

Faculty of Engineering and Technology

Electrical and Computer Engineering Department

Computer Networks Laboratory ENCS4130

**Access Lists EXP. No. 6**

Prepared by : Ansam Osama Hababeh

ID Number : 1201127

Instructor : Dr.Ahmad Shawahneh

Teaching Assistant : Eng.Ahed Mafarjeh

Section : 7

Birzit

20December – 2023

# **Abstract**

The purpose of the experiment is to acquaint participants with access control lists (ACLs), their kinds, and their variants. In addition to setting up and executing ACLs on a simple architecture made with Cisco Packet Tracer software. And introducing the methods of applying it like Standard ACL and Extended ACL. It is necessary to configure the different ACL types for different circumstances and goals.

The requirement for this experiment is preparing the topology, filling the IPs (Internet Protocol) and applying the OSPF (Open Shortest Path First) routing protocol.

Contents

[**Abstract** 2](#_Toc153914624)

[**1.** **Introduction** 7](#_Toc153914625)

[**1.1** **Overview** 7](#_Toc153914626)

[**1.2** **Named and numbered ACL** 7](#_Toc153914627)

[**1.3** **Wildcard Masks** 7](#_Toc153914628)

[**1.4** **Type Access List** 8](#_Toc153914629)

[**1.4.1** **Standard Access Control Lists** 8](#_Toc153914630)

[**1.4.2** **Extended Access Control Lists** 8](#_Toc153914631)

[**1.5** **Configuring ACL** 9](#_Toc153914632)

[**1.5.1** **Standard ACL Configuration** 9](#_Toc153914633)

[**1.5.2** **Extended ACL Configuration** 9](#_Toc153914634)

[**1.5.3** **Applying ACL to interface** 9](#_Toc153914635)

[**1.5.4** **The Three Rules of Configuring ACLs** 9](#_Toc153914636)

[**2.** **Procedure and Discussion** 10](#_Toc153914637)

[**2.1** **Building The Topology** 10](#_Toc153914638)

[**2.2** **Configuring OSPF Routing** 11](#_Toc153914639)

[**2.3** **Configuring Standard Access List** 12](#_Toc153914640)

[**A.** **Task 1:** 12](#_Toc153914641)

[**B.** **Task 2:** 13](#_Toc153914642)

[**C.** **Task 3:** 15](#_Toc153914643)

[**D.** **Task 4:** 16](#_Toc153914644)

[**2.4** **Configuring Extended Access List** 19](#_Toc153914645)

[**A.** **Task 1:** 19](#_Toc153914646)

[**B.** **Task 2:** 21](#_Toc153914647)

[**C.** **Task 3:** 22](#_Toc153914648)

[**D.** **Task 4:** 24](#_Toc153914649)

[**E. Task 5:** 25](#_Toc153914650)

[**Conclusion** 27](#_Toc153914651)

[**References** 28](#_Toc153914652)

# **Table of Figures**

[Figure 1:toplogy 12](#_Toc153914653)

[Figure 2:ospf router 0 13](#_Toc153914654)

[Figure 3:ospf router1 13](#_Toc153914655)

[Figure 4:Standard Task A Command 14](#_Toc153914656)

[Figure 5:Failed respond 14](#_Toc153914657)

[Figure 6:Standard Task A Result 15](#_Toc153914658)

[Figure 7:Standard Task B Command 15](#_Toc153914659)

[Figure 8:show access 16](#_Toc153914660)

[Figure 9:Standard Task B Result 16](#_Toc153914661)

[Figure 10:another result 16](#_Toc153914662)

[Figure 11:Standard Task C Command 17](#_Toc153914663)

[Figure 12:acces list 17](#_Toc153914664)

[Figure 13:Result for c 18](#_Toc153914665)

[Figure 14:Standard Task D Command 19](#_Toc153914666)

[Figure 15:Run 19](#_Toc153914667)

[Figure 16:result of D 20](#_Toc153914668)

[Figure 17: Extended Task A Command 21](#_Toc153914669)

[Figure 18:run fig 21](#_Toc153914670)

[Figure 19:Result of A 22](#_Toc153914671)

[Figure 20:Another Result 22](#_Toc153914672)

[Figure 21:Extended Task B Command 23](#_Toc153914673)

[Figure 22:Result of B 24](#_Toc153914674)

[Figure 23:adding server 24](#_Toc153914675)

[Figure 24:Extended Task C Command 25](#_Toc153914676)

[Figure 25:Result of c 25](#_Toc153914677)

[Figure 26:Extended Task D Command 26](#_Toc153914678)

[Figure 27:Result 26](#_Toc153914679)

[Figure 28:Extended Task E Command Part B 27](#_Toc153914680)

[Figure 29:Extended Task E Result 28](#_Toc153914681)

# **Table of Tables**

[Table 1:Networks IPS 12](#_Toc153914682)

# **Acronyms and Abbreviations**

OSPF Open Shortest Path First

ACL Access Control List

# **Introduction**

## **Overview**

Control Allowing Routing for devices or allowing them to do so, All this through Access list Or access control list (ACL), Where it filters network traffic by controlling if it is routing or redirecting, as it provides protection for networks, to more accurately match a particular network activity, extended access lists are employed. While providing a far higher level of control than standard access lists, extended access lists are more difficult to configure and require more processing time., where it works in two ways, the first is Configuring Standard Access List, the second is Configuring Extended Access List. While Extended ACL can match based on source and destination IP addresses as well as port number and protocol number, Standard Access List can only match based on source IP address. [1]

## **Named and numbered ACL**

There are two ways to identify access lists: by number or by name. The outdated method of identifying ACLs with numbers is difficult to remember and update. Alternatively, the more recent approach to identify access lists is using named ACLs. They perform better with bigger networks and are simpler to maintain and control.[1]

## **Wildcard Masks**

Access lists are created in many devices using wildcard masks, sometimes known as inverse masks. These masks are challenging to understand since they are subnet masks in binary reverse. In other words, 0.0.0.255 would be the wildcard mask used to match a range specified by the subnet mask 255.255.255.0. Here is a straightforward rule that will address the majority of subnet/wildcard mask issues you encounter: All 0s must be changed to 255s, and vice versa.

## **Type Access List**

There are several types of ACLs, that are used for different purposes. Two main types of these are standard and extended ACLs.[2]

### **Standard Access Control Lists**

Standard access lists are created using only the source IP address, allowing or denying access to all protocols. They use numbers 1-99 to identify the originating IP address. These lists are typically used near the destination and prohibit access to the entire network or sub-network. Rules cannot be deleted, and the entire list is destroyed if removed. Conventional access lists are used, copying the primary topology at every stage.

### **Extended Access Control Lists**

The Extended Access List (ACL) is a popular tool that distinguishes between different types of IP traffic, requiring source and destination IP addresses, port numbers, and allowing or blocking specific services. It operates between 100-199 and can be used near the source. However, rules cannot be deleted and the entire access list will be destroyed if removed.

## **Configuring ACL**

### **Standard ACL Configuration**

A standard ACL can be configured on a cisco router using the following command:

Router(config)# access-list <ACCESS-LIST-NUMBER> <permit|deny> <host | source sourceWildCardMask | any>

Where the access list number should be in the range 1-99 or 1300-1999. Standard ACLs are usually applied to the interface nearest to the destination.

### **Extended ACL Configuration**

An extended ACL can be configured on a cisco router using the following command:

Router(config)# access-list <ACCESS-LIST-NUMBER> <permit | deny>

<TRANSPOT-LAYER-PROTOCOL> <host |source sourcewildcardmask | any>

<host|destination destinationWildCardMask|any> eq <PORT-NUMBER>

Where the access list number should be in the range 100-199 or 2000-2699. Standard ACLs are usually applied to the interface nearest to the source. [2]

### **Applying ACL to interface**

After building an Access-Lists, it should be applied to an interface, and can be inbound or outbound. Inbound ACLs process the incoming packets, then drops or routes them to the outbound interface. On the other hand, for the outbound ACLs, the packets are routed and then processed at the outbound interface.[2]

### **The Three Rules of Configuring ACLs**

When configuring ACLs, keep in mind these three fundamental principles: one ACL per protocol, one ACL per direction, and one ACL per interface. These guidelines control the flow of traffic on a network, so following them is important.[2]

# **Procedure and Discussion**

## **Building The Topology**

Figure 2.1 topology was constructed with Cisco Packet Tracer. Two routers, three switches, and six PCs make up the topology, which consists of four networks in the same region (area0). Table 1 shows sub-netting the IPs of networks and PCs and identify the areas.

