UNIT-II

OVERVIEW OF PHYSICAL LAYER SWITCHING & MULTIPLEXING

Syllabus: **Physical layer and overview of PL Switching: Multiplexing**: frequency division multiplexing, wave length division multiplexing, synchronous time division multiplexing, statistical time division multiplexing, **introduction to switching**: Circuit Switched Networks, Datagram Networks, Virtual Circuit Networks.

MULTIPLEXING:

- **Multiplexing is the set of techniques that allows the simultaneous transmission of multiple signals across a single data link.**

In a multiplexed system, ***n*** lines **share the bandwidth of one link**. Following Figure shows the basic format of a multiplexed system. The lines on the left direct their transmission streams to a **multiplexer (MUX)**, which combines them into a single stream (many-to one). At the receiving end, that stream is fed into a **demultiplexer (DEMUX)**, which separates the stream back into its component transmissions (one-to-many) and directs them to their corresponding lines.

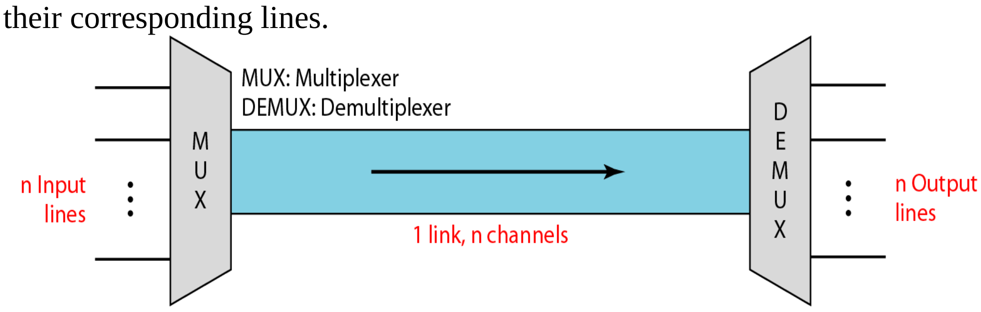


Figure: *Dividing a link into channels*

MULTIPLEXING TECHNQUIES:

There are **three** basic multiplexing techniques: **frequency-division multiplexing**, **wavelength-division multiplexing**, and **time-division multiplexing**. The first two are techniques designed for analog signals, the third, for digital signals.

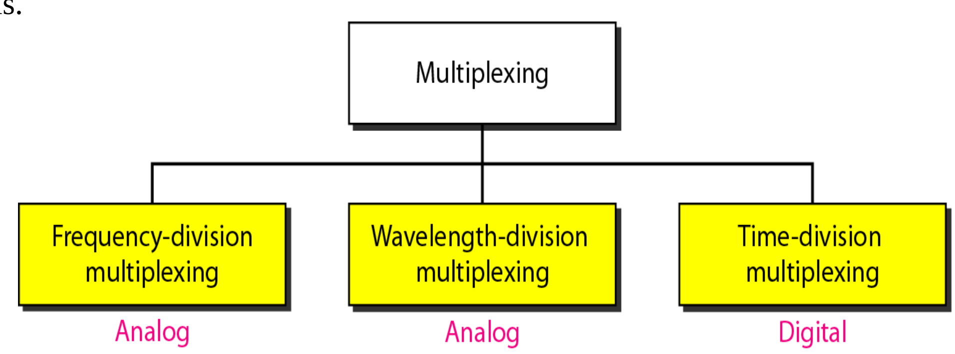


Figure: *Categories of multiplexing*

Frequency-division multiplexing (FDM):

- Frequency-division multiplexing (FDM) is an analog technique that can be applied when the bandwidth of a link is greater than the combined bandwidths of the signals to be transmitted.
- In FDM, signals generated by each sending device modulate different carrier frequencies.
- These modulated signals are then combined into a single composite signal that can be transported by the link.
- Carrier frequencies are separated by sufficient bandwidth to accommodate the modulated signal.
- These bandwidth ranges are the channels through which the various signals travel.
- Channels can be separated by strips of unused bandwidth-**guard bands**-to prevent signals from overlapping.
- In addition, carrier frequencies must not interfere with the original data frequencies.

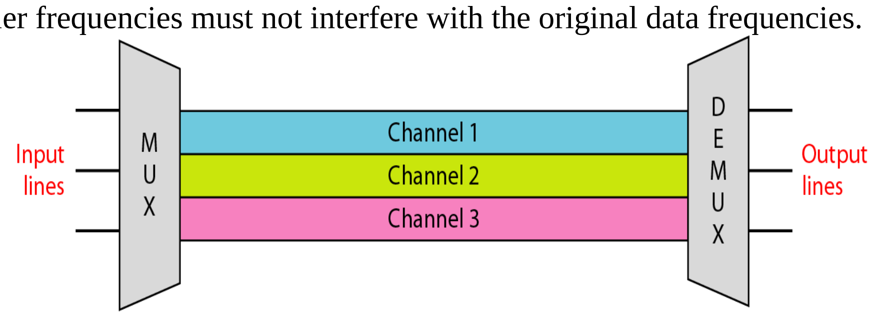


Figure: *Frequency-division multiplexing*

Above Figure gives a conceptual view of FDM. In this illustration, the transmission path is divided into three parts, each representing a channel that carries one transmission.

Multiplexing Process:

Following Figure is a conceptual illustration of the multiplexing process. Each source generates a signal of a similar frequency range. Inside the multiplexer, these similar signals modulates different carrier frequencies *(*f1,f2, and f3*)*. The resulting modulated signals are then combined into a single composite signal that is sent out over a media link that has enough bandwidth to accommodate it.

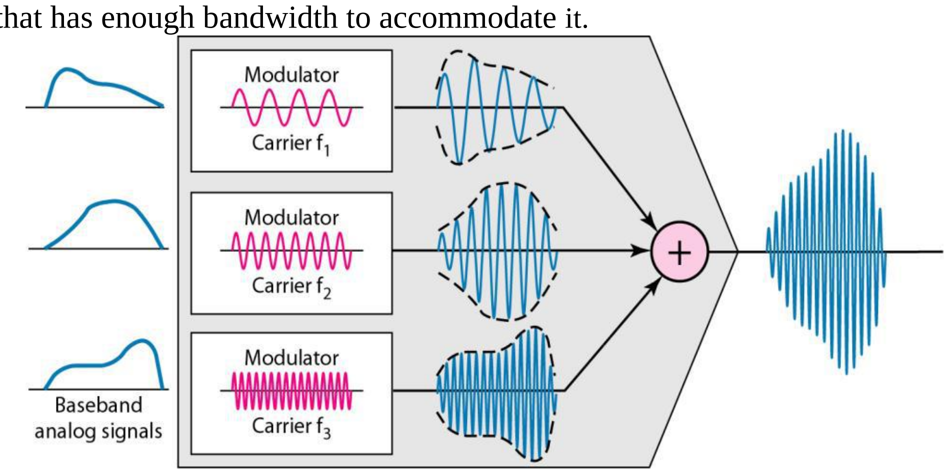


Figure: ***FDM process***

Demultiplexing Process:

The demultiplexer uses a series of filters to decompose the multiplexed signal into its constituent component signals. The individual signals are then passed to a demodulator that separates them from their carriers and passes them to the output lines. Following Figure is a conceptual illustration of demultiplexing process.

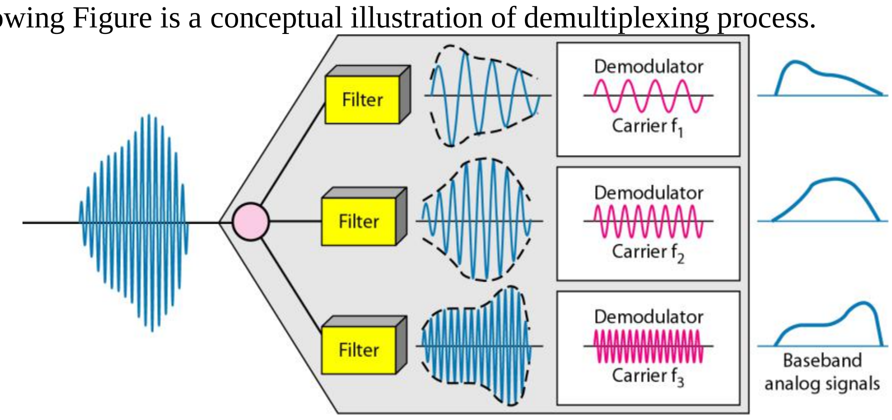


Figure: ***FDM demultiplexing example***

The Analog Carrier System:

To maximize the efficiency of their infrastructure, telephone companies have traditionally multiplexed signals from lower-bandwidth lines onto higher-bandwidth lines. In this way, many switched or leased lines can be combined into fewer but bigger channels. For analog lines, FDM is used.

