UNIT 4 MULTIPLEXING AND SWITCHING

Structure **Page Nos.**

4.0 INTRODUCTION

The transmission of data over a network, whether wireless or not, requires us to solve the problem of being able to send a large amount of data among different groups of recipients at the same time without the message of one being mixed up with those of the others. For example, think of the thousands of radio or television channels that are being broadcast throughout the world simultaneously, or of the millions of telephone conversations that are taking place every minute of the day. This feat is achieved by *multiplexing* the transmissions in some way. In this unit, we will look at the different methods of multiplexing that are commonly used, frequency division multiplexing and time division multiplexing. Time division multiplexing again can be synchronous or statistical.

There are some differences in the characteristics of voice and data communication. Although, finally everything is really data, these differences dictate the design of the telephone network as well as the techniques of switching. When one wants to transmit data over a telephone line, we have to make some changes to be able to get sufficiently high speeds. We will look at ADSL, a scheme for doing this at rates that can compete on cost and quality with those offered by cable television service providers. We will also look at the advantages and disadvantages of the two methods and of high-speed data access.

When constructing networks of any size, a fundamental requirement is that any two nodes of the network should be able to communicate with each other, for, if they cannot do so, they are not on the same network. This brings up the issue of how to switch the data stream, that is, to make sure it reaches its destination and not some other random location on the network. This unit will, describe the different switching techniques available to make this possible.

4.1 OBJECTIVES

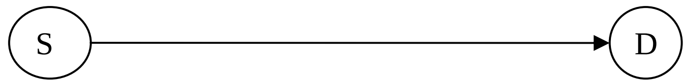
After going through this unit, you should be able to:

- state what multiplexing and switching mean;
- state how a telephone line can be used for transmitting data using ADSL;
- describe the differences between ADSL and cable television for data access;

- describe the different kinds of multiplexing and switching;
- state the characteristics of Frequency Division Multiplexing;
- describe the features of Time Division Multiplexing, both synchronous and statistical, and
- differentiate between the different kinds of switching.

4.2 MULTIPLEXING

If, one wanted to send data between a single source and a single destination, things would be comparatively easy. All it would need is a single channel between the two nodes, of a capacity sufficient to handle the rate of transmission based on Nyqist's theorem and other practical considerations. Granted that there would be no other claimants for the resources, there would be no need for sharing and no contention.



The transmission channel would be available to the sole users in the world at all times, and any frequency which they chose could be theirs. In a broadcasting case, we could have the single source transmitting at any time of its choosing and at any agreed upon frequency.

However, things are not so straightforward in the real world. We have a large number of nodes, each of which may be transmitting or receiving, from or to possibly different nodes each time. These transmissions could be happening at the same moment. So, when we need to transmit data on a large scale, we run into the problem of how to do this simultaneously, because, there can be many different sources and the intended recipients for each source are often different. The transmission resource, that is the available frequencies, is scarce compared to the large number of users. So, there will have to share the resource and consequently, the issue of preventing interference between them will arise.

If, all possible pairs of nodes could be completely connected by channels(?!), we would still not have a problem. But that is obviously out of the question, given the very large number of possible source and destination nodes. There would be over a billion telephones in the world today, for instance.

This problem is solved by performing what is called *multiplexing*. It can be done by sharing the available frequency band or by dividing up the time between the different transmissions. The former is called Frequency Division Multiplexing (FDM) while the other scheme is Time Division Multiplexing (TDM).

Another reason for multiplexing is that it is more economical to transmit data at a higher rate as the relative cost of the equipment is lower. But at the same time, most applications do not require the high data rates that are technologically possible. This is an added inducement to multiplex the data so that, the effective cost can be brought down further by sharing across many different, independent transmissions.

In this context, a multiplexer is a device that can accept *n* different inputs and send out 1 single output. This output can be transmitted over a link or medium to its destination, where to be useful, the original inputs have to be recovered. This is done by a demultiplexer. We need to realise that for this kind of scheme to work, the creation of the composite signal by the multiplexer needs to be such that, the original component signals can be separated at the receiving end – otherwise we would end up with just a lot of noise!

When we transmit data over a cable, we are really setting up a different medium. We can transmit data over different cables at the same frequency at the same time without interference because the media or links are different. Even in radio transmission, where the medium is space and hence is a single medium available to all transmissions, geographical distance can in many cases give rise to different links. A low power medium wave transmission happening in India can be done simultaneously with a similar transmission in Europe without interference or the need for any special precautions. However, throughout the rest of the unit, when we talk of multiplexing, we are referring to the simultaneous transmission of data over the same medium or link.

4.2.1 Frequency Division Multiplexing

Suppose, it is human voice that has to be transmitted, over a telephone. This has frequencies that are mostly within the range of 300 Hz to 3400 Hz. We can modulate this on a bearer or carrier channel, such as one at 300 kHz. Another transmission that has to be made can be modulated to a different frequency, such as, 304 kHz, and yet another transmission could be made simultaneously at 308 kHz. We are thus, dividing up the channel from 300 kHz up to 312 kHz into different frequencies for sending data. This is Frequency Division Multiplexing (FDM) because all the different transmissions are happening at the same time – it is only the frequencies that are divided up.

The composite signal to be transmitted over the medium of our choice is obtained by summing up the different signals to be multiplexed (*Figure1*). The transmission is received at the other end, the destination and there, it has to be separated into its original components, by demultiplexing. In practice, a scheme like this could result in interference or cross talk between adjacent channels because the bandpass filters that are used to constrain the original data between the agreed upon frequencies (300 to 3400 kHz) are not sharp. To minimise this, there are guard bands, or unused portions of the spectrum between every two channels.

Another possible cause of interference could arise because of the fact that the equipment, such as amplifiers used to increase the strength of the signal, may not behave linearly over the entire set of frequencies that we seek to transmit. Then, the output can contain frequencies that are the sum or difference of the frequencies used by the input. This produces what is called intermodulation noise.

In this kind of division, it should be realised that the actual modulation technique used is not of consequence. So, one could use analog modulation (AM) or Frequency Modulation (FM). Also the composite signal that we have produced could again be modulated over a different frequency altogether. For example, the three voice channels, that have been modulated for commercial broadcast radio to produce a spectrum from 300 kHz to 312 kHz could be modulated onto a 2 GHz satellite channel for long distance transmission to the other side of the earth. This second modulation could use a technique different from the first one. The only thing that needs to be take care of is, that the recovery of the original signals have to be done in the reverse order, complementing the method used for producing the composite signal.