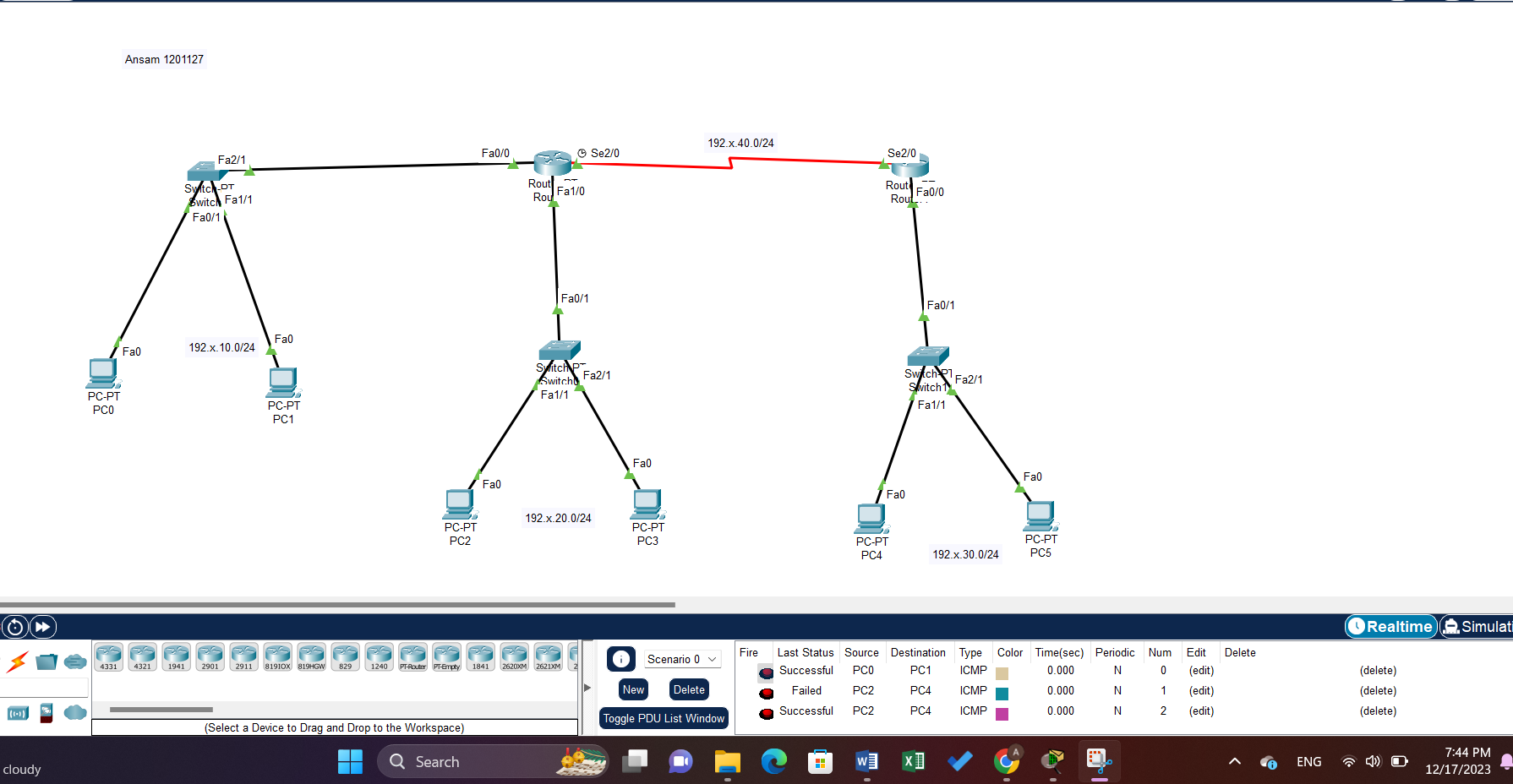
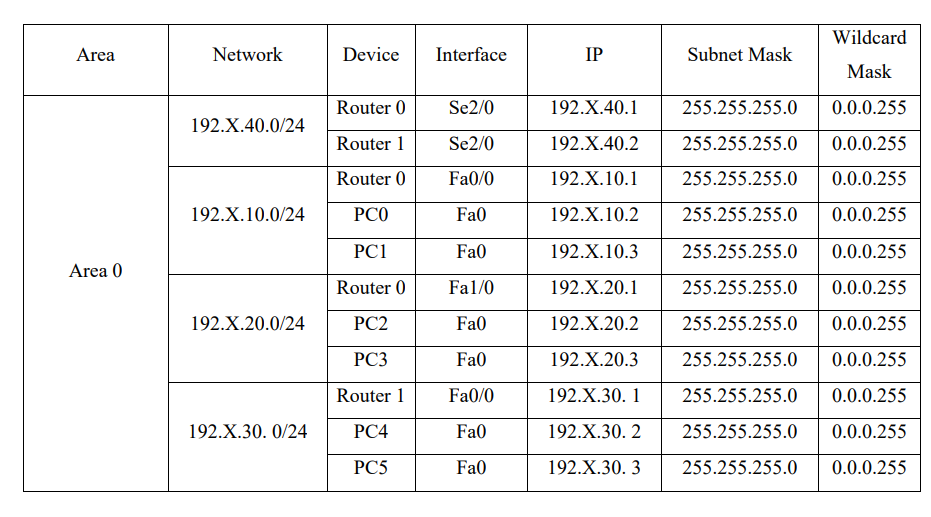


Figure 1:toplogy

Table 1:Networks IPS



## **Configuring OSPF Routing**

Each PC has its IP address, subnet mask, and gateway set according to Table 1. Additionally, each network interface that we utilized had a router activated for it and a unique IP address assigned to it. Then, by entering commands into the routers that specify the routing process ID and choose which networks contain the specified router by entering the network ID, wildcard mask, and area number, OSPF has been configured.

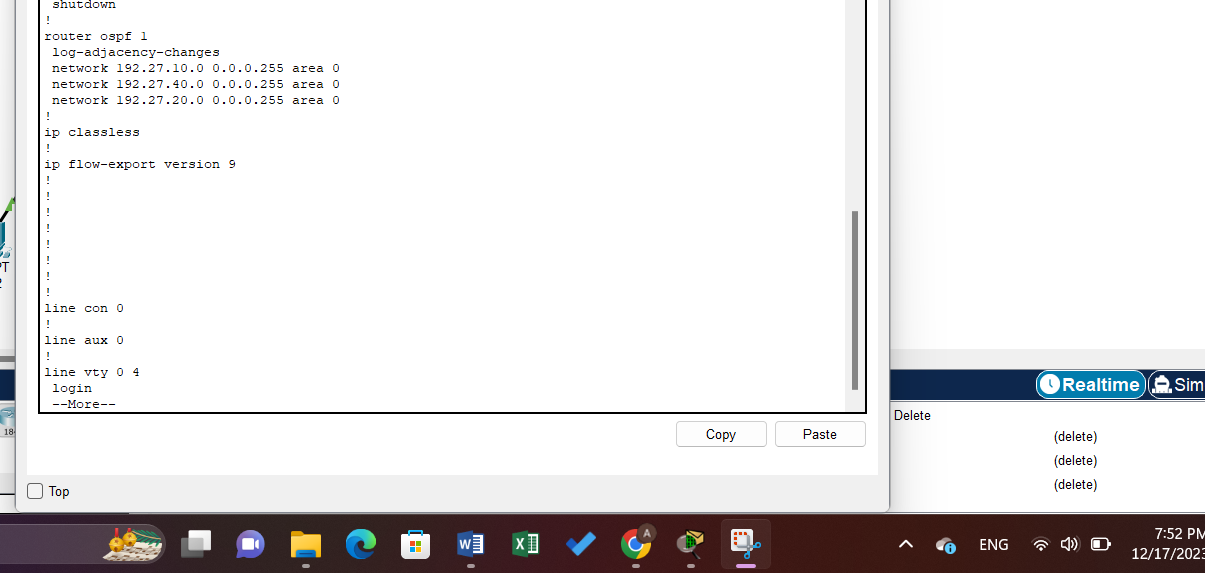


Figure 2:ospf router 0

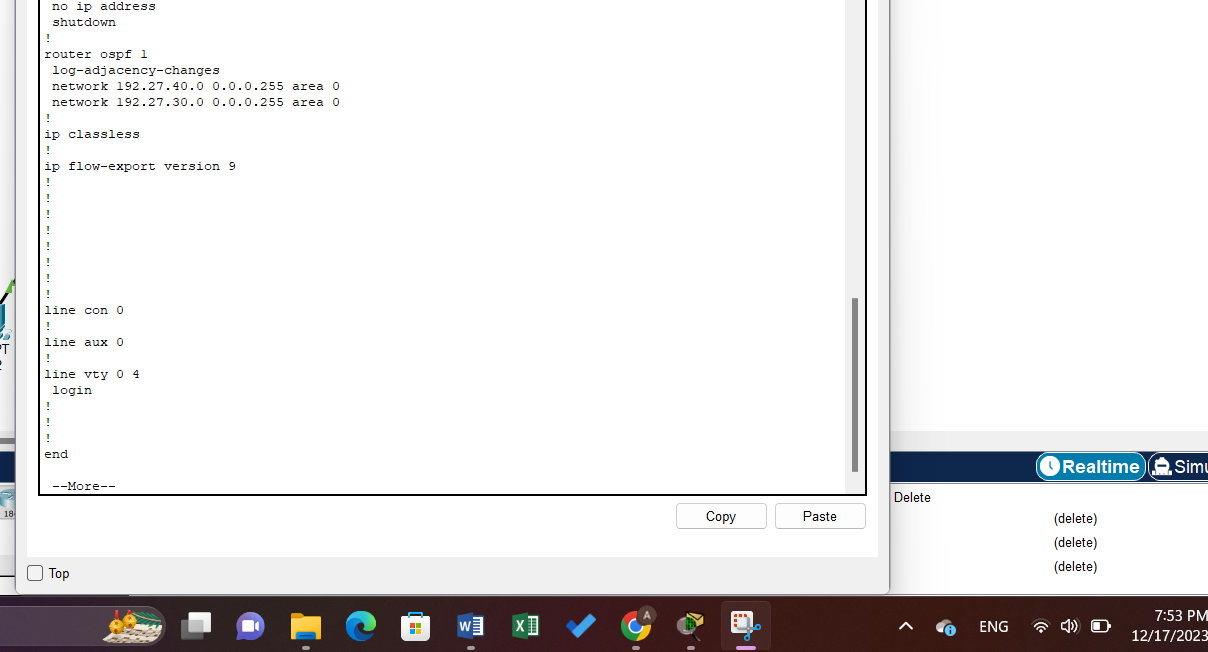


Figure 3:ospf router1

## **Configuring Standard Access List**

### **Task 1:**

**Prevent 192.27.10.2 (pc0) to access network 192.27.20.0/24.**

An access list to prevent PC0 from accessing network 192.27.20.0/24 was created on Router0 using the following commands:

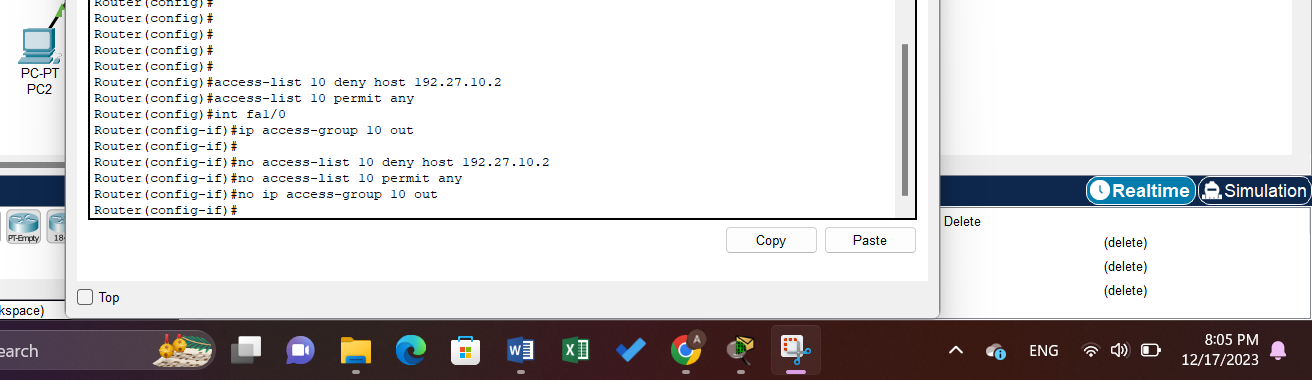


Figure 4:Standard Task A Command

When preventing the PC 0 from transmitting inside the router, any device outside the Swig and inside the router does not respond to it.

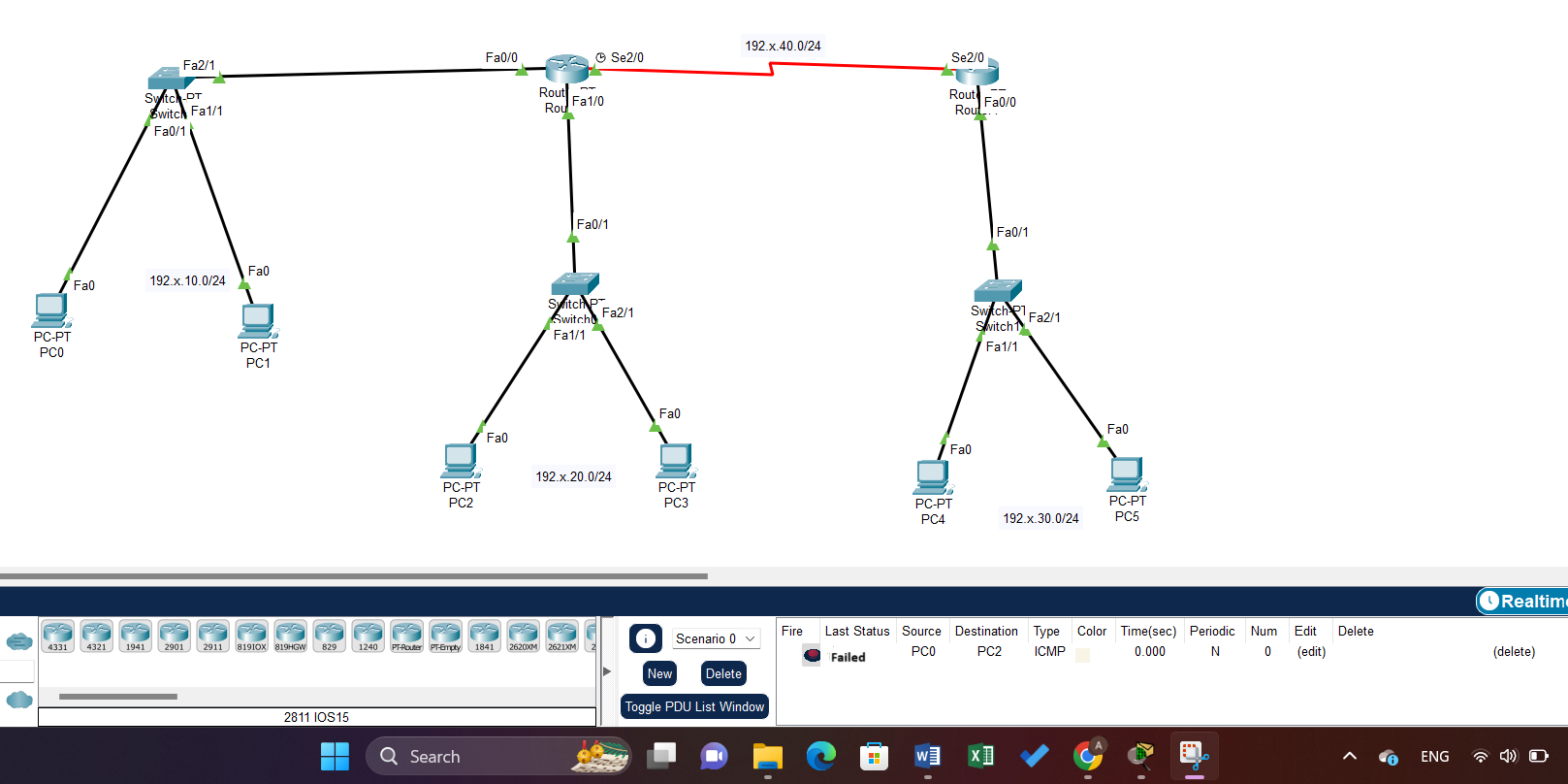


Figure 5:Failed respond

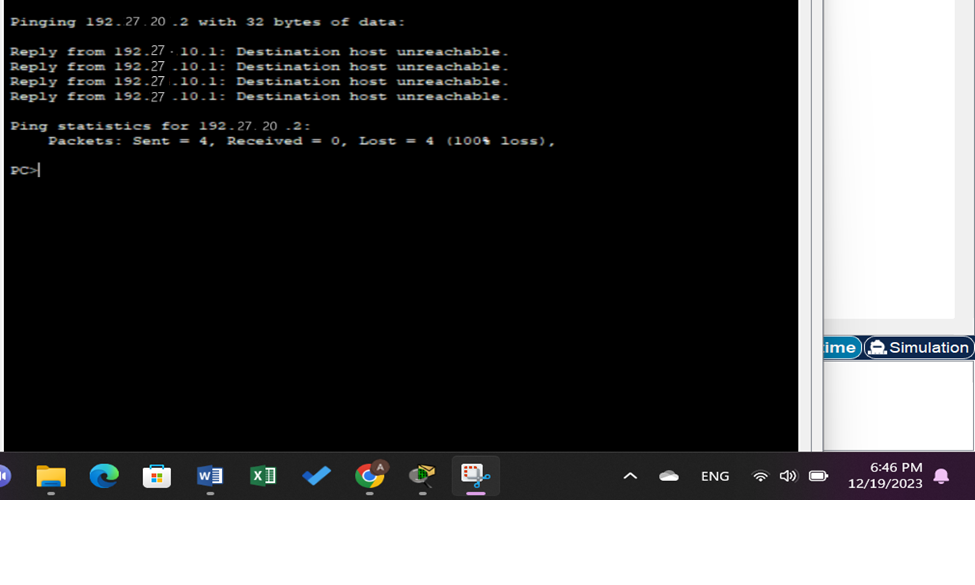


Figure 6:Standard Task A Result

### **Task 2:**

**Allow just PC0 to access network 192.27.20.0/24 using the Standard ACLs and deny any other traffic.**

An access list to do so was created on Router0, using the following command:

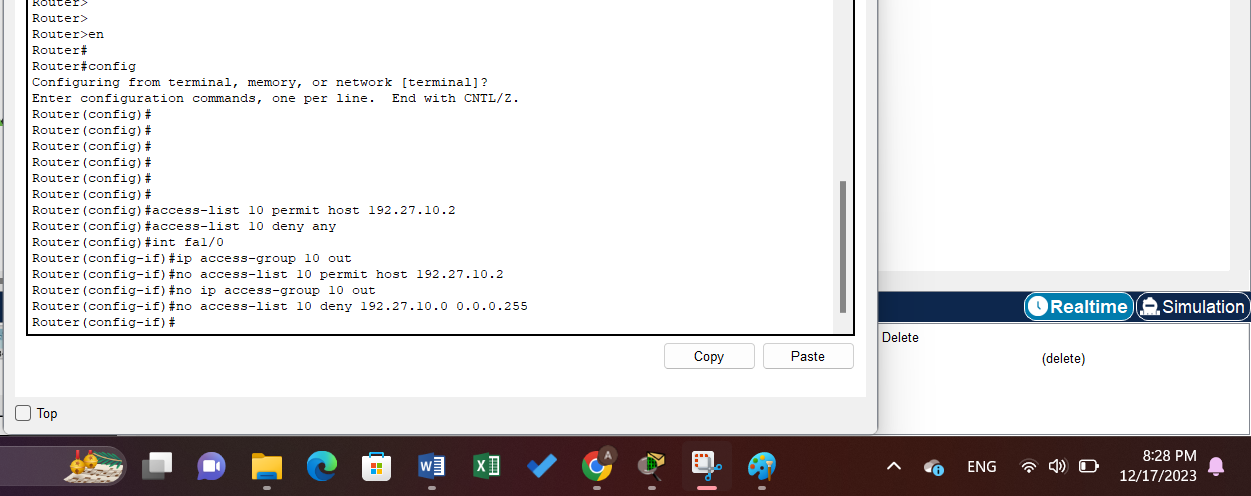


Figure 7:Standard Task B Command

Only PC0 can ping to network 192.27.20.0

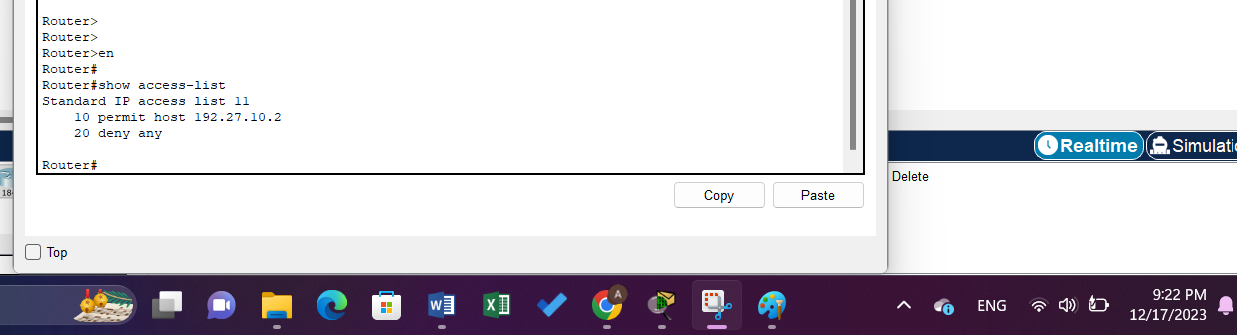


Figure 8:show access



Figure 9:Standard Task B Result

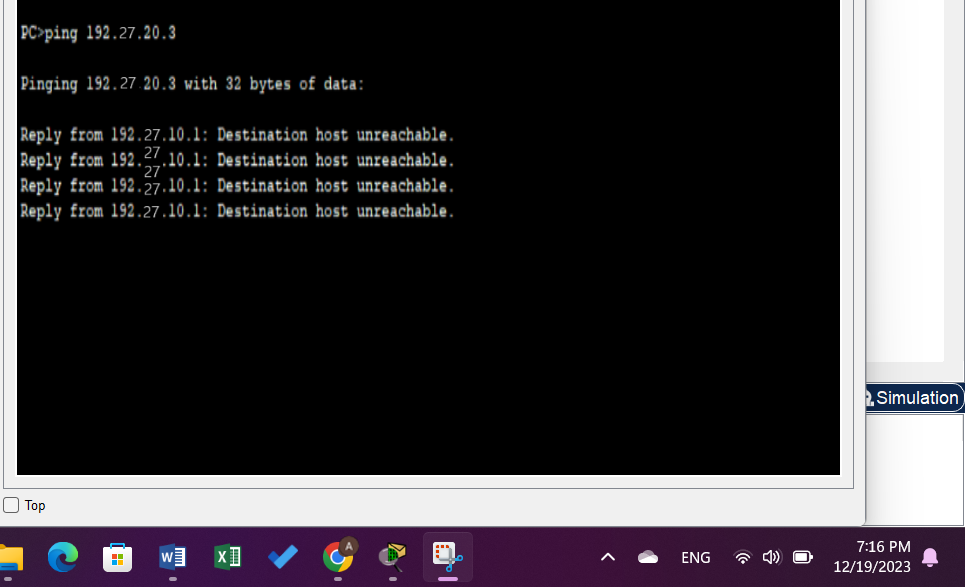


Figure 10:another result

### **Task 3:**

**Prevent network 192.27.10.0 from accessing network 192.27.20.0 only (use the wild-card, not ‘any’ option).**

An access list to do so was created on Router0, using the following commands:

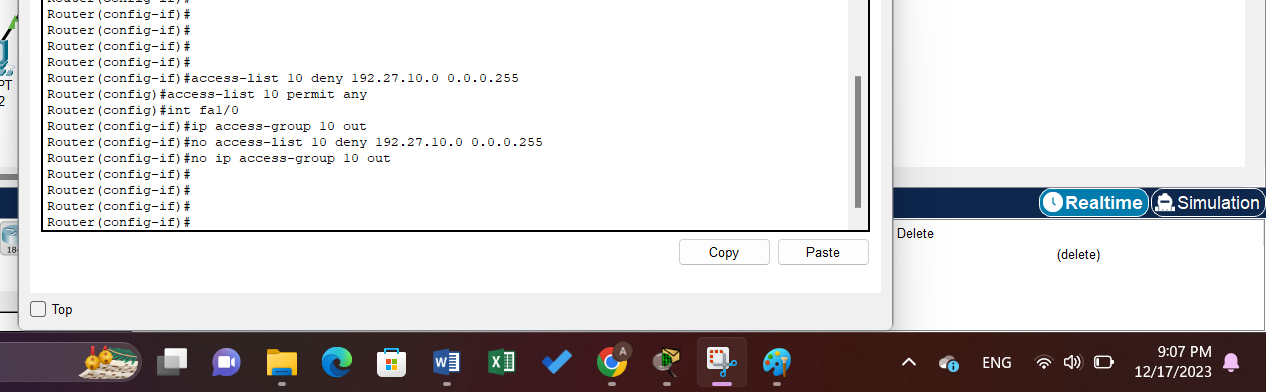


Figure 11:Standard Task C Command

PC0 and PC1 can’t reach PC3 and PC2.

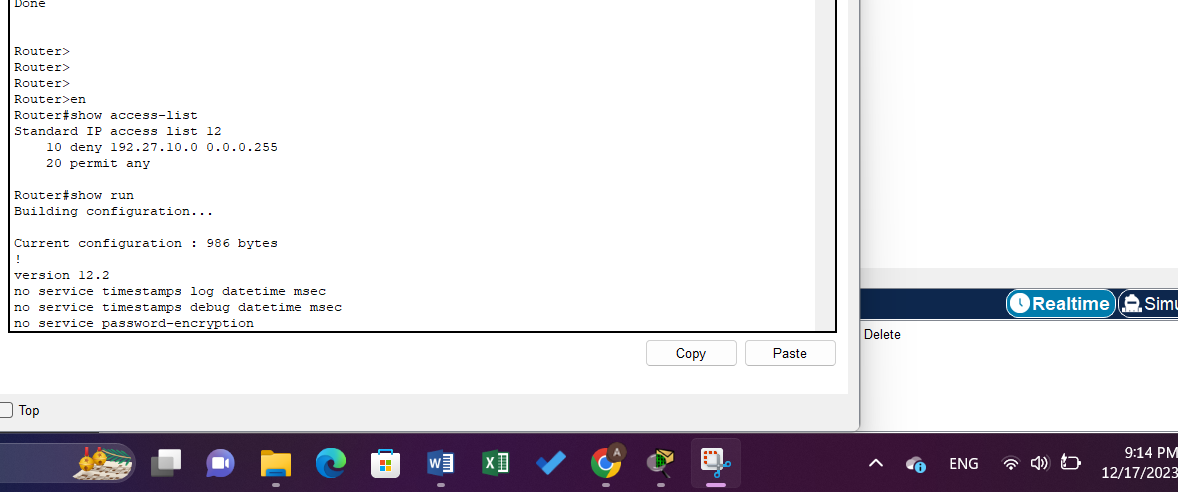


Figure 12:acces list

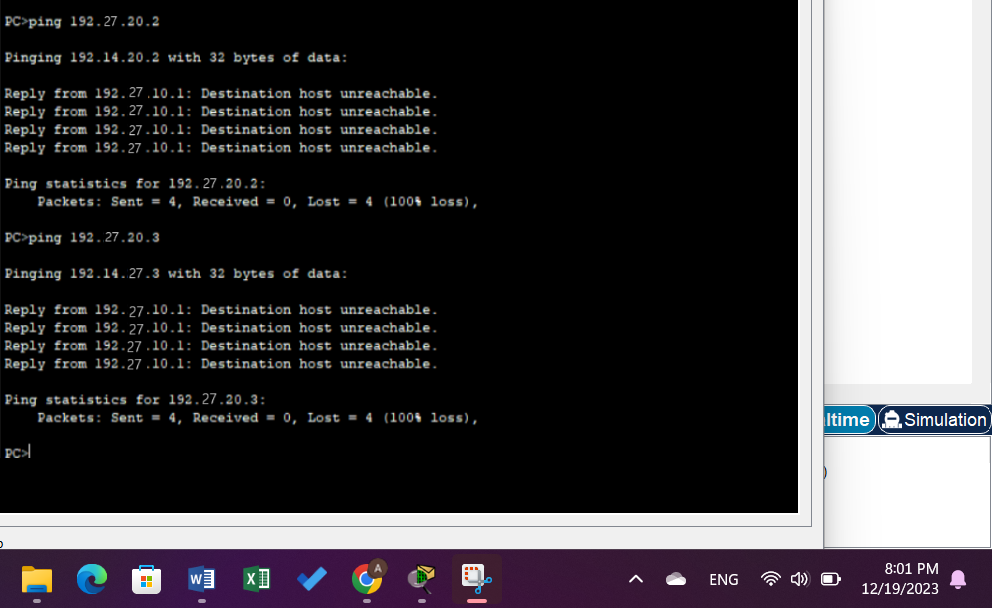


Figure 13:Result for c

### **Task 4:**

**Prevent PC0 from accessing Network 192.27.30.0 all other traffic is allowed.**

An access list to prevent PC0 from accessing network 192.27.30.0/24 was created on Router1, using the following commands:

In this PC0 can’t reach network 192.27.30.0.

