**Chapter 8: Switching**

Table of Contents

[8.1 Circuit-Switched Networks 3](#_Toc63188300)

[Three Phases of Circuit Switching 4](#_Toc63188301)

[Efficiency 4](#_Toc63188302)

[Delay 5](#_Toc63188303)

[8.2 Datagram Networks 6](#_Toc63188304)

[8.3 Virtual-Circuit Networks 8](#_Toc63188305)

[Virtual Circuit Identifiers 8](#_Toc63188306)

[Three Phases of Data Transfer 9](#_Toc63188307)

[Efficiency 11](#_Toc63188308)

[Delay 11](#_Toc63188309)

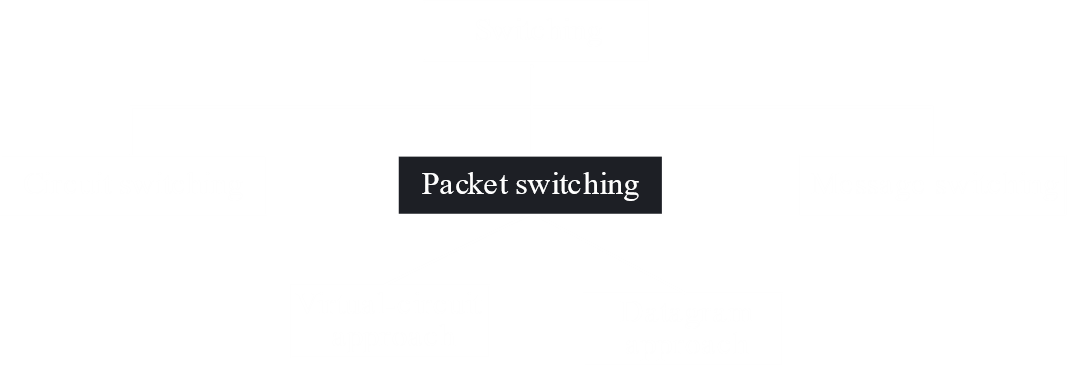
[8.4 Structure of a Switch 12](#_Toc63188310)

[Circuit Switching 12](#_Toc63188311)

[Packet Switching 18](#_Toc63188312)

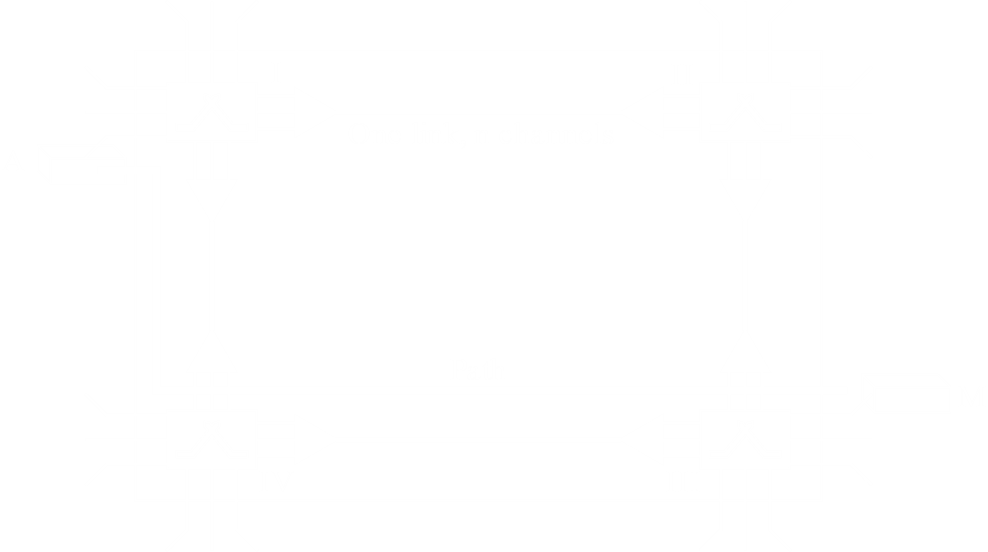
Switches are devices that are capable of creating temporary connections between two or more devices connected to the switch. Using the process of switching, data is forwarded from one place to another.

The taxonomy of switched networks looks like this:



## 8.1 Circuit-Switched Networks

Circuit switching works in the physical layer. It consists of a set of switches connected by physical links. When a connection is requested between two stations, a dedicated path is created using one or more links by physically switching the circuits using the switches.



As can be seen in the above diagram, a dedicated path is created between the two end devices. All the data packets follow this single path. Each connection only uses a dedicated channel on each link. The links are normally divided into multiple channels using FDM or TDM.

Circuit switching is fairly old and was heavily used with older, wired telephone connections. In those days, the end devices themselves were not sophisticated at all, so the core network had to be extremely sophisticated so as to be able to create the required connections. Nowadays the opposite is true, as we will see in the next section.

### Three Phases of Circuit Switching

There are mainly three phases to circuit switching:

1. Setup Phase – System needs to request a connection to system . This request must be accepted by all the switches as well as by system itself. The connection is made by reserving a channel on each link, the combination of which defines the dedicated path between the systems.
2. Data Transfer – After a dedicated path made of connected channels is established, data can be transferred. The existence of the dedicated path means that individual data packets do not need any headers, since they will just follow the path.
3. Teardown Phase – After all the data is transferred, the circuits are torn down.

