

Computer Networks
Prof. Sujoy Ghosh
Department of Computer Science and Engineering
Indian Institute of Technology – Kharagpur

Lecture - 13
Multiple Access

Good day. Today we will talk about multiple access. What is multiple access?

(Refer slide time: 00:55)



(Refer slide time: 00:56)



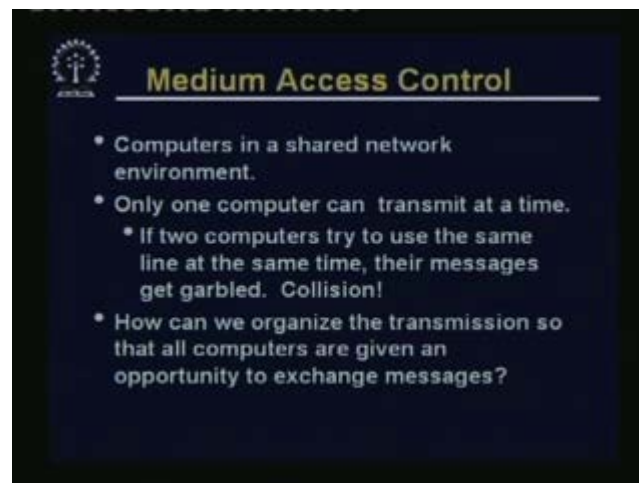
If you remember, we had talked about 7 layers in the OSI stack, the top one being application, then we have presentation, session, transport, network, link and physical. We had been mostly talking about the physical layer till now, although for optical networks, we sort of ventured into some of the higher layers. But from this lecture onwards, we want to concentrate on the link layer. What medium access does is, it coordinates competing requests for medium; that means, there is a medium, which may be an object of contention, meaning that several nodes may want to use it and this medium access control protocol has to do with how to handle that. Sharing of link and transport of data over the link is in general the description of what the data link layer does. When we share a link, there is a question of competing requests and we have to have some way of resolving that, and there is also a question of transport of data over a link. A link if you remember is two nodes, which are connected. The nodes may be computers or routers, switches, etc., the two networking nodes are directly connected; it might mean a cable, a copper cable, or a fiber, or it might mean a shared medium like the free space. There is some way of communicating directly between these two nodes; that is what we mean by link. That means it is just one hop in the network. That is what we are talking about in the data link layer. There is a question of reliable transfer of data over this link and if when you have a free space transmission with so many nodes in the network, many people would like to transmit. So there is a question of a sharing this medium and there is a question of who would access it when. Just as a shared free space could be a shared medium, Similarly, if you have some kind of a bus from which a number of nodes are hanging, that bus may also be an object of contention. Bus is the medium, through which the communication is taking place and there is some kind of competition of sharing between this shared medium between the nodes. So we have to handle; that is the other thing.

(Refer slide time: 03:52)



So actually this link layer may be divided into two sub layers, so to say: the upper one is the logical link control and the lower one is the medium access control. We will talk about logical link control later. We start with medium access control; there are a number of medium access control protocols and techniques. We will discuss a number of them.

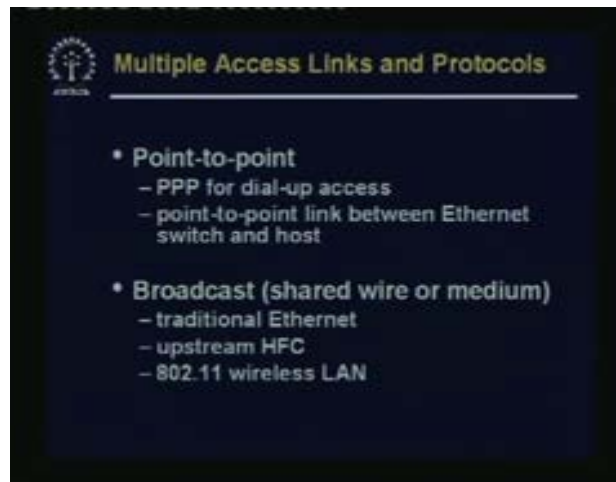
(Refer slide time: 4:15)



The situation we have is computers in a shared network environment and although the medium is shared, it is taken that only one computer can transmit at a time. If two computers try to use the same line at the same time for transmission, i.e., when one is transmitting one or more may

receive. Receiving is not a problem; but when two computers want to transmit at the same time, their messages get garbled. We say that there is a collision. How can we organize the transmission so that all computers are given an opportunity to exchange messages? If you remember, when we were talking about multiplexing, that is also in some sense sharing a medium. So we had this time division multiplexing and frequency division multiplexing; actually we also have time division multiple access and frequency division multiple access, etc., over here. The difference between the two cases is that in the multiplexing case, the lines are all coming together into the multiplexer, whereas in multiple access, the points are all geographically distributed and they may not know about each other. There may not be much of a coordination; in some schemes we do have some coordination in some schemes we do not have any coordination. So all these cases we will see.

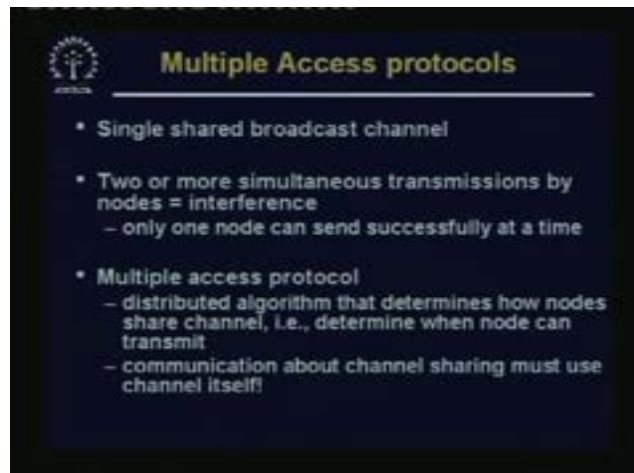
(Refer slide time: 06:03)



There may be point-to-point links in a network like PPP for dial-up access, etc., which is quite common – point-to-point link between Ethernet switch and a computer. So these are examples of point-to-point links; we will come to this later on. Then there are broadcasts, like a traditional Ethernet, upstream in an HFC. HFC is a hybrid fiber coaxial; some part it is fiber and some part of it is coaxial, and the upstream traffic there also uses that medium as a shared medium. Then we have 802.11, which is a wireless standard. We will have a separate lecture on this wireless networking, etc. 802.11 is the number of the standard, which defines some

form of a wireless networking. Wireless is a shared medium because it is sharing our common shared electromagnetic field that we already mentioned.

(Refer slide time: 7:14)



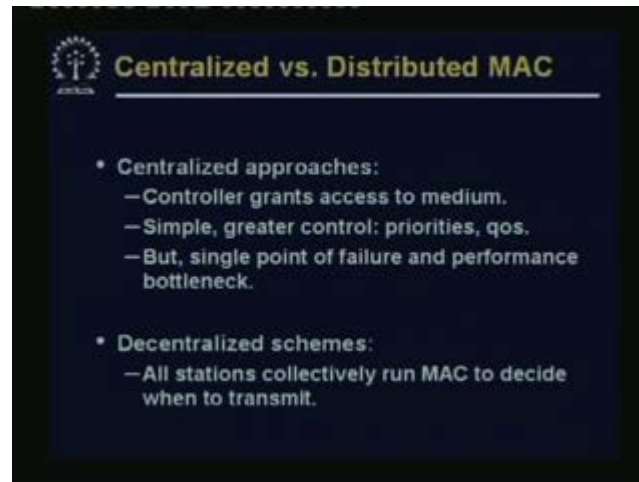
There may be a single shared broadcast channel, there may be many also, or more than one also. Two or more simultaneous transmissions by nodes will mean interference. Only one node can send successfully at a time. Multiple access protocol is a distributed algorithm; so this is where it differs from multiplexing. It is a distributed algorithm that determines how nodes share channels; that is determined when a node can transmit. Communication about channel sharing must use the channel itself; that means, we have what we earlier called in-band control usually. That means if you have a, let us say, wireless network, if you are running any kind of protocol for control purpose, etc., that would also be using the same medium. The same question of how to coordinate that with all the other competing nodes – which are trying to transmit data – all that comes into play.

(Refer slide time: 8:15)



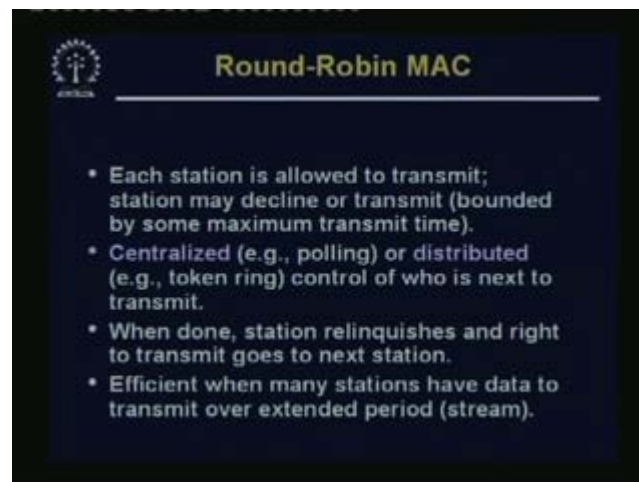
There are centralized approaches to a MAC, i.e., medium access control. The centralized approach is simpler in the sense that there is a controller, which grants access to the medium. It is simple, has greater control, priorities, quality of service, etc. – all this can be implemented very easily. For example, if you want to give higher priority to one particular node the controller simply just takes note of the priority when allocating the medium to a particular node. But there are problems with centralized approach also. One is that a single point of failure is a big problem. When we have a centralized system and if that system fails, then the whole network is down, which may not be a good idea in many many situations. Secondly, there will be a performance bottleneck; meaning, first of all everybody is communicating to it but when the network becomes bigger, the demand on its computational capacity, its control mechanism, etc., goes up proportionately and that is also another problem in centralized approaches. In decentralized schemes all stations collectively run MAC to decide when to transmit.

(Refer slide time: 09:46)



So in a decentralized scheme we are running some kind of a distributed algorithm.

(Refer slide time: 09:58)



There are various kinds of MAC protocols: one is called something like a round robin MAC. In round robin there are a number of stations and each station takes its turn in a round robin fashion. There are some protocols based on round robin MAC. So here, each station is allowed to transmit; a station may decline or transmit. That means, if it has nothing to transmit, it may decline. Otherwise, it will transmit. It could be centralized; for example, polling, or distributed;

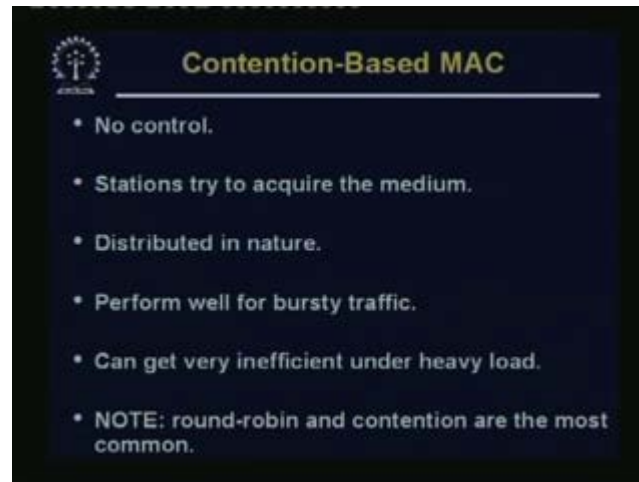
for example, token ring. Control may be centralized or distributed of who is next to transmit. When done, the station relinquishes and the right to transmit goes to the next station. This is efficient when many stations have data to transmit over an extended period. So that may be a good scheme.

(Refer slide time:10:56)



There are scheduled access MACs, like time is divided into slots just like our time division multiplexing; station reserve slots in the future; multiple slots for extended transmissions and suited to stream traffic. There is a question of reservation of some slots. That would be a part of the protocol about how reservation will be done, who can reserve, and if two people want to reserve simultaneously then what happens. So we have to handle issues like that. This is the scheduled access MAC, when it is scheduled, some reservation is done beforehand. Then there are some contention based MACs.

(Refer slide time: 11:41)



Contention means there is hardly any control; there is no control. Stations simply try to grab the medium. This is distributed in nature and surprisingly, for no control, it performs quite well for a bursty traffic, but it can get very inefficient under heavy load. Actually, the round robin MAC that I mentioned before, and the contention based MAC are very common. The most common example is the Ethernet, which we will see especially later on; how it is done, or how satellite communications use contention based MAC. That means nodes simply track and try to grab the control but over a long period of time, the amount of data or the number of times the nodes on an average try to communicate and the transmission time – these two factors together define the load on the system. If the load is light then this contention based protocols very surprisingly are much simpler and may be cheaper to implement. So examples of contention based are ALOHA and slotted ALOHA.

