LABORATORY 04 - APPLICATION LAYER AND PHYSICAL LAYER PROTOCOLS

Laboratory 4 corresponding to the second term

Santiago Amaya Zapata

Julian Camilo Tinjacá Corredor

Escuela Colombiana de Ingeniería Julio Garavito

NETWORK ARCHITECTURE AND SERVICES

GROUP 3

Bogotá DC, Colombia

2025 - 2

Lab No.04 - APPLICATION LAYER AND PHYSICAL LAYER PROTOCOLS

Objectives

- Monitor the application layer protocols.
- Review the structured cabling standard and its application.
- Perform cable punching with RJ-45 connectors and patch panels.

Tools to be used

Items provided by the university
- Computers
- Internet access
- Patch panel and face plate
- Punch tools (for patch cords and impact punches)
- Cable strippers and wire cutters
- Cable tester

Items to be brought by the students
- 4 to 6 meters of UTP/FTP CAT5 or CAT6 cable
- 8 RJ-45 connectors
- If available:
    * Cable stripper or utility knife, and wire cutters
    * Punch tool for patch cords
    * Cable tester

Introduction

Computer networks are the backbone of modern communication systems, allowing information to flow efficiently between devices, services, and users. To ensure reliable communication, it is essential not only to configure and interconnect devices but also to understand how messages travel across the network and how protocols at different layers interact. This laboratory aims to provide hands-on experience with network configuration, service deployment, and protocol analysis using tools such as Cisco Packet Tracer and Wireshark. Through practical exercises, students will configure DNS, HTTP, FTP, and email services, observe packet exchanges, and study structured cabling standards. The objective is

to strengthen the understanding of application and transport layer protocols while gaining technical skills in configuring and monitoring real and simulated networks.

Theoretical Framework

Computer networks are organized following the layered model, with the **OSI model** and the **TCP/IP model** being the most common references. In this lab, particular attention is given to the **application layer** and the **transport layer**. The application layer is responsible for providing network services to end-users, such as web browsing (HTTP), email communication (SMTP, POP3), domain name resolution (DNS), and file transfer (FTP). The transport layer, on the other hand, ensures that data is delivered reliably and in the correct order, typically through protocols such as TCP and UDP. Transport layer ports serve as logical endpoints that allow different services to coexist on the same machine.

The **DNS (Domain Name System)** service translates human-readable domain names into IP addresses, enabling communication between clients and servers without the need to memorize numerical addresses. **HTTP (Hypertext Transfer Protocol)** allows the exchange of web content, forming the foundation of the World Wide Web. **Email services** rely on protocols such as SMTP for sending messages and POP3/IMAP for retrieving them. **FTP (File Transfer Protocol)** enables file exchange between clients and servers, supporting both uploading and downloading operations.

Beyond services, **network simulation tools** like Cisco Packet Tracer are valuable for visualizing packet flow, testing configurations, and verifying connectivity. Similarly, **Wireshark** is a powerful packet analyzer that captures and decodes traffic, allowing users to study headers and payloads in detail, particularly at the application and transport layers.

Finally, the laboratory incorporates **structured cabling**, which ensures physical connectivity between network devices. Standards for cabling, such as straight-through and crossover cables, as well as the use of patch panels and faceplates, are crucial for building scalable and reliable infrastructures. Understanding both the logical and physical aspects of networking provides a comprehensive view of how communication systems are designed, implemented, and maintained.

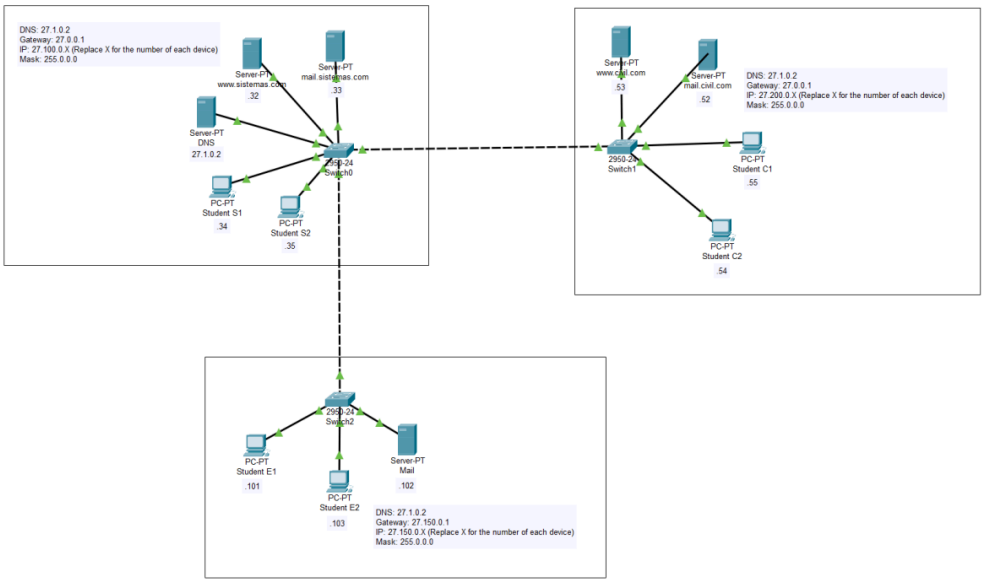
Through these theoretical foundations, the laboratory provides students with the necessary knowledge to configure, analyze, and evaluate network services while reinforcing the practical importance of standardized communication protocols and physical infrastructure.

Experiments

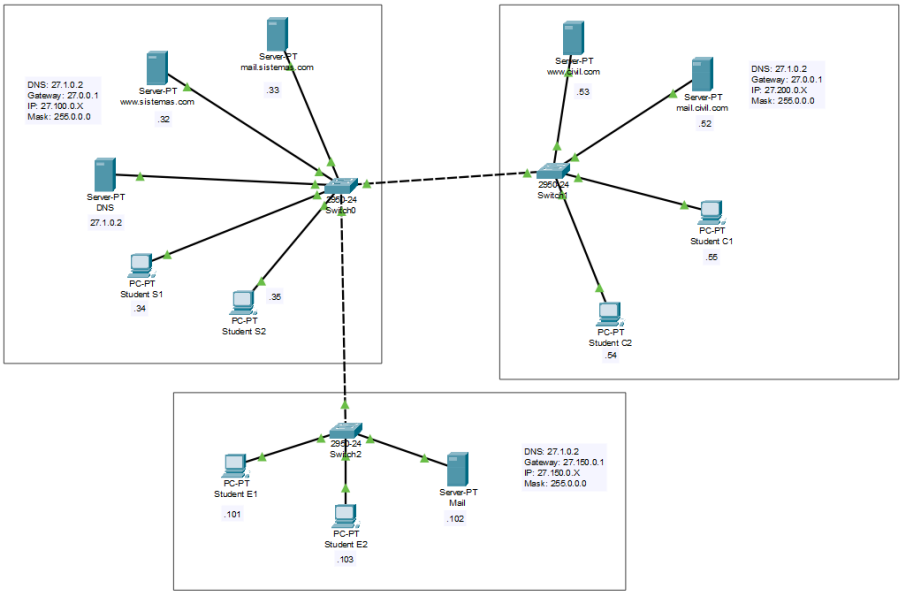
Understanding how messages circulate across the network and being able to analyze their contents is important for reviewing and fine-tuning the network. In this part of the lab, we will review the information on the application layer and transport layer protocols (only the ports) that we have covered.

1. Cisco Packet Tracer [For groups of 1, 2, and 3 students]

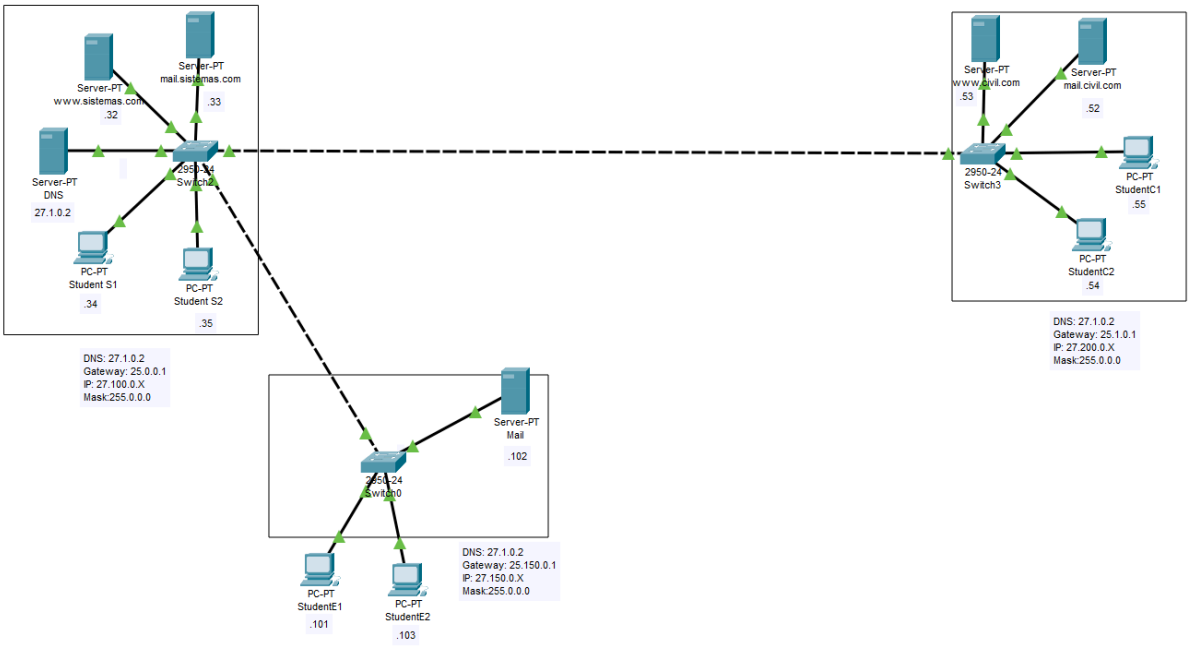
Using Packet Tracer, each student should configure the network presented below and document the experience. The services for DNS, HTTP, FTP, and email must be configured on the servers for the network presented.



Santiago´s diagram:



Julian´s Diagram:

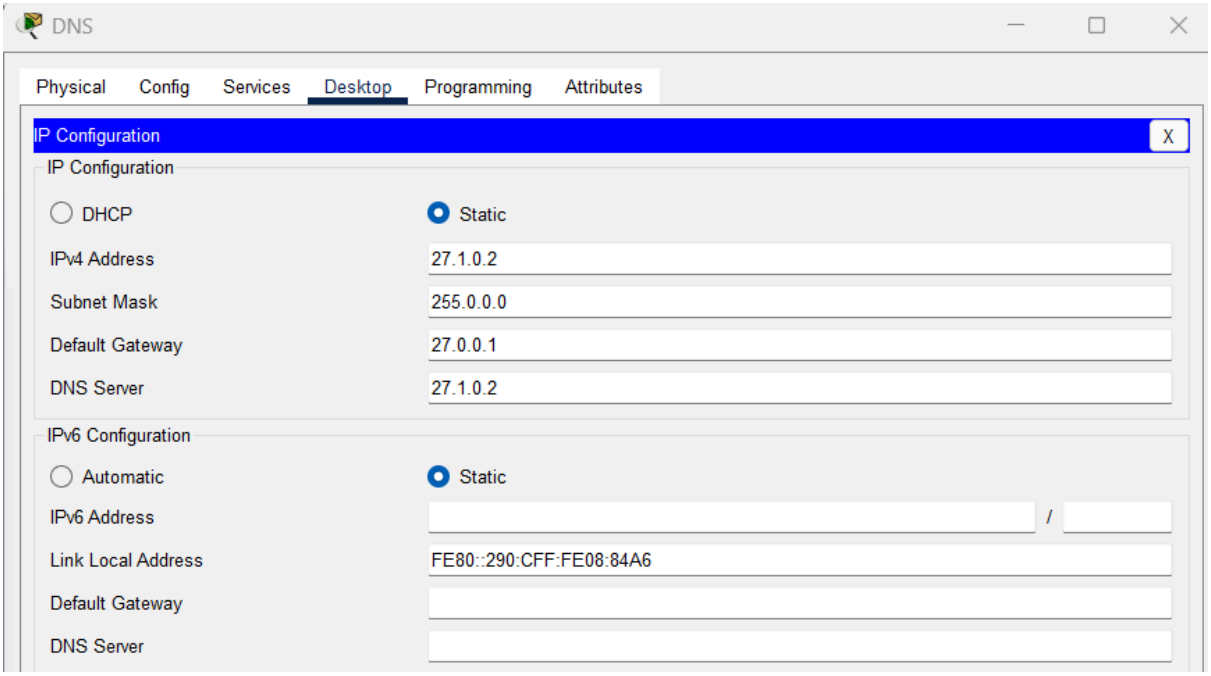


2. Network Configuration [For groups of 1, 2, and 3 students]

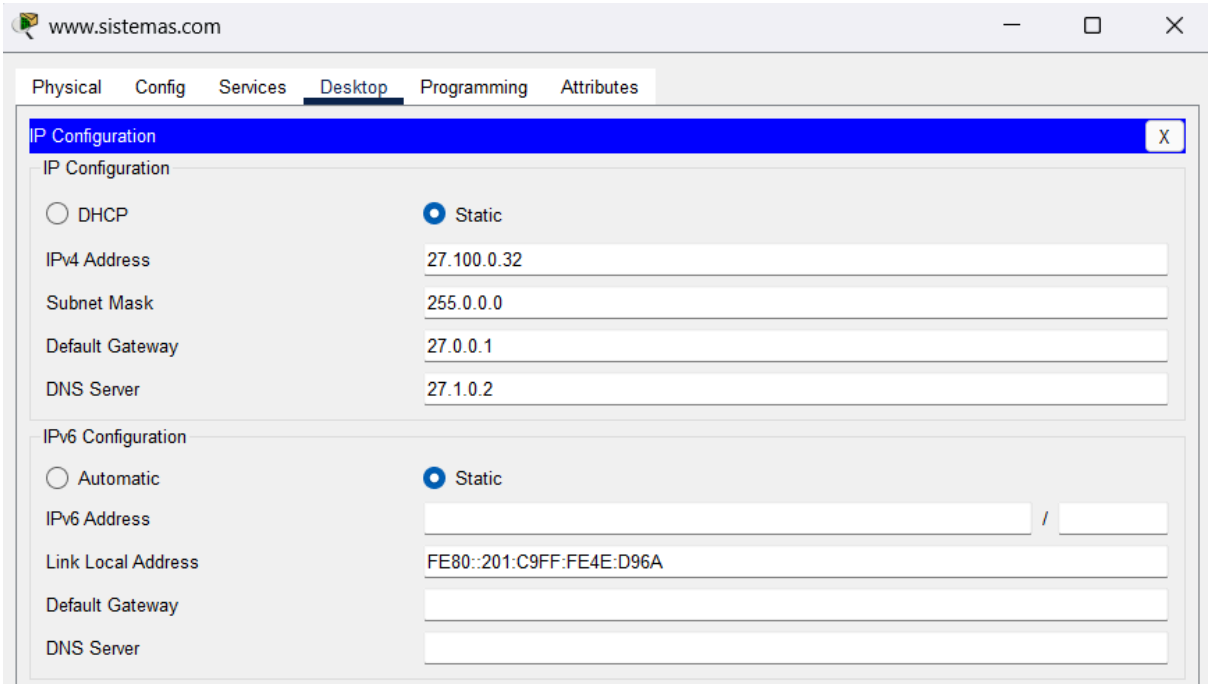
- Include the servers and clients presented, connect the cables, and assign DNS, Gateway, IP Address, and Subnet Mask to each device following the configuration shown in the diagram.

On sistemas.com

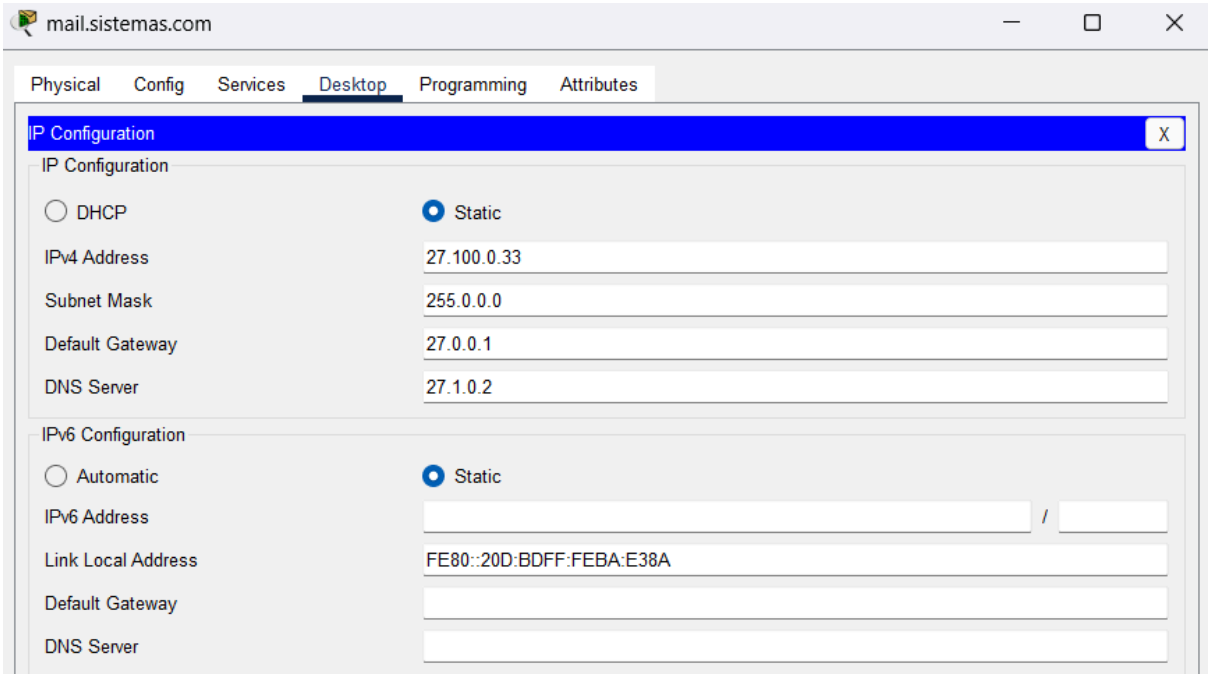
DNS Server:



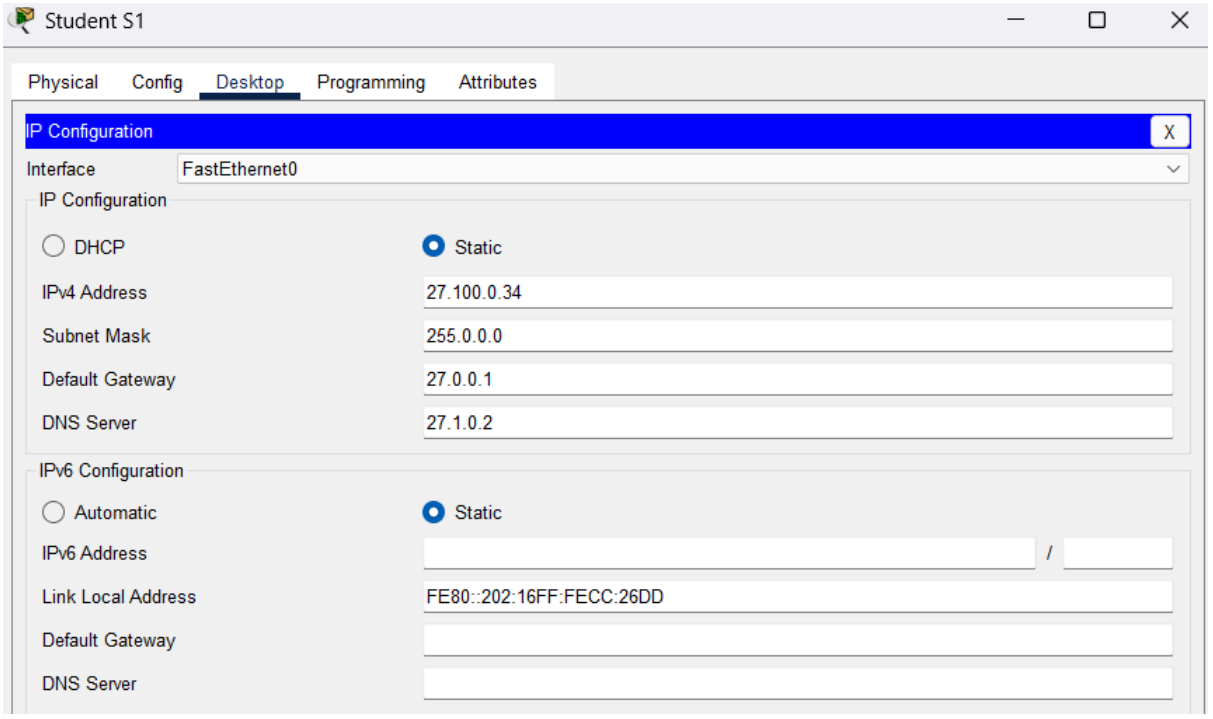
www.sistemas.com server:



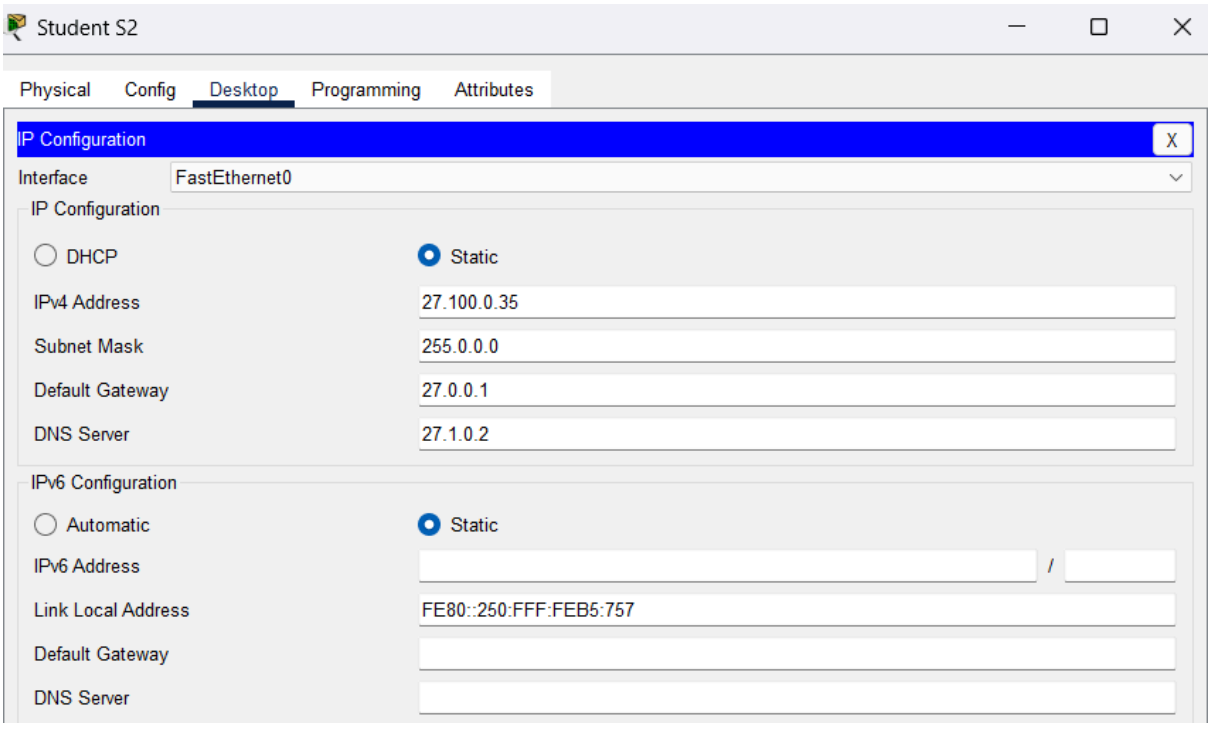
mail.sistemas.com server:



Student S1 PC:

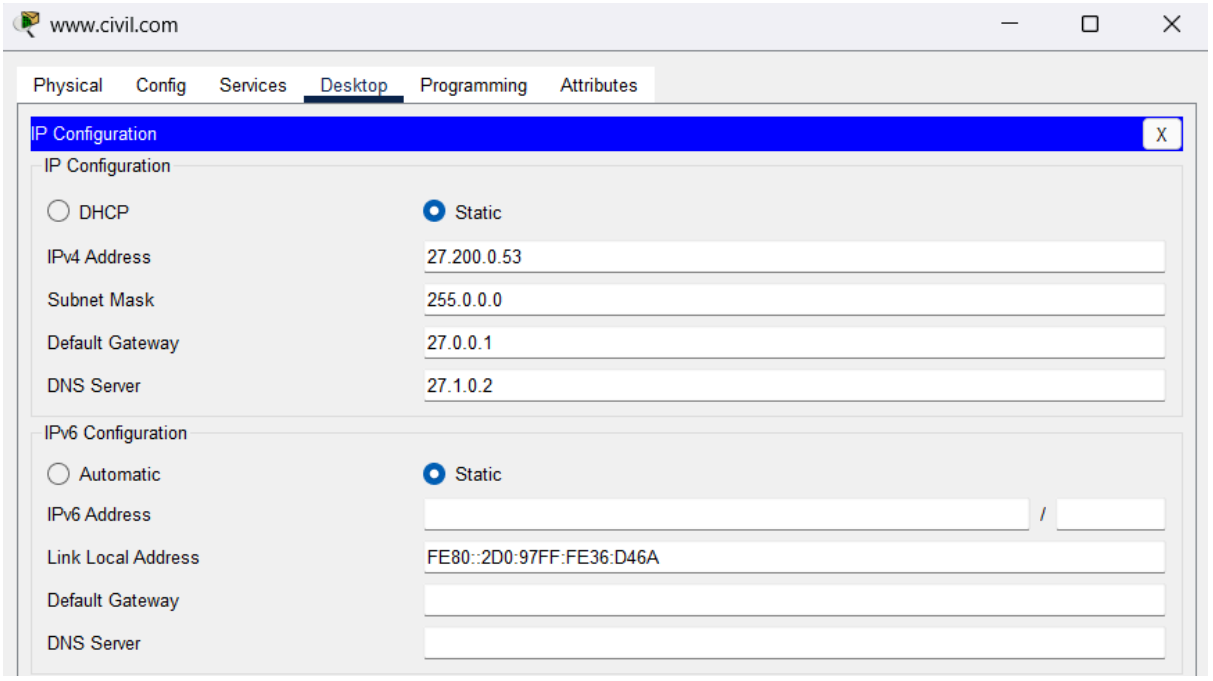


Student S2 PC:

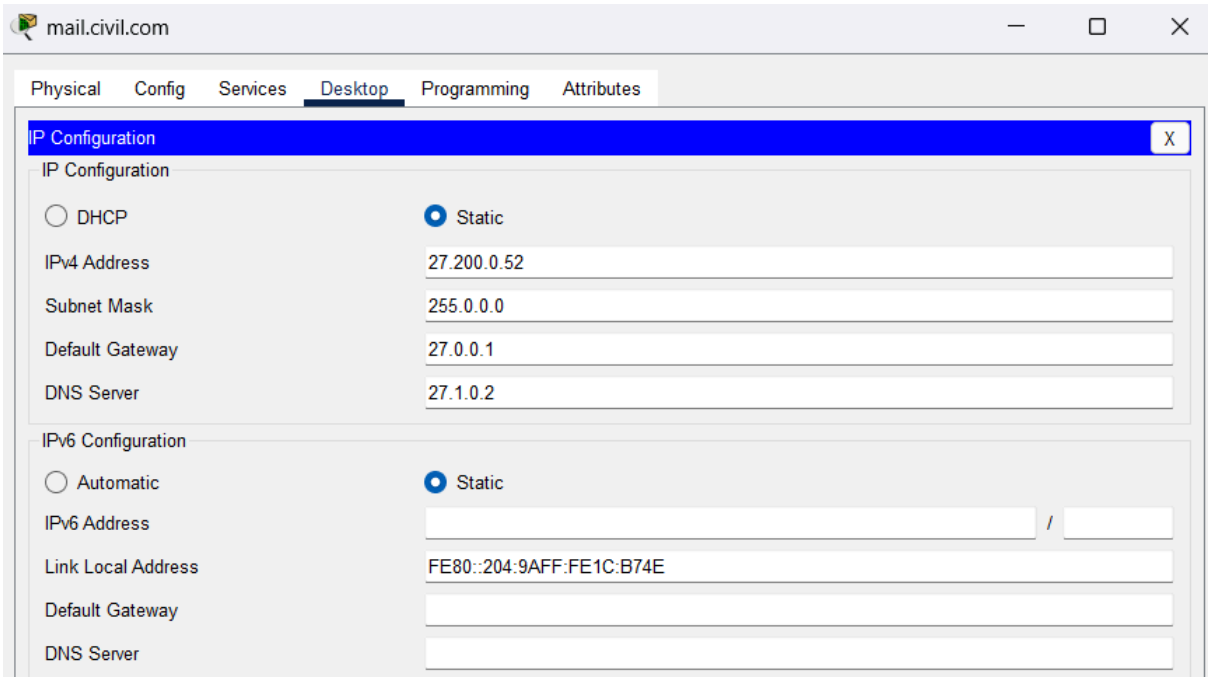


On civil.com

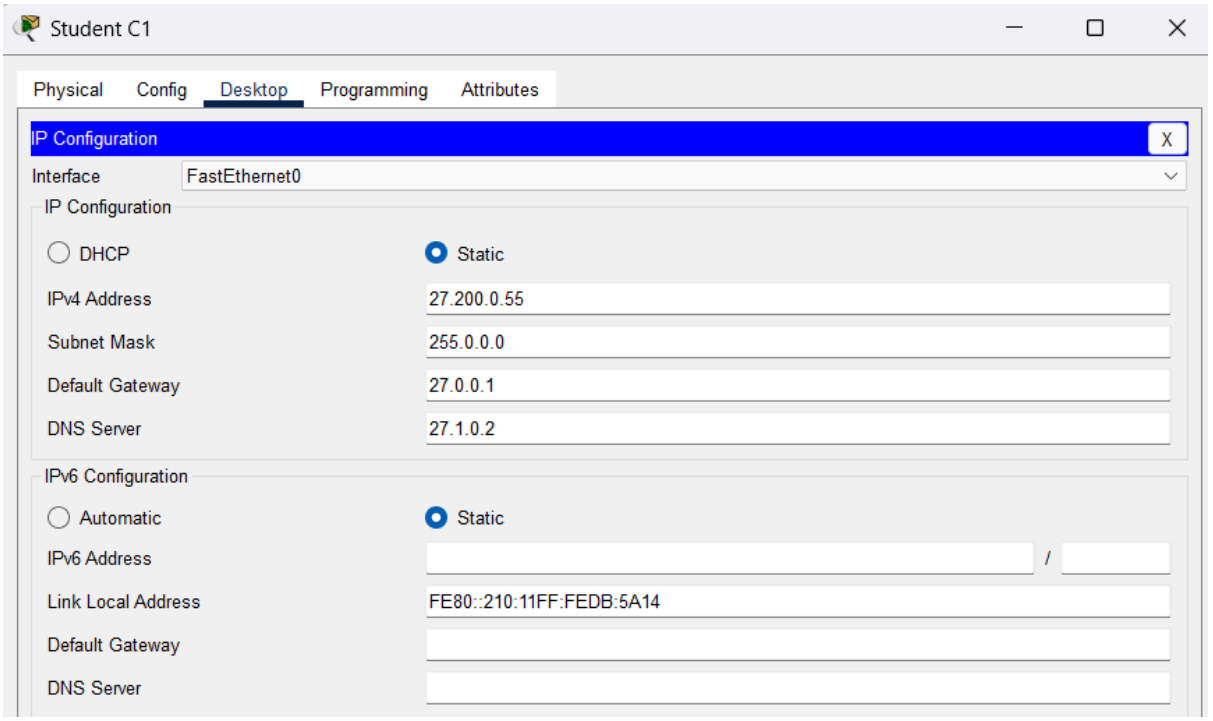
www.civil.com server:



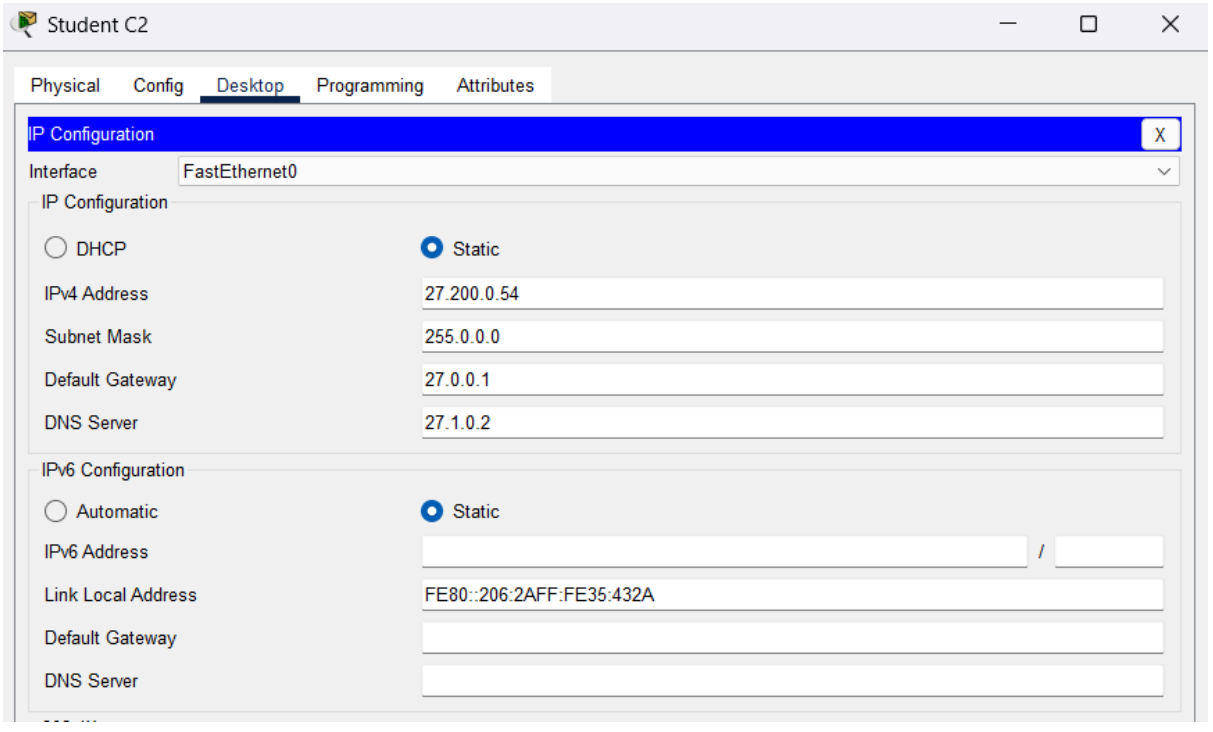
mail.civil.com server:



Student C1 PC:

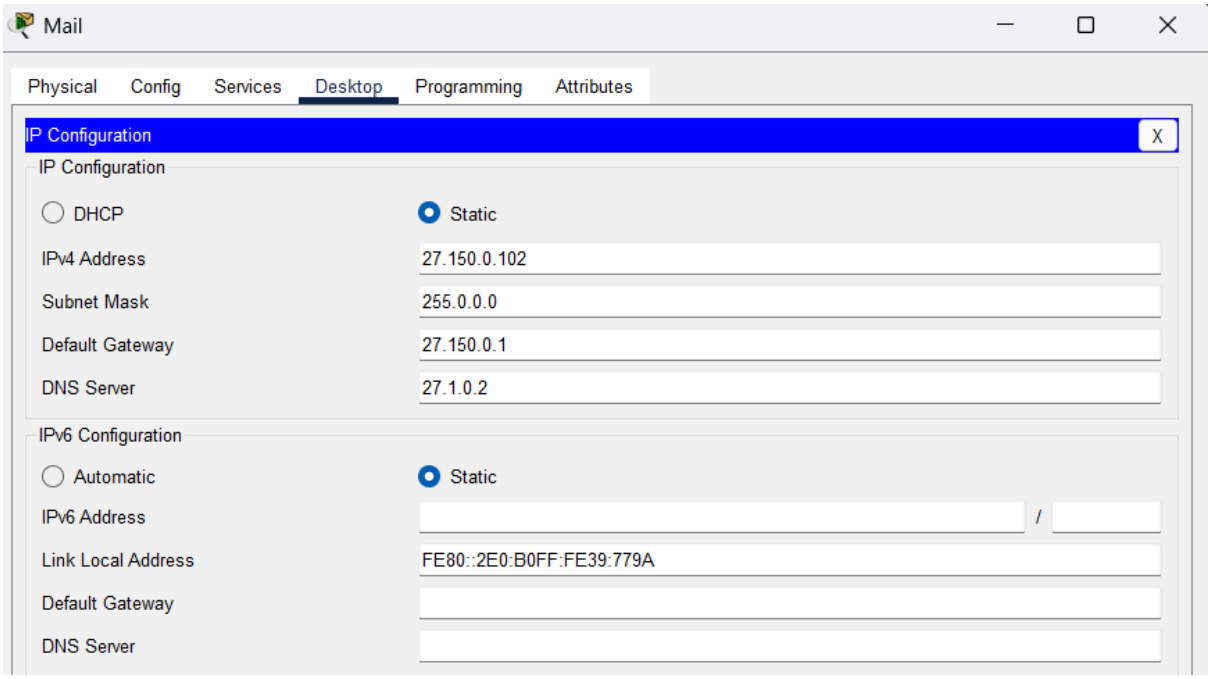


Student C2 PC:

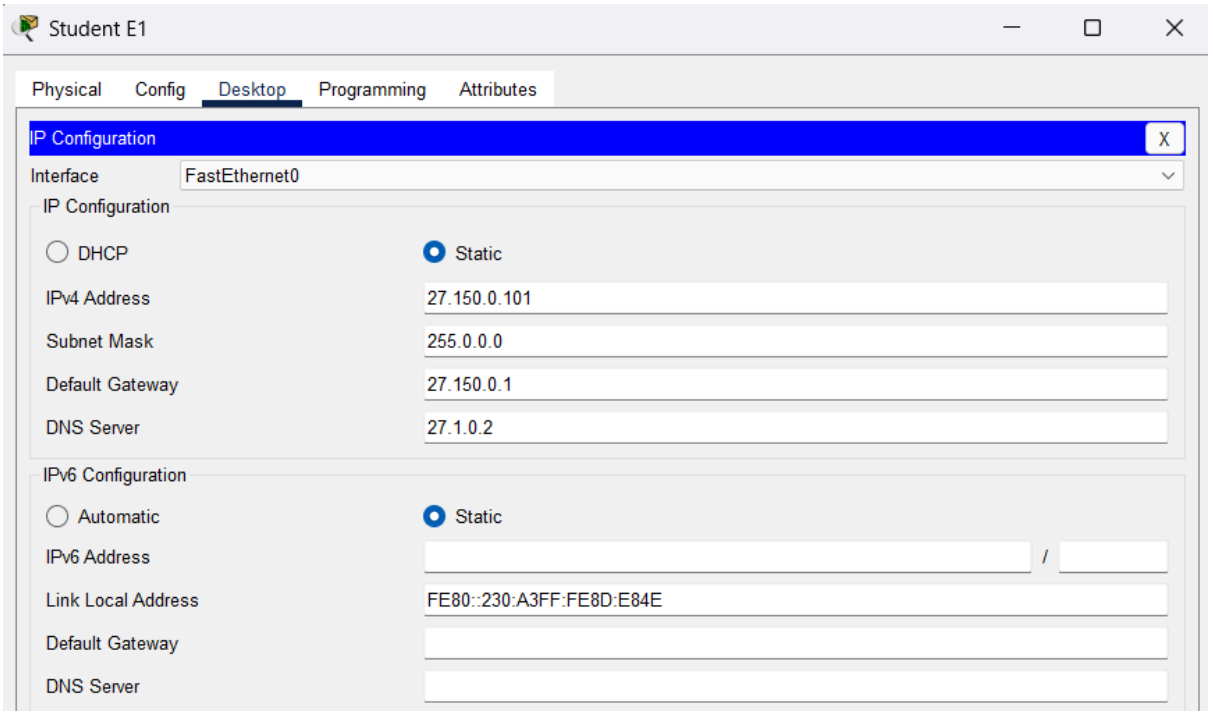


On electrica.com

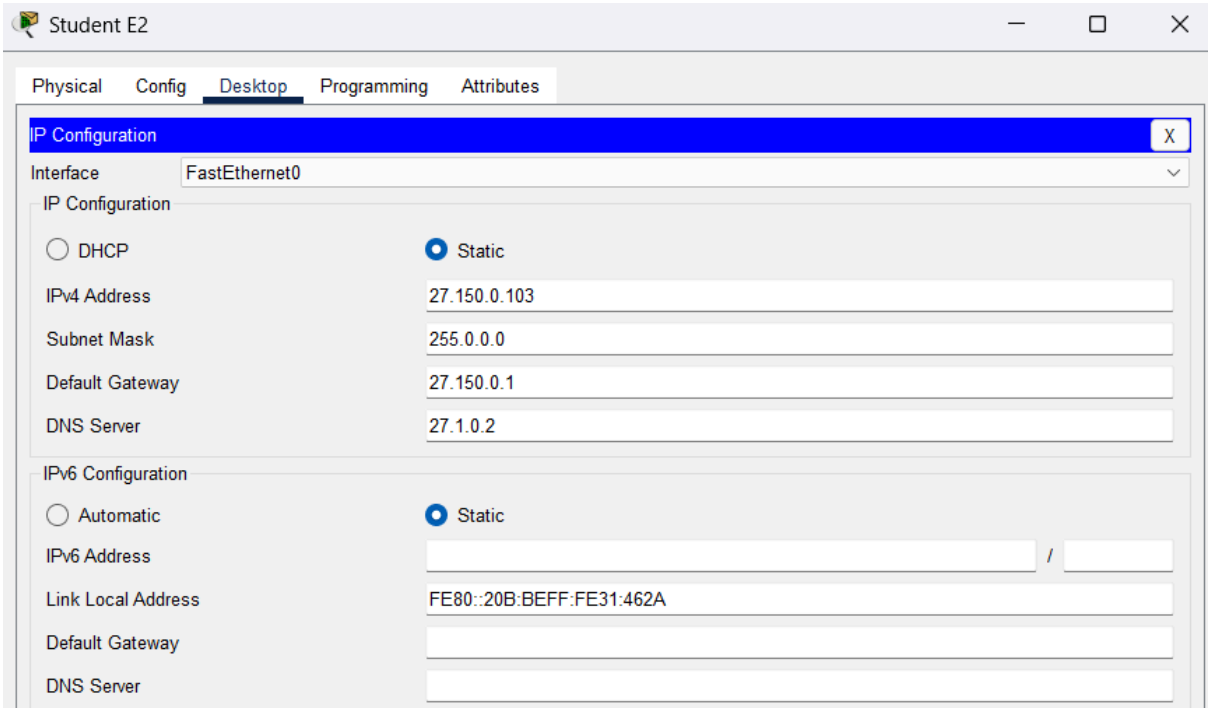
Mail server:



Student E1 PC:

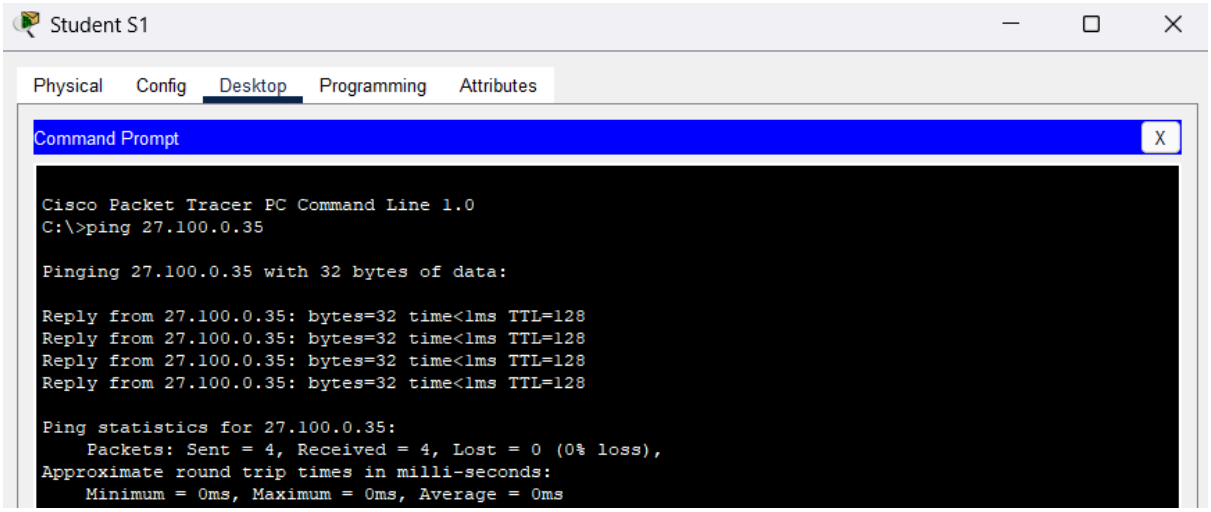


Student E2 PC:



- Send messages between the devices in the network and verify connectivity between all of them.

