Module 6

Spread Specturm Techniques

Spread Spectrum combine signals from different sources to fit into a larger bandwidth. It is designed to be used in wireless applications (LANs and WANs). In wireless applications, all stations use air (or a vacuum) as the medium for communication. Stations must be able to share this medium without interception by an eavesdropper and without being subject to jamming from a malicious intruder (in military operations, for example). To achieve these goals, spread spectrum techniques add redundancy; they spread the original spectrum needed for each station. If the required bandwidth for each station is *B*, spread spectrum expands it to Bss' such that Bss » *B*. The expanded bandwidth allows the source to wrap its message in a protective envelope for a more secure transmission.

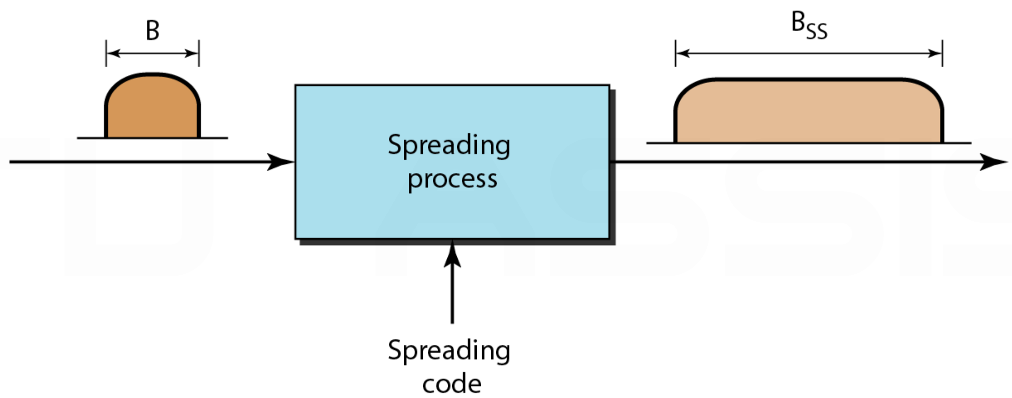


Figure above shows the idea of spread spectrum. Spread spectrum achieves its goals through two principles:

1. The bandwidth allocated to each station needs to be, by far, larger than what is needed. This allows redundancy.
2. The expanding of the original bandwidth *B* to the bandwidth Bss must be done by a process that is independent of the original signal. In other words, the spreading process occurs after the signal is created by the source. After the signal is created by the source, the spreading process uses a spreading code and spreads the bandwidth. The figure shows the original bandwidth *B* and the spreaded bandwidth *BSS*. The spreading code is a series of numbers that look random, but are actually a pattern.

There are two techniques to spread the bandwidth: frequency hopping spread spectrum (FHSS) and direct sequence spread spectrum (DSSS).

1) Frequency Hopping Spread Spectrum (FHSS)

The frequency hopping spread spectrum (FHSS) technique uses *M* different carrier frequencies that are modulated by the source signal. At one moment, the signal modulates one carrier frequency; at the next moment, the signal modulates another carrier frequency. Although the modulation is done using one carrier frequency at a time, *M* frequencies are used in the long run. The bandwidth occupied by a source after spreading is $B_{pHSS} » B$.

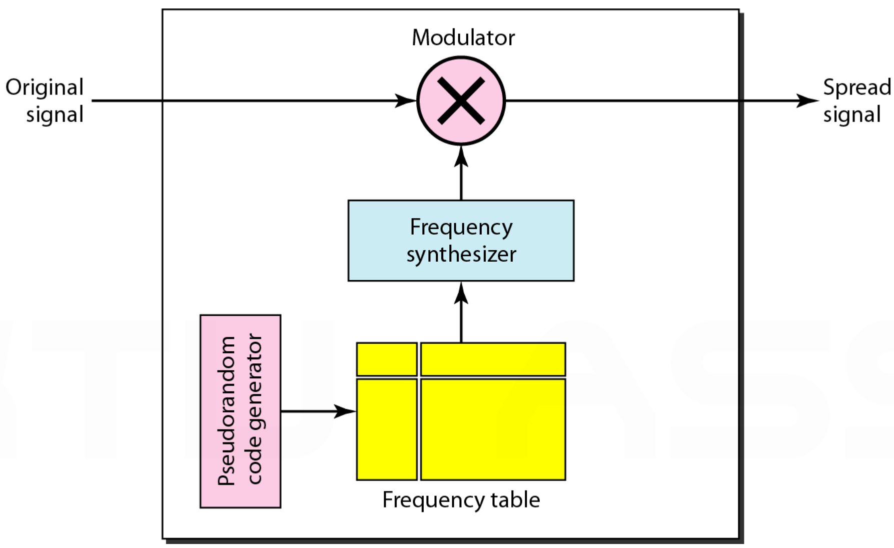


Figure above shows the general layout for FHSS. A pseudorandom code generator, called **pseudorandom noise (PN)**, creates a k-bit pattern for every hopping period T_h. The frequency table uses the pattern to find the frequency to be used for this hopping period and passes it to the frequency synthesizer. The frequency synthesizer creates a carrier signal of that frequency, and the source signal modulates the carrier signal. Suppose we have decided to have eight hopping frequencies. In this case, *M is* 8 and *k* is 3. The pseudorandom code generator will create eight different 3-bit patterns. These are mapped to eight different frequencies in the frequency table.

Example:

The pattern for the station is 101, 111, 001, 000, 010, all, 100. This means that at hopping period 1, the pattern is 101. The frequency selected is 700 kHz; the source signal modulates this carrier frequency. The second k-bit pattern selected is 111, which selects the 900-kHz carrier; the eighth pattern is 100, the frequency is 600 kHz. After eight hoppings, the pattern repeats, starting from 101 again.