One of these hierarchical systems used by AT&T is made up of groups, supergroups, master groups, and jumbo groups (see Following Figure).

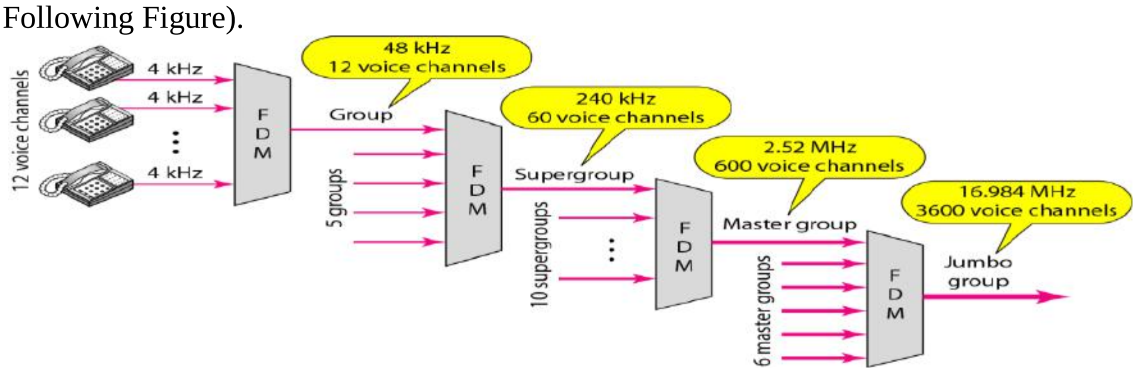


Figure: Analog hierarchy

Applications of FDM:

A very common application of FDM is AM and FM radio broadcasting.

Wavelength-Division Multiplexing:

- Wavelength-division multiplexing (WDM) is designed to use the high-data-rate capability of fiber-optic cable.
- Multiplexing allows us to combine several lines into one.
- WDM is conceptually the same as FDM, except that the multiplexing and demultiplexing involve optical signals transmitted through fiber-optic channels. The idea is the same: We are combining different signals of different frequencies. The difference is that the frequencies are very high.

Following Figure gives a conceptual view of a WDM multiplexer and demultiplexer. Very narrow bands of light from different sources are combined to make a wider band of light. At the receiver, the signals are separated by the demultiplexer.

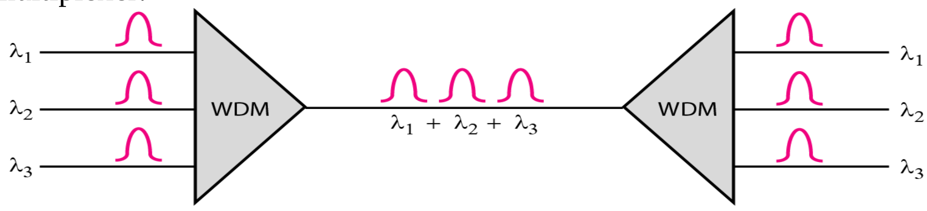


Figure: *Wavelength-division multiplexing*

- Although WDM technology is very complex, the basic idea is very simple. We want to combine multiple light sources into one single light at the multiplexer and do the reverse at the demultiplexer.
- The combining and splitting of light sources are easily handled by a prism. Recall from basic physics that a prism bends a beam of light based on the angle of incidence and the frequency.
- Using this technique, a multiplexer can be made to combine several input beams of light, each containing a narrow band of frequencies, into one output beam of a wider band of frequencies. A demultiplexer can also be made to reverse the process. Following Figure shows the concept.

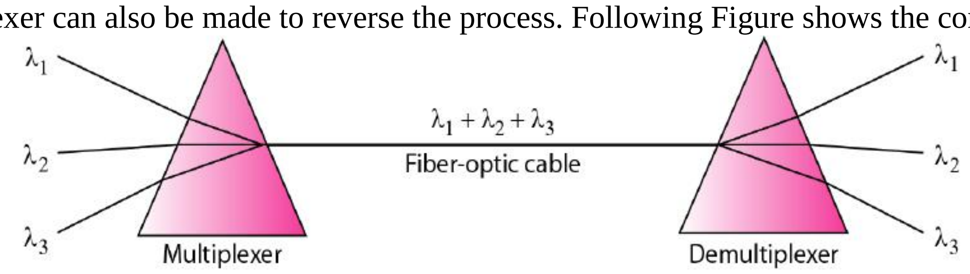


Figure: *Prisms in wavelength-division multiplexing and demultiplexing*

One application of WDM is the SONET network in which multiple optical fiber lines are multiplexed and demultiplexed.

Time-Division Multiplexing

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

In the figure, portions of signals 1, 2, 3, and 4 occupy the link sequentially.

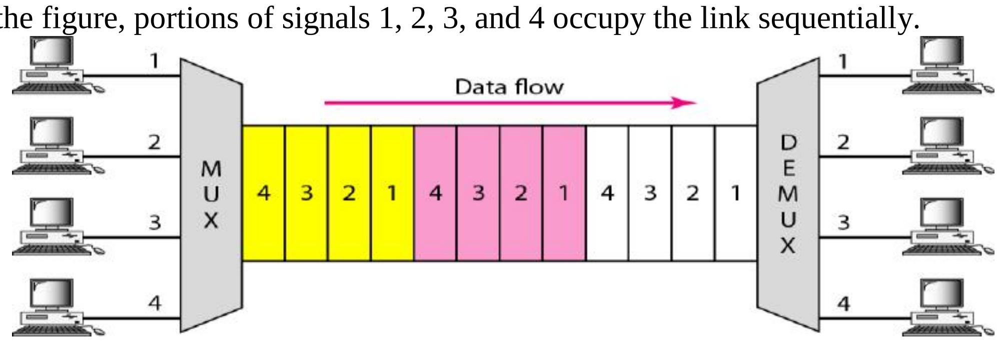


Figure: Time division multiplexing

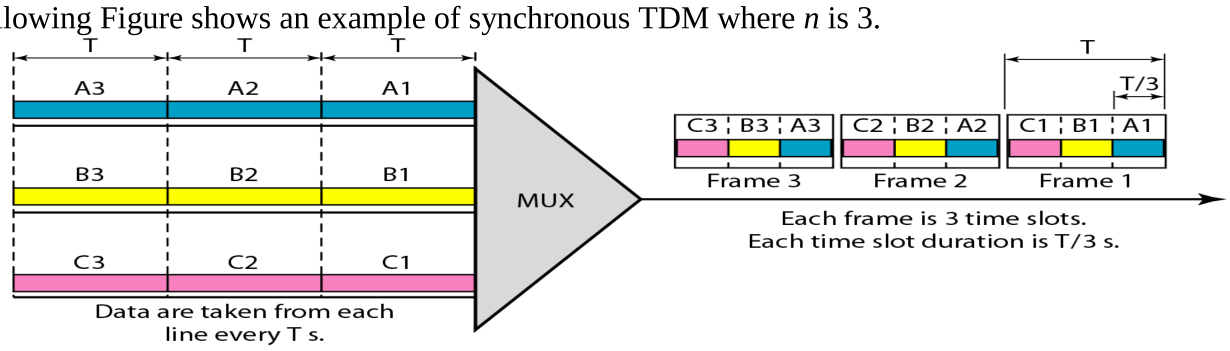
- TDM into two different schemes: **synchronous and statistical**.

Synchronous TDM: In synchronous TDM, each input connection has an allotment in the output even if it is not sending data.

Time Slots and Frames:

- In synchronous TDM, the data flow of each input connection is divided into units, where each input occupies one input time slot.
- A unit can be 1 bit, one character, or one block of data.
- Each input unit becomes one output unit and occupies one output time slot.
- However, the duration of an output time slot is n times shorter than the duration of an input time slot.
- If an input time slot is T s, the output time slot is T/n s, where n is the number of connections.

Following Figure shows an example of synchronous TDM where n is 3.



- In synchronous TDM, a round of data units from each input connection is collected into a **frame**.
- If we have *n* connections, a frame is divided into *n* time slots and one slot is allocated for each unit, one for each input line.
- If the duration of the input unit is *T*, the duration of each slot is ***T/n*** and the duration of each frame is *T*
- The data rate of the output link must be *n* times the data rate of a connection to guarantee the flow of data.

In above Figure, the data rate of the link is 3 times the data rate of a connection; likewise, the duration of a unit on a connection is 3 times that of the time slot.