Using Student S1 PC:
We ping Student S2 PC(IP: 27.100.0.35):



ping www.sistemas.com (IP: 27.100.0.32):

```
Pinging 27.100.0.32 with 32 bytes of data:

Reply from 27.100.0.32: bytes=32 time<1ms TTL=128
Reply from 27.100.0.32: bytes=32 time<1ms TTL=128
Reply from 27.100.0.32: bytes=32 time<1ms TTL=128
Reply from 27.100.0.32: bytes=32 time<1ms TTL=128

Ping statistics for 27.100.0.32:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

We also tried pinging devices from other domains.

ping www.civil.com (IP: 27.200.0.53):

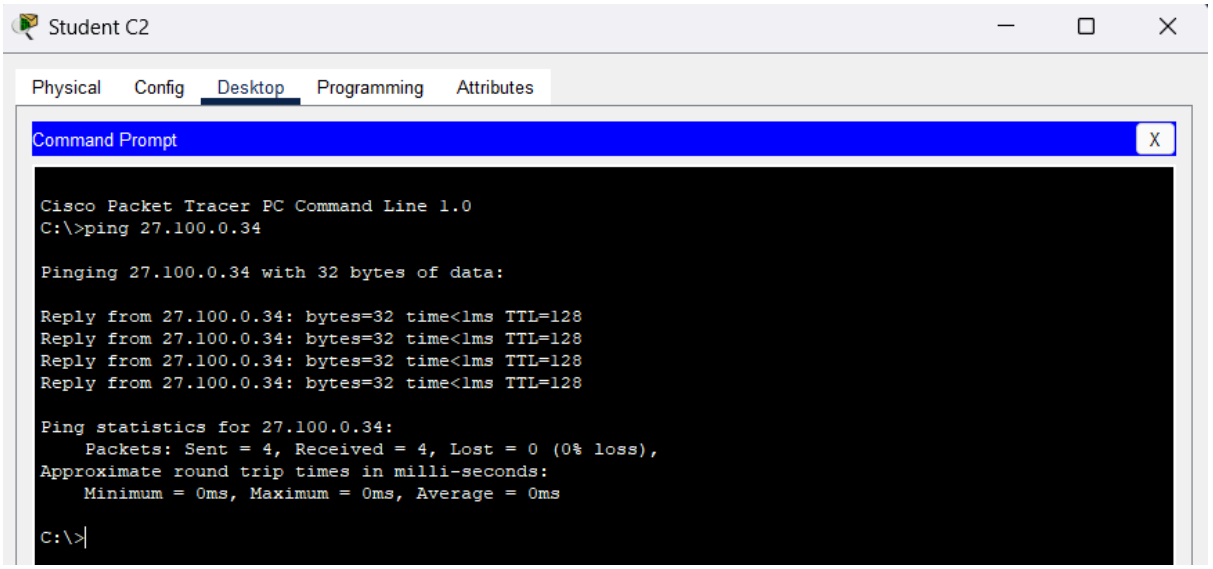
```
C:\>ping 27.200.0.53

Pinging 27.200.0.53 with 32 bytes of data:

Reply from 27.200.0.53: bytes=32 time<1ms TTL=128
Reply from 27.200.0.53: bytes=32 time=7ms TTL=128
Reply from 27.200.0.53: bytes=32 time<1ms TTL=128
Reply from 27.200.0.53: bytes=32 time<1ms TTL=128

Ping statistics for 27.200.0.53:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 7ms, Average = 1ms
```

ping Student E1 PC (IP: 27.150.0.101):

```
C:\>ping 27.150.0.101

Pinging 27.150.0.101 with 32 bytes of data:

Reply from 27.150.0.101: bytes=32 time<1ms TTL=128
Reply from 27.150.0.101: bytes=32 time<1ms TTL=128
Reply from 27.150.0.101: bytes=32 time<1ms TTL=128
Reply from 27.150.0.101: bytes=32 time<1ms TTL=128

Ping statistics for 27.150.0.101:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Now from another device we ping Student S1 PC:

Using Student C2 PC:

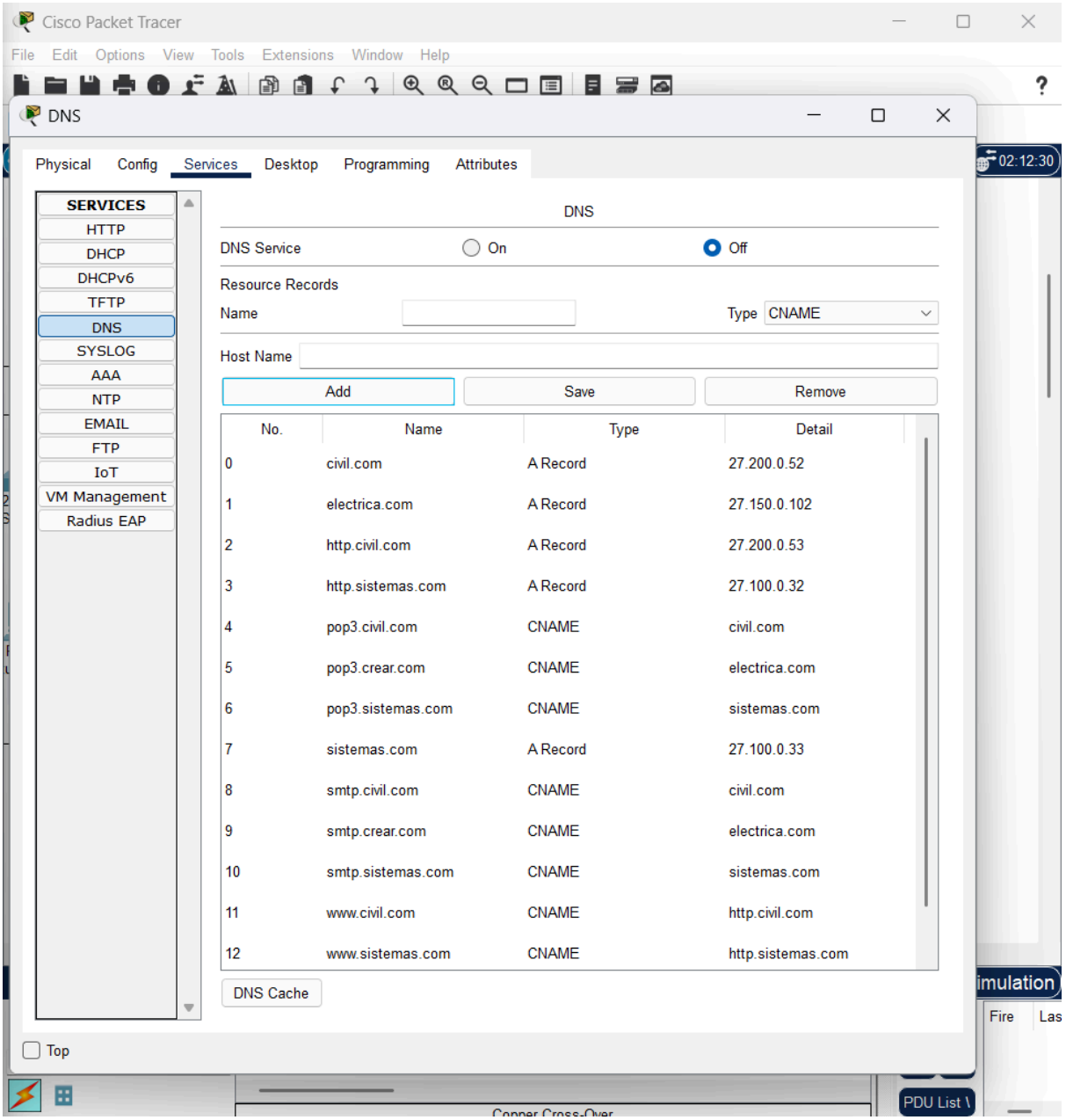


3. Service Configuration

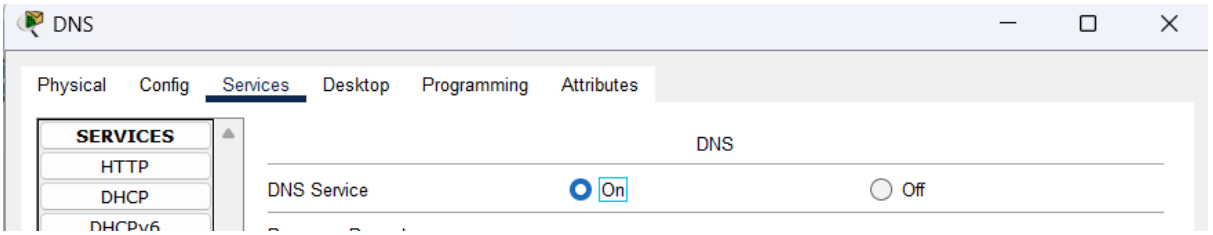
a. DNS.

In the DNS service with IP 27.1.0.2, include the following entries:
- sistemas.com with the IP of the mail server for sistemas.com
- pop3.sistemas.com as an alias for sistemas.com 2
- smtp.sistemas.com as an alias for sistemas.com
- http.sistemas.com with the IP of the web server for sistemas.com
- www.sistemas.com as an alias for http.sistemas.com
- civil.com with the IP of the mail server for civil.com
- pop3.civil.com as an alias for civil.com
- smtp.civil.com as an alias for civil.com
- http.civil.com with the IP of the web server for civil.com
- www.civil.com as an alias for http.civil.com

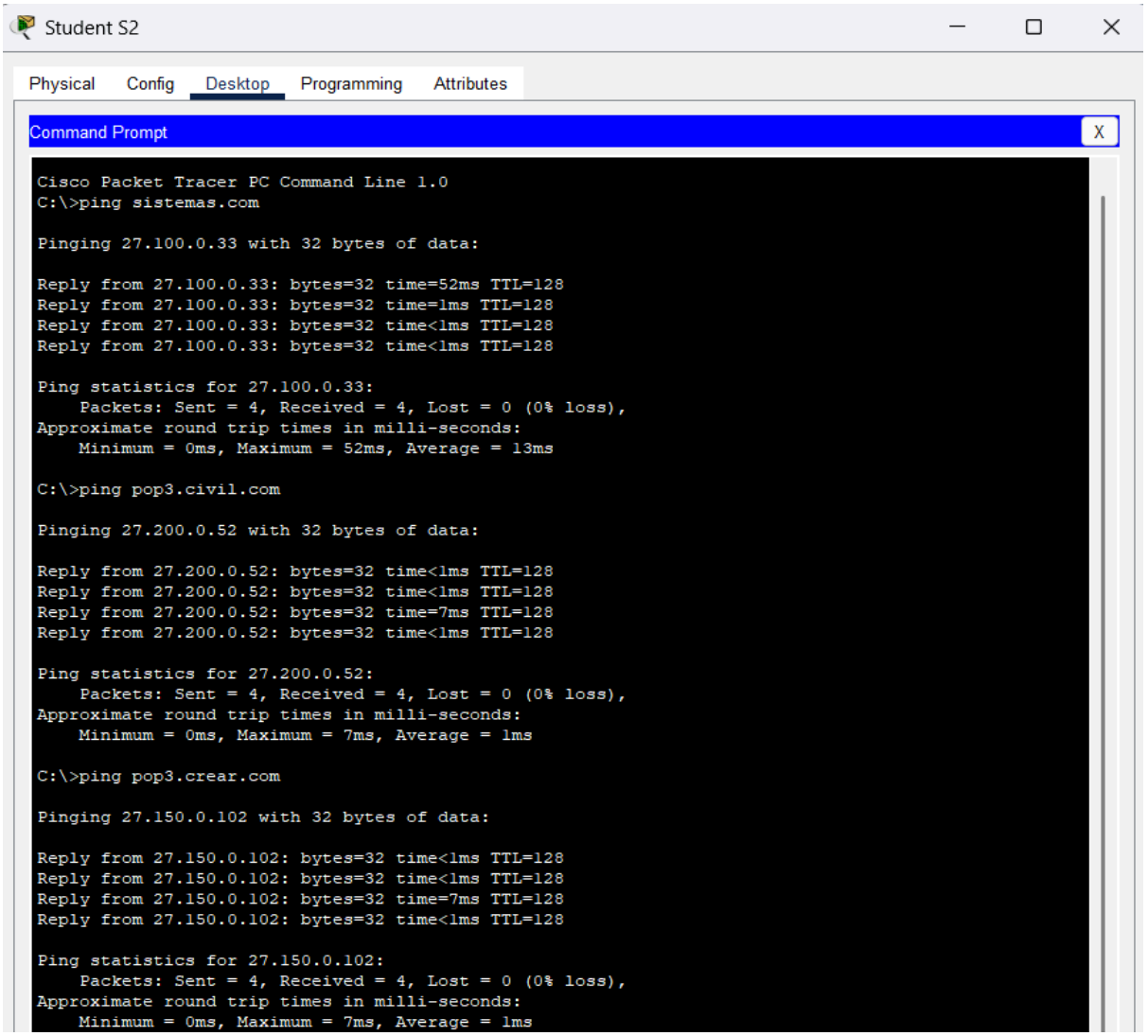
In the DNS server with IP 27.1.0.2, include the following entries:
- electrica.com with the IP of the mail server for electrica.com
- pop3.crear.com as an alias for electrica.com
- smtp.crear.com as an alias for electrica.com



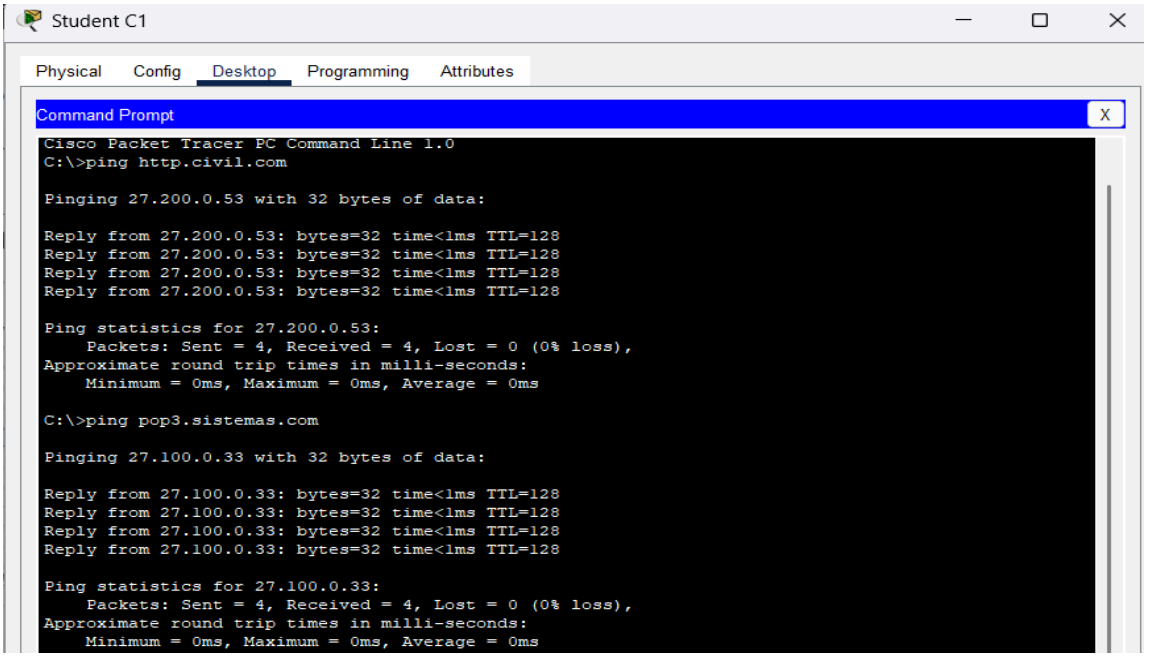
Start the service, and from a client machine in each company, use the ping command by name in the command line to verify that the service is working properly.



For the company sistemas.com, Student S2 PC was used for testing:

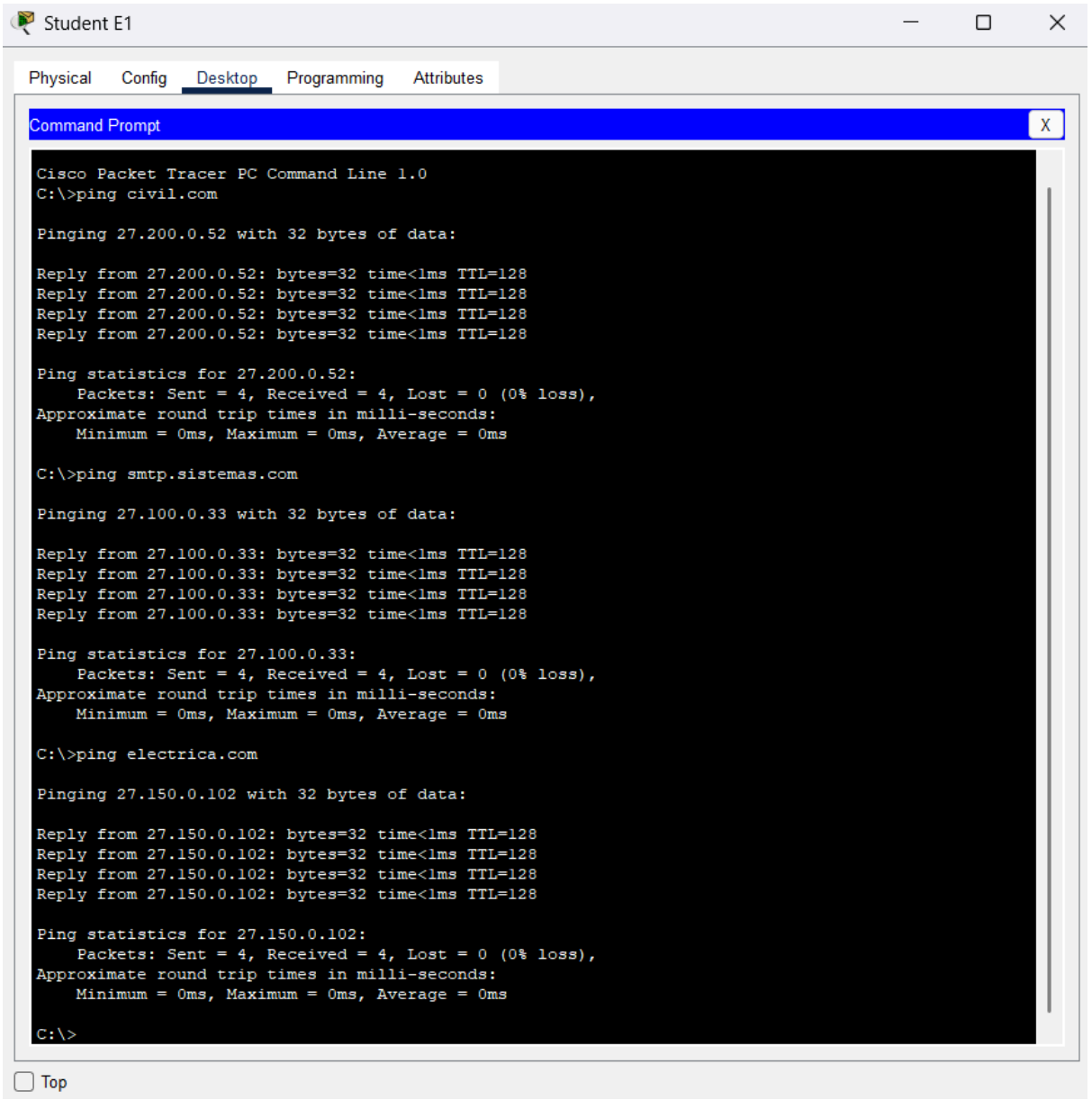


For the company civil.com, Student C1 PC was used for testing:



```
C:\>ping smtp.crear.com

Pinging 27.150.0.102 with 32 bytes of data:

Reply from 27.150.0.102: bytes=32 time<1ms TTL=128
Reply from 27.150.0.102: bytes=32 time<1ms TTL=128
Reply from 27.150.0.102: bytes=32 time<1ms TTL=128
Reply from 27.150.0.102: bytes=32 time<1ms TTL=128

Ping statistics for 27.150.0.102:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

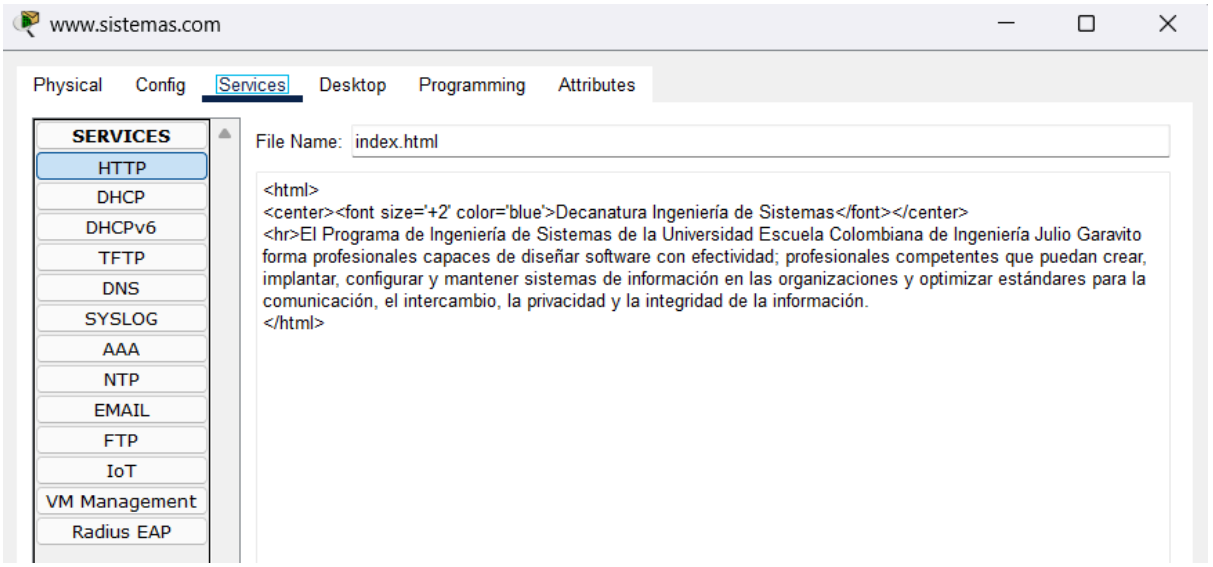
For the company electrica.com, Student E1 PC was used for testing:



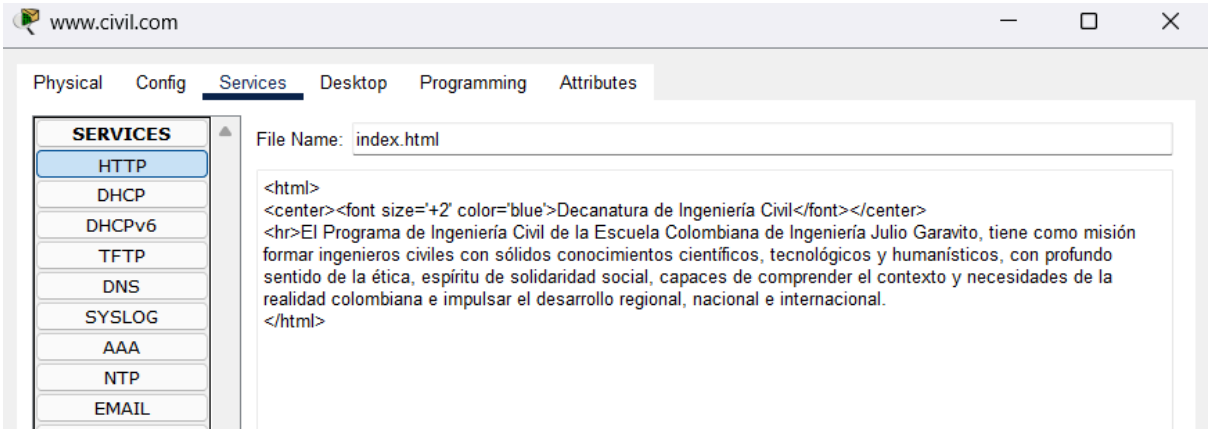
b. HTTP

On the web servers, configure the HTTP service. Modify the web pages of the servers to recognize which faculty they belong to (personalize it for each faculty). Start the service.

On www.sistemas.com server:

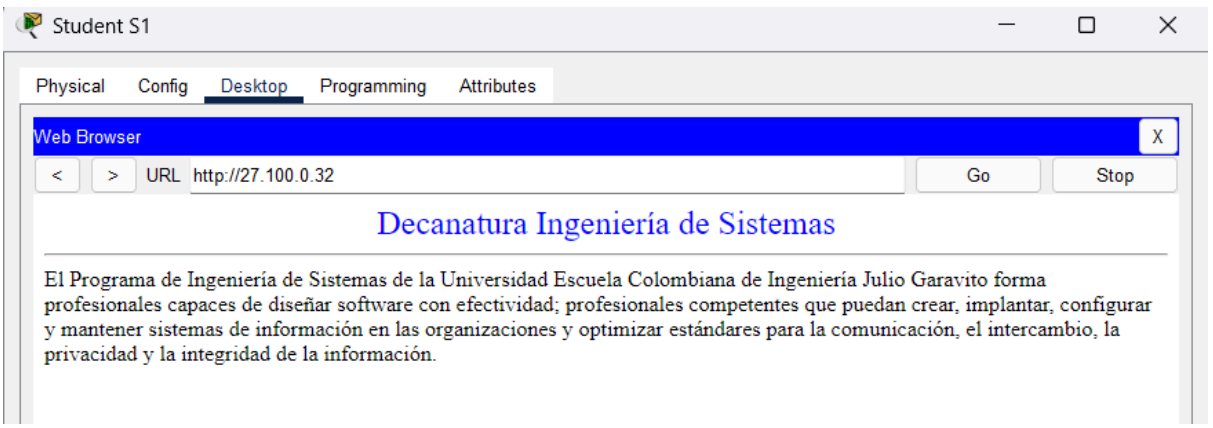


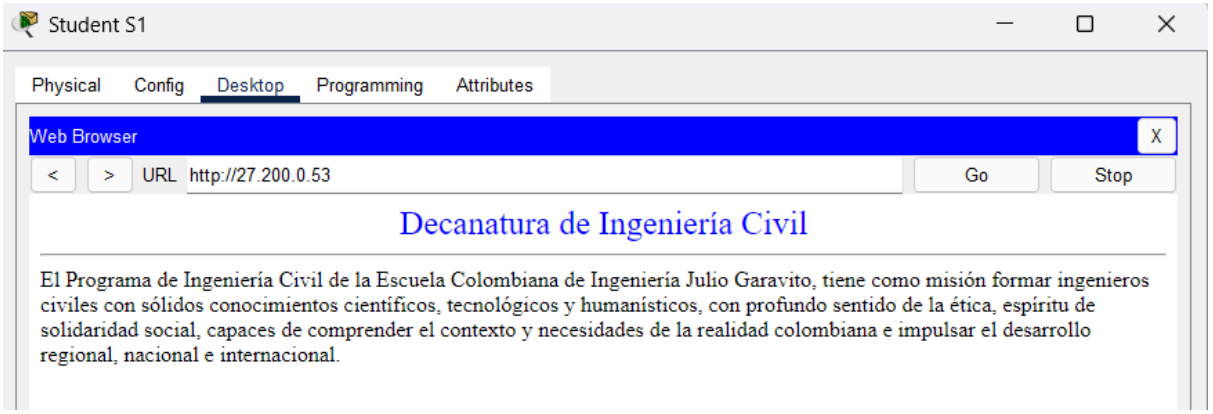
On www.civil.com server:



From the client stations, try connecting to the web servers:
- Make the request for the web page using the IP addresses of each server.

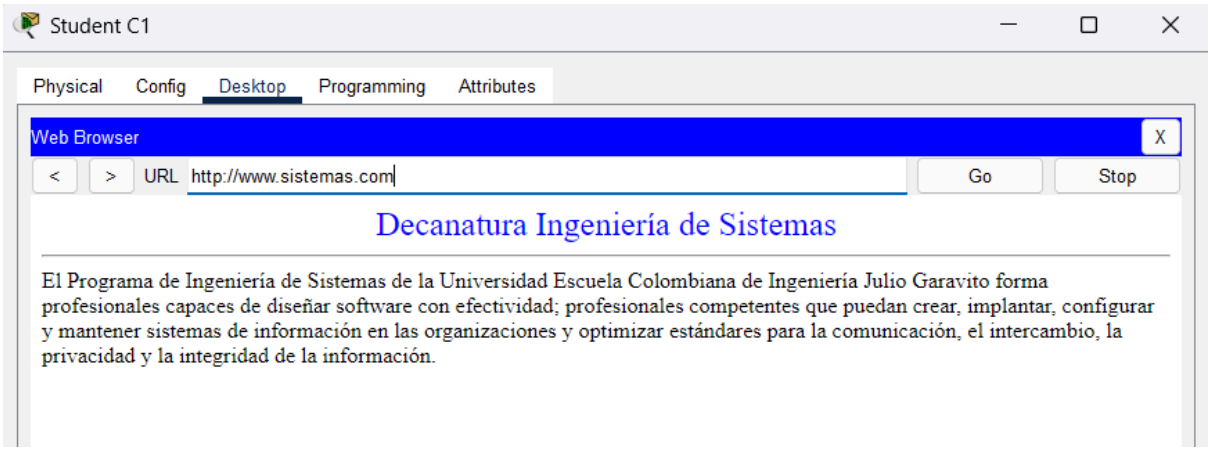
Student S1 PC was used for testing:

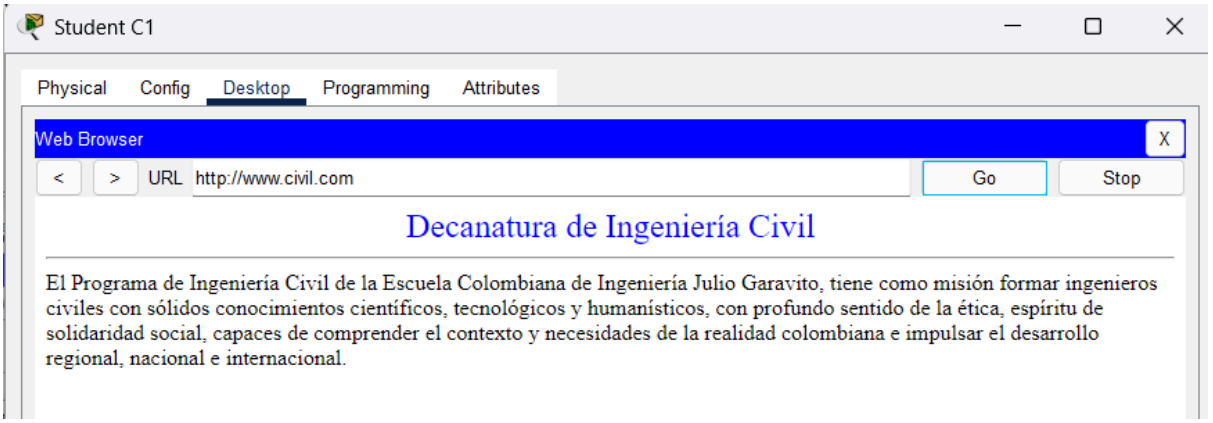




- Make the request for the web page using the URL of each server.

Student C1 PC was used for testing:



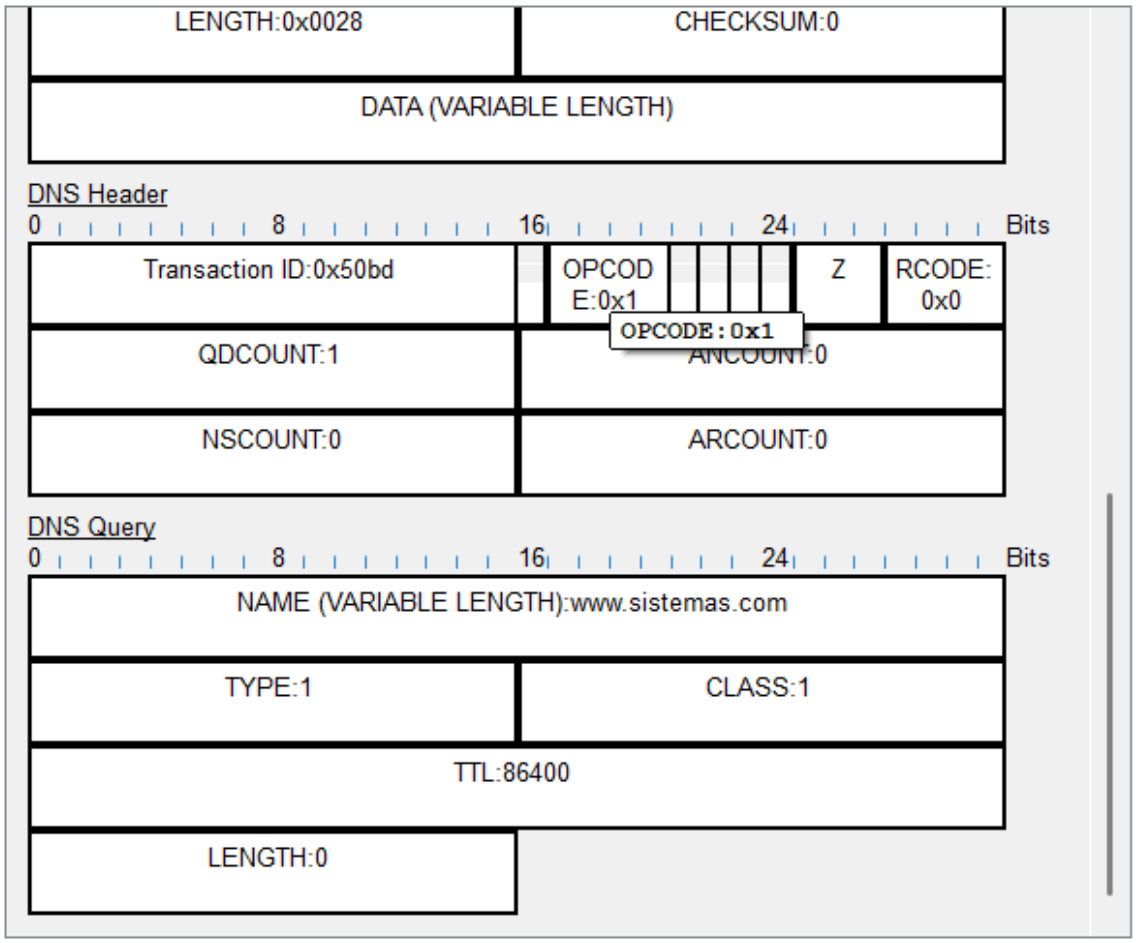


- Using simulation mode, check the contents of the PDU at the application layer.

PDU Information at Device: Student S1

OSI Model | Outbound PDU Details

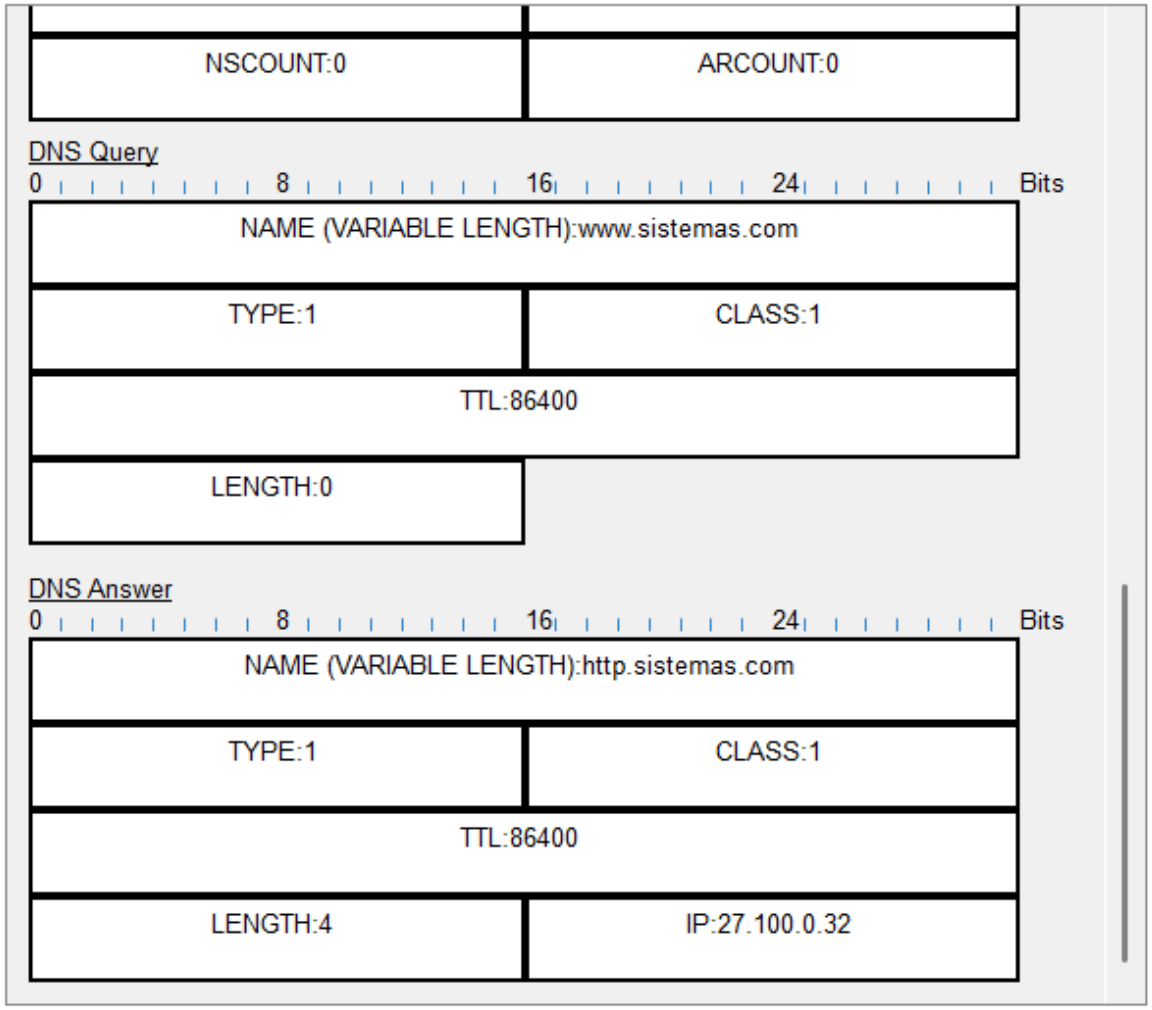
PDU Formats

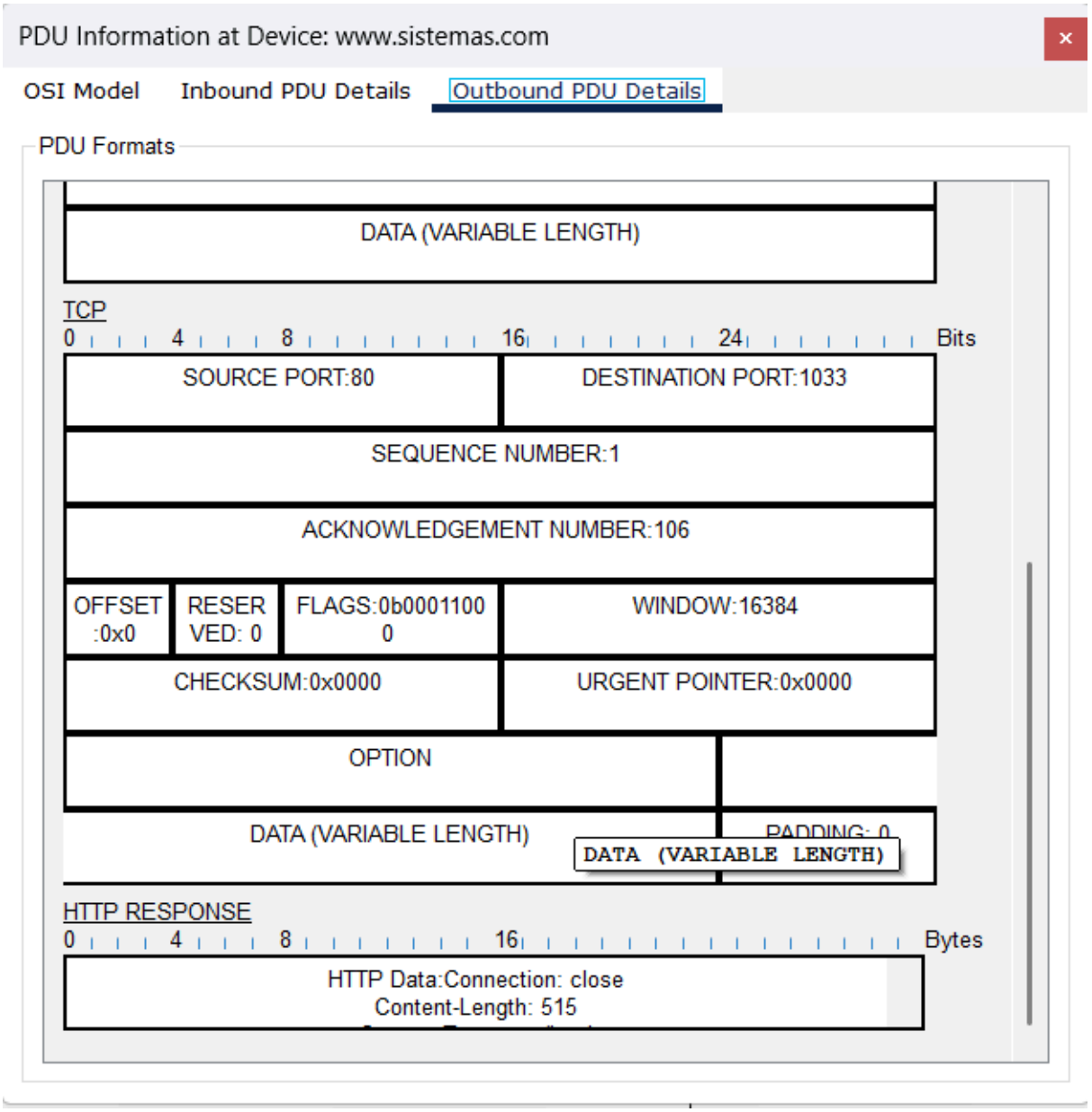


PDU Information at Device: DNS

OSI Model | Inbound PDU Details | Outbound PDU Details

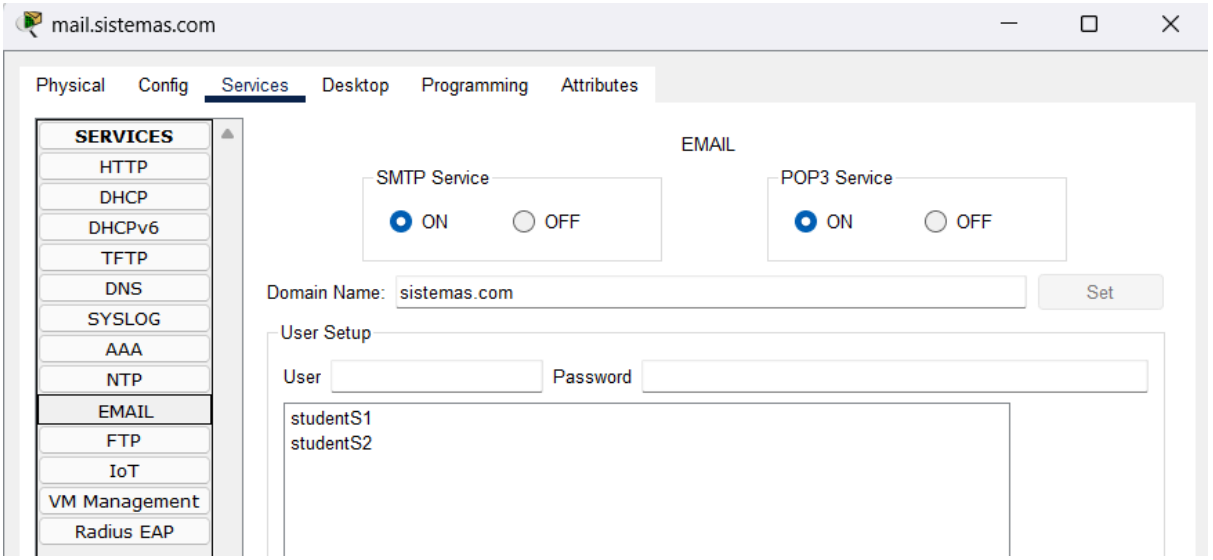
PDU Formats

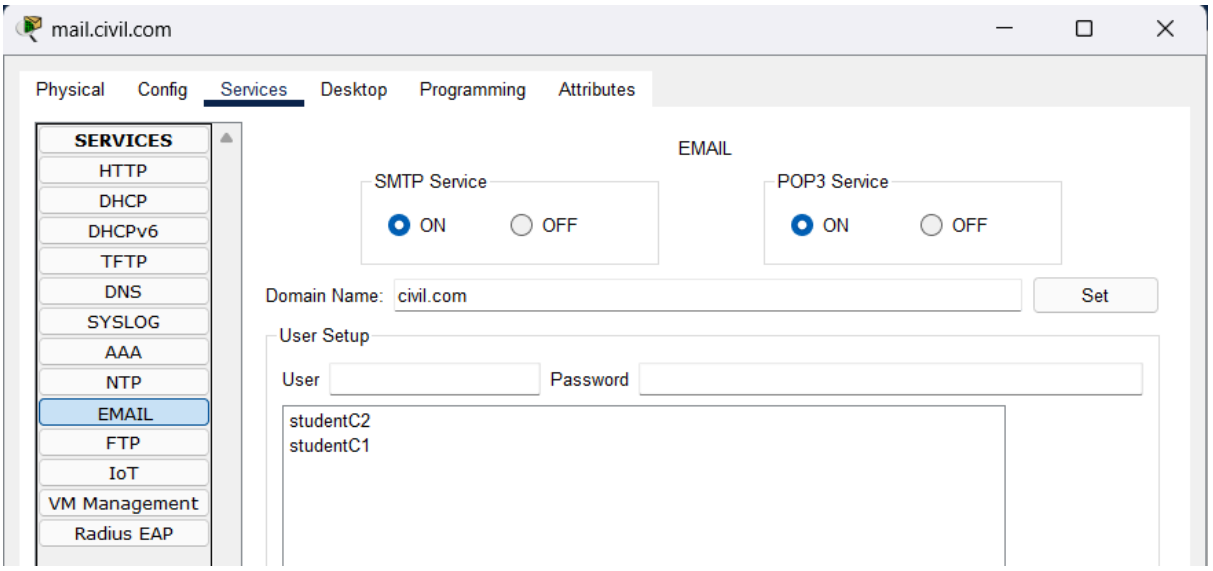


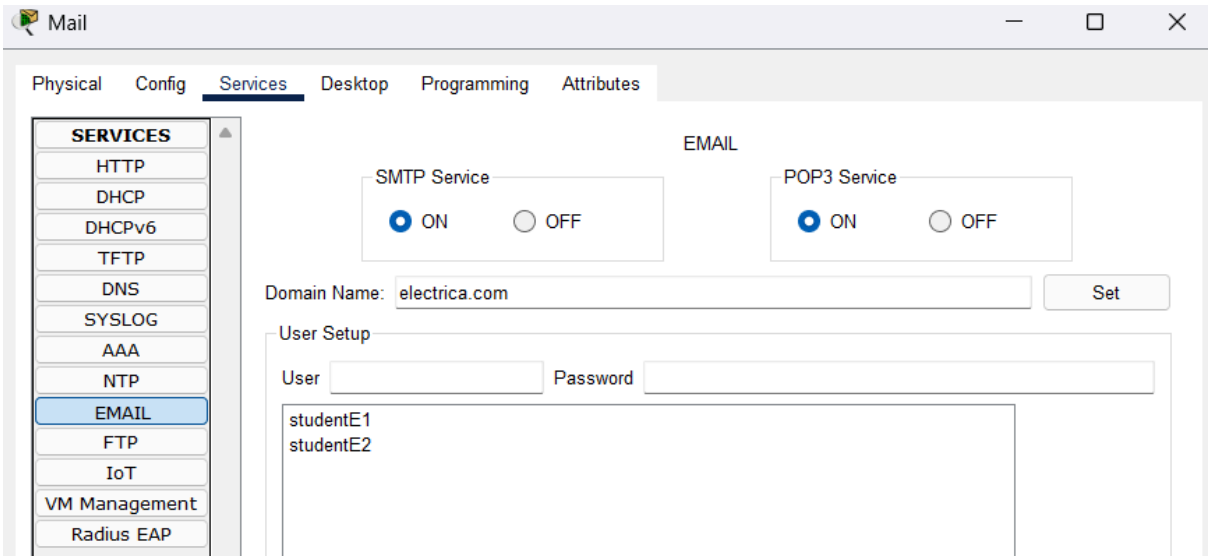


c. Email