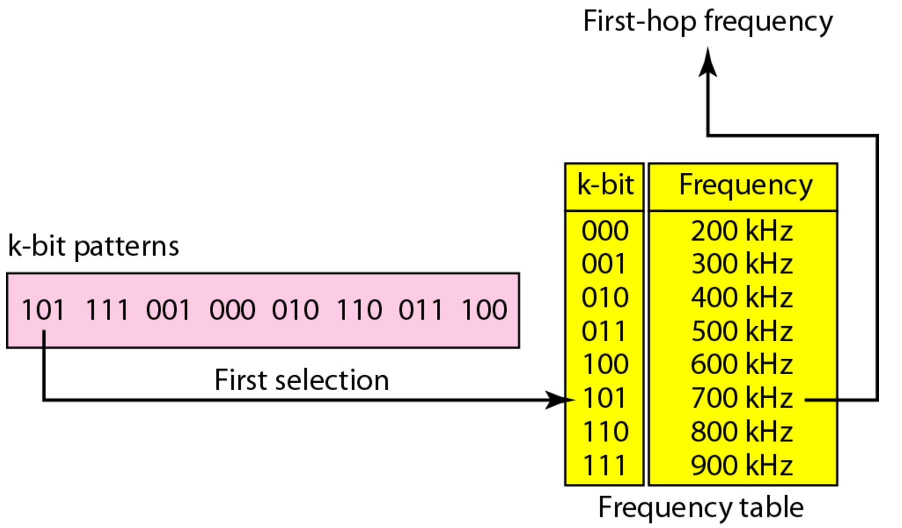
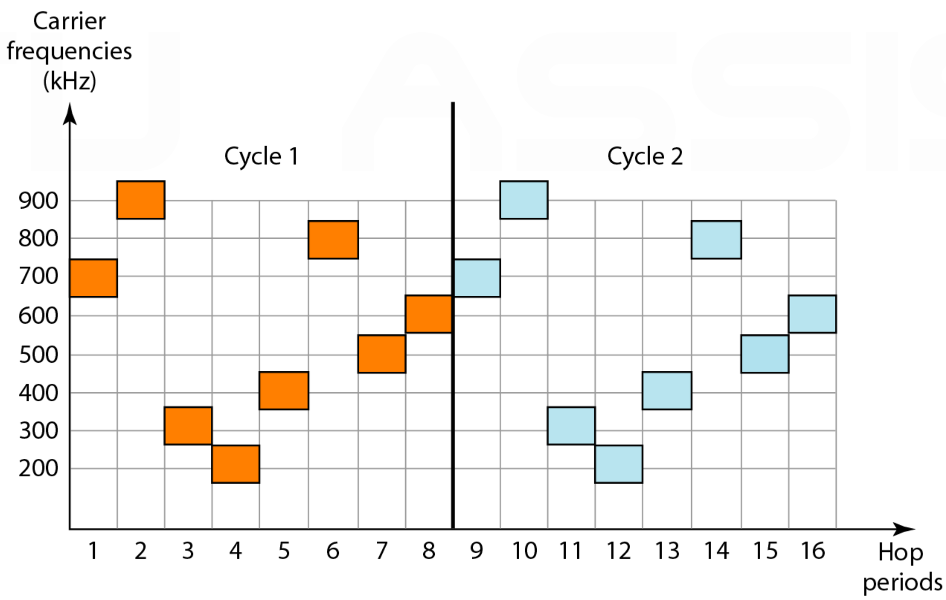


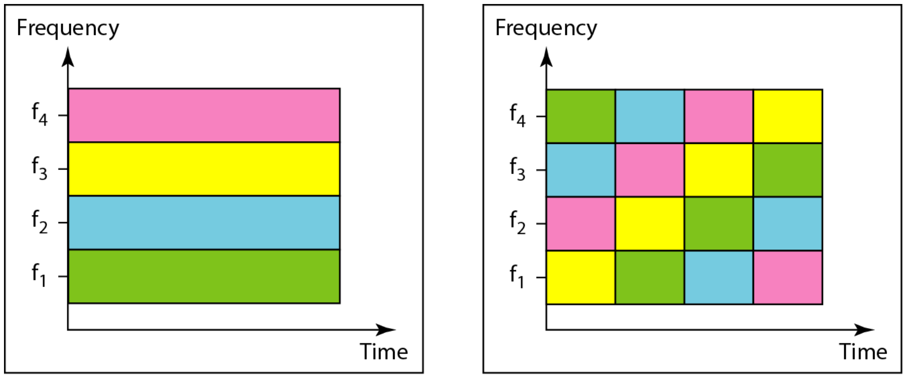
Figure below shows how the signal hops around from carrier to carrier. We assume the required bandwidth of the original signal is 100 kHz.



Advantages:

- Bandwidth sharing
- If there are many k-bit patterns and the hopping period is short, a sender and receiver can have privacy
- The scheme has also an antijamming effect. A malicious sender may be able to send noise to jam the signal for one hopping period (randomly), but not for the whole period.

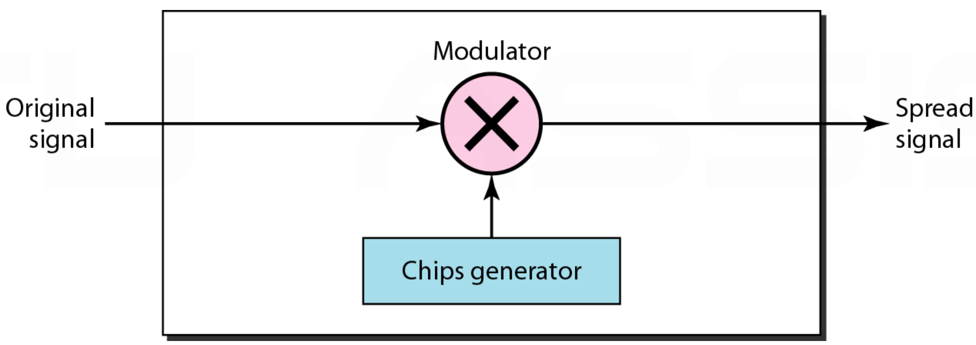
FHSS is similar to FDM, as shown in Figure below.



a. FDM b. FHSS

2)Direct Sequence Spread Spectrum

The direct sequence spread spectrum (DSSS) technique also expands the bandwidth of the original signal, but the process is different. In DSSS, we replace each data bit with *n* bits using a spreading code. In other words, each bit is assigned a code of *n* bits, called chips, where the chip rate is n times that of the data bit.