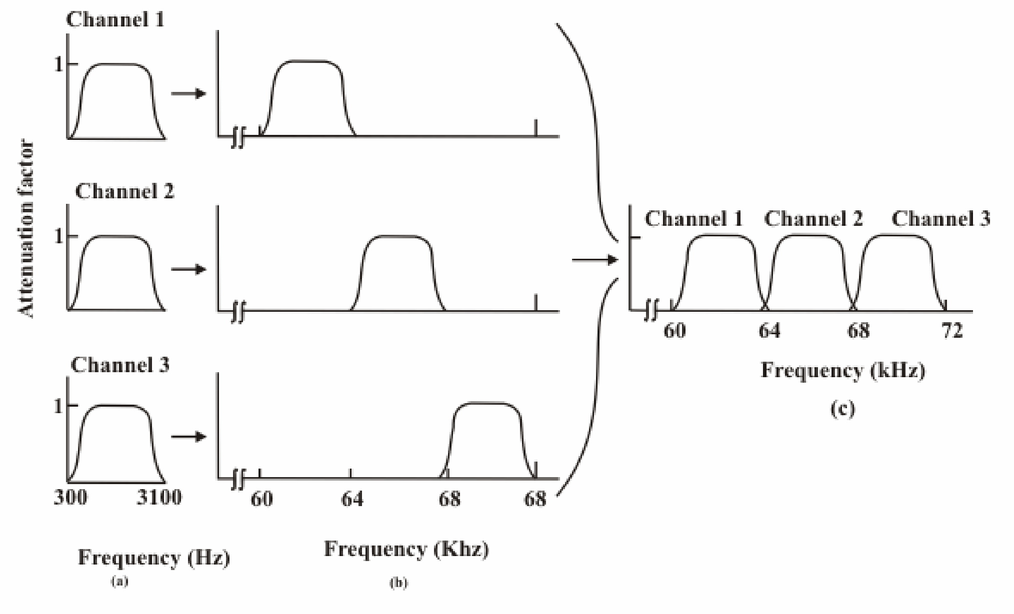


Figure 1: Frequency division multiplexing

[Source: Computer Network by A.S. Tanenbaum]

The International Telecommunication Union (ITU) has standardised a hierarchy of schemes that utilise FDM for the transmission of voice and video signals. To begin with, a cluster of 12 voice channels, each of 4 kHz is combined to produce a signal of bandwidth 48 kHz that is then, modulated and transmitted over the 60 to 108 kHz band. This is called a group.

The next level of the hierarchy is the supergroup, where 5 such groups are combined to use 240 kHz of bandwidth, occupying from 312 to 552 kHz. The next level is the mastergroup that consists of 5 such supergroups. This uses 1232 kHz from 812 to 2044 kHz. Again, 3 such supergroups form a supermaster group that contains 3.872 MHz from 8.516 to 12.388 MHz. The United States uses a similar hierarchy as decided by AT&T (the telecommunications company) that is not entirely identical to the ITU standard.

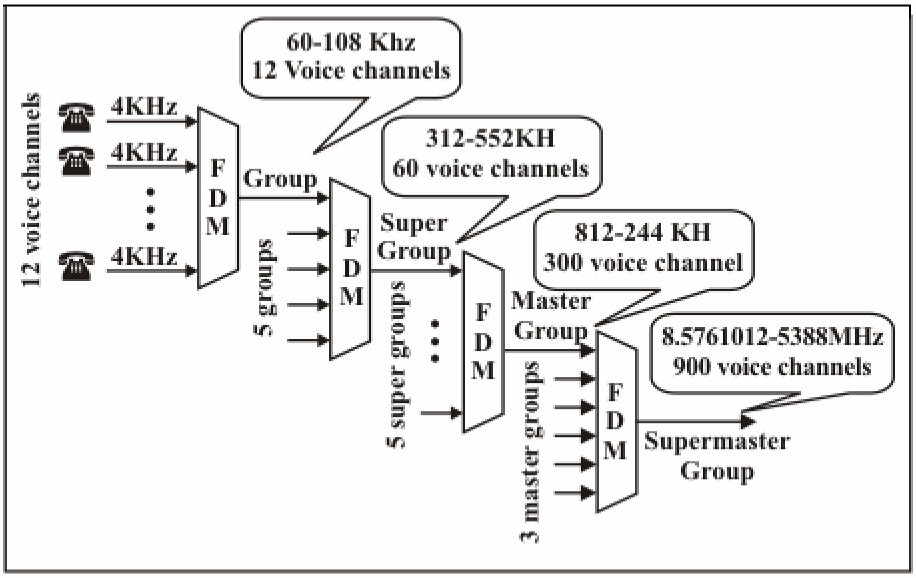


Figure 2: Analog hierarchy

We, thus, see that the original signal could find itself subjected to modulation several times, each of which may be of different kinds, before it is finally transmitted. This raises the possibility of interference and noise, but, with the very good equipment available today, this need not be a matter of concern.

FDM has the disadvantage of not being entirely efficient if the transmissions that are multiplexed together have periods of silence or no data. Since, a frequency band is dedicated to each data source, any such periods are simply not utilised.

4.2.2 Time Division Multiplexing

Another multiplexing scheme is to use the entire bandwidth for each channel but to divide it into different time slots, as shown below.