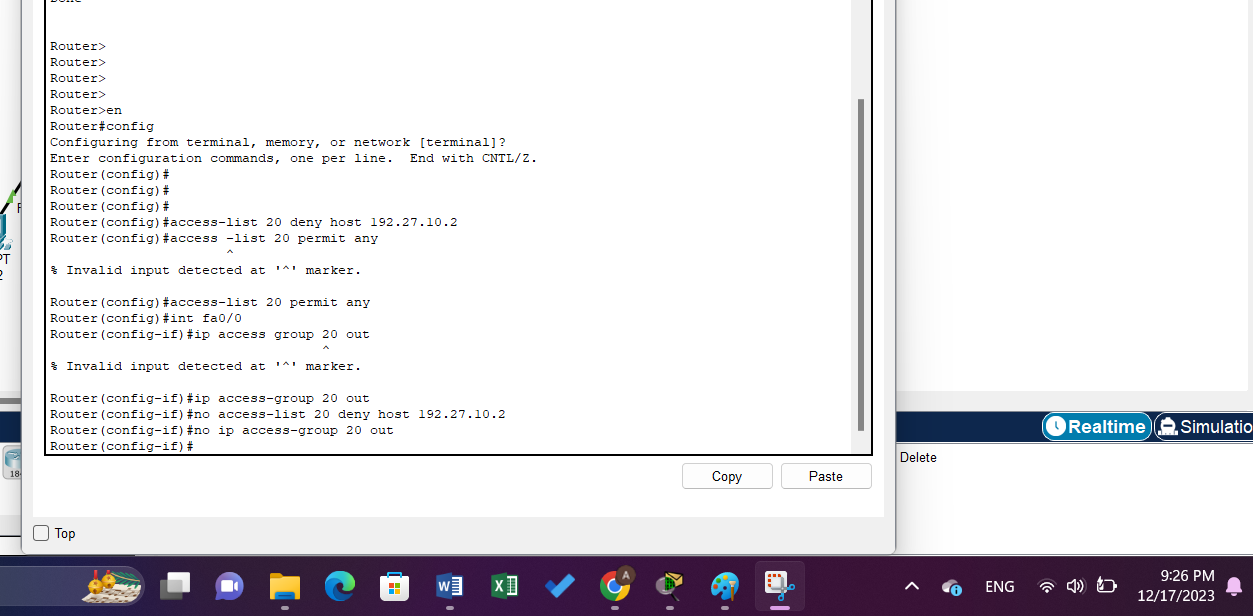


Figure 14:Standard Task D Command

In this PC0 can’t reach network 192.27.30.0.

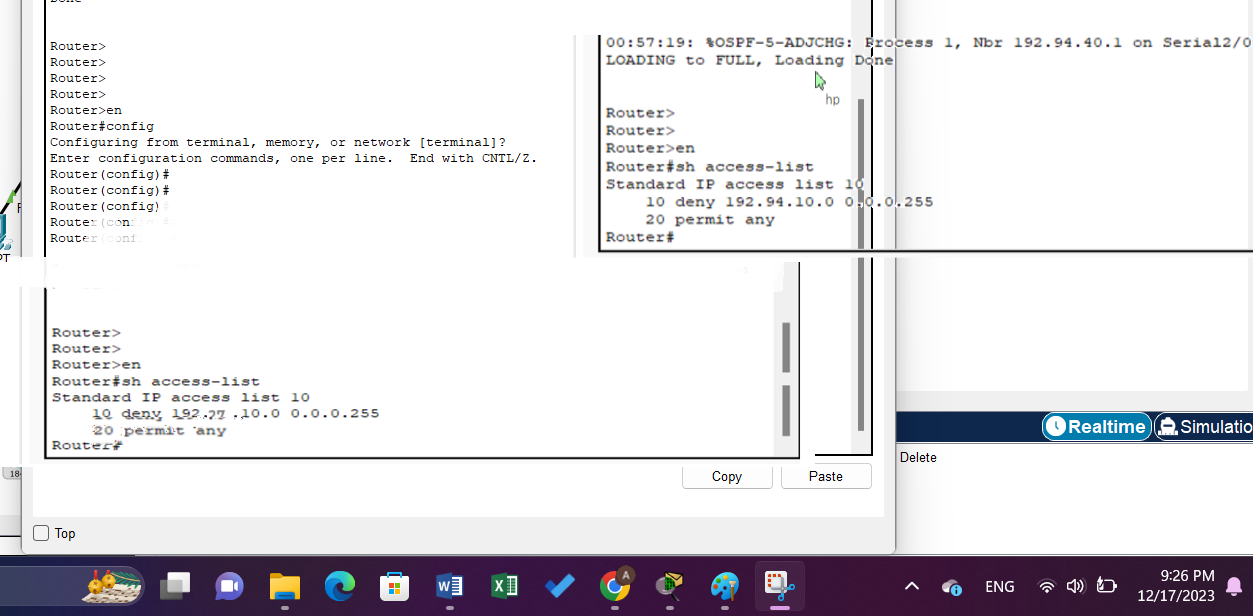


Figure 15:Run

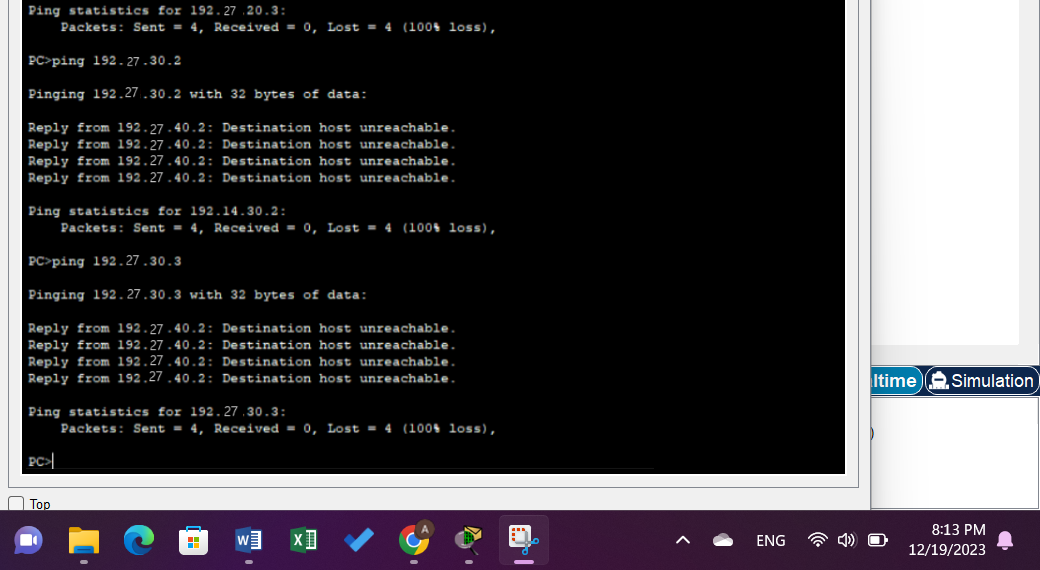


Figure 16:result of D

## **Configuring Extended Access List**

In this section, extended access list will be used , and make sure in each step to copy the main topology. Extended ACL takes IDs of 100 to 199.

### **Task 1:**

**Prevent PC0 (192.27.10.2) from accessing PC2 (192.27.20.2). (All other traffic is allowed).**

An access list to do so was created on Router0, using the following commands:

And in this PC0 can’t reach PC2 but can reach anything else.