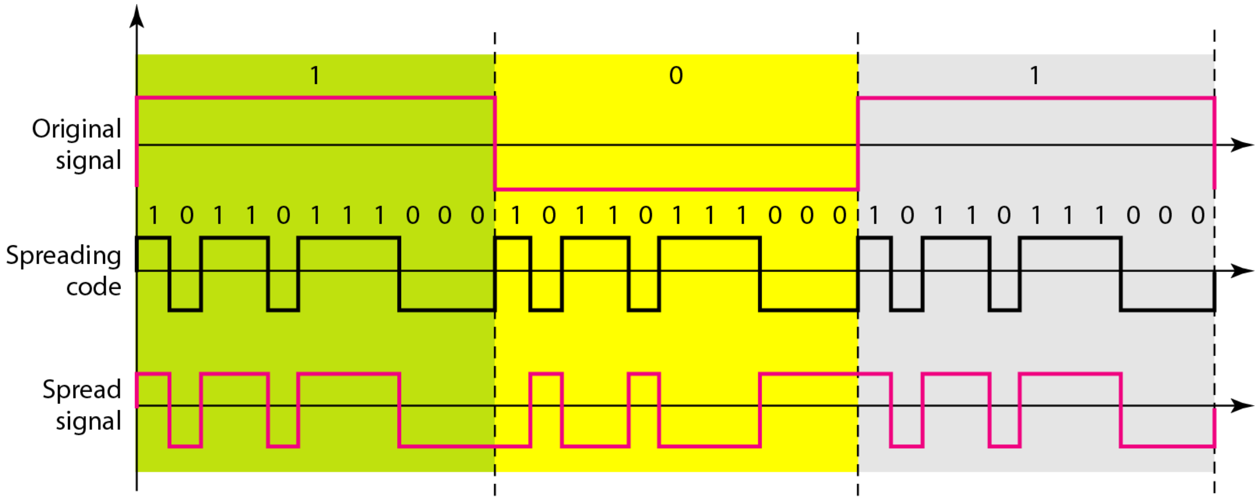


As an example consider the sequence used in a wireless LAN, the famous Barker sequence where *n* is 11. We assume that the original signal and the chips in the chip generator use polar NRZ encoding. Figure below shows the chips and the result of multiplying the original data by the chips to get the spread signal. In Figure, the spreading code is 11 chips having the pattern 10110111000 (in this case). If the original signal rate is *N*, the rate of the spread signal is 11*N*. This means that the required bandwidth for the spread signal is 11 times larger than the bandwidth of the original signal.

Input 1- Send original chip sequence

Input 0- Send complement of chip sequence

Example:

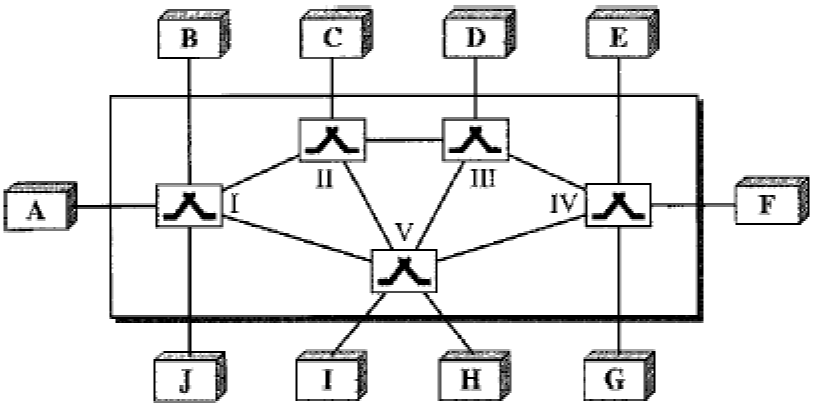


Advantages: The spread signal can provide privacy if the intruder does not know the code. It can also provide immunity against interference if each station uses a different code.

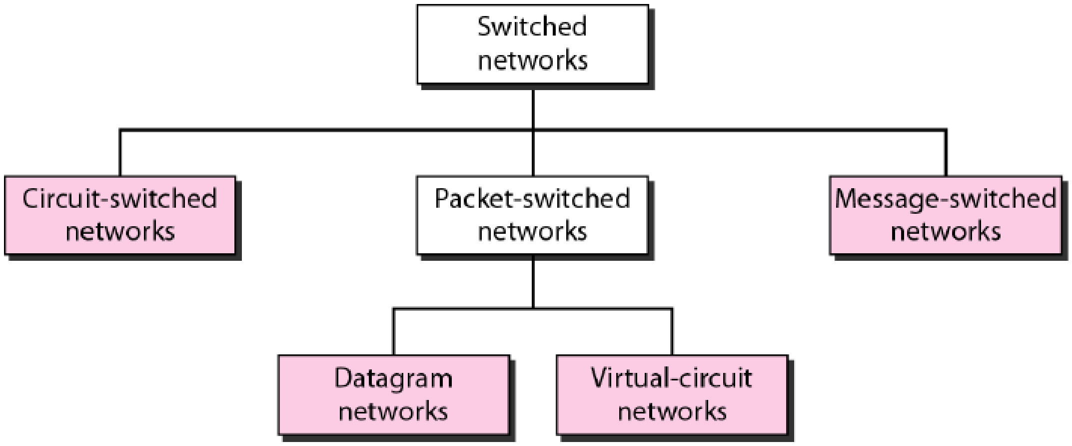
<u>Switching</u>

Switching is the technique to connect devices to make a one-to-one connection when we have multiple devices, using a switched network. The other methods like a point to point connection or using a central master slave connection are impractical and wasteful when using in a large network.

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 (computers or telephones, for example). Others are used only for routing.

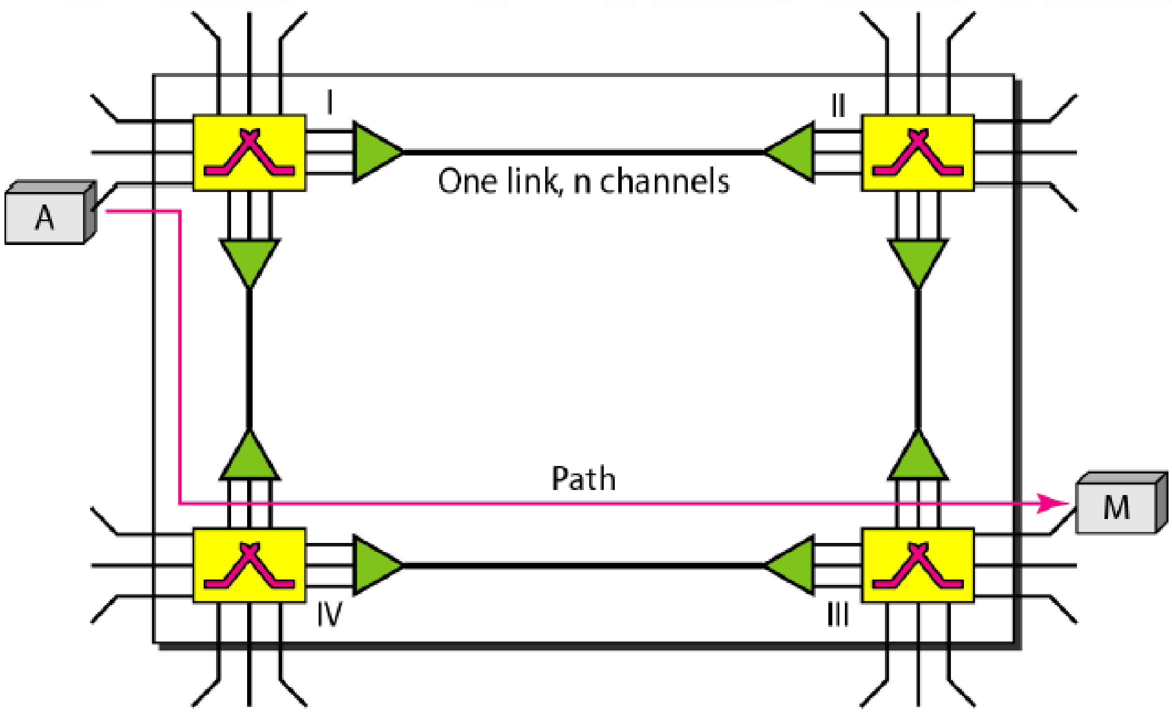


Types of switching



Circuit Switching

A circuit-switched network is made of a set of switches connected by physical links, in which each link is divided into *n* channels by using FDM or TDM. Figure 8.3 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.



