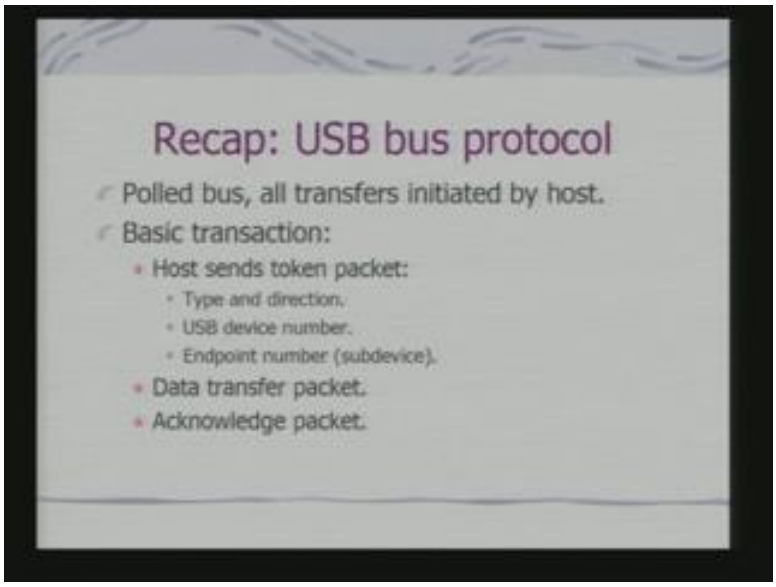
Embedded Systems
Dr. Santanu Chaudhury
Department of Electrical Engineering
Indian Institute of Technology, Delhi

Lecture - 17
Serial Interfaces

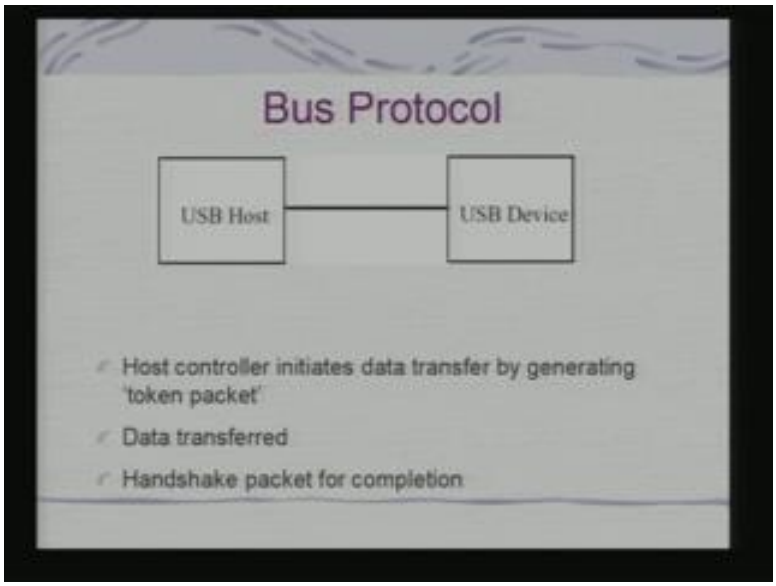
In the last class we had started looking at interfaces by which embedded Systems can be connected to the host we were looking at USB interface. Today, we shall look at USB IEEE 1394 as well as IrDA protocols for connecting devices to each other as well as to the host.

(Refer Slide Time: 01:33)



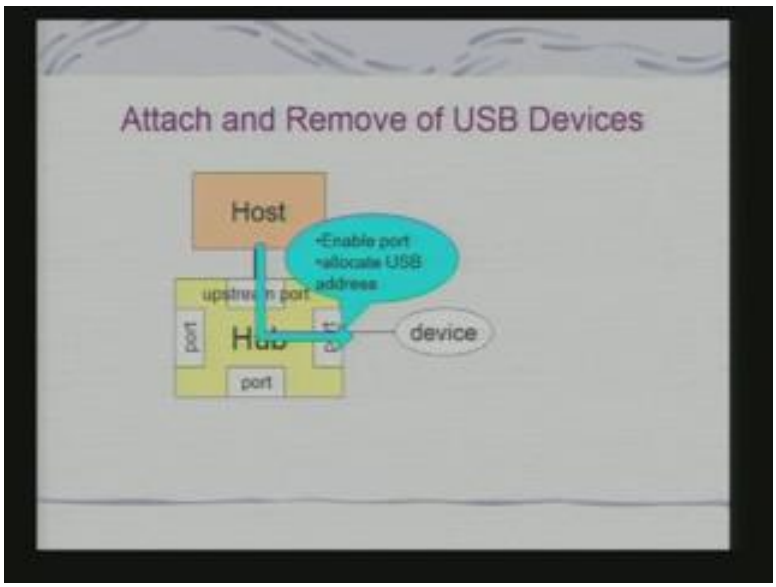
So, USB bus is basically a polled bus. In fact, here the all transfers are initiated by the host. The basic transaction steps are following host sends token packet which actually tells a type type of the data transaction that is suppose to take place as well as a direction. It also has the USB device number indicating the device for which the token is targeted and also the end point numbers, which is a sub device. Because USB devices can be even compound device which may correspond to more than one device also there is data transfer packets which flows on the bus as well as acknowledge packets.

(Refer Slide Time: 02:32)



So, basically this is how the whole operation works this is a USB host which initiates a token. These as got type of transaction direction of transaction and USB device address. So, once that is perceived depending on the type of transaction the data transfer takes place and the handshake packet is used for acknowledgement as well as completion for an transaction.

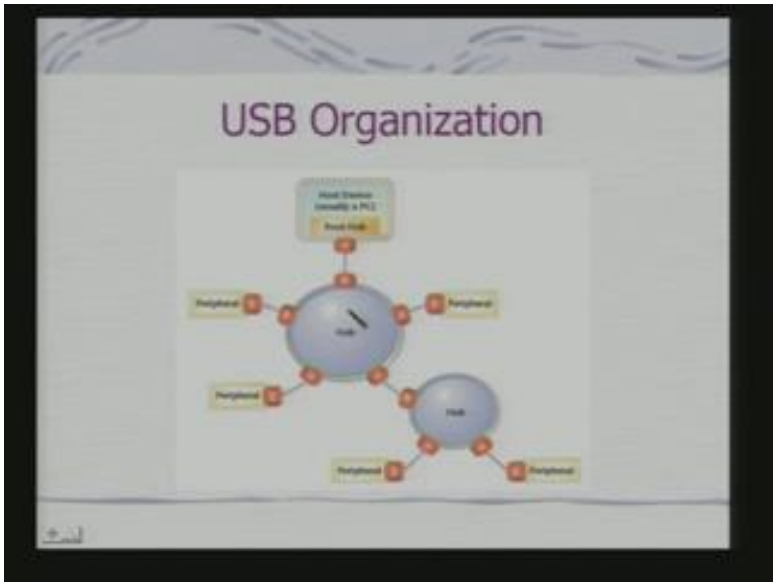
(Refer Slide Time: 03:06)



The another aspect of the USB bus protocol is that of device identification in the last class we had talked about the concept of enumeration. So, what happens is the device can

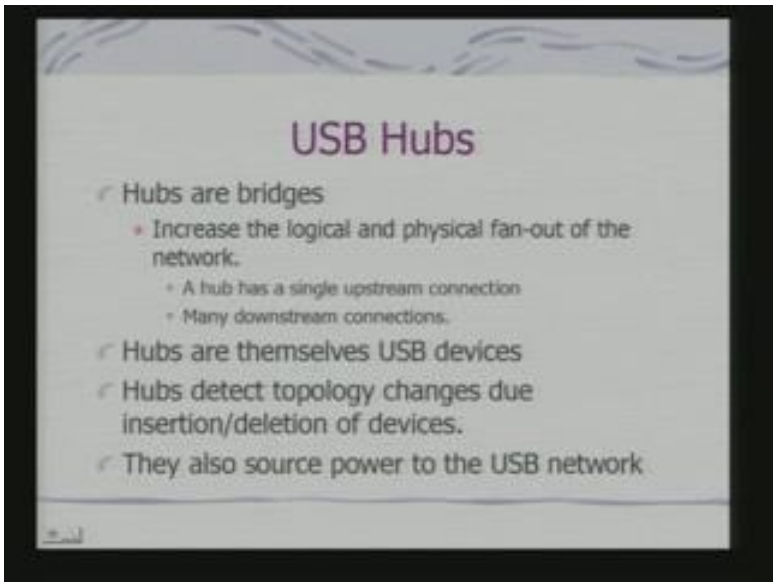
get connected to a Hub. Hub is nothing but something like a bridge in a USB system. So, device is connected to the port and once it gets connected to the port the indicator of the device flows via the port to the host. Now, what does host do? It enable the port allocate USB address to the device and also it identifies the device driver which is require for the device. If the devices removed the port actually gets disabled and you remove the host removes the indicator and host therefore, can sends automatically whether the device is removed or not.

(Refer Slide Time: 04:17)



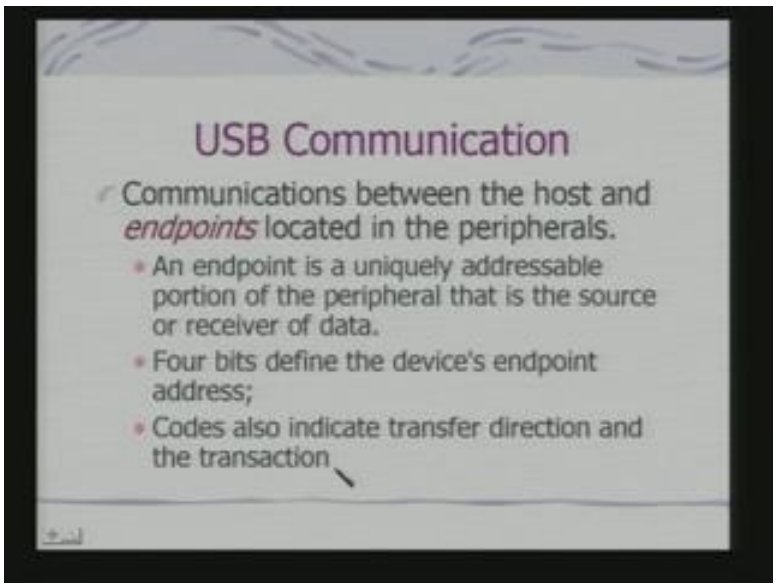
So, this USB organization we have also send in the last class. So, you have got these Hubs to which the peripherals are connected. So, it affects effectively increases the fan out of the network and Hubs are nothing but a bridge in a USB organization.

