ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

NEUMANN JÁNOS FACULTY OF INFORMATICS

THESIS

OE-NIK
2022

Student's name: **Saif Shalaan Abdulrazzaq Alkhniab**
Student's registration number: **T/008784/FI12904/N**

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

**NEUMANN JÁNOS
FACULTY OF
INFORMATICS**

Reducing the time response of computer networks for control instructions

Óbuda University
John von Neumann Faculty of Informatics
Institute for Cyber-physical Systems

DIPLOMA THESIS
TOPIC DESCRIPTION

Student name: **Saif Shalaan Abdulrazzaq Alkhniab**
Registration number: T/008784/FI12904/N
Neptun's code: [redacted]

Title of diploma thesis:

Reducing the time response of computer networks for control instructions
Számítógépes hálózatok vezérlési utasításainak válaszidejének csökkentése

Internal supervisor: Péter Fésüs
External supervisor

Deadline: 15th of December, 2022.

Topic description:

Redesign and implement a computer network protocol, and the redesigned one should be faster in relation and higher response to the control instructions through the network by limit the traffic effect on important control instructions such as medical robots control signals, remote controlled vehicles, alarm systems. This solution should apply only in overload cases not all the time, otherwise the network should work normally.
The solution should be high reliable in case that one route or more has been dropped down. The diploma thesis should be based on ideas of stander network protocols such as TCP IP, UDP, RIP and others. The solution should edit the communication protocols and control systems. The solution might be software or hardware based or both.

The diploma thesis should cover:

- the precise description and detailed analysis of the problem to be solved,
- possible implementation methods and solutions,
- the detailed system plan and its justification,
- presentation of the technologies used in the implementation,
- a description of the implementation process, work phases and experiences,
- test methods, test plans and test results,
- the application and further development possibilities of the completed system,
- The source code of the developed system and the prepared documentation on a CD or DVD attachment.

Ph.

.....
Dr. Rita Fleiner
head of institute

Term of limitation: **15th of December, 2024.**

The diploma thesis is adequate and can be submitted:

.....
external supervisor

.....
internal supervisor

STUDENT'S DECLARATION

I the undersigned student hereby declare that this thesis is the result of my own work; I have disclosed the references and tools used in an identifiable manner.
The results indicated in my thesis completed may be used by the university and the institution announcing the task for their own purposes free of charge.

Dated: Budapest, 28.04. 2022

student's signature

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Faculty of Informatics

CONSULTATION LOG

Student's name: Neptun code: Specialty:

Saif Shalaan Abdulrazzaq Alkhniab [redacted] Computer Science Engineering

Phone number: Mailing address (e.g. domicile):

[redacted]

Degree project / thesis[1] title in Hungarian:

Számítógépes hálózatok vezérlési utasításainak válaszidejének csökkentése

Degree project / thesis[2] title in English:
Reducing the time response of computer networks for control instructions

Institutional supervisor: External supervisor:

Fésüs Péter
Please write the data in printed capital letters!

Occ.	Date	Content	Signature
1.	17/2/2022	RECOMMENDING OF SOME FORMAL SOURCES OF PAPERS	Fésüs Péter
2.	24/2/2022	COLLECTING SOME PAPERS ABOUT RELATED WORK	Fésüs Péter
3.	3/3/2022	FILTERING USEFUL PAPERS FOR THE THESIS	Fésüs Péter
4.	10/3/2022	SELECTING WRITING STRUCTURE FOR THESIS WORK	Fésüs Péter

The Consultation log is required to be countersigned by either supervisor on a total of 4 occasions of consultation.

The student has complied with the requirements of the subject titled Degree project I / Degree project II (BSc) or Thesis 1 / Thesis 2 / Thesis 3 / Thesis 4[3], and is allowed to proceed for reporting / defense[4].

Dated: Budapest, 10/03/2022

Fésüs Péter
.....
institutional supervisor

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Faculty of Informatics

CONSULTATION LOG

Student's name:	Neptun code:	Specialty:
Saif Shalaan Abdulrazzaq Alkhniab	[redacted]	Computer Science Engineering

Phone number: Mailing address (e.g. domicile):

[redacted]

Degree project / thesis[1] title in Hungarian:

Számítógépes hálózatok vezérlési utasításainak válaszidejének csökkentése

Degree project / thesis[2] title in English:
Reducing the time response of computer networks for control instructions

Institutional supervisor: External supervisor:

Fésüs Péter
Please write the data in printed capital letters!

Occ.	Date	Content	Signature
1.	17/3/2022	WRITING THE LITERATURE REVIEW OF THE THESIS	
2.	24/3/2022	DESIGNING SOME NETWORK TOPOLOGIES ON CISCO PACKET TRACKER	
3.	31/3/2022	DID CALCULATIONS FOR SOFTWARE-BASED SOLUTION	
4.	7/4/2022	WRITING CHAPTER 2 OF THE THESIS	

The Consultation log is required to be countersigned by either supervisor on a total of 4 occasions of consultation.

The student has complied with the requirements of the subject titled Degree project I / Degree project II (BSc) or Thesis 1 / Thesis 2 / Thesis 3 / Thesis 4[3], and is allowed to proceed for reporting / defense[4].

Dated: Budapest, 07/04/2022

.....
institutional supervisor

INSERT THE SIGNED CONSULTATION LOG (4th SEMESTER, 1 PAGE)

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Faculty of Informatics

CONSULTATION LOG

Student's name: Neptun code: Specialty:

Saif Shalaan Abdulrazzaq Alkhniab [redacted] Computer Science Engineering

Phone number: Mailing address (e.g. domicile):

[redacted]

Degree project / thesis[1] title in Hungarian:

Számítógépes hálózatok vezérlési utasításainak válaszidejének csökkentése

Degree project / thesis[2] title in English:
Reducing the time response of computer networks for control instructions

Institutional supervisor: External supervisor:

Fésüs Péter
Please write the data in printed capital letters!

Occ.	Date	Content	Signature
1.	14/4/2022	DESIGNING HARDWARE-BASED SOLUTION, THEORETICALLY	Fésüs Péter
2.	21/4/2022	IMPLEMENTATION OF HARDWARE ELECTRONIC'S CIRCUIT ON PROTEUS	Fésüs Péter
3.	28/4/2022	ENHANCING THE DESIGN TO COVER MORE FEATURES	Fésüs Péter
4.	5/5/2022	DID MODIFICATION ABOUT THE FINAL VERSION OF THESIS DOCUMENTATION	Fésüs Péter

The Consultation log is required to be countersigned by either supervisor on a total of 4 occasions of consultation.

The student has complied with the requirements of the subject titled Degree project I / Degree project II (BSc) or Thesis 1 / Thesis 2 / Thesis 3 / Thesis 4[3], and is allowed to proceed for reporting / defense[4].

Dated: Budapest, 09/05/2022

Fésüs Péter
.....
institutional supervisor

Contents

Abstract

This thesis is concerned with lowering the time required for information to be transmitted from one device to another inside a network. This thesis contains all of the knowledge essential for the reader to comprehend the thesis's central issue. The side information presented in this thesis acts as a refresher for the reader who is familiar with computer network operation. We thoroughly study the issue of network delays, emphasizing the critical nature of minimizing information transmission time and the ramifications for certain applications. Additionally, we review existing solutions to the delay problem, highlighting the advantages and disadvantages of each strategy. This thesis present software and hardware solutions for resolving the delay issue. The offered solutions will be discussed in terms of their merits and limitations, as well as their applications. It be discussed when these solutions are most effective and what needs to be changed in present network protocols to achieve the suggested system's maximum performance. This research is concerned with control signals in particular, but the delay in the network will be treated generally, with the control signal treated as a normal signal, but with an emphasis on how to customize the proposed system to work only in relation to the delay time of control signals passing through the network. At the end of the thesis, the designed system will be discussed scientifically and objectively, with an emphasis on the concerns that could serve as the focal points of future study to further enhance the system.

1 Chapter 1: literature review

1.1 Computer network

A computer network is a collection of computers that are connected in a specific method that enables them to communicate data efficiently, reliably, and at a high rate of speed. The connection techniques can be wired, in which information is transmitted via electrical signals, wireless, in which data is transmitted by electromagnetic waves, or optical fibers, in which data is transmitted via light signals. Of course, there are other additional methods of connecting, but the primary objective is the same in each. Additionally, there are various types of network topologies, including the star technique, the ring method, and the mixed approach, also known as the complex method. The network may incorporate one or more types of transmission lines, as well as a variety of the previously described technologies for connection and topology. Nowadays, networks include a variety of devices as it shown in below figure; they are no longer limited to computers, as they formerly were. This diversity and growth in the number of devices generates a great deal of traffic on the network, necessitating the use of appropriate transmission protocols to ensure network stability while also ensuring that information reaches its intended destination without data loss or error, and at an acceptable speed. The simplest computer network consists of two devices connected to one another and exchanging data, whereas complex networks comprise of several devices such as computers, phones, televisions, and sensors. While you are sending an email to a specific user on the network, a variety of other operations are occurring concurrently. There is another person watching a football match, another making an Internet call, and a doctor performing surgery remotely via a medical robot, which is controlled via the same network. At the same time, there are smart sensors transmitting data to data centers located in other places around the world, and there is another person browsing a website, another making an Internet call, and yet another conducting numerous other operations via the Internet [1] [2]. As a result, various issues may arise, including how to coordinate all of these tasks occurring concurrently and in multiple regions around the globe. How is data transmitted between devices? In other words, how do devices communicate with one another within a network? These and other problems will be addressed in the subsequent chapters of this thesis.

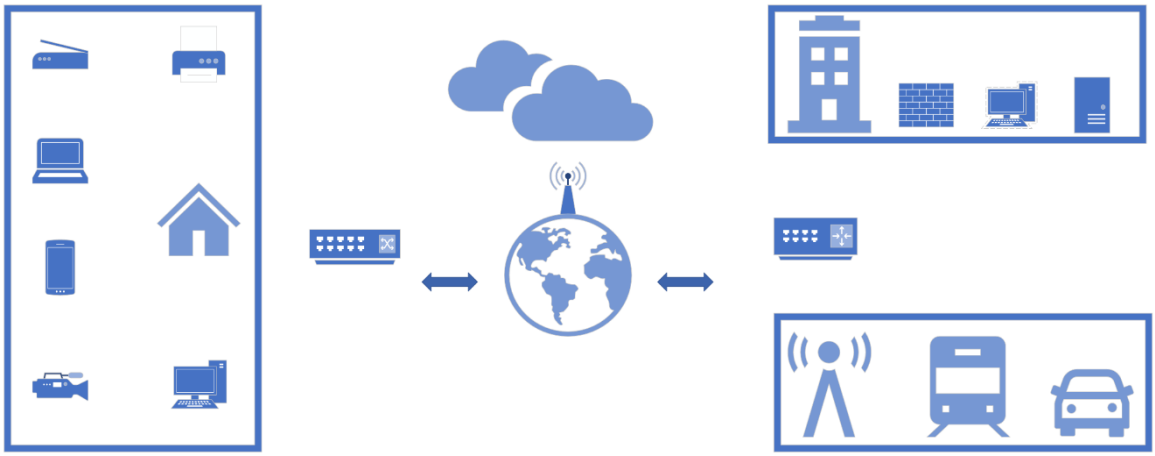


Figure 1 Network devices

1.2 Networking topologies

They are merely methods of connecting computers or other devices within a network to one another in order to create a network. There are two major types of representation: physical and logical.

Physical expression: It is the network's depiction based on the physical connectivity of devices.

Logical expression: It is determined by the data routes used for information sharing within the network.

Numerous topologies exist. We will discuss a few of the more prevalent ones (Circular topology, line topology, bus topology, star topology, fully connected topology).

A network's physical and logical representations may differ. For instance, we can observe a network that is connected in a star topology but operates in a line topology [3].

1.2.1 Star topology

This technique is represented by a central device in the network that is connected to all other devices and through which communication between any two devices within the network must occur. This architecture demonstrates the ease of connection and the low number of connections required to construct these types of networks, as well as the fact that the network is unaffected by terminal device failure. The downsides of this architecture include the lack of many data channels, which affects data transmission speed due to the inability to establish multiple connections concurrently. Additionally, we should notice that the network is wholly dependent on the central device; if the central device fails, the network would cease to function completely [3].

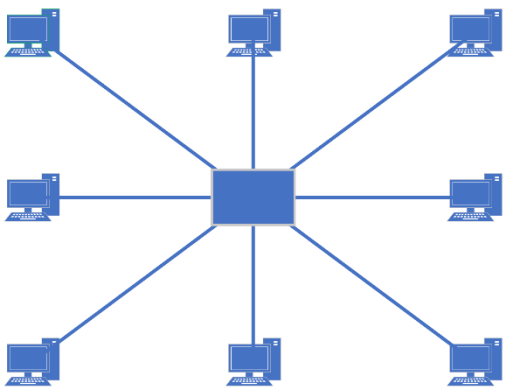


Figure 2 star topology

1.2.2 Ring topology

The devices are connected in this topology in the form of a ring, with each device connected to its two neighboring devices. The advantages of this network are its simplicity and lack of necessary connections, as well as the enumeration of data routes, which enables communication between any two devices on the network via two distinct paths, with the shortest or fastest path being selected. Additionally, we highlight that preventing one device

from communicating with another inside the network does not disrupt the entire network; rather, a different method might be chosen to interact with a specific device. The downsides of this method are that it is limited in terms of data transfer speed; while it is faster than the star method, it is far from optimal [3].

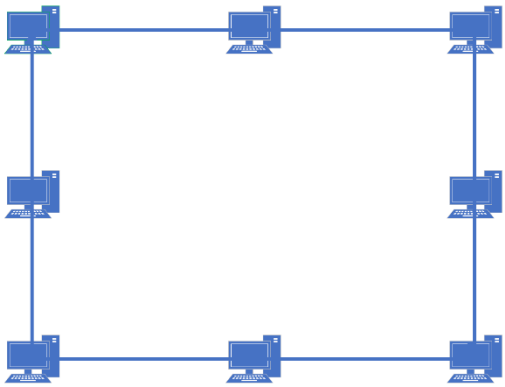


Figure 3 Ring topology

1.2.3 Fully linked (Mesh) topology

This topology establishes a connection between each device on the network. Among the network's advantages is its fast data sharing, which is assisted by a direct connection between all of the network's nodes. The other advantage is that the network's diversity of data paths allows for greater flexibility in choosing the optimal path between connected devices. Additionally, the failure of any device, or even a large number of devices, in the network has no effect on the operation of the remaining devices, as this is a unique property. Concerning the network's downsides, they include its complexity and expensive cost, as it necessitates the construction of several links [3].

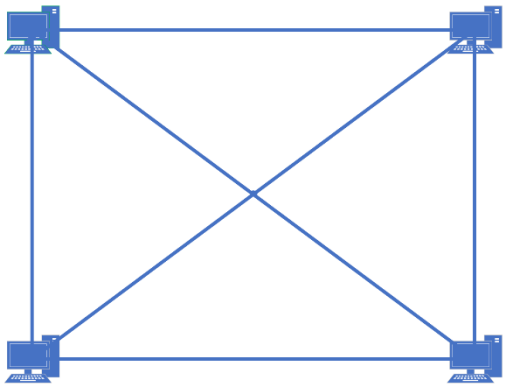


Figure 4 Fully linked topology

1.2.4 Line topology

Akin in concept to the ring topology, the only difference is that the final node of a network is not directly connected to the first node, but rather it is connected sequentially in a way similar to train cars.

The advantage of this network is the simplicity of the link and the lack of required connections, but the main disadvantage is the slowness with which data is transferred, as well as the possibility of network interruption in the event that one of the devices fails to function, although it should be noted that it is possible for some devices to remain connected to each other in the event that one of the intermediate devices fails, or in other words, we can say that a device malfunctions in the middle of the network may divide the network in two separated networks [3].

Figure 5 Line topology

1.2.5 Bus topology

In this design, each device in the network is connected to a central transmission line that connects all network devices via transmission lines. The message is delivered simultaneously to all network devices, but only the device with the same address as the message will get the data; the remaining devices are unconcerned about the message sent. One of the advantages of this network is that each device is self-contained; if one or more devices fail, this has no effect on the network's operation; but, if a fault develops in the central line, the network will cease to function or split into two distinct networks. Additionally, this topology has a slow data transfer rate. Communicating with more than two devices concurrently is really tough [4].

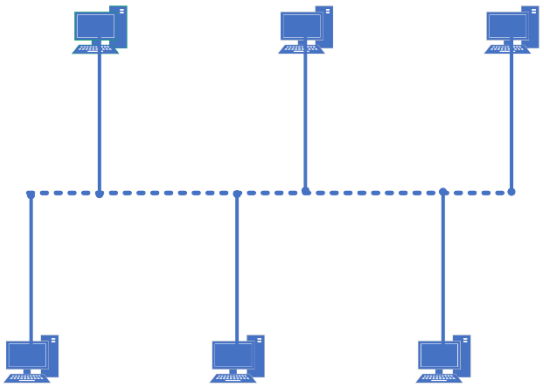


Figure 6 Bus topology

1.2.6 Mixed (Hybrid) topology

This sort of connection combines several of the previously stated connection topologies, and it is the most common, particularly in large networks, where the huge network is composed of several small sub-networks [3].

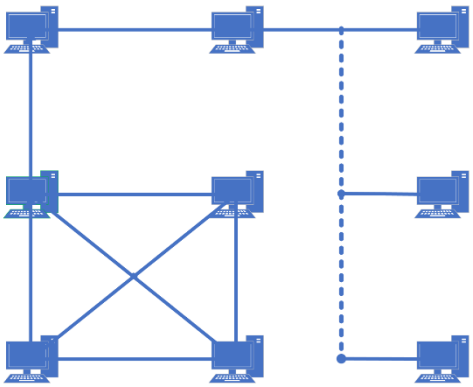


Figure 7 Mixed (Hybrid) topology

1.3 Transmission lines

They are a transmission technology used to transmit data or connect devices together. They come in a variety of forms. We shall summarize them briefly, highlighting the primary advantages and downsides of each of these methods.

1Coaxial cable

2_ Twisted cable

3_ Optical fiber

4_ Wireless communication

1.3.1 Coaxial cable

It is a metal carrier consisting primarily of copper (1) that is enclosed by an insulating layer (2) and then encircled by a mesh of wires (3) that surrounds the coaxial carrier and protects it from external effects. Apart from preventing interference with the original signal from surrounding radio signals, the mesh also prevents the radiation of electrical signals in the form of electromagnetic waves from the coaxial conductor where the surrounding mesh is connected to the ground, thereby enabling it to cancel any electromagnetic signal from or to the coaxial. Finally, an insulating coating (4) is applied to the entire cable body to shield it from the environment's effects [4].

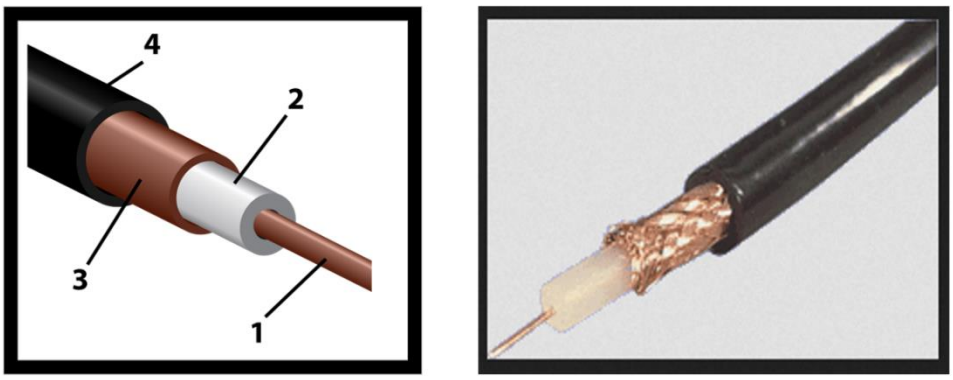


Figure 8 Coaxial cable structure

1.3.2 Twisted pair cable

It is composed of numerous pairs of conductors, which are typically made of copper and are insulated by an insulating material such as plastic. Each of the parallel pairs is twisted in conjunction with the other to mitigate the influence of the magnetic field created during the transmission of data at high frequencies through the transmission lines. The second purpose is to minimize the influence of magnetic fields around the transmission line on the data transferred across it. Twisting the wires together reverses the polarity of the electromagnetic fields, canceling each other out.

Wired transmission lines are defined by their ease of installation and connecting. The downsides include the high manufacturing cost and inability to operate in systems requiring high data transmission rates, as well as the limited distance that can be covered by a metal transmission line transmitting data via electric current [5].