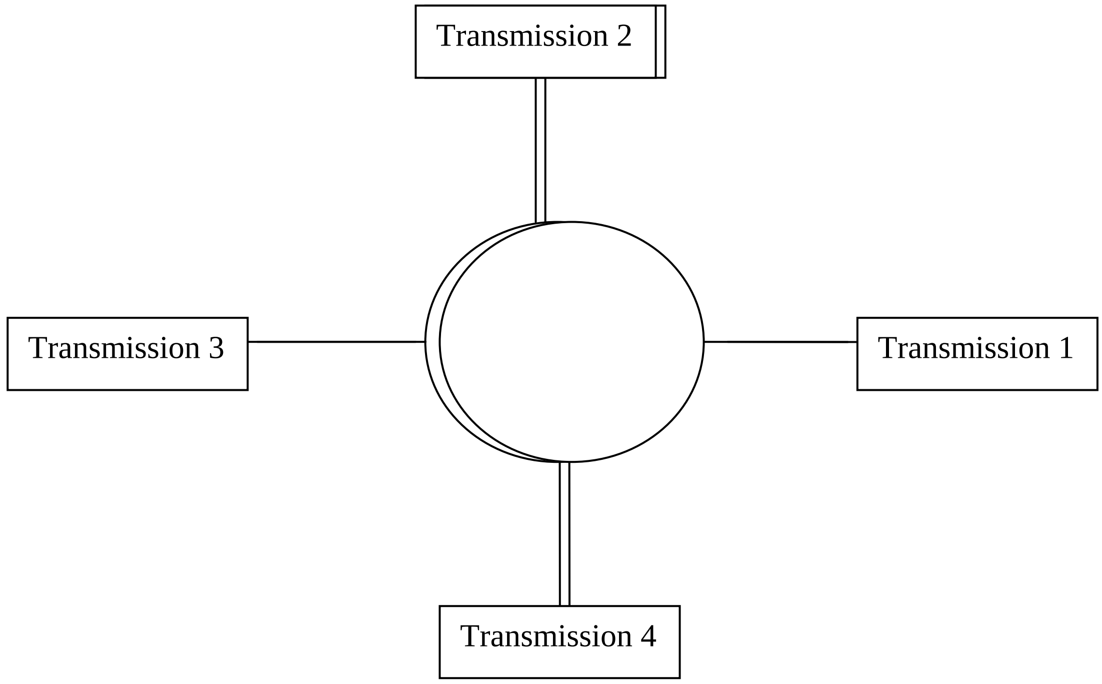


Figure 3: Time division multiplexing

In this case, we have four different transmissions occurring with each being ¼ of the time slice. The slice should be small enough so that the slicing is not apparent to the application. So, for a voice transmission a cycle of 100ms could be sufficient as we would not be able to detect the fact that there are delays. In that case, each transmission could be allotted a slice of 25ms.

At the receiving end, the transmission has to be reconstructed by dividing up the cycle into the different slices, taking into account the transmission delays. This synchronisation is essential, for if the transmission delay is, say 40ms, then the first transmission would start reaching 40 ms later, and would extend to 65ms. If, we interpreted it to be from 0 to 25 ms, there would be complete loss of the original transmission. This interleaving of the signal could be at any level starting from that of a bit or a byte or bigger.

Synchronous Time Division Multiplexing

In this kind of Time Division Multiplexing (TDM), the simpler situation is where the time slots are reserved for each transmission, irrespective of whether it has any data to transmit or not. Therefore, this method can be inefficient because many time slots may have only silence. But it is a simpler method to implement because the act of multiplexing and demultiplexing is easier. This kind of TDM is known as Synchronous TDM.
Here, we usually transmit digital signals, although the actual transmission may be digital or analog. In the latter case, the composite signal has to be converted into analog data by passing it through a modem. Here, the data rate that the link can support has to at least equal the sum of the data rates required for each transmission.

So, if one has to transmit four streams of data at 2400 bps each, we would need a link that can support at least 9600 bps. For each data stream, we would typically have a small, say 1 character buffer, that would take care of any data flow issues from the source.

Although, the transmission has to be synchronous, that is not the reason the scheme is called Synchronous TDM. It is because of the fact that each data source is given fixed slots or time slices. We can also support data sources with differing data rates. This could be done by assigning fewer slots to slower sources and more slots to the faster ones. Here, a complete cycle of time slots is called a frame. At the receiving end, the frame is decomposed into the constituent data streams by sending the data in each time slot to the appropriate buffer. The sequence of time slots allotted to a single data source makes up a transmission channel.

We have already seen that the transmission in TDM must be synchronous because otherwise, all the data would be garbled and lost. How is this achieved? Because this unit is concerned with the physical layer and not the data link layer, we will not concern ourselves with the problem of flow control or error correction or control. But even at the physical layer, we have to ensure that each frame is synchronised to maintain the integrity of the transmission.

One scheme to attain this is to add an extra bit of data to each frame. In succeeding frames, this extra bit forms a fixed pattern that is highly unlikely to occur naturally in the data. The receiver can then use this fact to synchronise the frames. Suppose each frame has n bits. Starting from anywhere, the receiver compares bit number 1, n+1, 2n+1 and so on. If the fixed data pattern is found, the frames are synchronised. Otherwise, the receiver commences to check bit number 2, n+2, 2n+2, … until it can determine whether the frames are in synchronization or otherwise. This continues till, it is able to synchronise the frames and can then start receiving the data from the various channels.

Even after this, it is important to continue to monitor the fixed pattern to make sure that the synchronisation remains intact. If it is lost, the routine of re-establishing it needs to be repeated as before.

The second problem in synchronising the transmission is the fact that there can be some variations between the different clock pulses from the different input streams. To take care of this, we can use the technique of pulse stuffing. Here, the multiplexed signal is of a rate that is a bit higher than that of the sum of the individual inputs. The input data is stuffed with extra pulses as appropriate by the multiplexer at fixed locations in the frame. This is needed so that the demultiplexer at the other, receiving, end can identify and remove these extra pulses. All the input data streams are thus synchronised with the single local, multiplexer clock.

Statistical Time Division Multiplexing

We have seen that synchronous TDM can be quite wasteful. For example, in a voice transmission, much of the bandwidth can be wasted because there are typically many periods of silence in a human conversation. Another example, could be, that of a time sharing system where many terminals are connected to a computer through the network. Here too, many of the time slots will not be used because there is no activity at the terminal. In spite of being comparatively simpler, synchronous TDM is therefore, often not an attractive option.

To take care of this problem, we can use statistical TDM. This is a method where there are more devices than the number of time slots available. Each input data stream has a buffer associated with it. The multiplexer looks at the buffer of each device that provides the input data and checks to see if it has enough to fill a frame. If, there are enough, the multiplexer sends the frame. Otherwise, it goes to the next device in line and checks its input buffer. This cycle is continued indefinitely. At the receiving end the demultiplexer decomposes the signal into the different data streams and sends them to the corresponding output line.

The multiplexer data rate is not higher than that of the sum of all the input devices connected to it. So statistical, TDM can make do with a lower transmission rate than needed by synchronous TDM, at the cost of more complexity. The other way in which this is of benefit is, by the ability to support higher throughput for the same available data rate of the multiplexer. This capability is easily realised during the normal, expected data transmission periods when the amount of data that the different input devices have available for transmission is, in fact, lower than the capacity of the multiplexing device. But the same attribute becomes a disadvantage during peak loads, where all or many of the devices may have data to transmit for a short time. In such a situation, the slack that was available to us for more efficient transmission is no longer present. We will see later how to handle peak load situations in statistical TDM.

In this kind of scheme it is not known to us which input device will have data to send at a given time. So we cannot use a round robin positional scheme that was possible in synchronous TDM. Each frame has to be accompanied by information that will tell the demultiplexer which input device the frame belongs to, so that it can be delivered to the appropriate destination. This is the overhead of statistical TDM, besides increased complexity of equipment.

If, we are sending input data from only one source at a time, the structure of a frame would need to have an address field for the source followed by the data for it. This is really the statistical TDM data frame. It would form the data part of a larger, enclosing HDLC frame if we are using HDLC as the transmission protocol. This frame would itself have various other fields that we will not discuss here.