(Refer Slide Time: 04:35)



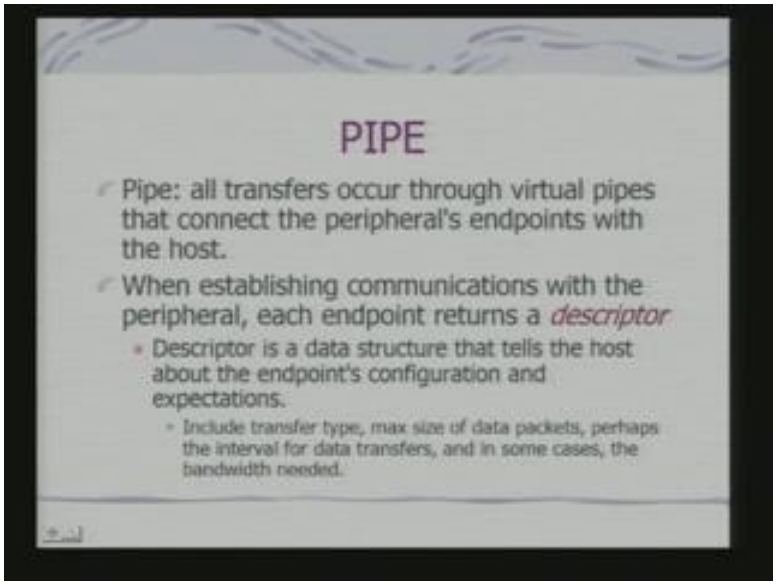
So, USB Hubs are bridges, which increase the logical and physical fan out of the network. A Hub typically has a single upstream connection and it can have many downstream connections. Hubs are themselves USB devices the Hubs detect the topology changes. We had already seen how this topology changes actually detected and propagated to the host and the host manages the USB device addresses with conjunction with Hub. They also other source of power to the USB network because USB devices meet not be separately powered if they are power demand is not substantial.

(Refer Slide Time: 05:34)



The USB communication takes place between the host and endpoints and endpoints are actually located in the peripherals. In fact, the end point each end point has got a unique address and therefore, if a peripheral consists of multiple devices then each of the device would have a unique address because each device will be make to an end point. In fact, this 4 points 4 bits define device's endpoint address and the codes also indicate transfer direction and the transaction type that is a token which is circulated by the host that indicate transfer direction as well as the transaction type. In fact, all transactions take place through pipes.

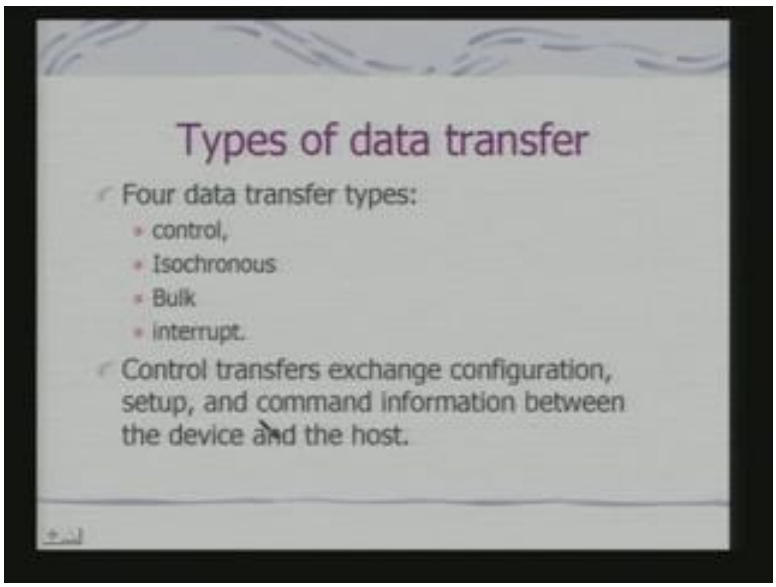
(Refer Slide Time: 06:26)



Pipes are like virtual connectors that connect peripherals end points with the host why it is a virtual connector? Because the path can go through a Hub, but logically there is only a single connection or a pipe between the device and that of the host. So, when establishing communications of the peripheral each endpoint returns what is called a descriptor. In fact, descriptor is a data structure that tells the host about the endpoints configurations and expectations. So, that guides the host to manage the transaction with the specific device that is get connected to the port. These includes in fact, this data structure has got specification for transfer type there are different types of data transfer which can take place on the USB bus. So, it indicates the transfer type. It also indicates the maximum size of data packets, because that would depend upon the capability of the client device which is getting connected to the bus. Perhaps also the interval for data transfers; that means the rate at which the data transfer is takes place as well as the

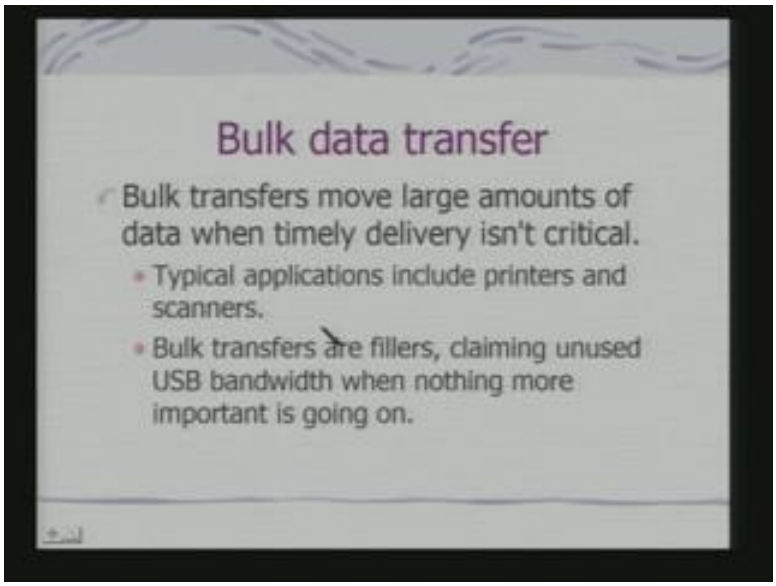
bandwidth needed. Why all these information's have to be provided by endpoint? Because it is not a preconfigured bus when a device get connected the connection is establish and the host by looking at the descriptor would know really the capability of the device and the connection. The device is looking for that enables the host to manage the device.

(Refer Slide Time: 08:23)



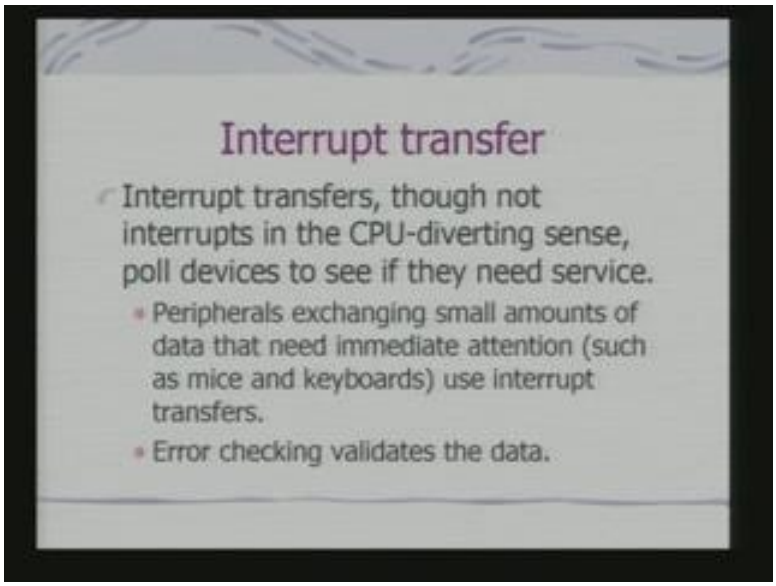
So, what are the different kinds of data transfer which can take place on the USB bus that of 4 types of transfer control, isochronous, bulk, interrupt. These are the different types which are formally identified control; obviously, is the transfer modality by which the configuration information, setup, and command information is exchanged between the host and the device.

(Refer Slide Time: 08:49)



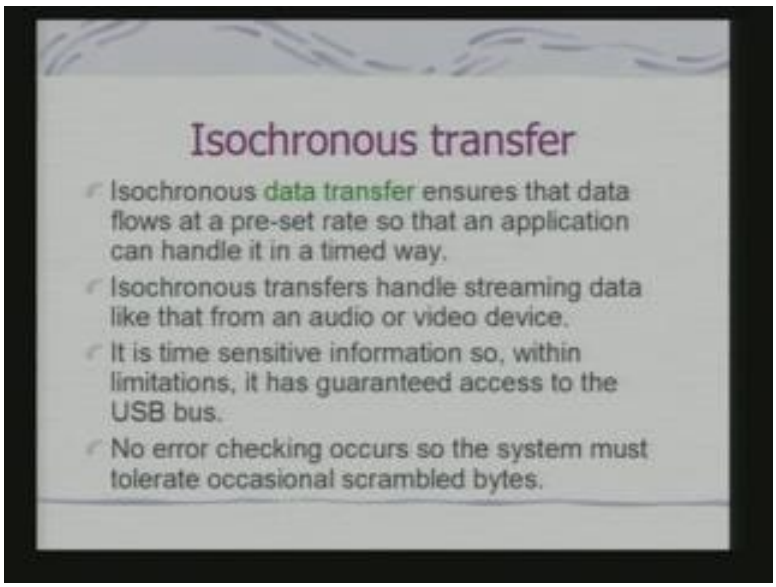
Bulk data transfer is that modality in which large amounts of data are moved, but these data really do not have time as a critical parameter. So, they are timely delivery is not really critical. So, examples could be printers as well as scanners. So, here we need to move large volume of data, but it is not necessary these data has to be moved with certain dead line in mind. So, in many cases bulk transfers become fillers claiming unused USB bandwidth when nothing more important is going on, because since there is no deadline to satisfy. So, it is not essential that the USB bus has to make a guaranteed bandwidth available to this transfer type. So, it can be put in whenever there is a gap, but large volume of data is to be transfer this is the basic aim.

(Refer Slide Time: 10:08)



The interrupt transfers I am not really associated with interrupting the CPU basically the whole objective of the interrupt transfer it, it should transfer small amounts of data small packets of data. So, peripherals which are exchanging small amount of data that need immediate attention such as mouse as well as keyboards use interrupt transfers. In fact, this is a type of a transfer you can see that this a fold bus the host initiates the transfer. So, interrupt transfers a periodically initiated by the host to check whether this device is required immediate attention or not and that is why they are called interrupt transfer.