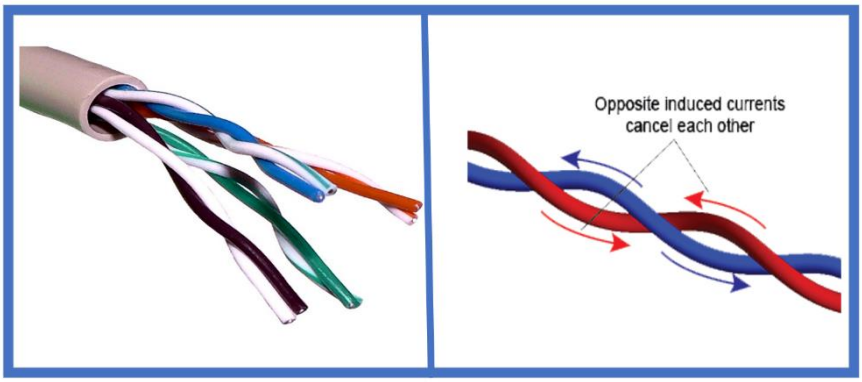


Figure 9 Twisted pair cable structure

1.3.3 Optical fiber

An optical fiber is a very fine thread constructed of silicon. Optical signals are used to transport data across this wire. This line is distinguished by its capacity to carry extremely high data speeds, as well as its low manufacturing cost in comparison to copper conductors. Additionally, it is capable of transmitting data at a high data rate over very long distances; these characteristics, among others, make this transmission line ideal for connecting countries or serving as the primary transmission line feeding remote locations. The difficulties associated with this type of line include manufacture, installation, and maintenance [5].

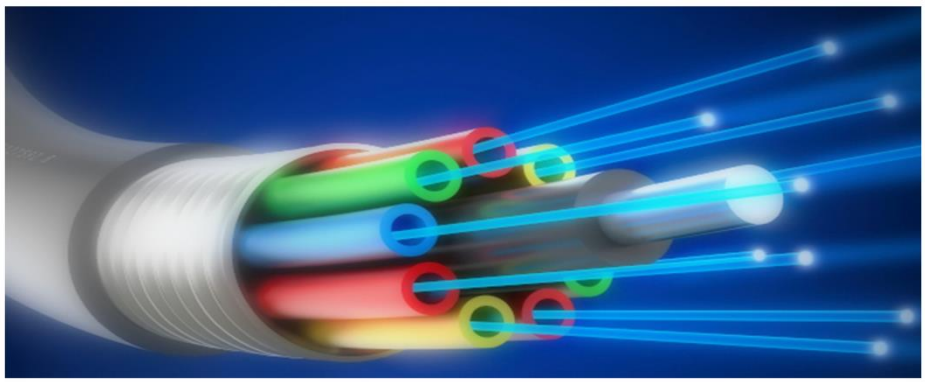


Figure 10 Optical fiber transmission line

1.3.4 Wireless connection

Electromagnetic waves are utilized to communicate between nodes in a network. We have referred to this kind of communication as a transmission line figuratively because it is not a tangible physical object but rather performs the same purpose as other transmission lines, with the same advantages and disadvantages. One of the benefits of employing this technology in networks is the ease with which electromagnetic waves propagate in open and rugged areas that cannot be connected via cables, as well as the large area covered by this technology, which enables mobile devices such as phones, cars, and planes to connect to the network smoothly. This method employs a broad frequency range with numerous bands, which prevents channels from overlapping. One of the most glaring flaws and issues with this technology is that it is employed simultaneously from multiple sides or multiple systems. For example, television and radio systems, as well as radar systems, all use the same technology, which means they must commit to a single frequency range and operate within it in order to avoid interfering with the other systems, limiting the number of devices that can be used within a given geographical area [4].

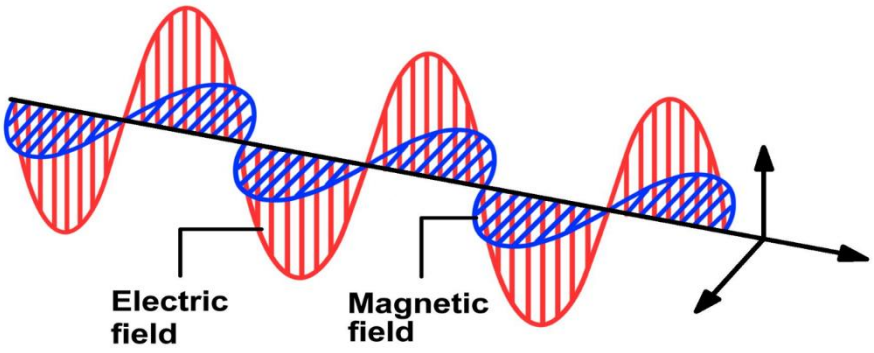


Figure 11 Electromagnetic wave

Transmission lines connect network devices. We can see the usage of all of the above forms of transmission lines in huge networks; for example, your mobile phone is connected to the network using Wi-Fi technology, which utilizes wireless communications. At home or at work, the phone is connected to the router, which is generally connected to the Internet via wired connections between the router and the sub-provider of Internet service. The sub-provider is connected to the service's primary distributor through an optical cable, and so on. We highlight that a particular technique was applied when necessary. Because the phone requires mobility within the house, electromagnetic wave technology must be employed to connect. Concerning the router, it was connected via Ethernet cable because its location is fixed and it does not require a huge amount of bandwidth because it only serves a small number of devices within the house or organization. Additionally, electromagnetic waves were not used for this purpose due to the difficulties of propagating them between buildings and the need to conserve frequencies for other applications. We still have the connection between the primary provider and the service sub, where the optical cable was chosen for this task due to the large bandwidth required by the sub-distributor, which cannot be transmitted via electromagnetic waves or electric wires, as well as the distance between the two parties. Which necessitates the usage of an optical cable to do this task. For the same reason as the last one, optical cables are used to

connect countries and continents where enough bandwidth is required to meet user needs [6] [5].

1.4 Network devices

There are several sorts of devices connected to the network, but they may all be categorized into two broad categories. [2]
1- end-user devices, such as laptops, iPads, smartphones, and sensors, among others
2- intermediate devices such as switches, hubs, routers, and so forth.

1.4.1 End user devices

These are the final elements in the network that do not serve any other devices. We can refer to the computer as an end user device in the network because it is the final element and does not serve any other components, but if other devices are linked via this computer, it will transition from an end user to an intermediary device. For instance, if a phone is connected to the network via a computer, the computer acts as an intermediary device in the network by serving other devices [2].

1.4.2 Intermediary devices

They are the devices that connect the network's components. They may be used to connect end user devices or to connect one end user device to another intermediary, and so forth.
Intermediaries' aim is to guarantee the network's stability, high security, reliability, and fast communication between other devices, as well as network scalability. This means that data is sent from one end device in the network to another end user device via a succession of intermediary devices. While intermediate devices do not benefit from this data, they are responsible for passing it from one device to another until it reaches its destination [2].
We will now quickly describe the devices (routers, switches, and hubs), but before we do, it is necessary to understand the concept of a critical component shared by all network devices: the interface.
Interface: It is a device that is attached to any product that has the ability of connecting to a network, regardless of whether the product is a computer, phone, router, printer, or Internet of things device. The interface of each gadget may be slightly different. The interface used to connect the computer to the network via the Ethernet connection varies in appearance. Regarding the interface that links the phone to the network via Wi-Fi technology, as well as another device that connects to the network via optical fiber technology, the goal is the same in both cases: to achieve a seamless connection to the network. Each interface is assigned a unique number that is unique to it. This is referred to as the Mac address, and it is used to identify various network interface devices. The Mac address will be discussed in further detail in subsequent chapters [6].

1.4.3 The hub

It is a gadget that is used to connect multiple computers to establish a network. It is frequently used to configure local area networks (LANs).

Numerous ports are located at the hub. Each computer is associated with a unique port. When a computer sends data to another device, the data is circulated to all devices connected to the Hub; because the Hub cannot interpret the data or determine which devices to give it to, all devices in the network receive the data, but only the concerned device responds to it. As for the remaining devices, they will discard this data and ignore it because it has been transmitted to another device. Computers identify the sender of the message based on the presence of the target device's Mac address within the message. While this device is simple and affordable, it is ineffective in big networks due to the inability to establish multiple connections simultaneously within the network. If a device delivers data to another device on the network, all other devices on the network must wait, even if they are unrelated to this Connection. They must wait for the shared channel to become available for another device-to-device connection [7].

1.4.4 The Switch

is quite similar to the Hub in that it connects computers together via the switch's ports, but the distinction is that this device analyzes the data that comes into a specific port from a certain device and determines who the data should be transferred to. Following that, this device directs data to the appropriate port exclusively; this provides network security and privacy, as data is delivered from one device to another without being circulated to all network devices, as is the case with the Hub device. Additionally, the remaining devices may connect concurrently. This device offers network isolation, and it is also used in local area network (LAN) networks. Later parts of this thesis will provide a practical example of delivering data from one device to another, encompassing all stages of the data's journey from the source to the destination [7].

1.4.5 The router

It is a network device that receives and analyzes data and determines the most appropriate path for it, even if the requested device's address is outside the local network. This is the characteristic that differentiates it from a switch. If the requested device's address is not within the network, the switch cannot forward the request beyond the network, in contrast to the router, which may forward the request outside the network. Additionally, information can be acquired from sources outside the network. In other words, the router can connect two or more distinct networks and coordinate their operations. Clarifying the concept of a router versus a switch. The switch can be thought of as a local mail, but the router is a worldwide mail, allowing information to be sent to any location in the world [7].

2 Chapter 2: How is the network configured?

The computer network's design is remarkably similar to the conventional postal system used from ancient times to the present day, in which houses, cities, and streets were numbered and addressed in a certain manner that ensured each house's address was unique. For instance, your house may be numbered 37, whereas your friend resides in a different neighborhood. However, by accident, his house number is also 37, but you live in an area called A, whilst your friend lives in an area called B. Here, your house address is A 37, while your friend's is B 37. Finally, your house address is absolutely unique from your friend's home address or any other home address anywhere in the world due to the highly organized numbering and coding of areas that prevents any home or institution's address from being similar to another. From this perspective, the computer network was structured in such a way that each device linked to it has a unique address distinct from that of any other device. Let us examine how this was accomplished.
Each interface on each device was assigned a unique number. This is referred to as the Mac address, which is a 12-digit hexadecimal or 6-byte binary number. The MAC address can be expressed in one of three distinct forms (a1:b2:c3:d4:e5:f6 , a1-b2-c3-d4-e5-f6 , a1b.2c3.d4e.5f6). The IEEE organization distributes these numbers to manufacturing businesses, who are required to follow these guidelines to avoid interference caused by the existence of two devices with the same Mac address on the same network. When a message is delivered from one device in the network to another device in the same network, the message must include the destination device's Mac address in order for the message to be routed correctly via the network's intermediary devices. This procedure appears to be reasonable in tiny networks with a small number of nodes. Although the devices are few, digging deeper reveals that depending solely on the Mac address is insufficient, despite the fact that it is unique across the network. The process of locating the required device's address is nearly impossible, and as a result of the massive increase in the number of devices worldwide, the so-called IP (Internet Protocol) address was added, which is a number composed of four decimal parts, each of which must be between 0 and 255. This IP address is divided in a way that approximates the concept of area numbering in the postal system. This number is assigned to each device connected to the network, as well as to the entire network. The IP addresses of the devices must be distinct from one another if they are connected to the same network, but they do not have to be distinct if they are connected to different networks. For instance, two devices can share the same IP address. One of them is connected to network A, while the other is connected to network B; nevertheless, we must keep in mind that the primary address of network A must be distinct from the primary address of network B. This technology, which enables the use of the same IP address in two distinct networks, is referred to as (DHCP), and so we might have had the same result. In traditional mail, this mechanism and the use of the IP allow for the rapid discovery of the desired device's address without the need to search through all the devices on the planet. The IP system is divided into geographical zones based on the primary services available in each region of the world. The distribution is handled by the Internet Corporation for Assigned Names and Numbers (ICANN). The IP address of the transmitted data must be provided in the packet, as this is how the right path for the data within the network is identified [8].

2.1 IP Address

It is a numerical address assigned to each device connected to the network; however, unlike the Mac address, the IP address of the same device may change depending on the network's location and connection time. You can connect to the Internet via a computer, and the network will assign your computer a unique address, such as 192.168.0.3. However, when you reconnect at a later time, you may not receive the same previous IP address; instead, you may receive a different address within the same range. IP version 4 is currently in use and consists of four bytes of binary numbers, although there is a project to switch to IP version 6 in the future. We will confine ourselves to explaining IP version 4 in this thesis because it is currently in use. The fourth version of IP consists of four decimal parts ranging from 0 to 255, separated by a dot (172.84.255.45). Throughout the network's operation, devices view this IP as having two parts, the first expressing the network's number and the second expressing the device's number within the network. Another number called a subnet divides the IP address into two portions, for example (255.255.255.0). The length of the subnet is identical to that of the IP address, and it also contains four parts of decimal numbers between 0 and 255. In the binary system, the AND operation is used between the subnet number and the IP address to sort the network address and device address parts [9].

2.2 Network protocols

We have obtained a general understanding of how data is transmitted from one device to another inside a network. Numerous producers of network products (computer routers, switches, and so on) exist in various nations throughout the world. A question comes up at this occasion: how does a network device created in one nation interact successfully and without faults with another network device manufactured by another firm located in a different location around the world? The solution is straightforward: through the international organization IEEE, which establishes standards and procedures that apply to all produced devices. These protocols are evaluated and recommended by networking professionals before being distributed to manufacturers. This avoids the issue of incompatibility. The OSI Model, which divides processes and devices in a network into seven layers, has been adopted in the field of networks. Each of these layers is comprised of a collection of devices, protocols, and processes. The protocols used in networks will be discussed implicitly in this thesis, along with an explanation of the seven layers. It's worth mentioning here that the sequence of layers varies depending on the sender and recipient. On the sender's side, the data transmission process begins at the application layer and concludes at the seventh layer, the physical layer. On the receiver side, the data processing process begins at the seventh layer, which is the physical layer, and ends at the application layer. As can be seen in certain sources, the layer numbering is inverse. The following figure shows OSI (Open Systems Interconnection) model layers.

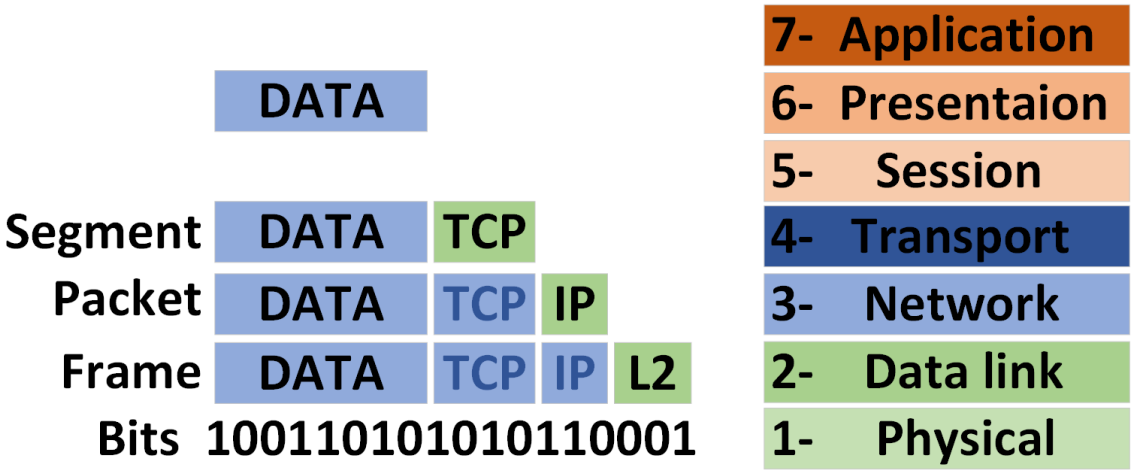


Figure 12 OSI model layers

Numbering is used in this thesis from the recipient's perspective, with the physical layer being numbered one and the application layer being numbered seven [9].

2.3 OSI Model layers

2.3.1 Physical layer

The physical layer's principal job is to transport data from one location to another, or from one device to another. This layer contains a variety of technologies and equipment, such as connection cables and Wi-Fi. As previously indicated, Wi-Fi is used to transmit data and connect network devices. Likewise, for the hub device [8]. Take note that each technique discussed in previous chapters has been explored in detail.

2.3.2 Data link layer

It is the layer responsible for properly preparing data for the physical layer, which includes a variety of devices and technologies such as (NIC - Network Interface Card) (WiFi Access Card). These devices transform data to electrical, optical, or electromagnetic signals in conjunction with the network's transmission line. This layer utilized the Mac address and added it to the transmitted data packet in order to move data from one network node to another. Another device that operates inside the boundaries of this layer is the switch [1].

2.3.3 Network layer

This layer is in charge of ensuring that data is transmitted end-to-end within the network. The IP address is appended to the packet at this layer. This layer utilizes technologies and devices such as hosts, routers, and any other device that works with IP addresses. There is a protocol known as ARP (Address Resolution Protocol) that operates at the layer 2 and layer 3 layers. This protocol works in conjunction with the Mac's IP address to ensure that data is routed correctly from the source to the destination [1].

2.3.4 Transport layer

This layer identifies and directs data within a single device to the appropriate application. For instance, an email was received on your computer while you were watching a football match. The question is how email data was separated from video or football data. This was accomplished by appending a port number to the data delivered. When a computer receives data, it determines the port number, which reflects the device's necessary application. At this layer, UDP and TCP operate. Each of them uses ports 0 to 65535, with each port being dedicated to a particular application on the device [1]. UDP (User Datagram Protocol) and TCP (Transmission Control Protocol) both accomplish the identical task of transferring data from one side to the other, but they do so in very different ways, making each one suitable for certain applications but not the other. The UDP protocol is used in broadcast and live communication applications because it provides a fast but unreliable connection. For instance, if you were on a phone call and the network went down for a period of time before reconnecting the call, the aforesaid information will be lost during the cutoff period, indicating that this protocol continues to send data regardless of whether it reaches the destination or not. In contrast to TCP, which assures that data is accurately routed from the source to the destination. If a network failure happens during data transmission and a portion of the data is lost, this protocol resends the portion of the data that was lost. This property enables this protocol to be utilized in applications requiring high reliability regardless of connection speed [10].

2.3.5 Session layer

This layer is where authorization and authentication occur. This layer is responsible for managing the network connection in terms of permissions and security. Additionally, to sorting the incoming data, which arrives in the form of packets. When a website is browsed, the host server provides information about the website in the form of packets comprising images, text, and so on. As explained previously, the program is aware of its own packet via the port number. Sorting the contents of the packet within the application is a duty for the session layer [4].

2.3.6 Presentation layer

This layer is where data is sent across systems according to the data destination. When data is received from a particular source, this layer converts it from machine language (1,0) to human language (numbers and letters). However, if we wish to communicate specific data to a particular recipient, this layer is responsible for the conversion of data from human to machine language. Additionally, there are additional processes such as data compression and encryption. Certain data is encrypted before transmission to the network for security and privacy reasons. Other data is compressed to reduce the network's bandwidth use [4].

2.3.7 Application layer

This layer's purpose is to communicate with the user. Numerous programs make use of protocols from this layer. For example, Internet browsers (Chrome, Firefox, and so on), email, and Skype all make use of application layer protocols such as http, https, ftp, and smtp.
FTP: File transfer protocol. It used to send files through the network.
HTTP: HyperText Transfer Protocol. Services devoted exclusively to web browsing

Https: HyperText Transfer Protocol secure. Dedicated to exploring the web safely and securely
SMTP: Simple Mail Transfer Protocol. It is used for sending email applications.
It is critical to notice that the data in this layer is in the form of readable text, numbers, and graphics. The application layer is the first layer in the data transmission process and the last layer in the data reception process [1].

2.3.8 Practical example

A practical example of data transmission between two devices through a network:
We have obtained a general understanding of the network's most critical processes, components, and devices. We will walk through the process of transmitting data from one computer to another on the same network. The peripheral devices are connected via an intermediary device called a switch. The network diagram is depicted in the following image.