On the mail server of each faculty, include email accounts for the users of each faculty. Use the client computer names as user names. Start the service.



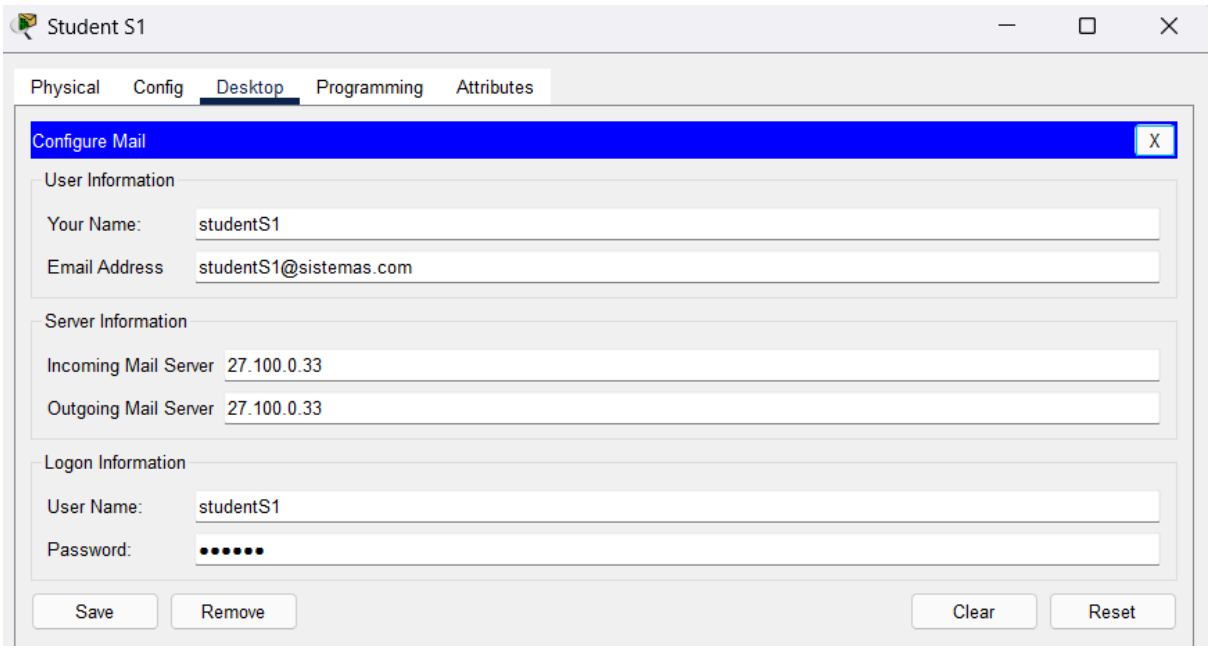


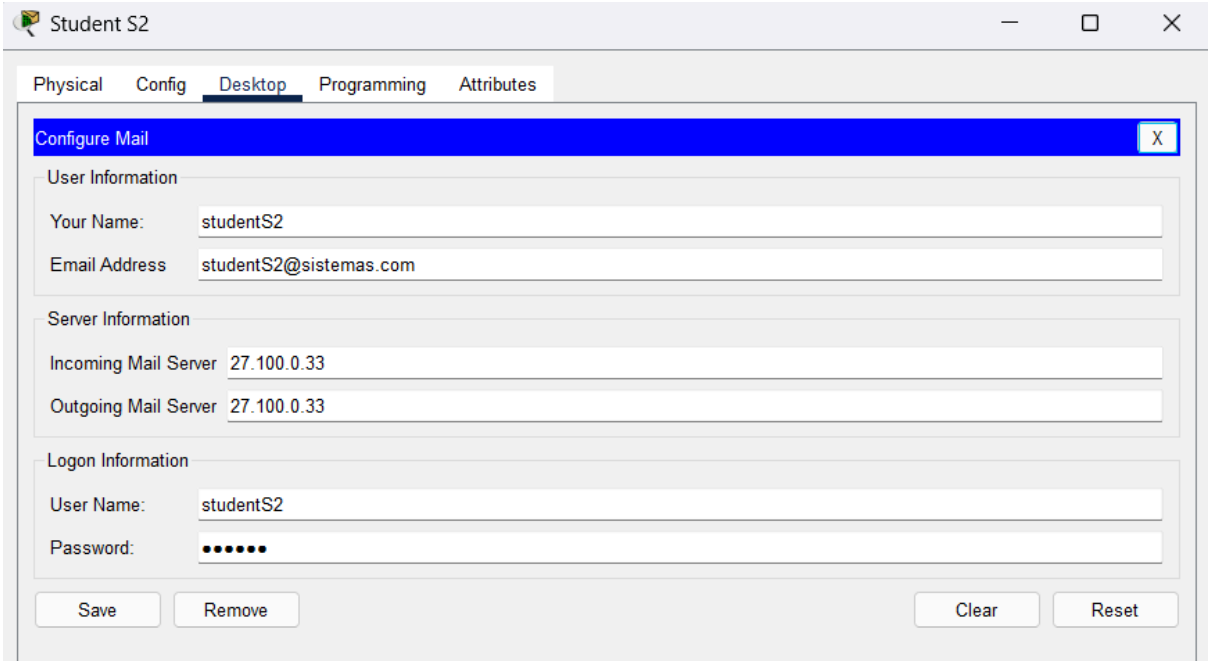


From the client stations, test the service:

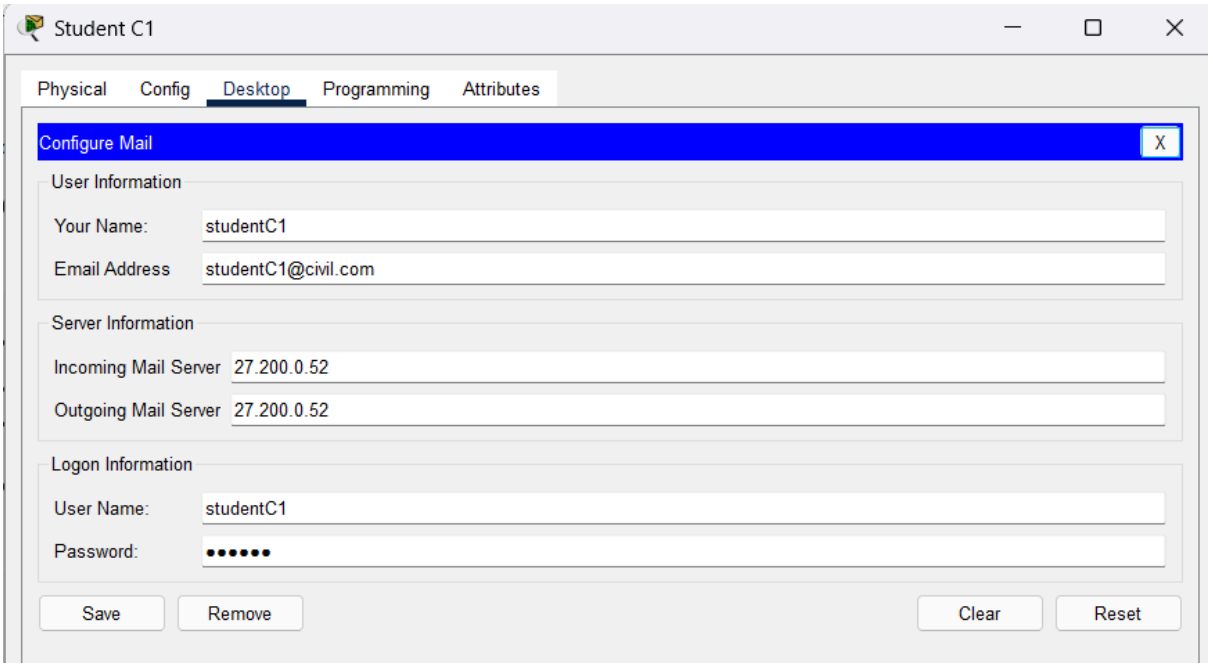
- Configure the email clients for each domain.

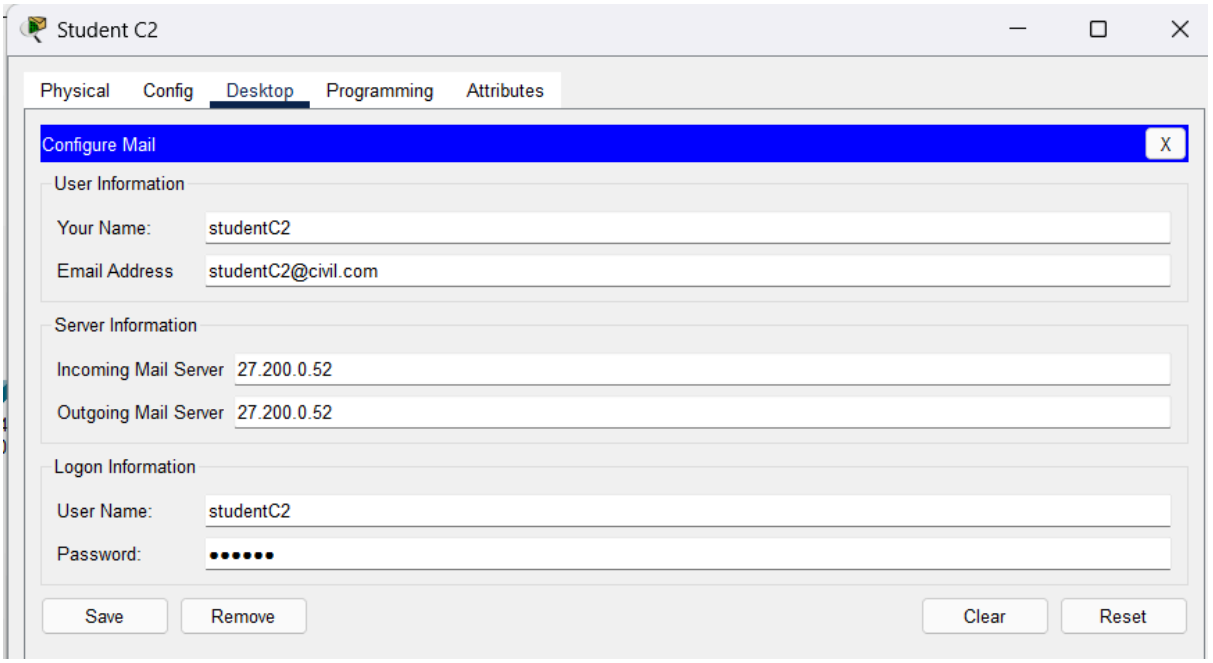
sistemas.com emails:



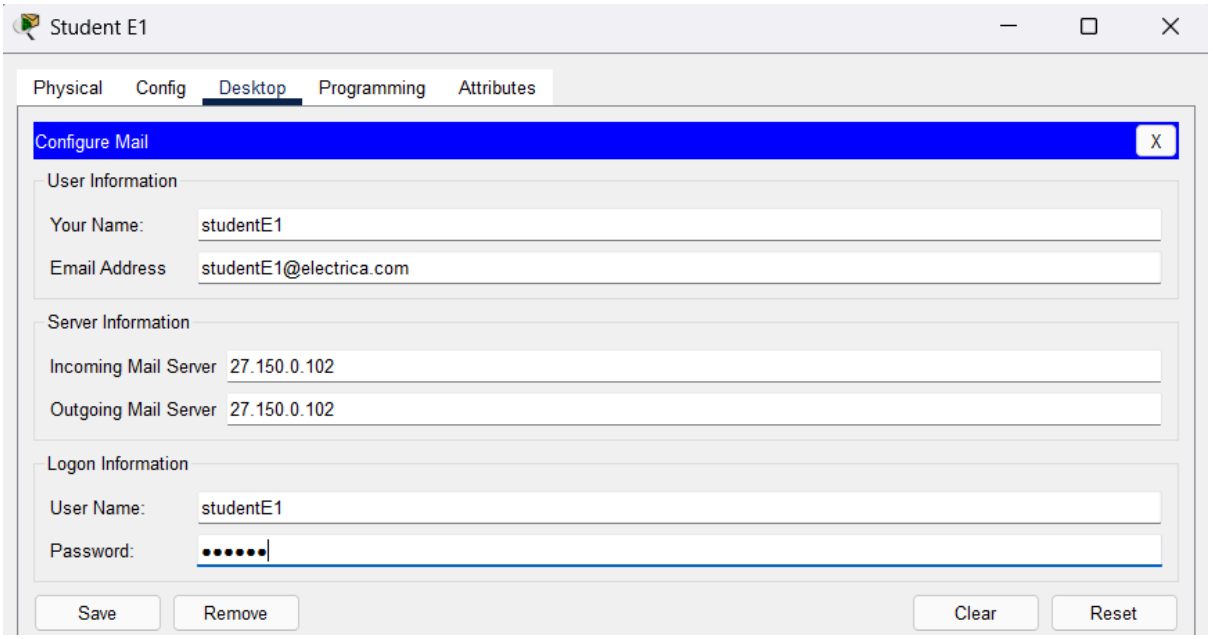


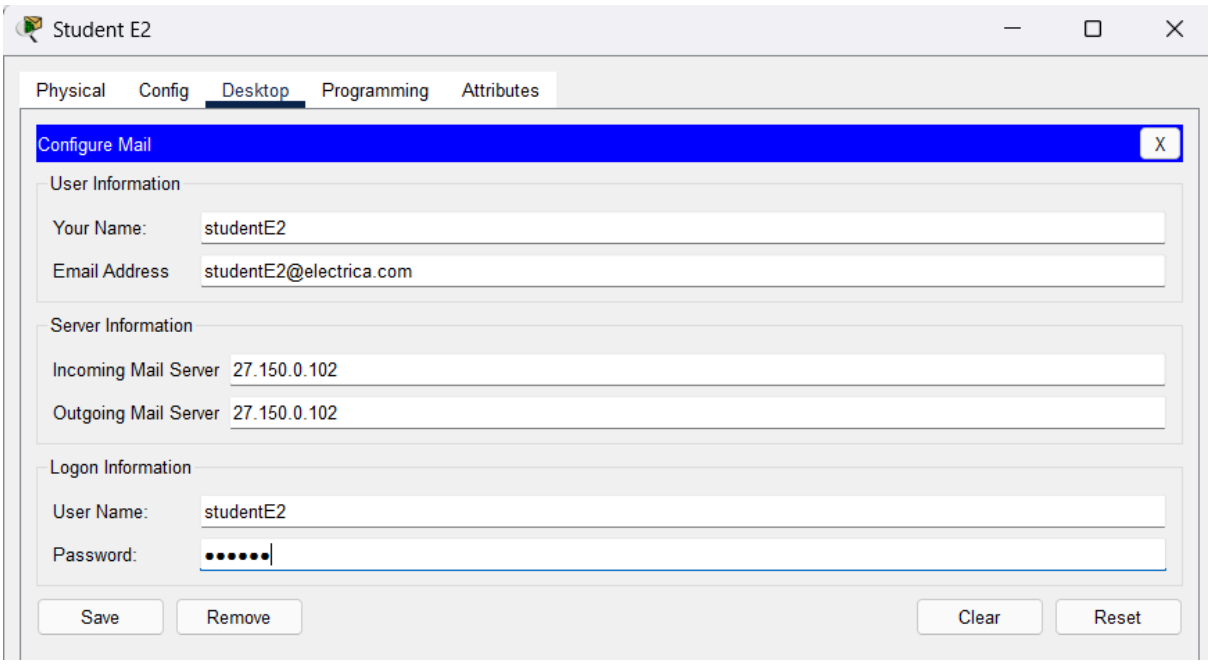
civil.com emails:



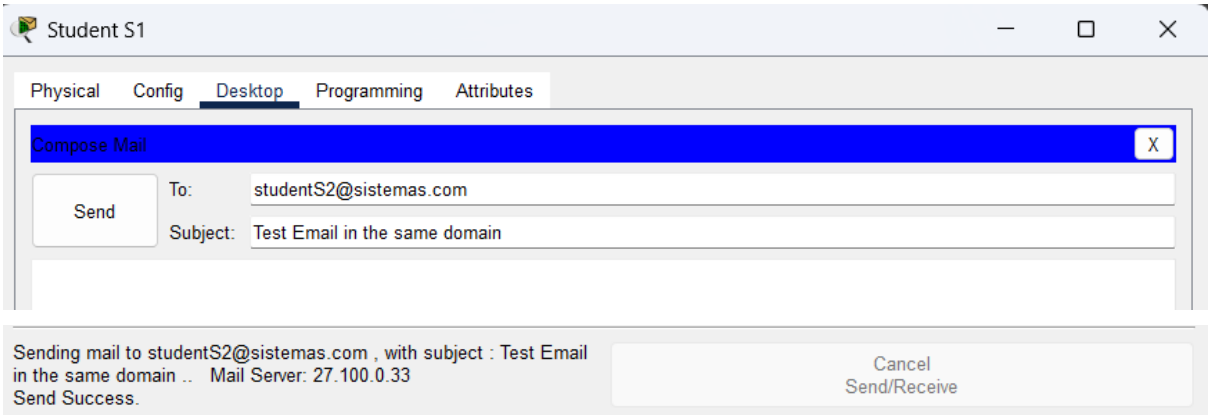


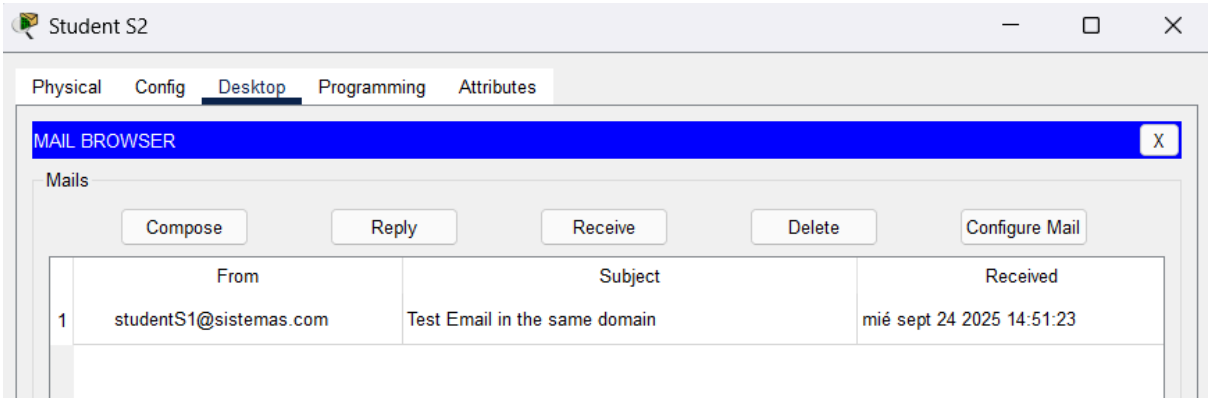
electrica.com emails:



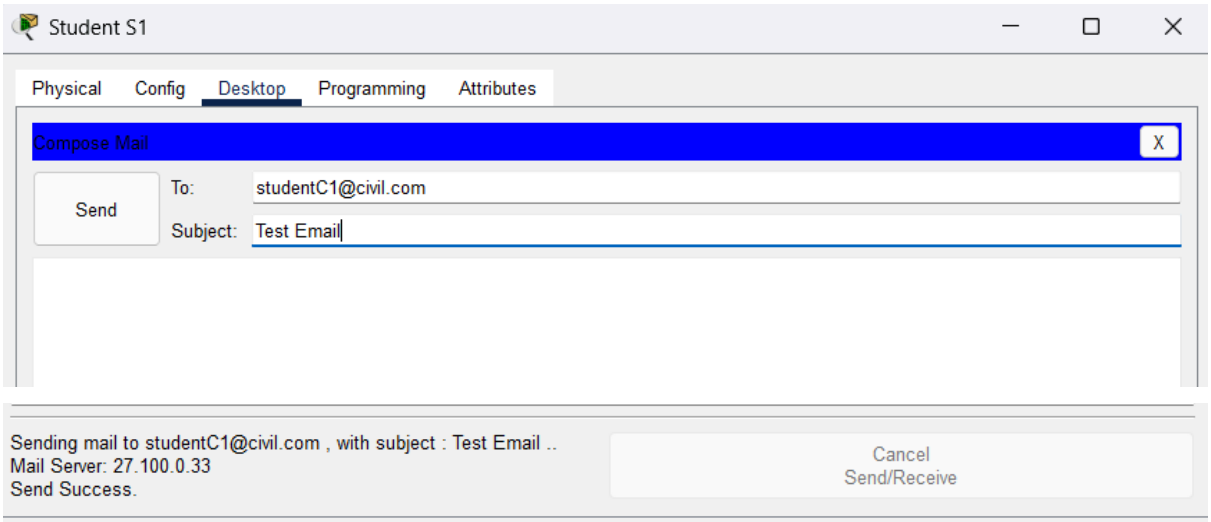


- Send emails between stations in the same domain.

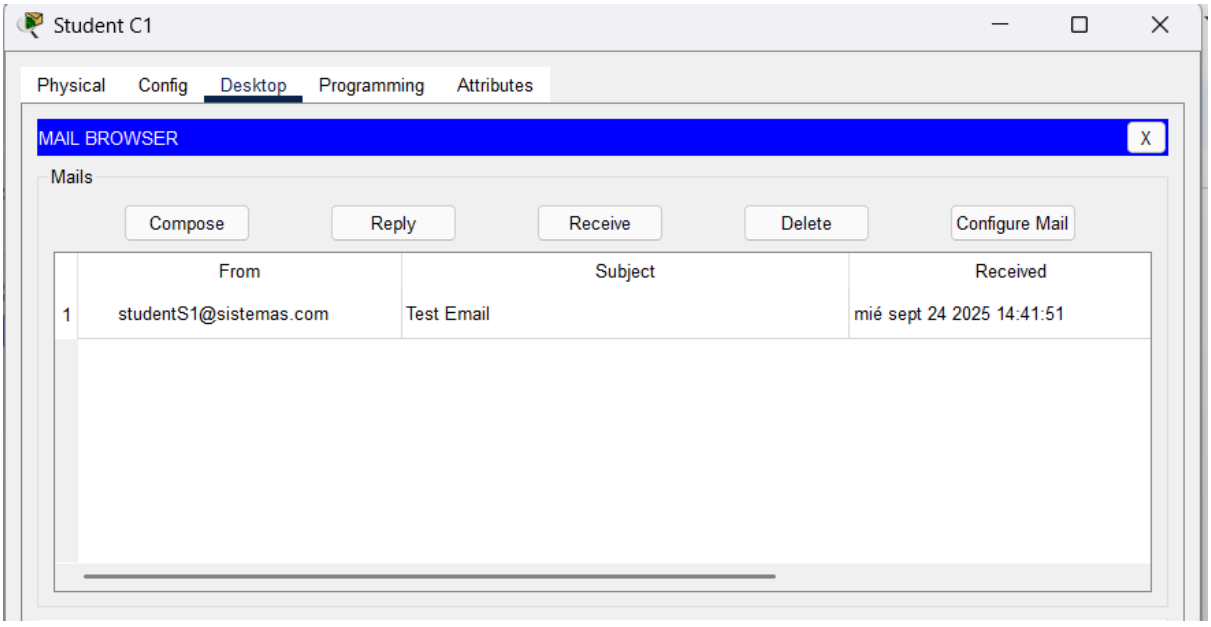


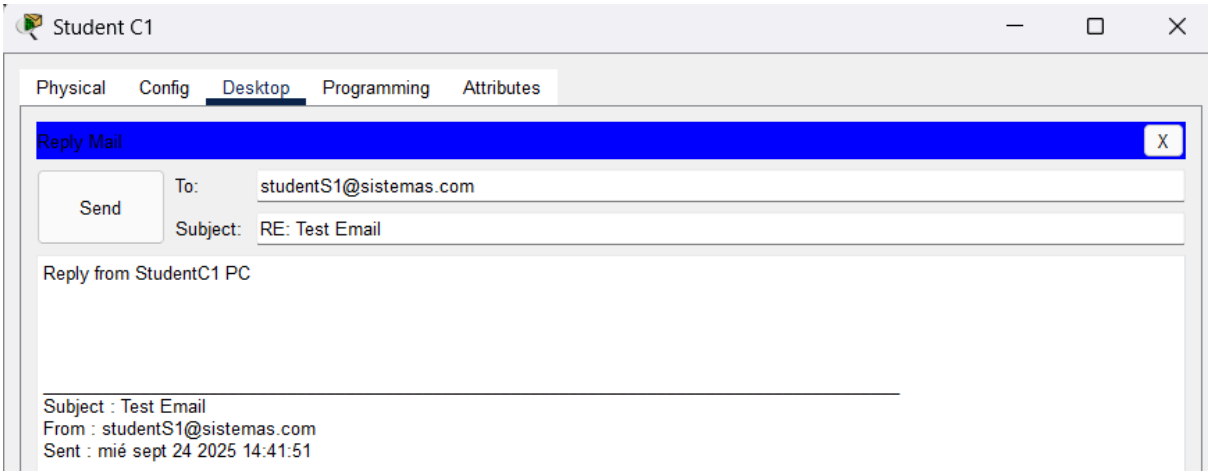


- Send emails to clients in other domains.
Student S1 PC was used for testing:

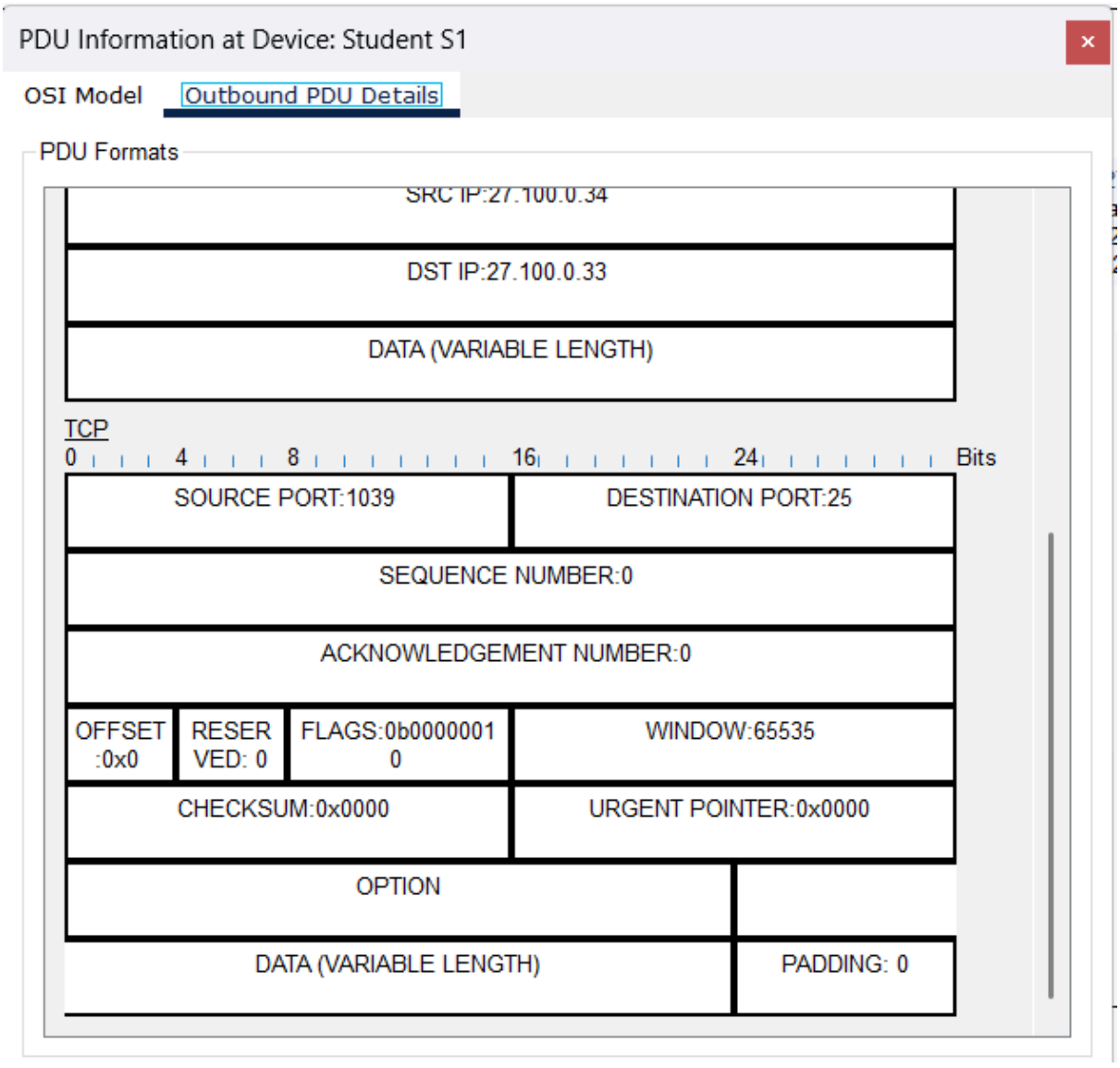


- Verify receipt of the emails on the stations and reply to the received messages.





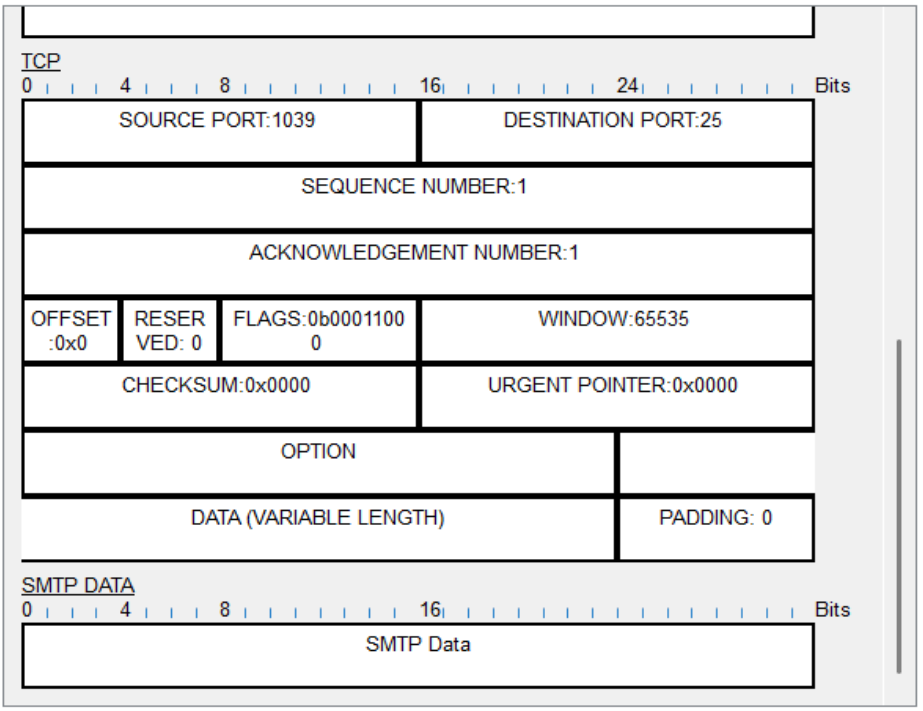
- Using the simulation tool, check the contents of the PDU at the transport and application layers during the email sending process between the sending client and its SMTP server and between the receiving client and its POP3 server.



PDU Information at Device: Switch0

OSI Model | Inbound PDU Details | Outbound PDU Details

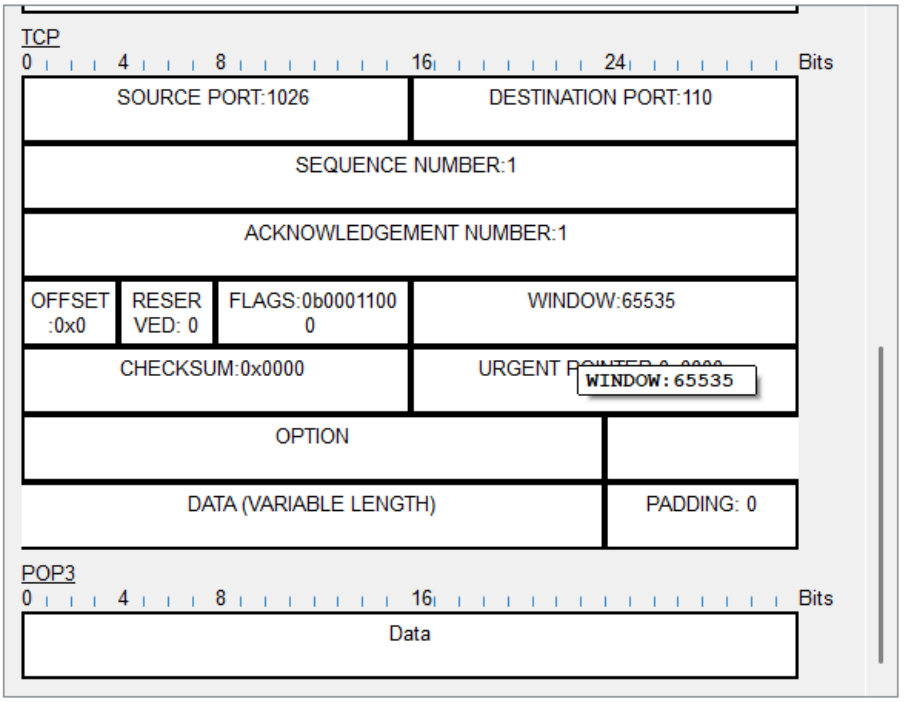
PDU Formats



PDU Information at Device: Student S2

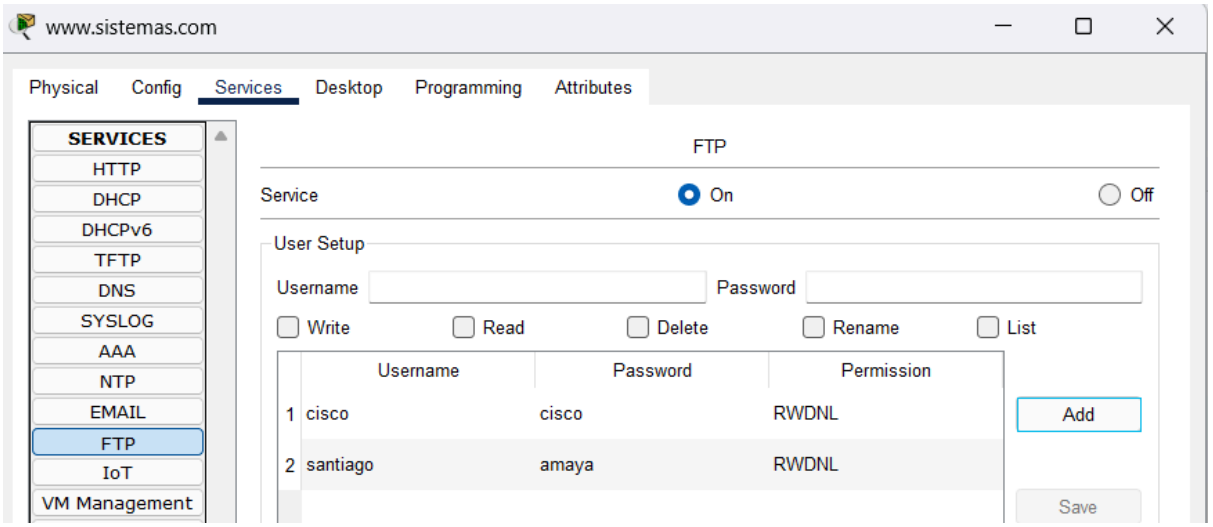
OSI Model | Outbound PDU Details

PDU Formats



d. FTP

On the sistemas web server, configure the FTP service. Create a user with your first name and password as your last name (e.g., in my case, it would be username: claudia and password: santiago). Start the service.



From the client stations, try connecting to the FTP server and download a file:

- From the command line, access the FTP server (by name or IP address) using the telnet command.