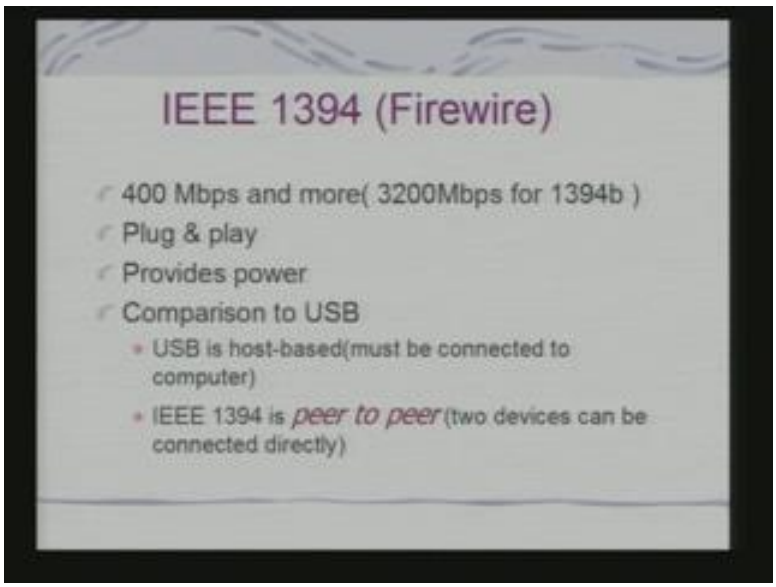
(Refer Slide Time: 10:58)



The last one is isochronous transfer what is isochronous data transfer. isochronous data transfer ensures that data flows at a pre set rate this is fixed rate. So, that an application can handle it in a time doing. So, isochronous transfer is primarily targeted for time critical delivery of data so; obviously, when we are streaming data from audio or video device we are required to support isochronous transfer. So, what is important is that in these context the USB bus has to provide guaranteed access. So, there has to be a guaranteed bandwidth allocated for these transfer. In these case no error checking occurs, because whenever there is an error checking an if there is an error detected. There would be a requirement of retransmission, but retransmission; obviously, cannot be ensured if I have a really critical deadlines to meet when I am transferring the data.

So, that is why with isochronous transfer typically you do not have error checking and system must tolerate occasional scramble bytes. If there are one or two bytes scrambled in a video frame which is being displayed it might not really affect the quality of the picture or even there can be software post processing to take care of this kind of scrambled bytes. In fact, this is a very basic principle which is we are discussing in the context of USB, but it is true for various other protocols which support isochronous transfers. Now, we shall look at another protocol which is IEEE 1394 this is again a serial protocol.

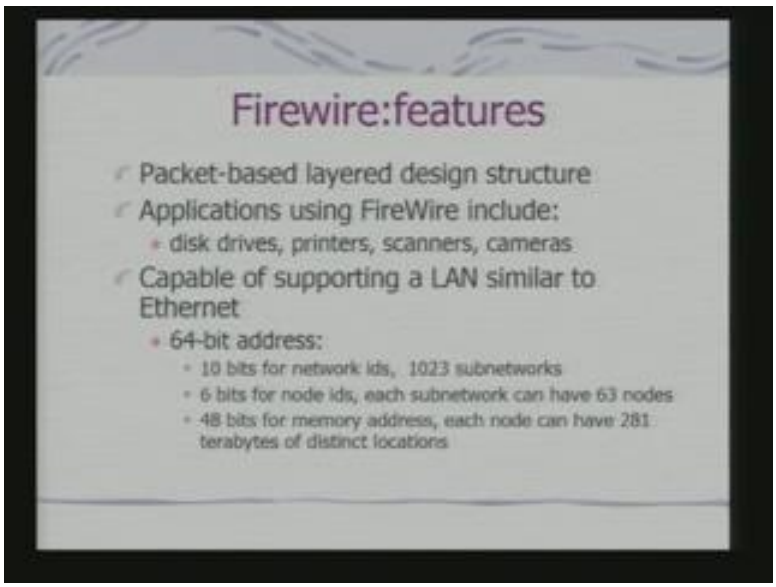
(Refer Slide Time: 13:14)



So, it is targeted for basically high speed transfers 400 Mbps and beyond an. In fact, this is what people at clubbing ends for the standard which is 1394b standard. This is a serial bus it has got the same plug and play capability just like in the USB bus. Whenever you connecting a device the device is recognized here also the target is to provide plug and play capability it also provides power. So, there is a power connection along with it for in comparison to USB the basic difference is here. USB is host based; that means, there has to be a computer to which a client device is connected, but IEEE 1394 is a peer to peer protocol, 2 devices can be connected directly. So, an embedded system is expected to support an USB interface when it is designed to communicate with the host typically you will find.

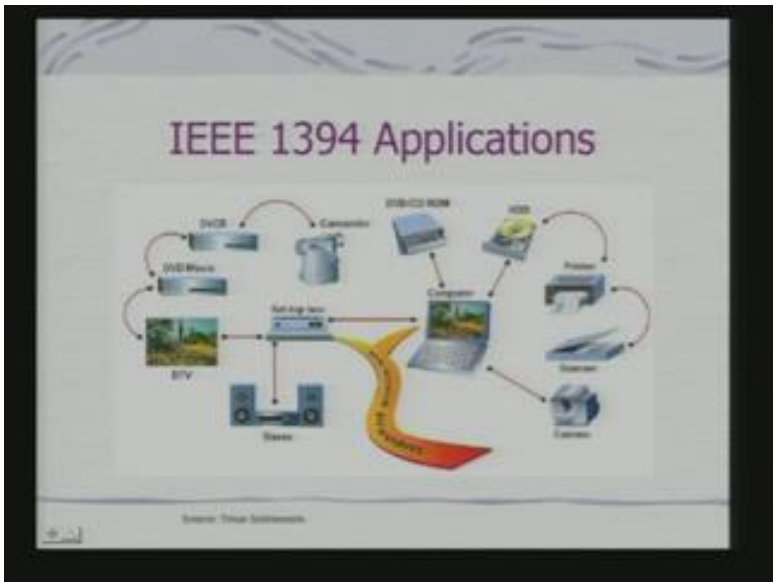
So, your PDA's or mobile phones the come with the USB cradle by which you connect to the host. So, that you can download software utilities your phone book all these things on to your PDA or mobile from the computer. So, in such cases and USB interface is expected this is also true for camera for loading the pictures. This is true for a digital recorder when you are loading a recorded music recorded music or a speech onto the computer. The other way round, also say an MP3 player with an USB interface can load the music, which you have obtained over the internet on your host space. On the other hand if you are talking about connecting two such embedded devices without involving a host or a PC. I would like to enable IEEE 1394 connection on such embedded Systems because then two such peer device can communicate with each other.

(Refer Slide Time: 15:39)



In fact, these Firewire IEEE 1394 is also known as Firewire. In fact, the original trade name because these whole standard was pushed forward by apple. They use the name Firewire later on when interest accepted and when in to and when into an IEEE standard it was term IEEE 1394. It is a packet based structure. So, it is a serial bus, but it has got more like a packet switching network. So, that what you transfer is a packets. So, it is not really a circuit allocated dedicatedly for a transfer operation. It application using disk drive printers scanners cameras there would be various of things which can be connected via this IEEE 1394. And it is capable supporting a LAN similar to Ethernet, but of a limited variety over a limited distance it has got 64 bits address. So, 10 bits for network IDs you can have 1023 sub networks 6 bits for node ids and each sub network can have 63 nodes, 48 bits for memory addresses each node can have 281 terabytes of distinct locations. So, directly device as are in these way addressable that is ports on the device as a directly addressable.

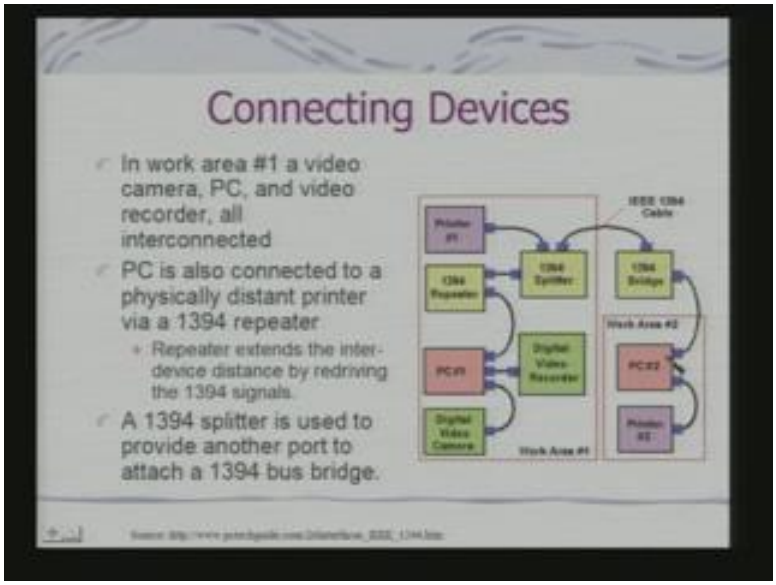
(Refer Slide Time: 17:19)



So, if you look into this the whole application this is a scenario which is for which is likely to be realized through your IEEE 1394 connection. And these all these are examples of your embedded system other than possibly a computer which is a general purpose computing device. But if you see here your stereo can be connected through IEEE 1394 to a set of box which gets connected to a DTV digital TV. It can be a DVD movie player it can be camcorder it can be when a separate hard disk communicating with the computer via your IEEE 1394 connections. So, in fact, if you see you can set up

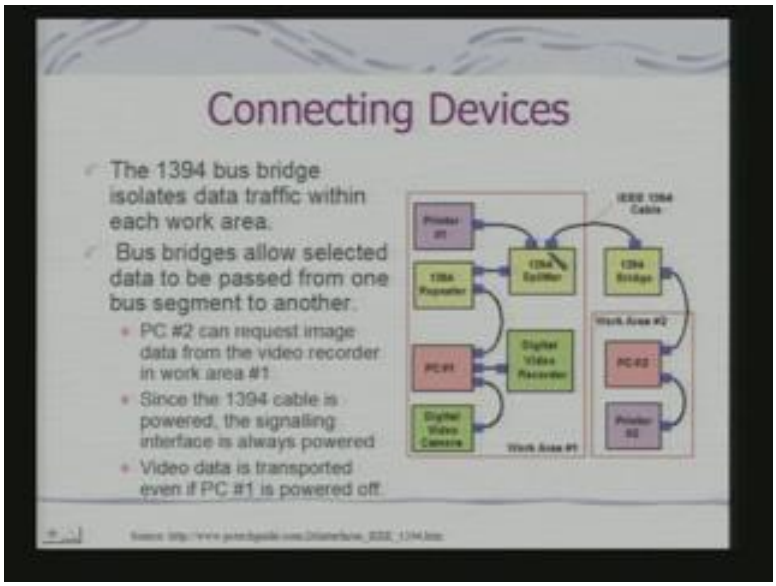
these kind of a complex network in your home environment or even in your office environment. And you can have may be a movie which your receiving on your digital TV which can be even put on to your hard disk. If you have really this kind of a connection built and this is really a peer to peer connection modality. So, let us see how such a network can be setup.