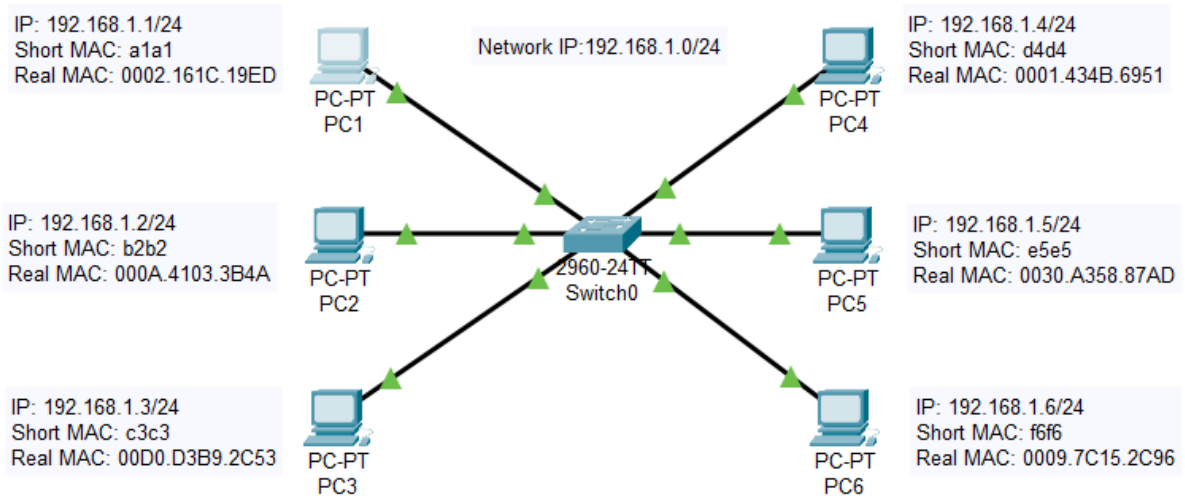


Figure 13 Star topology using CISCO switch device and 6 PCs

This form of link is referred known as the star topology, which was discussed previously in this thesis. To ease the simulation procedure, we've shortened the Mac address of Devices. We will analyze the process of data transmission from device 1 to device 4, assuming that device 1 wishes to send an email to device 4. The user on device number 1 composes the email in text format using the email program or one of the Internet browsers. When the user clicks the send button, the voyage of data transfer from device 1 to device 4 begins. The application layer, the presentation layer, and the session layer all initialize, encrypt, and format data in a network-compatible manner. Following that, the data is transferred to the transport layer 4 in the form of a block. The information is supplemented by the port number, constituting the so-called segment. This data is then forwarded to the network layer, where the source and destination IP addresses are added to ensure an end-to-end delivery procedure. The so-called packet is formed here. Following that, the packet is transferred to the data link layer, where a new header with the source and destination Mac address is appended to the package. At this moment, the data block is referred to as frame. It's worth noting here that when a new header is added, the prior data is considered as a single block of data. When the IP header was added, the preceding segment was considered as a single block of data; similarly, when the MAC address header

was added, the previous segment was treated as a single block of data as well, regardless of the type of data it contained. The frame is transmitted to the physical layer by electrical, wireless, or optical signals, depending on the interface and transmission line utilized. The switch transmits data via port number one. The switch determines the MAC address of the destination device and stores a table on it. According to the following figure, this table comprises the ports and MAC addresses of each device connected to the switch via any port. To show the MAC table in Cisco switch use command: **show mac address-table**

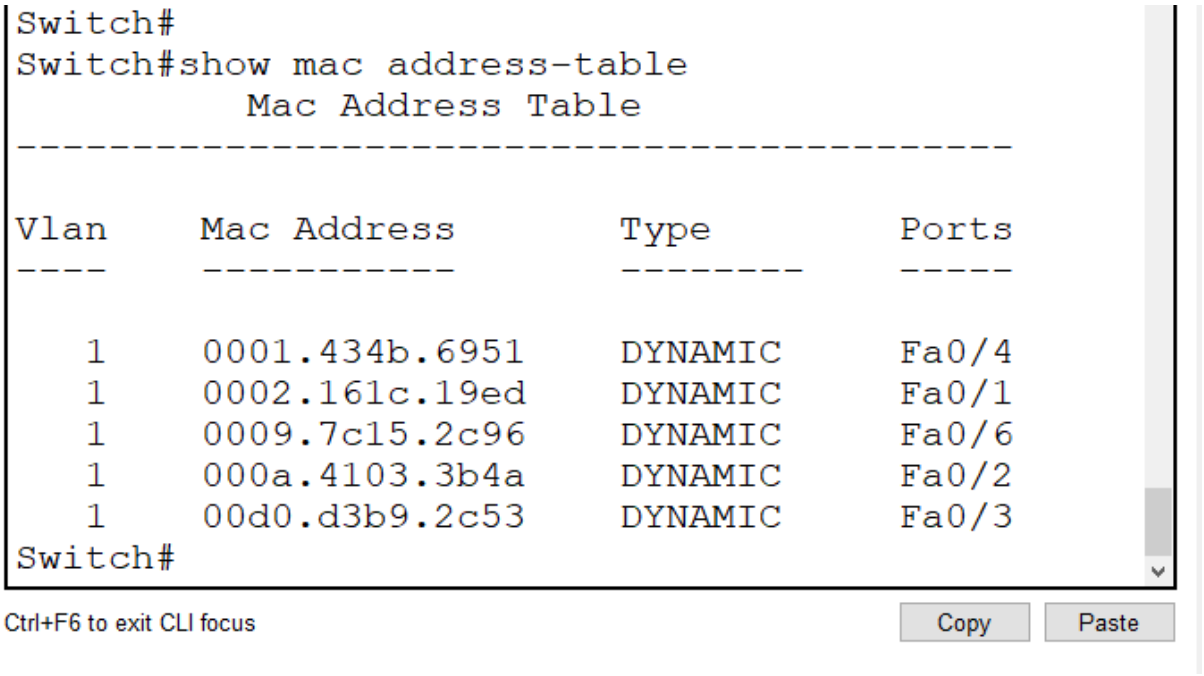


Figure 14 MAC address table in CISCO switch

The switch routes the data to the fourth port, which is occupied by device number 4, which receives it, analyses it, extracts the basic information sent, decrypts it, and displays the basic text to the user on the fourth computer's screen. The activities of the fourth machine are diametrically opposed to those of the first. Receiving commences at the physical layer and concludes at the application layer. The destination device was identified using the MAC address table contained within the switch, rather than the IP address, because it is connected to the same network and not to another. In other words, both the source and destination devices are linked to the same switch. When the network is powered on for the first time, the switch's MAC address table is empty. The switch uses a method called learning to determine the MAC addresses of the devices linked to it within the network. When a connection is established between two devices, the switch stores the MAC addresses of these devices together with the ports to which they are attached in the switch's MAC table. This is referred to as switch learning. A critical question may arise at this address: how do two devices communicate for the first time if the switch does not already include their Mac addresses? To show this technique, we'll utilize the same network and devices as before, but this time we'll assume that the switch's MAC address table remains empty. When device 1 reaches the data link layer to configure the frame, it will require the MAC of device 4 to finish the frame. However, the ARP

table in device number one remains empty and has no information about the MAC of device number four. The next figure depicts the ARP table in PC1 prior to it connecting to any network devices, indicating that it lacks information about the entire topology. To view the ARP table on a Windows computer, use the command (**arp – a**).

Figure 15 ARP table in windows machine

As a result, the transmission process will come to a halt.
The device No. 1 sends an ARP request. This request contains the IP address of device 4, the Mac address of device 1, and the destination MAC address FFFF (FF-FF-FF-FF-FF-FF); it will then be sent to the switch. The switch receives data via port 1 and analyzes it to determine the destination device's MAC address. However, you'll notice that the destination device's MAC address is FFFF. In the system, this MAC is reserved for the so-called broadcast MAC address. The switch from this MAC address will understand that it must broadcast the received data to all ports except the one from which it originated; this is exactly what the switch will do. The switch performs a broadcast operation, transmitting the received data through port number one to all switch ports except port number one without altering the data. Simultaneously, the switch inserts a new line into the table. This line contains the port number, which is one, and the source device's MAC address, which is one. The switch distributes information to all devices on the network. All network devices evaluate received data down to the IP address level, however all devices will discover that the IP address Except for the fourth device, whose IP is compatible with the one included in the message, the one contained in the message is distinct from the IP of each device, except for the fourth device, which will discover a match between its IP and the one contained in the message. Except for the fourth device, which will add a new line to the ARP Table, all devices will disregard the request. This line contains the device's IP and MAC addresses. Following that, device No. 4 will react to the request by sending a message with its MAC address included. This message is delivered to the switch's port No. 4, which creates a new line in the table. This line contains the fourth device's port number and MAC address. The switch analyzes the message received from device No. 4 and determines that it is directed to device No. 1 by the MAC Address included within the message. The switch is already aware that this Mac is connected to port No. 1 in this situation, as this information is now stored in the switch table. As a result, it will route to port number one, bringing you to device number one. The device number, in turn, adds a new line to the ARP Table, containing the fourth device's MAC address and IP address. Following that, device No. 1 resumes transmission, as the missing data is now available. The number one device puts the MAC address of the fourth device in the message and sends it to the Switch. The switch will receive the message from port number one, but because the private MAC is already in device number one and is saved in the switch table, it will not add it again, but will do a refresher on this information. The switch routes the request to port number four because it has determined that

the device is now connected to that port. The destination is connected to the fourth port as a result of the preceding scenario's ARP Table entry. As a result, the data is transmitted to device No. 4. The following figure depicts PC1's ARP table after it established connections to all devices on the network, indicating that it now knows information about the entire topology.

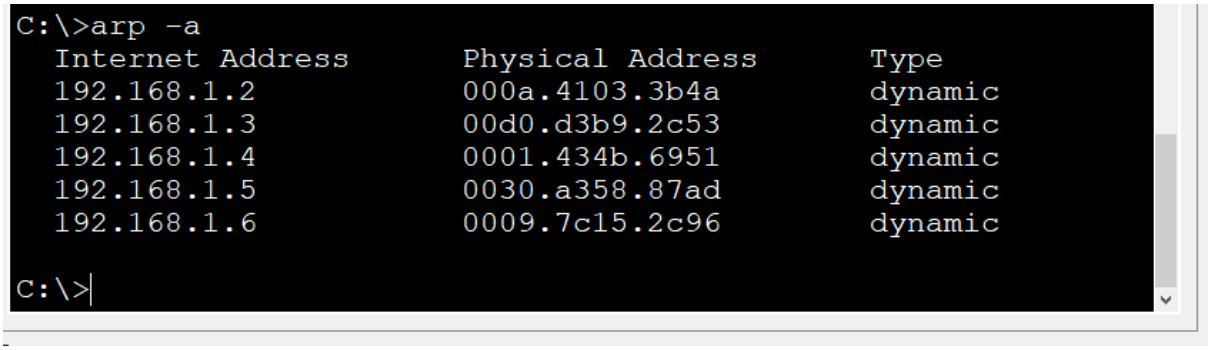


Figure 16 ARP table in windows machine

It's worth noting here that the ARP table is regularly updated and that its components are automatically removed after a specified time period. For instance, if no requests are received from or to device number three for a specified length of time, its information is erased from the ARP and MAC tables, and so on for the remaining devices [9] .

3 Chapter 3: Latency

It is the time required for information to be transmitted from the source device to the destination device, as this time period begins when the source device initiates the sending process and ends when the destination device completes the receiving process. Latency is divided into four categories. [11]

1- Transmission delay
2- Propagation delay
3- Queuing delay
4- Processing delay

Where latency is equal to the total of the preceding four variables, as in the following equation.
Latency = Transmission delay + Propagation delay + Queuing delay + Processing delay

3.1 Transmission delay

It is the time required to transmit the whole message sent from the sending device. Assume a computer wishes to convey the message "Hello world" to another device. The transmission time is the amount of time required to convey the data contained in this message from the sending device to the transmission line. The data is represented by zeros and ones. As for the transmission line, it refers to the medium over which this data is transmitted, regardless of whether it is an optical fiber, wire, or electromagnetic wave...etc.
The transmission time is dependent on two variables: the message's size and the available bandwidth; they exhibit the following relationship.[12]
Transmission delay = Message size / Bandwidth

3.2 Propagation delay

It is the amount of time it takes for a bit to go from the source device to the destination device via the transmission medium (transmission line). This time is dependent on two variables: the distance between the transmitter and receiver and the medium's propagation speed. The following equation expresses the diffusion time and its relationship to the two previously mentioned factors. [12]
Propagation delay = distance / Propagation speed

3.3 Queuing delay

It is the amount of time required for intermediate devices or even end devices to receive an entire message from the transmission line before processing it. In other words, it is the amount of time spent by data in the queuing interface prior to being processed by the processor. This time period is variable and not fixed. It is influenced by the network's load. When the network is under a heavy demand, the queuing time increases. However, if there is no load, this time will be extremely brief.[12]

3.4 Processing delay

It is the time taken by the node to process the message, regardless of whether it is sent or received. This time period is mostly dependent on the processor speed of the network devices. [12]
To minimize the time of any of the preceding four types of time, the causes controlling that duration must be managed in some way. To alter the transmission time, we must either vary the size of the message sent or the bandwidth available, or both. The same holds true for the propagation delay time, as either the distance or the speed of propagation, or both, must be managed. [11] [12]
This thesis is concerned with minimizing the latency of control commands traversing the network. This will be accomplished through decreasing the network's reaction time in general by minimizing queuing and processing times.

3.5 Congestion problem

The issue we are attempting to address in this thesis is as follows:
We demonstrated a fairly simple network in the preceding example. Today's networks are far more complicated; they may contain hundreds or even thousands of different devices (phones, computers, servers, sensors, etc.). This, of course, imposes a great load on the network's switch devices. This strain slows down the network's overall performance, as the switch is unable to handle this massive number of operations per second. Consider the design below.

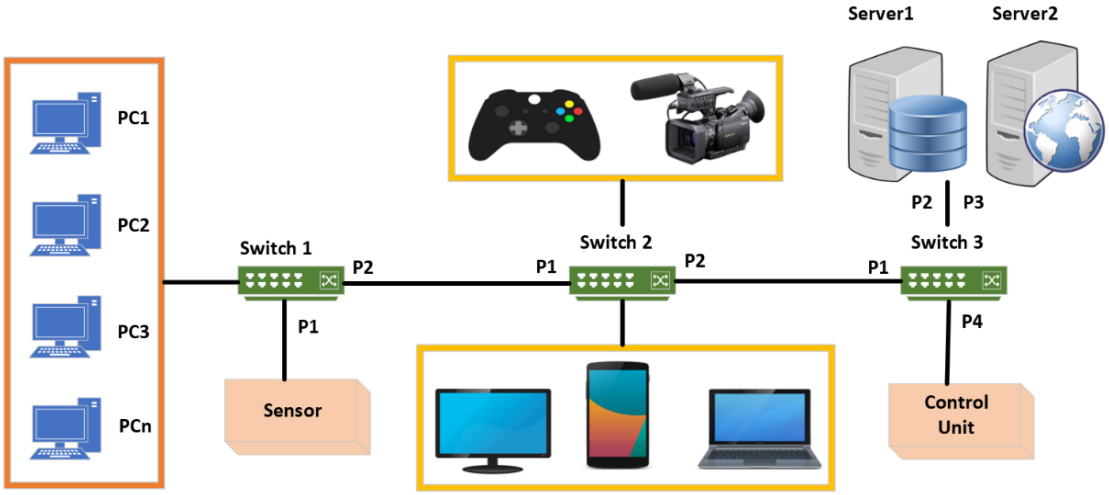


Figure 17 Network has multi types of devices

Consider that all the devices in the preceding network communicate with one another concurrently, and that all the different devices communicate with servers (1,2). It will be a significant strain on switch 3's port 1. Naturally, this results in a delay for all network devices attempting to communicate with servers (1,2), as all requests must transit through switch 3's port 1. The only solution for switch number three is to sequentially process requests. This circumstance of data congestion on a single port is referred to as a bottleneck, and it occurs frequently in networks. If we analyze the preceding network closely, we will see a large variety of devices. In an office, for example, there are computers. Additionally, there is a gaming device, a mobile phone, a TV, a laptop, and a sensor situated in a strategic location. It is nonsensical, for example, to order the requests on port No. 1 in switch No. 3 sequentially by

time of arrival. Because the gaming signal may be less critical than the sensor signal required to execute a particular control instruction. The same is true for television, telephone, and other signals. In such instances, network device designers and manufacturers must take the priority of each signal into account, which they have already done. There are numerous methods for resolving this issue. We will cover the most often utilized strategies and then discuss the enhancements proposed in this thesis.

3.6 Congestion management

Numerous procedures are employed to alleviate the condition of retinal congestion. We shall discuss the most often utilized ones.:

3.6.1 FIFO (First-In-First-Out) technology

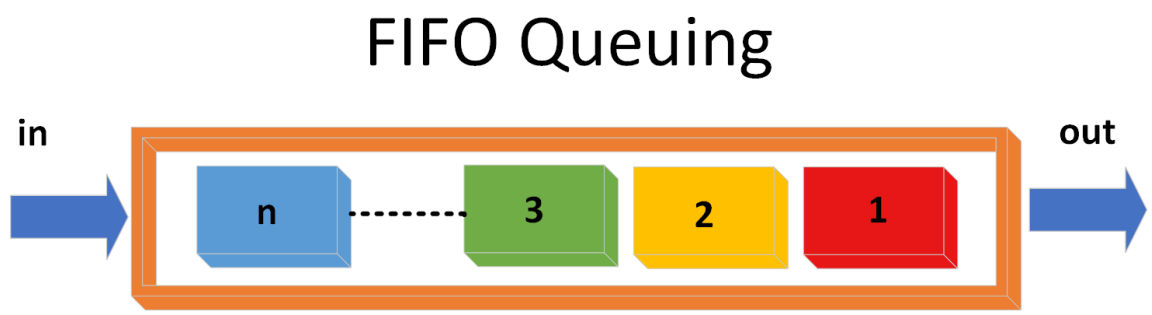


Figure 18 First In First Out Queue

This technology is applicable to both routers and switches. This strategy is based on allocating priority to arriving packets based on their arrival time at the interface. The packet that arrives first at the interface is processed first, and so on for the subsequent packets. The previous figure illustrates this algorithm. While this approach has several advantages, including its simplicity and speed of implementation, it is ineffective when some packets are more important than others. Although the packet must wait in line regardless of its priority, this algorithm is nevertheless utilized independently or as a component of another algorithm. [13]

3.6.2 PQ (Priority Queuing) technology

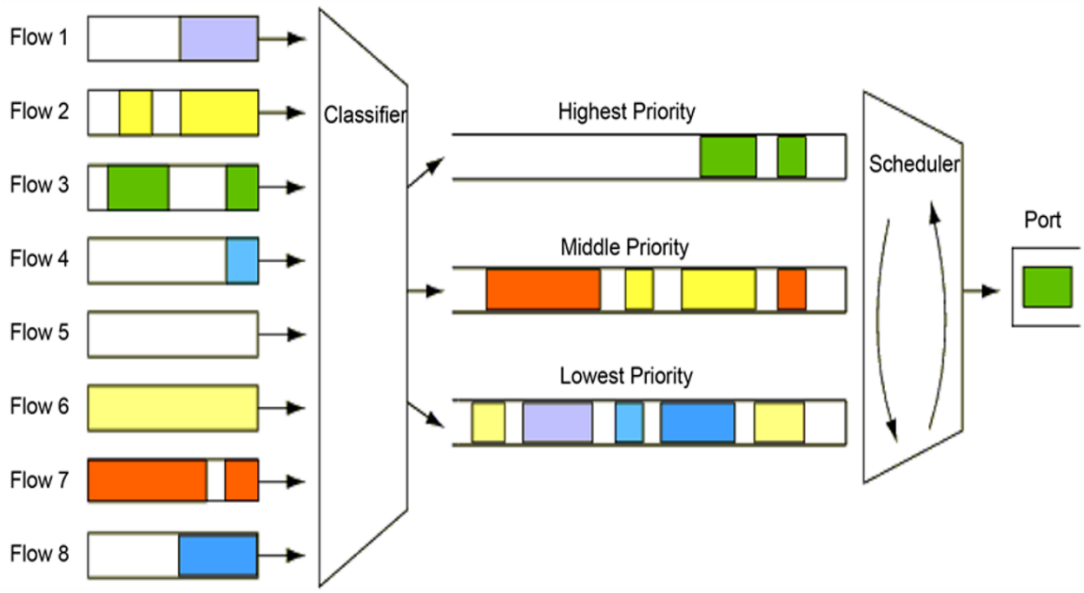


Figure 19 Priority Queue

The preceding figure depicts this algorithm. The algorithm does not isolate signals solely on the basis of their priority. Rather than that, each signal is assigned a specific priority level; there are signals classified as important, more important, and most important. This technology separates signals on multiple levels. For instance, if there are three levels of importance, this algorithm analyses the level 3 signals first; once all of the level 3 signals are processed, it moves on to the level 2 signals. Once all signals at level 2 have been processed, it will proceed to processing signals at level 1 and so on.