Three Phases

The actual communication in a circuit-switched network requires three phases:

1.Setup Phase

2.Data Transfer hase

3.Tear down Phase

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 tom down.

Setup Phase

Before the two parties (or multiple parties in a conference call) can communicate, a dedicated circuit (combination of channels in links) 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.

Data Transfer Phase

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

Teardown Phase

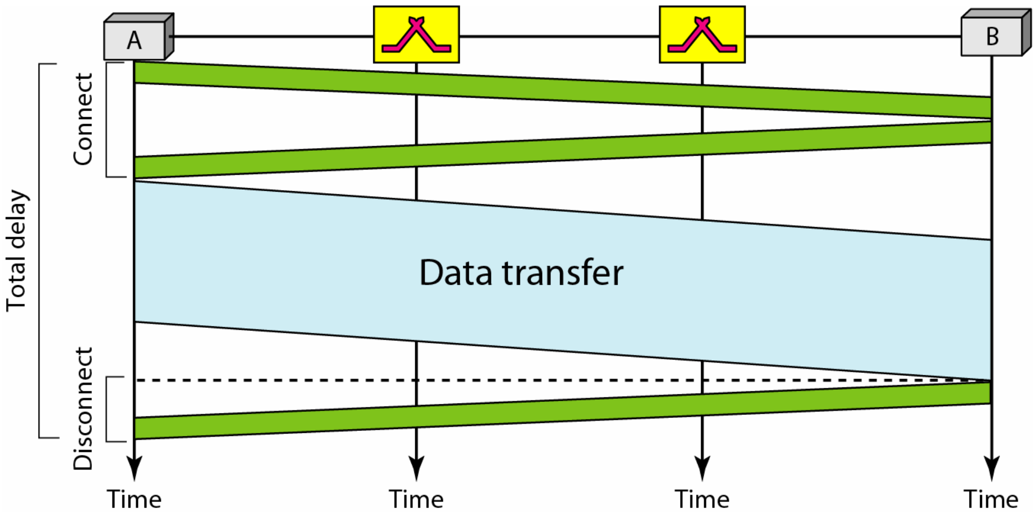
When one of the parties needs to disconnect, a signal is sent to each switch to release the resources.

Efficiency

Circuit-switched networks are not much efficient as resources are allocated during the entire duration of the connection. These resources are unavailable to other connections.

Delay

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. Figure below shows the idea of delay in a circuit-switched network when only two switches are involved.



Packet switching

In packet switching data are transmitted in discrete units called **packets**. The packet contains not only data but also a header with control information. The packets are sent over the network node to node.

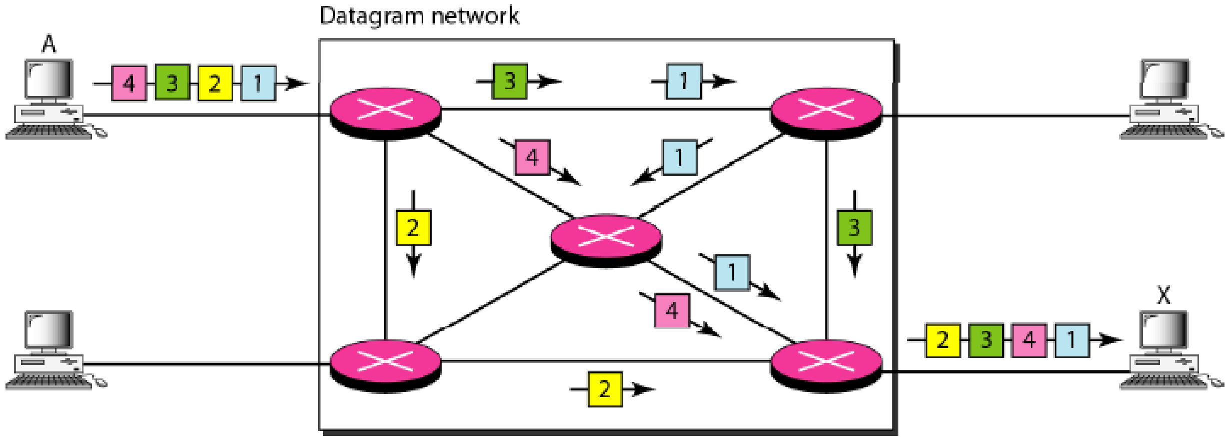
At the source ,the data is divided into **packets** using the packetizer. Then the address of the sender and the receiver is added. Then it makes sure that the packet is of proper size. If it is too large, it is fragmented in a fragmentation module. At any intermediate node, ie a switch is responsible for routing the packet. At the destination, it verifies the destination address of the packet. It also checks whether the packet is corrupted or not. If the packet is a fragment, then a reassembling is done at the destination.

There are two popular approaches for packet switching.

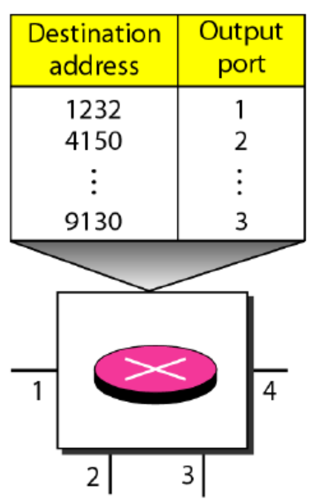
- Datagram approach
- Virtual circuit approach

Datagram approach

Here each packet is treated independently of others and each packet is routed independently. A packet in this approach is called **datagram**.



- Here 4 packets are sent from A to X. All the 4 packets are belonging to the same message but is taking different paths to reach the destination X. But here there is a chance for the packets to reach the destination out of order. It is responsibility of the upper layer to make it in the proper order.
- In datagram approach the decision of the routing is done at the packet switch only and no call setup is done at the beginning.So no resource allocation for a packet;it is done on demand.
- Each of the switches maintains a **routing table**. The destination addresses and the corresponding forwarding output ports are recorded in the tables.
- The routing tables are dynamic and are updated periodically.