Such a scheme is also not as efficient as we can make it. This is because the quantum of data available from the source, in that time slot, may not be enough to fill the TDM sub frame. So, while it may be an adequate method if the load is not heavy, we also need to think of a method that can utilise available resources better.

The way to do this would be, to have more than one input data source transmit in a single TDM frame. We would then need to have a more complex structure for the frame whereby, we would have to specify the different input devices in the frame followed by the length of the data field. More sophisticated approaches could be used, in order to optimise the number of bits, we need to encode all this addressing and data information.

For peak loads, there is need for some kind of buffering mechanism, so that, whenever there is excess input from the data sources that the multiplexer cannot immediately handle, it is stored until it can be sent. The size of the buffer needed will increase as the transmission capacity of the multiplexer decreases, as it will become more likely that an input data stream will not be transmitted immediately. It will also depend on the average data rate of the input devices taken together. No matter what the buffer size we choose, there is always a non-zero probability that the buffer itself will overflow, leading to loss of data. If, the average data rate of all devices is close to the peak rate, then we are approaching a situation of synchronous TDM where we are not able to take advantage of periods of silence that is the basis of statistical TDM.

4.3 DIGITAL SUBSCRIBER LINES

Digital Subscriber Lines are an example of multiplexing in action. They are really telephone lines that can support much higher data rates than are possible with an ordinary telephone connection. The concept came up in response to the high data rates of more than 10 Mbps offered by cable television providers. Compared to those rates, the 56 Kbps possible over a telephone dial up connection was a miserable offering.

The 56 Kbps rate is really an artificial barrier arising from the fact that voice telephone line standards limited the bandwidth to about 3.1 KHz to take advantage of the normal range of the human voice in conversation. Nyquist's theorem and the limitations of line quality ensured that we ended up with that rate. Once the 3.1 KHz limit is removed, the telephone line can support much higher data speeds.

Out of the several different technical solutions that came about, the ADSL (Asymmetric DSL) technology has proved to be the most popular. Asymmetric comes from the fact that the data rates in the two directions are different. The approach is to divide the 1.1 MHz bandwidth available over the Cat-3 telephone cables into 256 channels. Of these channels, 0 is used for normal voice communication. The next 5 channels are not used to ensure separation and non-interference between data and voice transmissions. The next 2 channels are used for upstream and downstream control. The remaining 248 channels are available for data transmission.

Like in voice circuits, it would have been possible to use half the channels for communication in each direction. But, statistics show that most users download much more data than they upload. So usually 32 channels are dedicated to uploading, that is, transferring data from the users to the provider and the remaining 216 channels are used for downloading data to the users. This typically translates into 512 Kbps to 1 Mbps download and 64 Kbps to 256 Kbps upload data rates. This then, is the asymmetric aspect of the DSL line.

A problem with ADSL is that the physics of the local loop is such that, the speed at which it can be driven depends heavily on the distance between the subscriber's premises and the provider's nearest termination point. The speed falls sharply with distance and so, distance can become a limiting factor in being able to offer competitive speed compared to that of the cable television providers.

However, ADSL is comparatively simple for a service provider to offer, given an existing telephone network, and does not require much change to its already available equipment. It necessitates two modifications, one each, at the subscriber end and at the end office. On the user's premises, a Network Interface Device has to be installed that incorporates a filter. This is called a splitter and it sends the non-voice portion of the signal to an ADSL modem. The signal from the computer has to be sent to the ADSL modem at high speed, usually done these days by connecting them over a USB port.

At the provider's end office, the signal from the users is recovered and converted into packets that are then sent to the Internet Service Provider, which may be the telephone company itself.

4.4 ADSL Vs. CABLE

At first glance, a comparison between ADSL and cable may seem like a no contest. Cable television, sent over coaxial cables, has a bandwidth that is potentially hundreds of times that of the twisted pair Cat-3 cable used for telephone connections. But, as we go along, it turns out that there are considerations in favour of both sides.

First, there are specific assurances regarding bandwidth that we get from telephone companies who provide ADSL connectivity. An ADSL link is a dedicated connection that is always available to the user, unlike television cable that is shared by scores or even hundreds of subscribers in the immediate neighbourhood. So the kind of speeds that we can get over cable can vary from one moment to the next, depending on the number of users that are working at the time.

There are security risks associated with the fact that cable is shared. Potentially other users can always tap in and read (even change) what you are sending or receiving. The problem does not exist on ADSL, because, each channel is separate and dedicated to the specific user. Though, cable traffic is usually encrypted by the provider, this situation is worse than ADSL where other users just do not get your traffic at all. Also because the channel is dedicated to you, the total number of users does not have any effect on your access speeds as there is no contention with other users.

4.5 SWITCHING

There are many potential sources of data in the world and likewise, many potential recipients. Just think of the number of people who may like to reach one another over the telephone. How can one ensure that every data source is able to connect to the recipient? Clearly one cannot have a physical link between every pair of devices that might want to communicate! Therefore, we need a mechanism, to be able to connect together devices that, need to transfer data between them. This is the problem of switching.

There are two basic approaches to switching. In circuit switching, we create a circuit or link between devices, for the duration of time for which they wish to communicate. This requires a mechanism for, the initiating device to choose the device that it wants to send data to. All devices must also have an identifying, unique address that can be used to set up the circuit. Then, the transmission occurs, and when it is over, the circuit is dismantled so that the same physical resources can be used for another transmission.

The packet switching approach can be used when we are using datagrams or packets to transmit data over the channel. Each datagram is a self-contained unit that includes within itself the needed addressing information. The datagrams can arrive at the destination by any route and may not come in sequence.

4.5.1 Circuit Switching

As mentioned already, in circuit switching, we create a link or circuit between the devices that need to communicate, for the duration of the transmission only. It entails setting up the circuit, doing the actual transmission that can be simplex, half duplex or full duplex, and then dismantling the circuit for use by another pair of devices. An example is shown in the *Figure 4* where there are 7 devices. These are divided into two groups of 3 on the left and 4 on the right. To ensure complete connectivity between them at all times, we would need 12 physical links. But, if connectivity is not required at all times, we can achieve connectivity between any of the devices by grouping them together and using switches to achieve temporary links.

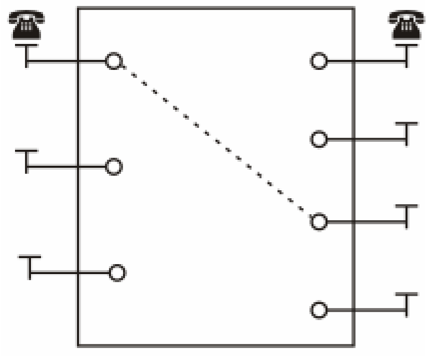


Figure 4: Circuit switching

For example, we can have 7 links that connect the devices A, B and C on the left to the switch. The other devices D, E, F and G on the right are also connected to the same switch. The switch can connect any two devices together using only these seven links as desired.