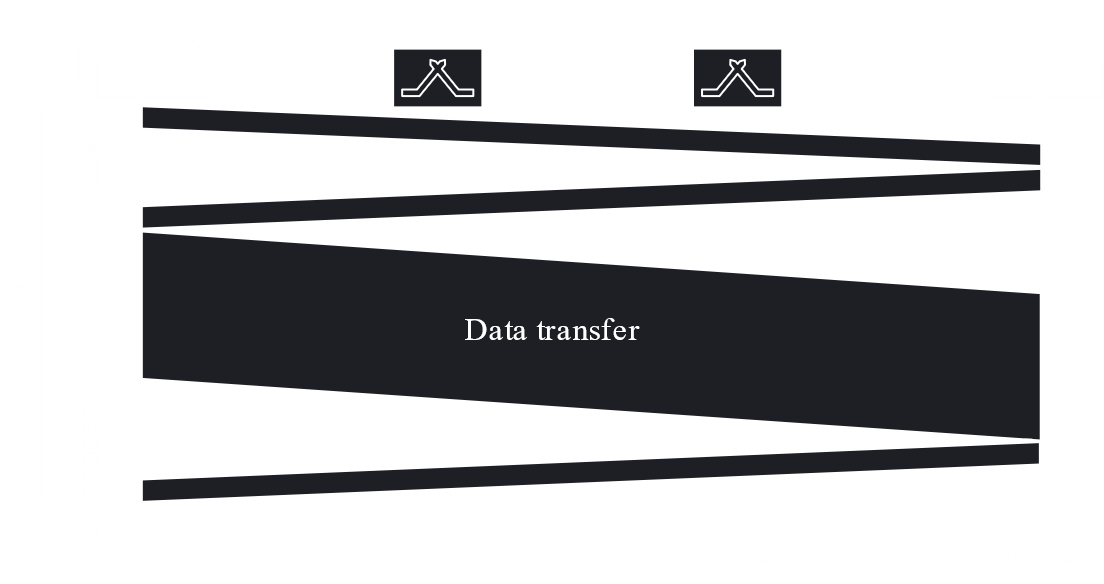
### Efficiency

Circuit switching is not very efficient, since resources (the channels on each link in this case) are allocated for the entire duration of the connection. These resources are then unavailable to other connections.

If data is only being transferred for a small portion of the time for which the connection is established, this will lead to very poor efficiency. As the number of users on the internet grew, this poor efficiency became a huge problem. Thus, even though the connection itself was great, the poor efficiency lead to circuit switching being abandoned in favour of datagram networks.

### Delay

The advantage to circuit switching is that there is very little delay. The only delay experienced is the initial delay where the connection is actually established. This also means it is extremely unlikely that there will be any jitter.



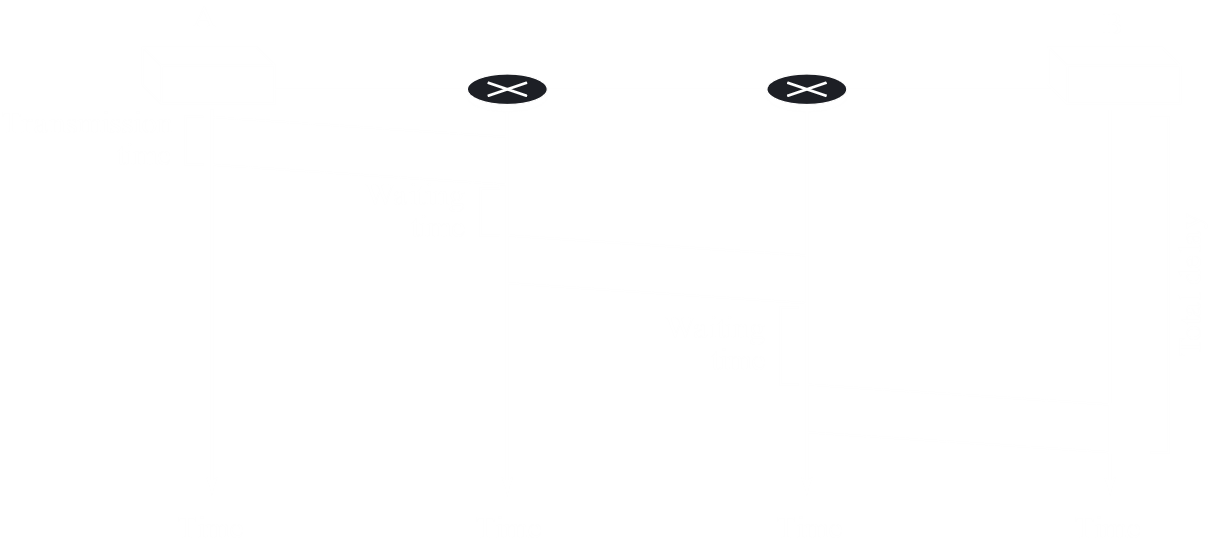
The figure above describes this. There is some delay at the beginning and the end while connecting and disconnected, but no separate delay during data transfer itself. This is different from other switching techniques.

## 8.2 Datagram Networks

In packet switching, there is no dedicated path as was available in circuit switching. Instead, the data is divided into packets and each packet is given a header file to allow identification of its source and destination. Resources to allow the transfer of packets are allocated to each individual packet only for the duration required to send the packet.