(Refer Slide Time: 18:33)



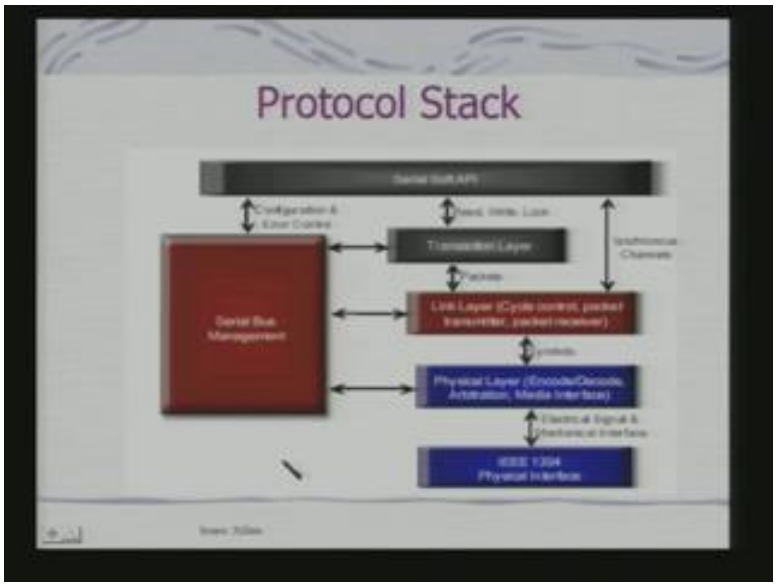
This is an example where we have identified two work areas work area one and work area 2 right in work area one a video camera a PC of video recorder all are interconnected. You have a digital video camera you have call got a printer you have got digital video recorder and this PC has got typically lets a three IEEE 1394 ports. So, they are connected and what it says the PC is also connected to a physically distant printer via 1394 repeater. Just like any repeater you use in a in for conditioning of the signal repeater extends the inter device distance by redriving the 1394 signals. So, it is a repeater and splitter is used to provide another port to attach a 1394 bus bridge. So, you have got the bus bridge by which you can connect other devices. So, this is another PC another printer.

(Refer Slide Time: 19:42)



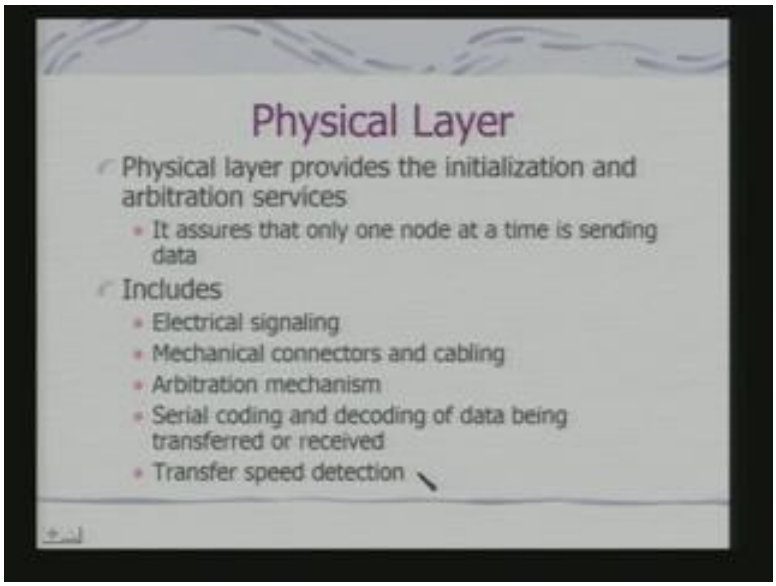
So, this bus bridge just like any other bridge in a network isolates data traffic within each work area. In fact, we have seen that also for any of the bridges that we use in the bus architectures. So, what does that mean? That means, I can have communication over this work area independent of the communication in the work area 2, but also the same time I can communicate between the two work areas. So, bus bridges allow selected data to be passed from one bus segment to another. So, PC2 can now request image data from the video recorder in work area 1. Since the 1394 cable is powered the signaling interface is always powered I said there would be power cading. So, video data is transported even if the PC is powered off. Although it is connected directly to the PC the whole the signaling interface which is expected to be powered on it shown. So, this it is not essential for this PC to work for the data to move from this digital video camera to this PC or the data to move from this PC to this printer for printing. So, you have actually got a device to device connection independent of the host.

(Refer Slide Time: 21:12)



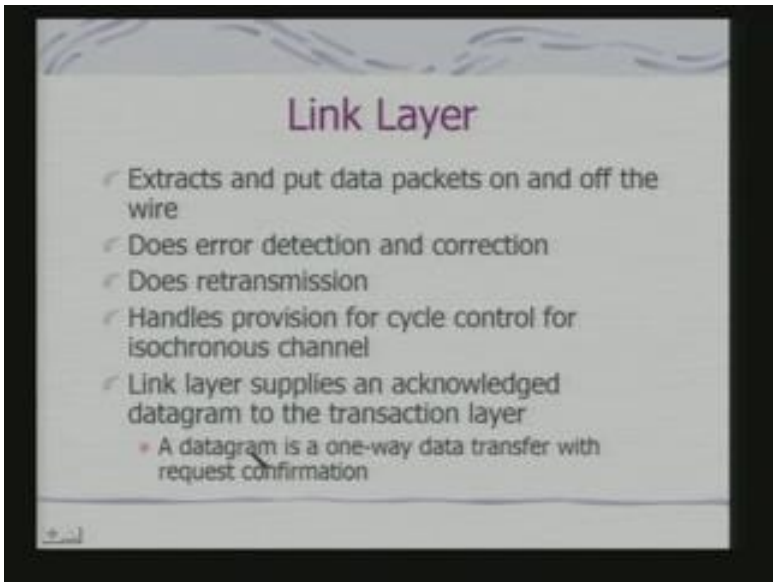
So, what is the protocol stacks? So, it is it has got a pretty pretty complex I should not say as complex as that of any network thin layers, but it is a pretty sophisticated protocol stack. So, other lowest level you have got the physical interface which provides a electrical signal and mechanical interface just like any bus structure we had discussed earlier. They knew of got a physical layer which provides the encoding decoding arbitration and media interface because the data when it has to flow ((refer time: 21:48)). It has be encoded and decoded appropriately they knew of got a link layer which controls a packet transmission and packet reception. This is a transaction layer which actually carries out the read write transactions over the bus and when it is an isochronous flow. We are already discussed isochronous flow that is media data delivery they not strictly read write transaction. So, under that circumstance is you really need that the transaction layer to come in play there would be direct transfer and that transfer is managed by the link way and all these things through a serial bus management system.

(Refer Slide Time: 22:37)



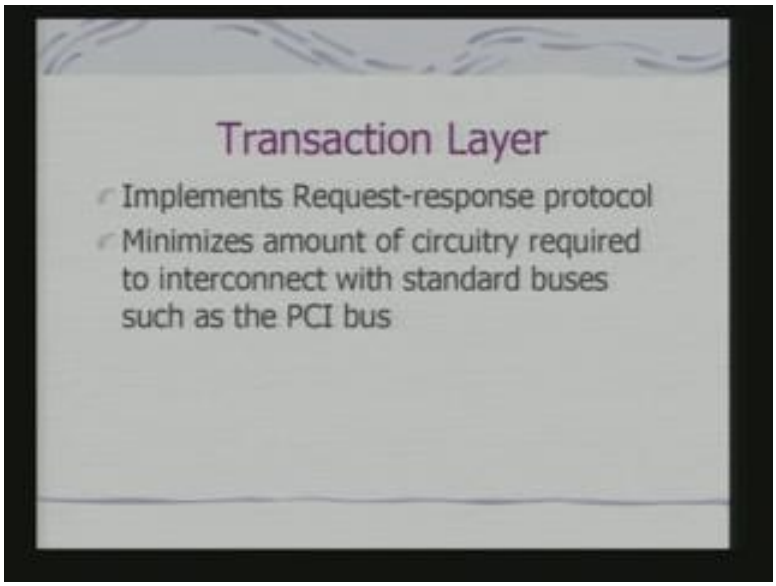
So, physical layer provides what we call the initialization an arbitration services. So, it assures that only one node at a time is sending data fine. Now, the other aspects which this physical layer management module includes or electrical signaling mechanical connectors specification of mechanical connectors and cabling. So, this is very important for any of these standards, because otherwise you cannot have devices going into the ports manufactured by a some other manufacture. This is truly when in the case of USB arbitration mechanism then serial coding and decoding of data being transferred or received as well as transfer speed detection. The speed at which really the device will respond unable to receive or send that it the link layer job is to extract and put data packets on and off the wire.

(Refer Slide Time: 23:37)



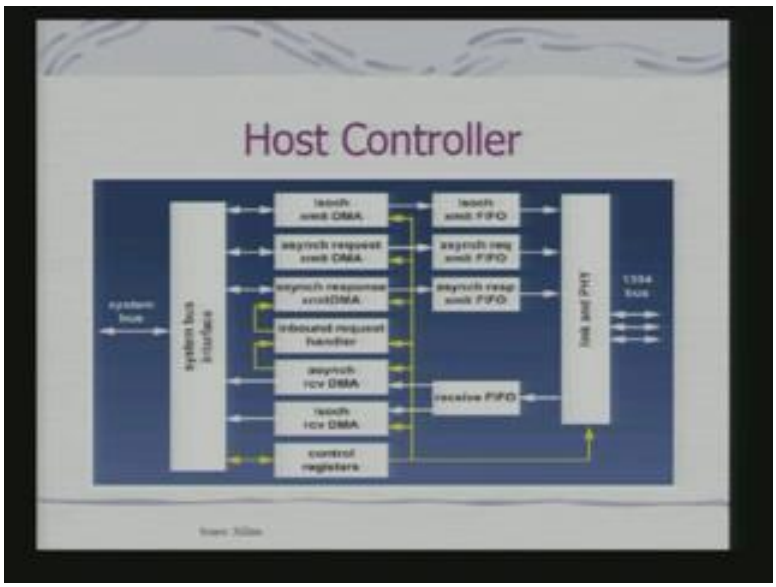
So, it is actually ((refer time: 23:45)) in the data packets in that sense it does error detection as well as correction because that is the basic necessity at that level. So, if I detect an error I need to take appropriate action one of the appropriate actions could be retransmission. It handles provision for cycle control for isochronous channel we have already seen then why it is an isochronous channel? We really do not go through the transaction layer. So, it is a continuous flow of data. So, you have to maintain the cycle by which the data is to be transferred and that is what is handle by the link layer. Link layer also provides an acknowledge datagram to the transaction layer what is the datagram? Datagram is a one way data transfer with a request confirmation. So, I can send the packet with a request confirmation and that thing is detected; obviously, the Linked Layer, but the whole management of the datagram has to take place at the transaction layer. Because that is keeping track of the data transfer and the nature of the data transfer that is taking place.