The capacity of the switch is determined by the number of circuits that it can support at any given time. In the above example, we have seen a switch with 1 input and 1 output. If devices C and E are communicating with each other, the others cannot communicate at the same time, although, there are available links from them to the switch. A circuit switch is really a device that has n inputs and m outputs (n need not be equal to m).

There are two main approaches to circuit switching, called space division or time division switches. Space division switches are so called because the possible circuit paths are separated from one another in space. The old, and now obsolete, crossbar telephone exchanges are an example of space division switching. The technique can be used for both digital or analog systems. There were other designs of such switching but the only one that went into large scale use was the crossbar. Because of the way it is constructed, such switching does not have any delays. Once the path is established, transmission occurs continuously at the rate that the channel can support.

In essence, a crossbar connects p inputs to q outputs. Each such connection is actually performed by connecting the input to the output using some switching technology such as, a transistor based electronic switch or an electromechanical relay. This requires a large number of possible connections, called crosspoints. For a 10,000 line telephone exchange, it would mean that, each of the 10,000 possible inputs be able to connect to any of the 10,000 possible outputs, requiring a total of 100,000,000 crosspoints. As this is clearly impractical, the pure crossbar design is not usable on a commercial scale. Moreover, there are inherent inefficiencies in this design as statistically, only a small fraction of these crosspoints are ever in use simultaneously.

The way to get around this limitation is, to split the switch into different stages. If we consider a 36 line exchange where each of 36 inputs needs to be connected to 36 outputs, we can do so in, say, 3 stages. The first stage could have 3 switches, each with 12 inputs and 2 outputs to the two second stage switches. These intermediate switches could each have 3 inputs from the 3 first stage switches and 3 outputs to the 3 third stage switches. The last stage of the switches would then have 2 inputs from the 2 second stage switches and 12 outputs, each to 12 of the 36 devices. It is thus, possible for each of the 36 inputs to connect to each of the 36 outputs as required, using only 72 + 18 + 72 = 162 crosspoints, instead of the 1296 crosspoints that would have been required without the multistage design. The following *Figure 5* shows the multistage switch.

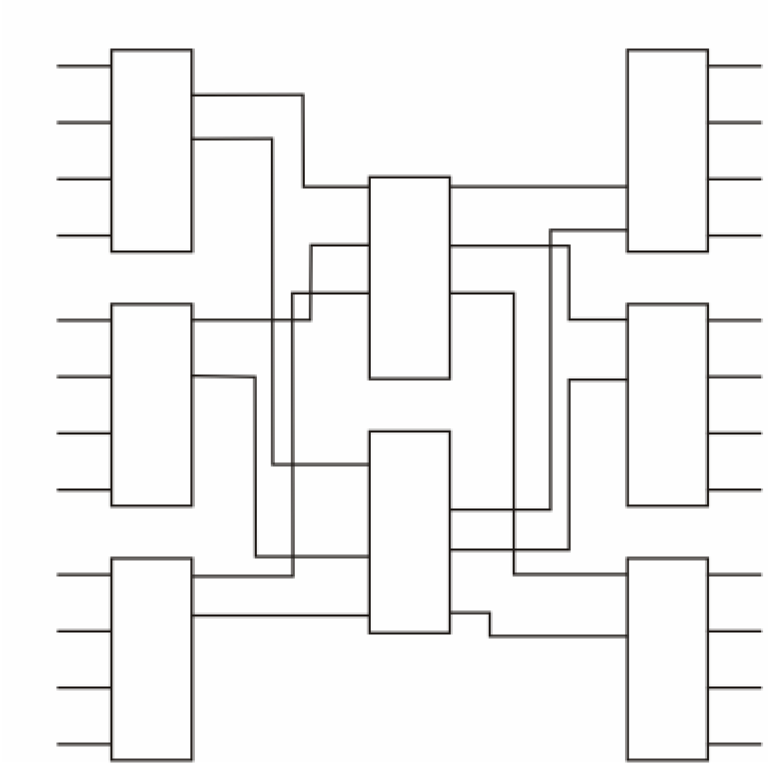


Figure 5: Multistage switch

If, we use a single stage switch, we will always be able to connect each device to any other device that is not already busy. This is because all the paths are independent and do not overlap. What this means is that, we will never be starved of circuits and will never suffer the problem of blocking. For multistage switching, the reduction in the number of crosspoints required comes at the cost of possible blocking. This happens when there is no available path from a device, to another free device, that we desire to connect to. In the *Figure 5*, if we have a switch in the first stage that is already serving 2 input devices, then there are no free outputs from that stage. We cannot therefore service more than 2 input devices connected to one switch at a time and the 3rd device would not be able to connect anywhere, getting a busy signal.

It is possible to minimise the possibility of blocking by using stochastic analysis to find the least number of independent paths that we should provide for. However, we will not go into such an analysis in this unit. But, we can see that the limiting factor in the *Figure 5* is the middle stage where there are only 3 inputs and 2 outputs. This is the point at which congestion is most likely to occur. You too may have experienced blocking when trying to make a telephone call and found that you got an exchange busy tone, indicating that it was not the number you called but the exchange (switch) itself that was congested. Such a problem is, most likely to occur during periods of heavy activity such as at midnight on a new year’s day, when many people might be trying to call one another to exchange greetings.

Another advantage of multistage switching is that, there are many paths that can be used to connect the input and output devices. So, in the above case, if there is a failure in one of the connections in the first stage switch, two input devices can still connect to it. In the case of single stage switching, that would have meant that we could not set up a circuit between those devices.

Let us, now look at another method of switching that uses time slots rather than spatial separation. You have already seen how synchronous TDM involves transmission between input and output devices using fixed time slots dedicated to each channel. But, that is not switching because the input-output device combinations are fixed. So, if we have three devices A, B and C transmitting and three devices D, E and F that are receiving the respective transmissions, there will be no way to change the circuit path so that A can transmit to E or F.

To achieve switching, we use a device called a Time Slot Interchange (TSI). The principle of such a device is simple. Based on the desired paths that we want to set up, the TSI changes the input ordering in the data streams. So, if the demultiplexer is sending the outputs to D, E and F in order, and if we want that A send data to E instead of F, and that B send data to F rather than E, then the TSI will change the input ordering of the time slots from A, B, C to C, A, B. The demultiplexer will not be aware of this and will continue to send the output the same way as before. The result will be, that our desired switching will be accomplished. However, unlike space division switching, our output will be subject to delays because we might have to wait for the input from the right device before it can be transmitted. This is unlike space division switching where the data can be sent in a steady stream.