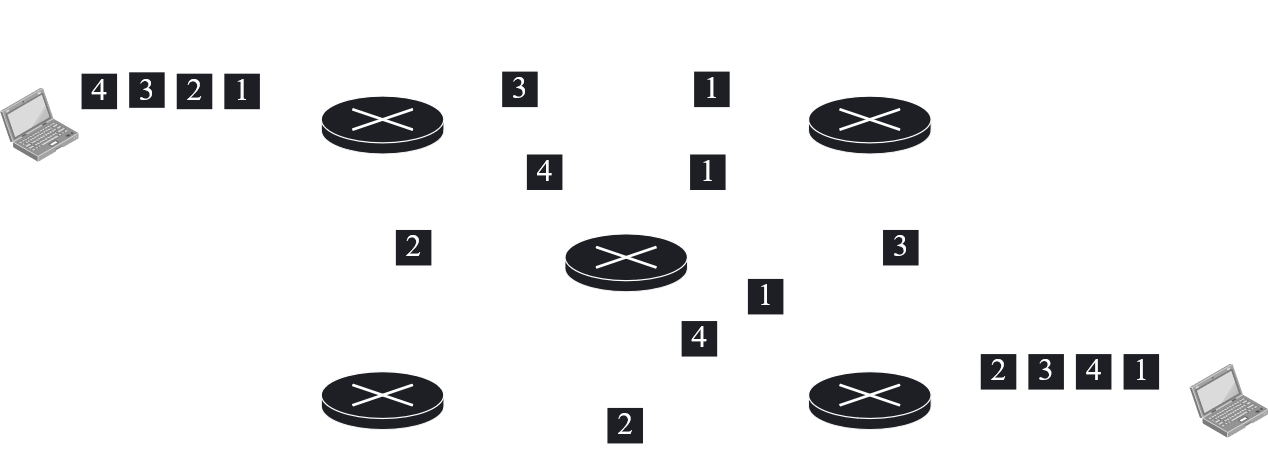
Datagram networks are a form of packet-switched networks. This is the approach we have seen over the previous couple of chapters, discussing things like the TCP/IP protocol suite, logical and physical addresses, routing tables, etc. Since routers deal with the actual paths followed by packets using routing tables, datagram networks work in the network layer.

This approach is much more efficient compared to circuit switching, especially with regard to the modern internet. However, the lack of a dedicated path also means each individual packet will experience some delay due to queues at each router, known as queuing delay, which causes a larger overall delay.



The packets could, in theory, follow different paths, since they each have headers, and even face different delays, which would cause jitter in real-time applications. This can also cause the packets to become out of order, but this is avoided using packet numbers. However, the packets normally try to follow the same path.

It could also be argued that the need for headers is a disadvantage, since they will require a little more bandwidth, but nowadays, this overhead is insignificant.



The datagram approach is also called the best effort approach. This is because the lack of connection means some data packets may be lost. The network tries its best to deliver all the packets, but delivery is not guaranteed.

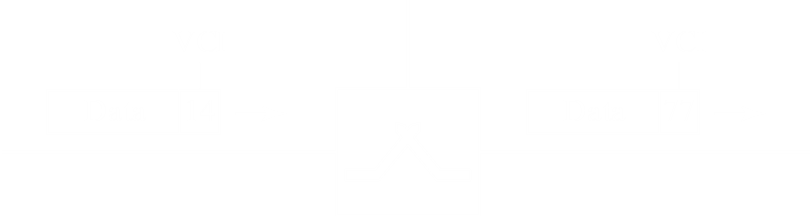
## 8.3 Virtual-Circuit Networks

Virtual circuits networks are also classified as packet-switched networks. They work in the data-link layer. They are a compromise between circuit switching and datagram networks. A dedicated connection is setup between end devices, meaning the setup phase, data transfer phase and teardown phase all exist, but this connection is not reserved for the duration in which no data is being sent. When it is not being used, the connection may be used elsewhere. Because of these advantages, the modern internet is trying to switch to virtual-circuit networks.

### Virtual Circuit Identifiers

Data is still packetized in virtual circuit switching, with each packet having a header file. However, the address in these headers are not global addresses like IP addresses. Instead, a much more local address, called a label, is used which only works inside the established connection and is recognizable only by the different hops.

A global address is still required though, and this is used to create the virtual circuit identifier, or VCI. The VCI is a small number that allows switches to identify which direction to send the data in. The switch does this using a table. Every switch can have a unique set of VCIs.

