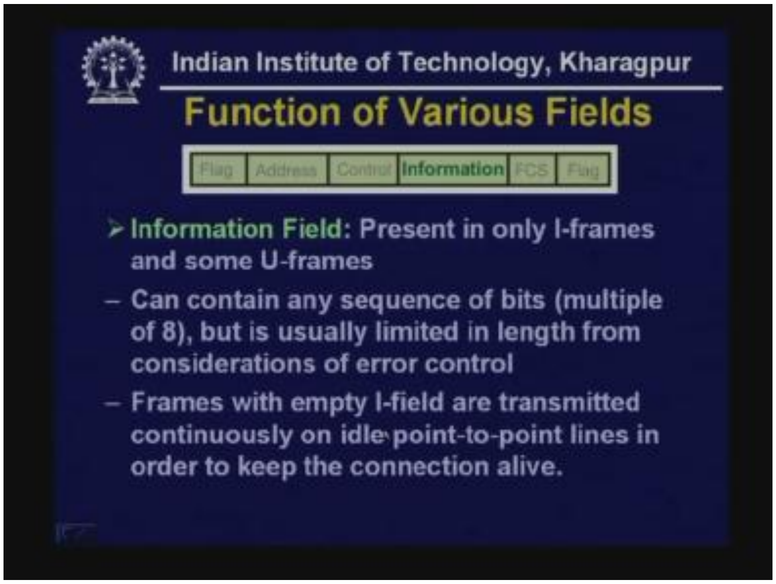
Finally in case of U-frame again the code is there which is used in S-frame and U-frame and the three more bits are used for coding. These frames are essentially for used for control and management purposes.

(Refer Slide Time: 43:32)



Now coming to the information field it is present only in I-frames and also in some U frames as I mentioned. And whenever it is present in I-frame it is user data that is being sent and when it is used in U-frame it is management and control information. So it can contain any sequence of bits but that has to be multiple of 8 but is usually limited in length from consideration of error control.

(Refer Slide Time: 50:30)

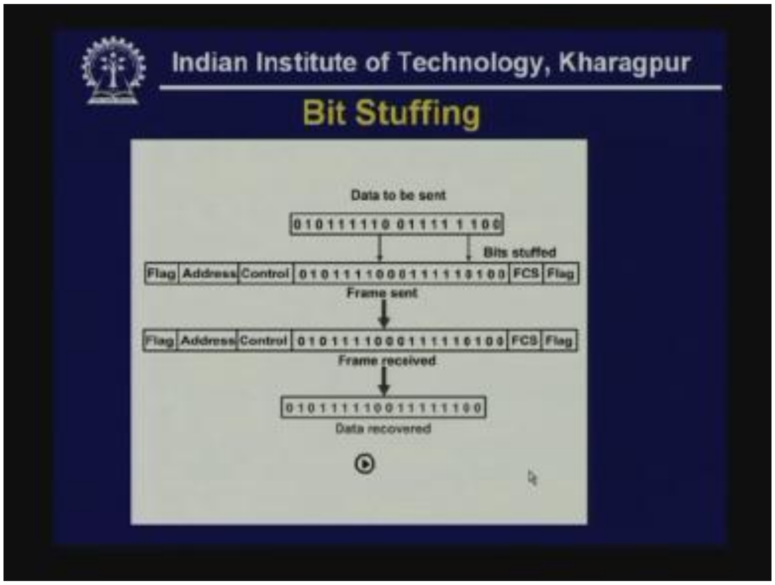


As I have mentioned although you can send unlimited number of bits it is restricted so that the frame does not suffer from error. That means if one bit gets corrupted then again retransmission has to be done so to avoid that it is better to restrict the length so that the error does not occur and probability of error increases with the length. Longer the length more the possibility of error is more. Frames with empty I field are transmitted continuously on idle point-to-point lines in order to keep the connection alive.