```
C:\>ftp 27.100.0.32
Trying to connect...27.100.0.32
Connected to 27.100.0.32
220- Welcome to PT Ftp server
```

- Log in with the created username/password.

```
Username:santiago
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>
```

- Download one of the files from the server.

```
ftp>dir

Listing /ftp directory from 27.100.0.32:
0   : asa842-k8.bin                                5571584
1   : asa923-k8.bin                                30468096
2   : c1841-advipservicesk9-mz.124-15.T1.bin       33591768
3   : c1841-ipbase-mz.123-14.T7.bin                13832032
4   : c1841-ipbasek9-mz.124-12.bin                 16599160
5   : c1900-universalk9-mz.SPA.155-3.M4a.bin       33591768
6   : c2600-advipservicesk9-mz.124-15.T1.bin       33591768
7   : c2600-i-mz.122-28.bin                        5571584
8   : c2600-ipbasek9-mz.124-8.bin                  13169700
9   : c2800nm-advipservicesk9-mz.124-15.T1.bin     50938004
10  : c2800nm-advipservicesk9-mz.151-4.M4.bin      33591768
11  : c2800nm-ipbase-mz.123-14.T7.bin              5571584
12  : c2800nm-ipbasek9-mz.124-8.bin                15522644
13  : c2900-universalk9-mz.SPA.155-3.M4a.bin       33591768
14  : c2950-i6q4l2-mz.121-22.EA4.bin               3058048
15  : c2950-i6q4l2-mz.121-22.EA8.bin               3117390
16  : c2960-lanbase-mz.122-25.FX.bin               4414921
```

```
ftp>get asa842-k8.bin

Reading file asa842-k8.bin from 27.100.0.32:
File transfer in progress...

[Transfer complete - 5571584 bytes]

5571584 bytes copied in 16.253 secs (78546 bytes/sec)
ftp>
```

- Exit the server and verify that the file is on the client.

```
ftp>quit

221- Service closing control connection.
C:\>
```

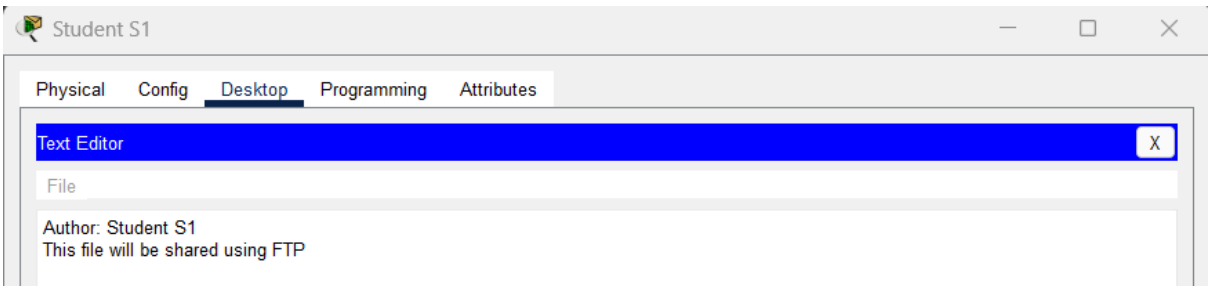
```
C:\>dir

 Volume in drive C has no label.
 Volume Serial Number is 5E12-4AF3
 Directory of C:\

12/31/1969  19:0 PM           5571584   asa842-k8.bin
12/31/1969  19:0 PM           26        sampleFile.txt
               5571610 bytes       2 File(s)
C:\>
```

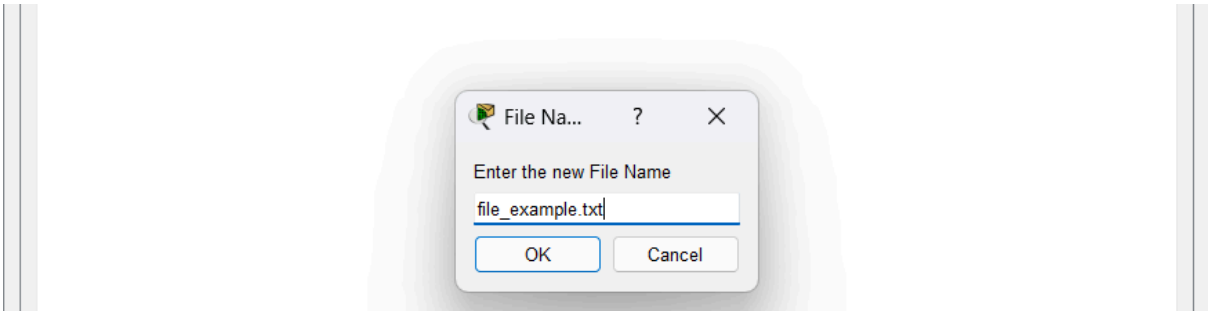
- Present the command log of the commands used.

Command	Use
ftp <FTP Server IP>	Connection to FTP server
get <File name>	Download file from server
dir	List files
quit	Exit connection to FTP server

From simulation mode, re-enter the FTP server and upload the .TXT file from the client. Check the headers at the application layer that indicate the connection, username and password submission, acceptance confirmation messages, file upload, and end of communication.

First, create a file to share





We enter through the console and upload the file using the put command

```
C:\>ftp 27.100.0.32
Trying to connect...27.100.0.32
Connected to 27.100.0.32
220- Welcome to PT Ftp server
Username:santiago
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>put file_example.txt

Writing file file_example.txt to 27.100.0.32:
File transfer in progress...

[Transfer complete - 53 bytes]

53 bytes copied in 0.086 secs (616 bytes/sec)
ftp>
```

We use the dir command to see if the file is on the server.

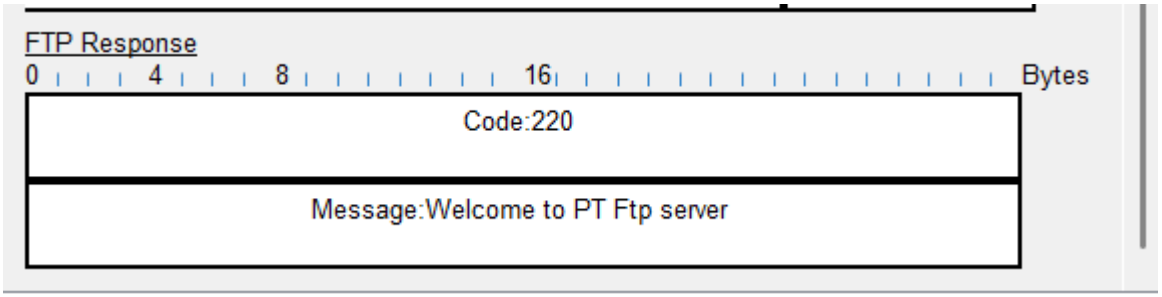
```
ftp>dir