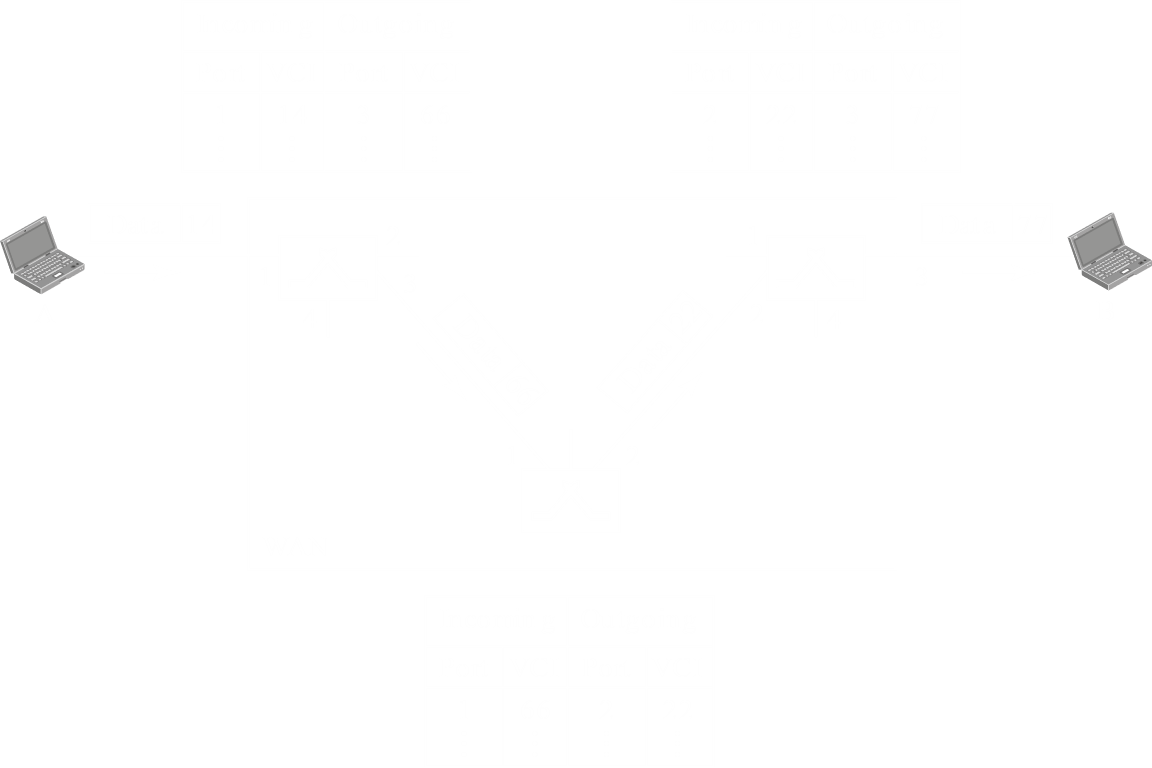


### Three Phases of Data Transfer

Virtual circuit networks have the same three phases as circuit switching, setup, data transfer and teardown. In the setup phase, the source and destination global addresses are used to let switches make table entries. We will look into more details of this later. In the teardown phase, the source and destination inform the switches to erase the table entries.

#### Data Transfer

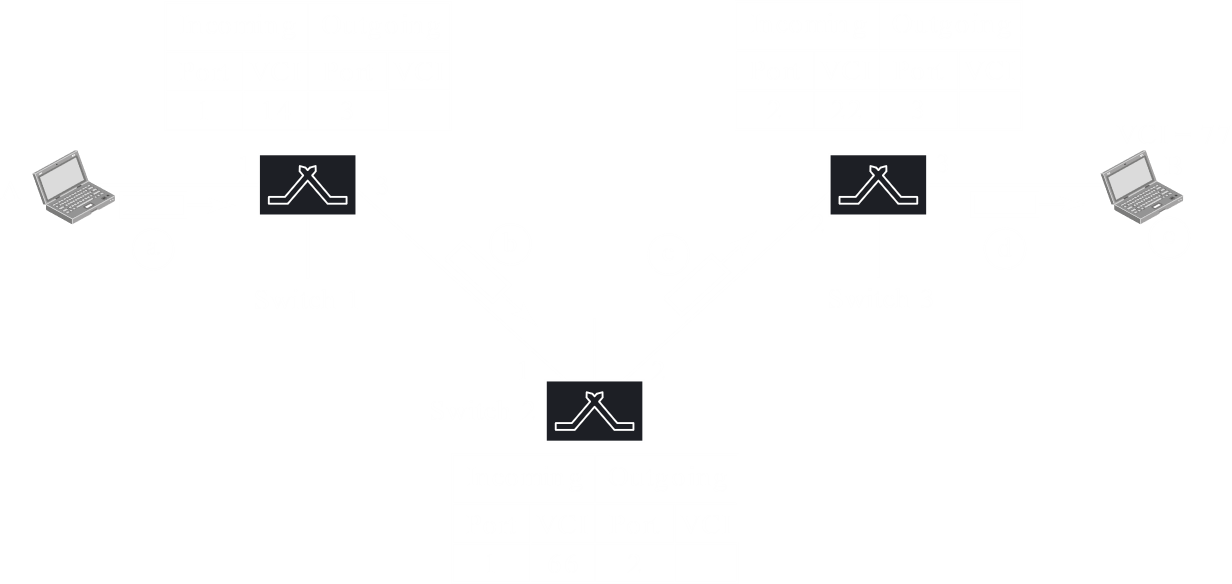
In the data transfer phase, each frame has a VCI, as discussed earlier. The VCI, along with the port at which the frame is incoming, allows the switch to identify which port it should be outgoing from. The VCI for the frame is also changed. Thus, every frame from a single source will have the same VCIs and will follow the same path.



The VCIs are chosen by the devices. For example, when device A wants to send data to device B, the first switch tells it to use the label 14. The second switch in turn tells the first switch that data from this device should use the label 66 and so on. Finally, device B allows data from device A to come to it through the last switch using the label 77.