(Refer Slide Time: 24:56)



So, transaction layer implements what is called a request response protocol; that means, on the basis of the request of the devices will send the data. And that request response protocol is the job of the transporting. And the way it is designed the basic specification is such that it minimizes the amount of circuitry require to interconnect with standard buses such as PCI bus. Because one of the time movers of these standard was PC manufactures and they would like the PCs to play a role in next should not be just among the devices.

(Refer Slide Time: 25:42)



So, the standard is specified such that there is a compatibility with the PCI bus and your host controller of IEEE 1394 is something like this. Now, let us try to understand what is what is specified here is that this is when we are talking about the system bus interface. These interface is with respect to say system bus like PCI and these layer is IEEE 1394 and this is really the interfacing hardware. What we say that the physical as well as the link layers of 1394 and typically this is what is to be always powered up. Because when I say that the course may be ((refer time: 26:34)) not and then also the data communication to should take place of the signals have to be should be powered on and this interfacing is what is interesting and should be noted. These are all buffers and they are all buffers. So, therefore, they have to organized as FIFO first in first out kind of a structure.

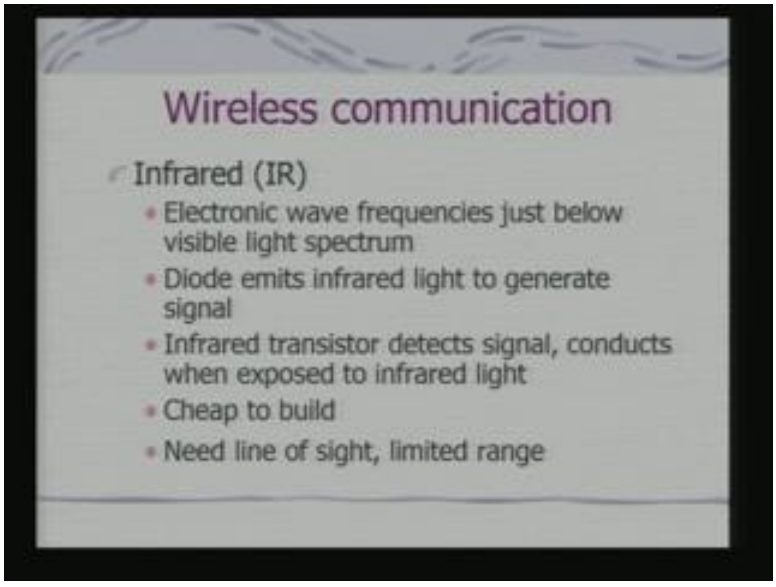
So, this is for an isochronous transfer transmit FIFO. This is asynchronous transmit FIFO and this is an asynchronous response transmit this is for the request this is for response, because it is a symmetric device. So, it can request as well as it can it would may be needed to send its response as well. Similarly, this is there is a receive FIFO which is for the data which is coming from the bus. Now, these FIFOs will be; obviously, connected to what if you are looking at a host interface there will be all connected to the various DMA channels. There would be connected to various DMA channel, because this data has to be transfer now because transfer to a transfer to the memory area. So, that the processor which is sitting inside the embedded system can make use of these data we are not looked at these interface for other buses. Other buses will also have conceptually similar interface architecture.

So, here you find that for each one of them I have got indicated the DMA channel fine. And what you have got? You have got the control registers. In fact, this control registers will essentially manage the overall DMA transfers. So, if it is really a DMA request. So, that a processor should relinquish the bus and they data get transfer. So, that information will be generated from here and what is the location where the data has to be low etcetera all these information will come from control registers. Now, here; obviously, you will find that for different kinds of data there has been different DMA channels which has been identified. And here you find that inbound request handler is something which is different from other transmit handlers. So, because they have to be configured are these DMA channels have to be handle differently depending on the how the data is mapped one to the memory for subsequent usage.

So, these interface is typically part of a PC when PC is one of the IEEE 1394 device. It would be part of any of their embedded system when that itself is participating in the IEEE 1394. So, typically you would expect similar kind of a controller with even you video camera which is IEEE 1394 enabled, but IEEE, but a video camera may not have an extended interface for receiving the data, because it is suppose to transmit the data in response to a request. There may be adaptations with the device specific characteristics, but all the devices would support this kind of a Host controller architecture. In fact, this is a very basic feature to support all these interfacing protocols although we a discussing only here in the context of IEEE 1394.

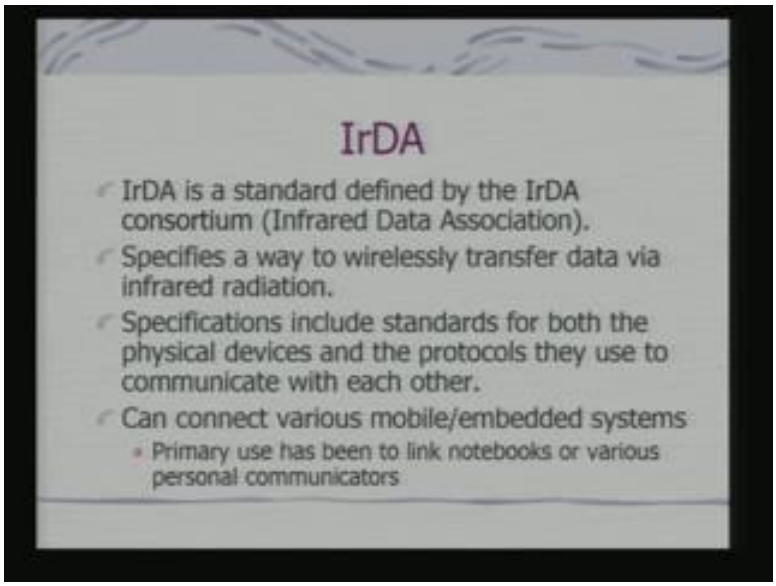
And since all the devices that we are talking about they are not passive devices in the sense a day it is not the day do not have a processor. Here all embedded devices having a processor and that makes the fact that you can really have this kind of a sophisticated protocol implemented for these kind of interfacing tasks. If you just consider you would like to connect simply a the modem which is not doing anything you cannot really do it in this way. Even if you have a USB modem USB will also have a small processor setting inside or a control logic setting inside implementing the whole interfacing logic. Next we shall look at wireless protocols these are all wire these two interfacing protocol that we have looked at USB as well as IEEE 1394 or wired protocols.

(Refer Slide Time: 31:45)



Now, we shall look at wireless protocol and in the wireless protocol the most common. In fact, very easily available today is what is infra red protocol. I possibly should not say most common, because other protocols that RF protocols are also have become very common place today. So, here you used infrared light for transmission of data which is the most distinctive feature of this protocol. So, basic concept is very straight forward. You have got infrared diode which emits light to generate signal. And there would be infrared transistors which detect signal and conducts when exposed to infrared light. And that is how it actually detects the signal is a very simplistic conceptual scenario for implementation of these infrared protocol. It is chip to built and; obviously, needs line of sight limited range, because if it is not in the line of sight if your infrared transmission is block then you cannot really have the communication. In fact, all of you use infrared control of your set TV your sound systems and these protocol is for doing transfer over such infra red channel.

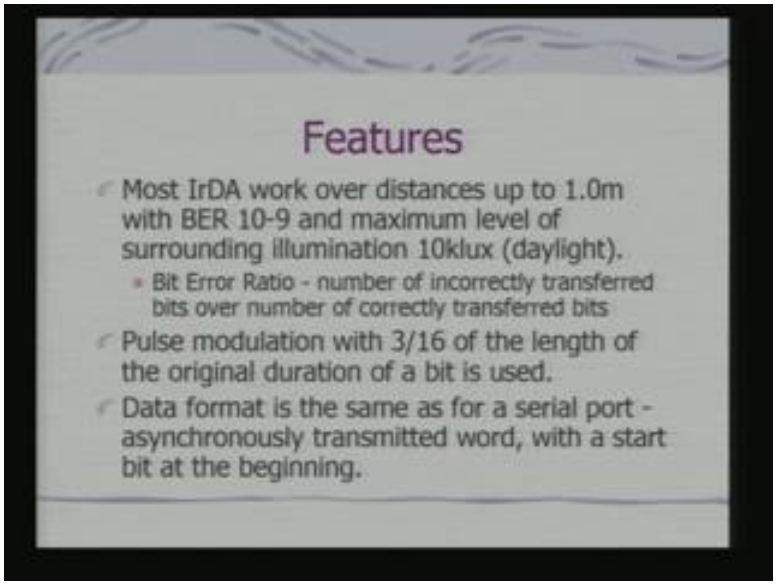
(Refer Slide Time: 33:19)



So, IrDA is a standard defined by the IrDA consortium infrared data association. Now, all these things have to be standard simplify otherwise things are not interoperable. It is specifies a way to wirelessly transfer data via infrared radiation. This specification includes standards for both the physical devices and the protocols they use to communicate with each other. Because here the physical device specifications have to be with, regard to the diodes or the transistors that have been used. And it can connect various mobile and embedded systems and primary use has been to link notebooks or

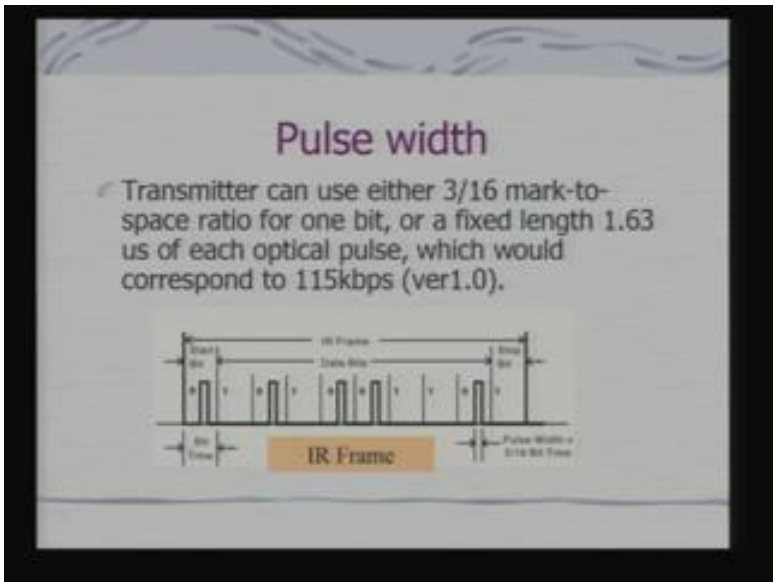
various what you call personal communicators which may be PDA with cell phones and their variance.