Interleaving:

- TDM can be visualized as two fast-rotating switches, one on the multiplexing side and the other on the demultiplexing side.
- The switches are synchronized and rotate at the same speed, but in opposite directions.
- On the multiplexing side, as the switch opens in front of a connection, that connection has the opportunity to send a unit onto the path. This process is called **interleaving.**
- On the demultiplexing side, as the switch opens in front of a connection, that connection has the opportunity to receive a unit from the path.

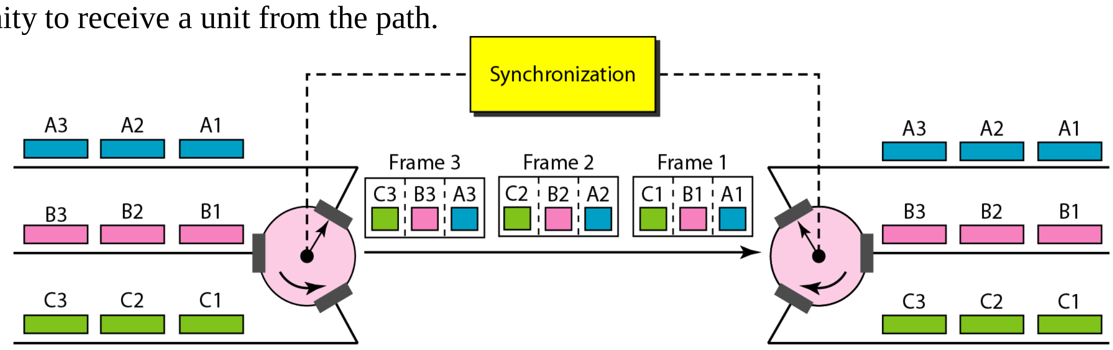


Figure: ***Interleaving***

Empty Slots

Synchronous TDM is not as efficient as it could be. If a source does not have data to send, the corresponding slot in the output frame is empty. Following Figure shows a case in which one of the input lines has no data to send and one slot in another input line has discontinuous data.

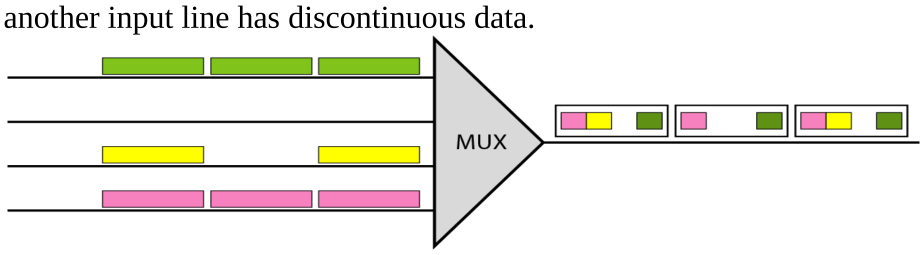


Figure: empty slots

Data Rate Management:

One problem with TDM is how to handle a disparity in the input data rates. We assumed that the data rates of all input lines were the same. However, if data rates are not the same, three strategies, or a combination of them, can be used. We call these three strategies **multilevel multiplexing, multiple-slot allocation,** and **pulse stuffing.**

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

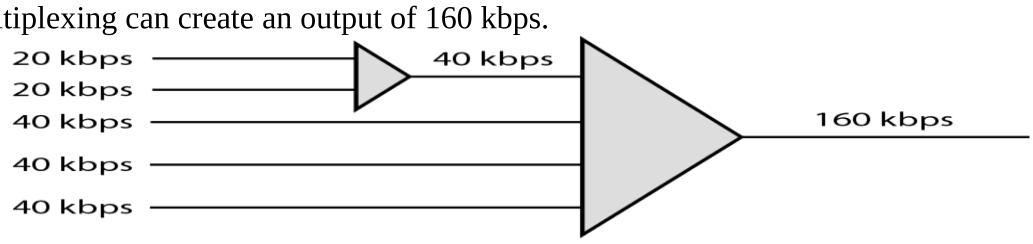


Figure: ***Multilevel multiplexing***

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

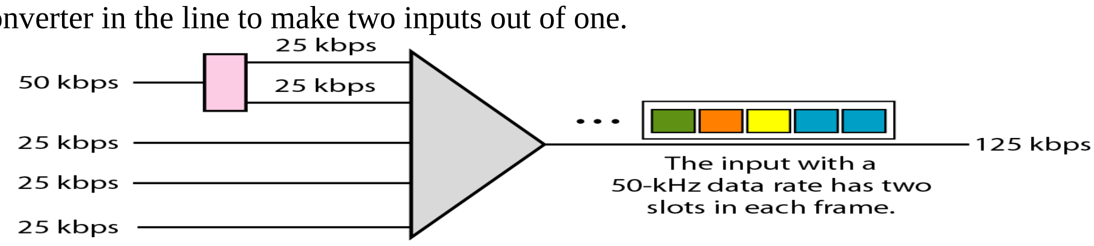
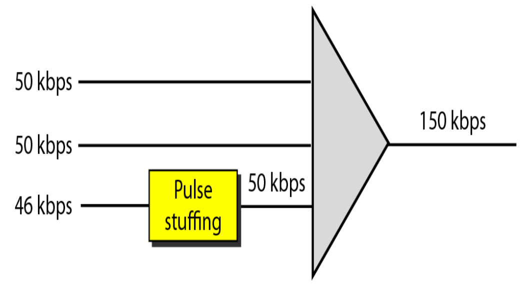


Figure: Multiple-slot multiplexing

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



Frame Synchronizing:

The implementation of TDM is not as simple as that of FDM. If the multiplexer and the demultiplexer are not synchronized, a bit belonging to one channel may be received by the wrong channel. For this reason, one or more synchronization bits are usually added to the beginning of each frame. These bits, called **framing bits**, follow a pattern, frame to frame, that allows the demultiplexer to synchronize with the incoming stream so that it can separate the time slots accurately. In most cases, this synchronization information consists of 1 bit per frame, alternating between 0 and 1, as shown in Following Figure.

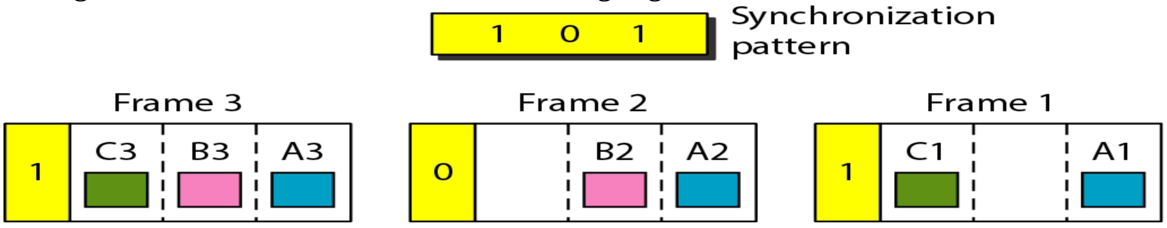


Figure: Framing bits

Digital Signal Service:

Telephone companies implement TDM through a hierarchy of digital signals, called **digital signal (DS)** service or digital hierarchy. Following Figure shows the data rates supported by each level.