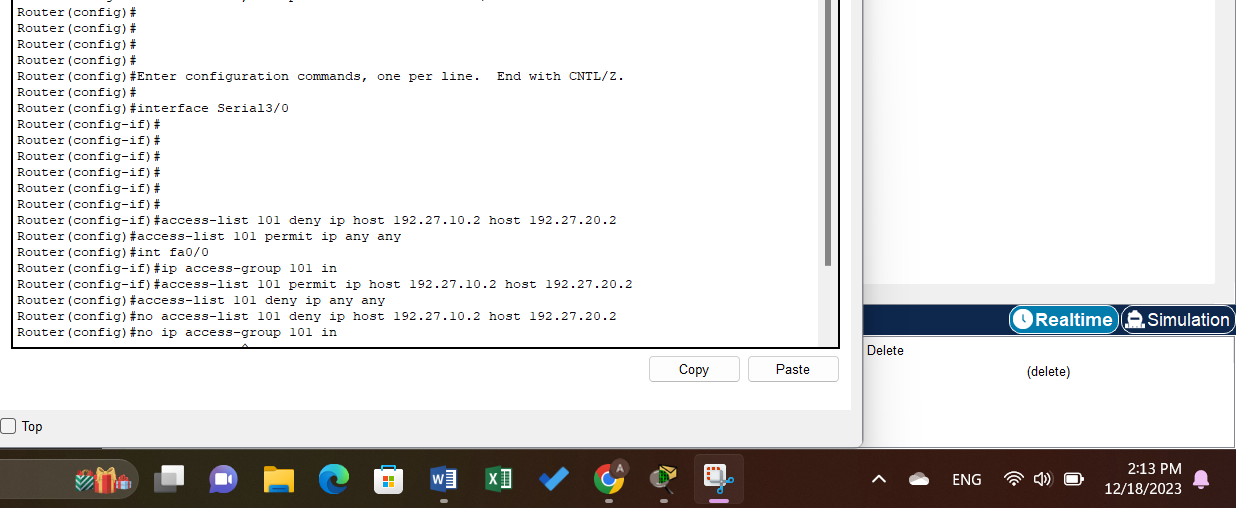


Figure 17: Extended Task A Command

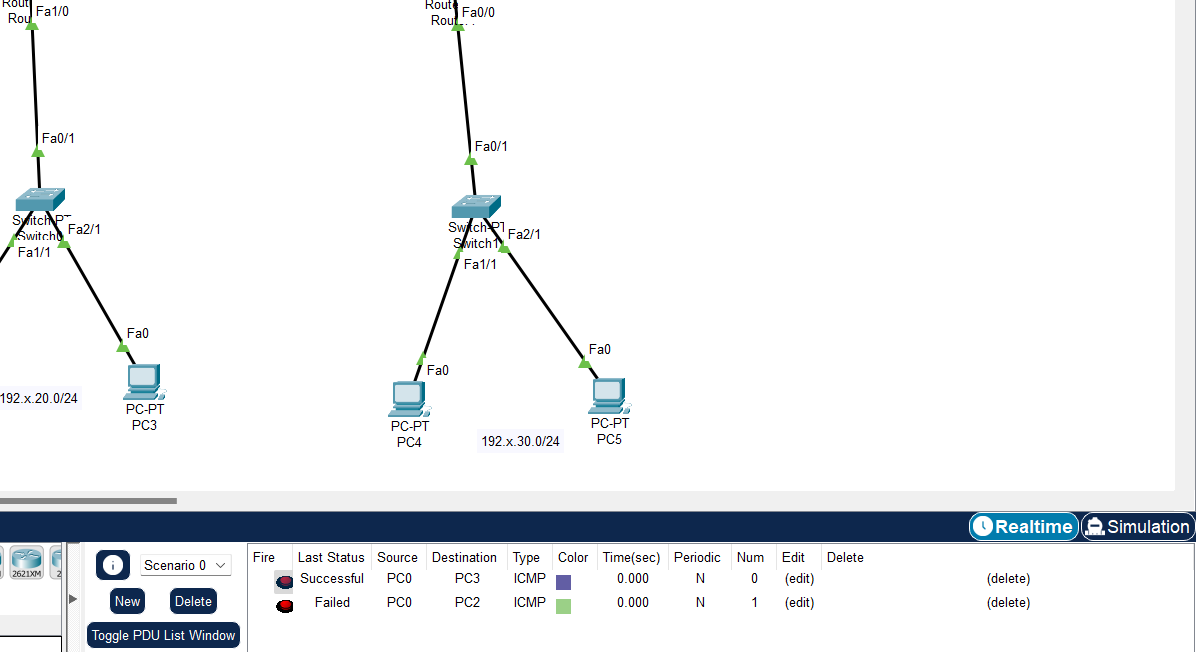


Figure 18:run fig

From pc0 and pc3 is successful but from pc0 and pc2 is failed.

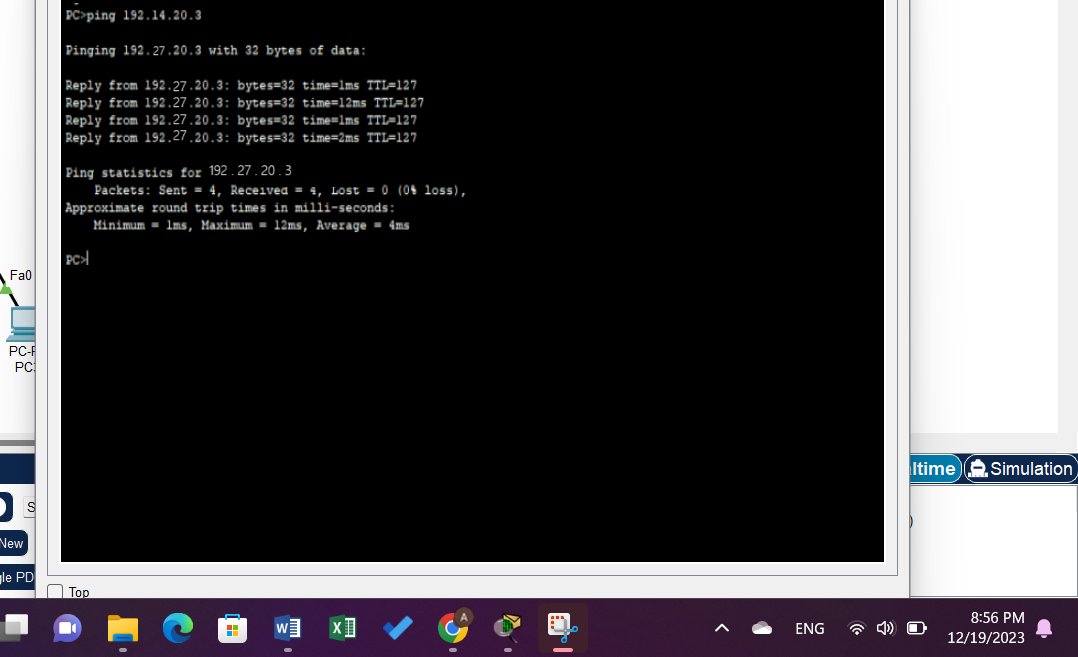


Figure 19:Result of A

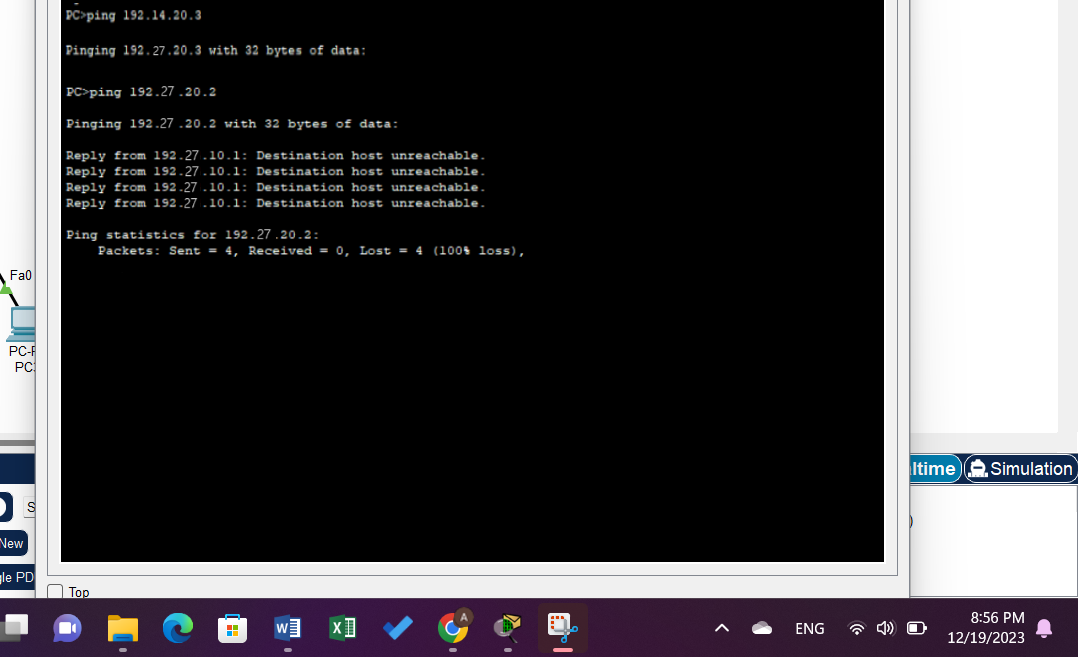


Figure 20:Another Result

### **Task 2:**

**Allow PC0 to access PC2 all other traffic is denied.**

PC0 can ping to PC2 but any ping between any two PC’s deny.

All traffic to PC2 was denied, except for the ones from PC0. This was done by configuring ACL on Router0 using the following commands:

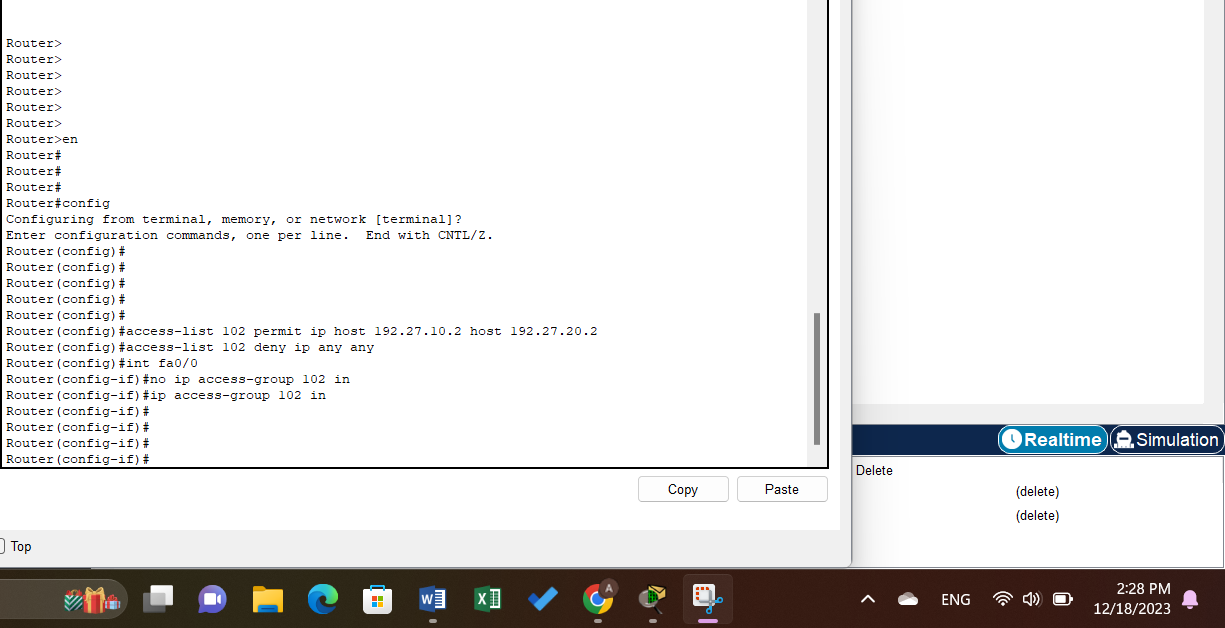


Figure 21:Extended Task B Command

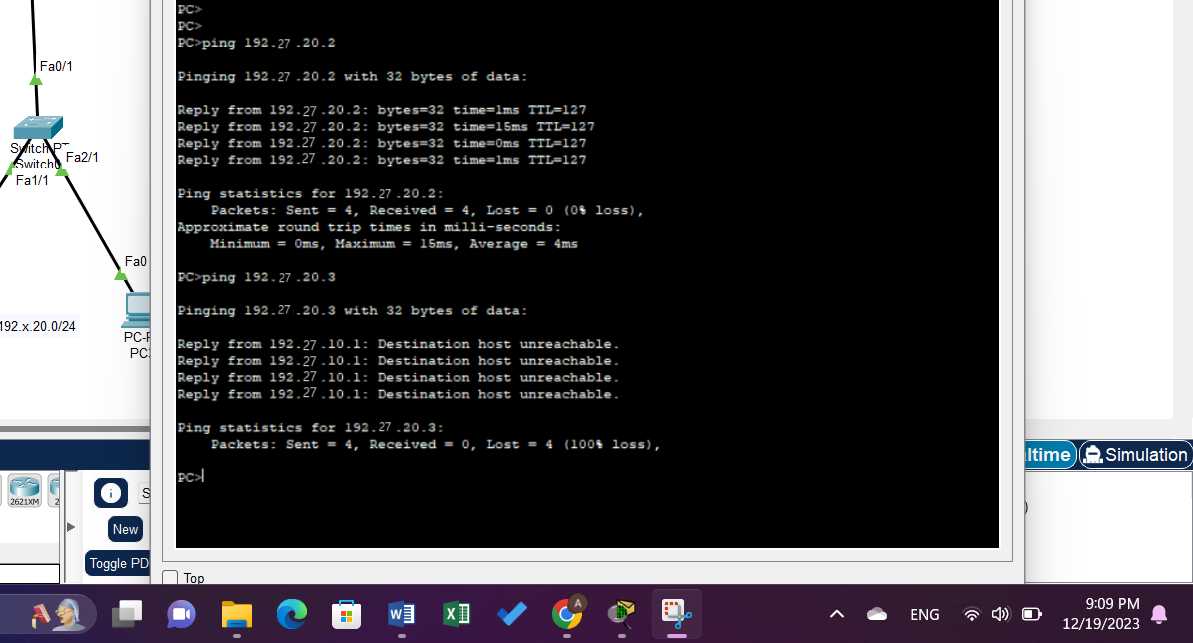


Figure 22:Result of B

### **Task 3:**

**Add a server to the topology to network 192.27.20.0/24 and activate http service on it, then deny PC0 to make HTTP request to this server.**

As seen in figure 15, the server was added to network 192.27.20.0/24's topology. The IP address 192.27.20.4 was assigned to the server, and HTTP services were turned on for it.

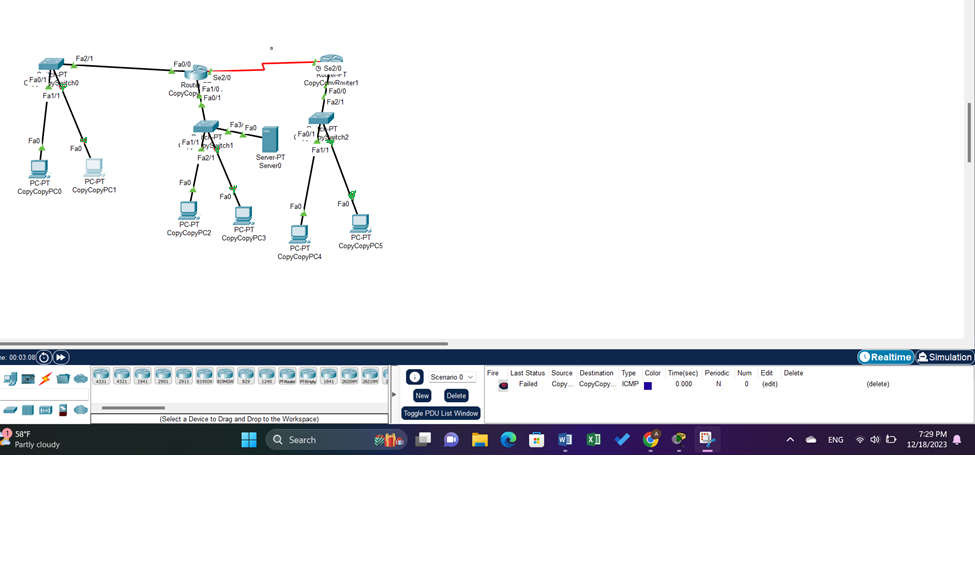


Figure 23:adding server

The ACL was configured on Router0 using the following commands:

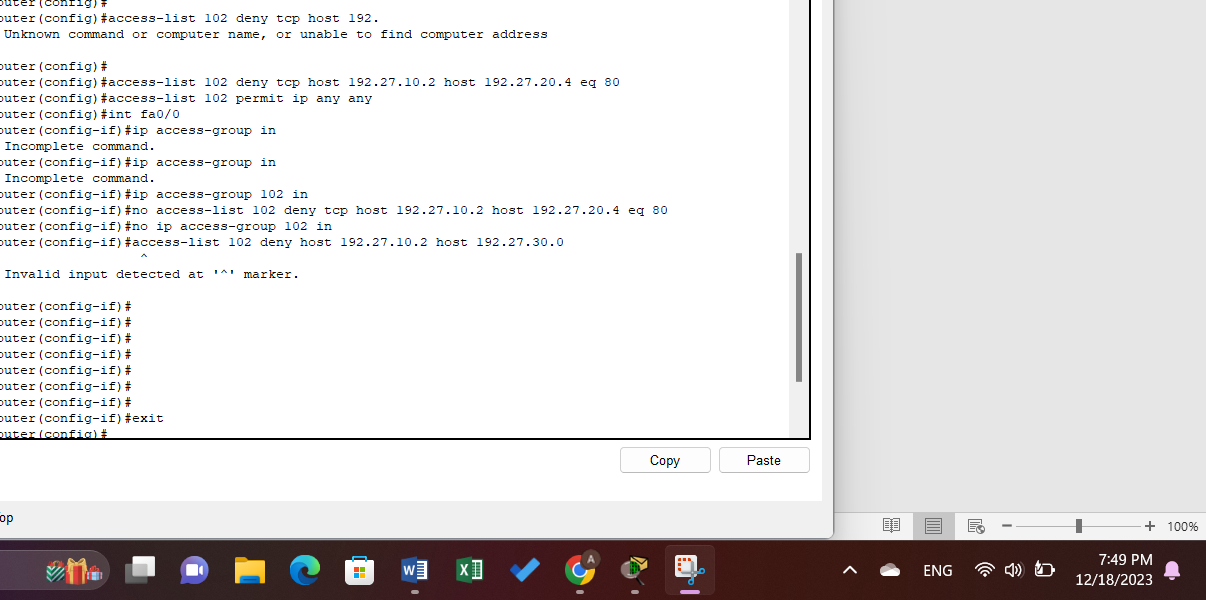


Figure 24:Extended Task C Command

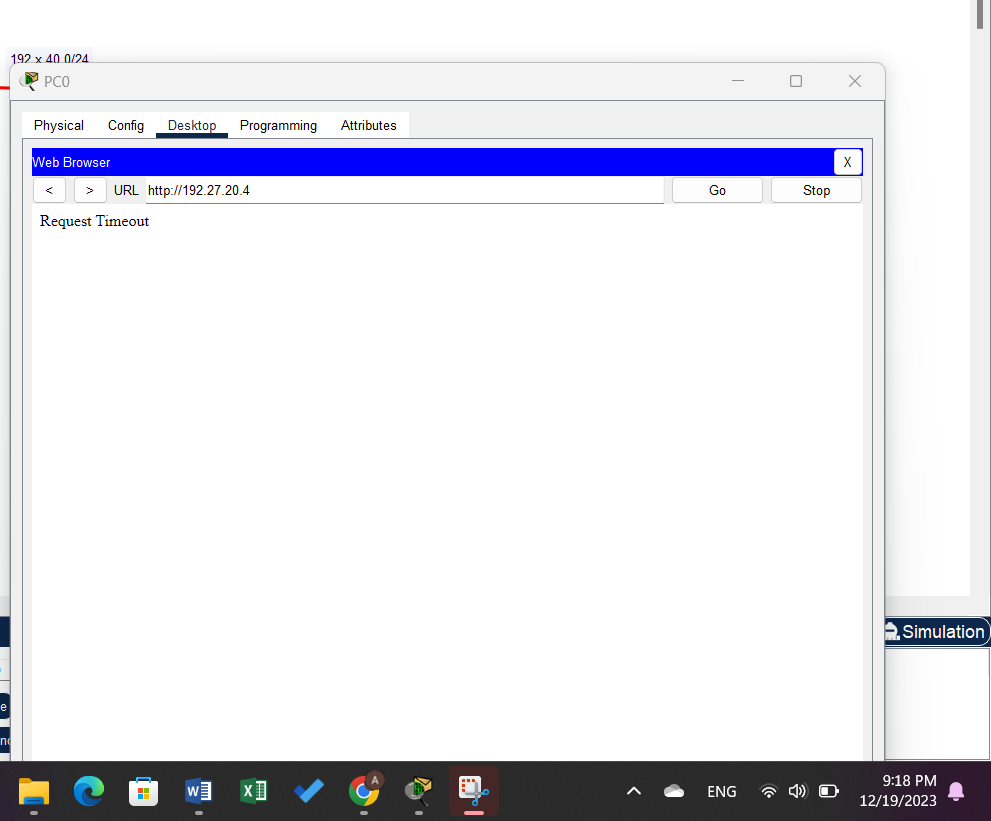


Figure 25:Result of c

### **Task 4:**

**Prevent PC0 from accessing PC4 (192.27.30.2) all other traffic is allowed.**

in this PC0 can’t ping to PC4.