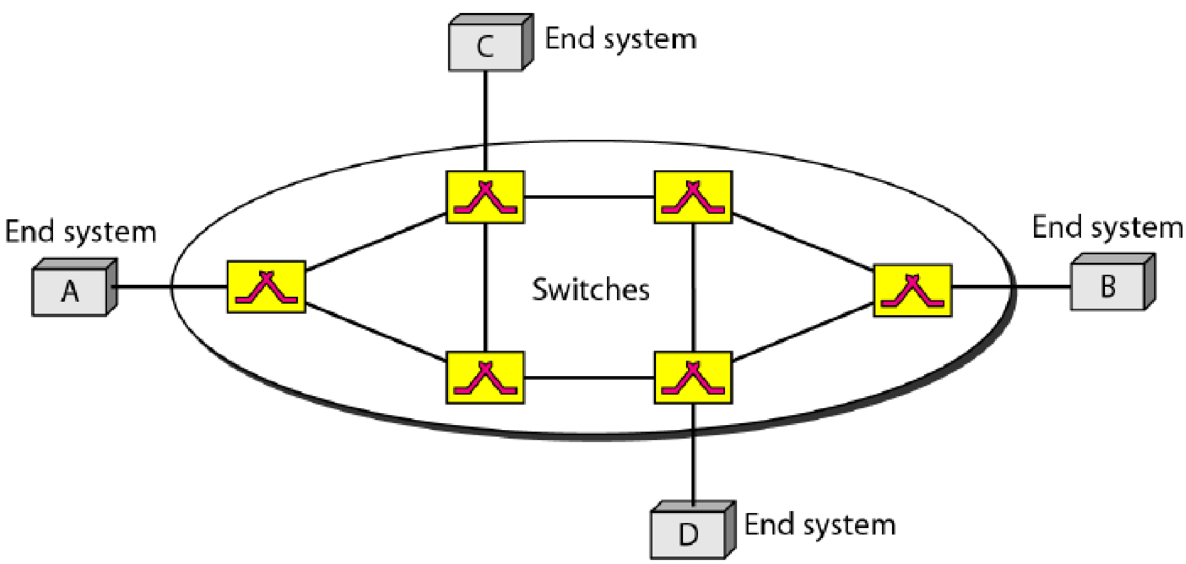
Routing Table in a datagram network

- Every packet in a datagram network carries a **header** that contains, among other information, the destination address of the packet and sequence number.

- Sequence number is used to reorder the packets at the destination.
- 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.
- Datagram approach has the advantage that it does not need any call setup at the beginning.
- The datagram networks are sometimes referred to as connectionless networks ie each packet is treated independent; there is no relationship between the packets of the same message while transmitting. Also no connection is established between the sender and receiver at the beginning.
- Packets that doesn't reach destination within a stipulated time are lost or dropped because of lack of resources.

Virtual circuit approach

In virtual circuit approach, the relationship between the packets of a message is preserved. All the packets of the same message are routed in the same path. For that a path is created at the beginning. Wide area network uses this approach.



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

- 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.
- Local address (Virtual-Circuit Identifier) is the identifier that is actually used for data transfer is called the virtual-circuit identifier .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. A VCI does not need to be a large number since each switch can use its own unique set of VCIs.
- The data communication using virtual circuit approach is accomplished through 3 phases. A **setup phase** to create the path, a **data transfer phase** where the data transmission is occurred. A **teardown phase** where the path created is deleted after the communication is completed.

Setup phase:

In the setup phase, a switch creates an entry for a virtual circuit. Two steps are required: the setup request and the acknowledgment. Each switch maintains a table, the switching table. It stores the details of the incoming and outgoing port and VCI for the packets. Its structure is :

Incoming		Outgoing	
Port	**VCI**	**Port**	**VCI**
1	14	3	22
2	33	3	16

Consider the first raw. Here the values indicate that, if a packet with VCI 14 is arriving at the 1st port, then it will be directed out thru the 3rd port with an outgoing VCI 22. This switching table is for that switch only. The outgoing VCI of this switch is the incoming VCI of the next switch connected thru that outgoing port.

Setup request :

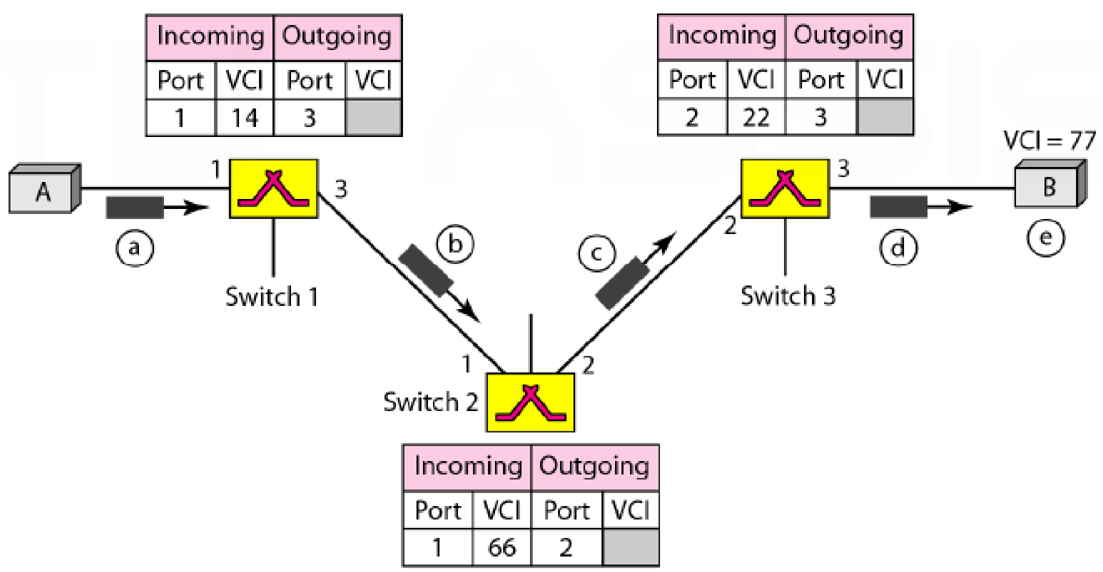
a. Source A sends a setup frame to switch 1.

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 to give information about the port. The switch creates an entry for the virtual ckt in the switching table. 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.



Setup request in a virtual-circuit network

***Acknowledgement* :**

A special frame, called the acknowledgment frame, completes the entries in the switching tables.