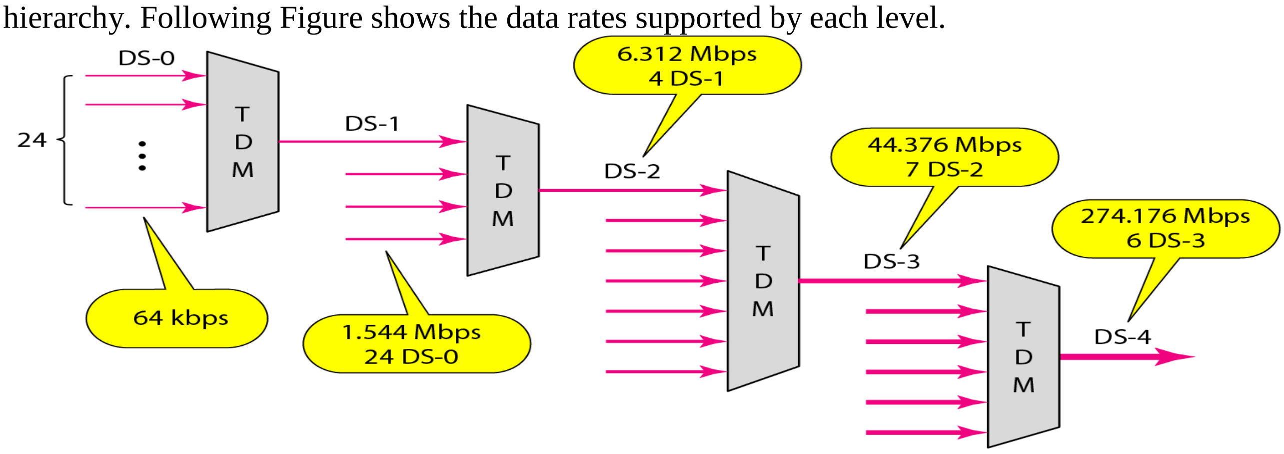


Figure: ***Digital hierarchy***

- A **DS-O** service is a single digital channel of 64 kbps.
- **DS-1** is a 1.544-Mbps service; 1.544 Mbps is 24 times 64 kbps plus 8 kbps of overhead. It can be used as a single service for 1.544-Mbps transmissions, or it can be used to multiplex 24 DS-O channels or to carry any other combination desired by the user that can fit within its 1.544-Mbps capacity.
- **DS-2** is a 6.312-Mbps service; 6.312 Mbps is 96 times 64 kbps plus 168 kbps of overhead. It can be used as a single service for 6.312-Mbps transmissions; or it can be used to multiplex 4 DS-l channels, 96 DS-O channels, or a combination of these service types.
- **DS-3** is a 44.376-Mbps service; 44.376 Mbps is 672 times 64 kbps plus 1.368 Mbps of overhead. It can be used as a single service for 44.376-Mbps transmissions; or it can be used to multiplex 7 DS-2 channels, 28 DS-l channels, 672 DS-O channels, or a combination of these service types.
- **DS-4** is a 274. 176-Mbps service; 274.176 is 4032 times 64 kbps plus 16.128 Mbps of overhead. It can be used to multiplex 6 DS-3 channels, 42 DS-2 channels, 168 DS-l channels, 4032 DS-O channels, or a combination of these service types.

T Lines:

DS-O, DS-l, and so on are the names of services. To implement those services, the telephone companies use **T lines** (T-l to T-4). These are lines with capacities precisely matched to the data rates of the DS-l to DS-4 services. So far only T-l and T-3 lines are commercially available.

Service	*Line*	*Rate (Mbps)*	*Voice Channels*
DS-1	T-1	1.544	24
DS-2	T-2	6.312	96
DS-3	T-3	44.736	672
DS-4	T-4	274.176	4032

Table: ***DS and T line rates***

T Lines for Analog Transmission:

T lines are digital lines designed for the transmission of digital data, audio, or video. However, they also can be used for analog transmission (regular telephone connections), provided the analog signals are first sampled, then time-division multiplexed.

24 voice channels can be multiplexed onto one T-1 line

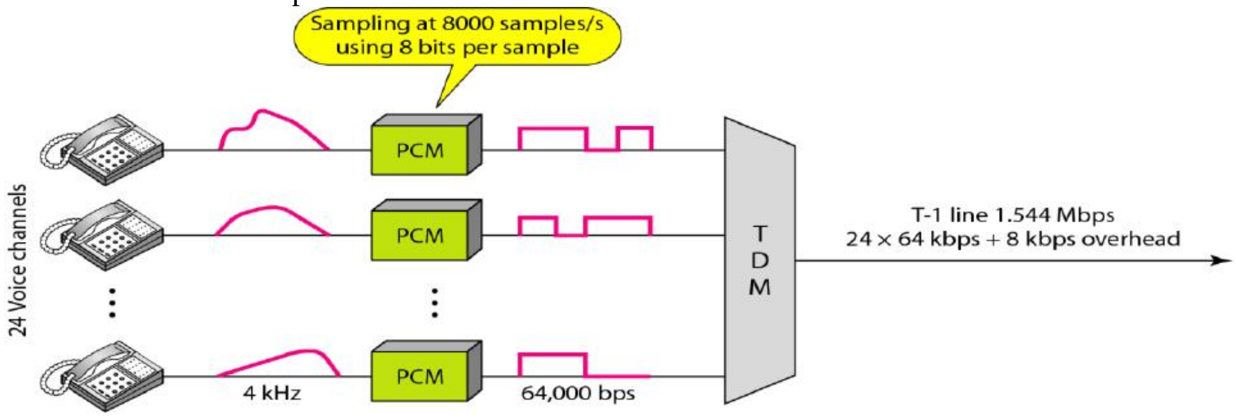


Figure: *T-1 line for multiplexing telephone lines*

E Lines:

Europeans use a version of T lines called E lines. The two systems are conceptually identical, but their capacities differ. Following Table shows the E lines and their capacities.

Line	*Rate (Mbps)*	*Voice Channels*
E-1	2.048	30
E-2	8.448	120
E-3	34.368	480
E-4	139.264	1920

Table : *E line rates*

Statistical Time-Division Multiplexing:

- In statistical time-division multiplexing, slots are dynamically allocated to improve bandwidth efficiency.
- Only when an input line has a slot's worth of data to send is it given a slot in the output frame.
- In statistical multiplexing, the number of slots in each frame is less than the number of input lines.
- The multiplexer checks each input line in round robin fashion; it allocates a slot for an input line if the line has data to send; otherwise, it skips the line and checks the next line.

Following Figure shows a synchronous and a statistical TDM example. In the former, some slots are empty because the corresponding line does not have data to send. In the latter, however, no slot is left empty as long as there are data to be sent by any input line.

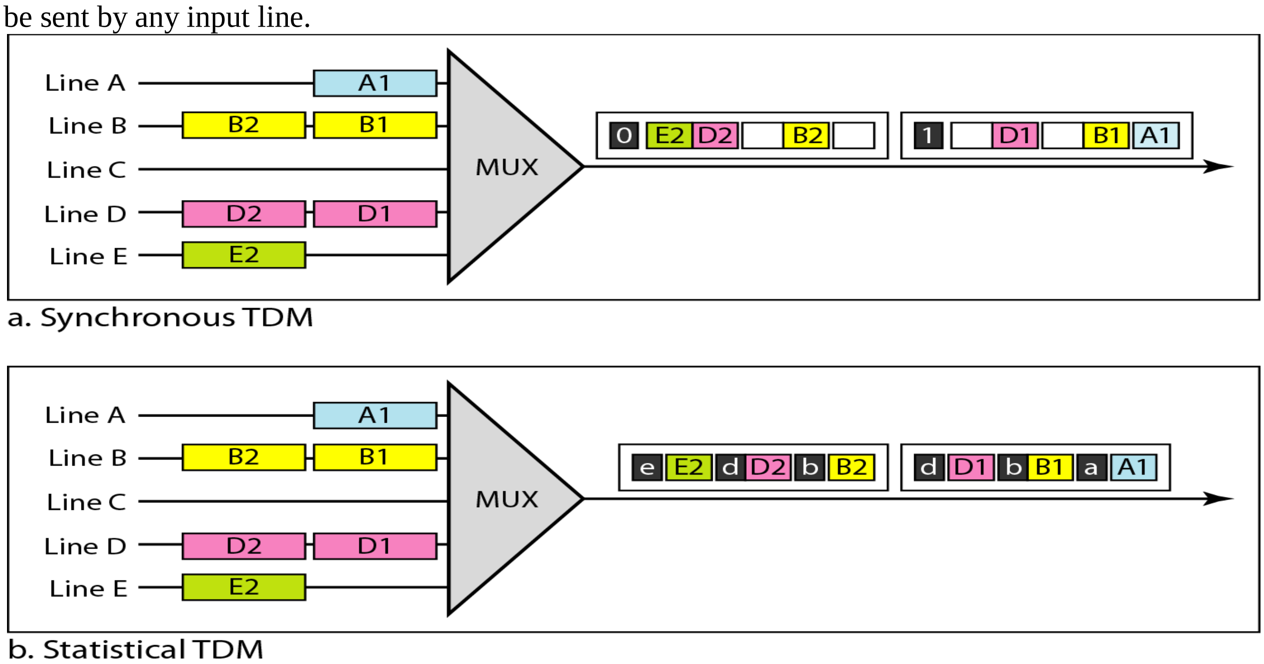


Figure :TDM Slot comparsion

Addressing:

- Above Figure also shows a major difference between slots in synchronous TDM and statistical TDM.
- An output slot in synchronous TDM is totally occupied by data; in statistical TDM, a slot needs to carry data as well as the address of the destination.
- In synchronous TDM, there is no need for addressing; synchronization and preassigned relationships between the inputs and outputs serve as an address.
- In statistical multiplexing, there is no fixed relationship between the inputs and outputs because there are no preassigned or reserved slots. We need to include the address of the receiver inside each slot to show where it is to be delivered. The addressing in its simplest form can be *n* bits to define *N* different output lines with $n=\log_2 N$. For example, for eight different output lines, we need a 3-bit address.