(Refer slide time: 13:06)



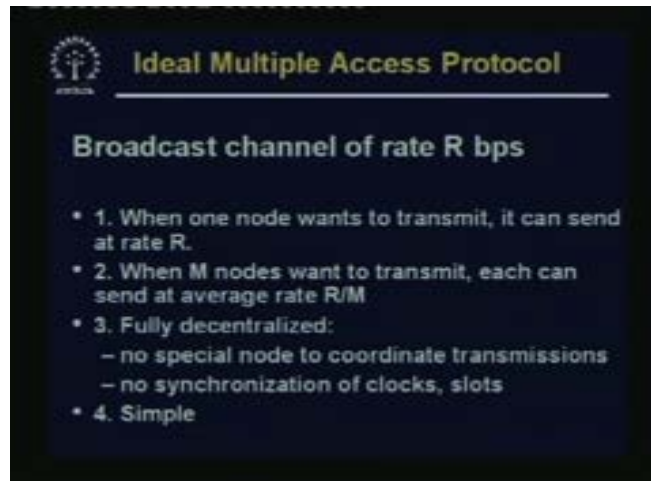
These refer to some protocols which we use in satellite communication; so we will discuss these when we talk about satellite communication. We have CSMA – it stands for carrier sense multiple access or CSMA CD, which is carrier sense multiple access with collision detection. There are other variants of this, like carrier sense multiple access with collision avoidance and things like that. ALOHA or slotted ALOHA are for satellite, CSMA, CSMA CD, specifically CSMA CD, is used by Ethernet in many situations; and then we have a CSMA CA, may be for cellular communication. These contention based MACs. Round Robin, are token based protocols and two very common ones are token bus and token ring. Actually, we will discuss these in this lecture as well as in the next.

(Refer slide time: 14:09)

	Topologies	
Techniques	Bus	Ring
Round robin	Token bus (802.4) Polling (802.11)	Token ring (802.5; FDDI)
Scheduled	DQDB (802.6)	
Contention	CSMA/CD (802.3) CSMA/CA (802.11)	

We can sort of summarize; first of all we can divide the networks into two types, depending on their topologies – one is the bus type. Bus means some communication channel and everybody is connected to the same communication channel. It may be a coaxial cable or some fibers or it could be a ring. Then we have this token bus, which is one particular MAC protocol. It has got IEEE standard number 802.4 or polling in 802.11. We will come to this later on. These are examples of bus type topologies using Round Robin techniques. Token ring is a ring type topology and 802.5 is the standard number. FDDI is a Round Robin, but the topology is the ring. There are scheduled approaches to medium access like DQ DB or distributed queue dual bus 802.6. We will be talking about this also. Contention based are CSMA CD, CSMA CA in 802.4 and 802.11. The first one is Ethernet and the second one is a wireless network; so we will look at these in later lectures.

(Refer slide time: 15:39)



The slide features a dark blue background with a light blue header. In the top left corner, there is a small circular logo containing a stylized 'T'. The title 'Ideal Multiple Access Protocol' is written in a light blue font. Below the title, the text 'Broadcast channel of rate R bps' is displayed. A bulleted list follows, detailing the characteristics of the protocol.

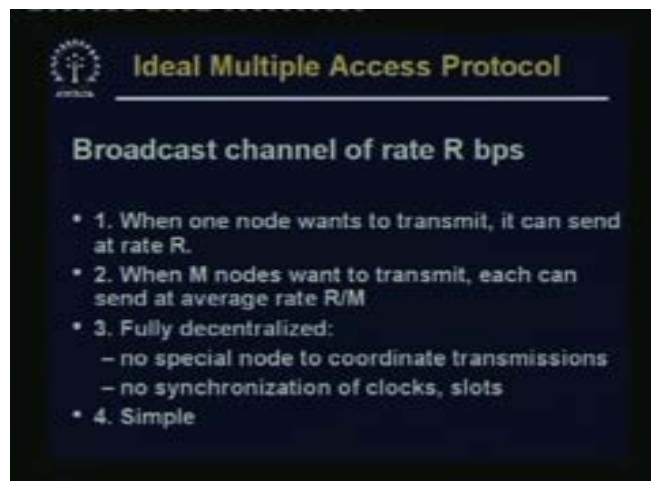
Ideal Multiple Access Protocol

Broadcast channel of rate R bps

- 1. When one node wants to transmit, it can send at rate R .
- 2. When M nodes want to transmit, each can send at average rate R/M
- 3. Fully decentralized:
 - no special node to coordinate transmissions
 - no synchronization of clocks, slots
- 4. Simple

What is the ideal of a multiple access protocol? Suppose we have a broadcast channel of rate R bps. Well, any broadcast channel would have some maximum limit or the rate you can communicate over; that depends on the characteristics of the medium. If their medium is different, this rate R may be different; for example, the rate for a fiber optical cable would be much higher than the rate for a coaxial cable. But the point is that there is always an upper limit due to various reasons; we need not go to that in this lecture. Due to various reasons of how we communicate, etc., there is a maximum rate R , which you can achieve on a particular medium using, let us say, the technology, which is present now.

(Refer slide time: 16:43)



This slide is identical to the one above, featuring a dark blue background, a light blue header with a circular logo, and a bulleted list of four points describing the ideal multiple access protocol.

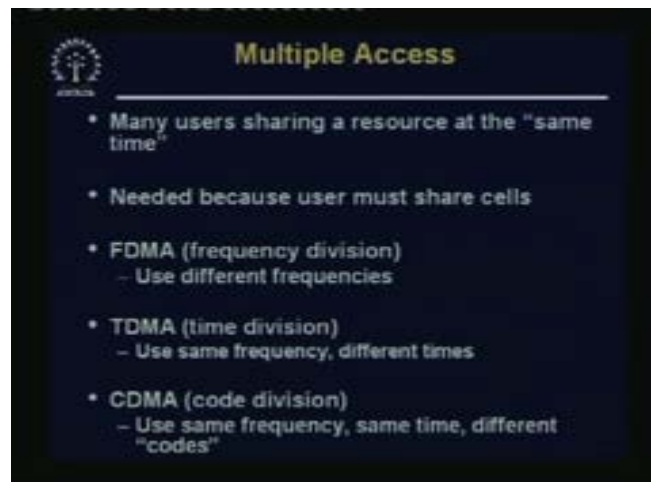
Ideal Multiple Access Protocol

Broadcast channel of rate R bps

- 1. When one node wants to transmit, it can send at rate R .
- 2. When M nodes want to transmit, each can send at average rate R/M
- 3. Fully decentralized:
 - no special node to coordinate transmissions
 - no synchronization of clocks, slots
- 4. Simple

When one node wants to transmit, it can send at rate R . There is no problem; it can send all the way up to rate R if the technology permits. When M nodes want to transmit, this R is not going to change. So what you want to do is that you want to divide up this rate R and distribute it among these M nodes, which want to communicate. So each can send at an average rate R/M . If that happens, naturally we have an ideal multiple access protocol; we can not do better than that. But it is difficult to achieve this theoretical rate because there will be some overhead for the protocol itself, which will eat up something. None of the protocols is 100 % efficient. Secondly, we would ideally like it to be fully decentralized. So there is no special node to coordinate transmissions, no synchronization of clocks and slots. Lastly, it will be simple. Naturally if it is decentralized, no synchronization, etc., that is one aspect of simplicity then the protocol itself will be simple. This is the other thing; and we want a fully decentralized scheme, I mean we preferred them because there would be no failure of single nodes, etc., and then there is no bottleneck. These problems should be avoided if you have a fully distributed system.

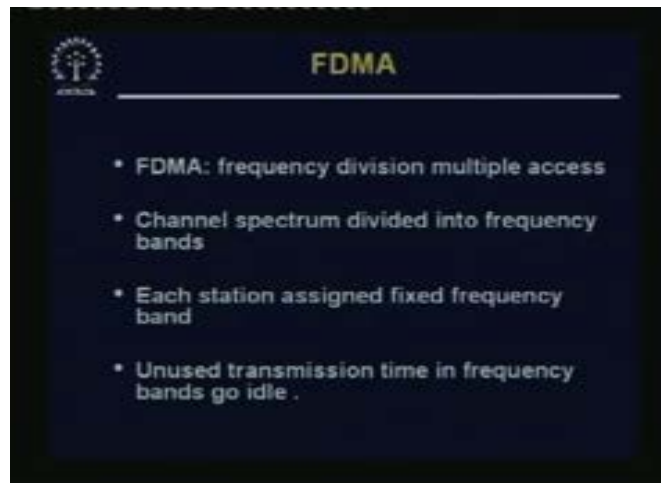
(Refer slide time: 18:18)



In multiple accesses as I mentioned, many users are sharing a resource at the same time; it is needed because users must share the cells. First let us talk about three different multiple accesses: frequency division multiple access, time division multiple access and code division multiple access. Frequency division multiple access and time division multiple access are very simple. We have already talked about multiplexing; only thing is that the connotation is

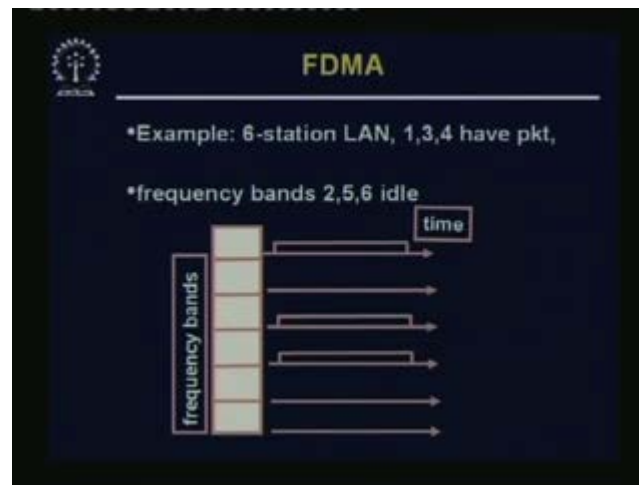
slightly changed over here. Code division multiple access is somewhat different; so we will discuss it in some more detail. So they use the same frequency, same time, but different codes; we will come to that.

(Refer slide time: 18:58)



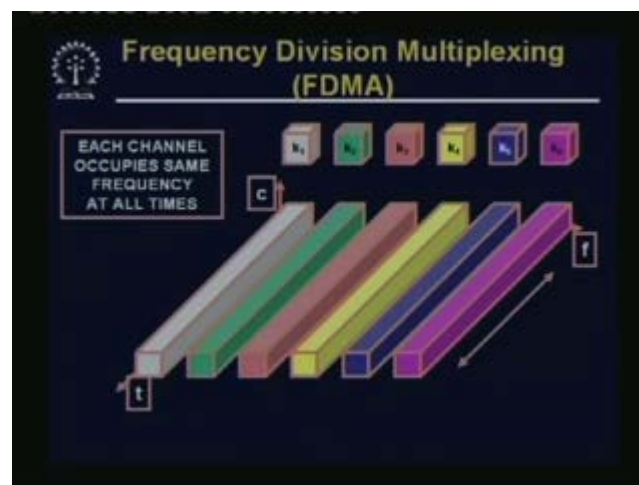
Let us start with FDMA. FDMA is frequency division multiple access. Just like frequency division multiplexing, channel spectrum is divided into frequency band. Each station has an assigned fixed frequency band. Unused transmission time in frequency bands goes idle. This is the simplest possible scheme. It is also not very efficient because many of the stations will be idle and that frequency band for that particular time is sitting idle; that part of the bandwidth is wasted. Also, we know that in a data networks the net traffic is very bursty; that means people want to transmit for a very short amount of time. Suppose the transmission time is T ; then may be for $500 \times T$ amount of time, it just sits idle. It is very bursty; so a frequency division multiple access would be very inefficient. But this is the oldest; you know, radio stations are doing multiplexing since the communication is only one way.

(Refer slide time: 20:26)



But if you are using some other kind of say radio frequency transmission using fixed channels, when there is both way of communication, then it is some kind of multiple access also. So suppose we have six stations LAN 1,3 and 4 have some packets to send. So they are sending; frequency bands 2,5,6 are idle. This happens quite often an FDMA.

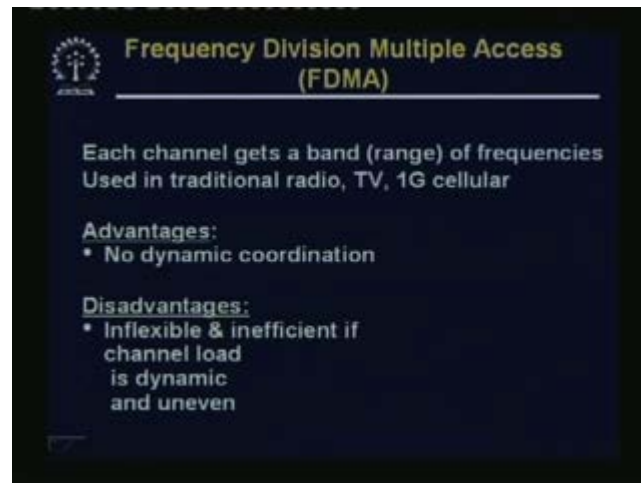
(Refer slide time: 20:35)



This is another picture; we have tried to show with colour the graphs of the different channels. The channels are differently colored (these bars), and we have the three accesses – f is the frequency, c are the channels and t is the time. There is only one channel at all times that uses the particular frequency; another channel for all times uses another frequency, and so on.