Listing /ftp directory from 27.100.0.32:
0    : asa842-k8.bin                                5571584
1    : asa923-k8.bin                                30468096
2    : c1841-advipservicesk9-mz.124-15.T1.bin       33591768
3    : c1841-ipbase-mz.123-14.T7.bin                13832032
4    : c1841-ipbasek9-mz.124-12.bin                 16599160
5    : c1900-universalk9-mz.SPA.155-3.M4a.bin       33591768
6    : c2600-advipservicesk9-mz.124-15.T1.bin       33591768
7    : c2600-i-mz.122-28.bin                        5571584
8    : c2600-ipbasek9-mz.124-8.bin                  13169700
9    : c2800nm-advipservicesk9-mz.124-15.T1.bin     50938004
10   : c2800nm-advipservicesk9-mz.151-4.M4.bin      33591768
11   : c2800nm-ipbase-mz.123-14.T7.bin              5571584
12   : c2800nm-ipbasek9-mz.124-8.bin                15522644
13   : c2900-universalk9-mz.SPA.155-3.M4a.bin       33591768
14   : c2950-i6q4l2-mz.121-22.EA4.bin               3058048
15   : c2950-i6q4l2-mz.121-22.EA8.bin               3117390
16   : c2960-lanbase-mz.122-25.FX.bin               4414921
17   : c2960-lanbase-mz.122-25.SEE1.bin             4670455
18   : c2960-lanbasek9-mz.150-2.SE4.bin             4670455
19   : c3560-advipservicesk9-mz.122-37.SE1.bin      8662192
20   : c3560-advipservicesk9-mz.122-46.SE.bin       10713279
21   : c800-universalk9-mz.SPA.152-4.M4.bin         33591768
22   : c800-universalk9-mz.SPA.154-3.M6a.bin        83029236
23   : cat3k_caa-universalk9.16.03.02.SPA.bin       505532849
24   : cgr1000-universalk9-mz.SPA.154-2.CG          159487552
25   : cgr1000-universalk9-mz.SPA.156-3.CG          184530138
26   : file_example.txt                             53
27   : ir800-universalk9-bundle.SPA.156-3.M.bin     160968869
28   : ir800-universalk9-mz.SPA.155-3.M             61750062
29   : ir800-universalk9-mz.SPA.156-3.M             63753767
30   : ir800_yocto-1.7.2.tar                        2877440
31   : ir800_yocto-1.7.2_python-2.7.3.tar           6912000
32   : pt1000-i-mz.122-28.bin                       5571584
33   : pt3000-i6q4l2-mz.121-22.EA4.bin              3117390
ftp>
```

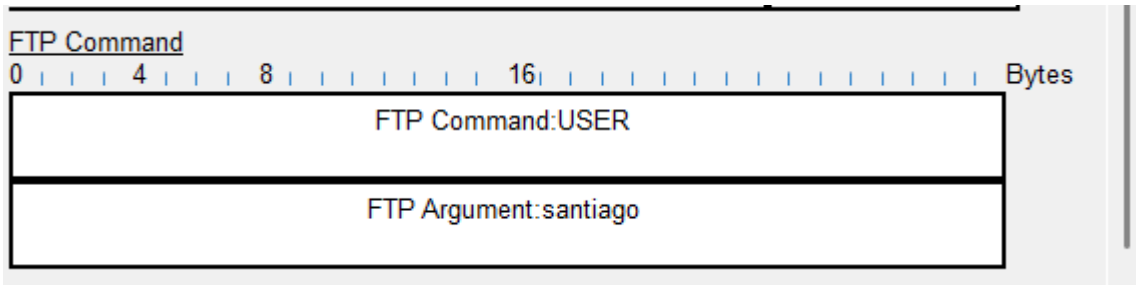
We see this headers:

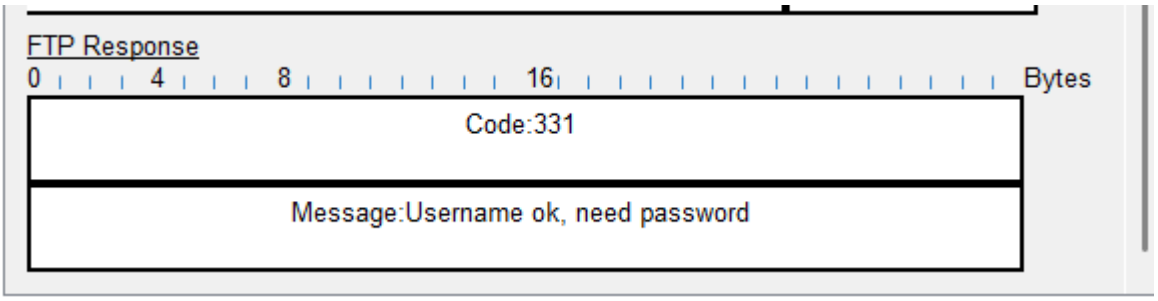
Simulation Panel

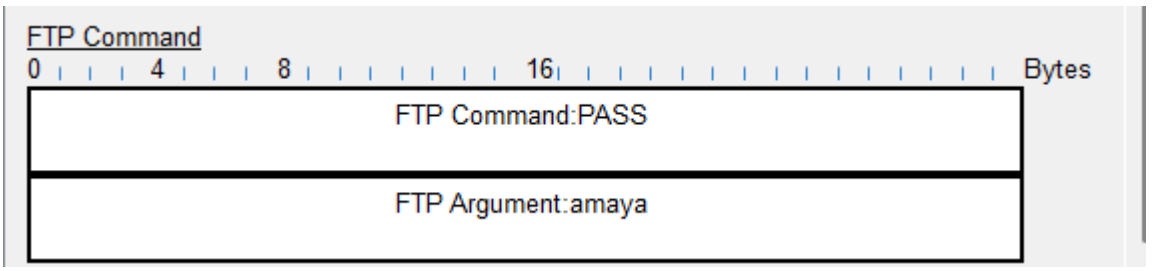
Event List

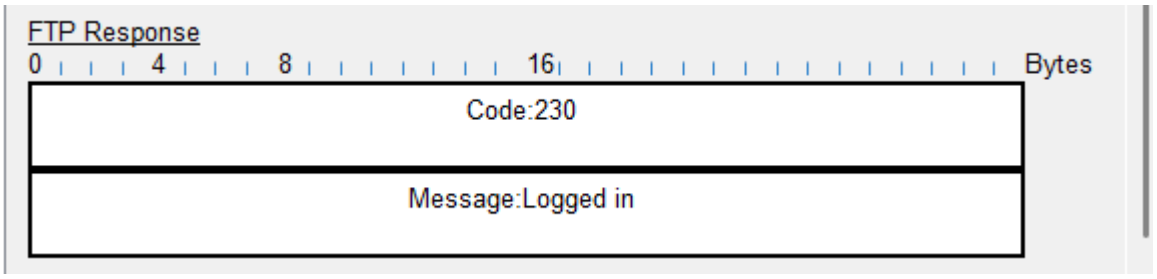
Vis.	Time(sec)	Last Device	At Device	Type
	0.126	Switch0	www.sistemas.com	FTP
	0.126	--	www.sistemas.com	FTP
	0.127	www.sistemas.com	Switch0	FTP
	0.128	Switch0	Student S1	FTP
	0.128	--	Student S1	FTP
	0.129	Student S1	Switch0	FTP
	0.130	Switch0	www.sistemas.com	FTP
	0.130	--	www.sistemas.com	FTP
	0.131	www.sistemas.com	Switch0	FTP
	0.132	Switch0	Student S1	FTP
	0.132	--	Student S1	FTP
	0.133	Student S1	Switch0	FTP
	0.134	Switch0	www.sistemas.com	FTP
	0.134	--	www.sistemas.com	FTP
	0.135	www.sistemas.com	Switch0	FTP
	0.136	Switch0	Student S1	FTP

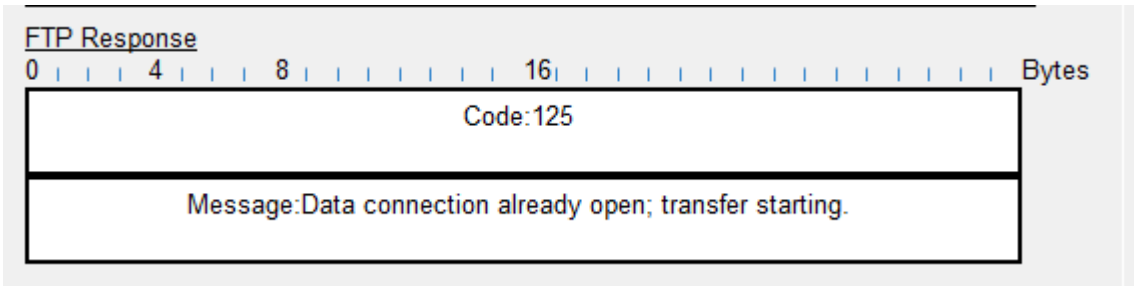


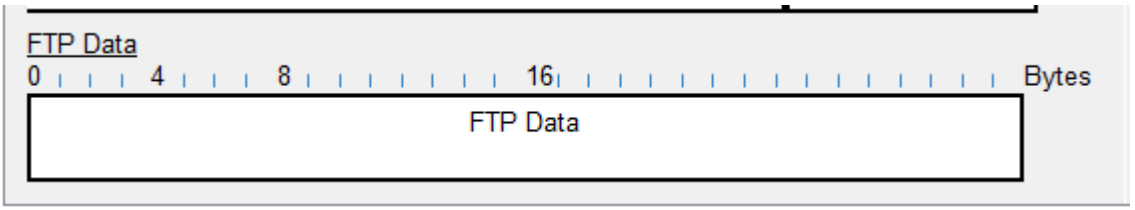


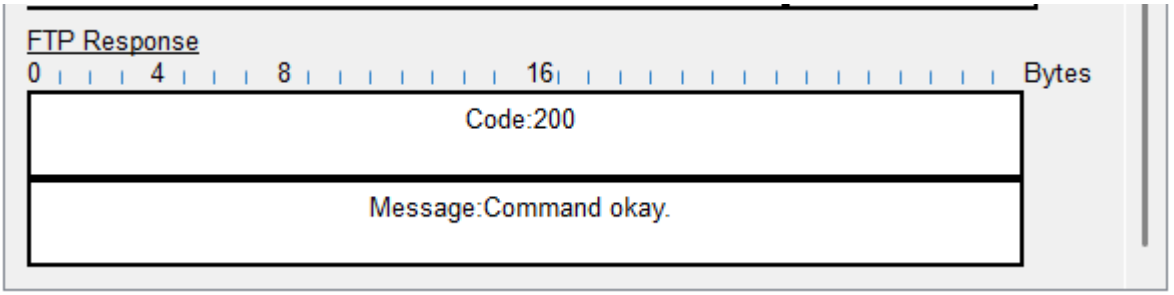


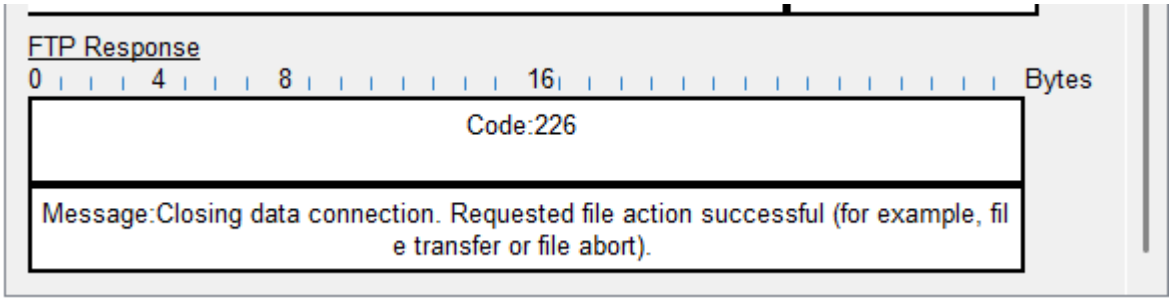

FTP Command
0
4
8
16
Bytes
FTP Command:PASS
FTP Argument:amaya

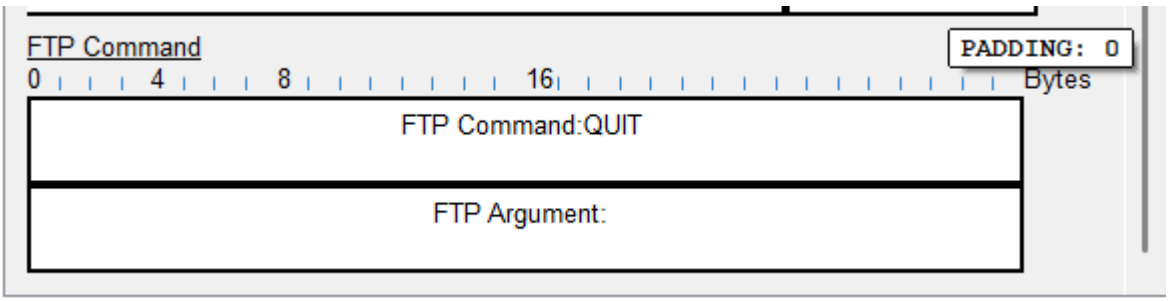

FTP Response
0
4
8
16
Bytes
Code:230
Message:Logged in

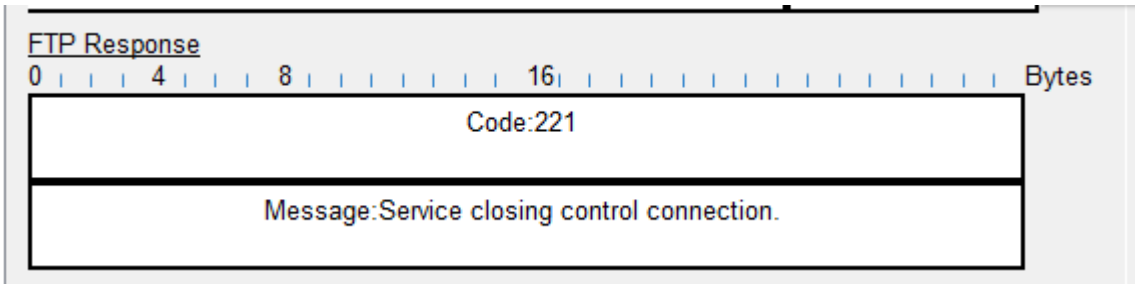

FTP Response
0
4
8
16
Bytes
Code:125
Message:Data connection already open; transfer starting.


FTP Data
0
4
8
16
Bytes
FTP Data


FTP Response
0
4
8
16
Bytes
Code:200
Message:Command okay.


FTP Response
0
4
8
16
Bytes
Code:226
Message:Closing data connection. Requested file action successful (for example, fil
e transfer or file abort).





In the Real Network

---

1. Wireshark

Using the Wireshark tool, perform and document the following tests:

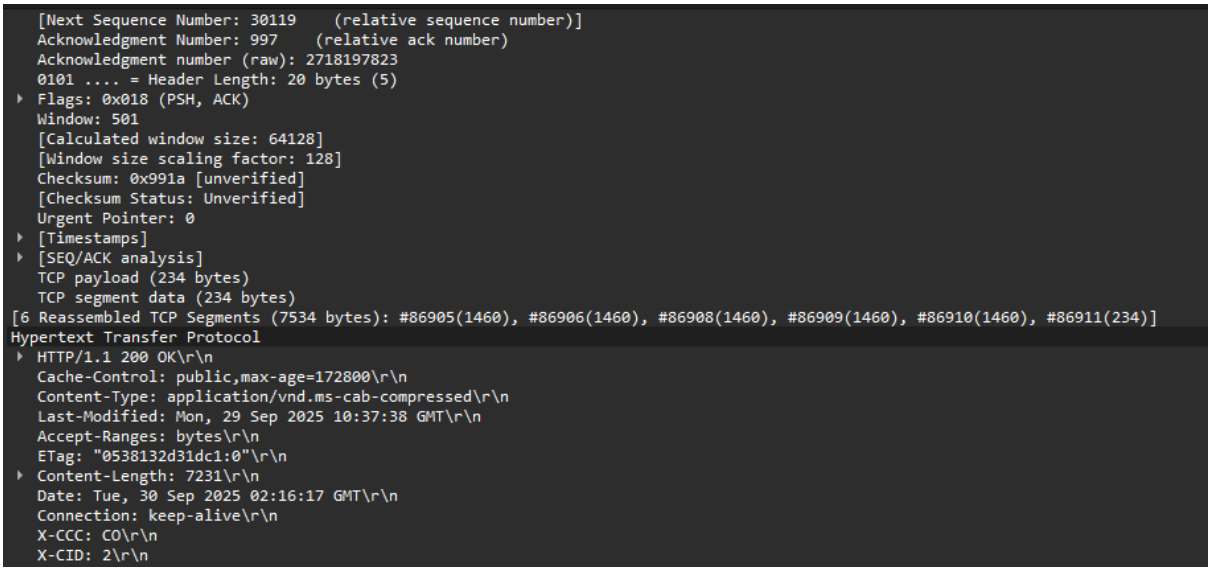
a. Make a web query to the Computer Lab website and see which application layer protocols are active. Analyze the application layer information and transport layer ports in the content of the captured packets.

Here´s the TCP protocol when making the connection, showing the ACK, SYN, FIN, PSH flags, and the sequence and acknowledgement numbers. To obtain index.html, when you perform a GET, you can view it through the HTTP protocol and see how the server returns a 200 (ok) response code

No.	Time	Source	Destination	Protocol	Length	Info
86881	237.446413	181.54.161.17	192.168.1.2	TCP	1514	80 → 52401 [ACK] Seq=5841 Ack=250 Win=64128 Len=1460 [TCP PDU reassembled in 86882]
86882	237.446413	181.54.161.17	192.168.1.2	HTTP	280	HTTP/1.1 200 OK (application/vnd.ms-cab-compressed)
86883	237.446430	192.168.1.2	181.54.161.17	TCP	54	52401 → 80 [ACK] Seq=250 Ack=7527 Win=65280 Len=0
86884	237.447082	192.168.1.2	181.54.161.17	HTTP	303	GET /c/msdownload/update/others/2025/09/43960788_4dcd71113997110a18242881d9dea6a8e2ef7294.cab HTTP/1.1
86885	237.451361	181.54.161.17	192.168.1.2	TCP	60	80 → 52401 [ACK] Seq=7527 Ack=499 Win=64128 Len=0
86886	237.453220	181.54.161.17	192.168.1.2	TCP	1514	80 → 52401 [ACK] Seq=7527 Ack=499 Win=64128 Len=1460 [TCP PDU reassembled in 86892]
86887	237.453220	181.54.161.17	192.168.1.2	TCP	1514	80 → 52401 [ACK] Seq=8987 Ack=499 Win=64128 Len=1460 [TCP PDU reassembled in 86892]
86888	237.453238	192.168.1.2	181.54.161.17	TCP	54	52401 → 80 [ACK] Seq=499 Ack=10447 Win=65280 Len=0
86889	237.453443	181.54.161.17	192.168.1.2	TCP	1514	80 → 52401 [ACK] Seq=10447 Ack=499 Win=64128 Len=1460 [TCP PDU reassembled in 86892]
86890	237.453443	181.54.161.17	192.168.1.2	TCP	1514	80 → 52401 [ACK] Seq=11907 Ack=499 Win=64128 Len=1460 [TCP PDU reassembled in 86892]
86891	237.453443	181.54.161.17	192.168.1.2	TCP	1514	80 → 52401 [ACK] Seq=13367 Ack=499 Win=64128 Len=1460 [TCP PDU reassembled in 86892]
86892	237.453443	181.54.161.17	192.168.1.2	HTTP	280	HTTP/1.1 200 OK (application/vnd.ms-cab-compressed)
86893	237.453465	192.168.1.2	181.54.161.17	TCP	54	52401 → 80 [ACK] Seq=499 Ack=15053 Win=65280 Len=0
86894	237.453925	192.168.1.2	181.54.161.17	HTTP	303	GET /c/msdownload/update/others/2025/09/43960787_ffbd236d873ebd515a0c6cda3f6476a46189aeaa.cab HTTP/1.1
86895	237.458423	181.54.161.17	192.168.1.2	TCP	60	80 → 52401 [ACK] Seq=15053 Ack=748 Win=64128 Len=0
86896	237.461980	181.54.161.17	192.168.1.2	TCP	1514	80 → 52401 [ACK] Seq=15053 Ack=748 Win=64128 Len=1460 [TCP PDU reassembled in 86902]
86897	237.461980	181.54.161.17	192.168.1.2	TCP	1514	80 → 52401 [ACK] Seq=16513 Ack=748 Win=64128 Len=1460 [TCP PDU reassembled in 86902]
86898	237.461997	192.168.1.2	181.54.161.17	TCP	54	52401 → 80 [ACK] Seq=748 Ack=17973 Win=65280 Len=0
86899	237.462203	181.54.161.17	192.168.1.2	TCP	1514	80 → 52401 [ACK] Seq=17973 Ack=748 Win=64128 Len=1460 [TCP PDU reassembled in 86902]
86900	237.462203	181.54.161.17	192.168.1.2	TCP	1514	80 → 52401 [ACK] Seq=19433 Ack=748 Win=64128 Len=1460 [TCP PDU reassembled in 86902]
86901	237.462203	181.54.161.17	192.168.1.2	TCP	1514	80 → 52401 [ACK] Seq=20893 Ack=748 Win=64128 Len=1460 [TCP PDU reassembled in 86902]
86902	237.462203	181.54.161.17	192.168.1.2	HTTP	286	HTTP/1.1 200 OK (application/vnd.ms-cab-compressed)
86903	237.462219	192.168.1.2	181.54.161.17	TCP	54	52401 → 80 [ACK] Seq=748 Ack=22585 Win=65280 Len=0
86904	237.462711	192.168.1.2	181.54.161.17	HTTP	303	GET /c/msdownload/update/others/2025/09/43960786_3e69ebc6244b7561dbbf42c2ca339b9f74fc80e7.cab HTTP/1.1
86905	237.468815	181.54.161.17	192.168.1.2	TCP	1514	80 → 52401 [ACK] Seq=22585 Ack=997 Win=64128 Len=1460 [TCP PDU reassembled in 86911]
86906	237.468815	181.54.161.17	192.168.1.2	TCP	1514	80 → 52401 [ACK] Seq=24045 Ack=997 Win=64128 Len=1460 [TCP PDU reassembled in 86911]
86907	237.468833	192.168.1.2	181.54.161.17	TCP	54	52401 → 80 [ACK] Seq=997 Ack=25505 Win=65280 Len=0
86908	237.469047	181.54.161.17	192.168.1.2	TCP	1514	80 → 52401 [ACK] Seq=25505 Ack=997 Win=64128 Len=1460 [TCP PDU reassembled in 86911]

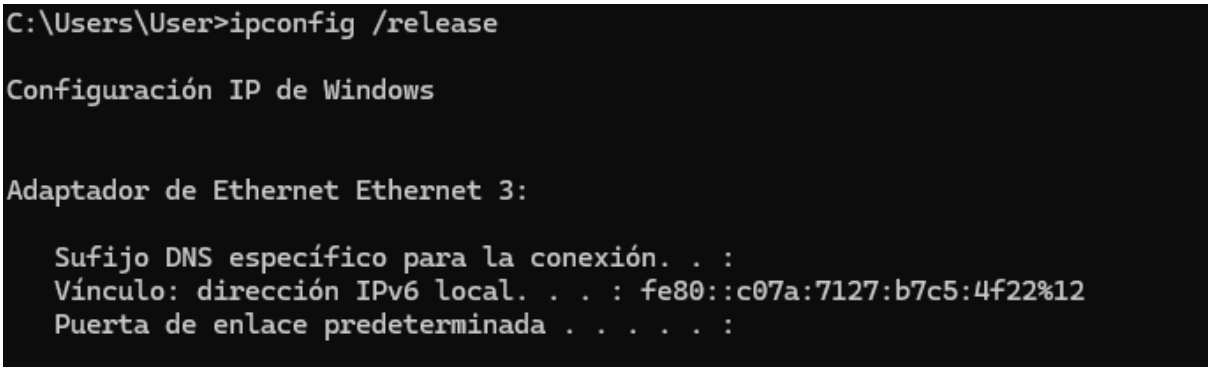
We also saw what the TCP segments looked like and how the transport addresses were assigned, between the receiving windows and the sequence numbers for communication.

We can also see the HTTP messages corresponding to the GET of the lab page.



b. Capture the DHCP traffic from your computer and analyze the packets that circulate between the client and the servers offering addresses. Check the content of the packets at the application layer and transport layer ports. Note: To perform this activity, capture the traffic, remove the IP address from the machine (type ipconfig /release in the command line) and request it again (type ipconfig /renew in the command line).

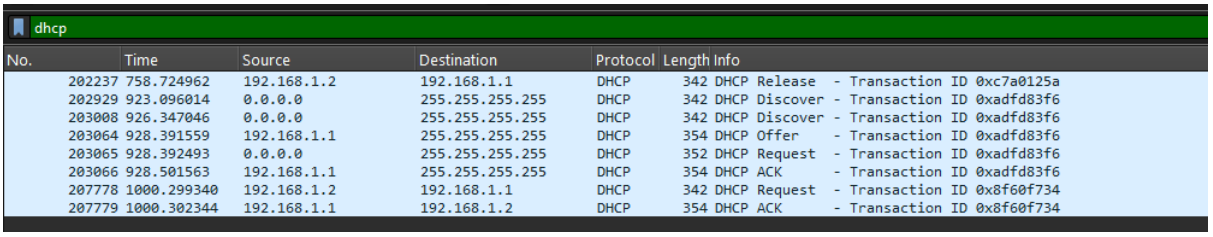
First we have to release the IP that we have configured with the command: ipconfig /release



Then we see how the new IP address is provided by capturing the packets with Wireshark using the command: ipconfig /renew

```
C:\Users\User>ipconfig /renew

Configuración IP de Windows


Adaptador de Ethernet Ethernet 3:

   Sufijo DNS específico para la conexión. . :
   Vínculo: dirección IPv6 local. . . : fe80::c07a:7127:b7c5:4f22%12
   Dirección IPv4. . . . . . . . . . . . . . : 192.168.1.2
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 192.168.1.1

C:\Users\User>
```

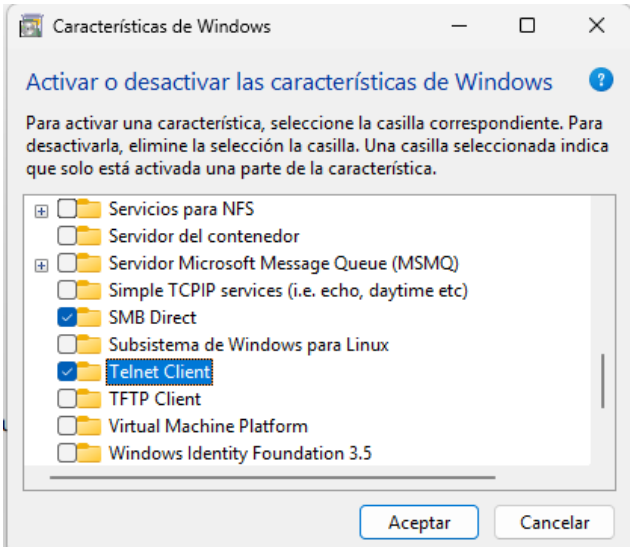
The captured traffic is presented below. In it, we can see the DORA process used by the DHCP protocol to assign the IP address.



dhcp

No.	Time	Source	Destination	Protocol	Length	Info
202237	758.724962	192.168.1.2	192.168.1.1	DHCP	342	DHCP Release - Transaction ID 0xc7a0125a
202929	923.096014	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xadfd83f6
203008	926.347046	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xadfd83f6
203064	928.391559	192.168.1.1	255.255.255.255	DHCP	354	DHCP Offer - Transaction ID 0xadfd83f6
203065	928.392493	0.0.0.0	255.255.255.255	DHCP	352	DHCP Request - Transaction ID 0xadfd83f6
203066	928.501563	192.168.1.1	255.255.255.255	DHCP	354	DHCP ACK - Transaction ID 0xadfd83f6
207778	1000.299340	192.168.1.2	192.168.1.1	DHCP	342	DHCP Request - Transaction ID 0x8f60f734
207779	1000.302344	192.168.1.1	192.168.1.2	DHCP	354	DHCP ACK - Transaction ID 0x8f60f734

c. Analyze the application layer information and ports (transport layer) in the content of the captured packets in an HTTP connection:
- Unlock the use of the TELNET protocol on your computer.
For this we have to access features of Windows and we activate Telnet Client



- Capture the packets when using TELNET and HTTP protocols and show the application layer messages generated in the following queries:

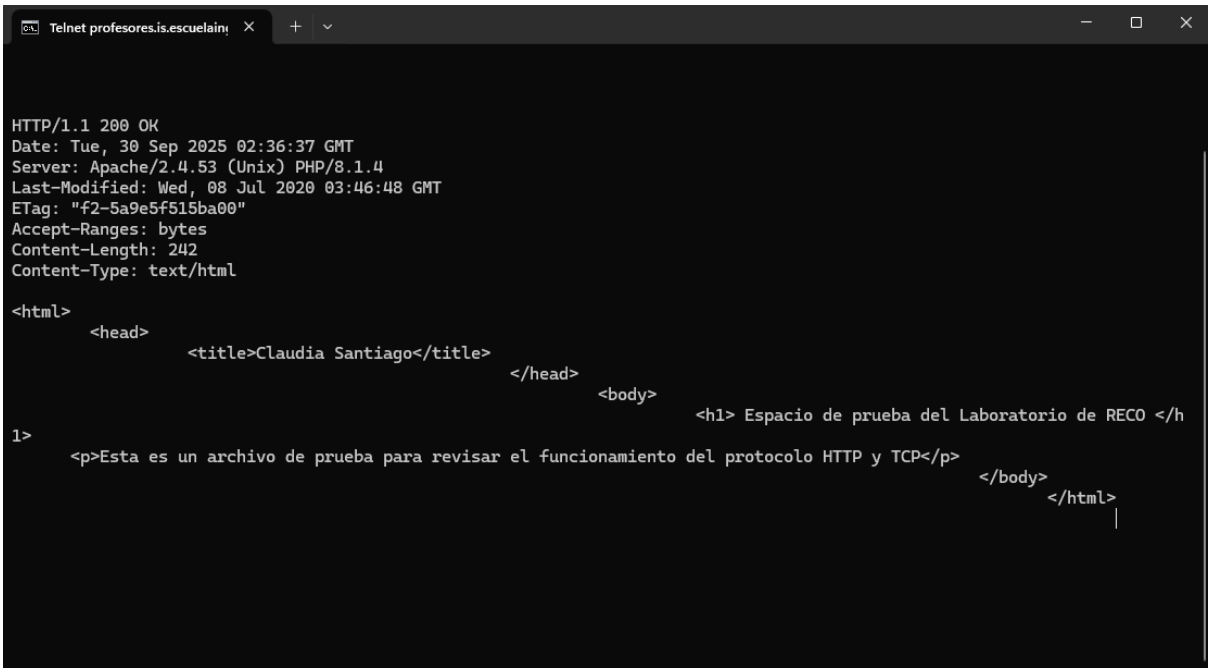
- Capture the following web page http://profesores.is.escuelaing.edu.co/csantiago/RECO/index.html using the protocols:

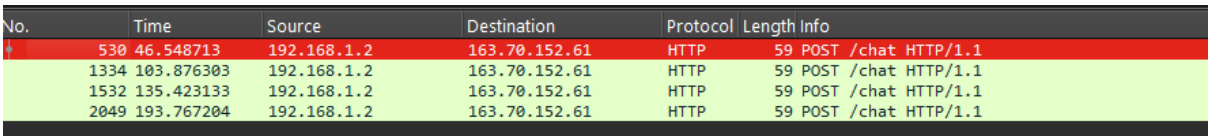
- Telnet

* telnet profesores.is.escuelaing.edu.co 80

* GET path/file (e.g., GET /index.html).

The content in index.html is displayed with a “Espacio de prueba del laboratorio de RECO” message! And in addition, how do the TCP and HTTP headers look when the GET request is made?

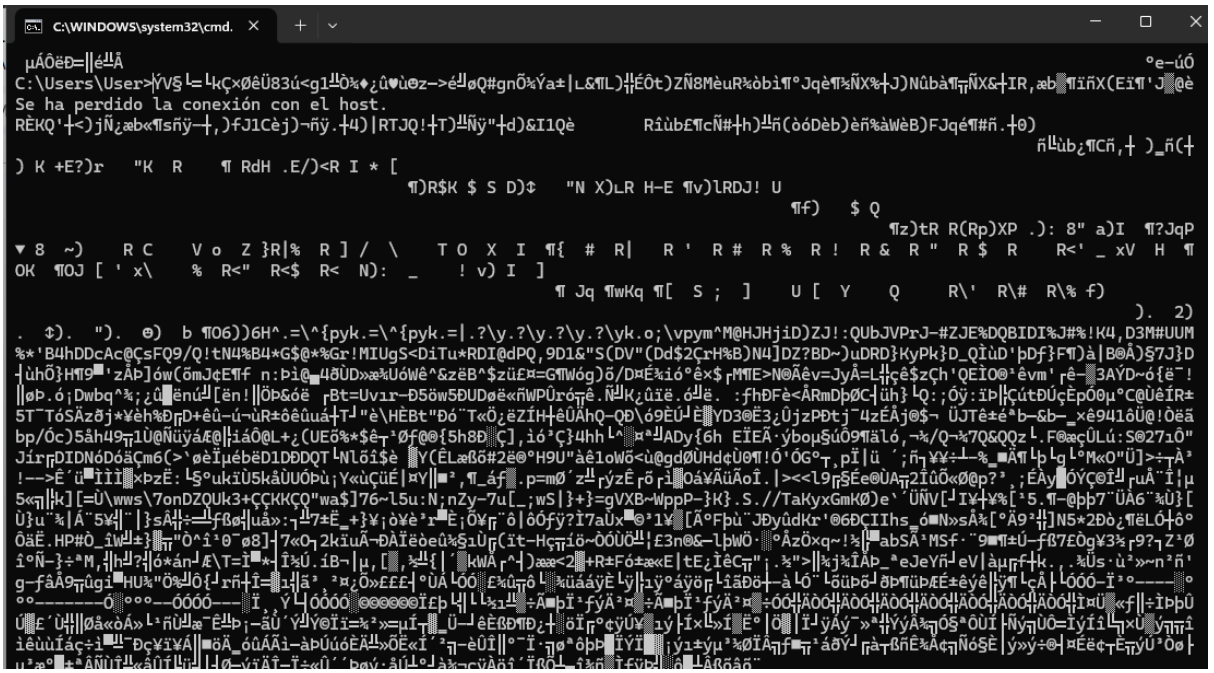




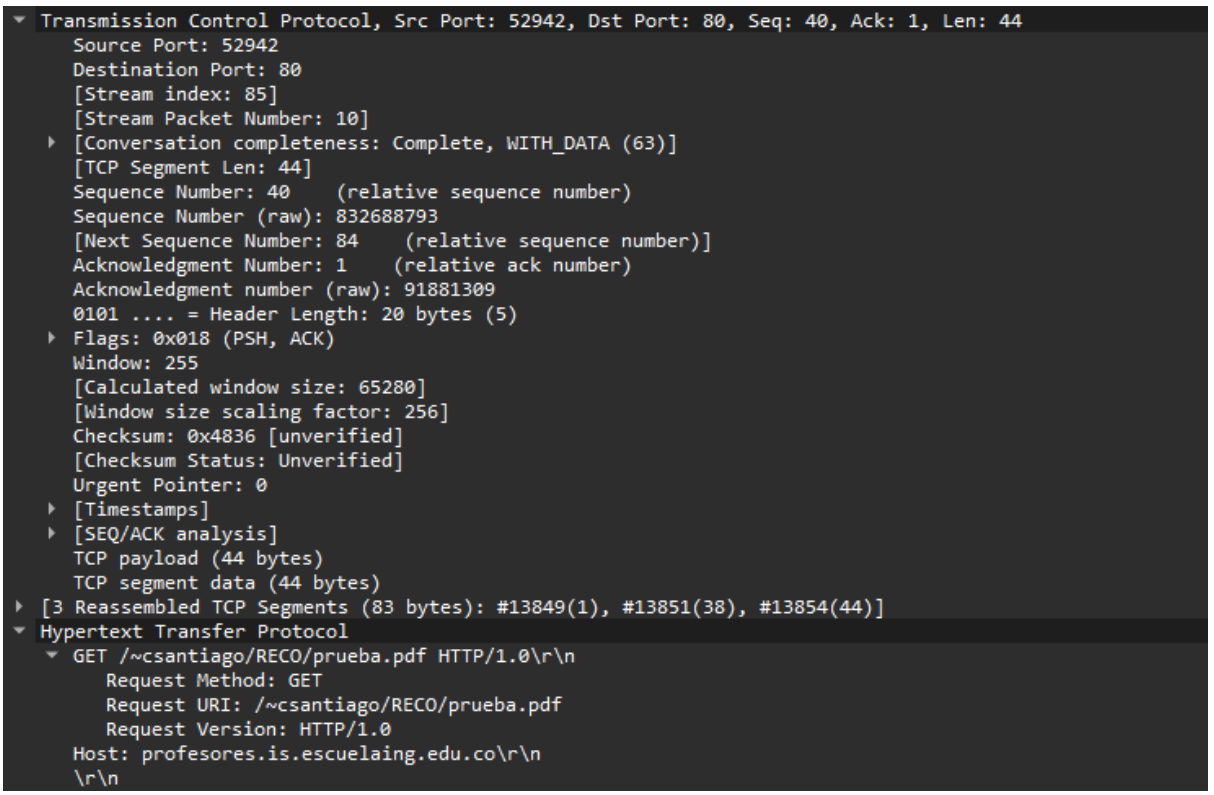
```
   [Stream index: 39]
   [Stream Packet Number: 32]
 ▸ [Conversation completeness: Complete, WITH_DATA (31)]
   [TCP Segment Len: 5]
   Sequence Number: 845    (relative sequence number)
   Sequence Number (raw): 1477079674
   [Next Sequence Number: 850    (relative sequence number)]
   Acknowledgment Number: 1266    (relative ack number)
   Acknowledgment number (raw): 3994696912
   0101 .... = Header Length: 20 bytes (5)
 ▸ Flags: 0x018 (PSH, ACK)
   Window: 251
   [Calculated window size: 64256]
   [Window size scaling factor: 256]
   Checksum: 0xfd4d [unverified]
   [Checksum Status: Unverified]
   Urgent Pointer: 0
 ▸ [Timestamps]
 ▸ [SEQ/ACK analysis]
   TCP payload (5 bytes)
   TCP segment data (5 bytes)
[6 Reassembled TCP Segments (849 bytes): #1800(442), #1806(57), #1811(41), #1812(137), #1818(167), #2049(5)]
Hypertext Transfer Protocol, has 12 chunks (including last chunk)
 ▸ POST /chat HTTP/1.1\r\n
   Host: c.whatsapp.net\r\n
   User-Agent: Mozilla/5.0 (compatible; WAChat/1.2; +http://www.whatsapp.com/contact)\r\n
   Transfer-Encoding: chunked\r\n
   \r\n
   [Full request URI: http://c.whatsapp.net/chat]
```

* Download the PDF file prueba.pdf

The PDF cannot be displayed correctly in the console. If we want to view it, we must copy the entire content to a file and save it in .pdf format to view it. Additionally, how do the TCP and HTTP headers appear when performing a GET request



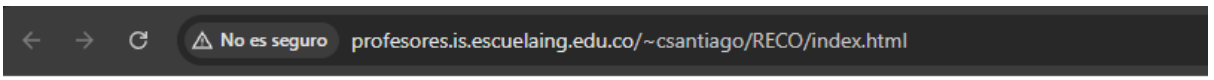