(Refer Slide Time: 34:16)



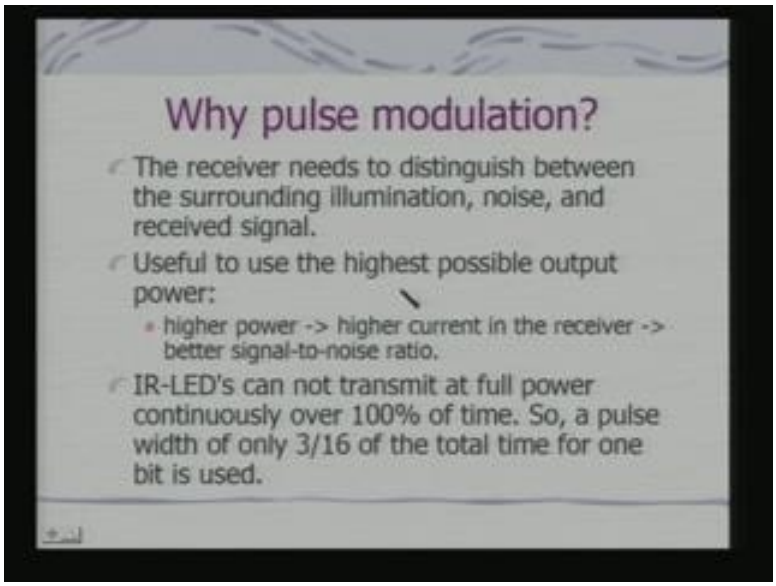
What are the features of IrDA? So, what we say that most IrDA work over distances up to 1.0 m and not it is not a very long distance with BER that is called Bit Error Ratio between 10-9 and maximum level of surrounding illumination of 10 kilo lux, because the surrounding illumination will have an effect in the detection of the signal. So, that is become a important point. So, in fact, bit error ratio is the number of incorrectly transferred bits over number of correctly transferred bits. In fact, it is 10-9 over 100 bits now pulse modulation with 3 by16 of the length of the original duration of a bit is used. Data format is same as for a serial port when it is asynchronously transmitting a word with start bit at the beginning.

(Refer Slide Time: 35:22)



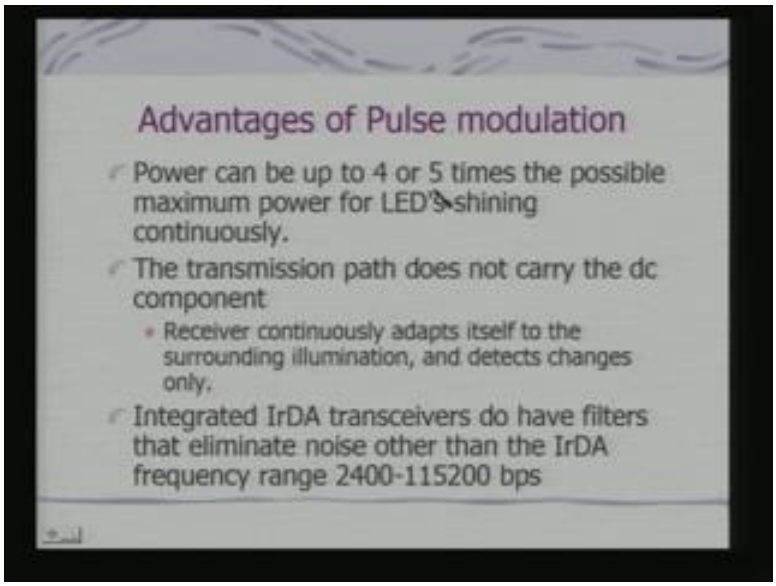
So, this is how the pulse width looks like . So, what is really asynchronous transmission? If you look at an URT we have seen that almost all the devices all the microcontrollers they have an URT interface. Typically URT supports asynchronous transmission an in an asynchronous transmission what happens typically clock is not part of the data. You have a start bit which indicates beginning of the data and a stop bit which indicates end of the data. So, if you look at a simple example of RS232C which is a most common asynchronous protocol. We say normally the signal lines or in what you call marking state? Marking state is typically high state for the signals and when a data transmission is to start that bit goes down. So, typically you have got a start bit. So, this start bit indicates beginning of a data transmission that is followed by the data bits and then you have got a stop bit this pulse widths a typically 3 by16 of the bit time. So, it is not that corresponding to the bit it will be always on. So, it what we are using is it is effectively what effectively a pulse width modulated transmission the pulse width is. So, transmitter can use either 3 by16 mark to space ratio for one bit or fixed length 1.63 micro second or each optical pulse which would typically correspond to 115 Kbps. This is version one there are other versions where you can have faster transfers now why pulse modulation?

(Refer Slide Time: 37:31)



The basic requirement is that the receiver needs to distinguish between the surrounding illumination, noise and received signal. And therefore, if you have to have better detection capability you would like to use the highest possible output power for the LED's. Now, therefore, what we say that higher power implies higher current in the receiver because as you get infrared the transistor starts conducting. So, you have higher current flow and that would imply better signal to noise ratio, but IR LED's cannot transmit at full power continuously over 100 percent of time. So, a pulse width of only 3 by16 of the total time for one bit is used. So, what it implies it it implies that you can deliver higher PIC power and you do not operate at high average power, because that is what your diode can not really handle.

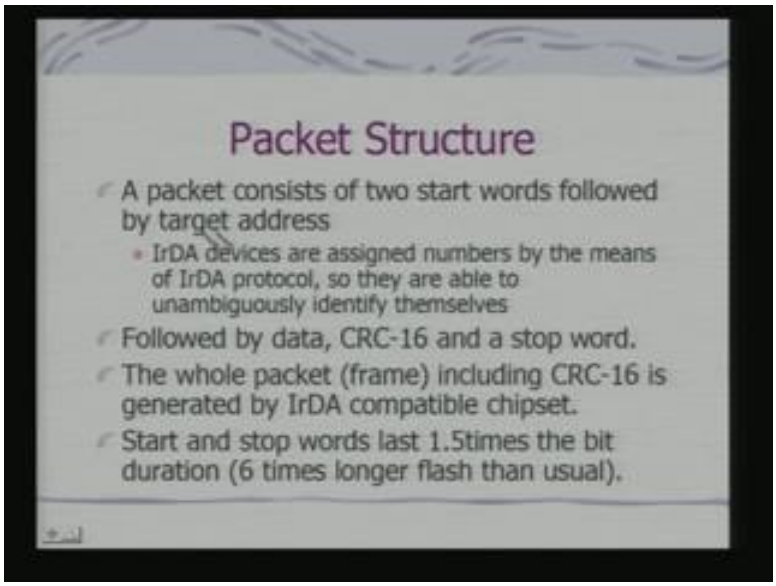
(Refer Slide Time: 38:42)



And there is another interesting thing is that is power can be up to 4 or 5 times of possible maximum power for LED's shining continuously. These scale up really is a reason of using this pulse width modulation at the same time the transmission part does not carry the dc component and. So, the receiver can continuously adopts itself to the surrounding illumination and detects changes only, because if you are using that. So, pulse essentially means change in the with respect to a ambient at the surrounding illumination in the noise. So, it is not a continuous transmission. So, you can detect the changes that is precise the reason why this pulse width is used. And what we say that integrated IrDA transceivers do have filters that eliminate noise other than the IrDA frequency range.

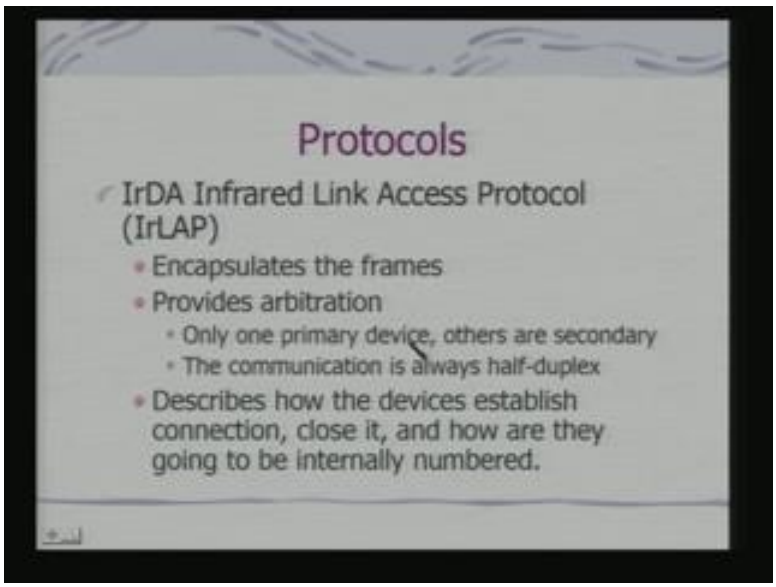
So, other noise is eliminated. So, this detection becomes important and critical in view of the surrounding illumination as well as the noise sources. And in fact, if you have used such a link for transferring the data you will find a slight disturbance. In many cases there are disturbances cost the data interrupt and in various ways you will find that message coming up. Particularly it is very sensitive to line of sight, because if it is blocked then; obviously, cannot really have a data transfer. Now, since it is a protocol. So, these so, far what we have talked about? We have talked about simply how the data is transmitted in terms of in terms of the bytes and it is an asynchronous transmission. So, you have got a start bit and a stop bit to indicate that. So, what we really have is in these case a definition of a packet or a packet structure.

(Refer Slide Time: 40:42)



Packet consists of 2 start words followed by the target address. IrDA devices are assigned numbers by means of IrDA protocol. So, they are able to unambiguously identify themselves. So, these properties similar to that of your USB as well as IEEE 1394 because here also I can just plug in a device just bring a device near to a receiver and then they can start really communicating with each other. The whole packet of the frame including CRC 16 is generated by IrDA compatible chipset. What is the CRC 16? This is for doing error detection as well as subsequent error correction.