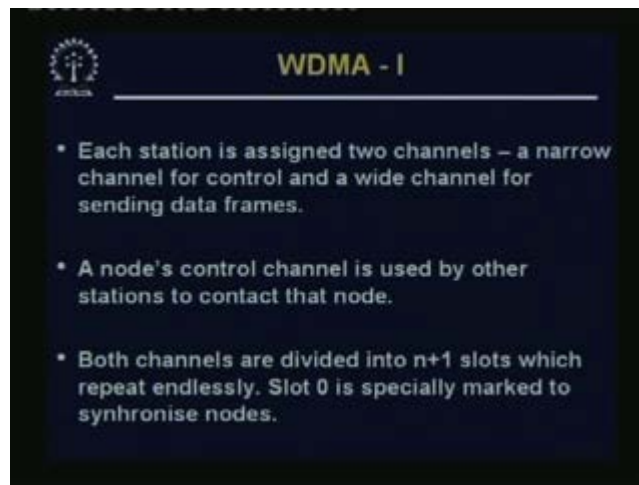
Another frequency means it is not a single frequency; this is a frequency band kind of thing. You always require a band; you cannot communicate anything with a single frequency that will be just a single tone.

(Refer slide time: 21:27)



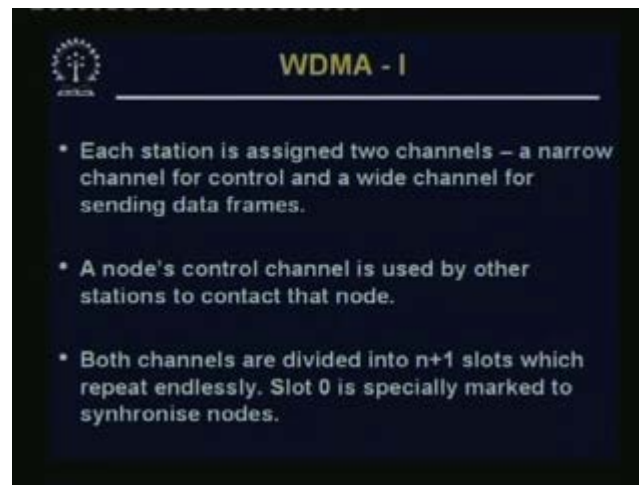
Frequency division multiple access for each channel gets a band (range) of frequencies used in traditional radio, TV, first generation cellular, etc. Advantage is that there is no dynamic coordination; it is absolutely distributed. Disadvantage is that it is inflexible and inefficient if channel load is dynamic and uneven. By today's standards, it is quite high, but at one point of time it was a quite good.

(Refer slide time: 21:57)



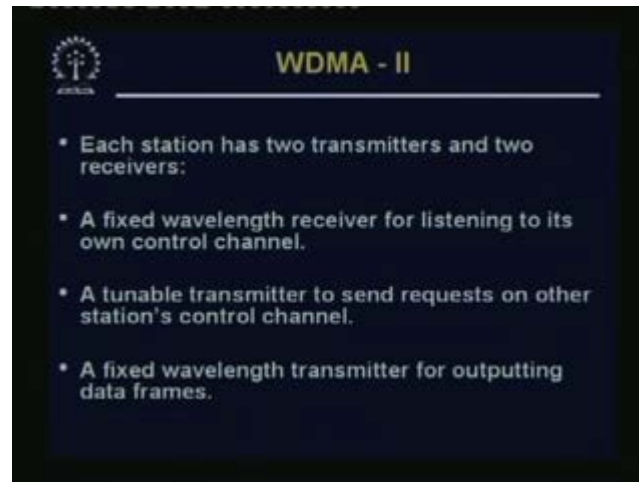
We will see another version of this FDMA, which is WDMA. WDMA means wavelength division multiple access; and as you can suspect, now we are talking about fibers. We just have a frequency division multiplexing and wavelength division multiplexing. They were actually the same thing; the only thing is that for fiber we choose to call it wavelength division multiplexing. Here also, we are talking about wavelength division multiple access. Previously all our fiber links that we talked about were point to point. Suppose the fiber link somehow is used for multiple access, two stations in a very dynamic fashion, we want to give this bandwidth. A number of stations can connect to the same fiber through some splitter, coupler, etc. So we are not going into that. Suppose somehow they are hooked into this fiber and we want to use wavelength division multiple access. Let us just look at the scheme.

(Refer Slide time: 23:03)



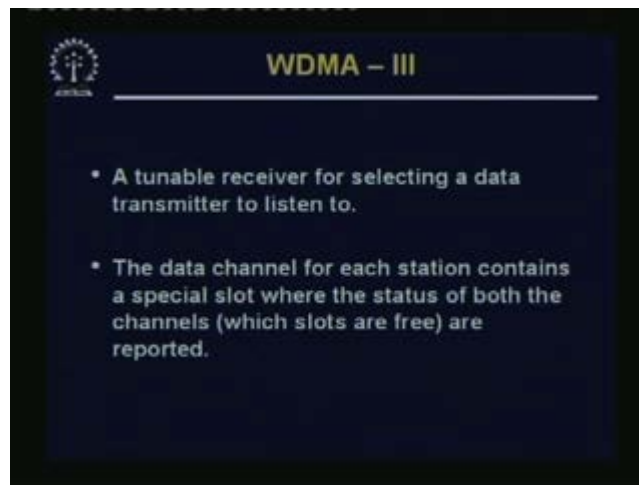
Each station is assigned two channels; a narrow channel for control and a wide channel for sending data frames. The channels could also be of the same size, does not matter, but then for control we do not require so much bandwidth. For a particular node, these two are assigned. A node's control channel is used by other stations to contact that node. So its control channel is fixed. Some other station, if it wants to communicate with A, will first talk to A on its control channel. Both channels are divided into $n + 1$ slots, which repeat endlessly. Slot 0 is especially marked to synchronize the nodes. In each channel, there are, say, $n + 1$ slots and then a number of nodes; let us say, three nodes, B, C and D can communicate with A at the same time. Because of communicating to A it has to send to A in the channel or the wave length which has been assigned to A. But how can all three send at the same time? If there are slots in that channel and if they have booked separate slots, in different slots their data can go. So B, C, and D simultaneously communicate with A.

(Refer slide time: 24:33)



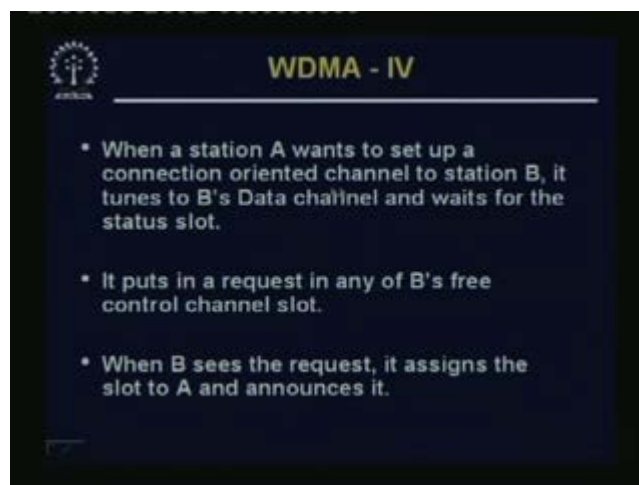
Each station has two transmitters and two receivers. First it has to have a fixed wavelength receiver for listening to its own control channel. Its control channel is fixed, so it requires a fixed wavelength receiver. Then it wants to communicate to somebody else's control channel. For that it requires another transmitter and that has to be tunable because A may want talk to B at one point of time and then talk to C at some other point of time B and C's control channels are different. So we sort of have to have a tunable transmitter. We have a fixed wavelength transmitter for outputting data frames. Suppose it wants talk to B it can talk to B's control channel, when A wants to communicate over the wavelength assigned to it. Then it can output the data frame at a fixed wavelength.

(Refer slide time: 25:43)



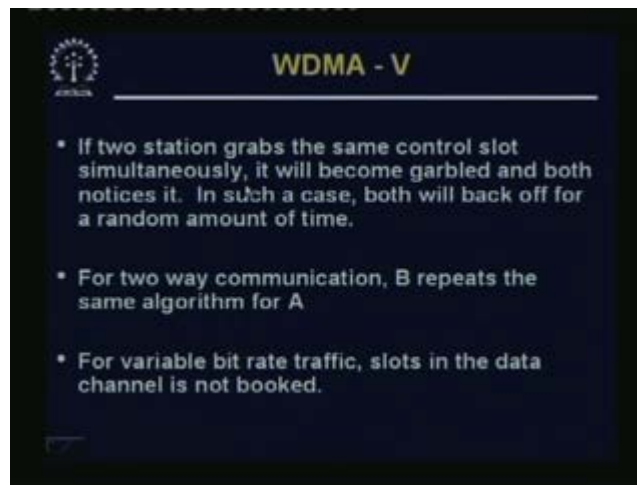
It requires the tunable receiver for selecting a data transmitter to listen to. A is talking to B, so for talking to or listening to B it requires a tunable transmitter because it may want to listen to B now and later on, it may want to listen to C. The data channels for each station contains a special slot where the status of both the channels, which slots are free, etc., are reported.

(Refer slide time: 26:10)



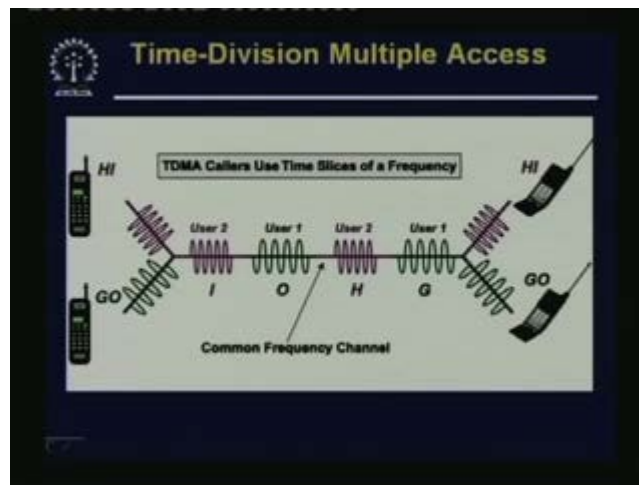
When station A wants to set up a connection oriented channel to station B, it tunes to B's data channel and waits for the status slot. It puts in the request in any of B's free control channel slots. B's status slot will tell whatever control channel slots are free. So B just puts in a request over there. When B sees the request, it assigns the slot to A and announces it to others.

(Refer slide time: 26:37)



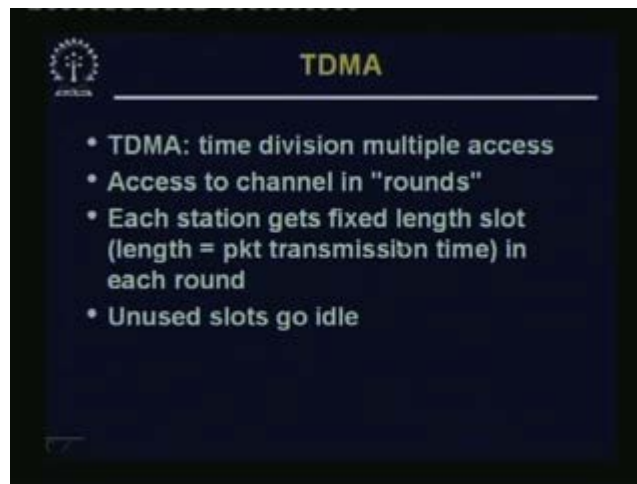
There is a problem in a distributed system, in the sense that you always have to worry about this kind of thing. If two stations grab the same control slots simultaneously, both of them want to talk to B and both of them sort of want to grab that same control slot by some chance simultaneously. When we say simultaneously, they both are close together, and so the information will get garbled. So naturally if both of them are trying to put things in the same box the information will get garbled and both of them will notice. In that case both will back off for a random amount of time. When they do back off for a random amount of time, the random number generated by one station and the random number generated by the other station will be different. So both of them will back off for two different amounts of time, and whichever has the shorter waiting time, comes back and tries to put again on this control slot. Hopefully there will be no collision. This is one kind of collision and backing off. This kind of scheme is used elsewhere also; we will come to that. For two-way communication, B repeats the same algorithm for A for variable bit rate traffic. This is for synchronizing, so one slot can be sort of reserved for this communication between A and B. For variable bit rate traffic, slots in data channel are not booked so they can be sent by some other mechanism. Let us not go into that at the moment.

(Refer slide time: 28:19)



Next, we come to time division multiple access. If you notice the figure, this looks just like time division multiplexing. We have the information or signal from this top one, high one coming and then it is sort of interspersed with the signal from this source go. They are going in a time division multiplex fashion. Only thing to notice over here is that this line and this line they may not be actually physical wires. That is why some wireless has been put over here. So they are actually sharing the same medium. So immediately you see the problem – what happens if this synchronizing behavior breaks down. So somehow you have to make sure that they do not break down and they do communicate at precise slots of time, which is assigned to them. Then you have the additional trouble of ensuring that the clocks synchronize. If he is running a clock of different zone these two clocks may not agree, so how to synchronize them, etc. These are the problems which you solve in a TDMA, time division multiple access. Otherwise the basic approach is the same as time division multiplexing.

(Refer slide time: 30:00)

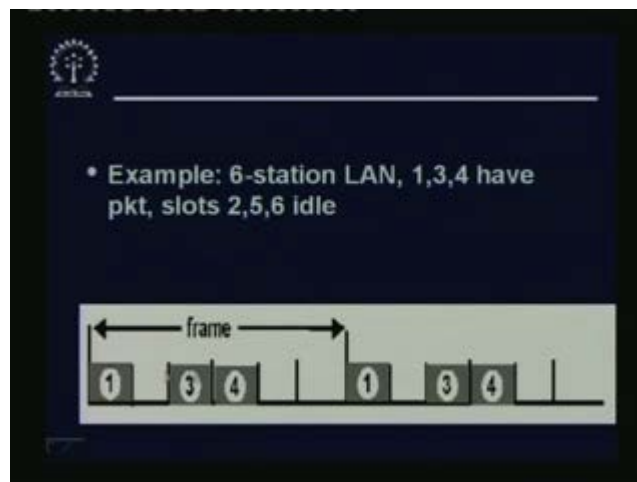


TDMA

- TDMA: time division multiple access
- Access to channel in "rounds"
- Each station gets fixed length slot (length = pkt transmission time) in each round
- Unused slots go idle

TDMA is time division multiple access. It enables access to channel in rounds. Each station gets fixed length slots, length is equal to packet transmission time in each round. The unused slots go idle; this is a simple scheme.