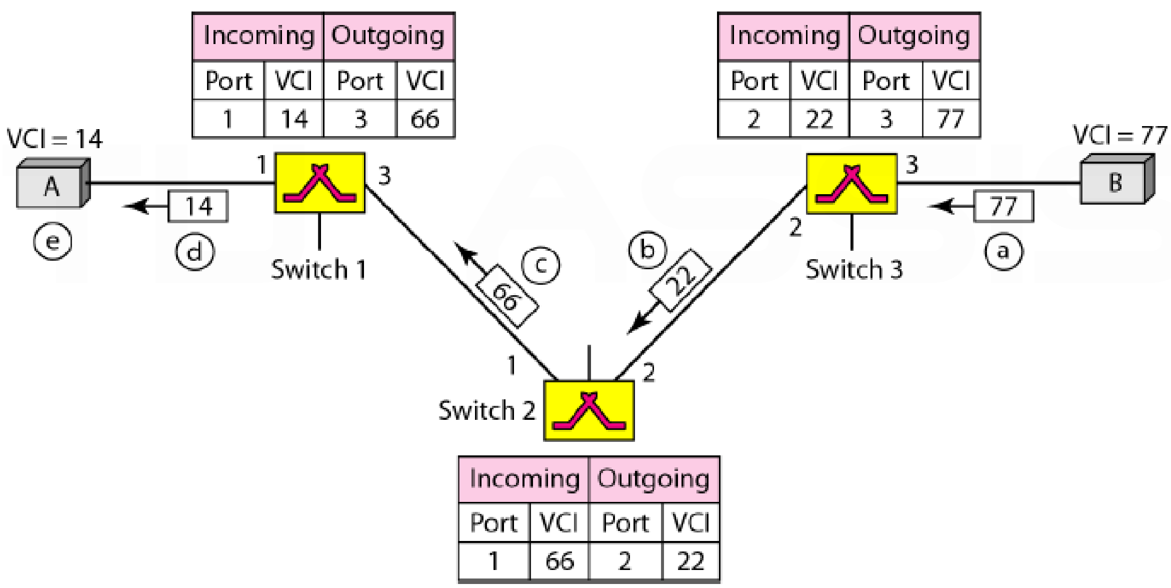
a. The destination sends an acknowledgment to switch 3. The acknowledgment carries the global source and destination addresses so the switch knows which entry in the table is to be completed. The frame also carries VCI 77, chosen by the destination as the incoming VCI for frames from A. Switch 3 uses this VCI to complete the outgoing VCI column for this entry. Note that 77 is the incoming VCI for destination B, but outgoing VCI for switch3.

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.



Setup acknowledgment in a virtual-circuit network

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. When a packet is arrived at a particular port with a particular VCI, the switch searches the switching table for the entry with that port and VCI and gets the outgoing port and outgoing VCI. Then the packet is assigned the new VCI and is sent out thru the outgoing port. 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.

Teardown phase :

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.

Application :

1) WAN (Virtual ckt approach)
2) Internet (datagram approach)

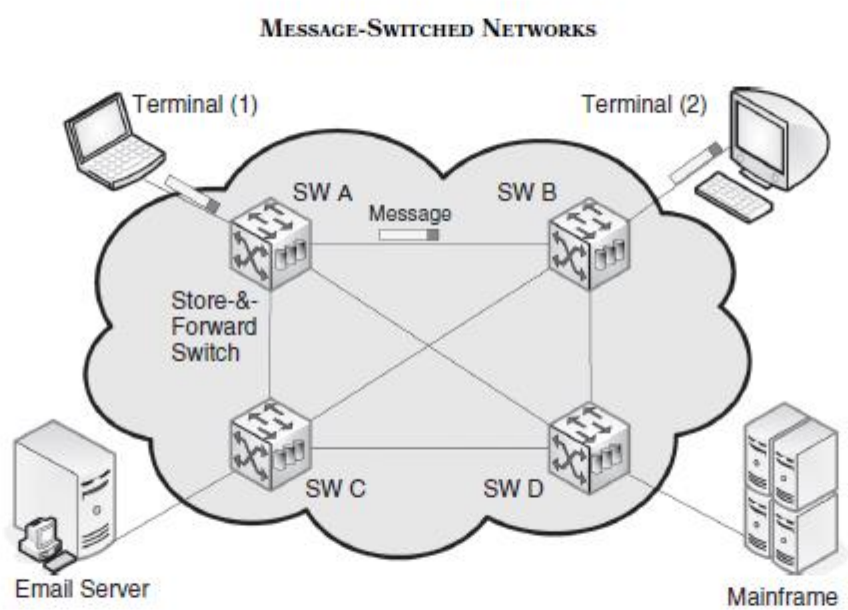
Internet chooses the connectionless approach of packet switching. It uses the universal addresses defined in the network layer to route packets from the source to the destination. The identifier used to uniquely identify the devices connected to the internet is internet protocol address or IP address. In the current version it is a 32 bit address. IP address is unique and if a system has 2 connections to the internet via 2 networks, it has 2 IP addresses.

Comparison of circuit and packet switching techniques

Circuit switching	Datagram packet switching	Virtual-circuit packet switching
Dedicated transmission path	No dedicated path	No dedicated path
Continuous transmission of data	Transmission of packets	Transmission of packets
Messages are not stored	Packets may be stored until delivered	Packets stored until delivered
The path is established for entire conversation	Route established for each packet	Route established for entire conversation

Call setup delay: negligible transmission delay	Packet transmission delay	Call setup delay; packet transmission delay
Busy signal if called party busy	Sender may be notified if packet not delivered	Sender notified of connection denial
Usually no speed or code conversion	Speed and code conversion	Speed and code conversion
Fixed bandwidth transmission	Dynamic use of bandwidth	Dynamic use of bandwidth
No overhead bits after call setup	Overhead bits in each packet	Overhead bits in each packet

Message switching



- Message switching is a network switching technique in which data is routed in its entirety from the source node to the destination node, one hope at a time.

- During message routing, every intermediate switch in the network stores the whole message. If the entire network's resources are engaged or the network becomes blocked, the message-switched network stores and delays the message until ample resources become available for effective transmission of the message. As message switching involves storing the data and forwarding subsequently it is also termed as **store and forward switching**.
- When this form of switching is used, no physical path is established in advance between sender and receiver. Instead, when the sender has a block of data to be sent, it is stored in the first switching office (i.e. router) then forwarded later at one hop at a time.
- Since message switching stores each message at intermediate nodes in its entirety before forwarding, messages experience an end to end delay which is dependent on the message length, and the number of intermediate nodes.
- In a message-switching centre an incoming message is not lost when the required outgoing route is busy. It is stored in a queue with any other messages for the same route and retransmitted when the required circuit becomes free.
- Every message should include a header, which typically consists of routing information, such as the source and destination, expiry time, priority level, etc.