How does the TSI work? It consists of a control unit that does the actual reordering of the input. For this, it has to first, buffer the input it gets, in the order it gets it. This would be stored in some kind of volatile memory. The control unit then sends out the data from the buffer in the order in which it is desired. Usually the size of each buffer would be that of the data that the input generates in one time slice.

We do not have to confine ourselves to a single type of switch. There can be switches that are based on a combination of both kinds of switching. For example, we could have a multistage switch where, some of the stages are space division switches whiles others are time division switches. With such an approach, we can try to optimise the design by reducing the need for crosspoints while keeping the delays in the whole system to a minimum. For example, we could have a TST three stage switch where the first and last stages use time division switching while the middle stage uses crosspoints.

Circuit switching is useful for voice based communication because of the characteristics of the data transfer. Although, a voice conversation tends to have periods of silence in between, those periods are usually brief. The rest of the time there is data available for transmission. Secondly, in such communication, we cannot tolerate delays of more than about 100 ms as that becomes perceptible to the human ear and is quite annoying to the speaker and listener.

Again, because it is human beings that are present at both ends, the rate at which data is generated at both ends is similar, even if one person talks a bit faster than the other! Also, the other human can usually understand what is being said even if he cannot at the same rate. And if really required, one can communicate to the other to speak slower or louder. But, when we are dealing with data generating devices, there can always be a mismatch between the rate at which one device generates data and at which the other device can assimilate it. Moreover, there can be long periods when there is no data generated at all for transmission. In such a situation, circuit switching will not be a suitable method and we have to look at something that takes care of the characteristics of data communication between devices.

4.5.2 Packet Switching

Packet switching *Figure 6* is a method of addressing these problems of circuit switching. It will be further elaborated in the Block 3. It is based on the concept of data packets, called datagrams. These are self contained units that include the address of the destination, the actual data and other control information. It takes care of the sporadic nature of data communication where transmissions tend to occur in bursts. So, we do not waste transmission capacity by keeping circuits connected but, idle while there are periods of silence. Even if we multiplex the channel, we cannot cater to a situation where all or most of the devices are silent, leading to underutilisation of the capacity. The concept is so interesting that it is worth devoting few words to it. Whenever a user wants to sends a packet to another user, s/he transmit to the nearest router either on its own LAN or over a point-to-point link to the caviar. The packet is stored for verification and then transmitted further to the next router along in way until it reaches for fixed destination machine. This mechanism is called packet switching.

Some of the other limitations of circuit switching for sending data are:

- It is not possible to prioritise a transmission. In a circuit switching mechanism, all the data will be sent in the order in which it is generated, irrespective of its importance or urgency. This is fine for voice transmission, in which we want to hear things in the sequence in which they are spoken at the other end, but this does not work for data where, some packets may be more important than others and the order of delivery does not have to be sequential.

- A circuit, once set up, defines the route that will be taken by the data until it is dismantled and set up again. Sometimes, that circuit may have been set up via a less advantageous set of links because that was the best route available at the time it was set up (best could be in terms of channel capacity, delays, line quality or other parameters). Now, subsequently, even if another, better route is released by other devices, we cannot change over to this better route without disconnecting the previous circuit and forcing the participants to set up the call again.

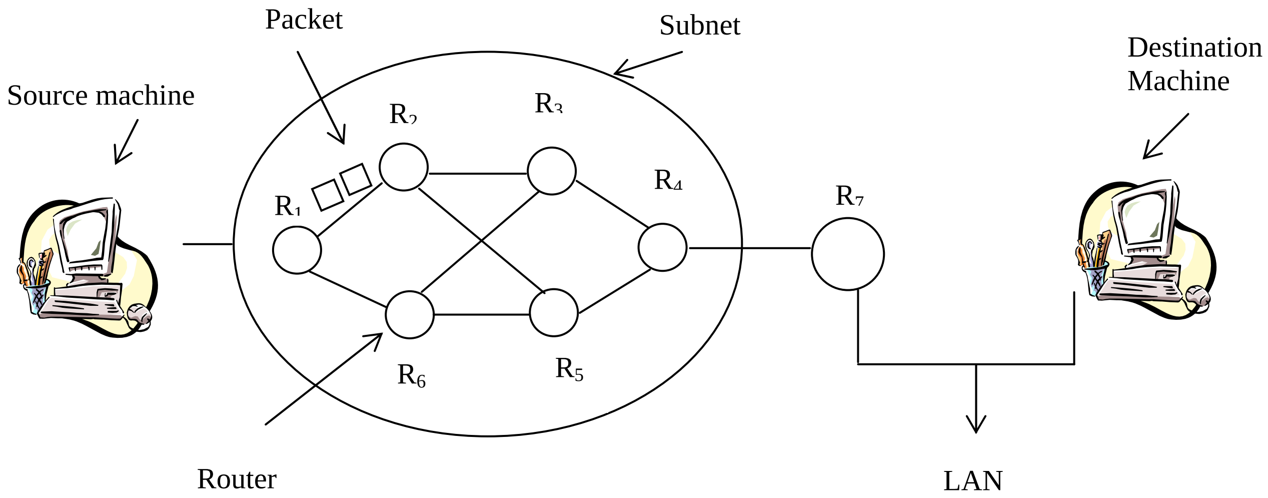


Figure 6: Packet switching

A datagram can contain different lengths of data. Besides, it would have control information such as the address to which it needs to be sent, the address from which it has originated, its type and priority and so on. From its source node it is sent to a neighbouring node, depending on its final destination. The format of the datagram depends on the protocol that is used and it is the responsibility of each node to route the datagram onto the next node that would take it to its destination. There can be several routes possible and each node would make the decision based on availability, traffic conditions, channel quality and so on.

Although datagram length can vary, long messages would have to be split into more than one datagram as the network protocol would have a limit on the maximum size of a datagram. Because each datagram is routed independently, the different datagrams that make up a message may arrive out of sequence at the destination, as different routes may take different times to traverse. In communication protocols it is the responsibility of the network layer to make sure that routing gets done, while it is the transport layer that ensures that the message at the receiving end is in the proper sequence.
Again, the physical link between two nodes can be of any type and is of no consequence at the higher layers of the communication protocol in use. That link itself may be multiplexed and may carry several transmissions simultaneously, for the

same or different pairs of source and destination nodes. Moreover, these transmissions may be happening in different directions.

This was the datagram approach to packet switching. It is also possible to look at packet switching in terms of a virtual circuit, which again has two different approaches that are in current practice.

In a virtual circuit, when the first datagram is sent out, we decide on the route that will be followed, and subsequent datagrams continue to follow that route. So, it is like circuit switching to a large extent. But, in circuit switching the link is dedicated to the pair of nodes and there is no multiplexing that happens at the level of individual switches. In virtual circuits there can be multiplexing at the switches as well.