If Level 2, Level 1 and a Level 3 signal are received, the Level 3 signal will be processed first, even if it is received after the other signals. In other words, the signal with the highest priority was processed first, followed by the signal with the second lowest and finally the signal with the lowest priority. While this strategy is extremely advantageous for signals with many priorities, it does have certain drawbacks. One of the most significant disadvantages is the issue of signal starvation for both regular and low-importance signals. If many signals with a significance level of three are received, this algorithm will focus exclusively on the signals with a significance level of three. Here the issue of signal starvation for the remaining signals becomes apparent. Numerous modifications and research have been performed to address the algorithm's shortcomings. The solutions are aimed at decreasing the phenomenon of starvation in this algorithm and ensuring that all packets waiting to be processed are treated fairly. [14]

3.6.3 WFQ (Weighted Fair Queuing) technology

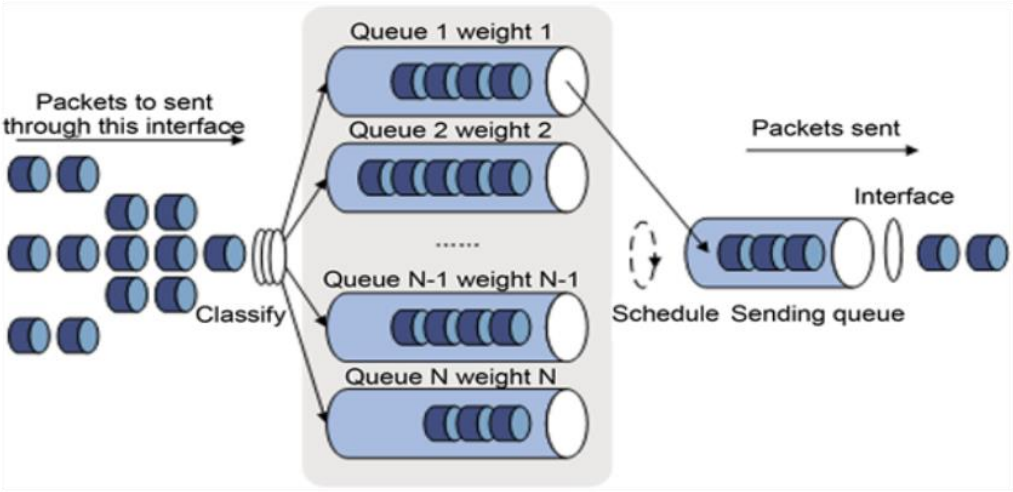


Figure 20 WFQ (Weighted Fair Queuing)

This method is quite similar to the PQ technique, in which signals are prioritized and divided into many queues. As shown in the picture above each queue is assigned a unique weight, which is a numerical value. Assume that an algorithm is built to function with four queues, and that the incoming signals are categorized and queued accordingly. Assume additionally that the first queue was assigned a weight of 5, the second queue was assigned a weight of 7, and the third and fourth queues were assigned a weight of 1. When this algorithm is in operation, it will take five requests from the first queue and then move to the second queue to take seven requests. Move to queue 3 to accept one request, then to queue 4 to accept one request as well, before returning to queue 1 to repeat the process. We remark that this approach overcomes the problem of famine, as all signals in all queues receive processing, regardless of how late the time or how many vital signals are there. Additionally, this algorithm has been updated multiple times. These studies concentrated on determining the queue's weight in order to optimize the system's performance and assure the maximum feasible fairness of queuing evidence . However, regardless of how far forward this algorithm's development is, there will always be a gray area in which the decision is made at random or via the FIFO process. When you want to choose between two signals with the same priority level, for example, both signals from level 3 or both signals from level 2, the system either chooses the first signal or chooses randomly amongst them, given that both signals have the same priority level. [14] Cisco devices employ the CBWFQ (Class Based Weighted Fair Queuing) algorithm, an upgrade to the WFQ algorithm, in which data is classified according to its nature (http requests, images, videos, phone conversations, and emails, for example), and then a bandwidth is assigned to each type of classified signal. Another algorithm, LLQ (Low Latency Queuing), is an extension of the CBWFQ algorithm that is utilized in Cisco products. [15]

3.7 Related work

We will describe some of the work of earlier scholars in the subject of designing algorithms for data transmission in networks in this section of the thesis. In [16] Using the PQ and WFQ queue scheduling algorithms, the author suggested an integrated packet scheduling method to reduce network congestion and boost performance. In [17] Considered an IEEE 802.11b WLAN-based wireless router with an FWQ algorithm for cognitive management of traffic flows with optimized packet sizes. As a result, the algorithm's weight coefficients, which determine the bandwidth share, are constantly being changed. The wireless router outputs throughput and latency for each service class. The new weight routers compute coefficients on a regular basis. The fuzzy model that has been suggested simplifies both the real-time UDP and the non-real-time TCP traffic sharing of a single connection TCP traffic with the best possible effort. In [18] For the chip verification platform, the research provides an EDWFQ task scheduling method, which alleviates the problem of starvation of SJF and decreases the Completion Time. According to the results, the test platform can operate more efficiently thanks to the algorithm that was presented. On the BR0101 chip verification platform, the suggested method is being used. In [19] When it comes to demand side management, weighted fair queuing (WFQ) is used. The algorithm's implementation is based on three factors: power demand, wait time, and group-to-group fairness. Remainder of renewable power supply and suggested fairness measures are compared with a Greedy strategy. In [20] Researchers in this work used an analytical technique to compare SPE and HPE models in SDN in order to gain insights into the performance of software and hardware switches. A hardware switch outperforms a software switch in terms of average delay and packet loss probability, according to our research Stable networks with low numbers of fresh flows arriving at switches for decision-making are required for this to work. It is becoming more and more common for a hardware switch to use the CPU for forwarding (as opposed to ASICs and TCAM) because of controller involvement. Experimentation with measurements that can provide actual detail insights will be an important part of our future effort in validating these models. In [21] To get around this problem, an improved fog computing architecture has been developed, which extends cloud services to the network's edge to reduce latency and congestion on that network. A number of problems must be overcome before the full potential of fog and IoT for real-time analytics can be realized. For real-time data transfer in IoT devices and cloud, fog computing includes all of the latency-minimization factors. Reinforcement learning is used to develop a fuzzy-based multi-agent system. [22] Fog computing in the Internet of Things is the foundation of this paper's four-layer system. In this concept, they employ JStorm to combine geographically large-scale distributed fog nodes into numerous clusters so that system can handle massive data. As a result, we propose a multi-channel data scheduling method based on network latency and oscillations in the industrial environment to solve the problem.

3.8 Deep Priority Queuing (DPQ)

We will concentrate on this case in this thesis, namely when two consecutive signals arrive at the same level of importance. We will improve decision-making and make it more effective in order to benefit the network's overall performance, we will call the new algorithm what we are designing (Deep Priority Queuing) because it makes decision-making even deeper. It is worth mentioning that while the proposed approach will be effective in some scenarios, it will be less effective or perhaps ineffective in others. However, it is beneficial to apply it to the network in order to improve overall performance. Consider the following network scenario.

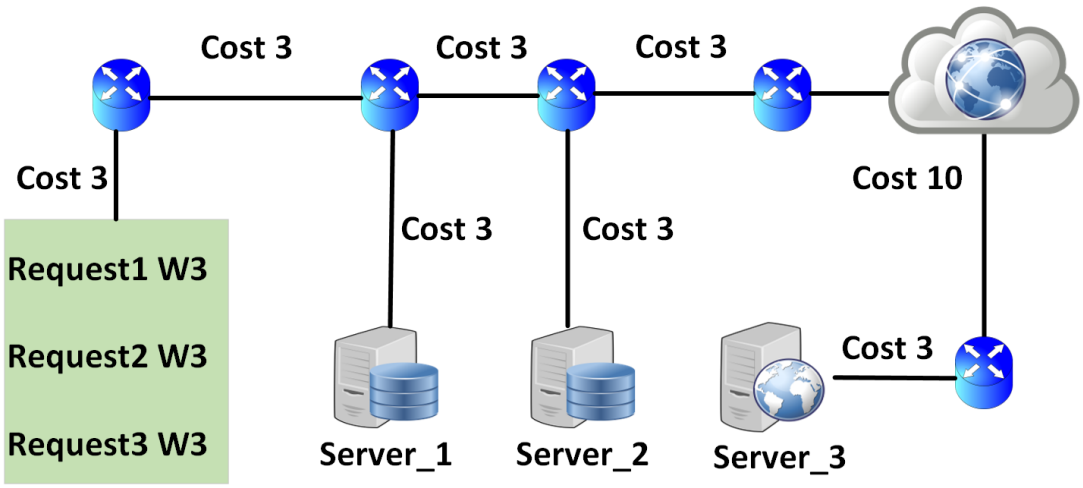


Figure 21 Deep Priority Queuing

Nota bene: The numbers in this diagram are not accurate; they are merely designed to demonstrate the concept. We notice that all three requests have the same priority level, implying that they are equally important, yet each takes a different routing. Signal No. 1 requires a cost of six to reach server No. 1, whereas signal No. 3 requires a cost of twelve. While signal number three requires a cost of 25 to reach server number three. Assume that each signal requires a processing cost of five. We desired to send messages in the logical order 1,2,3. As a result, the following will occur:

Request 1 => server 1 = Process cost + rout cost = 5 + 9 = 14
Request 2 => server 2 = 2* Process cost + rout cost = (5+5) + 12 = 22
Request 3 => server 3 = 3* Process cost + rout cost = (5+5+5) + 25 = 40

We observe a significant difference in the cost of each signal, despite the fact that all signals are of similar relevance. If the cost represents the delay time, this means that signal No. 3 has been delayed about three times as long as signal No. 1, despite the fact that both signals are of the same significance. As a result of this scenario, we can conclude that the state of fairness has been lost between the signals. Now we'll reverse the processing procedure and compute the cost. This time, we'll begin by processing the signal with the greatest distance, number 3,

followed by the signal with the closest proximity, number 2, and finally the signal with the closest proximity, number 1.

Request 3 => server 3 = Process cost + rout cost = 5 + 25 = 30
Request 2 => server 2 = 2* Process cost + rout cost = (5+5) + 12 = 22
Request 1 => server 1 = 3* Process cost + rout cost = (5+5+5) + 9= 24

Take note that the end results in the second case are more similar to those in the first, implying that the second scenario is more stable or fairer than the first. Thus, processing a signal that must travel a long distance or incur a high cost should come first, followed by processing the signal that must travel a short distance or incur a low cost, and so on. Given the length of time required for this type of signal to reach its destination, it is illogical to delay it further during the processing process. Unlike signals that travel to a nearby or low-cost route, where it is possible to delay processing of these signals while the more distant signals with the same level of priority are processed. The system is made as fair as possible by taking into account the signal's probable route and the cost of that route. The system must prioritize and process the most costly signals first, before proceeding to the next signal, and so on. A question may emerge at this point: how is the cost of a potential path determined? The answer is already implemented in routers via a variety of algorithms, including (OSPF, EIGRP), which determine the shortest or easiest path for the signal after it has been filtered for transmission by the system. [23] We will create a simple modification to the original algorithm, in which the path will be calculated and the shortest path for the signal will be picked first, followed by the signal being filtered at the cost of its journey.

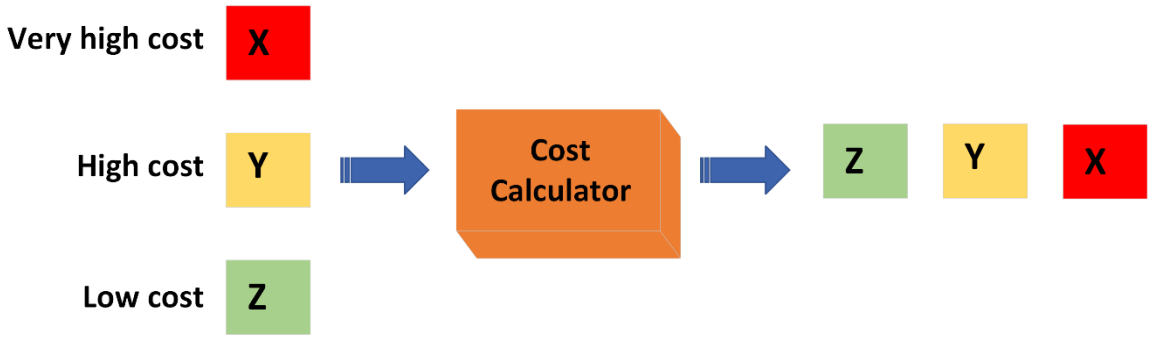


Figure 22 The new order of request for DPQ

And then another question arises: what if we did extra calculations, extending the signal processing time and resulting in a longer delay? The fact is that we did not introduce any new computational processes; rather, we submitted and delayed certain activities at the expense of others. The system would select the signal and then determine the optimal path for transmission. The new algorithm determines the optimal path for the existing signals and then sends the most costly first, followed by the easiest, and so on. This strategy increases the network's fairness. Prior to transmission, the route calculation step can also be used to cancel some signals in specific circumstances where it is necessary to cancel some packets to reduce network load. It is possible to remove packets with a longer route or a greater cost, as these

signals arrive at the destination too late to benefit the recipient. If the system is given an option between two signals to delete, it must delete the one that is the most distant or has a high cost.

3.9 Packet Drop method

Several ways are available for removing packets from the queue. We will discuss the most often used method in networks, referred to as weighted random early detection (WRED). This approach deletes packets from the queue based on their weight or priority, with the lower priority packets being removed first. This raises the question, what is the benefit of removing some packets? How to enhance network performance by removing some packets and avoiding the congestion problem. The purpose of deleting certain requests from the queue is to keep the network operational. This technology ensures that the network continues to run smoothly for important or high-priority signals, even in the event of a network overload. Consider the following situation:

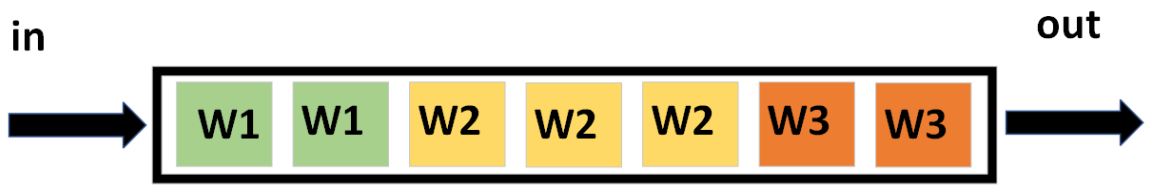


Figure 23 Full queue with 7 request's size

Seven requests are currently queued for processing. These requests are listed in order of priority, starting with number three, which is the most important, and ending with number one, which is the least important. But we note that the memory is full assuming that the memory capacity is only 7 requests. What happens if a new request at level 2 or 3 is added to the queue? This new request will be denied due to memory exhaustion. This is referred to as tail drooping. In this instance, there will be an obvious hiccup in the network's operation, since level 1 requests will be processed while level 2 or 3 requests are deleted. What happens if one of the first-level requests is deleted, leaving at least one empty slot in the queue?

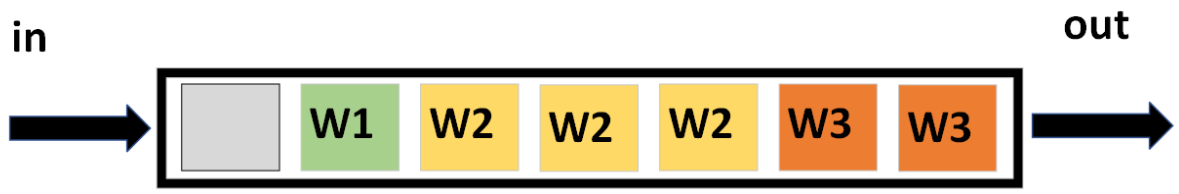


Figure 24 The queue has space for one more new request

In this situation, the new request will be processed and assigned the proper priority level; it will be placed after level 2 packets if it originates from level 2, or after level 3 requests if it originates from level 3. From this point forward, the concept of eliminating Certain memory

elements maintains a clear space in the queue to allow for the processing of signals of high importance. A fixed percentage of the queue's free memory is set, for example 75%, to ensure that 25% of the memory remains free and ready to receive new requests. If the queue's data content surpasses 75%, it will delete some elements (lower priority) in order to retain 25% of available memory space. Another advantage of adopting the method of eliminating insignificant parts is that it instructs the sending party to lessen the rate of data transmission in order to alleviate the receiving party's strain. This is a relatively regular occurrence due to the inconsistency of network devices' data transfer rates. One device operates at a 100 Mbps rate, another at a 10 Mbps rate, and another at a 10 Gbps rate, and so forth [24]. If we examine the above example attentively, we can see that the queue has been filled with requests from the third level, and that newer requests are now entering the system from the third level as well. The device needs to delete certain items from the queue in this case, but there is a basic issue: all requests in the queue are of equal priority. In this circumstance, the existing algorithms will randomly delete a value or will destroy the queue's last value. However, we can also utilize data analysis to eliminate requests that travel to remote locations because the signal will arrive too late for the recipient to benefit from it. As a result, in this scenario, it is best to erase it. It is worth emphasizing here that when relying on the notion of deleting packets that travel to remote devices and incur a significant cost, the technique relies on this principle. Consider the processor speed of the router and switch. If we wish to determine the packet's destination before removing it, this adds additional processing time, in contrast to passing. We discovered that while sending the farthest and then the closest packets, the overall processing time remains constant. For example, if we process four packets and then send four packets, the total processing time remains constant. The situation will be different in the event of deletion. Processor time may be used attempting to determine the destination of a particular packet, and as a result, this packet is erased. If four packets are processed as in the preceding example, but one of these packets must be deleted, the processor will ultimately send three packets but will consume the processing time of four packets. Here, we see that time has been lost; this time is sufficient to process the full packet. If the CPU in intermediate devices such as switches or routers is not fast enough, this wasted time affects the network's performance in general, resulting in poor network operation. On the other hand, if the processor speed is great, the time required to process the packet is minimal, and its loss has no effect on the network's performance.

3.10 Hardware based congestion solution:

While we have already discussed a software solution to this problem, it is also feasible to optimize the hardware utilized in switches to speed up packet processing and routing. This is accomplished by modifying the architecture of these devices. What follows now requires an understanding of the switching process within switch devices; therefore, if the reader is unfamiliar with this process, they will likely have problems grasping the principle of work. You are likely to view a section of the switch as well as a practical demonstration of the process of data transmission between two devices before proceeding to this section of the thesis. Both of the thesis's proposed subjects have been previously discussed. As previously stated, the table is contained within the switch device. This table lists the switch ports together with the MAC

addresses of each device connected to each of them. For instance: (7 1D:EE:A9:F6:CC:3E) This indicates that a device is connected to the switch's port number 7, and that device contains the MAC address specified. The MAC address of the device is determined throughout the process of connecting to or disconnecting from the needed device. The previous example recorded the information about the device connected to port 7. By connecting one of the devices to the device on port 7, or by the device attempting to connect to another device that should be connected to the network. The benefit of adding device information to the MAC table is that it speeds up subsequent connection processes. That is, if you submit a packet to the switch and the switch is already aware of the destination device, the switch will deliver the data directly to the port specified in the table, rather than querying all devices. The act of searching through the MAC table consumes both time and energy on the processor's part. The information received at the port is retained in the switch's memory until the MAC table search is complete, at which point the process of routing the packet to the requested port begins. The proposed solution in this section of the thesis is to decrease the time required to search the MAC table and get it closer to real time. This is accomplished through the addition of hardware. This hardware performs real-time matching, search, and data routing. Several steps will be taken to clarify and simplify the concept. We'll start with the hexadecimal value one from the device's MAC address. We know that the MAC has a length of 48 bits, but for the sake of simplification, we will create the electronic circuit assuming that the MAC has a length of 4 bits. The similar concept may be used to the MAC address's true length, which is 48 bits, but with the addition of additional electronic components. In any case, at the conclusion of this section of the thesis, it will be shown how to implement and extend this solution to include 48 bits, which is the actual length of the MAC address.

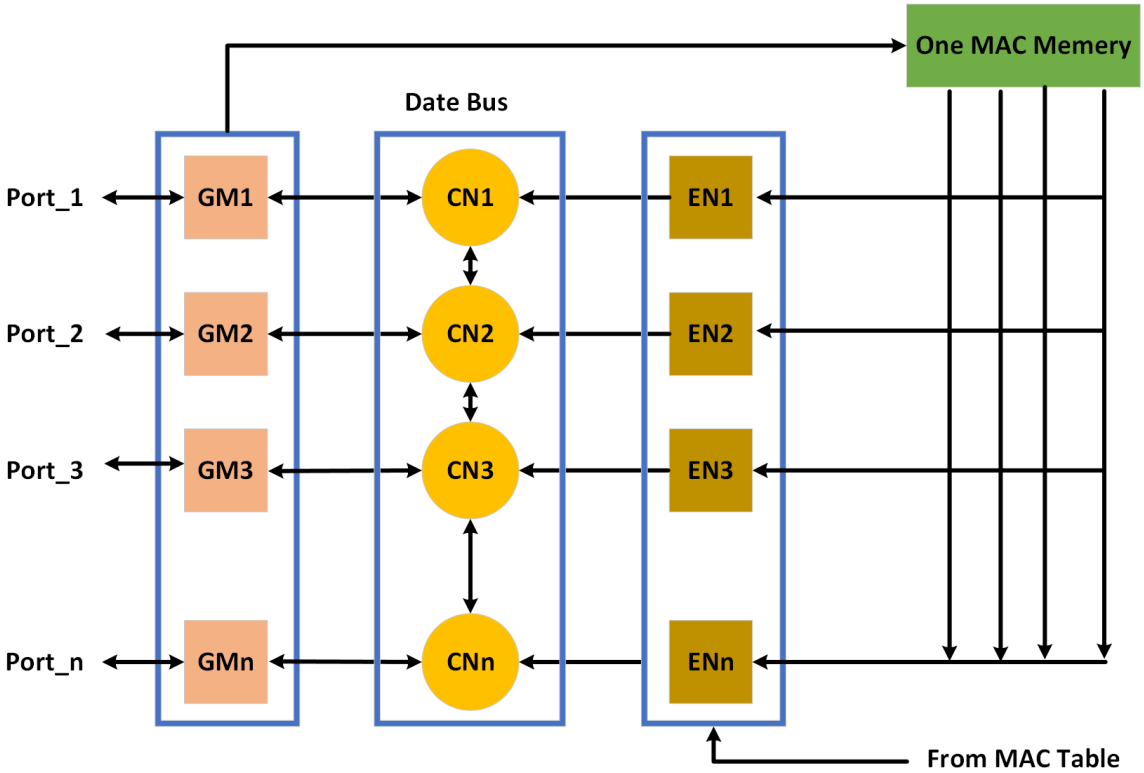


Figure 25 Hardware based switching architecture

As indicated in the preceding diagram, the structure is composed of four major components (Data buss, Enable Node, Connection Node, and the matching MAC "get MAC").