(Refer slide time: 30:16)

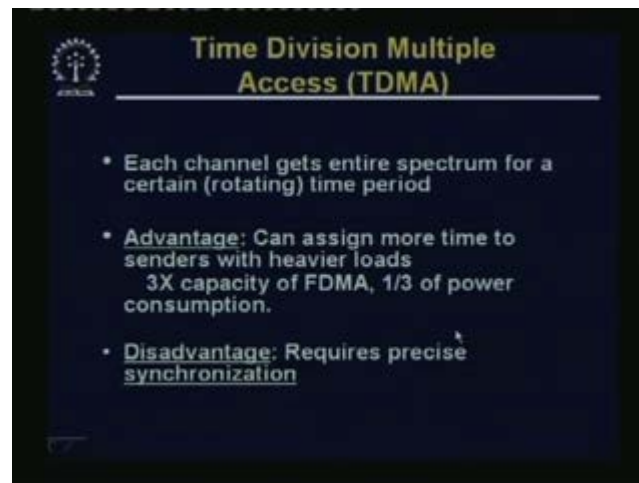


• Example: 6-station LAN, 1,3,4 have pkt, slots 2,5,6 idle

Diagram illustrating a frame structure for a 6-station LAN. The frame is divided into six slots. Slots 1, 3, and 4 contain packets (labeled 1, 3, and 4 respectively). Slots 2, 5, and 6 are idle. The sequence of slots is 1, 3, 4, 1, 3, 4, indicating a repeating pattern.

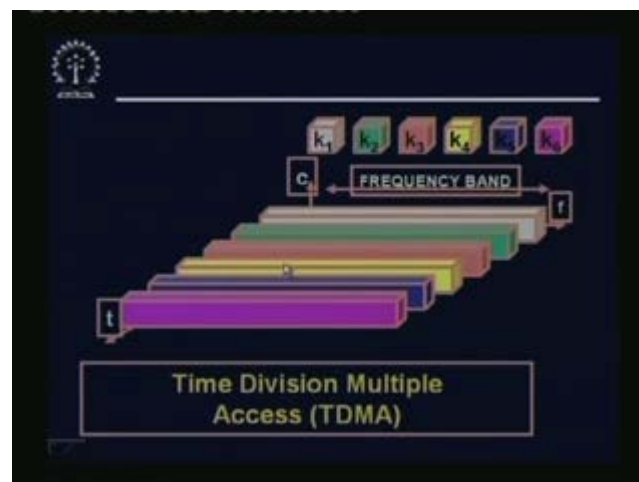
For example, suppose we have six stations: LAN 1,3,4 have packet and slots 2,5 and 6 are idle. This kind of frames sequence keeps on repeating. In each frame, 1 will have a slot, 2 will have a slot which is going idle, 3 will have a slot, 4 will have a slot, 5 and 6 will also have slots which are again going idle and so on. So this is a TDMA system.

(Refer slide time: 30:50)



Each channel gets entire spectrum for a certain time period, rotating time period. So whenever the time is allotted to that particular channel it gets the entire spectrum to itself, so it can send the data at a very high rate. The advantage is that it can assign more time to senders with heavier loads. By doing that, it gains some kind of efficiency advantage over FDMA and another side of the same point is that you can reduce the power consumption. The disadvantage is that it requires precise synchronization, which we talked about earlier.

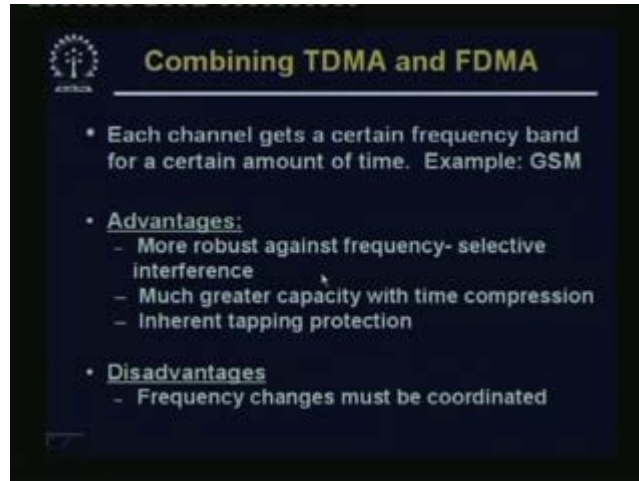
(Refer slide time: 31:37)



This is a diagram for the time division multiple access. The entire frequency band is given to one particular channel for a short amount of time. Then, for the next short amount of time,

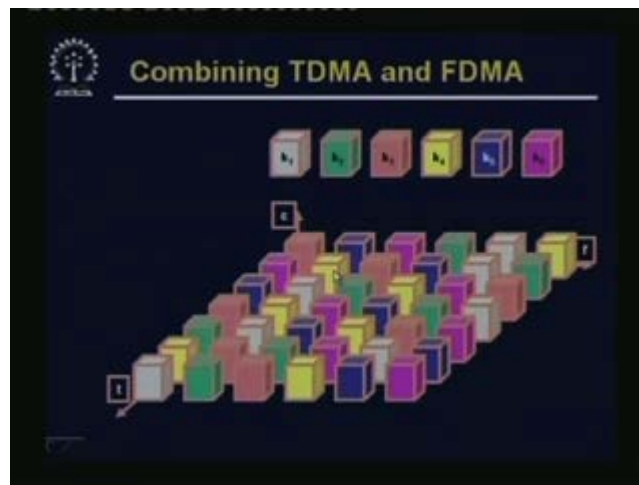
another channel gets the entire frequency band and so on. As it is there in standard time division multiplexing, the same thing applies.

(Refer slide time: 32:00)



Now we can combine TDMA and FDMA. Each channel gets a certain frequency band for a certain amount of time. An example is GSM; we will come to this may be in slightly more detail later on. So, this is combining both TDMA and FDMA; that means, we are breaking it up not only in the time domain but also in the frequency domain. The advantage is that there is more robust against frequency selective interference because for a particular communication the frequency is sort of changing; there is much greater capacity with time compression; and there is inherent tapping protection. So it has got a lot of advantages and it is quite efficient. The disadvantage is that, once again, just as in TDMA, we have to synchronize the clocks, etc. So here, frequency changes must also be coordinated.

(Refer slide time: 32:52)



This is the picture; basically, we have small blocks. So for this particular yellow block over here, at this particular time, this particular frequency range is given to this channel and so on.

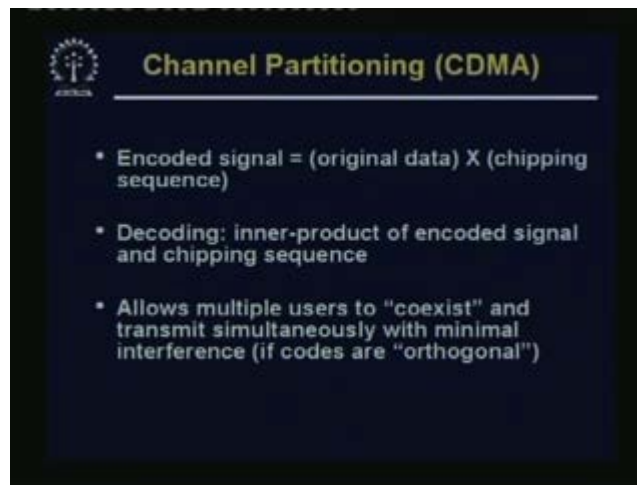
(Refer slide time: 33:10)

The slide, titled "Channel Partitioning (CDMA)", features a dark blue background with yellow text. It includes a logo in the top left corner. The content is organized as follows:

- CDMA (Code Division Multiple Access)
- Unique "code" assigned to each user; i.e., code set partitioning
- Used mostly in wireless broadcast channels (cellular, satellite, etc)
- All users share same frequency, but each user has own "chipping" sequence (i.e., code) to encode data

Next we come to channel partitioning or CDMA. CDMA is code division multiple access. This may be a little interesting and new. We have a unique code assigned to each user, that is, code set partitioning. What is that code? We will come to that. It is used mostly in wireless broadcast channels like cellular satellites, etc. All users share the same frequency, but each user has his own chipping sequence, that is, code, to encode data. We will come to the details.

(Refer slide time: 33:54)

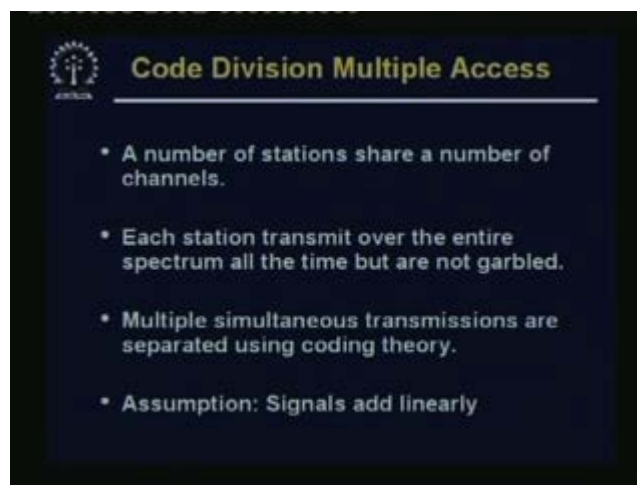


Channel Partitioning (CDMA)

- Encoded signal = (original data) X (chipping sequence)
- Decoding: inner-product of encoded signal and chipping sequence
- Allows multiple users to "coexist" and transmit simultaneously with minimal interference (if codes are "orthogonal")

Encoded signal is the original data; and then there is some operation with the chipping sequence. We will come to what is chipping sequence. Decoding is the inner product of encoded signal and chipping sequence; it allows multiple users to coexist and transmit simultaneously with minimal interference if codes are really orthogonal.

(Refer slide time: 34:14)



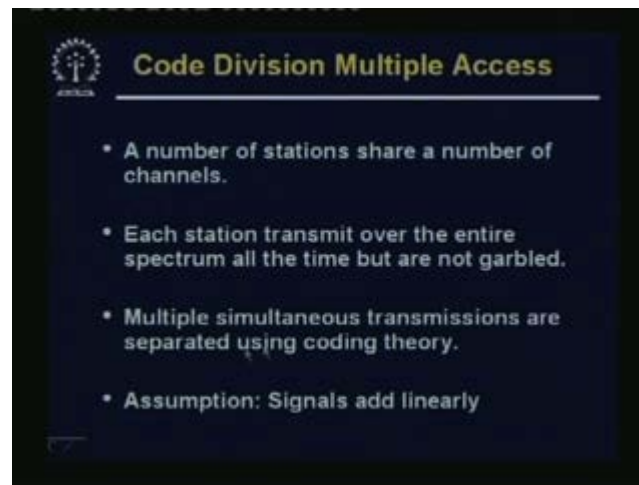
Code Division Multiple Access

- A number of stations share a number of channels.
- Each station transmit over the entire spectrum all the time but are not garbled.
- Multiple simultaneous transmissions are separated using coding theory.
- Assumption: Signals add linearly

We have a number of stations sharing a number of channels. Each station transmits over the entire spectrum all the time, which means that it is not excluded either in a time dimension or in a frequency dimension. Each channel can go on communicating in the entire frequency band all the time. Now how will that happen – because other people are also communicating in

the entire frequency band all the time. All their signals will get mixed up; as a matter of fact, they do get mixed up, but the point is that the signals are encoded in such a clever fashion that from that mixed signal you can separate out all the different streams of communication that have gone into it.

(Refer slide time: 35:06)

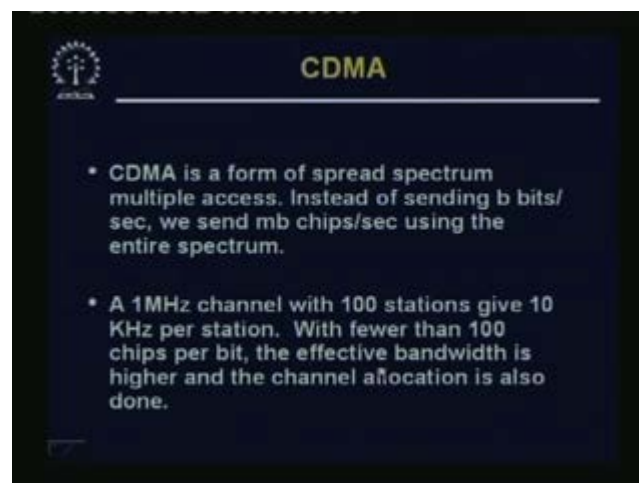


The slide is titled "Code Division Multiple Access" in yellow text on a dark blue background. It features a small circular logo with a cross-like symbol in the top left corner. The content consists of four bullet points in white text:

- A number of stations share a number of channels.
- Each station transmit over the entire spectrum all the time but are not garbled.
- Multiple simultaneous transmissions are separated using coding theory.
- Assumption: Signals add linearly

So multiple simultaneous transmissions are separated using coding theory. There is an assumption that the signals add linearly; if they do not, then you have to do some adjustment, etc. We will not go into that.