So, whenever a particular station has no data to send but a particular session has started in such a case frames are sent continuously with no information in the information field that is also possible. Particularly this is necessary to keep this connection alive otherwise the link will be disrupted.

Now as I mentioned that your information part may contain the flag which is your 0111111 and 0 and this should not appear in the information field. That means information field should not contain a flag character. So, to do that, bit stuffing is used.

(Refer Slide Time: 52:07)



Bit stuffing is explained with the help of this animation. Let us assume that this is the sequence of bits or data to be sent present in that information field (Refer Slide Time: 52:44). So, what is being done is after high consecutive ones a 0 is forcibly introduced so here you can see 11111 after five ones a 0 is introduced and here also 1111110 is introduced. Thus after each occurrence of five consecutive ones a 0 is introduced known as bit stuffing. So here you see a flag at this control and after this a 0 bit has been introduced and here also a 0 bit has been introduced so bit stuffing has taken place and this is the frame to be sent with two extra zeros and this is the frame that is being sent and this is the frame received.

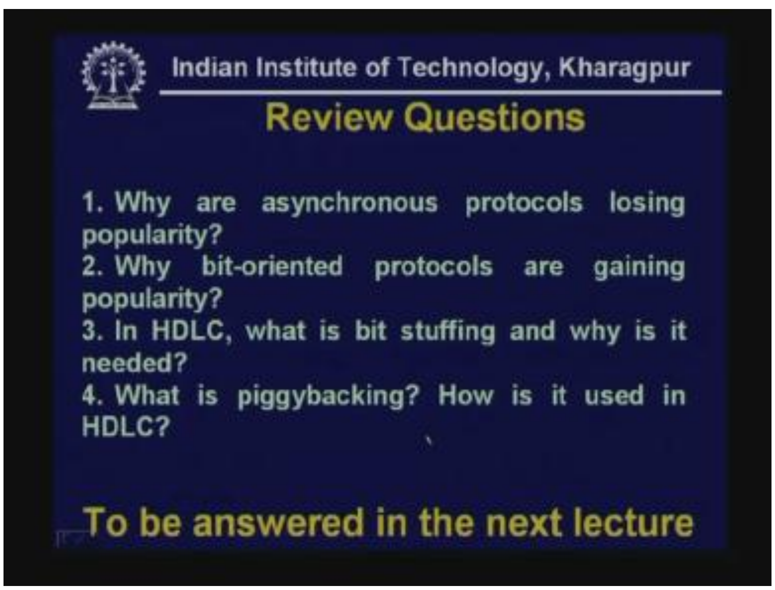
At the receiving end the receiver knows that after each five ones there is a so these zeros are replaced to get back the data. So the receiver will remove that redundant zeros to get back the data. This is how the frame is recovered at the receiving end. This is how the bit stuffing is done for data transparency.

Finally we have the Frame Check Sequence which can use either 16 or 32 bit CRC code computed using address, control and data fields for the purpose of error detection using the Cyclic Redundancy Code. We have already discussed about it in more detail. Now as I mentioned several protocols have been developed based on HDLC known as Link Access Procedure Protocols and LAPD and LAPB, LAPM are the three examples. This LAPB works in ISDN and uses Asynchronous Balanced Mode of transmission to connect two devices in a combined type. Link Access Protocol D is used in ISDN using ABM Asynchronous Balanced Mode that is in both the cases we find it is used for point-to-point link. The link access procedure for modems used in modems essentially is used to design synchronous and asynchronous conversion error detection and retransmission. These functionalities are not provided in dcdt interface so this is added on top of the dcdt interface to provide the HDLC features.

(Refer Slide Time: 55:40)

So now it is time to give you the Review Questions.