Permanent virtual circuits are dedicated channels set up for communication between a pre-decided pair of nodes. It is thus, akin to a leased line where the routing is decided once the virtual circuit is set up. There is no call set up required because, the link is always available to the pair of nodes. On the other hand, we can also have switched virtual circuits where the routing is decided every time a pair of nodes want to communicate. This is like a dial-up connection where any two pairs of nodes can communicate by setting up a call that is terminated when the conversation is over.

Another difference between circuit and virtual circuit switching has to do with reliability and failure recovery. In circuit switching, if the link breaks because of a failure in any portion, the call is terminated. It would have to be established again for the conversation to continue. In virtual circuits, although the route is decided at the beginning when the session is set up, in case of failure of any part of the route, an alternate route will be agreed upon and set up.

Also, in circuit switching there is no possibility of failure due to congestion after the link is set up. In virtual circuits, congestion can occur even later because of multiplexing at the switches.

☞ Check Your Progress 1

1) What is the problem in offering high speed ADSL connection to all subscribers that have a telephone?
……………………………………………………………………………………
2) Which method is more efficient in terms of capacity utilisation?
……………………………………………………………………………………
3) Do you think voice transmission could be done using packet switching?
……………………………………………………………………………………
4) What sort of problems could arise in trying to completely connect all data transmission devices in the world?
……………………………………………………………………………………
5) What are some of the requirements for a multiplexed signal to be useful?
……………………………………………………………………………………
6) Is multiplexing needed for transmissions over different links?
……………………………………………………………………………………
7) What are the problems that can occur in Frequency Division Multiplexing?
……………………………………………………………………………………
8) What is the method that FDM uses to mix signals that can be recovered later?
……………………………………………………………………………………
9) List the considerations that make FDM useful and possible.
……………………………………………………………………………………
10) What are the considerations in choosing the length of the time slice for Time Division Multiplexing?
……………………………………………………………………………………

11) Why must the multiplexer and demultiplexer be synchronised? Why then is synchronous TDM so called?
………………………………………………………………………………………

12) What are the problems in synchronising the transmission and how can they be taken care of?
………………………………………………………………………………………

13) What are the inefficiencies inherent in synchronous Time Division Multiplexing and how does statistical TDM seek to reduce them?
………………………………………………………………………………………

14) What price do we have to pay for increased efficiency in statistical TDM?
………………………………………………………………………………………

15) What would happen if the load from the input devices exceeds the capacity of the multiplexer?
………………………………………………………………………………………

16) How does ADSL enable high speed data access although voice lines are so slow?
………………………………………………………………………………………

17) Why is ADSL called asymmetric and why is it not kept symmetric?
………………………………………………………………………………………

18) How can a cable provider improve quality of service after the number of users becomes larger?
………………………………………………………………………………………

19) Why is ADSL more secure than cable for data communication?
………………………………………………………………………………………

20) Why is switching necessary?
………………………………………………………………………………………

21) What is circuit switching?
………………………………………………………………………………………

22) What is packet switching? How does it differ from circuit switching?
………………………………………………………………………………………

23) What is meant by multistage switching? Is the capacity of such a switch limited?
………………………………………………………………………………………

24) List five important features of space division switching.
………………………………………………………………………………………

25) What are the characteristics of time division switching?
………………………………………………………………………………………

26) How can we combine space and time division switching? What are the advantages of such an approach?
………………………………………………………………………………………

27) Why is circuit switching suitable for voice transmission?
………………………………………………………………………………………

28) What are the features needed for a switching mechanism that transmits data?
………………………………………………………………………………………

29) What is a virtual circuit? How does it differ from circuit switching?
………………………………………………………………………………………

30) What are datagrams and how can they be used to transmit data?
………………………………………………………………………………………

4.6 SUMMARY

Multiplexing is needed in communication networks because of the scarcity of bandwidth compared to the large number of users. Frequency Division and Time Division Multiplexing are the two ways of multiplexing, of which Time Division multiplexing can be synchronous or asynchronous. The asynchronous method is more complex but more efficient for data transmission.

ADSL is a means of utilising the existing capacity of the local loop in the telephone system for providing subscribers with high speed data access. Such access is also provided by companies over the cable television network. ADSL is more secure and predictable in terms of service quality, while cable does not have the limitations of distance from the end office that ADSL has.

Switching is necessary to connect two nodes or devices over the network that intend to communicate for a limited duration. Circuit switching is more suitable for voice communication, and can be done using space division, time division or a combination of both kinds of switches. Multistage switching is needed to optimise the number of crosspoints needed. Packet switching is used for data transmission and allows for prioritising of data packets, alternate routing as needed and is also more efficient for the bursty traffic pattern of data communication. Datagrams are self contained packets of data that are routed by the intermediate nodes of the network. Switched or permanent virtual circuits can also be utilised, where the route is established at the beginning of the session, but can be altered without disrupting the channel in case of failure of any part of the route.

4.7 SOLUTIONS/ANSWERS

Check Your Progress 1

1) First, the subscriber may be located too far from the provider's end office for the local loop to give sufficiently high data speeds to be competitive with cable. Secondly, providing a connection means a visit to the subscriber's premises to install the Network Interface Device, splitter and ADSL modem.

2) Cable is more efficient because bandwidth is not dedicated to a user. In ADSL, a user's dedicated bandwidth remains unutilised if s/he is not active at any given time.

3) Packet switching can be and is used for voice transmission, such as in the highly popular Voice Over Internet Protocol (VOIP) used for Internet Telephony. It is cheaper than voice transmission though there is some deterioration in voice quality.

4) There are a very large number of such transmission devices. For example, there may be over a billion telephones in the world today. The number of data channels to always connect even one phone to another billion phones alone would be one billion. It is therefore, not practically possible to have so many distinct data channels.

5) Multiplexing is the creation of a composite signal from various constituent signals at the transmitting end. There must be a way to separate the different signals at the receiving end, otherwise all we would have at that end would be a lot of noise. Secondly, there must be a way to send each separate signal to the proper device for which it was meant, otherwise we may have a useful signal but delivered to the wrong device.

6) Multiplexing is needed because we want to send several signals over the same link. If there are different physical links available, then no multiplexing is needed because the transmissions would be independent and would not interfere with each other.

7) One problem with FDM is that we have to use filters to constrain the constituent signals between the agreed upon frequencies. For example, in telephone transmission, the voice signal is filtered to lie between 300 and 3400 Hz. Other frequency components in the voice, such as very low or high frequencies, are therefore not transmitted. This filtering is not very sharp and can result in cross talk between adjacent channels.