(Refer slide time: 35:21)

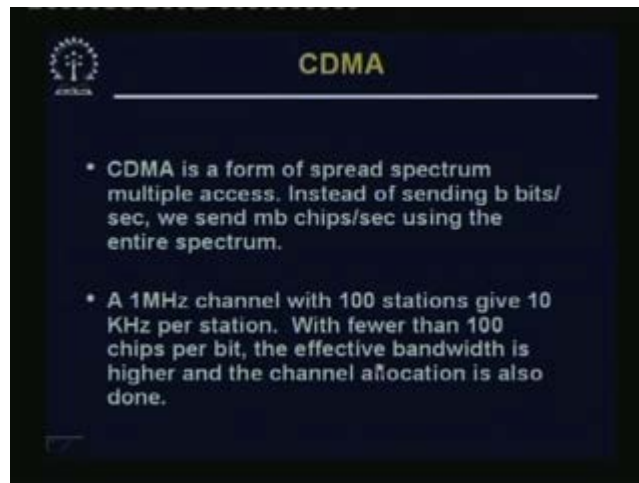


The slide is titled "CDMA" in yellow text on a dark blue background. It features a small circular logo with a cross-like symbol in the top left corner. The content consists of two bullet points in white text:

- CDMA is a form of spread spectrum multiple access. Instead of sending b bits/sec, we send mb chips/sec using the entire spectrum.
- A 1MHz channel with 100 stations give 10 KHz per station. With fewer than 100 chips per bit, the effective bandwidth is higher and the channel allocation is also done.

CDMA is a form of spread spectrum multiple access. It is spread over the entire spectrum. Instead of sending b bits per second for a particular node, we send mb chips per second. What is a chip? Each bit is encoded by m number of chips.

(Refer slide time: 35:29)



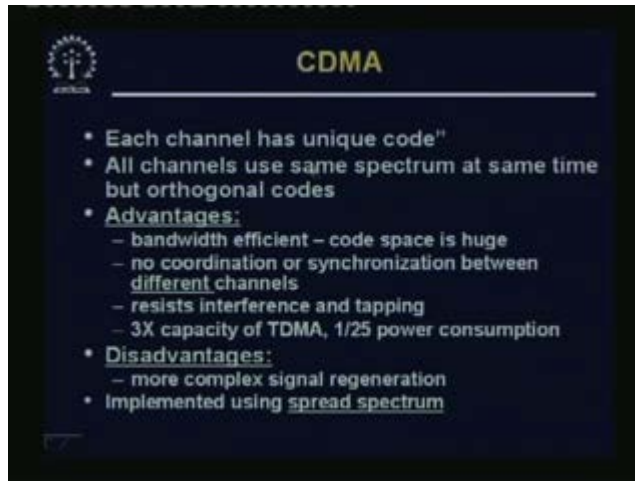
They are sort of tiny fragments of bits and these chips may again be 0 and 1. There is a 0 1 sequence code for these bits of one particular station; another station will have another code. A 1 MHz channel with 100 stations gives 10 KHz per station. With fewer than 100 chips per bit, the effective bandwidth is higher and the channel allocation is also done at the same time; we will see how.

(Refer slide time: 36:18)



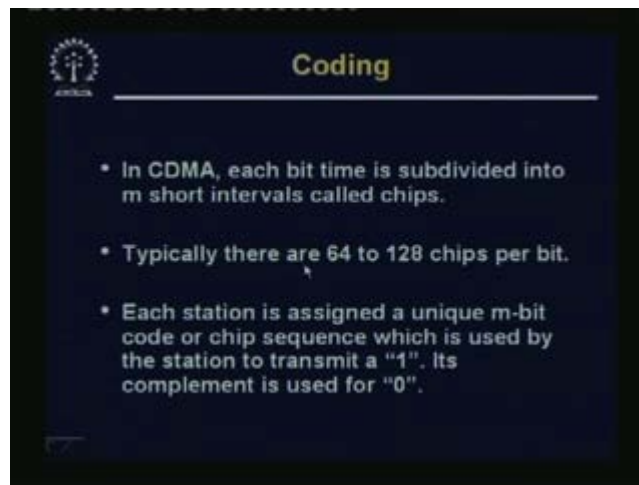
This is the picture, which is funny because for the entire time for the entire frequency band, all the channels are using it at the same time

(Refer slide time: 36:30)



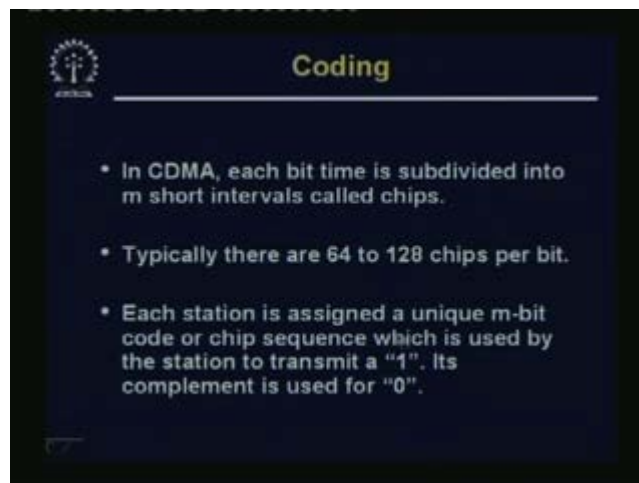
Now each channel has a unique code. All channels use the same spectrum at the same time but orthogonal codes. It is bandwidth efficient; its capacity also is quite good, when I talk about simple TDMA. But you can sort of mix up FDMA and TDMA and get a good efficiency over there also. This fight between GSM and CDMA is ongoing and it will go on for some time. The disadvantage is that it has more complex signal regeneration. So this is how it is implemented.

(Refer slide time: 37:10)



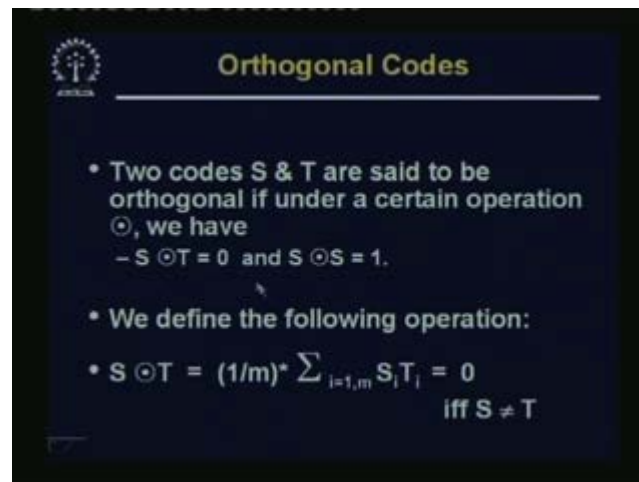
As I said, each bit time is subdivided into short intervals called chips and typically there are 64 to 128 chips per bit. That means for each bit we have a code, which is 64 bits long. The chips are again in a sequence of 1s and 0s. So we are sending a long sequence of 1s and 0s for sending may be 1. But we are sending it very fast; and we can send it very fast because we are using the entire spectrum. The entire spectrum is at our disposal; so we can send it very fast.

(Refer slide time: 38:04)



Each station is assigned a unique m bit code or chip sequence, which is used by the station to transmit 1; its complement is used for 0.

(Refer slide time: 38:13)

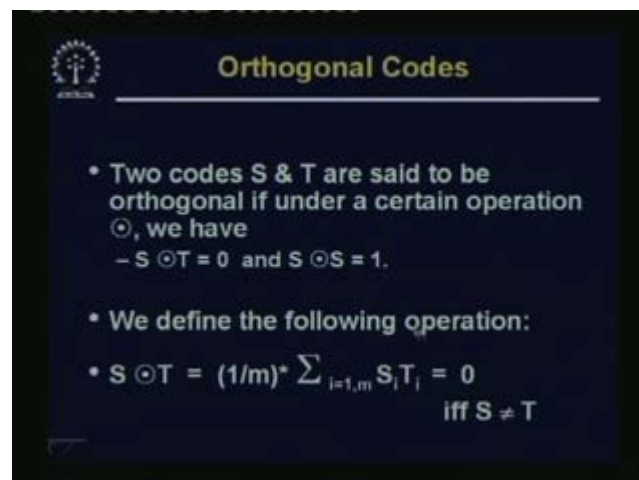


Orthogonal Codes

- Two codes S & T are said to be orthogonal if under a certain operation \odot , we have
– $S \odot T = 0$ and $S \odot S = 1$.
- We define the following operation:
- $S \odot T = (1/m) * \sum_{i=1,m} S_i T_i = 0$
iff $S \neq T$

We will see an example: two codes, S and T, are said to be orthogonal; we have been talking about orthogonal codes. If under a certain operation we have S let me call it a dot product kind of thing at the moment. So S dot T is equal to 0 and S dot S is equal to 1. You see that this is very similar to dot products in vector. If you have two orthogonal vectors, the dot product is going to be 0, whereas on the same vector when we take its dot product with itself it is going to give some value. We are representing it as 1. We define the following.

(Refer slide time: 38:57)

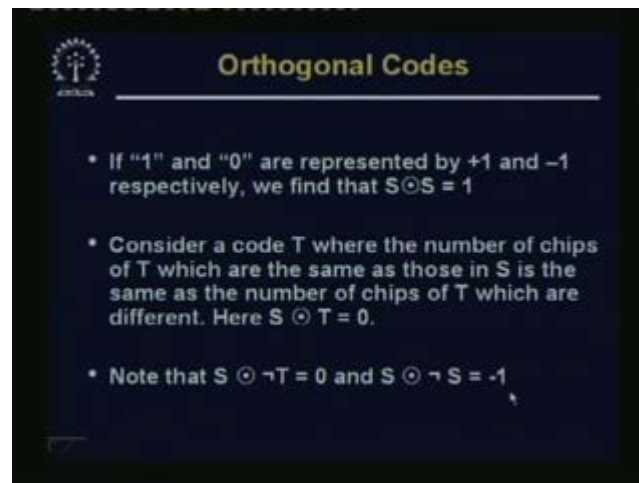


Orthogonal Codes

- Two codes S & T are said to be orthogonal if under a certain operation \odot , we have
– $S \odot T = 0$ and $S \odot S = 1$.
- We define the following operation:
- $S \odot T = (1/m) * \sum_{i=1,m} S_i T_i = 0$
iff $S \neq T$

Let us define this dot operation for our case. So S dot T is equal to $1/M \times$ summation of $S_i T_i$; i is equal to 1 to m and this will be equal to 0 if and only if S is not equal to T. We will see this.

(Refer slide time: 39:20)

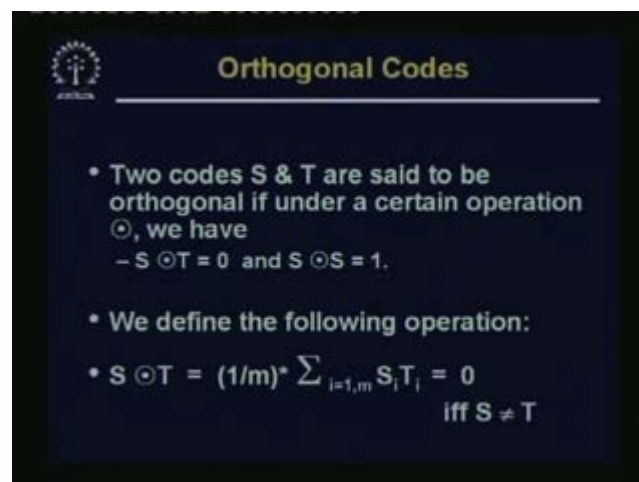


Orthogonal Codes

- If "1" and "0" are represented by +1 and -1 respectively, we find that $S \odot S = 1$
- Consider a code T where the number of chips of T which are the same as those in S is the same as the number of chips of T which are different. Here $S \odot T = 0$.
- Note that $S \odot \neg T = 0$ and $S \odot \neg S = -1$

If one and zero are represented by +1 and -1 respectively, we find that S dot S equal to one. This is a very simple because if the ones have been represented as one and zero has been represented as a -1, so S dot S so -1 will get multiplied with -1 giving you +1 and one and one will also give you one. so all of them would be one. so all M of them would be one.

(Refer slide time: 39:53)

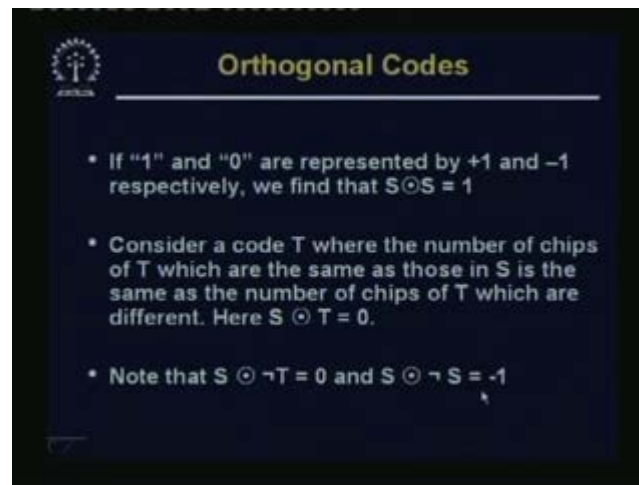


Orthogonal Codes

- Two codes S & T are said to be orthogonal if under a certain operation \odot , we have
 $- S \odot T = 0$ and $S \odot S = 1$.
- We define the following operation:
- $S \odot T = (1/m) \cdot \sum_{i=1, m} S_i T_i = 0$
iff $S \neq T$

The sum total in the previous if you look at the previous definition the sum $S_i S_i$ is going to be a sigma $S_i S_i$ is going to be M; you divide that with M so that will give you a 1. so s dot s would be 1 whereas for any other code, any other T, where the T is orthogonal to S, this sum is going to come out as 0. That is how we cleverly assign the codes.