Slot Size:
Since a slot carries both data and an address in statistical TDM, the ratio of the data size to address size must be reasonable to make transmission efficient. For example, it would be inefficient to send 1 bit per slot as data when the address is 3 bits. This would mean an overhead of 300 percent. In statistical TDM, a block of data is usually many bytes while the address is just a few bytes.

No Synchronization Bit:
There is another difference between synchronous and statistical TDM, but this time it is at the frame level. The frames in statistical TDM need not be synchronized, so we do not need synchronization bits.
Bandwidth:
In statistical TDM, the capacity of the link is normally less than the sum of the capacities of each channel. The designers of statistical TDM define the capacity of the link based on the statistics of the load for each channel. If on average only *x* percent of the input slots are filled, the capacity of the link reflects this. Of course, during peak times, some slots need to wait.

Introduction to switching:

A switched network consists of a series of interlinked nodes, called switches. Switches are devices capable of creating temporary connections between two or more devices linked to the switch. In a switched network, some of these nodes are connected to the end systems

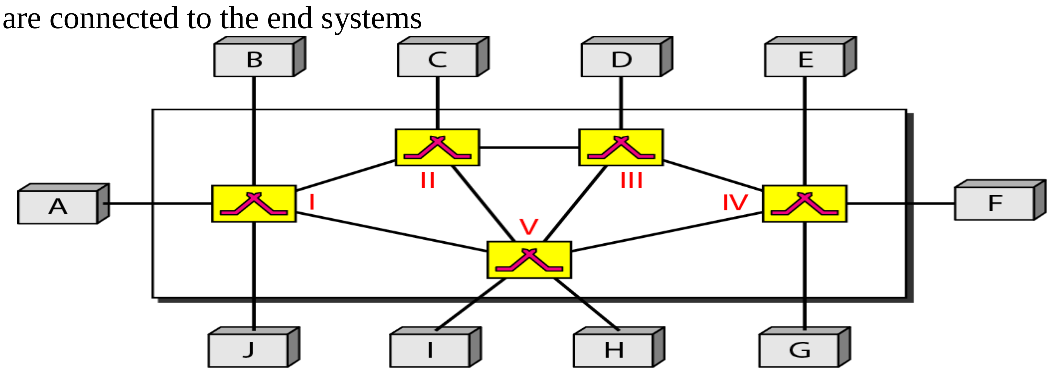


Figure: ***Switched network***

The end systems (communicating devices) are labeled A, B, C, D, and so on, and the switches are labeled I, II, III, IV, and V. Each switch is connected to multiple links.

TYPES OF SWITCHED NETWORKS:

Traditionally, three methods of switching have been important: **circuit switching, packet switching, and message switching.**
We can then divide networks into **three** broad categories: **circuit-switched networks, packet-switched networks, and message-switched**. **Packet-switched networks** can further be divided into two subcategories **virtual-circuit networks and datagram networks.**

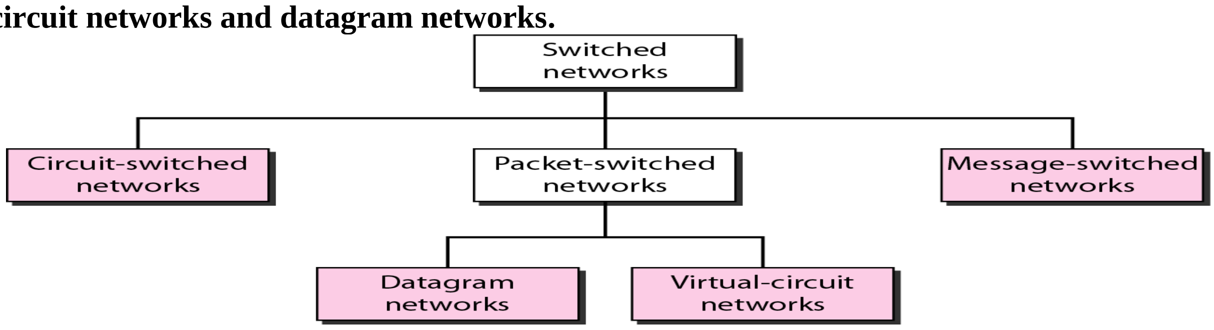


Figure: *Taxonomy of switched networks*

CIRCUIT-SWITCHED NETWORKS:

A circuit-switched network consists of a set of switches connected by physical links. A connection between two stations is a dedicated path made of one or more links. However, each connection uses only one dedicated channel on each link. Each link is normally divided into *n* channels by using FDM or TDM.
Following Figure shows a trivial circuit-switched network with four switches and four links. Each link is divided into *n (n* is 3 in the figure) channels by using FDM or TDM.

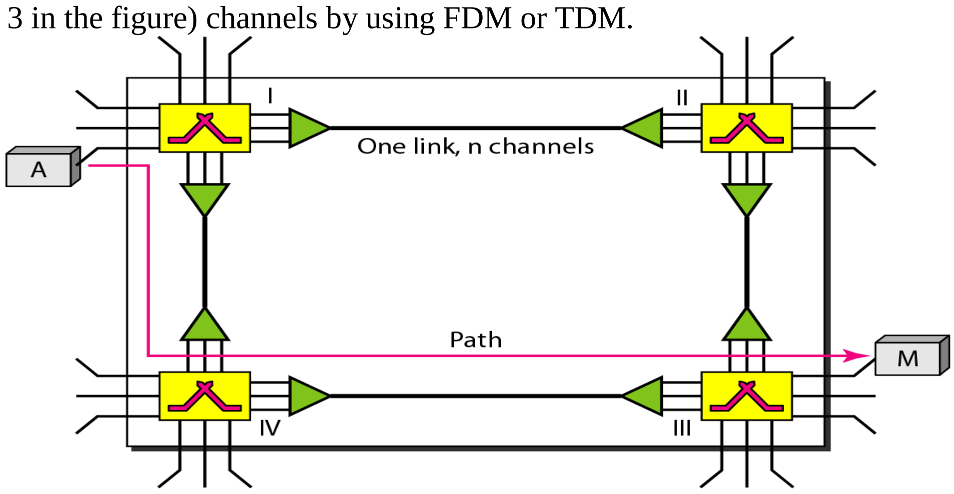


Figure: ***A trivial circuit-switched network***

We have shown only two end systems for simplicity. When end system A needs to communicate with end system M, system A needs to request a connection to M that must be accepted by all switches as well as by M itself. This is called the **setup phase**; a circuit (channel) is reserved on each link, and the combination of circuits or channels defines the dedicated path. After the dedicated path made of connected circuits (channels) is established, data transfer can take place. After all data have been transferred, the circuits are torn down.

Important points here:

- Circuit switching takes place at the physical layer.
- Before starting communication, the stations must make a reservation for the resources to be used during the communication. These resources, such as channels, switch buffers, switch processing time, and switch input/output ports, must remain dedicated during the entire duration of data transfer until the teardown phase.
- Data transferred between the two stations are not packetized. The data are a continuous flow sent by the source station and received by the destination station.
- There is no addressing involved during data transfer. The switches route the data based on their occupied band (FDM) or time slot (TDM). Of course, there is end-to end addressing used during the setup phase.

Three Phases of circuit switched network:

The actual communication in a **circuit-switched network** requires three phases: **connection Setup, data transfer, and connection teardown**.

Setup Phase

Before the two parties can communicate, a dedicated circuit needs to be established. The end systems are normally connected through dedicated lines to the switches, so connection setup means creating dedicated channels between the switches. For example, in above Figure, when system A needs to connect to system M, it sends a setup request that includes the address of system M, to switch I. Switch I finds a channel between itself and switch IV that can be dedicated for this purpose. Switch I then sends the request to switch IV, which finds a dedicated channel between itself and switch III. Switch III informs system M of system A's intention at this time.

In the next step to making a connection, an acknowledgment from system M needs to be sent in the opposite direction to system A. Only after system A receives this acknowledgment is the connection established.

Note that end-to-end addressing is required for creating a connection between the two end systems. These can be, for example, the addresses of the computers assigned by the administrator in a TDM network, or telephone numbers in an FDM network.

Data Transfer Phase

After the establishment of the dedicated circuit, the two parties can transfer data.

Teardown Phase

When one of the parties needs to disconnect, a signal is sent to each switch to release the resources.

Efficiency of circuit switched network:

It can be argued that circuit-switched networks are not as efficient as the other two types of networks because resources are allocated during the entire duration of the connection. These resources are unavailable to other connections.