3.10.1 Data bus section

This section is used to connect all switch ports physically or electrically. The number of lines in this section indicates the number of simultaneous connections that the switch can create between ports. If we use a single line, the system circuit can connect just two ports, and the remaining ports must wait for it to complete. The process of forwarding packets between these two ports, followed by the establishment of another connection between two further ports, and so forth. For instance, suppose a packet from port 3 wants to interact with port 1, while an inbound request from port 7 wishes to communicate with port 20. The second request must wait until the initial connection between ports 1 and 3 is complete, at which point the channel or transmission line becomes available for another connection between ports 20 and 7, and so forth. If we wish to utilize more than one transmission line, we must build another mechanism to detect which channel is available for communication. For instance, if we utilize six lines and a connection is in progress on line 2, and the device receives another request between two other ports in the meanwhile, there must be a part of the system responsible for selecting the available line number for the new connection process. After the system is completely explained to function through one transmission line, we will demonstrate how to use several transmission lines in practice.

3.10.2 Connection Node circuit

This circuit is used to connect the port to the transmission line electronically. This circuit is activated by an instruction received from the Enable node circuit. This command is generated using the MAC address of the destination device contained within the packet.

Suppose that an incoming packet originates from port number 1 and that the destination device's MAC address is assigned to port number 3 in accordance with the MAC table. In this situation, the Enable Node 3 circuit will activate the Connection Node 3, which links port No. 3 to the transmission lines. The procedure of connection is depicted in the diagram below.

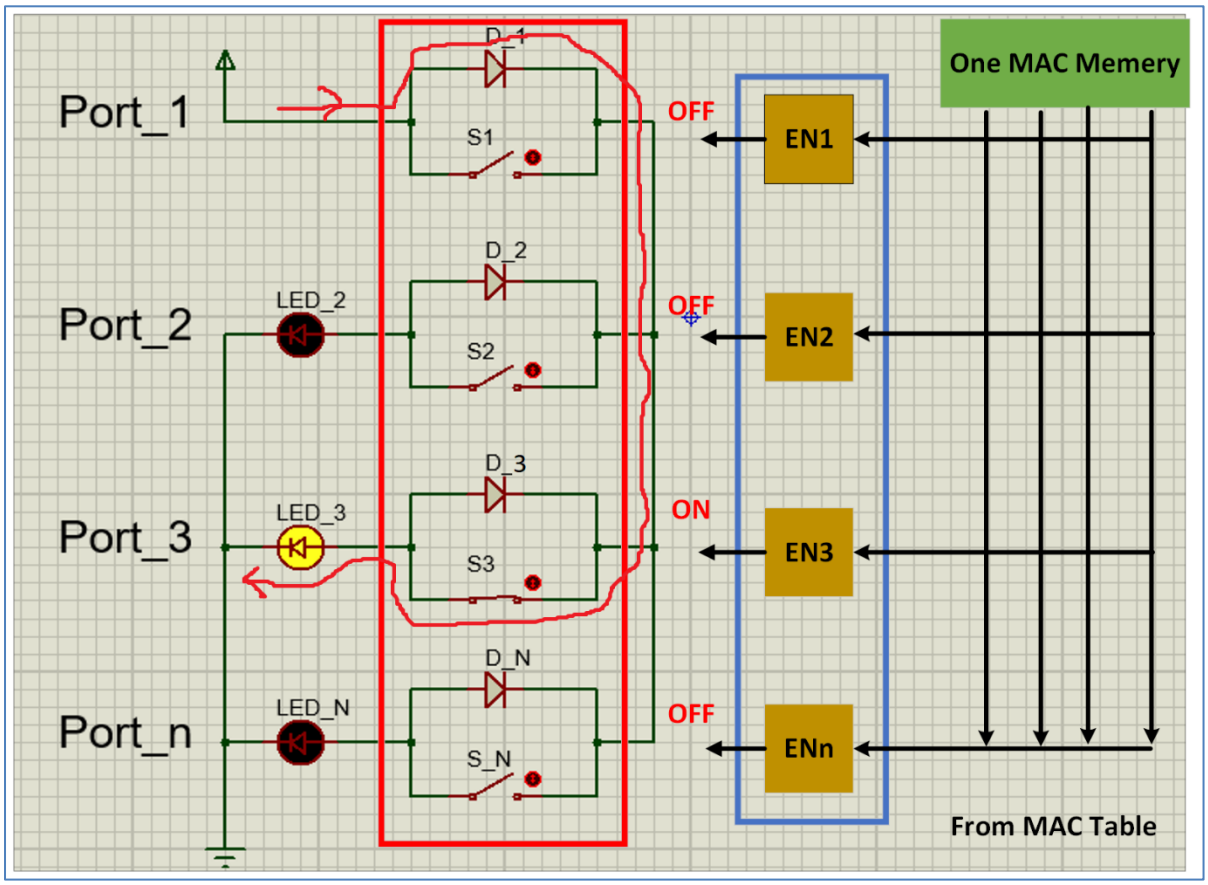


Figure 26 Connection Node circuit using mechanical switches

The data will flow in the direction indicated by the red line on the chart. The switch receives data via port No. 1, where it is routed to the transmission line via diode D1. The transmission line provides data to all ports, but only port No. 3 is connected through switch S3, as the activation circuit No. 3 is in the ON state, while the remaining circuits are in the OFF state. The remaining switches are in the OFF position. As a result, only port number three will connect with port number one, as is necessary. This situation is replicated at each subsequent port. In other words, just two ports are connected to the transmission line, while the remaining ports are isolated from it. Data from any port is routed to the data line via the CN circuit's diode. The transmission line is electrically connected to all CN units, however the EN circuit connected to it enables just one of the CN circuits. This leads to the establishment of an electrical connection between the data-receiving port and the data-sending port. It should be noted that the CN circuit is intended to be illustrative rather than practical. In practice, the

switch S is replaced with an AND gate, which performs the same function as a switch. The previous diagram used the key sign because it is more straightforward and understandable.

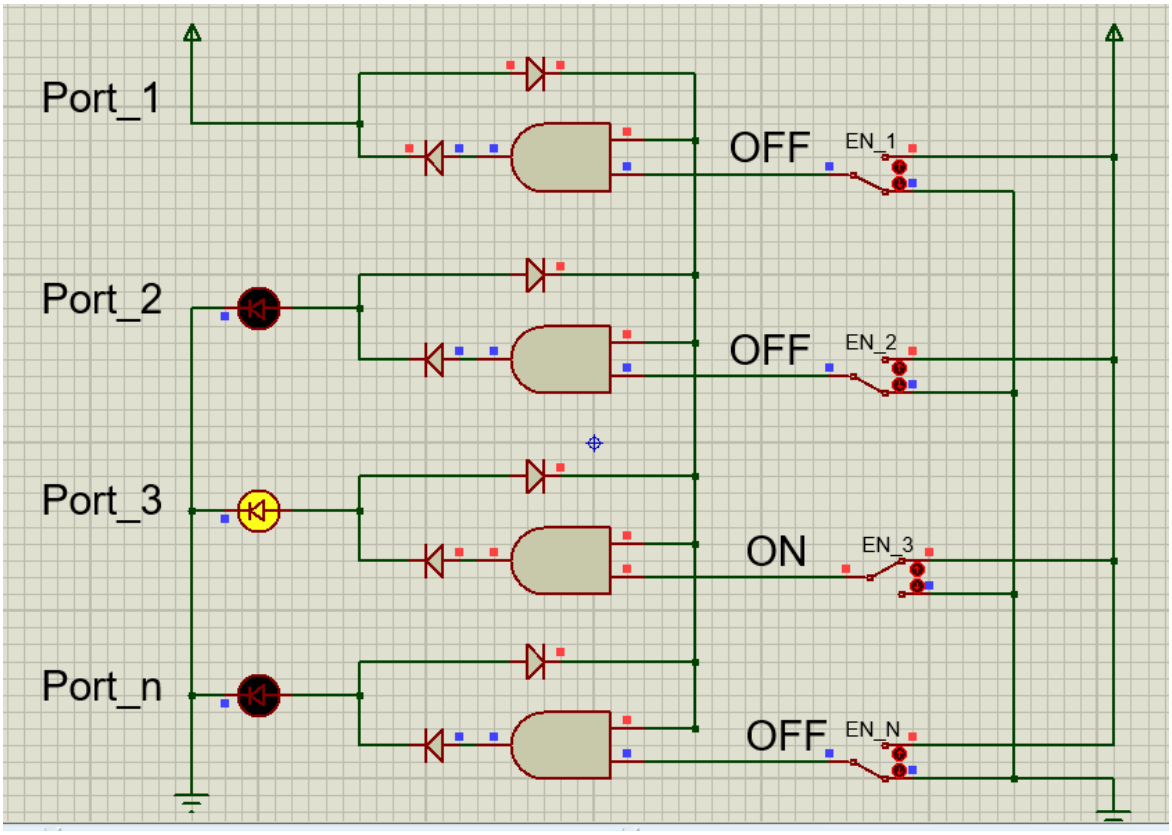


Figure 27 Connection Node circuit using AND gate as switch

A CN circuit controlling an AND gate is illustrated in the following image. If the activation signal from the activation circuit is zero or off, the AND gate output will be zero regardless of the transmission line signal; however, if the activation signal from the activation circuit is one, the AND gate output will be dependent on the information in the transmission line. In other words, the transmission line will be electronically and digitally connected to the port. When data is received into the switch, the diode associated to the AND gate output protects the gate from the electric current flowing from the port.

3.10.3 Enable Node circuit

The enable node circuit is a critical component of this system. It checks the received packet's MAC address to the MAC address stored in the circuit's memory. If these two conditions are met, the output of this circuit will be logical. If they do not match, this circuit will output a logical zero. This circuit's output is identical to the activation signal used by the CN circuit.

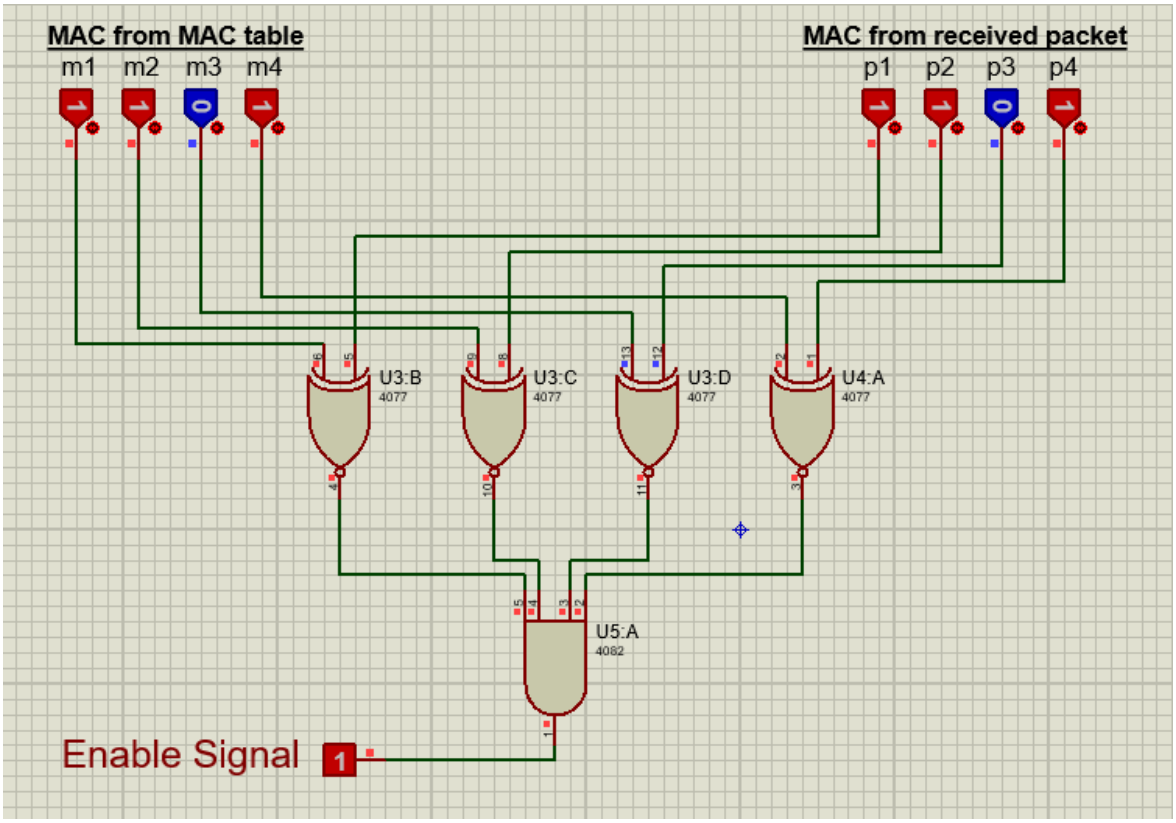


Figure 28 Enable Node internal circuit for 4-bit MAC address

The above image depicts the actual circuit for the EN component of the system. The values are kept in the switch's MAC table, more precisely in the node's memory. If the MAC table contains the following line (port=3, mac=1A:4B:E5:AD:CC:E2), the processor copies the MAC and stores it in the memory of the activation circuit EN3 for port No. 3. Naturally, the procedure is repeated for the remainder. The lines in the table where it copies the MAC and puts it in the memory of the activation circuit associated with that MAC based on the table and port number.

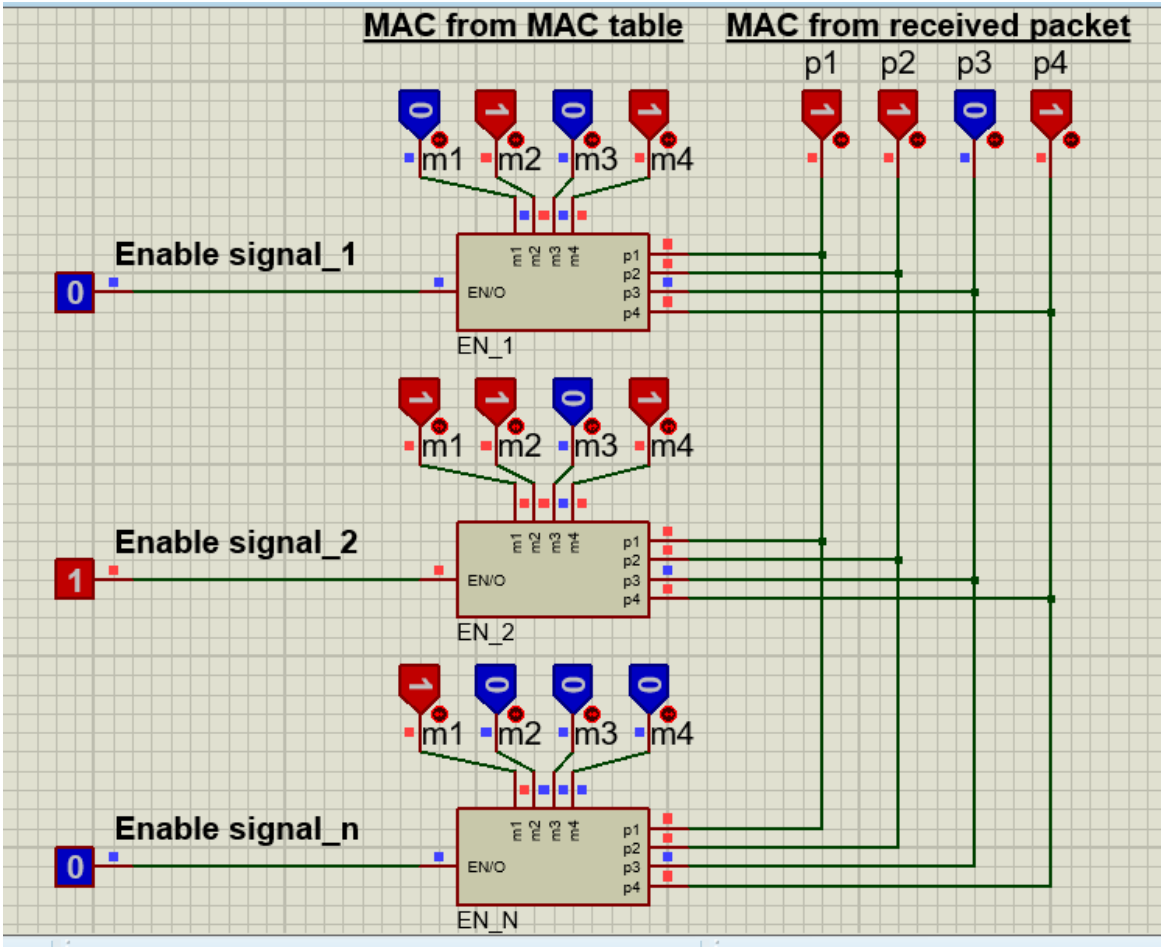


Figure 29 Enable Nodes 4-bit as blocks

When the switch gets a packet from the system via one of the switch ports, the system creates a copy of the destination device's MAC and stores it in the enable circuits EN's temporary memory. The enable circuits EN's temporary memory is a shared memory shared by all enable circuits ENs. If the MAC address of the incoming packet matches one of the MAC addresses recorded in the EN circuits, this circuit generates a logic 1 activation signal. This is accomplished using logic circuits comprised of XNOR gates and an AND gate. The accompanying illustration serves as a reminder of the truth table containing the equations for the logic gates utilized in this study.

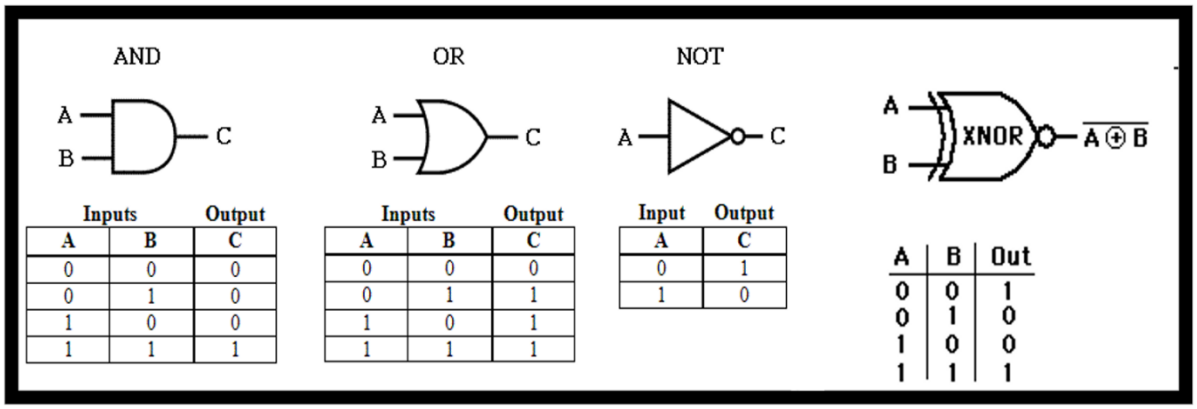


Figure 30 Logic gates truth table

Notably, the XNOR gate outputs a logical one if both entries are identical, but a zero if the two entries are dissimilar. The AND gate generates a logical zero if any of its entries is equal to zero, but only a logical one if all of its inputs are equal to one. For an OR gate, it produces one if any of its entries is equal to one and zero if all of its entries are equal to zero. The negation gate simply inverts the input value; if the input is one, the output is zero; if the input is zero, the output is one. [25] When the concept of logic gates is combined with the prior architecture of the enable circuit, the following results:
The XNOR gate produces the following output. X=1 only when m equals p. This indicates that the MAC address recorded in the EN circuit corresponds to the MAC address included in the incoming packet.

XNOR1 = m1 xnor p1 = X1
XNOR2 = m2 xnor p2 = X2
XNOR3 = m3 xnor p3 = X3
XNOR4 = m4 xnor p4 = X4

The AND gate output is via the following expression

AND(out) = X1 . X2. X3 .X4
Enable Signal = (m1 xnor p1) . (m2 xnor p2) . (m3 xnor p3). (m4 xnor p4)

If only X1=X2=X3=X4=1, the AND gate output is equal to one, indicating that all the bits in the EN memory match all the bits in the incoming packet. The matching procedure is logical, electronic, and electrical in nature and does not need a search of the switch's MAC table. This functionality enables the switch to operate in real time, without requiring the processor to perform any processing. Because the aforementioned EN circuit operates on a 4-bit MAC, we required four XNOR gates and one AND gate with four inputs. To construct this circuit and test it on the actual length of the Mac, we will want 48 XNOR gates and one AND gate with 48 inputs. The following is the method of connection.