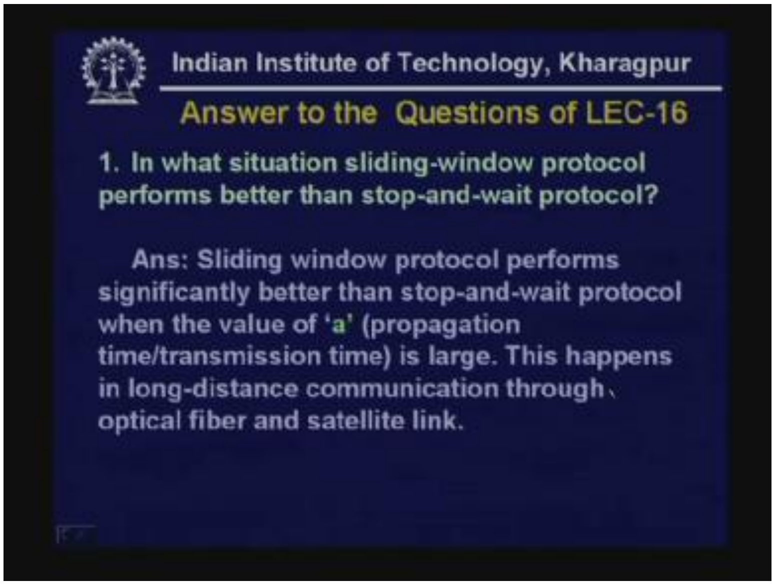
(Refer Slide Time: 56:10)



1) Why are asynchronous protocols using popularity?
2) Why bit oriented protocols are gaining popularity?
3) In HDLC what is bit stuffing and why is it needed?
4) What is piggybacking? How is it used in HDLC?

Now it is time to give you the answers to the questions of lecture minus 16.

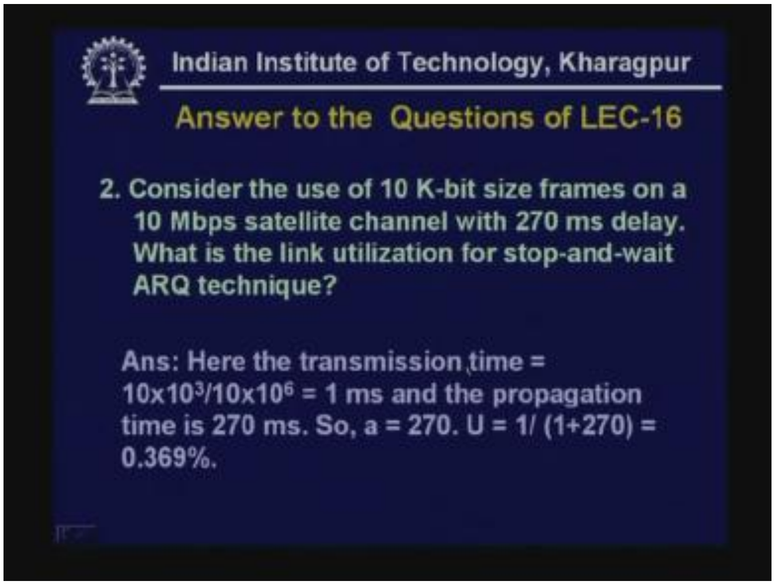
(Refer Slide Time: 56:21)



1) In what situation sliding-window protocol performs better than stop-and-wait protocol?

Obviously sliding-window protocol performs significantly better than stop-and-wait protocol when the value of 'a' propagation time by transmission time is large. this happens in long distance communication through optical fiber and satellite link or when you are using high speed communication.