Delay of circuit switched network:

Although a circuit-switched network normally has low efficiency, the delay in this type of network is minimal. During data transfer the data are not delayed at each switch; the resources are allocated for the duration of the connection. Following Figure shows the idea of delay in a circuit-switched network when only two switches are involved. As Figure shows, there is no waiting time at each switch. The total delay is due to the time needed to create the connection, transfer data, and disconnect the circuit. The delay caused by the setup is the sum of four parts: the propagation time of the source computer request (slope of the first gray box), the request signal transfer time (height of the first gray box), the propagation time of the acknowledgment from the destination computer (slope of the second gray box), and the signal transfer time of the acknowledgment (height of the second gray box). The delay due to data transfer is the sum of two parts: the propagation time (slope of the colored box) and data transfer time (height of the colored box), which can be very long. The third box shows the time needed to tear down the circuit.

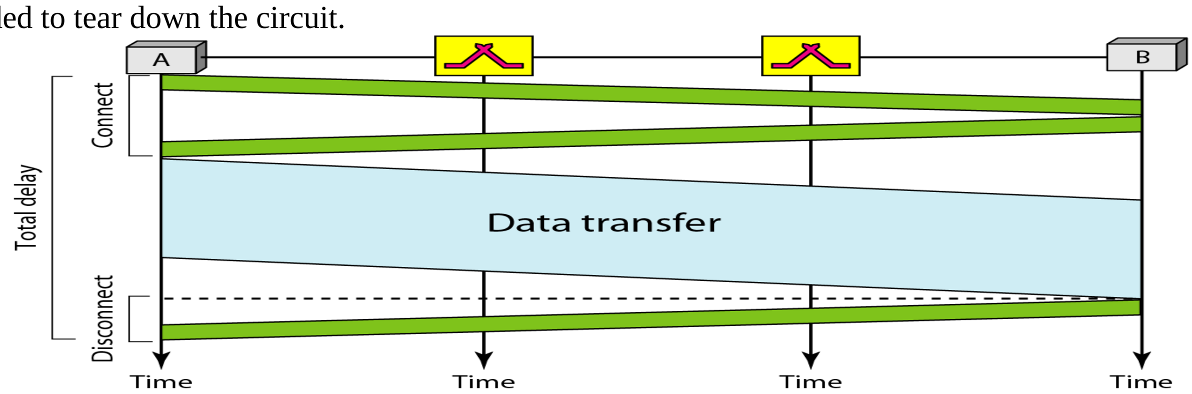


Figure: *Delay in a circuit-switched network*

- Circuit switched technology is used in telephone networks

DATAGRAM NETWORKS:

In data communications, we need to send messages from one end system to another.

- If the message is going to pass through a packet-switched network, it needs to be divided into packets of fixed or variable size.
- The size of the packet is determined by the network and the governing protocol.
- In packet switching, there is no resource allocation for a packet. This means that there is no reserved bandwidth on the links, and there is no scheduled processing time for each packet.
- Resources are allocated on demand.
- The allocation is done on a first come, first-served basis.
- When a switch receives a packet, no matter what is the source or destination, the packet must wait if there are other packets being processed.
- In a datagram network, each packet is treated independently of all others.
- Even if a packet is part of a multipacket transmission, the network treats it as though it existed alone.
- Packets in this approach are referred to as **datagrams**.
- Datagram switching is normally done at the **network layer**.

Following Figure shows how the datagram approach is used to deliver four packets from station A to station X. The switches in a datagram network are traditionally referred to as routers.

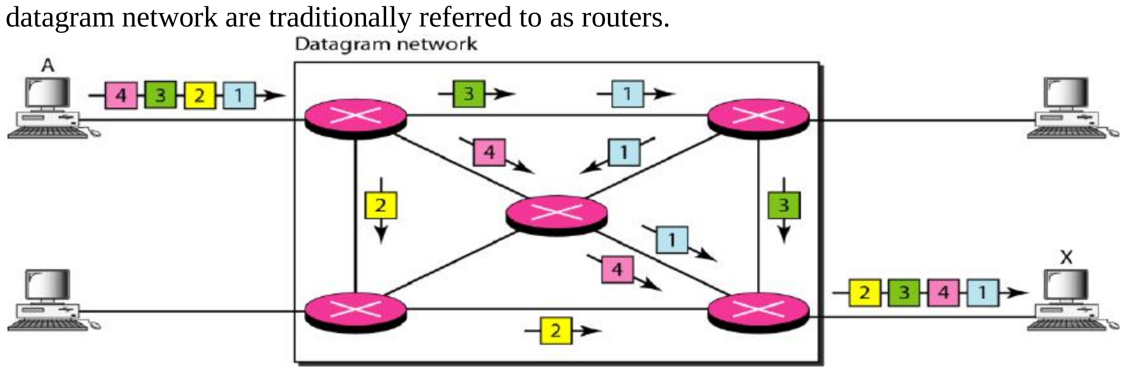
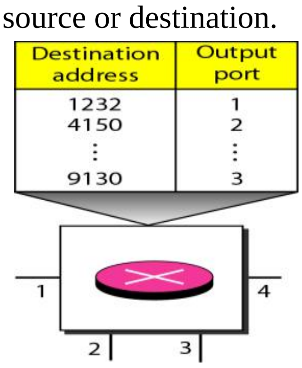


Figure: *A datagram network with four switches (routers)*

In this example, all four packets belong to the same message, but may travel different paths to reach their destination. This is so because the links may be involved in carrying packets from other sources and do not have the necessary bandwidth available to carry all the packets from A to X. This approach can cause the datagrams of a transmission to arrive at their destination out of order with different delays between the packets. Packets may also be lost or dropped because of a lack of resources. In most protocols, it is the responsibility of an upper-layer protocol to reorder the datagrams or ask for lost datagrams before passing them on to the application.

The datagram networks are sometimes referred to as connectionless networks. The term *connectionless* here means that the switch (packet switch) does not keep information about the connection state. There are **no setup or teardown phases**. Each packet is treated the same by a switch regardless of its source or destination.

Routing Table:



If there are no setup or teardown phases, how are the packets routed to their destinations in a datagram network? In this type of network, each switch has a **routing table** which is based on the destination address. The routing tables are dynamic and are updated periodically. The destination addresses and the corresponding forwarding output ports are recorded in the tables. Figure shows the routing table for a switch.

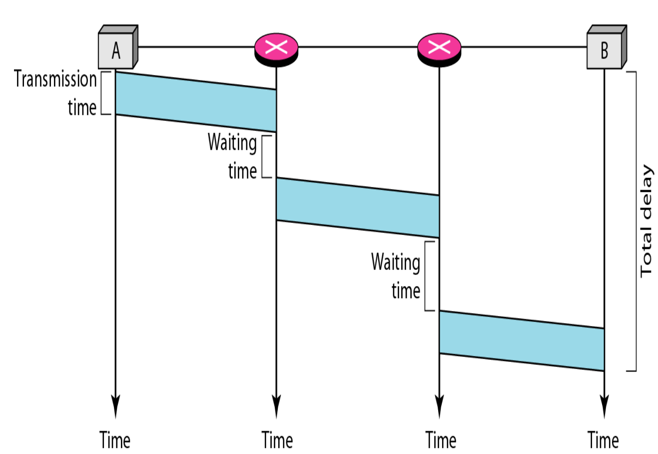
Destination Address:

Every packet in a datagram network carries a header that contains, among other information, the destination address of the packet. When the switch receives the packet, this destination address is examined; the routing table is consulted to find the corresponding port through which the packet should be forwarded. This address, unlike the address in a virtual-circuit-switched network, remains the same during the entire journey of the packet.

Efficiency

The efficiency of a datagram network is better than that of a circuit-switched network; resources are allocated only when there are packets to be transferred. If a source sends a packet and there is a delay of a few minutes before another packet can be sent, the resources can be reallocated during these minutes for other packets from other sources.

Delay



There may be greater delay in a datagram network than in a virtual-circuit network. Although there are no setup and teardown phases, each packet may experience a wait at a switch before it is forwarded. In addition, since not all packets in a message necessarily travel through the same switches, the delay is not uniform for the packets of a message. Figure gives an example of delay in a datagram network for one single packet.

The packet travels through two switches. There are three transmission times *(3T),* three propagation delays (slopes 3'Ŧ of the lines), and two waiting times ($w_1 + w_2$). We ignore the processing time in each switch.