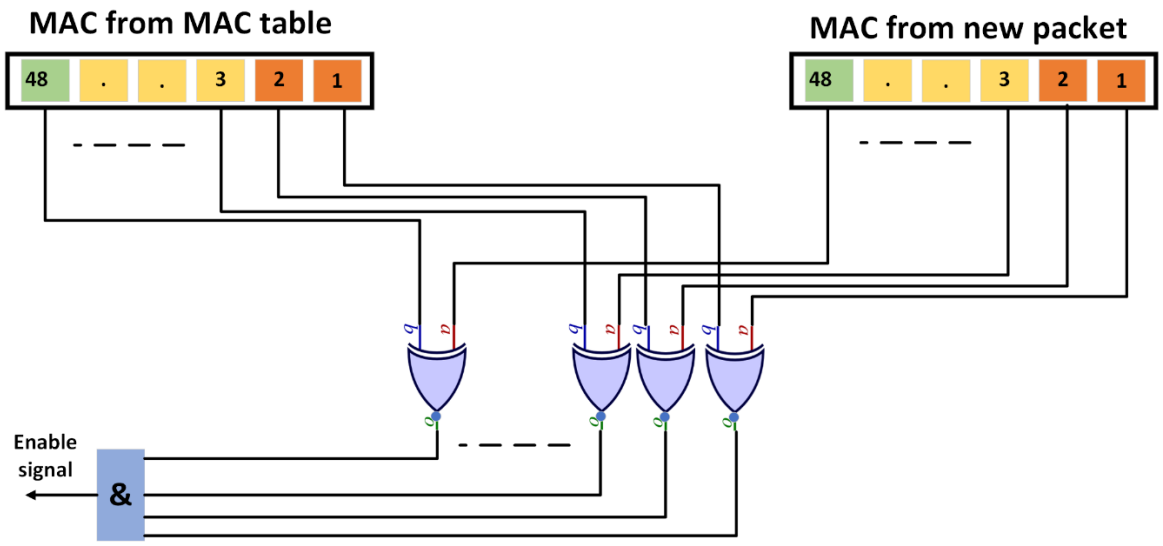


Figure 31 Enable Node internal circuit for 48-bit MAC address

The same equation as in the 4bit example holds true here, but with an increased number of terms in each equation. This configuration is effective if the switch's MAC table does not contain duplicate ports. Occasionally, as illustrated in the following graphic, more than one MAC is linked to a single port.

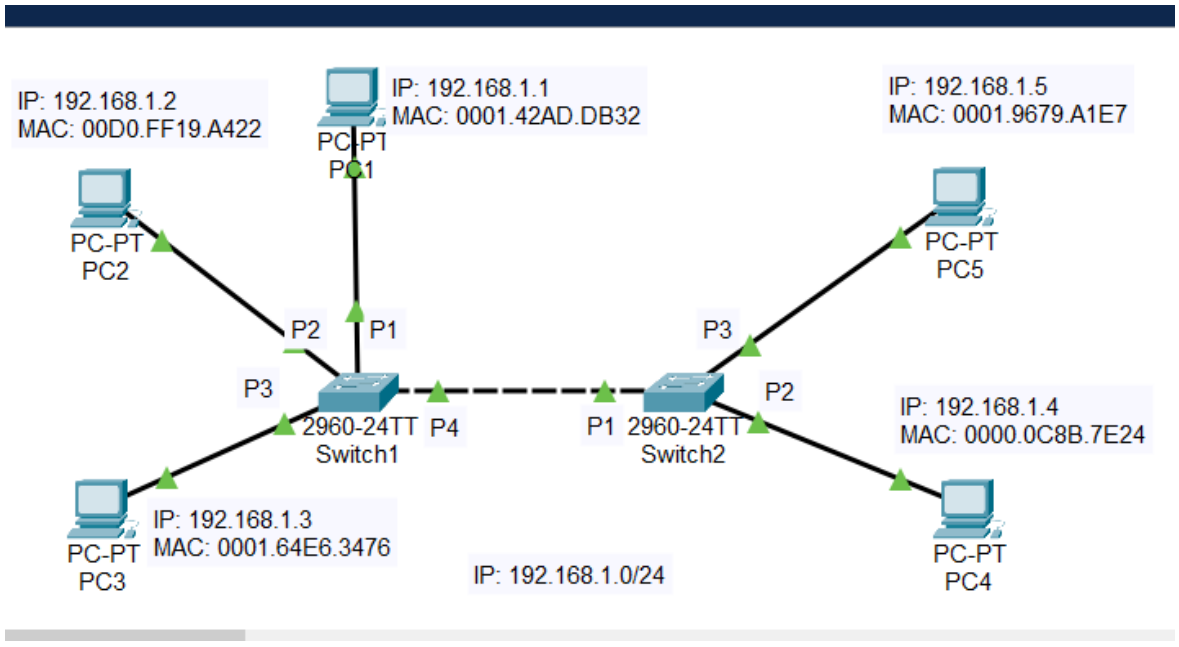


Figure 32 Network topology with two CISCO switches

The following table describes the MAC table in switches one and two.

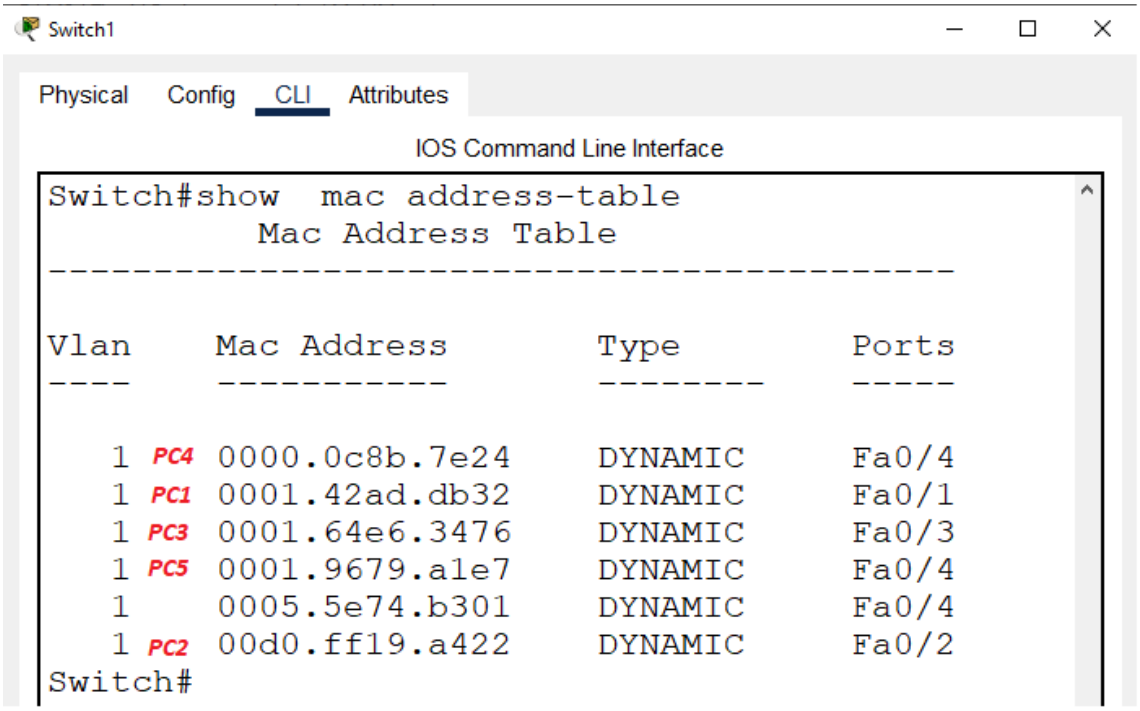


Figure 33 The MAC address table of switch_1

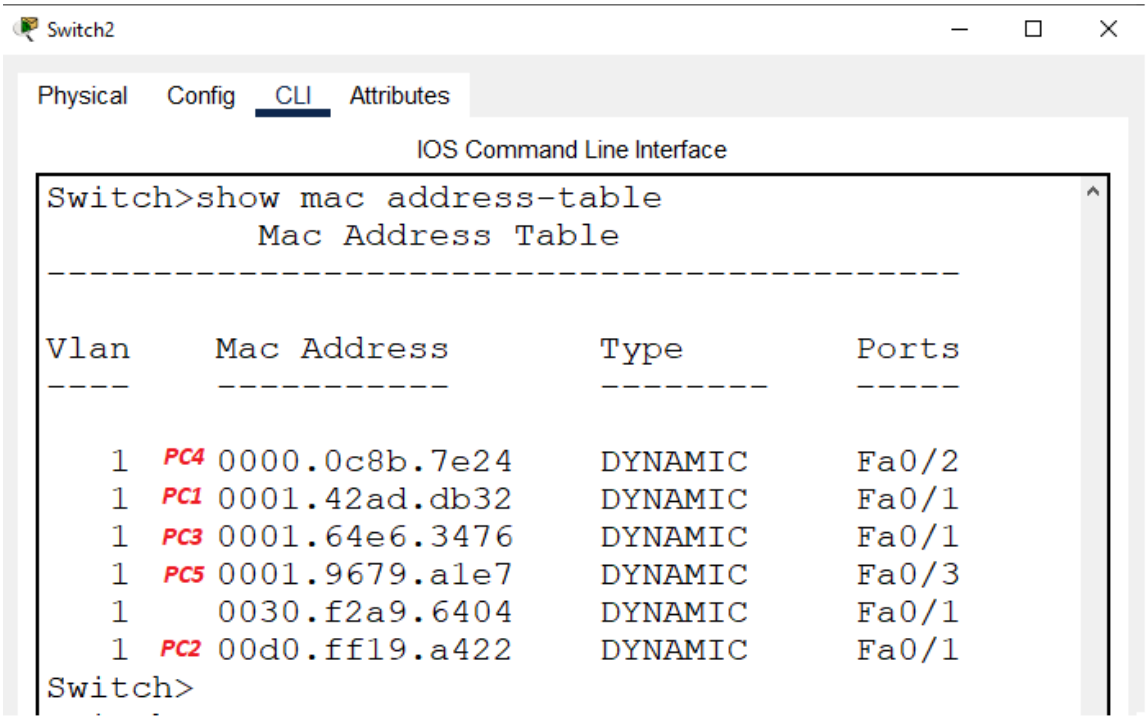


Figure 34 Figure 33 The MAC address table of switch_2

On both switches, we'll see that the port number is repeated. Port number one serves devices (1,2,3) in switch number one because each of them is connected to a single switch via port number one. If one of the devices (4,5) wants to communicate with one of the devices (1,2,3), the communication must pass through switch port number one. As a result, the MAC table records port number one three times. The same is true for the switch's fourth port. If one of the

devices (1,2,3) wants to communicate with another device (4,5), the connection must go through port number 4 on switch number two. The concept is that the circuit in the preceding image directs the incoming packet to a specific port if that port appears only once in the MAC table. Because the memory can only hold one MAC and the logical circuit can only match one MAC, they can only be matched. The question is how to enhance the preceding circuit to make it work when more than one MAC is attached to a single port. To illustrate the answer to this problem, we'll take the circuit schematic that works with four bits and change it to work with multiple MACs. The same technique can be used for a circuit that manages the 48bit MAC's real length. The approach is to include many sections of EN memory as well as the circuits generated by the XNOR and AND gates. The following figure illustrates how to modify the activation circuit so that it runs on a MAC with a 4 bit length.

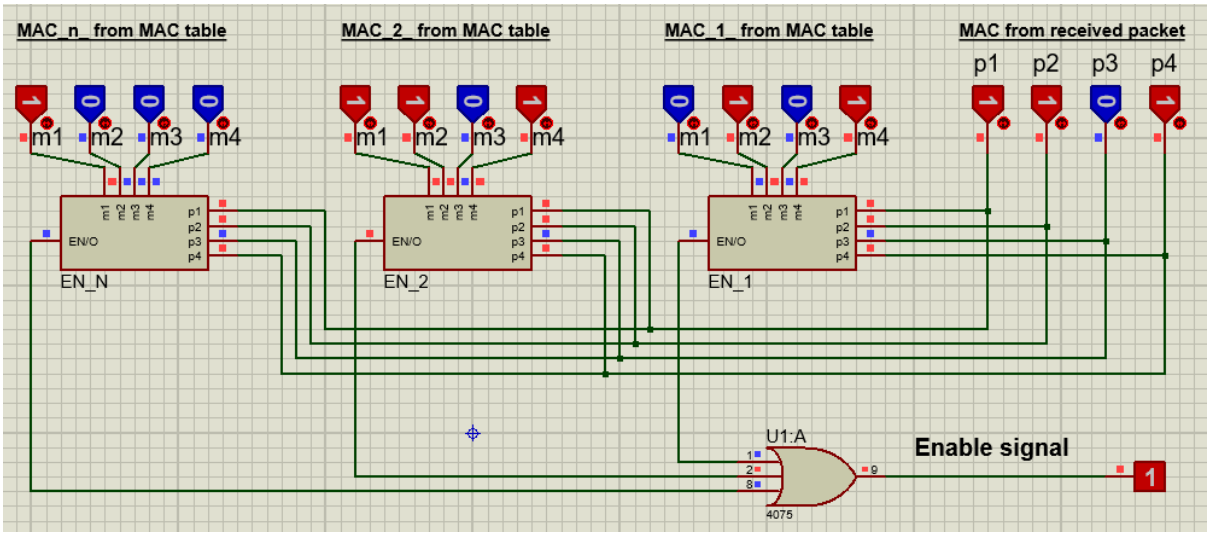


Figure 35 Multi 4-bit-MAC Enable Node circuit

This circuit is split up into different blocks that operate independently of one another throughout the matching process. The number of sections is determined by the maximum number of devices that can be connected to a given port. It is obvious that increasing the number of sections increases the manufacturing cost due to the increased electrical energy consumption of the device, but in exchange, we gain a high degree of flexibility in using the device, as the increased number of sections has no effect on the device's performance or data routing speed. As illustrated in the preceding chart, the MAC of the destination device is compared in real time and in parallel to all the MACs stored in the clips' memories. Each portion generates a signal of type 0 or 1. If there is a match, the section's output is a logical one; if there is no match, the section's output is a logical zero. The outputs for the OR gate sections are gathered. The OR gate outputs a one if one of the sections' outputs is equal to one, and a zero if all sections' outputs are zero. The OR gate's output reflects the enable signal that is used to activate the post's conduction node (CN) circuit. The following are the logical equations that describe how this circuit works.

out_1 = (m11 xnor p1) . (m12 xnor p1) . (m13 xnor p1) . (m14 xnor p1)
out_2 = (m21 xnor p1) . (m22 xnor p1) . (m23 xnor p1) . (m24 xnor p1)

…..
out_n = (m_n1 xnor p1) . (m_n2 xnor p1) . (m_n3 xnor p1) . (m_n4 xnor p1)
Enable signal = out_1 + out_2 + out_3 + out_4

The number of gates necessary to construct a multi-segment enable circuit is mostly determined by the number of devices that will most likely connect to a single port. If we wish to create a circuit that works when the MAC's true length is 48 bits and the number of devices attached to a port is n, we must do the following:

48*n number from XNOR gates
n number from AND gates with 48 inputs for each one
1 OR gate with n number of inputs

These are the gates for a single port. If the switch has K ports, the preceding values must be multiplied by K.

48*n*K number from XNOR gates
n*K number from AND gates with 48 inputs for each one
1*K OR gate with n number of inputs

This can be quite costly, as in most circumstances, just one or a few devices are connected to the switch port. A switch with two or four specific ports is possible to manufacture. These ports are capable of supporting the connection of a huge number of devices. Each of the four ports contains 1,000 segments on the inside, which means that each of these ports is capable of communicating with 1000 separate devices. The remaining ports in the switch can contain ten or twenty segments from the inside, which means that each of these ports can support ten or twenty connected devices. The preceding figures are illustrative but not exhaustive. While it is true that if all ports support a big number of devices, the gadget will be superior, the production cost may also be high.

3.10.4 Getting MAC from the received packet

Extraction of the destination device's MAC address from the received data packet: The primary function of this section is to extract the destination device's MAC address from the received data packet and then generalize it to the ENs units so that it can be matched with the MAC address stored in the memory of each of the units. Only one enable circuit (EN) should match the generalized MAC and thus generate an activation signal in accordance with the circuit sequence in the device representing the same port sequence. For instance, enable circuit number six is related with port number six, while enable circuit number four is associated with port number four, and so forth. This component of the system is comprised of the following: (shift register, digital counter, isolation circuit and delay circuit). As illustrated in the accompanying graphic.

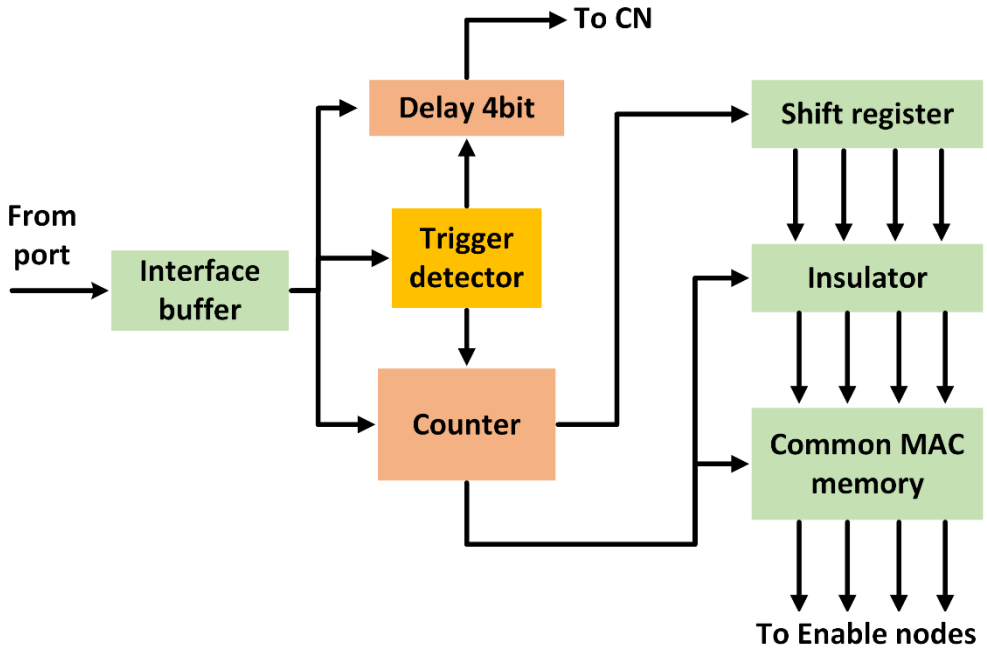


Figure 36 Getting MAC from the received packet block diagram of the circuit

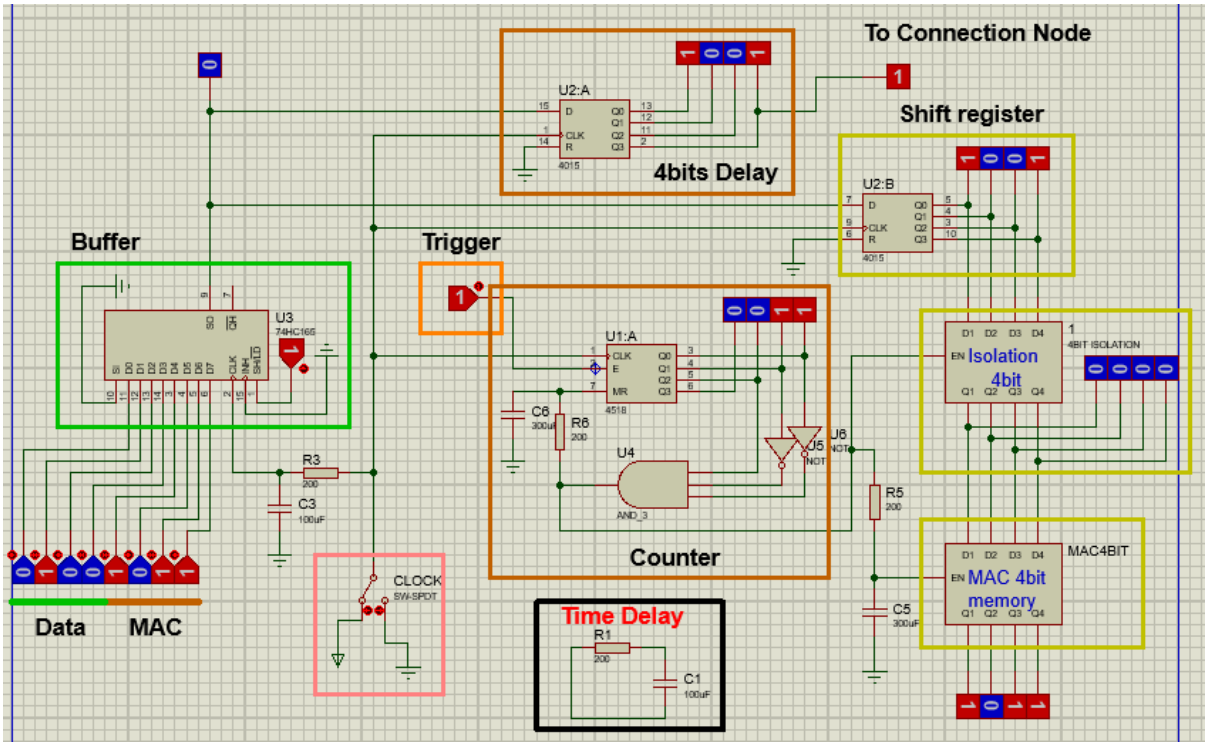


Figure 37 Getting MAC from the received packet circuit

The preceding graphic illustrates a straightforward architecture for the system, which involves extracting the MAC address of the destination device from the received packet. The following figure depicts the optimal shape of a packet of data transmitted via a network.