   Secondly, the amplifiers and other equipment used may not behave linearly over the whole frequency range. This gives rise to other frequency components that were not present in the original signal, giving rise to distortion of the original signal. This is called intermodulation noise.

8) FDM uses different methods to mix signals. We can use analogue modulation (AM) where the carrier amplitude is modulated with the signal, or frequency modulation (FM) where the carrier frequency is modulated with the signal.

9) FDM is useful because we are able to make use of the available spectrum more efficiently. We can use different carrier frequencies to send different signals at the same time.

10) The length of the time slice must be such that the application that is using the signal is not able to perceive the slicing, or it does not matter to the application. For example, in

voice transmission, a delay of 300 ms is quite apparent to a human speaker. So the time slice must be much smaller than this.

11) The multiplexed signal consists of signals from the different data sources, each of which has a fixed time slot. To send the received data to the correct data source, the demultiplexer must be in synchronization with the multiplexer, else the data would go to the wrong destination. That is why synchronous TDM is so called.

12) One problem is that the clocks of the different input signals may not be in synchronization. So we use a multiplexer clock to do the synchronization. The multiplexed signal is sent at a rate slightly higher than the sum of the input signals. At fixed points in the transmitted data frames, we stuff the frame with synchronization pulses that can be identified and removed by the demultiplexer at the receiving end.

    Secondly, the start of the frame from different sources needs to be known. This is done by adding extra bits to the frames that form a fixed pattern unlikely to occur in an actual transmission. This pattern is used by the demultiplexer to establish synchronization with the input from the multiplexer.

13) In synchronous TDM, every data source has a fixed time slot, irrespective of whether it has any data transmit during that slot or not. For example, a human conversation is often interspersed with periods of silence. Such slots are really wasted because no useful data is sent in that slot. Statistical TDM seeks to improve efficiency by ensuring that every slot is utilized. If a particular data source has no data to send, then the slot is given over to another data source. To make it even more efficient, if a particular data source does not have enough data to utilize the whole time slot, then within one time slot we send data from more than one data source.

14) Statistical TDM is a more complex scheme to implement and requires more complex equipment at both the multiplexing and demultiplexing ends. But this is well worth the increased throughput one gets.

15) Such a situation would cause loss of data. To prevent such loss, we have a buffer in the multiplexer to store such excess data. The size of the buffer depends on how often we expect such a situation to occur and for how long it would persist. If we have a situation where the peak load from the devices is close to the average load, then there is little advantage to using statistical TDM and a simpler synchronous TDM approach would be adequate. This is because the basic premise of statistical TDM that each data source has periods of silence.

16) Voice lines are not slow because of any inherent limitation. They are kept artificially slow because that is the nature of voice transmission, given the 3400 Hz upper limit on most normal voice conversations. They therefore do not need to cater to higher frequencies and do not need a higher transmission rate. Once we remove these constraints for data transmission, high speed data access is possible over the same lines.

17) ADSL is called asymmetric because the data transmission rate available to the user for uploads and downloads are different. Of the 248 channels available for the transmission, 32 are dedicated to uploads and 216 to downloading. This is done because in practice most users download much more than they upload. So it improves the performance of the line. However, it could have easily been kept symmetric.

18) The option he would have would be to install an additional cable to serve some of the users.

19) ADSL is more secure because each link is physically independent of that used by other users. In cable, all users share the same channel with the potential to tap into the data of other users. However, the problem can be mitigated by encrypting all traffic.

20) Switching is necessary because we cannot completely connect all the devices that may have a need to communicate, as that number is very large. So whenever needed, we have to use some mechanism to connect them together.

21) In circuit switching, a dedicated path or channel is set up between the devices that want to communicate. This requires each device to have an identifying address by means of which the initiating device can request for a channel. For the duration of the conversation, the channel is dedicated to the two devices. Once the conversation is over, the circuit is dismantled and the same physical resources can be used for another pair of devices that might want to communicate.

22) In packet switching there is no dedicated channel set up for the two devices to communicate. The transmitting device merely sends out the message over the transmission medium that can be accessed by all devices. The intended recipient finds out that the message is meant for it from the address and retrieves the message.

23) In multistage switching, there are different points at which switching is done to set up the circuit, rather than having a direct path between each device to be connected. The capacity of such a switch is limited by the number of available paths at a time. So the reduction in the number of crosspoints is at the cost of possible blocking.

24) In space division switching
   a. The different paths are separated in space
   b. There are no delays during the transmission
   c. There is a set up time while the circuit is established
   d. During periods of silence, resources are wasted
   e. The number of crosspoints limits the capacity of the switch

25) Here switching is achieved by multiplexing the transmission from different sources and demultiplexing them at the output stage. There can be delays while the switching is being done. Flow control is also more of a problem here.

26) Different stages of a multistage switch can use different approaches. The time division switches help reduce the number of crosspoints while the space division switches reduce the delays. By using a combination of both one can try to optimize the number of crosspoints without having too much delay.

27) Voice conversations do not have many periods of silence, so that the circuit is used effectively. Humans can communicate with each other and ensure that flow control is done, say, if somebody is talking too fast for the other to understand. Moreover, in voice conversations we cannot tolerate delays and so circuit switching is the better choice as it gives us a dedicated channel.

28) For transmitting data, we need a switching mechanism that allows for
   - Efficiency during periods of silence. In circuit switching, the available resources are wasted during such periods as the link is dedicated to the devices and cannot be used by others.
   - Flow control is required because the sending and receiving devices may not have the capacity to communicate at the same rate. In circuit switching, such a situation would lead to loss of data.
   - If a route becomes unavailable, we should be able to change over to another route. Similarly if a better route becomes available, we should be able to utilize it.

   All the above features are provided by packet switching.

29) A virtual circuit is set up at the time a pair of devices begins to communicate using packet switching. Unlike circuit switching, we can change the path or route used if there is a failure in the route. Virtual circuits can also be permanent, that is, dedicated to a pair of devices, or can be set up for a limited time between different devices. Another difference is that in virtual circuits multiplexing can happen at the level of individual switches as well.

30) Datagrams are self contained units of information that consist of a control part that has data about the recipient, the priority and so on, as well as the actual data to be sent. Each node that receives a datagram is responsible for sending it out onto the next node till it reaches its destination. Datagrams may arrive out of sequence at the destination because they have taken different routes.

4.8 FURTHER READINGS

1) *Computer Networks*, Andrew.S.Tannenbaum, PHI, New Delhi.
2) *Data and Computer Communication*, William Stalling, PHI, New Delhi.
3) *Introduction to Data Communications and Networking*, by Behrouz Forouzan, Tata McGraw-Hill, 1999, New Delhi.