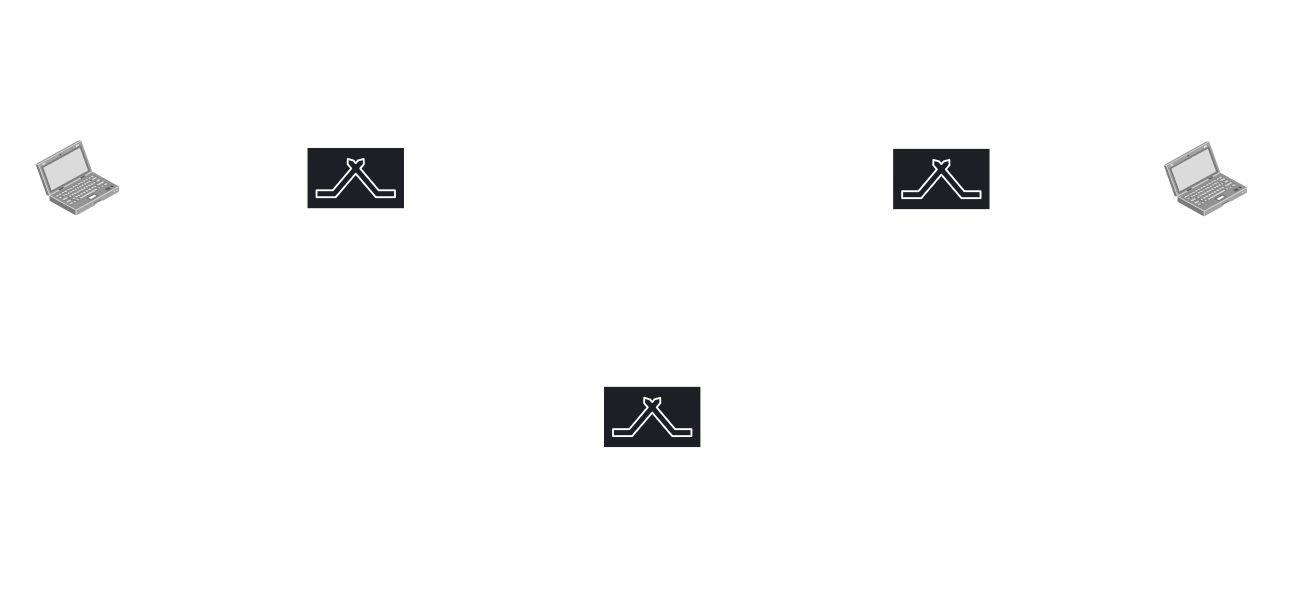
#### Setup

The first step to setting up the connection is to send a setup request.



When the request is sent, the first switch sees that data from A is using port 1. It can assign a VCI label to data from A, which it does. It forwards the data to the second switch through port 2. However, it does not yet know what label to use.

This same pattern continues until the request reaches device B, which then becomes aware that device A wants to communicate. B thus needs to send an acknowledgement. This is done by telling the third switch what label to use with frames that are coming from A. This setup acknowledgement informs each of the other switches what labels to use in turn, and finally reaches A, which is informed by the first switch what label to use with its frames.



#### Teardown

When A has finished sending all its frames, it will send a special frame called a teardown request. B in turn responds with a teardown confirmation. Once this is done, all the switches erase the corresponding entries from their tables.

### Efficiency

In virtual-circuit switching, all the packets travel the same path. Thus, there are no out of order packets. However, the packets may still arrive with different delays if resource allocation is on demand.

### Delay

There is a one-time delay for setup and teardown, however, assuming that resources are allocated during the setup phase, there is no queueing time.

## 8.4 Structure of a Switch

### Circuit Switching

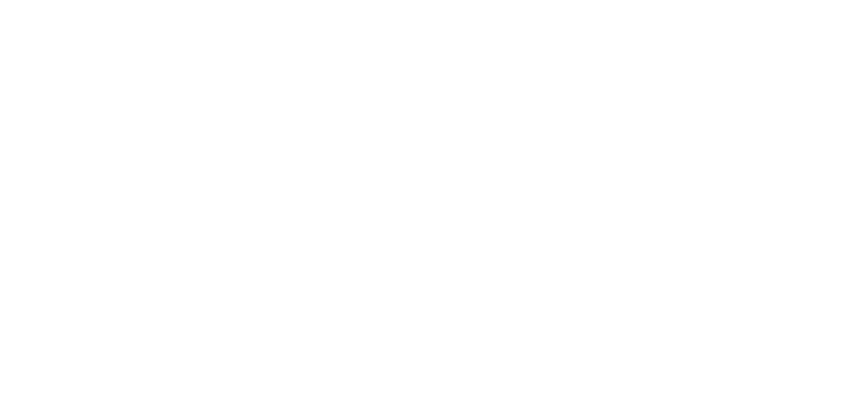
Circuit switching uses two technologies, space-division switching and time-division switching.

#### Space Division Switching

In space division switching, the paths in the circuit are separated spatially.

Crossbar Switch

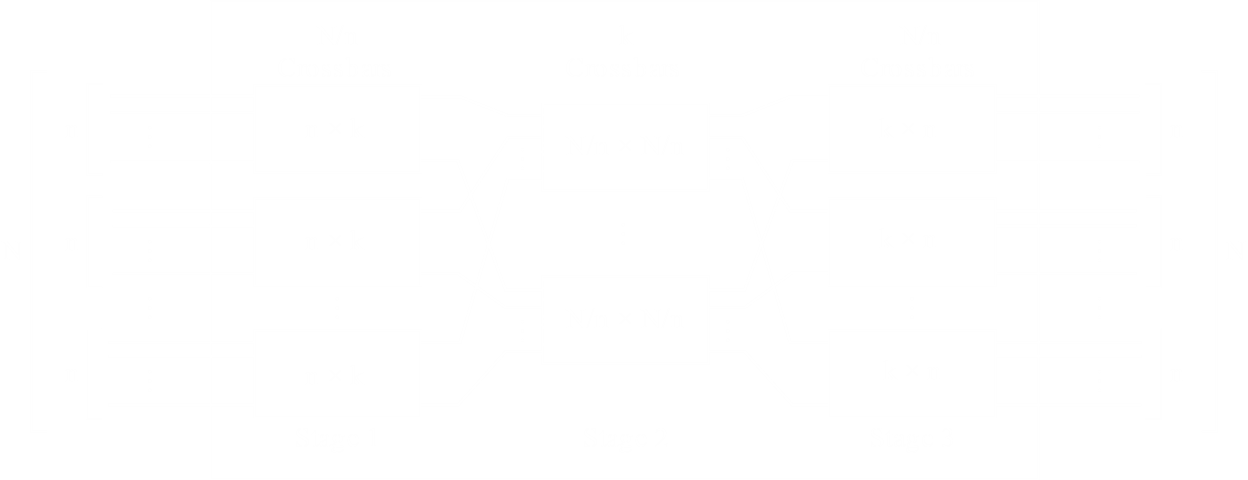
A crossbar switch connects inputs to outputs using a grid with crosspoints.



Each crosspoint is controlled using a separate transistor. Thus, the limitation is that a huge number of crosspoints becomes impractical. Also, statistically, fewer than of the cross-points are in use at any time, while the rest remain idle. This is extremely inefficient.