On Router0, an access list akin to the one used in Task 1 was generated, with PC4's IP in place of the destination. The following commands were applied:

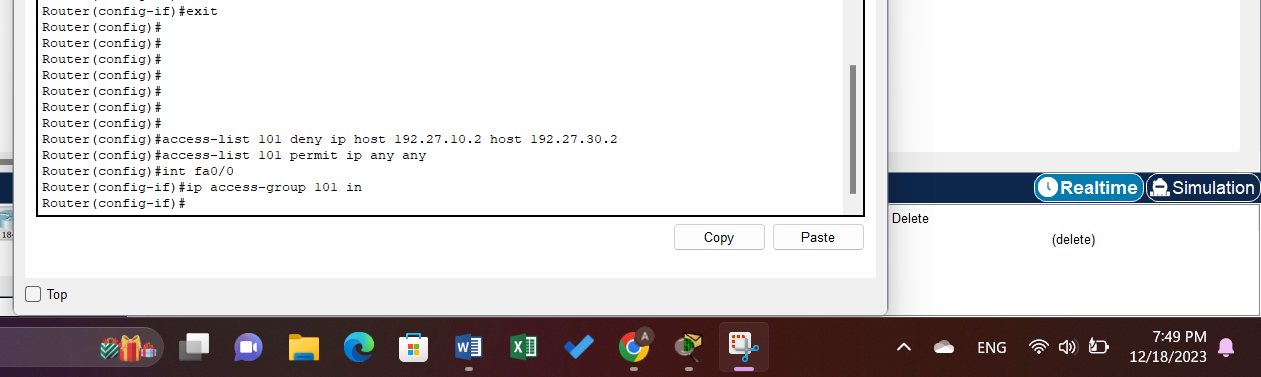


Figure 26:Extended Task D Command



Figure 27:Result

### **E. Task 5:**

**Enable telnet on Router1 then, deny all the hosts from making telnet with interface se2/0(192.27.40.2) of Router1 expect PC0, it can make telnet with any interface. [try to minimize the traffic on the serial line as much as possible]. All other traffic should be allowed.**

To do this, the following commands were used:

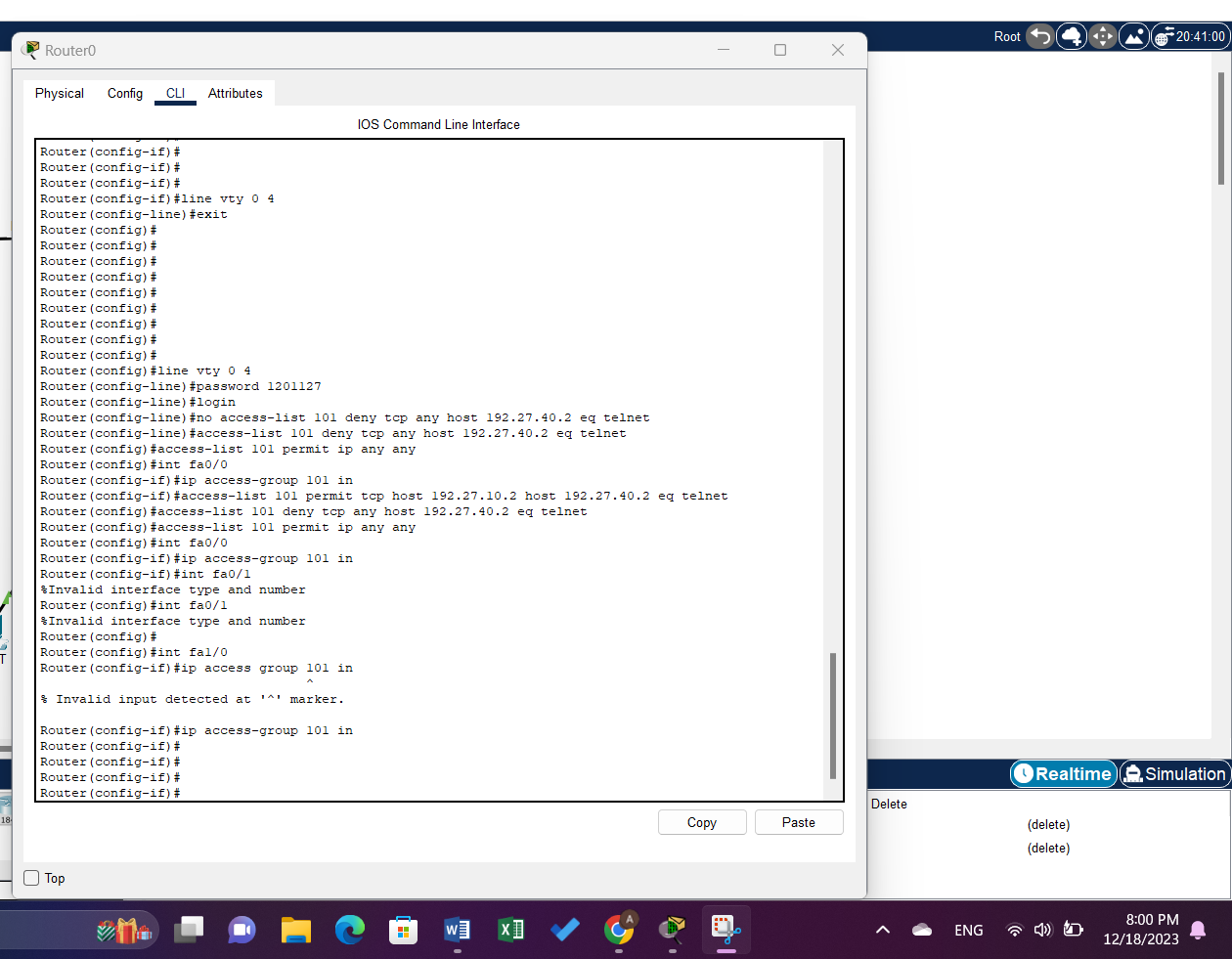


Figure 28:Extended Task E Command Part B

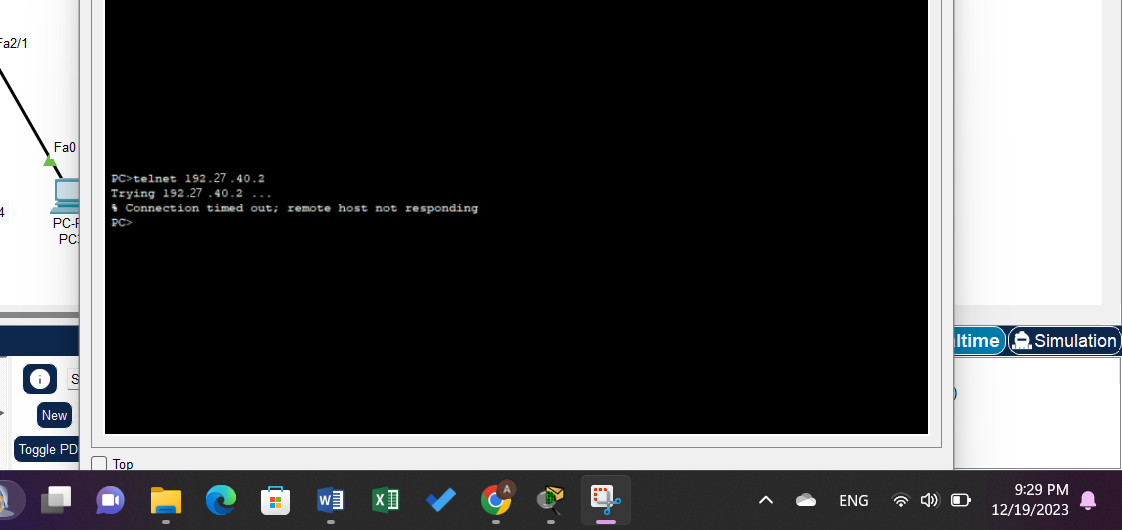


Figure 29:Extended Task E Result

# **Conclusion**

In this experiment, we started by talking about what an ACL is, its advantages, and how to set it from preventing unauthorized traffic from accessing certain devices for security and privacy concerns to regulating the kinds of information that are carried over the network. I believe it's crucial to understand how to set up ACL in order to block unauthorized access to networks or to punish misbehaving hosts there. Fortunately, because I was using the Packet CISCO tracer application on my laptop, I did not experience any issues. I sincerely hope that we can set up a genuine router and create genuine networks.

# **References**

[1]. <https://www.techtarget.com/searchnetworking/definition/access-control-list-ACL>

Accessed on 18 December 8:12pm.

[2]. lab Manual