Figure 38 The ideal packet structure for the system

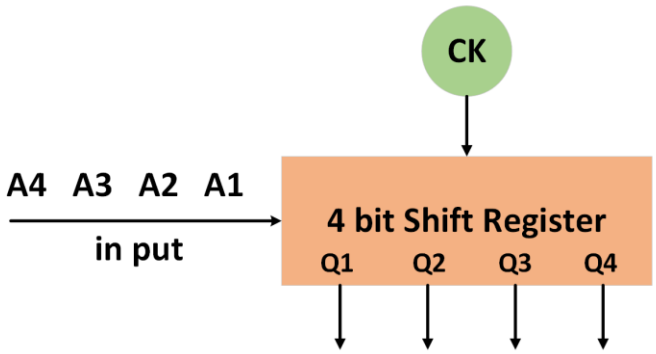
Although this format is not currently utilized in networks, the system we are developing performs optimally when the packet information is organized as indicated in the preceding figure. In any instance, the system operates regardless of the location of the destination device's MAC address within the received packet, however the closer the MAC address is to the packet's beginning, the faster the system responds. For instance, if the MAC is 48 bits in length and is present at the beginning of the packet, the system requires the same amount of time as it takes to pass 48 bits of data for the device to begin then routing the packet to the proper port. To connect the ports involved in the routing procedure, the system must delay the signal by a specified number of bits until it reaches the MAC address of the destination device. If the MAC address in the packet is after 100 bits of data, the system requires a time of 100 + 48 = 148 bits to determine the packet's destination. In other words, the system requires the time necessary to pass the MAC plus the time required to pass the data preceding the MAC in the packet sequence. We will implement the simulation assuming that the received packet is as shown in the preceding picture, which means that the MAC is at the head of the packet in order to compute the system's best performance. The packet depicted previously is made up of four components, the first of which is the trigger, which is a digital signal with a predefined width. This signal is used to notify the system that new information is on its way. The second section provides the MAC for the detention device, which is designed to be 48 bits in length but will be reduced to 4 bits to simplify the remainder of the system parts. The third component is the source device's MAC address, which is largely irrelevant at the moment. The fourth section contains the remainder of the packet's data. The fourth section may contain IP addresses, protocol types, encryption keys, image or audio data, and so on. It is also irrelevant during the design process. The preceding packet is currently stored in the interface buffer, which serves as a temporary storage device for data entering the switch port. The buffer circuit will simultaneously transmit data to the trigger circuit, the counter circuit, and the delay circuit. The trigger circuit's job is to detect the trigger signal at the beginning of the packet and instruct the counter to begin bit counting, which is then sent to the shift register. The trigger signal is unique and should not exist in the data, as it is a logical one for ten nanoseconds, followed by a logical zero for three nanoseconds, and lastly a logical one for two nanoseconds. While another strategy for triggering is feasible, we will employ this one in this thesis. The digital information must be fully distinct from the trigger signal, with each bit having a time longer than 15 nanoseconds, which is greater than the period of the trigger signal, in order for the system to function properly and identify the trigger signal from the original information. We reiterate that the numbers used are fictitious, but the concept is the same in all circumstances.

Delay Circuit: It delays the signal from the buffer by the same amount of time as the destination device's MAC information. Given that the MAC is considered to be four bits long,

this circuit will delay the signal by four bits. After receiving an activation signal from the trigger circuit, this circuit begins to operate. In this circuit, a shift register with a length of four bits can be utilized to accomplish the same task.

Counter circuit: This circuit also begins operating when it receives an activation signal from the trigger circuit. This circuit is based on the amount of data sent to the shift register. When the counter reaches 4, it sends one logical activation signal to the isolation circuit and another to the shared memory, at which point it resets to zero and ceases operation; it then remains in the condition of waiting for a new signal.

Shift register circuit: The shift register is a component of the system that converts MAC data from serial to parallel format. Shift registers come in a variety of configurations (Right circular shift, left circular shift, linear right shift, linear left shift... etc.). The following figure illustrates the operation of our present system's four-bit right linear shift register.



NO.	Q1	Q2	Q3	Q4
0	0	0	0	0
1	A1	0	0	0
2	A2	A1	0	0
3	A3	A2	A1	0
4	A4	A3	A2	A1
5	0	A4	A3	A2
6	0	0	A4	A3
7	0	0	0	A4
8	0	0	0	0
9	0	0	0	0

Figure 39 Shift register truth table

The table illustrates the state of the shift register for each clock generator pulse. All outputs are initially set to zero. At first glance, the shift register is loaded with the first A1 bit of data and stored in the first bit. The second pulse shifts the first bit A1 to the second bit and inserts the second bit A2 in its place. The third impulse shifts A1 and A2 and replaces A3, and so on for the remainder of the cycles.

As seen in the previous table, at pulse number four, all serial data A1,A2,A3,A4 is present on the shift register Q1,Q2,Q3,Q4's outputs. As a result, the counter was set to stop at 4 because this is the point at which the MAC is changed from serial to parallel form and stored on the shift register's outputs. As a result, the counter comes to a halt and resets to zero when an activation command is issued to the isolation circuit and shared memory of the MAC.

Isolation circuit: We found that the shift registers circuit's outputs are constantly changing. Only when the fourth pulse matches the input signal are the remaining signals wrong. To address this issue, an isolation circuit has been installed with an output equal to zero on all outputs independent of the signal on the input. When the counter reaches 4, the number on the shift register output is the MAC's true number. The isolation circuit is engaged in this case by the counter circuit, and its output is equal to the input. Meanwhile, the output of a circuit is sent to a shared MAC, which retains the MAC value for an extended period of time. (Unless a

different value is received). The isolation circuit retains the output for a very little period of time, typically less than one bit, before returning to the zero state, when all outputs are zero. There are other methods to create this circuit, but we will go with the simplest. The following illustration depicts a four-bit isolation circuit.

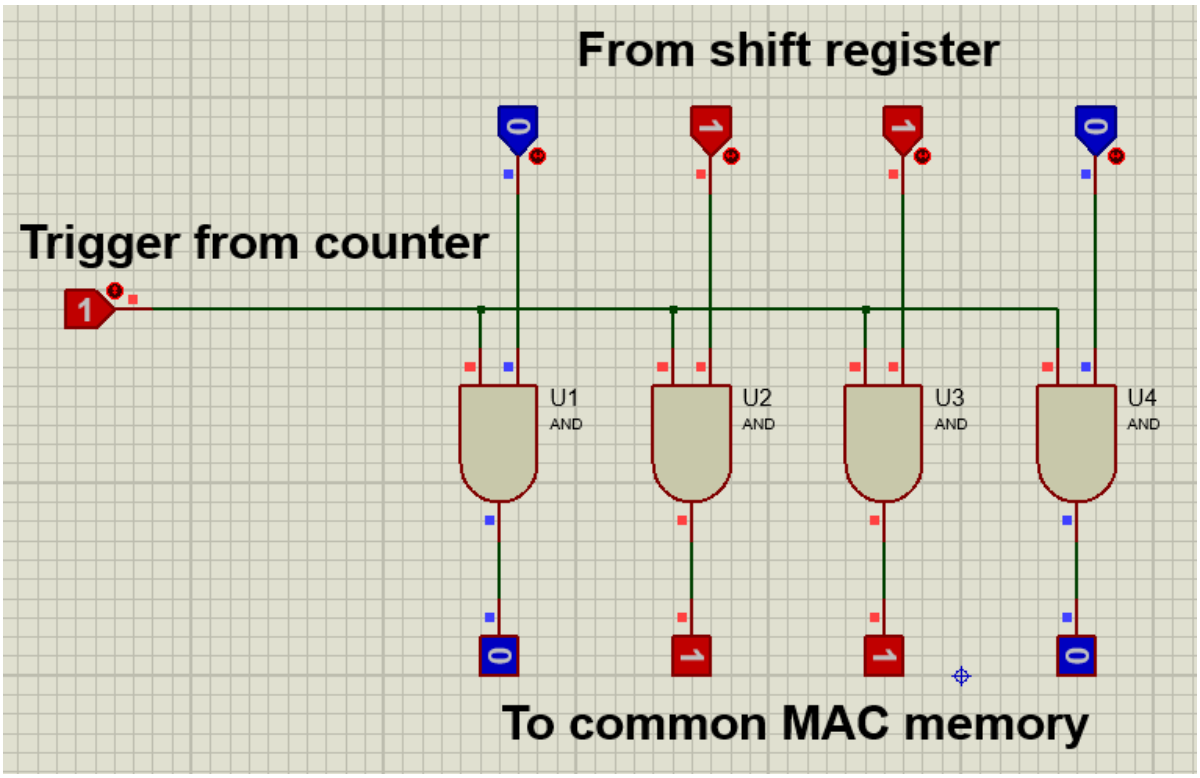


Figure 40 Isolation internal circuit

If the enable signal is zero, all AND gates output zero; however, if the enable signal is one, the gates output equals the input states D1, D2, D3, D4. The enable signal equals one only when the counter reaches four; otherwise, it equals zero. When the counter hits four, the isolation circuit transmits the shift register's outputs to the shared MAC memory circuit, where the value is stored. After that, the MAC is sent to the matching circuits. The isolation circuit must be enabled concurrently with the shared MAC memory, with the isolation circuit enabled first, followed by the MAC memory, and then the isolation circuit deactivated. If a sync error occurs, it will result in a significant system failure. If the isolation circuit is active and then removed prior to activating the MAC, this results in the shared MAC storing zeros, as the isolation memory output was zero when the shared MAC was activated. This synchronization issue can be resolved by including delay circuits in the path of the triggering signal, ensuring that one circuit is engaged before another.

3.10.5 MAC shared memory

This is a 4-bit memory (The size of memory depends on the length of the MAC used). This memory maintains the target device's MAC address for a specific duration, until the packet is forwarded to the correct port, at which point the memory is cleared and configured to receive a new MAC address. There is no problem if this memory is not cleaned because the new MAC replaces the old MAC in this memory. This memory operates similarly to an isolation circuit,

but the distinction is that this memory is retained in the output even in the absence of an activation signal. As with the other components of this system, the shared MAC memory can be implemented in a variety of ways, but we will employ D-type flip flops. We require four D flip flops, which corresponds to the number of MAC bits used. The internal structure of this memory is depicted in the following diagram.

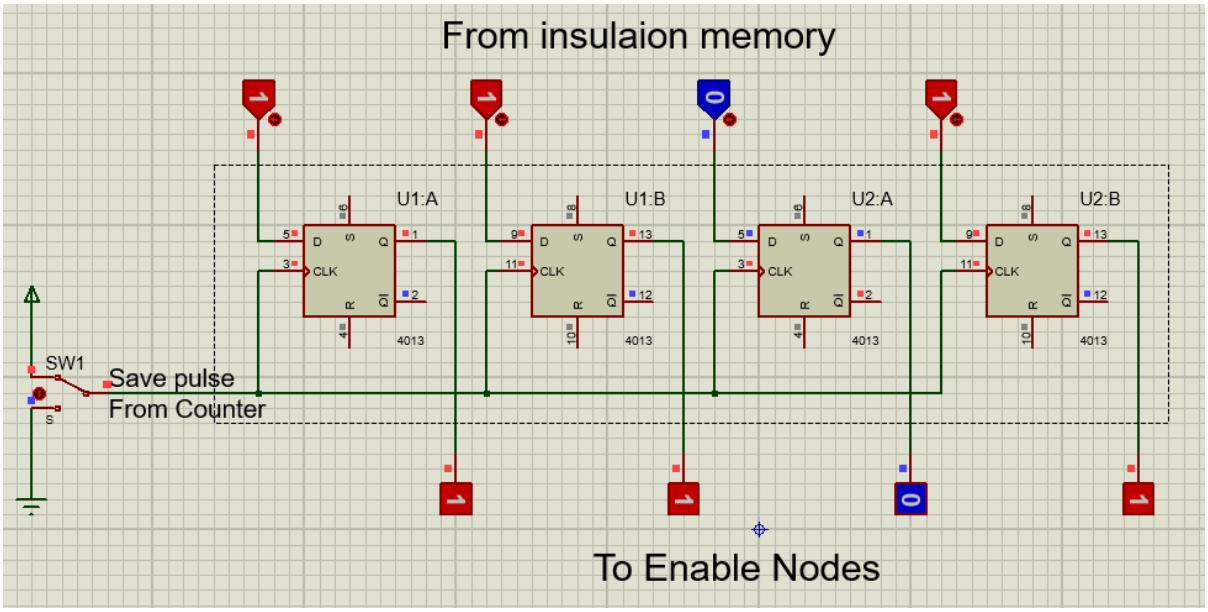


Figure 41 MAC shared memory of 4-bit MAC address

The preceding figure depicts the shared MAC memory's internal structure, including the D-flip flop's truth table. We notice that the memory's internal structure consists of four type D flip-flops. We can see from the truth table of Flip flop D that when no clock pulse is applied to the flip-flop CK port, the flip-flop output is identical to the preceding case regardless of the input value. When the flip flop receives a clock pulse via CK, the flip flop's output equals the value on the input port D. In other words, the output is equal to the input when there is a pulse (Q = D). All clock ports are connected in series to function concurrently and in parallel.

The common line connecting the clock pulse ports of the D flip flops functions as the memory's activation signal line. When the memory receives an activation pulse, the four flip-flops work cooperatively to store the signal on the inputs through ports D1, D2, D3 , D4 and to send it to the memory output ports Q1 , Q2 , Q3 , Q4. Without an activation pulse, the output ports retain their original value regardless of the input value change. This memory remains associated with the MAC until a new packet is received at the switch, at which point the aforementioned circuits extract the new MAC and transfer it to this memory, along with the activation command from the counter, and so on. The components depicted in figure 36 are unique to each switch port, with the exception of the MAC memory, which is shared by all switch ports. As a result, a tiny portion of the MAC memory must be added to enable communication with all switch ports in order to prevent interfering with the information received from the ports. If MAC1 originates from port number 5 and MAC2 originates from port number 8, for example. Two values will overlap in this case because the MAC gets both MAC1 and MAC2 values concurrently. Even if we assume that each port operates independently, if one of the MACs attached to the memory input is equal to (0000), this will influence the real value of the MAC in memory, even though

the value of the MAC will be taken electrically and converted to a new value, but this is false. To address this issue, an OR gate network must be added to each memory entry, as illustrated in the following image.

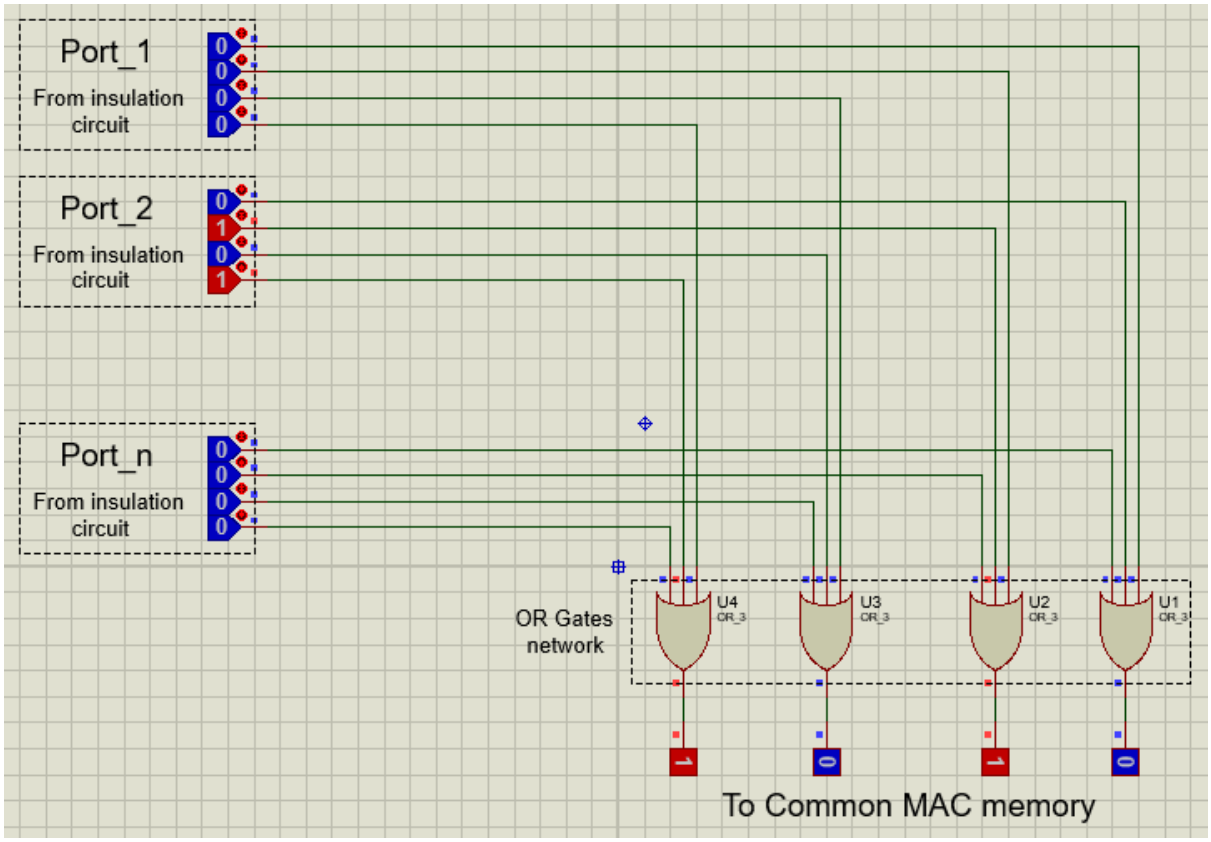


Figure 42 OR gates network

A MAC memory circuit with a matching stage is illustrated in the figure (OR gates). If all MACs except one are equivalent to (0000), just this MAC is stored in the MAC memory, and thus the interference problem is resolved. The number of OR gates necessary for this purpose is proportional to the MAC's length. We used only four gates because we were using a four-bit MAC. In a real-world implementation, the MAC is 48 bits long, necessitating the use of 48 OR gates. Each gate has a fixed number of inputs determined by the number of switch inputs. If the switch has sixteen ports, the OR gate must have sixteen entries, and so forth.

3.11 Simulation example

We have now thoroughly explained every component of the system. Now we will replicate the process of data transmission between two network devices via a switch, however this switch will be based on the system that we developed in the previous chapters. We'll be utilizing the following network.

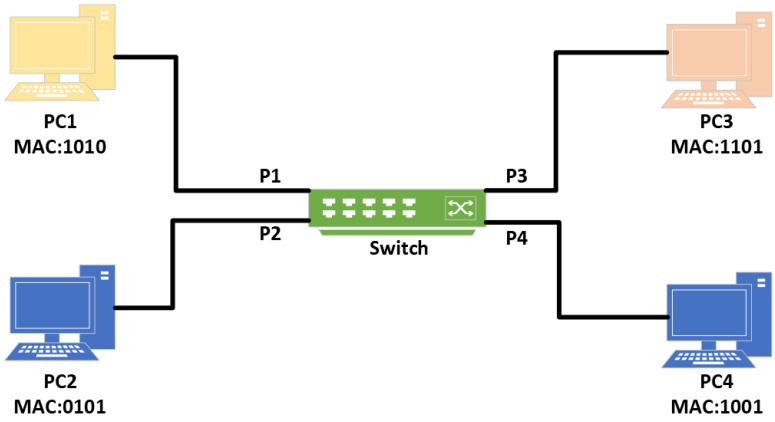


Figure 43 Star topology for simulation example

We will now proceed with the transfer from device PC1 to device PC3. This time, we'll concentrate on the operation of the system we designed, rather than on the transmission process as a whole. To simplify and clarify, we will assume that the MAC utilized is just four bits long and that the Mac of the destination device is at the beginning of the packet, as previously stated. The internal construction of the switch that we designed is depicted in the following figure.