The advantages to message switching are:

- Data channels are shared among communication devices, improving the use of bandwidth.
- Messages can be stored temporarily at message switches, when network congestion becomes a problem.
- Priorities may be used to manage network traffic.
- Broadcast addressing uses bandwidth more efficiently because messages are delivered to multiple destinations.

However, this technique also has several disadvantages:

- Because the messages are fully packaged and saved indefinitely at every intermediate node, the nodes demand substantial storage capacity.

- Message-switched networks are very slow as the processing takes place in each and every node, which may result in poor performance.
- This technique is not adequate for interactive and real-time processes, such as multimedia games and voice communication.

Application

- E Mail and telex transmission.

Structure of a switch

Structure of circuit switches

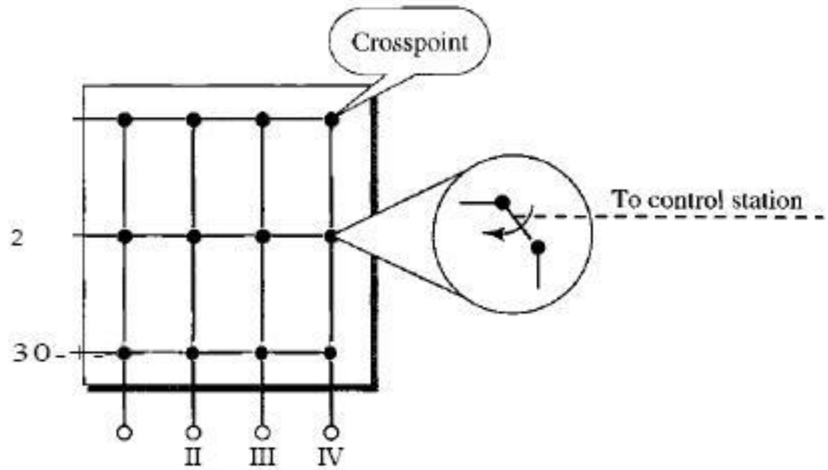
Circuit switching today can use either of two technologies: the space-division switch or the time-division switch.

Space division switch

In space-division switching, the paths in the circuit are separated from one another spatially. This technology was originally designed for use in analog networks but is used currently in both analog and digital networks. It has evolved through a long history of many designs.

Crossbar switch

A crossbar switch connects *n* inputs to *m* outputs in a grid, using electronic microswitches (transistors) at each crosspoint. The major limitation of this design is the number of crosspoints required. To connect *n* inputs to *m* outputs using a crossbar switch requires $n \times m$ crosspoints. For example, to connect 1000 inputs to 1000 outputs requires a switch with 1,000,000 crosspoints. A crossbar with this number of crosspoints is impractical. Such a switch is also inefficient because statistics show that, in practice, fewer than 25 percent of the crosspoints are in use at any given time. The rest are idle.

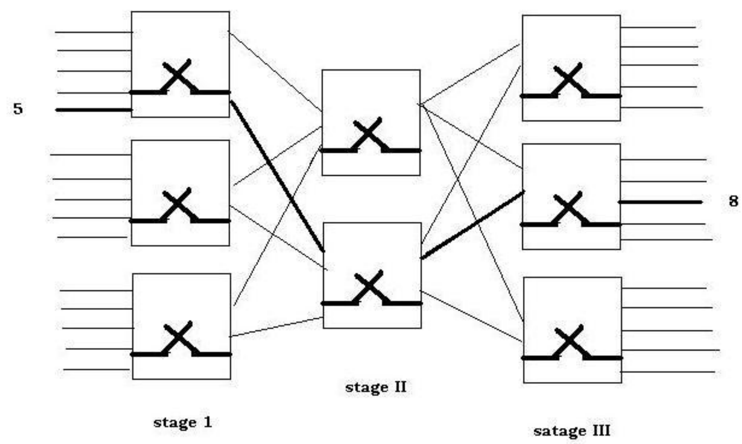


Multistage switch

The solution to the limitations of the crossbar switch is the multistage switch, which combines crossbar switches in several (normally three) stages. In a single crossbar switch, only one row or column (one path) is active for any connection. So we need N x N crosspoints. If we can allow multiple paths inside the switch, we can decrease the number of crosspoints. Each crosspoint in the middle stage can be accessed by multiple crosspoints in the first or third stage.

The design of multistage switch depends on number of stages and number of switches in each stage. Normally middle stage has fewer switches. In the below figure, there are 3 stages to connect between 15 inputs and 15 outputs.

Each of the first stage switches have an output to each of the intermediate switches ie 2 outputs. Each 3rd stage switches have inputs from each middle stage ie 2 inputs each. Each middle stages switches must have inputs from each 1st stage switches and output to each 3rd stage switches.



If we are using a crossbar switch to implement a 15X15 line, 225 crosspoints are needed. Here in multistage switch the crosspoints required are:

Three first stages each have to make 2 output from 5 inputs ie 10 (5 x 2) cross pints. So total 30 crosspoints.

Two middle stage switches each have to make 3 outputs from 3 inputs ie 9 (3 X 3) crosspoints. So total 18 crosspoints.

Three 3rd stage switches each have to make 5 outputs from 2 inputs ie 10 (2 X 5) crosspoints. So total 30 crosspoints.

So total number of crosspoints needed is 78. It means multistage switch reduces the number of crosspoints required.

Multistage switch provide several options for connecting each pair of linked devices. The above figure shows a path from input line 4 to output line 8 thru switches 1, 2, 2 of each stage respectively. Some other path is also possible say thru the switches 1, 1, 2 to reach line 8 from line 4.

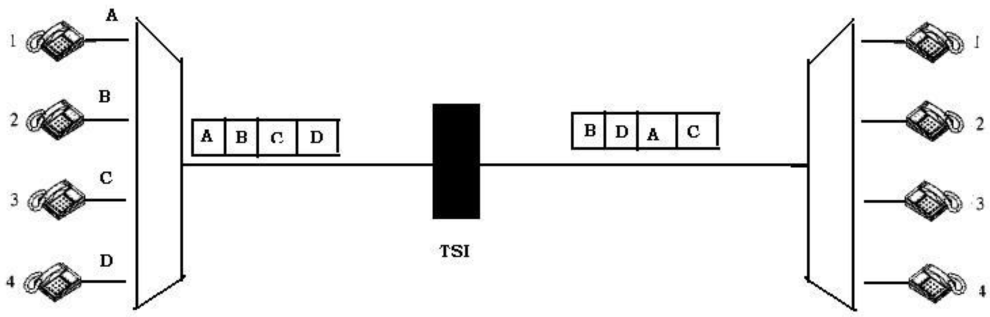
The design of multistage switch has a drawback that it results in blocking during heavy traffic. It is the condition when an input cannot be connected to output because no path is available at that moment. All the paths are occupied. In crossbar switch (single stage switch), blocking will not occur because for every pair of input and output devices a dedicated path is available. As the number of stages increases possible blocking increases.