Total delay $= 3T + 3\tau + w_1 + w_2$

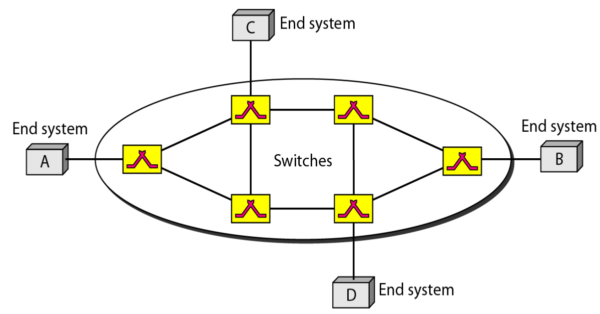
- **Datagram networks are used in Internet.**

VIRTUAL-CIRCUIT NETWORKS:

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

1. As in a circuit-switched network, there are setup and teardown phases in addition to the data transfer phase.
2. Resources can be allocated during the setup phase, as in a circuit-switched network, or on demand, as in a datagram network.
3. As in a datagram network, data are packetized and each packet carries an address in the header.
4. As in a circuit-switched network, all packets follow the same path established during the connection.
5. A **virtual-circuit network** is normally implemented in the **data link layer**, while a **circuit-switched network** is implemented in the **physical layer** and a **datagram network** in the **network layer**.

Figure is an example of a virtual-circuit network. The network has switches that allow traffic from sources to destinations. A source or destination can be a computer, packet switch, bridge, or any other device that connects other networks.



Addressing

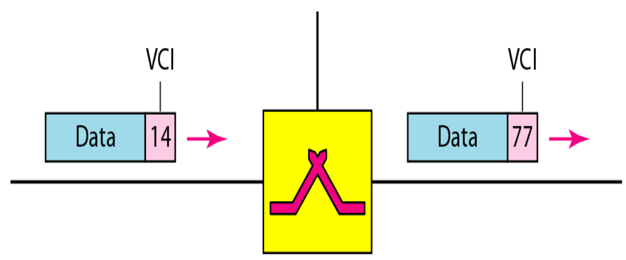
In a virtual-circuit network, two types of addressing are involved: **global and local** (virtual-circuit identifier).

Global Addressing

A source or a destination needs to have a global address-an address that can be unique in the scope of the network or internationally if the network is part of an international network. However, we will see that a global address in virtual-circuit networks is used only to create a virtual-circuit identifier.

Virtual-Circuit Identifier

The identifier that is actually used for data transfer is called the virtual-circuit identifier (VCI). A VCI, unlike a global address, is a small number that has only switch scope; it is used by a frame between two switches. When a frame arrives at a switch, it has a VCI; when it leaves, it has a different VCI. Figure shows how the VCI in a data frame changes from one switch to another.



Three Phases

As in a circuit-switched network, a source and destination need to go through three phases in a virtual-circuit network: **setup, data transfer, and teardown**. In the setup phase, the source and destination use their global addresses to help switches make table entries for the connection. In the teardown phase, the source and destination inform the switches to delete the corresponding entry. Data transfer occurs between these two phases.

Data Transfer Phase

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

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

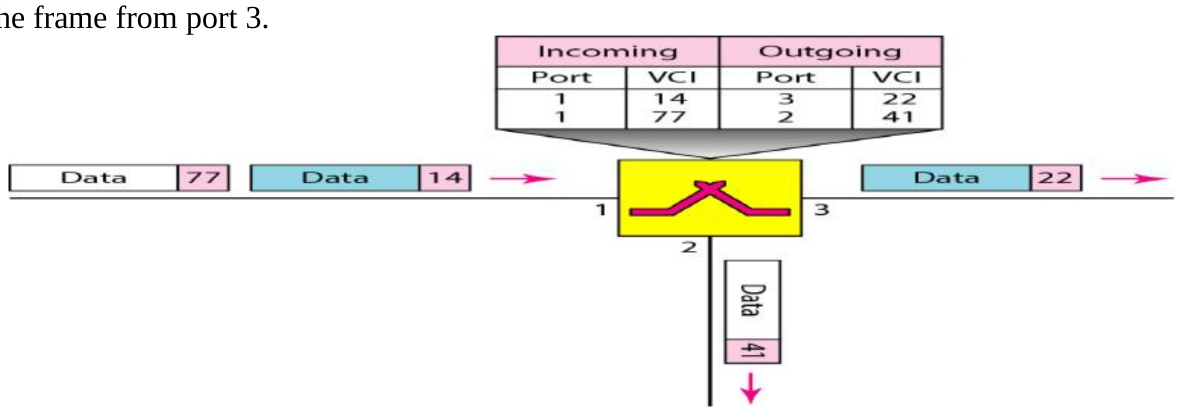


Figure: ***Switch and tables in a virtual-circuit network***

Following Figure shows how a frame from source A reaches destination B and how its VCI changes during the trip. Each switch changes the VCI and routes the frame.

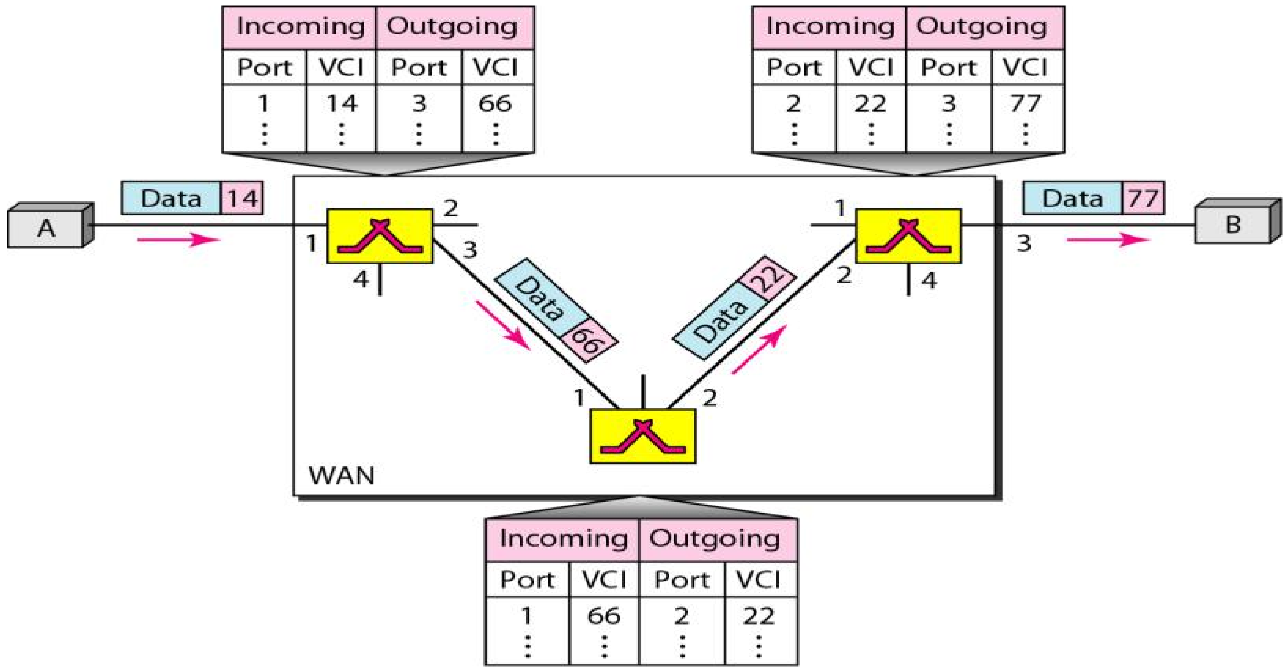


Figure: ***Source-to-destination data transfer in a virtual-circuit network***

The data transfer phase is active until the source sends all its frames to the destination. The procedure at the switch is the same for each frame of a message. The process creates a virtual circuit, not a real circuit, between the source and destination.

<u>Setup Phase</u>

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

<u>Setup Request</u>: A setup request frame is sent from the source to the destination. Following Figure shows the process.