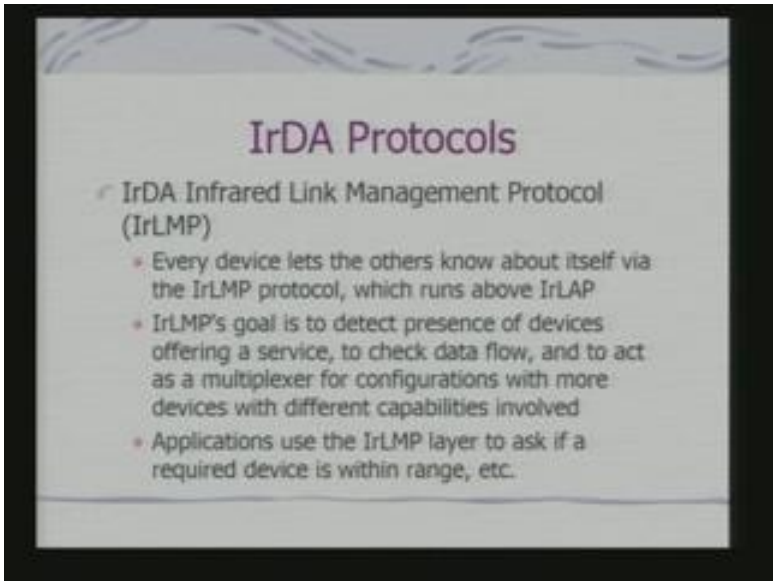
(Refer Slide Time: 41:33)



There are set of protocols which have been defined for these purpose just like you have got an IEEE 1394 layers of protocols here also you have got layers of protocol. First is what is called Infrared Link Access Protocol although typically it is not like a big network which gets it is setup. It is more like a one to one connection in many cases in maturity of the cases one to one connection and the connection gets establish when the devices are in vicinity of each other. Because you have to have them in the line of sight, but still then you need to have this proper protocol to ensure that is a correct transfer which takes place. So, you have got Infrared Link Access Protocol each encapsulates the frames provides arbitration, because they has to be only one primary device and others as a secondary device.

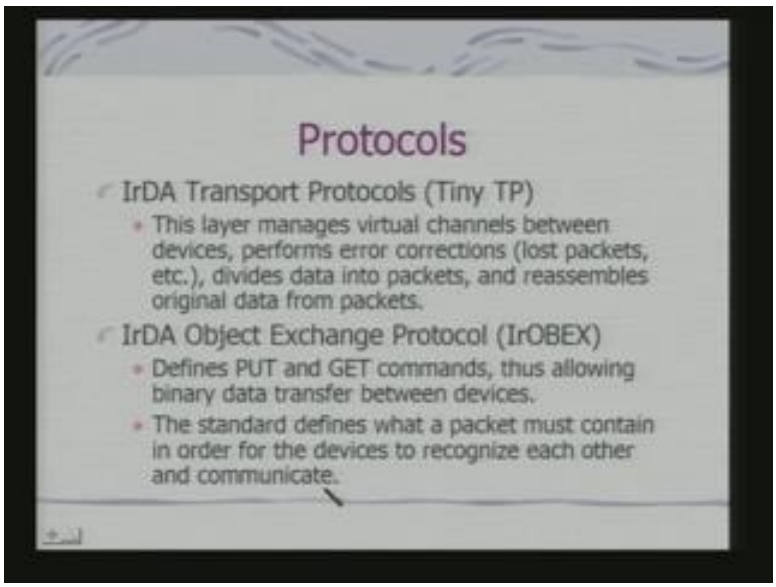
And communication is always half duplex both the sides really does not communicate and these protocol describes how devices establish connection close it and how are they going to be internally numbered. Because until an analyze their internally numbered they cannot be accessed. Even if there are multiple devices and if I have to access them they got to have a unique number and what is interesting to note is that although we are not telling and it is not a host based kind of a protocol. But there is typically always one primary device and others are secondary and top of these you have got Infrared Link management Protocol.

(Refer Slide Time: 43:22)



So, every device lets know others about itself via this IrLMP protocol which runs above IrLAP. So, LMP's goal is to detect presence of devices offering a service to check data flow and to act as a multiplexer for configurations with more devices with different capabilities involved. If there are not more devices then really this multiplexing capability is not really of significants. So, applications use this layer to find out if a required device is within a range or not. So, typically if you are using a PC with your infrared support you will find the moment you get a device inside it is a vicinity the PCR's responding it says that there is a infrared recipient in its vicinity. So, an application when it is drawn you are managing an infrared port typically uses this protocol to figure out whether there is any device with each it can establish a connection.

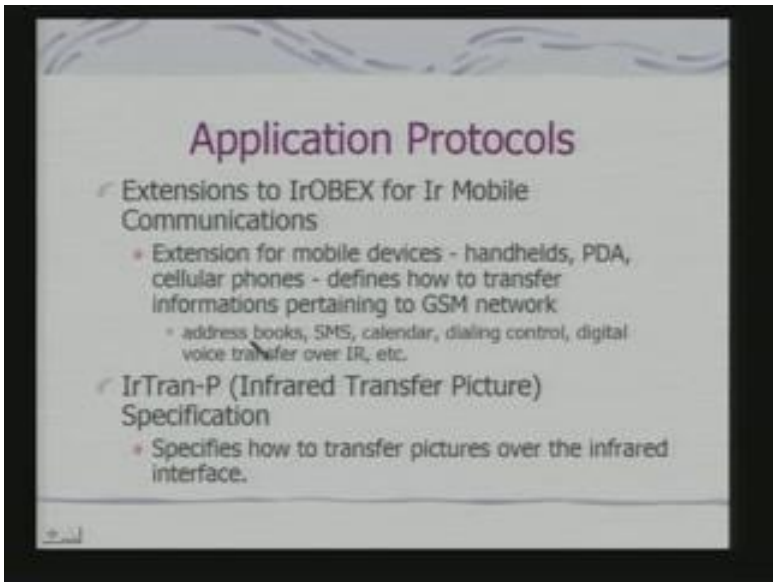
(Refer Slide Time: 44:45)



Then you have got transport protocols which is IrDA transport protocol which is called also Tiny TP, because transport protocol is essentially to maintain this virtual channels that is point to point connections It performs error corrections if there are packets loss how the packet loss has to be taken care of then a divides data into packets reassembles original data from packets and if you are familiar with the concept of networking. This is exactly the functionality which is associate with the transport layer of any network an top of these you have got a object exchange protocol. So, these actually have got a kind of transactions which are defined over here say defines PUT and GET commands thus allowing binary data transfer between the devices. So, actual data transfer which takes place between the devices that goes to your PUT and GET. In fact, when you say when

you find out on a on a PC if you are doing this transfer the rather this data to be accepted or received or sent that gets translated into effectively this PUT GET commands. This standard device what a packet must contain in order for the device to recognize each other and communicate. In fact, these layers a protocol had you defined these basically defines a generic kind of data exchange. Now, what is interesting will find that in order to make devices exploit this protocol effectively.

(Refer Slide Time: 46:43)

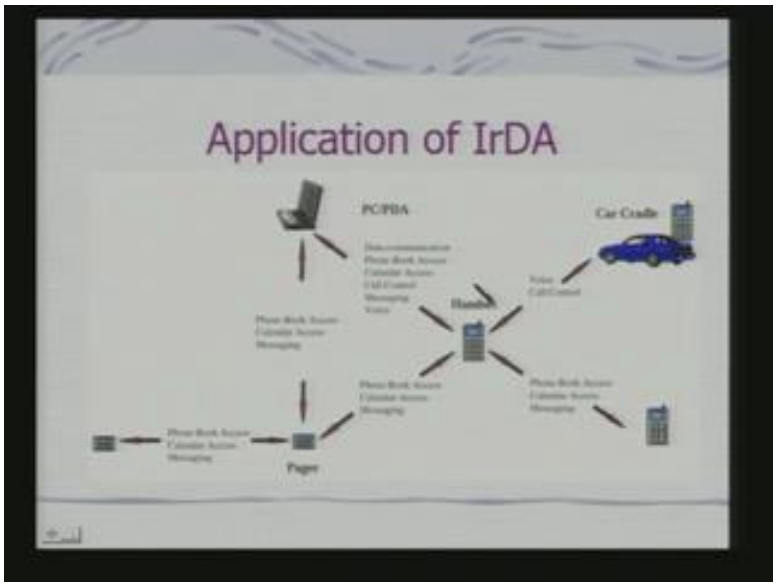


There are also application level protocols defined. So, there are extends for defining what is called extends to IrOBEX for Ir mobile communications. So, these are extension for mobile devices like handhelds PDA cellular phones and it defines how to transfer information's typically pertaining to GSM network. Address books, SMS, calendar, dialing control, digital voice transfer over IR, etc. So, you are actually having a GSM device it has got it GSM capability, but you have providing the interface for the GSM functionalities to higher link. And so, you define protocols specifically for this kind of applications then you say infrared transfer picture. This is another protocol how to transfer pictures over infrared interface.

So, what I wanted to show here through this I am not going into any of the details of the protocols that depending on the application. You have got the application will have mechanisms which have been specified over these kind of interfacing generic interfacing specifications. Obviously, these are not full fledged computer networks, but what you

can realize is many of the networking related concepts have been flow in and have been used to specify the protocols simply. Because that makes the management of multiple devices more efficient and ensures the data transfer is robust and you can handle errors in a better fashion.

(Refer Slide Time: 48:39)

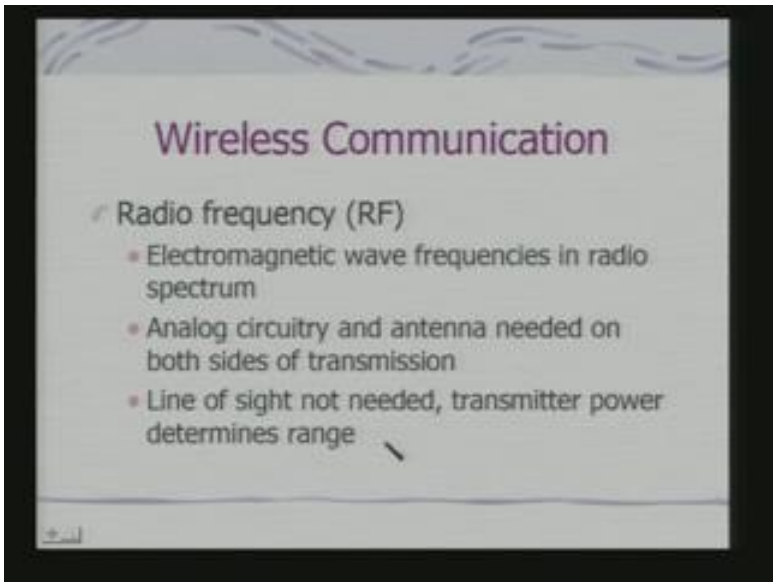


This a kind of an application scenario which people conceive of and its being used in the context of embedded systems. So, what you see that all these are embedded systems. In fact, other than this PC all these are embedded Systems this can be a digital diary which is again and embedded system which may be may have an IR link to a pager. This pager can have a IR link to the PC to load your phone book calendar etc. Now, here you have got a handset a mobile and mobile can communicate with any of these devices. This is a mobile to mobile communication a link which have showing here it is a mobile to mobile communication with the IR link. And then you have can have a car cradle and this is a voice call control; that means, you can have a mobile kept there. You are not really using your hand to control the mobile; you can have a voice interface to the mobile and the whole voice interface could be through an infraredly through an wirelessly.