Multistage Switch

Multistage switches solve the limitations of crossbar switches by combining the crossbar switches in several stages, normally 3.



In a single crossbar switch, only one row or column can be active for any connection. Thus, we need crosspoints. If we could allow multiple paths, we would need fewer crosspoints. Each crosspoint in the middle stage can be accessed by multiple crosspoints in the first and third stages.

To design a three-stage switch, there are three steps:

1. Divide input lines into groups of lines. Thus, each group needs a crossbar of size .
2. Use crossbars of size each in the middle stage.
3. Use crossbars, each of size in the third stage.

Thus, the total number of cross points is:

This value is much smaller than , the number of crosspoints we would need in a single stage switch. If we have extremely large numbers of inputs and outputs, we can increase the number of stages further to cut down the number of crosspoints even more.

One major problem with the multistage switch is blocking. Since the middle stage is shared, heavy traffic can cause some inputs to not be able to find any path to their outputs, since the intermediate switches are occupied. The more stages we have, the higher the chances of blocking. Such problems do not occur with single stage crossbars.

Keep in mind that we are not talking about two inputs trying to connect to the same output, since that means the output is just busy. Blocking refers to no points being available to form a path at all.

Blocking causes problems like people being unable to connect through public telephone systems during crises. The system is experiencing far higher loads than normal, which causes the issue.

A person named Clos discovered that blocking can be avoided if we have at least middle-stage switches. Thus, .

Using this information, we want to minimize the number of crosspoints for a fixed value of . Since is the only variable, we can take the derivative of the equation with respect to . Thus must be greater than or equal to . Thus, the total number of crosspoints is greater than or equal to . Thus, according to the Clos criterion, the minimum number of crosspoints is .

Example

Design a three stage switch with and .

In the first stage, we need crossbars, each of size .

In the second stage, we have crossbars, each of size .

In the third stage we have crossbars, each of size .

Thus, there are crosspoints. With a single stage, we would have had crosspoints.

If we wish to follow the Clos criterion,

First Stage: crossbars, each with crosspoints

Second Stage: crossbars, each with crosspoints

Third Stage: crossbars, each with crosspoints

Thus, there are crosspoints, compared to the crosspoints needed in a single stage switch.

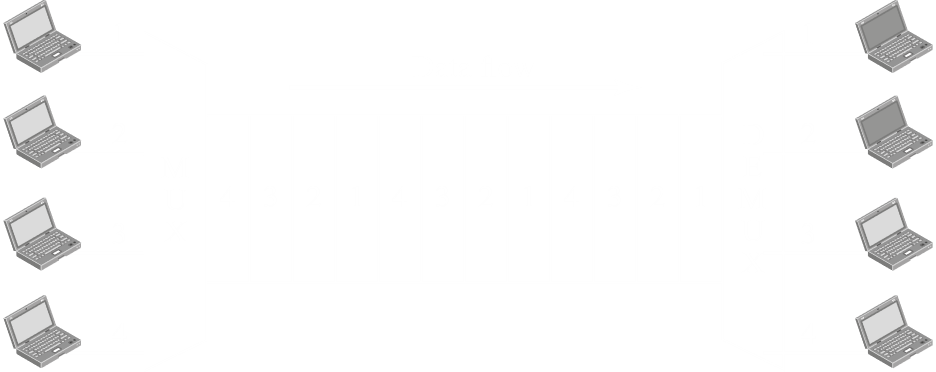
#### Time Division Switching

Today’s telephone companies use time-division switching, or at least a combination of space- and time-division switching. Time division switching uses time-division multiplexing (TDM) inside a switch. The most popular technology that does this is called time-slot interchange (TSI).

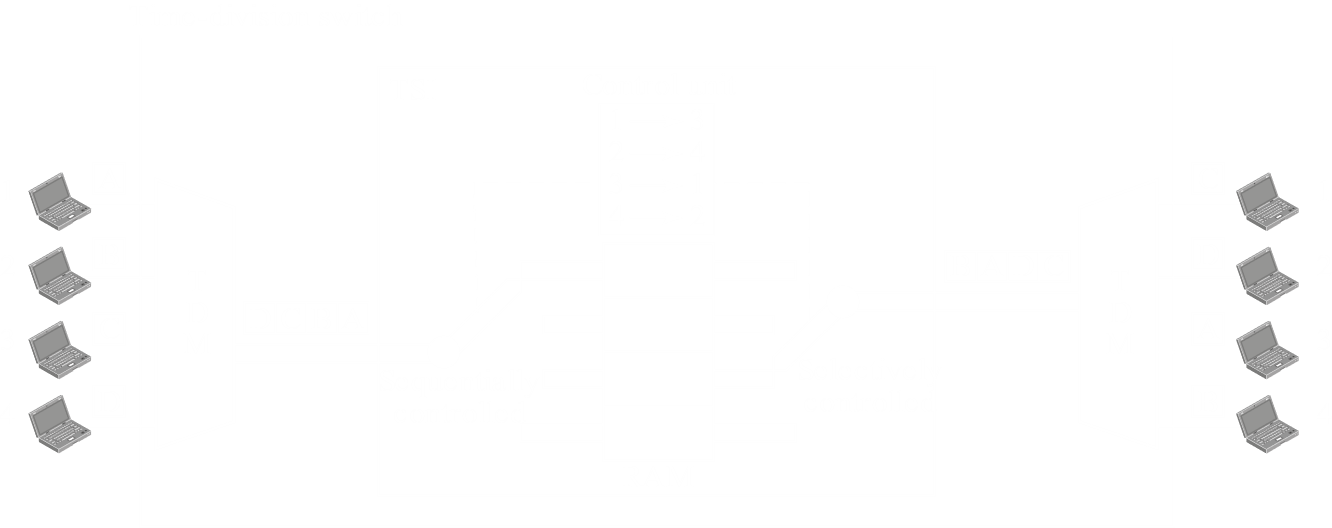
Time Slot Interchange

A time division switch uses a combination of a TDM multiplexer, a TDM demultiplexer and TSI consisting of RAM with several memory locations.