Time division switch

Time-division switching uses time-division multiplexing (TDM) inside a switch. The most popular technology is called the time-slot interchange (TSI).

Time slot interchange

Here the common time division MUXing is employed. But here the data are output on the same order as they are sent. So a time slot interchange device is added between the sender and the receiver. It changes the order of the frames sent to the desired sequence.

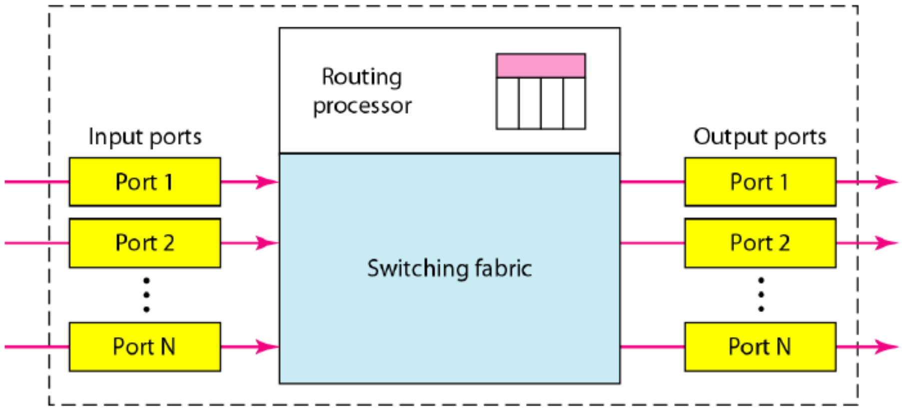


Here the TSI reorder the frames sent by 4 senders. If TSI was not there, A will go to terminal 1, B to terminal 2, and so on as in normal TDM. TSI is the device to make the proper order. It consists of a RAM and a control unit. The RAM fills up with the incoming data from time slots in the order received. But the frames are sent out in an order on the decision of the control unit.

A combination of the space division and time division switching also possible. It takes the advantages of both the systems. Advantage of space division switching is that it is instantaneous. But its large requirement of crosspoint is not a good feature. Time division switching need not any crosspoints, but the processing each connections creates delay. Multistage switches of this sort can be designed as time-space-time (TST), time-space-space-time (TSST) space-time-time-space (STTS) etc.

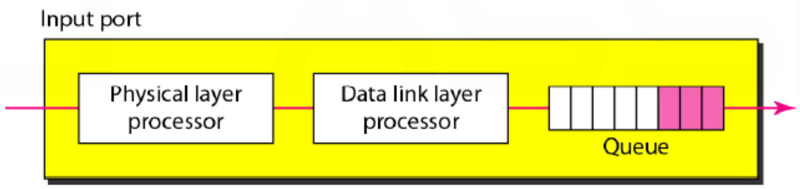
Structure of Packet Switches

A switch used in a packet-switched network has a different structure from a switch used in a circuit-switched network.We can say that a packet switch has four components: input ports, output ports, the routing processor, and the switching fabric, as shown in Figure.



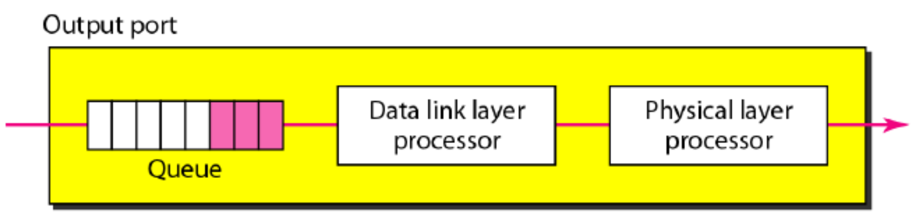
Input Ports

An input port performs the physical and data link functions of the packet switch. The bits are constructed from the received signal. The packet is decapsulated from the frame. Errors are detected and corrected. The packet is now ready to be routed by the network layer. In addition to a physical layer processor and a data link processor, the input port has buffers (queues) to hold the packet before it is directed to the switching fabric. Figure 8.22 shows a schematic diagram of an input port.



Output Port

The output port performs the same functions as the input port, but in the reverse order. First the outgoing packets are queued, then the packet is encapsulated in a frame, and finally the physical layer functions are applied to the frame to create the signal to be sent on the line. Figure below shows a schematic diagram of an output port.



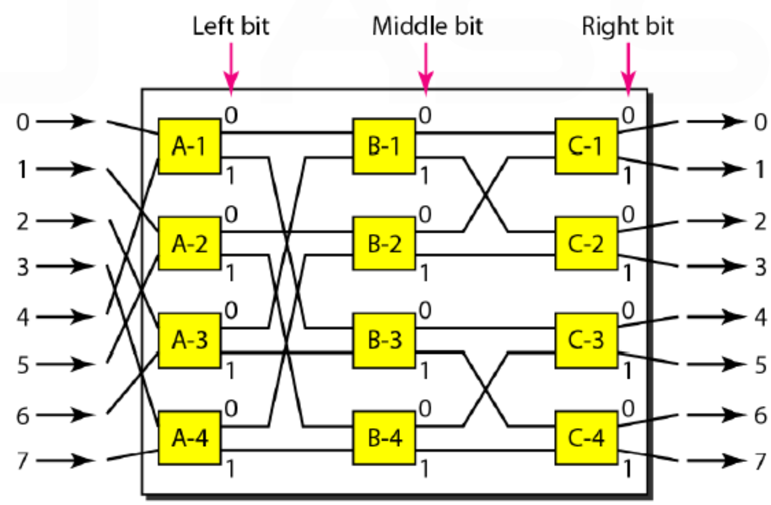
Routing Processor

The routing processor performs the functions of the network layer. The destination address is used to find the address of the next hop and, at the same time, the output port number from which the packet is sent out. This activity is sometimes referred to as **table lookup** because the routing processor searches the routing table.

Switching Fabrics

The most difficult task in a packet switch is to move the packet from the input queue to the output queue. The speed with which this is done affects the size of the input/output queue and the overall delay in packet delivery. In the past, when a packet switch was actually a dedicated computer, the memory of the computer or a bus was used as the switching fabric. The input port stored the packet in memory; the output port retrieved the packet from memory. Today, packet switches are specialized mechanisms that use a variety of switching fabrics. We briefly discuss some of these fabrics here.

- **Crossbar Switch** The simplest type of switching fabric is the crossbar switch.
- **Banyan Switch** A more realistic approach than the crossbar switch is the banyan switch (named after the banyan tree). A banyan switch is a multistage switch with microswitches at each stage that route the packets based on the output port represented as a binary string.



END