(Refer slide time: 40:17)

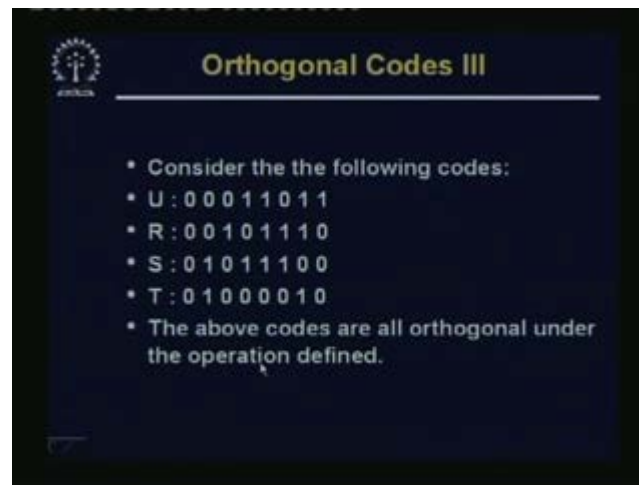


The slide is titled "Orthogonal Codes" and features a logo in the top left corner. It contains three bullet points:

- If "1" and "0" are represented by +1 and -1 respectively, we find that $S \odot S = 1$
- Consider a code T where the number of chips of T which are the same as those in S is the same as the number of chips of T which are different. Here $S \odot T = 0$.
- Note that $S \odot \neg T = 0$ and $S \odot \neg S = -1$

Consider a code T where the number of chips of T, which are the same as those in S is the same as the number of chips of T, which are different. Consider what it is saying: at code T, where the number of chips of T, which are the same as those in S are the same as the number of chips of T which are different, the number of chips of T which are the same as those in S. So the corresponding chips, when multiplied together, will give you so many 1s. If it is the same as the number of chips of T which are different, now if the chips are different in T and S when the corresponding chips are multiplied, they would give you -1. The number of +1s and the number of -1s, if they are the same when you add them together in the previous summation, the sum total will become 0. $S \cdot T$ will be zero. Note that $S \cdot \neg T$; $\neg T$ is the complement of T. That means the 1s and 0s are presented with 1s and -1s. So 1s and -1s are flipped in one of them. So naturally the number of chips which are different become the number of chips which are the same and the number of chips which were same earlier, the chip positions which were same earlier, now become different. But any way their numbers are equal. So once again we have $S \cdot \neg T$ is 0 and $S \cdot \neg S$ – all the 1s in S will be -1 in $\neg S$ and all the -1s in S would be 1 in $\neg S$. So in either case, we will get a product of -1. M -1s added together will give you -M divided by M will give you -1. We have $S \cdot S$ equal to 1; $S \cdot T$ is equal to 0; $S \cdot \neg T$ is equal to 0; $S \cdot \neg S$ equal to -1. So these are the nice properties if we have orthogonal codes.

(Refer slide time: 42:36)

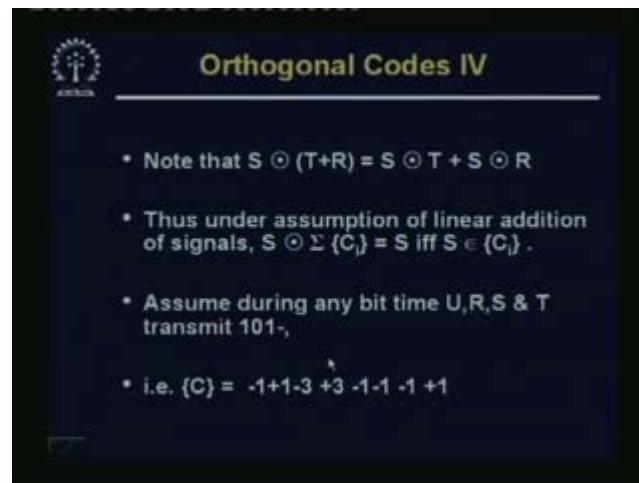


The slide is titled "Orthogonal Codes III" and features a list of four codes: U, R, S, and T. Each code is an 8-bit binary sequence. A bullet point at the bottom states that these codes are all orthogonal under a defined operation.

- Consider the the following codes:
- U : 0 0 0 1 1 0 1 1
- R : 0 0 1 0 1 1 1 0
- S : 0 1 0 1 1 1 0 0
- T : 0 1 0 0 0 0 1 0
- The above codes are all orthogonal under the operation defined.

So consider the following codes: suppose this U, R, S, and T are four different stations. Here a simple example has been shown using only eight chips. You note that between U and R, in one position they are same; in this position they are same; two they are different; in this fifth position they are same. That is the number three of the position in which they are same and then there is the number four position, which is the seventh position, where they are the same. In four positions they are the same and four positions namely the third position, fourth position, sixth position, and eighth position they are different. The number of positions in which they are the same is the same as the number of positions in which they are different. We know that if instead of 0 we have a -1 these codes will come out to be orthogonal. Similarly you would see that U, R, S, T are all orthogonal to each other. So the above codes are all orthogonal under the operation defined.

(Refer slide time 43:45)

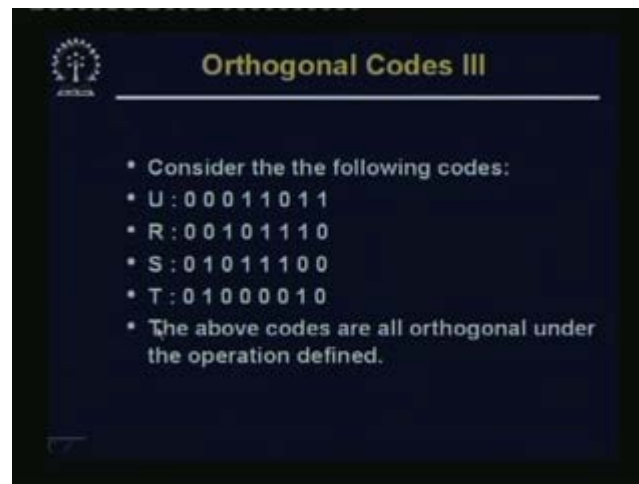


The slide is titled "Orthogonal Codes IV" and features a list of four bullet points. The first bullet point states a distributive property of the dot product. The second bullet point discusses the assumption of linear addition of signals and its implication for the dot product of a signal with a sum of other signals. The third bullet point sets a scenario where four signals (U, R, S, T) transmit the bit sequence 101. The fourth bullet point provides the resulting vector for the sum of these signals, C.

- Note that $S \odot (T+R) = S \odot T + S \odot R$
- Thus under assumption of linear addition of signals, $S \odot \sum \{C_i\} = S$ iff $S \in \{C_i\}$.
- Assume during any bit time U, R, S & T transmit 101,
- i.e. $\{C\} = -1+1-3+3 \ -1-1 \ -1+1$

Note that $S \cdot T + R$ is $S \cdot T + S \cdot R$, which means that under normal addition, if you take this is normal addition, this dot product is going to distribute. Thus under the assumption of linear addition of signals, $S \cdot \sum C_i$ is equal to S if and only if S is in C_i . That means suppose there are a number of channels, which are transmitting. In that case, we have already assumed linear addition of signal strengths. This is nothing but some of the signal strengths like Q, T, R, U, S , etc., have been all added up. Now $S \cdot$ will distribute over this summation, so we have $S \cdot T$ and $S \cdot R$ etc. If S happens to be in this set C_i , then that $S \cdot S$ will come out as 1. It should be 1 if and only if S is in C_i ; otherwise it will come out as 0. Assuming during any bit time U, R, S , and T transmit 101 and nothing, U is trying to transmit 1, R is trying to transmit 0, S is trying to transmit 1, and T is not transmitting at all.

(Refer slide time: 45:10)

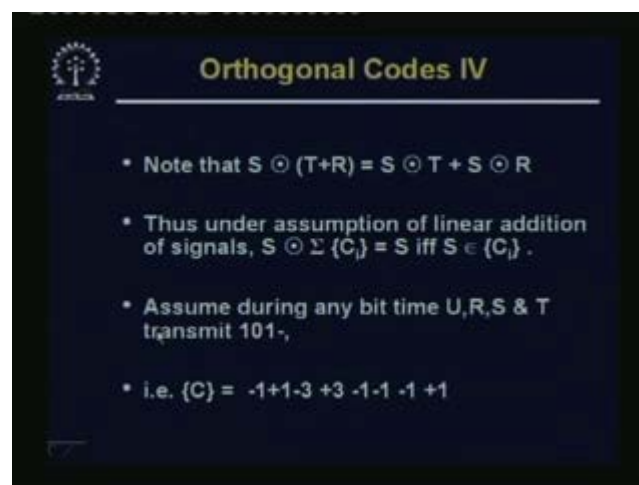


Orthogonal Codes III

- Consider the the following codes:
- U : 0 0 0 1 1 0 1 1
- R : 0 0 1 0 1 1 1 0
- S : 0 1 0 1 1 1 0 0
- T : 0 1 0 0 0 0 1 0
- The above codes are all orthogonal under the operation defined.

If you use the code shown in the previous slide over here and then if you do the calculation,

(Refer slide: 45:14)

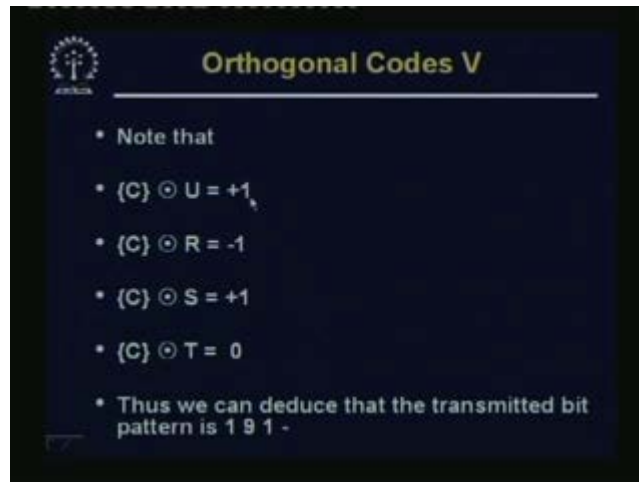


Orthogonal Codes IV

- Note that $S \odot (T+R) = S \odot T + S \odot R$
- Thus under assumption of linear addition of signals, $S \odot \sum \{C_i\} = S$ iff $S \in \{C_i\}$.
- Assume during any bit time U,R,S & T transmit 101-,
- i.e. $\{C\} = -1+1-3+3 -1-1 -1+1$

This will come out to be like this: the signal strength as we see it, will come out as $-1 +1 -3 +3$
 $-1 -1 -1 +1$ – so these are the eight signal levels we get.

(Refer slide time: 45:31)

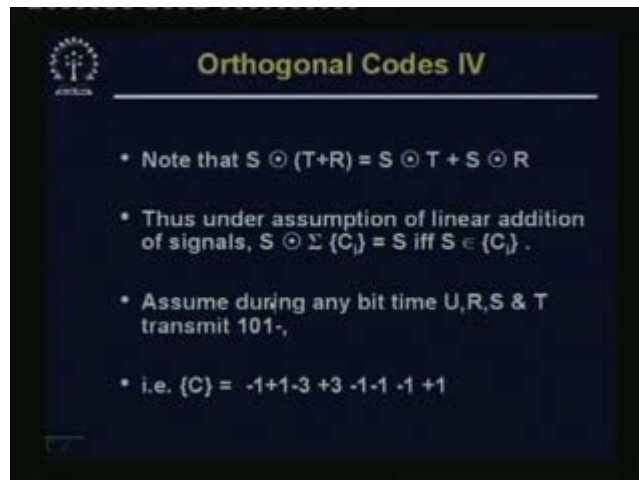


Orthogonal Codes V

- Note that
- $\{C\} \odot U = +1$
- $\{C\} \odot R = -1$
- $\{C\} \odot S = +1$
- $\{C\} \odot T = 0$
- Thus we can deduce that the transmitted bit pattern is 1 0 1 -

Now if we do that, C dot U will get a +1. That means C dot is trying to send 1; C dot R will be -1; that means R is trying to send 0; C dot S will be +1; that means S is trying to send 1; and C dot T would be 0; that means C is not sending anything at all.