With just the TDM multiplexer and demultiplexer, we can divide the incoming data into time slots on a single line.



If we add a TSI in between this, the incoming data will fill up the RAM, and a control unit will make decisions on which data to send out in what order.



Notice how the arm on the left of the TSI moves sequentially, since the incoming data from the TDM multiplexer is sequential. Once the data is in the RAM, the arm on the right moves selectively, based on the decision of the control unit.

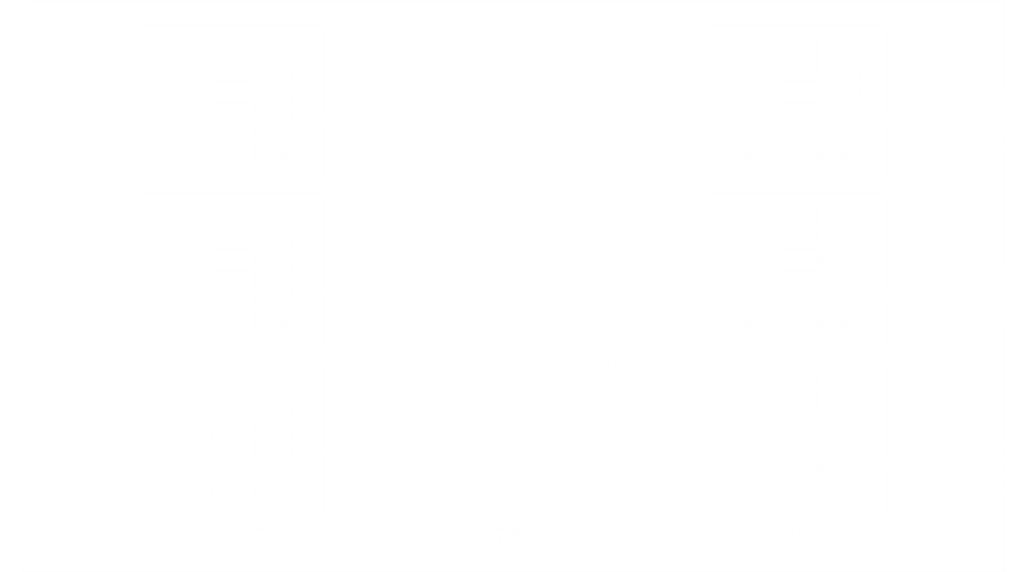
#### Space and Time Division Switch Combinations

If we compare space division switches and time division switches, we will notice that the prior has the advantage of being instantaneous, but is forced to deal with a huge number of crosspoints and potentially, blocking issues, while the latter has no crosspoints at all but transmission is not instantaneous, since each connection must face some processing.

Combining the two technologies can help us take the advantages of both. Such combinations can follow one of several patterns

* Time-Space-Time
* Time-Space-Space-Time
* Space-Time-Time-Space

For example, consider the Time-Space-Time (TST) configuration.



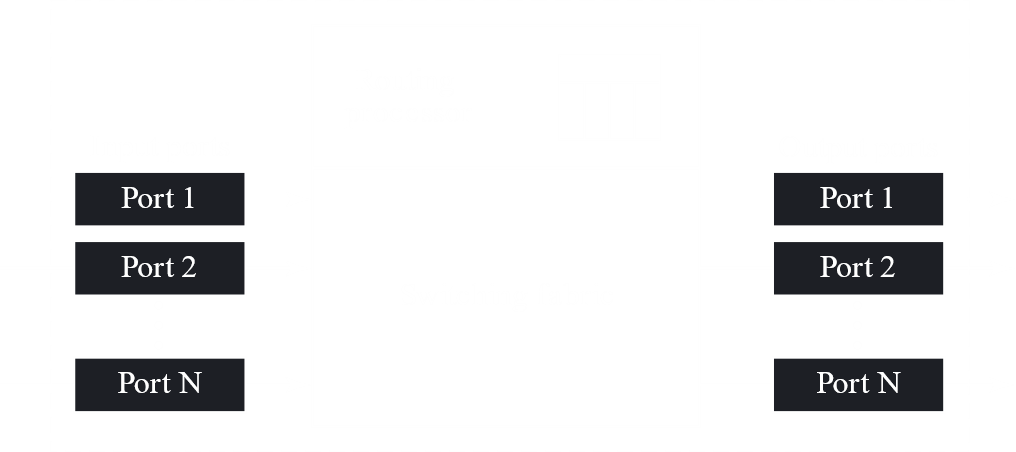
This particular configuration uses a total of 12 inputs divided into 3 groups, with each group going to a time division switch. The three outputs of the switches are instantaneously transmitted using a space division switch. Finally, another set of time division switches process the incoming data and sends them to the outputs accordingly.

The result of this configuration is a reduction in the average delay to one-thirds of what it would have been had a single time-slot interchange been used.

### Packet Switching

Packet switches have four components:

1. Input Ports
2. Output Ports
3. Routing Processor
4. Switching Fabric

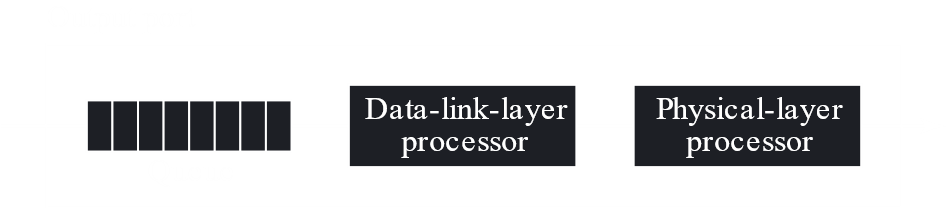


The input ports perform physical and data-link functions. The bits from the incoming signals are constructed, the packets are decapsulated and errors are detected and corrected. At the end of this, the packets are ready to be routed by the network layer.