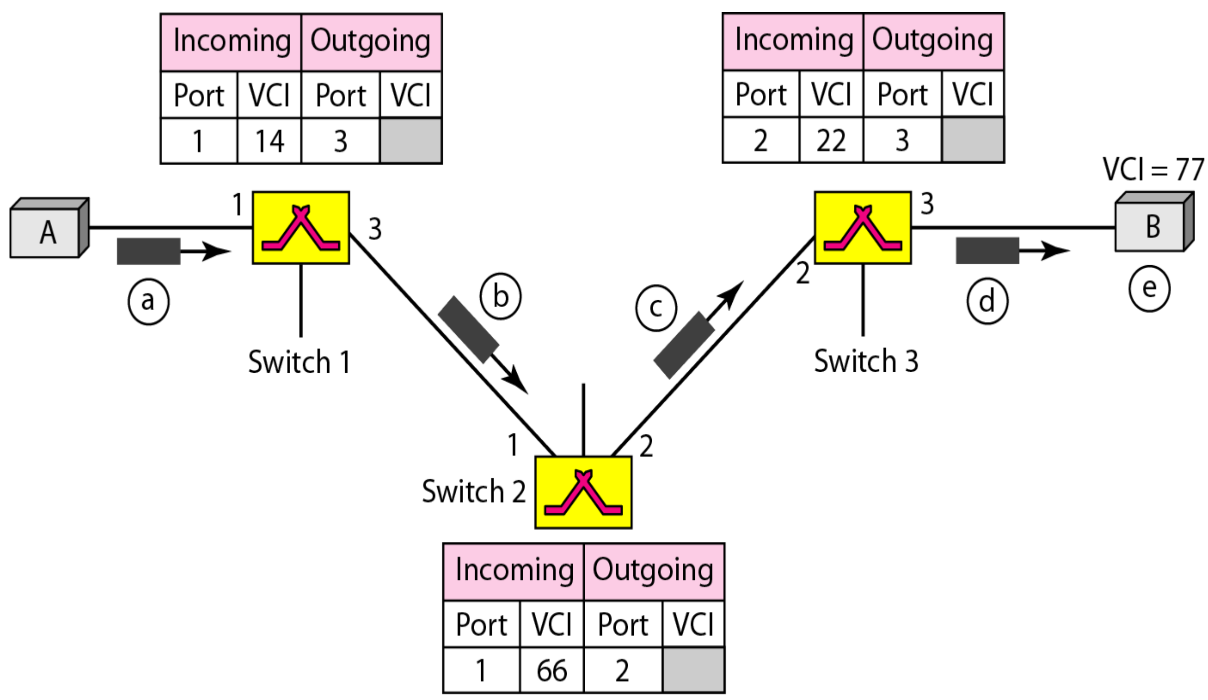


Figure: ***Setup request in a virtual-circuit network***

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

<u>Acknowledgment</u> A special frame, called the acknowledgment frame, completes the entries in the switching tables. Following Figure shows the process.

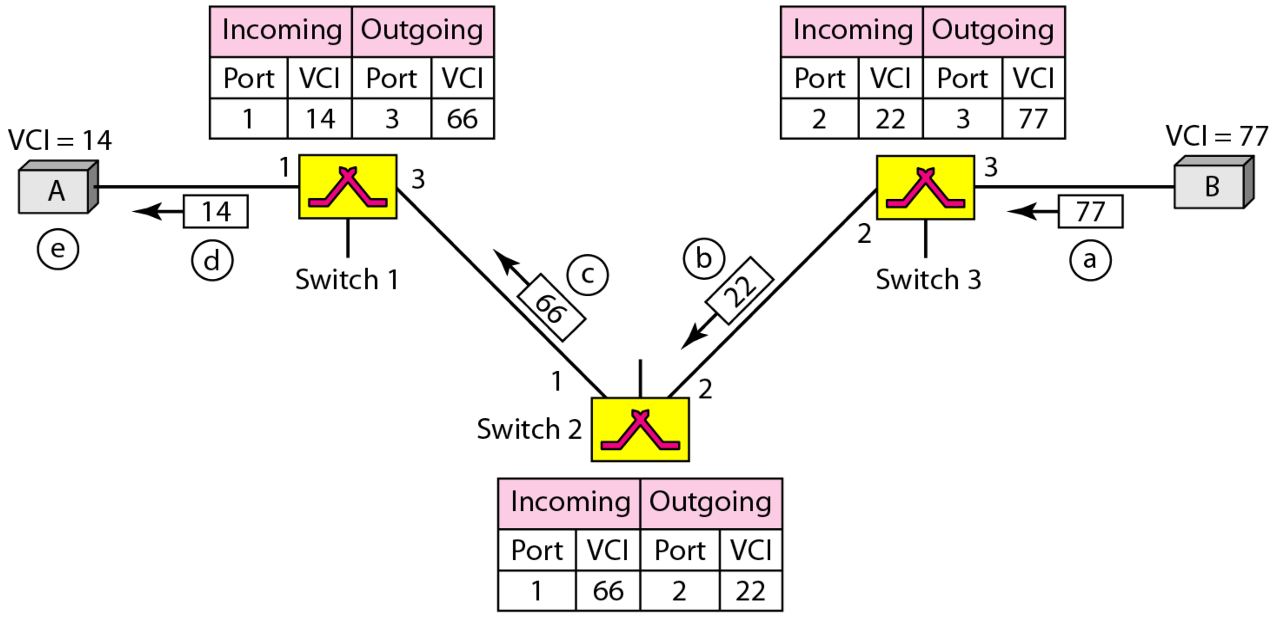


Figure: Setup acknowledgment in a virtual-circuit network

a. The destination sends an acknowledgment to switch 3. The acknowledgment carries the global source and destination addresses so the switch knows which entry in the table is to be completed. The frame also carries VCI 77, chosen by the destination as the incoming VCI for frames from A. Switch 3 uses this VCI to complete the outgoing VCI column for this entry. Note that 77 is the incoming VCI for destination B, but the outgoing VCI for switch 3.
b. Switch 3 sends an acknowledgment to switch 2 that contains its incoming VCI in the table, chosen in the previous step. Switch 2 uses this as the outgoing VCI in the table.
c. Switch 2 sends an acknowledgment to switch 1 that contains its incoming VCI in the table, chosen in the previous step. Switch 1 uses this as the outgoing VCI in the table.
d. Finally switch 1 sends an acknowledgment to source A that contains its incoming VCI in the table, chosen in the previous step.
e. The source uses this as the outgoing VCI for the data frames to be sent to destination B.

<u>Teardown Phase</u>

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

<u>Efficiency:</u>

Resource reservation in a virtual-circuit network can be made during the setup or can be on demand during the data transfer phase. In the first case, the delay for each packet is the same; in the second case, each packet may encounter different delays. There is one big advantage in a virtual-circuit network even if resource allocation is on demand. The source can check the availability of the resources, without actually reserving it. Consider a family that wants to dine at a restaurant. Although the restaurant may not accept reservations, the family can call and find out the waiting time. This can save the family time and effort.

In virtual-circuit switching, all packets belonging to the same source and destination travel the same path; but the packets may arrive at the destination with different delays if resource allocation is on demand.

<u>Delay in Virtual-Circuit Networks:</u>

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

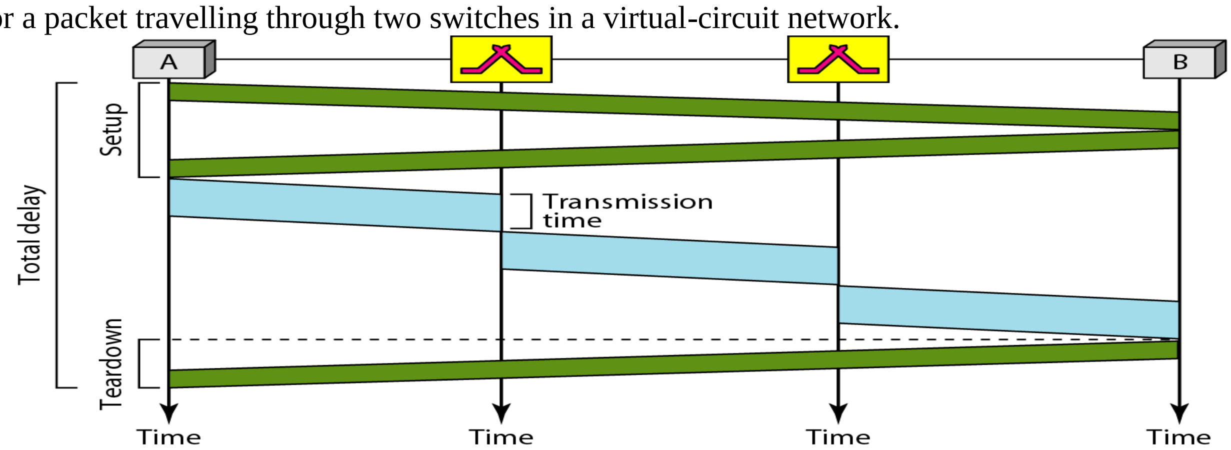


Figure: ***Delay in a virtual-circuit network***

The packet is travelling through two switches (routers). There are three transmission times *(***3T***)*, three propagation times (**3Ŧ**), data transfer depicted by the sloping lines, a setup delay (which includes transmission and propagation in two directions), and a teardown delay (which includes transmission and propagation in one direction).We ignore the processing time in each switch. The total delay time is

Total delay = 3T+3Ŧ+setup delay+ teardown delay

- **Virtual Switched Technology is used in WAN's.**