So, you just spick out to a microphone through an wireless ling it can be connected ((refer time: 50:01)). So, the whole point is if you looking do it this is some kind of a personal network it also its not truly a personal network or technical what today is being define as personal network. But you can see that what you can established through this

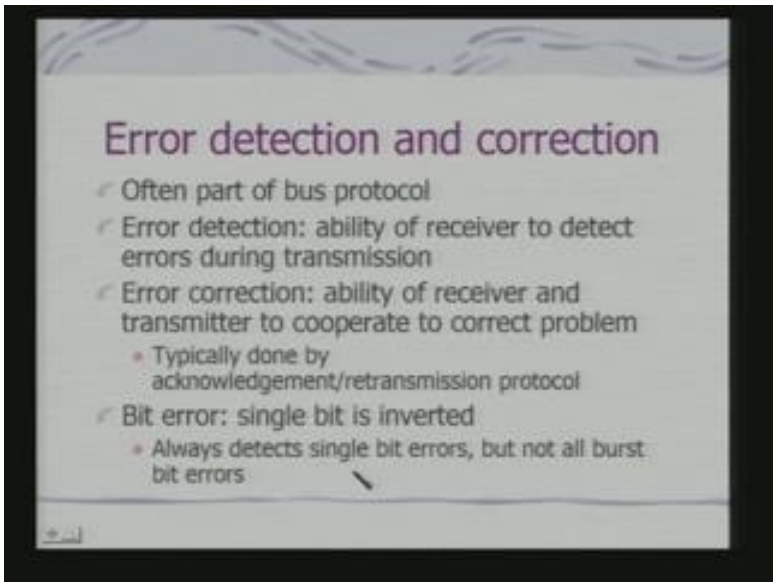
kind of a IrDA protocol is lots of functional utilities which are required for individual operations that can be network typically we talk about networking of computers this is an example of networking of appliances. The other option of this kind of a communication networking devices together is using protocols in radio frequency.

(Refer Slide Time: 50:50)



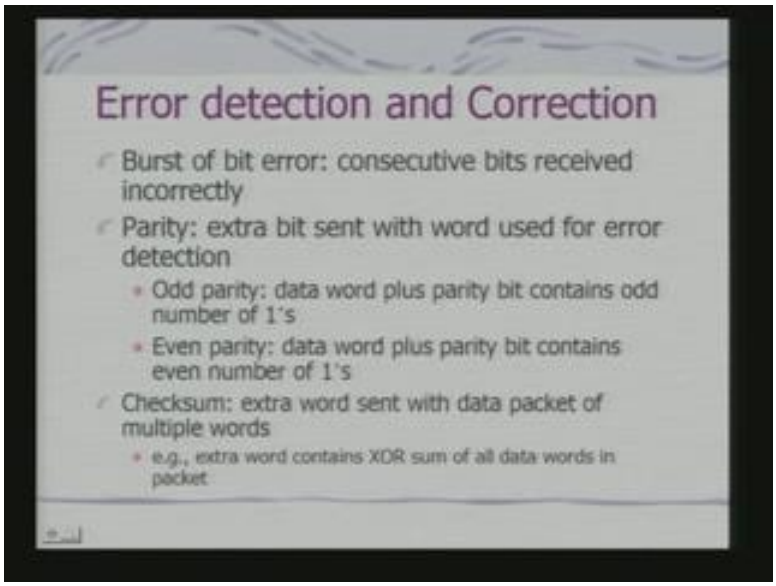
We had looked at till infrared in radio frequency. So, here the electromagnetic wave frequencies and radio spectrum is what is used. Obviously, you require the analog circuitry and antenna needed on both sides of transmission. The advantage here is the line of sight not needed and transmitter power determines the range. Now, we shall not going into more details of these wireless communication links and how to have a complete full fledged network built around wireless communication link? These we shall discussed when we really we shall be discussing embedded Systems with full fledged networking capability in that context I shall pickup these protocols, but this also another way of getting embedded systems communicating with each other. But there are some generic issues which sends you are finishing of this bus structure an interfacing protocol I would like to touch a call.

(Refer Slide Time: 51:58)



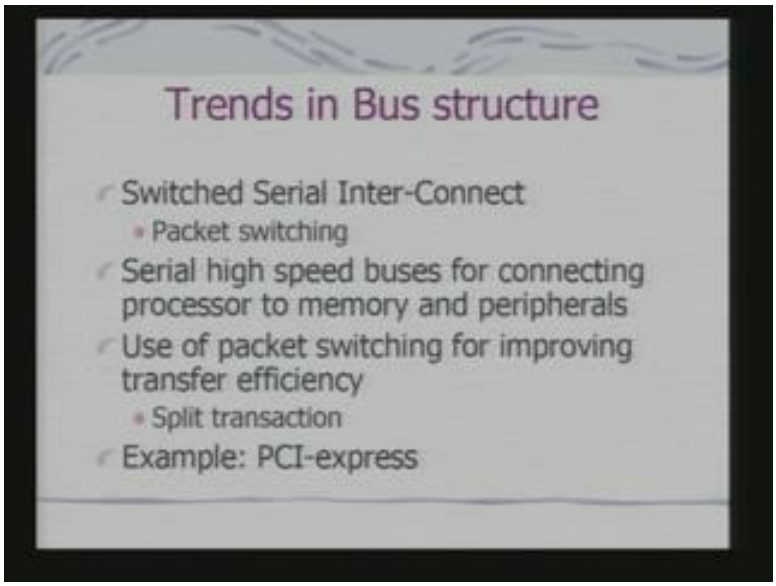
One very basic issue is error detection and control. In fact, you will find that almost all these protocols they have got error detection and correction capabilities built in as part of the lower layers. So, that is a part of past protocol. So, what is error detection? Ability of the receiver to detect errors during transmission error correction ability of receiver and transmitter cooperate to correct problem. So, it is typically done by acknowledgement or retransmission protocol, but when you are doing on isochronous transfer you really do not have a provision for retransmission. So, there has to be some kind of a post processing to take care of ((refer time: 52:39)). What kind of errors can occur the simplest error is bit errors? So, here the single bit is effectively inverted. So, when we try to detect a single bit error it kind we can detect a single bit errors, but not all burst bit errors can be always detective.

(Refer Slide Time: 53:01)



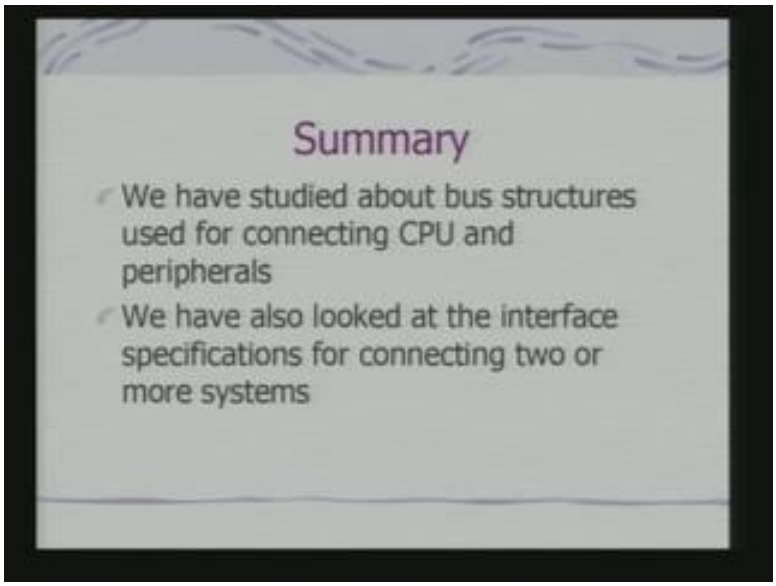
So, what is the burst bit error? Consecutive bits received incorrectly now if you are using typically a data level tools and we use data level tools to check whether we are correctly receive the bits or not. So, the most common mechanism is parity. An extra bit is sent with word used for error detection if there is an isolated error we can detected if there is always on even numbers. So, did get must I cannot detect that error if I am using a parity. So, it can be an add parity or an even parity which is used an. In fact, this is a basic error detection scheme which is there in as part of any of these asynchronous protocol even as simple as that of RS32C protocol. Checksum; now checksum can be generated using variety of algorithms is an extra word sent with data packet of multiple words. So that word really indicate. So, whether the data packet that you have received is correct or not so; that means, a receiver it uses the data part of the packet to recalculate the checksum and matches with the checksum byte which has been transmitted. If the two matches it ensures the data has been received correctly. So, an simple example of an checksum is an extra word which contains ((refer time: 54:23)) all data words in the packet. Obviously, we cannot use a very complex checksum, because that also puts the load on the interface circuitry bit for the purpose of verification and what are the current trends? So, we have seen that when we looked at the interfacing protocols.

(Refer Slide Time: 54:40)



Primarily they have in fact, serial interfacing it has move to into a kind of a switched network. In fact, packet switching is what is typically that run. In fact, today, we have bus structures when we refer to bus structures we are referring to how the CPU is to be connected to peripherals and the memory. We are also the trend is towards designing high speed switch serial interconnects; that means, effectively bringing in packet switching of the. So, use a packet switching for improving what we call transfer efficiency an. In fact, we had discussed if you remember. We are discuss the basic concept of split transaction; that means, you carry out a transaction. And if my slave is not currently able to do it the transaction it remembers the identity of the master and later on carries of the transaction. This is the basic concept and very similar to that of your packet switching. In fact, PCI express which is the current standard which is moving into the PC PC scenario and soon will move into embedded systems and there are other standards moving into embedded systems. What the basic trend is to go into high speed packet switching serial interconnect for connecting the peripherals in even the memory. And in fact, the bus structure or these bus structure protocols becomes a vary key issue, because you can design fast processors you can design fast memory, but everything will go to win. If you do not have a proper bus protocol and high speed connection that is why this is a very important issue for design and development. So, this brings us to the end of this topic as such.

(Refer Slide Time: 56:50)



So, what we have studied is bus structures used for connecting CPU and peripherals. We have also looked at interface specifications for connecting two or more embedded systems. We shall switch to some other some other topic in the next class in particular we shall look at how to manage power in an embedded system in the next class. Any questions? What we are telling is in case of an IRD application if every devices moving I cannot do it absolutely true, but the whole idea is that when there in the line of sight I can use them. So, it is not like a network connection with the kind of a persistent computation just like you look at in an ordinary network that computers are connected over the network and you continue continuously do a job. It is not like that it is connected when it is needed so, connection on need. So, I shall connect my device in line of side when I need it when I need to load my phone book or my diary later on it is not needed. So, I can move around.

.