Additionally, there is a buffer to hold the packets before they are directed to the switching fabric.



The output ports perform the same functions as the input ports, but in the opposite order. Packets are placed on the buffer, encapsulated in frames, and finally the signal is created that will be sent out.



The routing process performs the functions of the network layer. The destination address is used to find the address of the next hop and the output port. This process is also called table lookup, since the routing table needs to be searched. In newer versions of packet switches, this function is being moved to the input ports to make the process faster.

#### Switching Fabrics

The most difficult task is that of the switching fabric. It moves packets from the input queues to the output queues. The speed with which this is done affects the buffer sizes as well as the overall delay in packet delivery.

In the past, packet switches were actually dedicated computers. The input ports stored packets on the memory, and the output ports retrieved it from the memory. Thus, the memory or the bus was the switching fabric.

Nowadays, packet switches are specialized mechanisms that use a variety of switching fabrics.

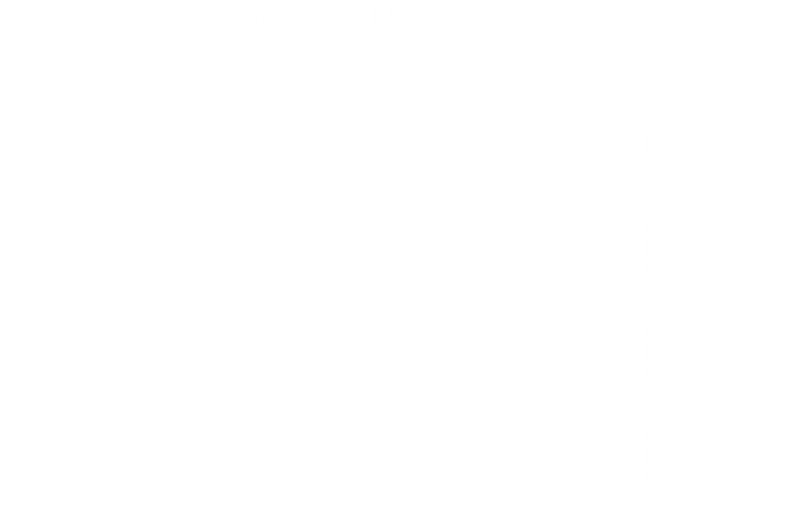
Crossbar Switch

This is the same as the crossbar switches discussed with circuit switching.

Banyan Switch

A banyan switch is more realistic than a crossbar switch. It is a multistage switch with microswitches at each stage that route the packets based on the output port represented as a binary string. For inputs and outputs, there will be stages with microswitches at each stage.

For example, consider the banyan switch below. There are inputs and outputs, so there are stages.

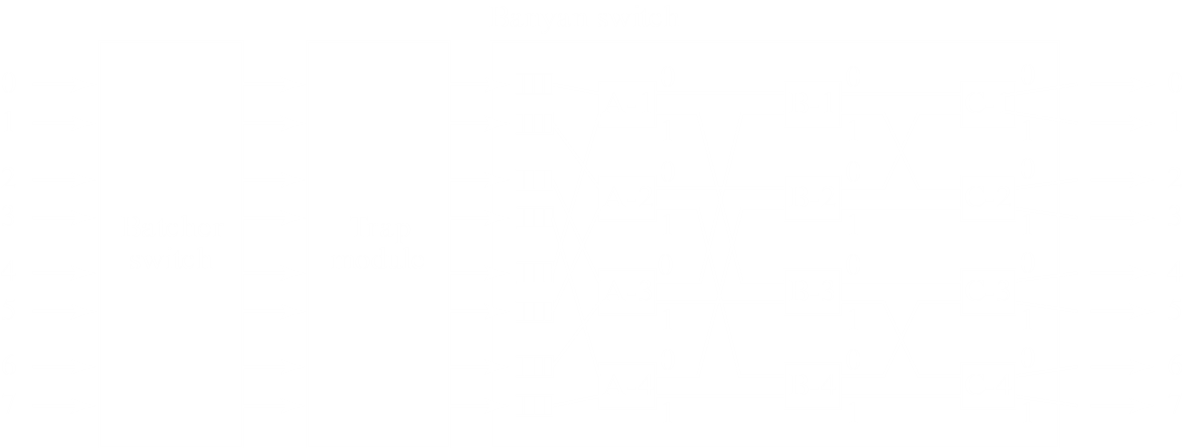


The first stage routes the packet based on the first bit, the second stage based on the second bit and the third stage based on the third bit.

As an example, consider input 1 being sent to output 6 (110). Input 1 would go to A-2. The first bit of the output is 1, so A-2 would send it to B-4. The second bit is also 1 so B-4 would send it to C-4. Finally, the third bit is 0, so C-4 sends it to output 6.

Batcher Banyan Switch

The problem with the Banyan switch is that inputs could collide internally, even when two packets are not headed for the same output port. This problem can be solved by sorting the arriving packets based on their destination port. This is done in the Batcher Banyan switch. The actual sorting is done by the batcher switch, which comes before the banyan switch.



Additionally, there is another module called a ‘trap’ in between the batcher switch and the banyan switch. This prevents inputs that are headed for the same output from being sent to the banyan switch simultaneously. If there are multiple such inputs, they are sent one after another.