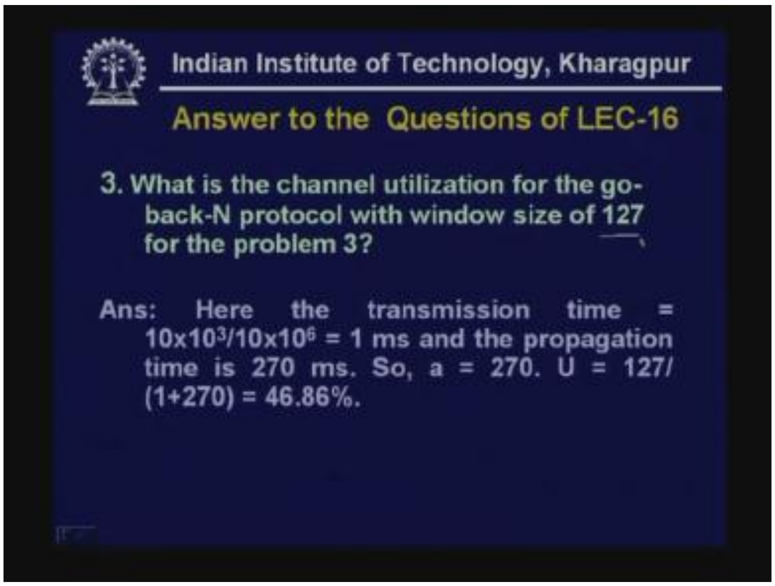
(Refer Slide Time: 56:55)



2) Consider the use of 10 kilo bit size frames on a 10 mega bit satellite channel with 270 millisecond delay. What is the link utilization for stop-and-wait ARQ technique?

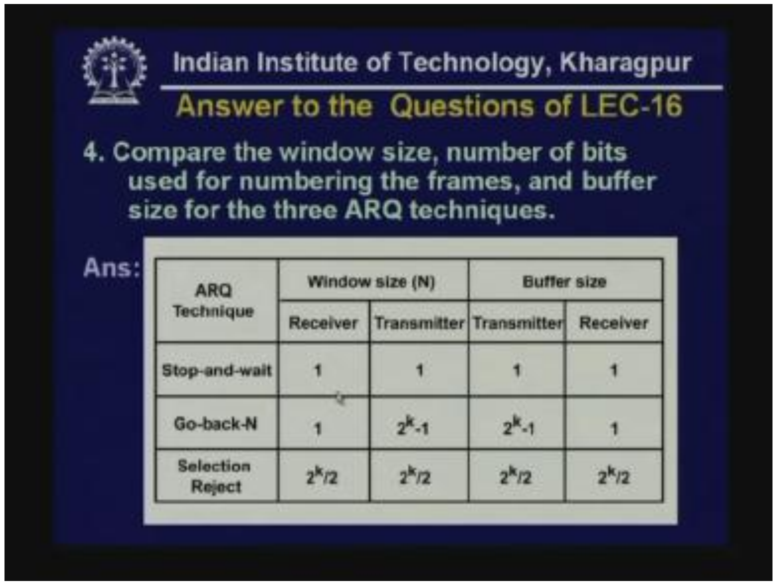
So here the value of a equal to 270 because transmission time is 1 millisecond, propagation time is 270 millisecond so utilization is only 0.369%. On the other hand, when you are using go-back-N ARQ with window size 127 that means you are using 8 bit for the frame number so in such a case the value of a is equal to 270 and obviously the utilization is 46.86%.

(Refer Slide Time: 57:25)



And if we use more number of bits that means larger window then it can be even 1.

(Refer Slide Time: 57:52)



ARQ Technique	Window size (N)		Buffer size	
	Receiver	Transmitter	Transmitter	Receiver
Stop-and-wait	1	1	1	1
Go-back-N	1	2^k-1	2^k-1	1
Selection Reject	$2^k/2$	$2^k/2$	$2^k/2$	$2^k/2$

3) Compare the window size, the number of bits used for numbering the frames and buffer size for three ARQ techniques.

So, in stop-and-wait window size is 1, buffer size requirement is 1, in go-back-N receiver has got window size of only 1, transmitter will have window size 2 to the power k minus 1, buffer size in transmitter is 2 to the power k minus 1 and receiver will have only 1.

In case of selection reject however as we know the receiver and transmitter both will have window size 2 to the power k by 2 and buffer size also 2 to the power k by 2 both for transmitter and receiver. So with this we come to the end of today’s lecture, thank you.