(Refer slide time: 45:51)

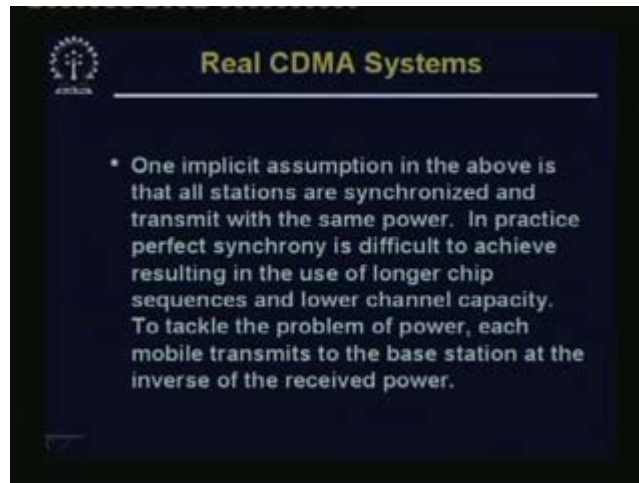


Orthogonal Codes IV

- Note that $S \odot (T+R) = S \odot T + S \odot R$
- Thus under assumption of linear addition of signals, $S \odot \sum \{C_i\} = S$ iff $S \in \{C_i\}$.
- Assume during any bit time U, R, S & T transmit 101-,
- i.e. $\{C\} = -1+1-3+3-1-1-1+1$

If you remember, 101- that means a S, R, U and T are transmitting in this fashion.

(Refer slide time: 46:04)



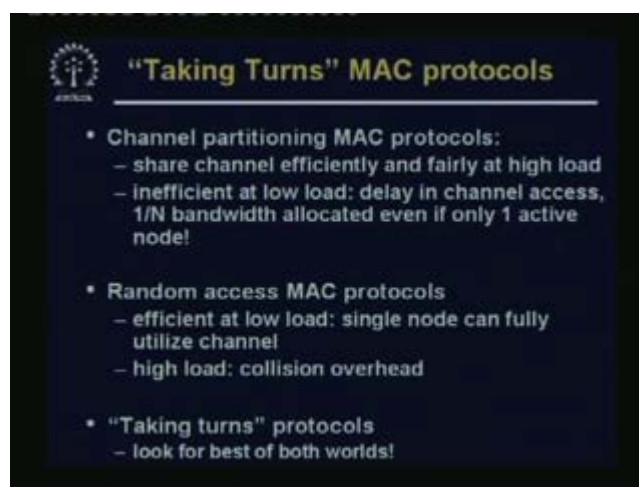
The slide features a dark blue background with a small circular logo in the top left corner. The title "Real CDMA Systems" is centered at the top in a yellow font. A single bullet point is listed below the title.

Real CDMA Systems

- One implicit assumption in the above is that all stations are synchronized and transmit with the same power. In practice perfect synchrony is difficult to achieve resulting in the use of longer chip sequences and lower channel capacity. To tackle the problem of power, each mobile transmits to the base station at the inverse of the received power.

One implicit assumption in the above is that all stations are synchronized and transmit with the same power. In practice, perfect synchrony is difficult to achieve resulting in the use of longer chip sequences and lower channel capacity. Secondly, to tackle the problem of power, each mobile transmits to the base station at the inverse of the received power. So there are practical limitations. Although this looks very nice, what happens is that there are practical limitations and the theoretical maximum that we could achieve is less than that but this is an elegant system.

(Refer slide time: 46:50)



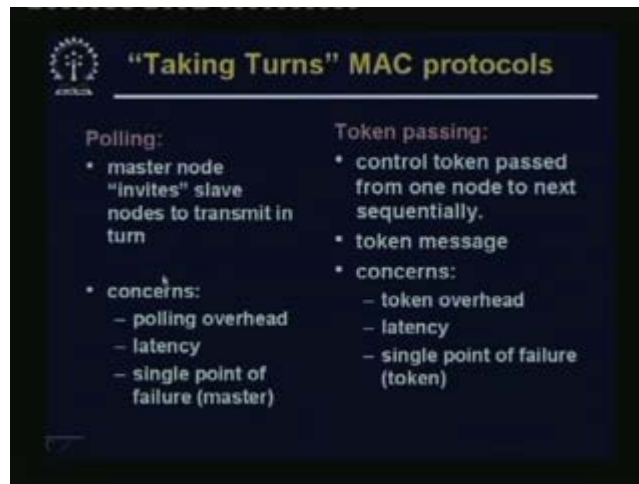
The slide features a dark blue background with a small circular logo in the top left corner. The title ""Taking Turns" MAC protocols" is centered at the top in a yellow font. Three bullet points are listed below the title.

"Taking Turns" MAC protocols

- Channel partitioning MAC protocols:
 - share channel efficiently and fairly at high load
 - inefficient at low load: delay in channel access, $1/N$ bandwidth allocated even if only 1 active node!
- Random access MAC protocols
 - efficient at low load: single node can fully utilize channel
 - high load: collision overhead
- "Taking turns" protocols
 - look for best of both worlds!

Now we will look at some MAC protocols: two of them, two small ones we will discuss in this lecture, and then, in the next lecture, we are going to take two of the more involved ones. Taking turns MACs protocols: do you remember that we had channel partitioning MAC protocols and random access MAC protocols and taking turns protocols? In taking turns, what it tries to do is that it sort of allocates the turn and we will see how.

(Refer slide time: 47:19)

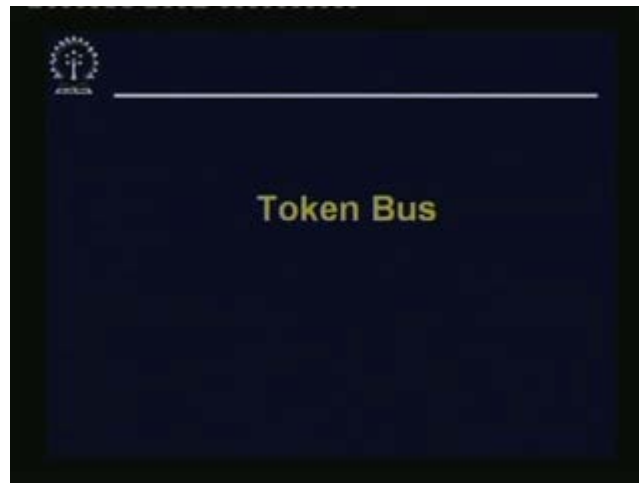


The slide is titled "Taking Turns" MAC protocols. It compares two protocols: Polling and Token passing. The Polling section lists that a master node "invites" slave nodes to transmit in turn, with concerns for polling overhead, latency, and a single point of failure at the master. The Token passing section lists that a control token is passed sequentially from one node to the next, with a token message and concerns for token overhead, latency, and a single point of failure at the token itself.

Polling:	Token passing:
<ul style="list-style-type: none">• master node "invites" slave nodes to transmit in turn	<ul style="list-style-type: none">• control token passed from one node to next sequentially.
<ul style="list-style-type: none">• concerns:<ul style="list-style-type: none">– polling overhead– latency– single point of failure (master)	<ul style="list-style-type: none">• token message• concerns:<ul style="list-style-type: none">– token overhead– latency– single point of failure (token)

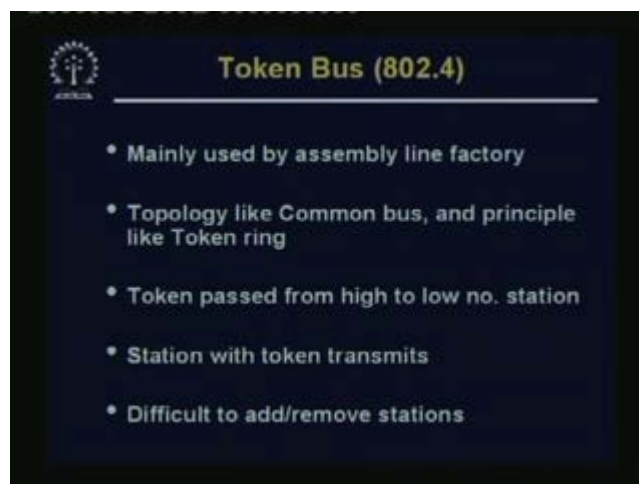
One could be through polling by a master node, which is some kind of a centralized system. There is polling overhead, latency, single point of failure, etc., we are not going to discuss this at this point of time. We are going to focus on this in the next; that is token bus control. A token is passed from one node to the next sequentially; there is a token message that concerns token overhead, latency, single point, etc. We will see some systems based on tokens now, which are sort of using MAC taking turns.

(Refer slide time: 47:56)



And the first example we are going to talk about is the token bus.

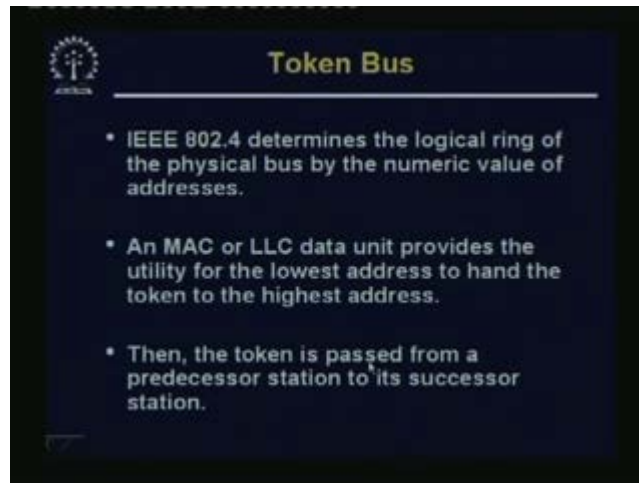
(Refer slide time: 48:01)



So they may be mainly used by assembly line factory. It is used in factories for the main reason that in the other kind of system, which is the contention based system, there is some randomness in the way communication can happen. There may not be any hard and fast guarantee, which people in process control and factories, etc. may not like. So they may use this token bus, which has got the IEEE number 802.4. Token bus is just like a common bus and the principle is like a token ring. Token is passed from high to low number of station. What is a token? Token is some kind of a bit pattern, which is passed from high to low number of

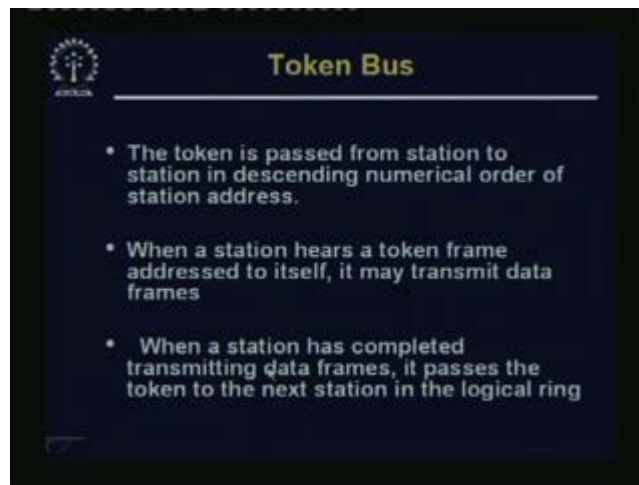
stations. The station with token will transmit. It is difficult to add and remove stations in this particular case.

(Refer slide: 49:06)



So IEEE 802.4 determines the logical ring of the physical bus by the numerical value of the addresses. A MAC or LLC data unit provides the utility for the lowest address to handle the token to the highest address. Otherwise, the higher address gives the token to the lower address so the predecessor gives the token to the successor then the successor gives the token to the next successor and so on, all the way down the chain. Then there is some protocol for sending from the lowest one to the highest one in one group. Then the token is passed from a predecessor station to the successor station. so this a sort of taking turns kind of thing.

(Refer slide time: 49:51)



The token is passed from stations to stations in a descending numerical order of station address. When a station hears a token frame, that means it gets a token frame addressed to itself, it may transmit data frames. That means when it gets the token, which is addressed to itself, and which has been sent by the node which is just higher in number, which is its predecessor, that will be sending the token address to the next station. At that point of time, it may transmit its data frame. When a station has completed transmitting data frames, it passes the token to the next station in the logical ring. So if all of them are trying to transmit they will sort of form a nice queue and then they will come back in a very regular fashion; that is the worst case.

(Refer slide time: 50:36)



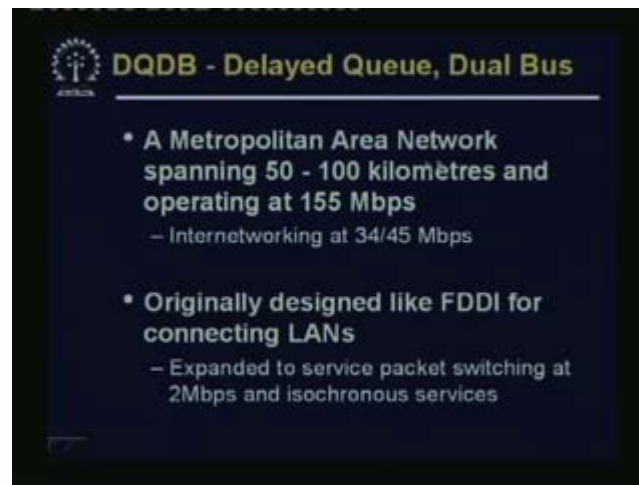
At some particular point of time, a node may not have anything to communicate. In that case, it will simply pass on the token to the next lower address, next lower MAC address. By the way, I have used the term MAC address earlier also. MAC is for Medium Access Control; for that, we require some kind of address. A particular node, if it has nothing to transmit, will give the control or the token to the next lower MAC address and then the next lower MAC address will, if it has something to transmit, will transmit. What is the worst case? The worst is that when everybody wants to transmit; that will take some time, but there is a bound to that time and after that time your turn will come back again. So there is a bound to the worst case performance in this token bus. But this is not very efficient, and not very fast. So it is getting replaced now, but even in some factories it is still there. We are going to talk about another technology, named DQDB, which was once proposed – it also had a name S M P S – as a solution for metro networks and that means for metro networking, DQDB was suggested and some were implemented. But DQDB is once again going out. It has a similar kind of principle; only thing is that instead of a single bus we now have two buses. So let us look at DQDB very quickly.