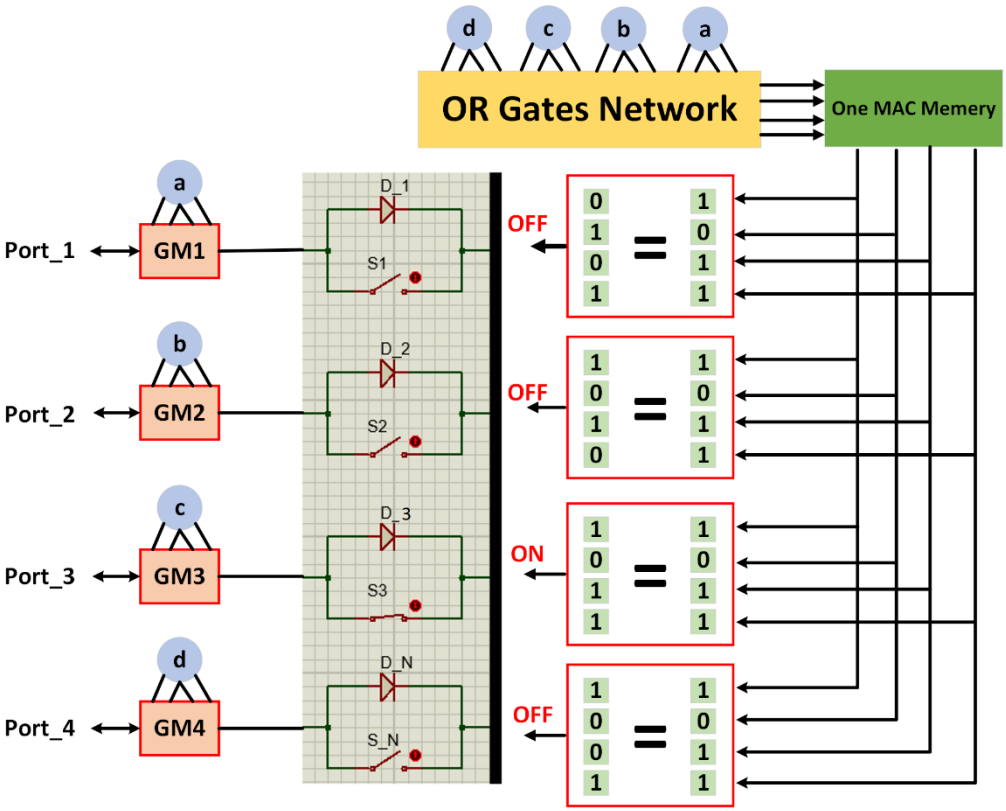


Figure 44 internal structure of enhanced switch

When the packet arrives at port number one, the trigger signal at the beginning of the packet provides the MAC acquiring section with the work signal. The MAC table of the switch shown in below figure.

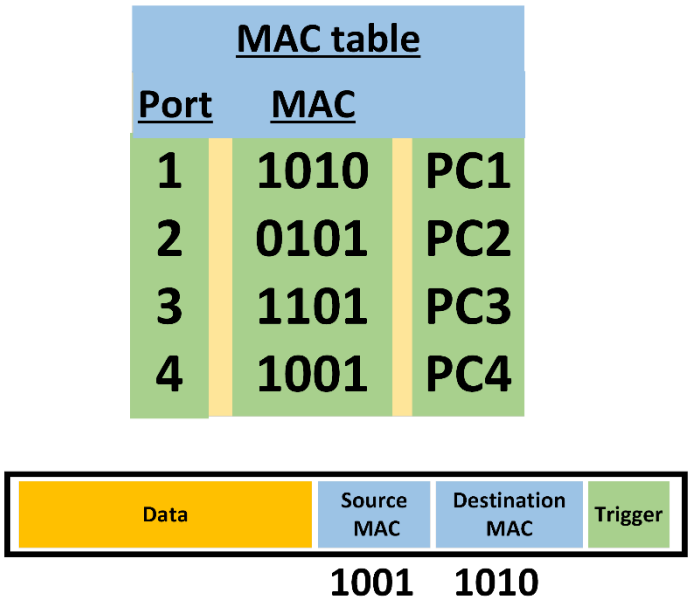


Figure 45 Binary MAC table with ideal packet

This circuit retrieves the MAC address of target device 3, which is 1101, and transfers it over the network via the OR gates network to the common MAC address. The shared MAC memory circuit communicates the MAC to all Enable Node circuits for comparison purposes. Enable node circuits ENs have the same information as the switch's MAC table. As seen in the previous figure's MAC table, EN1 = 1010, EN2 = 0101, EN3 = 1101, and EN4 = 1001 entirely according to the MAC table. The ENs circuits match the MAC address contained in the received packet, and their outputs are EN1 = EN2 = EN4 = 0. Only EN3 equals 1 since the EN3 Enable Node circuit's memory has the same MAC received (1101). The EN3 Enable Node circuit's Enable Node signal closes the Connection Node CN3's switch S3. By closing the switch S3, a closed circuit is formed consisting of port No. 1 connected to the diode D1 connected to the transmission line and then to the switch S3 connected to port No. 3. This operation took the period of the trigger signal plus the duration of 4 bits. That is, the signal has been delayed by 4 bits in addition to the trigger signal. We can see from the above example that the time of information redirection is quite near to real time. If we construct this system to function on a 48-bit MAC, routing the signal requires 48 bits plus the duration of the trigger signal. The more information included prior to the target device's MAC address, the longer the delay necessary for the system to determine the data's destination. The optimum situation for this system is to place the target device's MAC address at the beginning of the packet. While we have completed the design and testing of the system at this point, tweaks and improvements to the previous design can be made to make it more efficient and cost effective. While we have concentrated on the overall concept in this thesis, it is acceptable to mention some enhancements that the developers of this research may bring to the system in the future.

4 Chapter 4: Design discussion

Although the system has been tested theoretically and practically and has proven to be effective, it is necessary to delve deeper into the details of this system in order to discuss its capabilities, advantages, and disadvantages in order to determine its appropriate use. Additionally, we will discuss some areas that could be the subject of future research for the system's development.

4.1 System uses

This system has a very fast switching process, but it is also quite expensive, hence it is recommended for usage in network nodes that act as a gateway to a particular network with a high probability of congestion. This system is compatible with the current systems used in switch devices. That is, the switch operates normally, but when the node is under a heavy load, the device activates the hardware system temporarily until the load is lessened and then returns to its usual condition. This method reduces the device's energy consumption, as the designed hardware may require a significant amount of energy, while maintaining all the benefits associated with the processor's presence in the node, such as protection and monitoring. Due to the diversity of devices in the network and the variety of communication goals, it is impossible to decide whether to use the system in one case or in all circumstances, and thus the question of system use must be presented in the context of a debate rather than expressly. Notably, the system operates after the second attempt to communicate between two devices, that is, when the MAC address of the destination device is saved in the switch's MAC table. The first connection process occurs normally, as mentioned above, but the second connection process between the devices (where a MAC for the connected devices exists in the switch's MAC table) activates the new system. This characteristic demonstrates that this approach is effective in networks with lengthy communication processes between devices. In other words, this approach is extremely effective when the MAC has been stored in the MAC table for a long period of time. Keeping the MAC in the MAC table for an extended period of time indicates that the device is performing a continuous connection process or that there is a brief period between connection processes. We know that if a device does not establish a connection for a specified amount of time, its MAC address is erased from the switch. The converse is true; if the system is used in a network without lengthy communication processes, it will not perform as intended, as the switch must perform a broadcast to obtain the MAC address of the destination device each time, and the system will function only after the second connection process.

For example, if the devices in the network connect every 30 minutes, the MAC table deletes the devices' information after five minutes if no new connections to or from these devices occur. In this circumstance, the switch searches for the MAC address of the destination device on each connection and using the new system we devised is pointless because the system works only if the king is present in the switch's MAC table, which implies that the mac is kept in the system's memory.

4.2 Broadcast Process

The (FF-FF-FF-FF-FF-FF-FF) MAC is used as the destination device's MAC during the podcast process, but the system is unaware that this MAC means circulating the packet to all ports. To resolve this issue, it must be straightforward. This MAC address is kept in each port's circuitry. This implies that the port's memory must contain a clip dedicated to detecting the Broadcast MAC, as illustrated in the following figure.

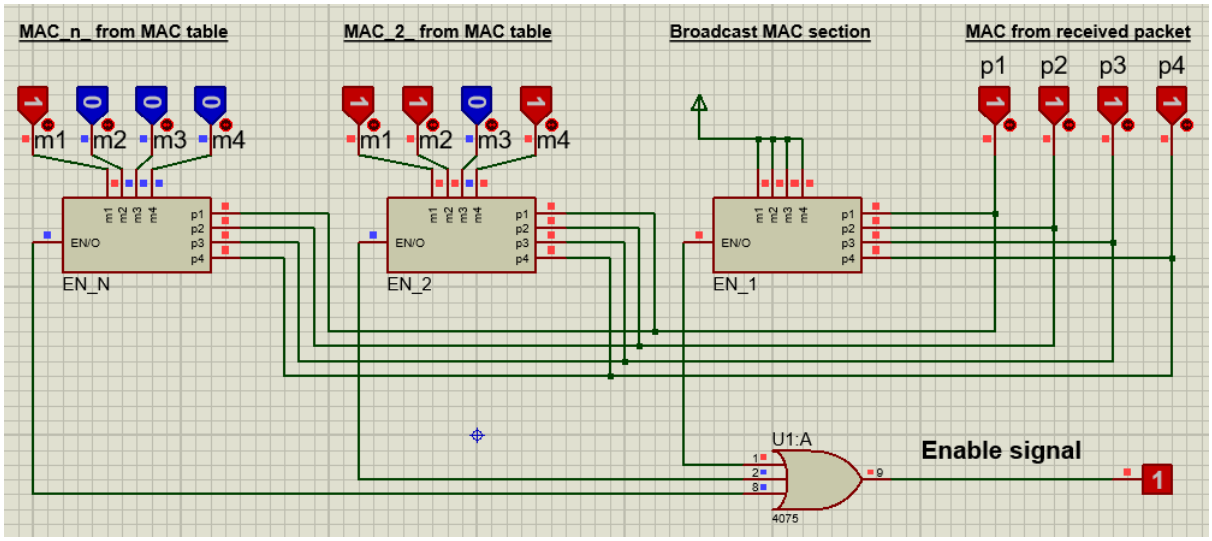


Figure 46 Enable Node with broadcast section

The preceding diagram illustrates an EN circuit capable of detecting the broadcast for a 4-bit MAC, but the same approach may be used to an actual 48-bit MAC. The XNOR gate's inputs were grouped together to conserve memory cells. It was feasible to use the same enable node circuit EN to store several MACs, with the MAC broadcast saved in one of the clips, resulting in the circuit functioning normally and without issues. The design has been updated to eliminate four memory cells in favor of a single voltage source. When the MAC is 48 bits long, the same thing happens since 48 memory cells are saved and only one voltage source is used.

4.3 Use Multi data buss

In its current configuration, the system operates on a single transmission line and relies on the receipt of a single piece of information from a single port in each. Of course, in practice, the switch will receive a huge quantity of data on practically all ports, necessitating the addition of a synchronization component to the design. Synchronization guarantees that one port operates as an input while the other operates as an output. The remainder of the ports must be placed on hold pending completion of the communication procedure between the two preceding ports.
The capability of utilizing multiple transmission lines will also be shown in a future work, as it involves considerable effort and design. While this thesis may not accommodate this, we will make some suggestions to the developers of this system on the use of numerous lines for data transfer between switch ports. The total number of transmission lines that can be added to the system is limited to half of the total number of ports. If the switch has n ports, the maximum number of transmission lines that can be employed is n/2, as each pair of ports communicates over a single transmission line. It may be necessary to add more than one transmission line to

the system's addition and modification in order to accomplish the intended purpose of adding numerous transmission lines. The purpose of using multiple transmission lines is to increase the device's performance in terms of speed, as each transmission line connects two distinct ports, providing us with many connections simultaneously. It's worth noting that many transmission lines are beneficial in some situations but unsuccessful in others. If all the ports in the switch wish to interact with a single port, for example, employing multiple transmission lines is not possible, as only one line will operate while the remaining lines will remain unaffected due to the fact that all the data wishes to flow to a single port.
However, if the requests on the switch ports are directed to numerous ports, the routing speed is doubled. This is because several transmission lines work in tandem, as each transmission line connects two distinct ports. Three transmission lines with synchronization and occupant line selection circuits are illustrated in the accompanying schematic.

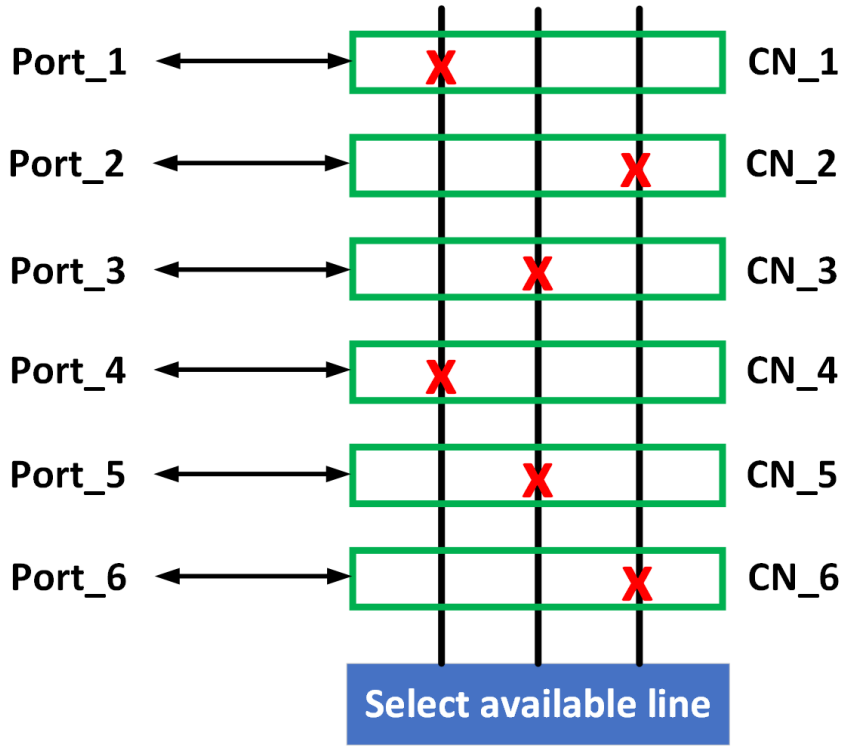


Figure 47 Data Buss with 3 lines

We observe that three communication processes are currently taking place without interruption through the switch. Although the preceding situation is ideal, it is conceivable. All system resources are being utilized in this case, including all transmission lines, because all two separate ports wish to communicate with one another. In this situation, the switch operates at a speed equal to three times that of a single transmission line. Due to the rarity of this case, it is advisable to employ more than one transmission line, say n/2, to save expenses. For instance, the usage of four transmission lines for a switch with sixteen ports, rather than eight, because the case of its arrival from communications to eight distinct regions is nearly impossible in practice. Modifications must be made to the connection node circuits to enable them to be connected to more than one transmission line in some fashion. Additionally, the shared MAC memory must be adjusted to match the MAC for the present connection and prepare to receive

another MAC from another port, rather than waiting for the communication process to complete, as is the case with transmission line One. Additionally, a circuit must be built to track free lines in order for them to be used in subsequent communication activities.

The proposed development procedures are illustrative only. Naturally, developers have complete freedom to apply different technologies in proportion to the system. The goal of this thesis was to decrease the response time of control signals traveling across a computer network, as these signals are extremely sensitive to delay, and any delay might result in major flaws in a particular system. As a result, methods must be discovered to ensure that these signals arrive at their destination with the least amount of delay feasible. We drastically decreased processing times throughout the intended system. In this circumstance, network operations are often expedited, which enhances the speed of control signal transfer as well. The proposed solutions can be utilized just for control signals and not for other signals, but this is entirely up to the network engineers, who have the option of using only system control signals or all other signals traveling through the network.

Conclusion

We have reviewed the computer network's mechanism in this thesis, with a focus on the most critical devices and protocols that comprise the network. Additionally, we discussed the subject of network response delay in detail, including the ramifications that may occur in certain applications as a result of the network response delay problem. This thesis proposed two solutions to this problem; one was software-based, while the other was hardware-based. We have gone into detail about the proposed system and discussed the critical features that must be included in order for it to be suited for the widest possible range of applications. The software solution improved the state of fairness of data transmitted over the network, while the hardware solution significantly increased the data processing process in the switch. It has been demonstrated how to configure the system to focus exclusively on control signals and ignore all other signals. Additionally, this thesis discusses some of the current viable solutions to the problem of delay, outlining the most significant advantages and downsides of each strategy. We have achieved in this thesis the creation of specialized hardware for switch devices; this hardware is highly efficient at evaluating packets and routing them to the appropriate port without the usage of a processor. We have achieved a switching speed that is extremely near to real time. At the conclusion of this research, we addressed the system we built and the relevant applications for its use, while also leaving some suggestions for future development research.

Table of Figures

References

[1] A. S. Tanenbaum and D. J. Wetherall, “Computer-Networks-5th-Edition,” 2011.

[2] J. F. Kurose and K. W. Ross, *Computer networking : a top-down approach*. Pearson, 2013.

[3] R. Jiang, “A review of Network Topology,” 2015.

[4] Y. Nureni, “DATA COMMUNICATION & NETWORKING E-Learning View project NETWORK SECURITY View project,” 2015. [Online]. Available: https://www.researchgate.net/publication/288180515

[5] S. Babani, A. A. Bature, M. I. Faruk, and N. K. Dankadai, “Comparative Study Between Fiber Optic And Copper In Communication Link,” 2014. [Online]. Available: www.ijtra.com

[6] Stephen McQuerry, “CCNA Self-Study: Network Media (The Physical Layer),” *CISCO*, Apr. 09, 2004. https://www.ciscopress.com/articles/article.asp?p=169686 (accessed Feb. 19, 2022).

[7] “Routers and Switches and Hubs, Oh My! An Introduction to Network Devices and Their Functions,” *CISCO*, Oct. 12, 2015. https://www.ciscopress.com/articles/article.asp?p=2434666 (accessed Feb. 19, 2022).

[8] J. F. Kurose and K. W. Ross, *Computer networking : a top-down approach*. 2017, 2017.

[9] P. Dordal, “An Introduction to Computer Networks Release 2.0.6 Peter L Dordal,” 2022.

[10] M. Ali *et al.*, “Performance Comparison between TCP and UDP Protocols in Different Simulation Scenarios,” 2018. [Online]. Available: www.sciencepubco.com/index.php/IJET

[11] C. ’Buan, *Computer Networks / Computernetze*. 2019.

[12] P. ’Larry L and D. ’Bruce S, *Computer Networks / A systems approach*, 6th ed. Birtcher, Katey, 2021.

[13] H. Attar, M. R. Khosravi, S. I. Shmatkov, K. N. Georgievan, and M. Alhihi, “Review and performance evaluation of FIFO, PQ, CQ, FQ, and WFQ algorithms in multimedia wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 16, no. 6. SAGE Publications Ltd, Jun. 01, 2020. doi: 10.1177/1550147720913233.

[14] S. Szabolcs and A. Béla, “A Review of Congestion Management Algorithms on Cisco Routers.”

[15] I. Zakariyya, M. Nordin, and A. Rahman, “Bandwidth Guarantee using Class Based Weighted Fair Queue (CBWFQ) Scheduling Algorithm.”

[16] T. A. Assegie and H. D. Bizuneh, “Improving network performance with an integrated priority queue and weighted fair queue scheduling,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 1, pp. 241–247, 2020, doi: 10.11591/ijeecs.v19.i1.pp241-247.

[17] M. Jutila and T. Frantti, “Cognitive fuzzy flow control for wireless routers,” 2018.

[18] F. Ye, T.-A. Tang, Institute of Electrical and Electronics Engineers. Beijing Section, C. Fu dan da xue (Shanghai, and Institute of Electrical and Electronics Engineers, *Proceedings, 2019 IEEE 13th International Conference on ASIC (ASICON 2019) : Oct. 29 - Nov. 1, 2019, Chongqing, China*.

[19] T. Haslak, “Weighted Fair Queuing as a Scheduling Algorithm for Deferrable Loads in Smart Grids,” in *Trends in Mathematics*, Springer, 2020, pp. 123–141. doi: 10.1007/978-3-030-32157-4_8.

[20] D. Singh, B. Ng, Y. C. Lai, Y. D. Lin, and W. K. G. Seah, “Modelling Software-Defined Networking: Software and hardware switches,” *Journal of Network and Computer Applications*, vol. 122, pp. 24–36, Nov. 2018, doi: 10.1016/j.jnca.2018.08.005.

[21] S. Shukla, M. F. Hassan, L. T. Jung, and A. Awang, “Architecture for latency reduction in healthcare internet-of-things using reinforcement learning and fuzzy based fog computing,” in *Advances in Intelligent Systems and Computing*, 2019, vol. 843, pp. 372–383. doi: 10.1007/978-3-319-99007-1_36.

[22] W. Wang, G. Wu, Z. Guo, L. Qian, L. Ding, and F. Yang, “Data scheduling and resource optimization for fog computing architecture in industrial IoT,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11319 LNCS, pp. 141–149. doi: 10.1007/978-3-030-05366-6_11.

[23] A. Manzoor, M. Hussain, and S. Mehrban, “Performance Analysis and Route Optimization: Redistribution between EIGRP, OSPF & BGP Routing Protocols,” *Computer Standards and Interfaces*, vol. 68, Feb. 2020, doi: 10.1016/j.csi.2019.103391.

[24] “Cisco IP Telephony Flash Cards: Weighted Random Early Detection (WRED),” *CISCO*, 2004. https://www.ciscopress.com/articles/article.asp?p=352991&seqNum=8 (accessed Mar. 21, 2022).

[25] A. Kumar. Maini, *Digital electronics : principles, devices and applications*. John Wiley & Sons, 2007.