(Refer slide time: 52:20)



DQDB is 802.6.

(Refer slide time: 52:24)

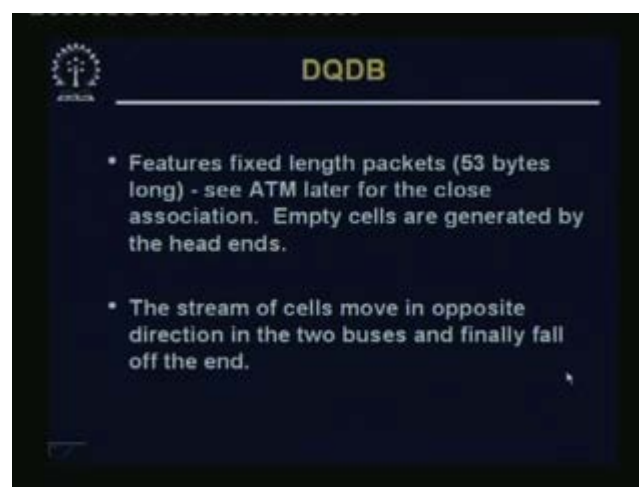


DQDB - Delayed Queue, Dual Bus

- A Metropolitan Area Network spanning 50 - 100 kilometres and operating at 155 Mbps
 - Internetworking at 34/45 Mbps
- Originally designed like FDDI for connecting LANs
 - Expanded to service packet switching at 2Mbps and isochronous services

So it is for a metropolitan area network spanning may be 50 to 100 kilometers and operating at a 34 to 45, even 155 mbps speed was talked about at one point of time. This did not work out quite well later on; but any way, that is a difference story. It was originally designed like FDDI – we will be talking about FDDI in the next lecture – for connecting LANs; expanded to service packet switching at 2 Mbps and isochronous services. Isochronous means nearly synchronous.

(Refer slide time: 52:54)

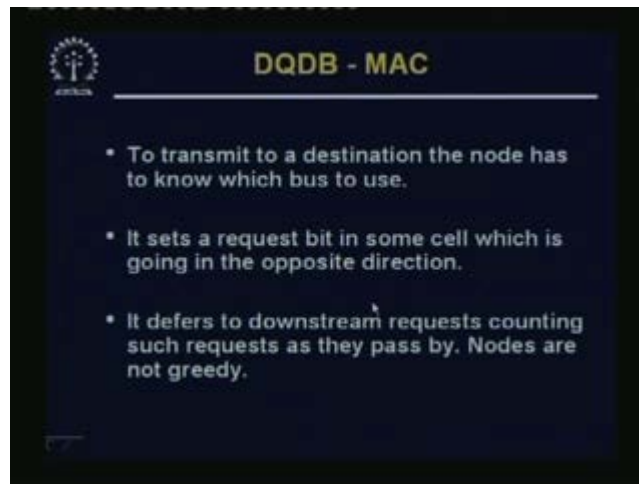


DQDB

- Features fixed length packets (53 bytes long) - see ATM later for the close association. Empty cells are generated by the head ends.
- The stream of cells move in opposite direction in the two buses and finally fall off the end.

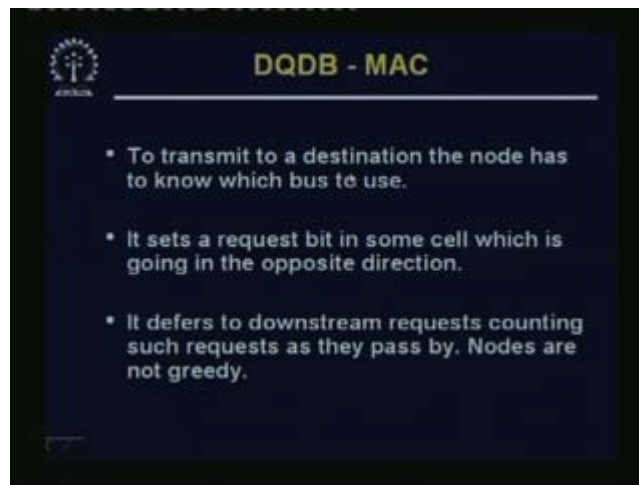
It features fixed length packets like 53 bytes long, and we will see ATM later, which was inherited from this DQDB later on. ATM is still a strong technology even today. Empty cells are generated by the head ends. So there are two buses as I mentioned, and then, at the end of the bus, there are these head ends, which generate a stream of empty cells. The streams of cells move in the opposite direction in the two buses and finally fall off the other end.

(Refer slide time: 53:31)



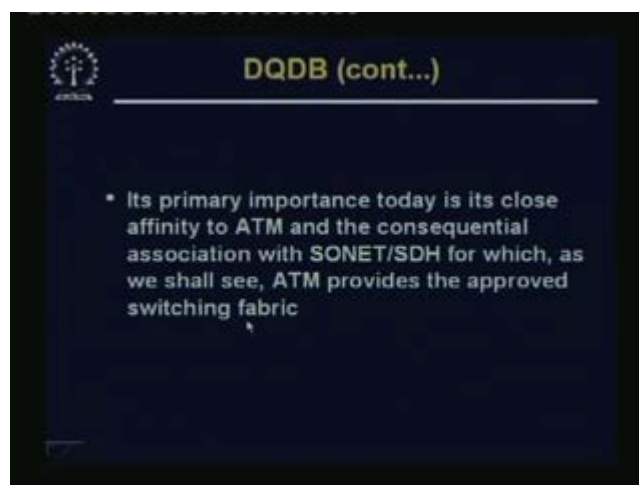
So to transmit to a destination, the node has to know which bus to use; that means whether it is to the left or right, or up or down. So it has to know which bus to use. So every node must be having this information. It sets a request bit in some cell, which is going in the opposite direction. This is for telling suppose the node to which it wants to communicate is downstream, say towards the left, then it tells all the other nodes towards the right that it wants to communicate to this node on the left. It defers to downstream requests, counting such requests as they pass by; nodes are not greedy. So this is the heart of the protocol. Actually what it does is that if you simply get an empty cell and want to put in your data, then those nodes, which are towards the end, are favored. I mean it is not a fair system any longer. But then, in this scheme, if you think about it, you will queue the request on a first come first serve manner and that is why you put in your request to the other side so that those nodes on the other side would know that he is going to send something.

(Refer slid time: 55:00)



So to transmit to a destination, the node has to know which bus to use, etc., and the nodes are not greedy.

(Refer slide time: 55:07)



Its primary importance today is its close affinity to ATM, and the consequential association with SONET and SDH, for which we shall see ATMs provide the approved a switching fabric. Thank you. In the next lecture, we are going to discuss two more token-based protocols, namely, token ring, which was more common than this token bus or DQDB and FDDI. Thank you.

(Refer slide time: 55:37)

Preview of the next lecture

Lecture Name # 14

Token Based Mac

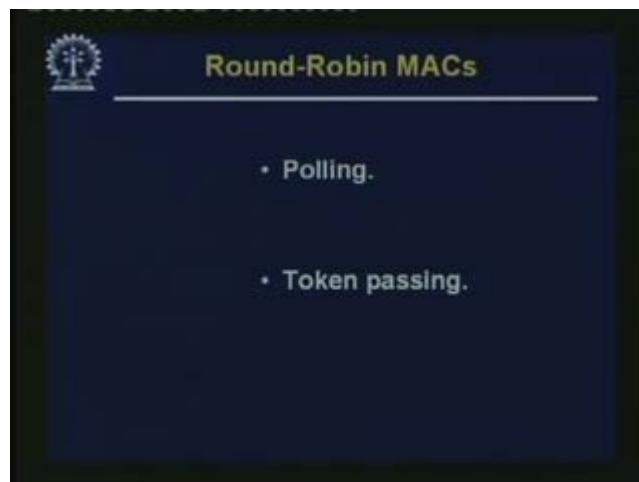
good day in the last lecture we talked about various multiple access schemes and a one of this set of schemes in token bus and DQDV etc where using tokens ok now we will use the we will see two other variance of it namely token ring and a

(Refer slide time: 56:05)



FDDI so we are going to talk about token based MAC and specifically

(Refer slide time: 56:11)



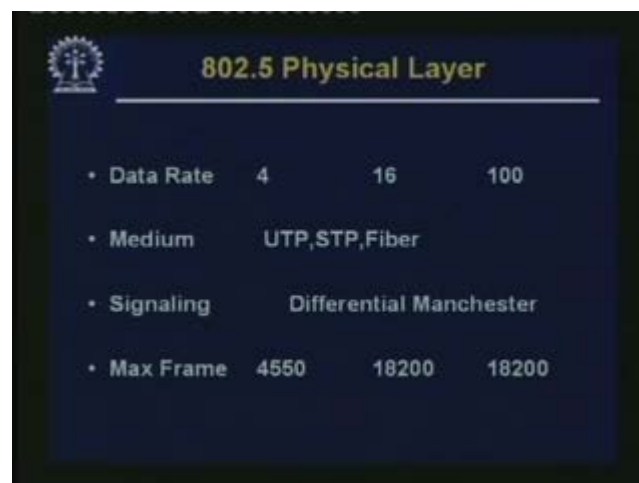
so they are some kind of Round Robin MACs that means a the chance to transmit comes to each of the station in a Round Robin fashion and this can be done as I mentioned earlier through poling or token busing here we will be specifically talking about token busing

(Refer slide time: 56:30)



so the first a system that will talk about is the token ring so as the name itself suggests that it is a ring topology it is a ring on a token ring MAC works with a special pattern or token which is three bytes long so it is a three bytes words of bits called token which moves from one computer to the next priority indicators are used within the token how the priority indicators are used we will see later so

(Refer slide time: 57:04)



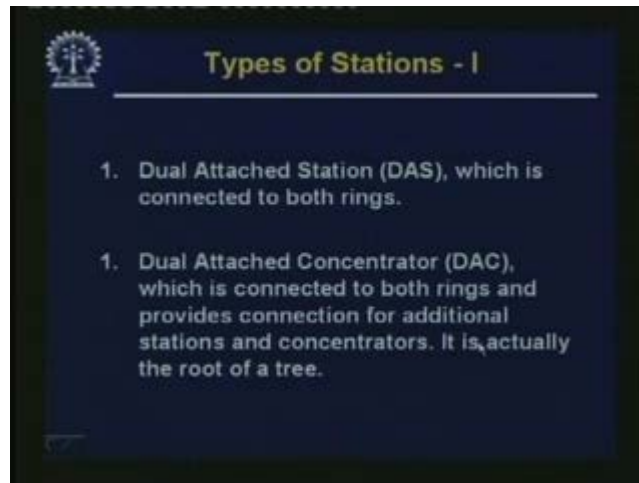
The slide is titled "802.5 Physical Layer" and contains a table with four rows of specifications. The slide has a dark blue background with a logo in the top left corner.

• Data Rate	4	16	100
• Medium	UTP,STP,Fiber		
• Signaling	Differential Manchester		
• Max Frame	4550	18200	18200

so data rate may be four sixteen or hundred mbps medium may be UTP STP or fiber signaling may is usually differential Manchester we mention this earlier what is differential Manchester that how you represent here zeros and ones by electrical signals or optical signals as the case

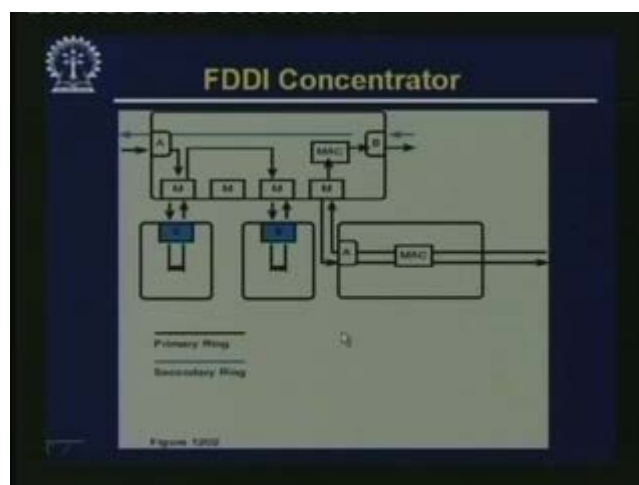
may be and the maximum frame size would be about this four thousand five hundred and fifty bytes or write up to eighteen point two kilo hertz

(Refer slide time: 57:38)



now let us talk about the type of network stations which may be a connected to an FDDI ring one is a dual attached station which is connected to both the rings that means it is a station which is connected to both the rings that is why it is called dual attached then we have dual attached concentrator DAC which is connected to both rings and provides connection for additional stations and concentrators it is actually the root of a tree this is where the tree comes from I have a picture.

(Refer slide time: 58:12)



So we have a so this is the picture of an FDDI concentrator so you can see that this the concern this is the main part of the concentrator and the two rings are there the counter rotating this is the primary ring and this is the secondary ring so the primary ring is coming like this from A to B and the secondary ring is going like this it has some additional ports from which other stations may hang.