```
        13854 884.749713    192.168.1.2          45.239.88.86         HTTP       98 GET /~csantiago/RECO/prueba.pdf HTTP/1.0
        13989 884.782228    45.239.88.86         192.168.1.2          HTTP     1463 HTTP/1.1 200 OK  (application/pdf)
```



* Download the image file network.png – HTTP

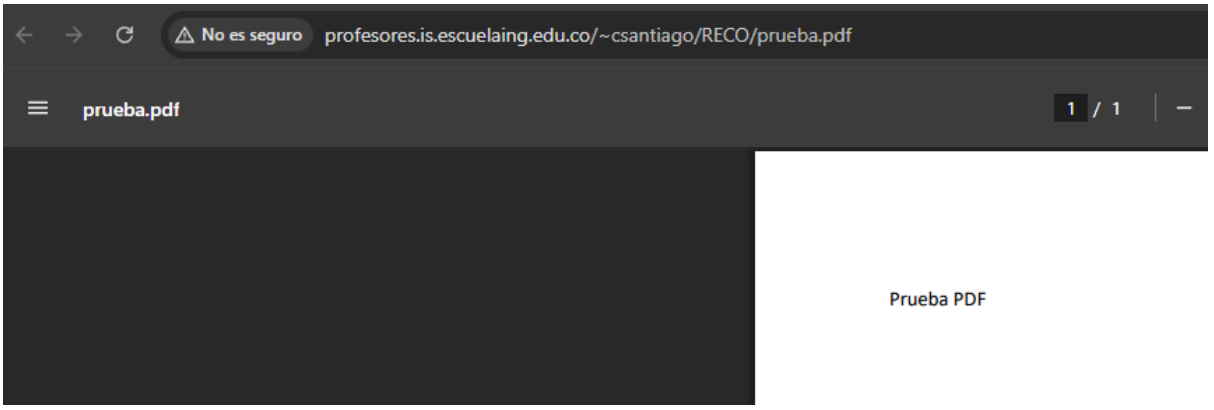
Here we couldn’t download the network PNG because the server only had the prueba.pdf; requesting prueba.png returned a 404 Not Found, showing how HTTP depends on the correct resource path and file availability.

* Use the browser to view the same pages you accessed with TELNET. * Present and explain the capture results.

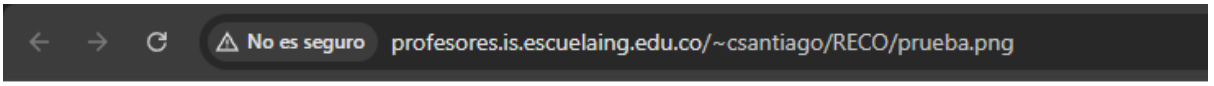


Espacio de prueba del Laboratorio de RECO

Esta es un archivo de prueba para revisar el funcionamiento del protocolo HTTP y TCP



Here is shown that we couldn't find a resource connected to the png view



Not Found

The requested URL was not found on this server.

```
25565 1191.495161   192.168.1.2     45.239.88.86    HTTP     100 GET /~csantiago/RECO/prueba.png HTTP/1.0
28357 1271.395474   192.168.1.2     45.239.88.86    HTTP      98 GET /~csantiago/RECO/prueba.png HTTP/1.0
41496 1520.419916   192.168.1.2     45.239.88.86    HTTP    1008 GET /~csantiago/RECO/prueba.pdf HTTP/1.1
41666 1520.450684   45.239.88.86    192.168.1.2     HTTP    1500 HTTP/1.1 200 OK  (application/pdf)
44448 1583.479280   192.168.1.2     45.239.88.86    HTTP    1008 GET /~csantiago/RECO/prueba.png HTTP/1.1
44461 1583.486557   45.239.88.86    192.168.1.2     HTTP     475 HTTP/1.1 404 Not Found  (text/html)
```

Packet captures show that there's not much difference between the use of the TELNET protocol. However, we saw that packets with the TELNET protocol were observed with "Continuation" information, indicating that the file was being downloaded, and this could be viewed in the console. With TELNET, we used the HTTP protocol without a web browser.

* What difference do you find between the files downloaded via TELNET and via the browser?
The main difference is the display. With TELNET, to view each file, we had to copy the downloaded content, paste it into a file, and open it in the specified format. With the browser, it offered us the way to view the files directly.

2. DNS Service Test

Perform the following DNS tests. Go to https://centralops.net/co and check the domains listed below and perform the indicated tests:

- escuelaing.edu.co
- jbb.gov.co
- google.com

Domain	# NS	Assigned Date	Registrar	Entity ID	Last Updated	Valid Until	IP Range / Authority	Company / Institution
escuelaing.edu.co	2	1998-06-02	.CO internet S.A.S	1111111	2025-07-19	2028-12-31	45.239.88.0/22 (LACNIC / AS266862)	Escuela Colombiana de Ingeniería Julio Garavito
jbb.gov.co		—	—	—	—	—	—	Jardín Botánico de Bogotá
google.com	4	1997-09-15	MarkMonitor Inc.	—	2019-09-09	2028-09-14	Multiple global IP ranges (Google ASNs)	Google LLC
ox.ac.uk	6	2003-09-17	Fastly, inc.	SKYCA-3	2024-04-24	2024-04-26	163.1.88.0/24 (Oxford IT)	University of Oxford

```
escuelaing.edu.co  IN    NS    ns2.escuelaing.edu.co
escuelaing.edu.co  IN    NS    ns1.escuelaing.edu.co
```

Domain Whois record

Queried **whois.nic.co** with "**escuelaing.edu.co**"...

```
Domain Name: escuelaing.edu.co
Registry Domain ID: D596567-CO
Registrar WHOIS Server:
Registrar URL: www.cointernet.com.co
Updated Date: 2022-11-06T18:30:02Z
Creation Date: 1998-06-02T00:00:00Z
Registry Expiry Date: 2025-12-31T23:59:59Z
Registrar: .CO Internet S.A.S.
Registrar IANA ID: 111111
Registrar Abuse Contact Email: soporte@cointernet.com.co
Registrar Abuse Contact Phone: +57.16169961
Domain Status: ok https://icann.org/epp#ok
Registry Registrant ID: REDACTED FOR PRIVACY
Registrant Name: REDACTED FOR PRIVACY
Registrant Organization: Escuela Colombiana de Ingenieria
Registrant Street: REDACTED FOR PRIVACY
Registrant Street: REDACTED FOR PRIVACY
Registrant Street: REDACTED FOR PRIVACY
Registrant City: REDACTED FOR PRIVACY
Registrant State/Province: Bogota
Registrant Postal Code: REDACTED FOR PRIVACY
Registrant Country: CO
Registrant Phone: REDACTED FOR PRIVACY
Registrant Phone Ext: REDACTED FOR PRIVACY
Registrant Fax: REDACTED FOR PRIVACY
Registrant Fax Ext: REDACTED FOR PRIVACY
```

```
Domain:
        ox.ac.uk

Registered For:
        University of Oxford

Domain Owner:
        University of Oxford

Registered By:
        University of Oxford

Servers:
        dns0.ox.ac.uk    129.67.1.190
        dns1.ox.ac.uk    129.67.1.191
        dns2.ox.ac.uk    163.1.2.190
        auth4.dns.ox.ac.uk       45.33.127.156
        auth4.dns.ox.ac.uk       2600:3c00:e000:19::1
        auth5.dns.ox.ac.uk       93.93.128.67
        auth5.dns.ox.ac.uk       2a00:1098:0:80:1000::10
        auth6.dns.ox.ac.uk       185.24.221.32
        auth6.dns.ox.ac.uk       2a02:2770:11::21a:4aff:febe:759b
```

```
Registrant Contact:
        Domain Registration

Registrant Address:
        IT Services, University of Oxford
        7-19 Banbury Road
        Oxford
        Oxfordshire
        OX2 6NN
        United Kingdom
        +44 1865 273619 (Phone)
        domain-registration@it.ox.ac.uk

Renewal date:
        Friday 26th Jul 2024

Entry updated:
        Friday 10th March 2023

Entry created:
        Wednesday 17th September 2003
```

3. NTP Server

Why is it important to ensure all computing devices in an infrastructure have the same time?

It is important to ensure all computing devices in an infrastructure have the same time because synchronized clocks are essential for maintaining security, consistency, and reliability in the network. Without accurate time, authentication mechanisms like Kerberos or digital certificates may fail, logs and events can appear out of order making troubleshooting difficult, and distributed systems or databases may experience conflicts in data integrity. Using an NTP server ensures that all devices share a common and reliable time reference.

Install an NTP server on one of your machines and configure the other machines (Linux Slackware, Solaris, Windows Server, and CentOS for groups of 3) to sync their time with the NTP server. A total of 1 NTP server should be configured on Solaris or Linux Slackware, and the other operating systems installed in Lab No. 1 and 2 on virtual machines, with all the physical machines configured as NTP clients. For example, in the case of 2-student groups, the installation could be:

Team No.01:
- Solaris → NTP Server

First, we verified that NTP is installed

```
root@solaris:~# pkg info network/ntp
             Nombre: service/network/ntp
            Resumen: Network Time Protocol Daemon v4
       DescripciÃ³n: Network Time Protocol v4, NTP Daemon and Utilities
         CategorÃ-a: System/Services
             Estado: Instalado
             Editor: solaris
           VersiÃ³n: 4.2.8.11 (4.2.8p11)
           Sucursal: 11.4.0.0.1.14.0
Fecha de empaquetado: 14 de agosto de 2018, 17:23:28
            TamaÃ±o: 5.80 MB
              FMRI: pkg://solaris/service/network/ntp@4.2.8.11-11.4.0.0.1.14.0
:20180814T172328Z
     URL de proyecto: http://www.ntp.org/
      URL de origen: http://archive.ntp.org/ntp4/ntp-4.2/ntp-4.2.8p11.tar.gz
root@solaris:~#
```

Now, create /etc/inet/ntp.conf, include this inside

```
root@solaris:~# cat /etc/inet/ntp.conf
# Use public NTP servers as a reference
server 0.pool.ntp.org
server 1.pool.ntp.org
server 2.pool.ntp.org

# Allow clients in this network to synchronize
restrict default kod nomodify notrap nopper noquery
restrict 127.0.0.1
restrict ::1
restrict 10.2.77.0 mask 255.255.0.0 nomodify notrap
root@solaris:~#
```

Now, we enable and start the server

```
root@solaris:~# svcadm enable ntp
```

```
root@solaris:~# svcs -xv ntp
svc:/network/ntp:default (Network Time Protocol (NTP) Version 4)
 Estado: online desde 25 de septiembre de 2025, 16:36:00 -05
   Consulte: man -M /usr/share/man -s 4 ntp.conf
   Consulte: man -M /usr/share/man -s 8 ntpd
   Consulte: man -M /usr/share/man -s 8 ntpq
   Consulte: /var/svc/log/network-ntp:default.log
Impacto: ninguno.
root@solaris:~#
```

Now, check NTP peers and their synchronization with **ntpq -p**

```
root@solaris:~# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
+cronos.unad.edu 200.25.3.11      3 u   10   64  377    4.427    2.747  12.773
*0.co.ntp.edgeun 129.6.15.28      2 u   27   64  377    3.238    4.287  15.259
root@solaris:~#
```

- Linux Slackware → NTP Client

First verify that NTP is installed

```
root@local:~# which ntpd
/usr/sbin/ntpd
root@local:~# ntpd --version
ntpd 4.2.8p15@1.3728-o Fri May 21 19:02:16 UTC 2021 (1)
root@local:~# _
```

Now we mount the ISO to download the file libedit

```
root@local:~# mount /dev/cdrom /mnt/cdrom
mount: /mnt/cdrom: WARNING: source write-protected, mounted read-only.
```

Then we look for the file path and proceed to download

```
root@local:~# ls /mnt/cdrom
ANNOUNCE.15.0          ChangeLog.txt     README_LVM.TXT     extra/
CHANGES_AND_HINTS.TXT  EFI/              README_RAID.TXT    isolinux/
CHECKSUMS.md5          FILELIST.TXT      README_UEFI.TXT    kernels/
CHECKSUMS.md5.asc      GPG-KEY           RELEASE_NOTES      pasture/
COPYING                PACKAGES.TXT      SPEAKUP_DOCS.TXT   patches/
COPYING3               README.TXT        SPEAK_INSTALL.TXT  slackware64/
COPYRIGHT.TXT          README.initrd     Slackware-HOWTO    testing/
CRYPTO_NOTICE.TXT      README_CRYPT.TXT  UPGRADE.TXT        usb-and-pxe-installers/
root@local:~# ls /mnt/cdrom/slackware64
CHECKSUMS.md5      FILE_LIST     PACKAGES.TXT@  ap/  e/  k/    l/  t/    x/    xfce/
CHECKSUMS.md5.asc  MANIFEST.bz2  a/             d/   f/  kde/  n/  tcl/  xap/  y/
root@local:~# ls /mnt/cdrom/slackware64/l | grep libedit
libedit-20210910_3.1-x86_64-1.txt
libedit-20210910_3.1-x86_64-1.txz
libedit-20210910_3.1-x86_64-1.txz.asc
root@local:~#
```

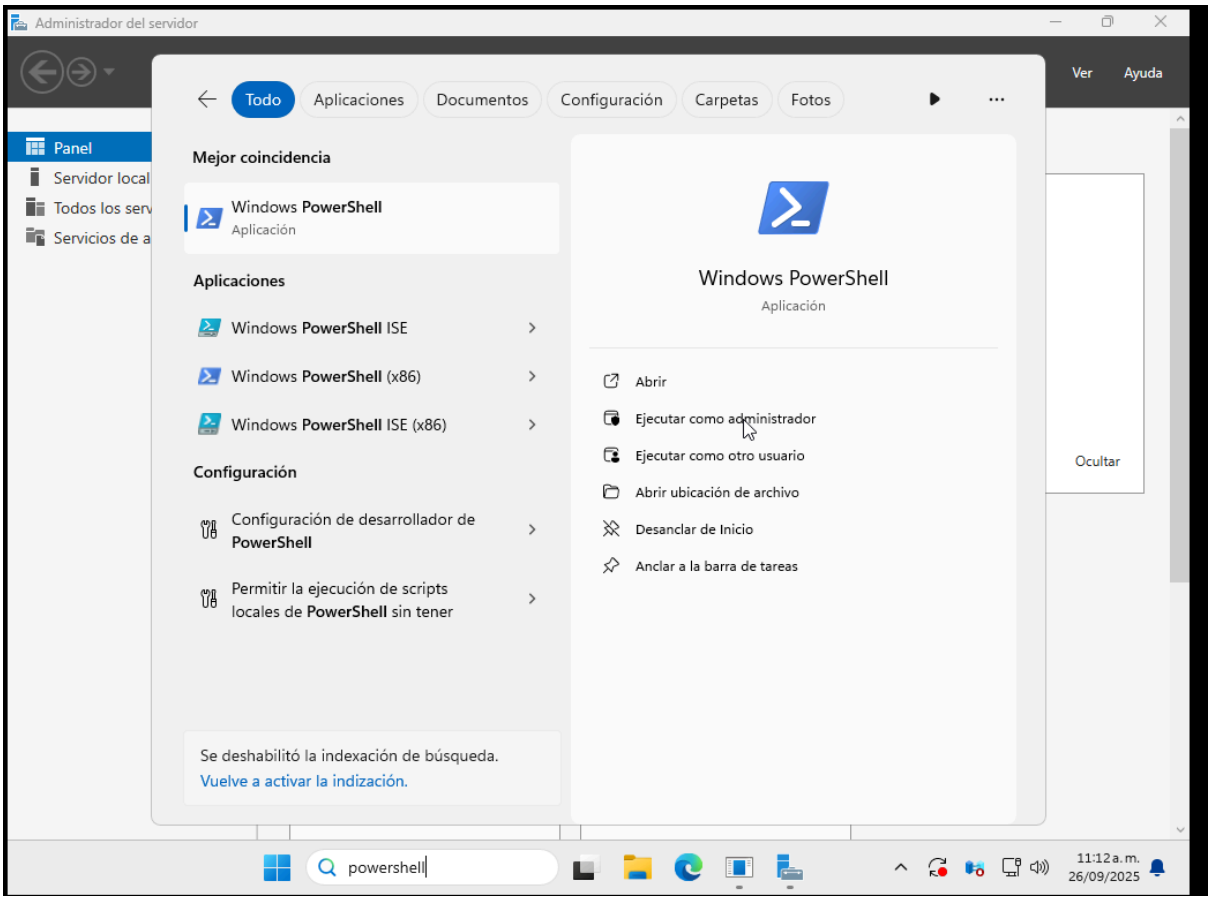
```
root@local:~# installpkg /mnt/cdrom/slackware64/l/libedit-*.txz
Verifying package libedit-20210910_3.1-x86_64-1.txz.
Installing package libedit-20210910_3.1-x86_64-1.txz [REC]:
PACKAGE DESCRIPTION:
# libedit (Command line editor library)
#
# This is an autotool and libtoolized port of the NetBSD Editline
# library (libedit). This Berkeley-style licensed command line editor
# library provides generic line editing, history, and tokenization
# functions, similar to those found in GNU Readline.
#
# Homepage: http://www.thrysoee.dk/editline/
#
Executing install script for libedit-20210910_3.1-x86_64-1.txz.
Package libedit-20210910_3.1-x86_64-1.txz installed.
root@local:~#
```

Now we restart the service and verify that the connection to the server is in correct conditions.

```
root@local:~# /etc/rc.d/rc.ntpd restart
Stopping NTP daemon...
Starting NTP daemon:  /usr/sbin/ntpd -g -u ntp:ntp
root@local:~# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 10.2.77.35      200.25.3.17      3 u    1   64    1    0.272  -12.052   0.306
root@local:~# _
```

- Windows Server with GUI → NTP Client

First we open Powershell as administrator



We restart the w32time service, and configure the client

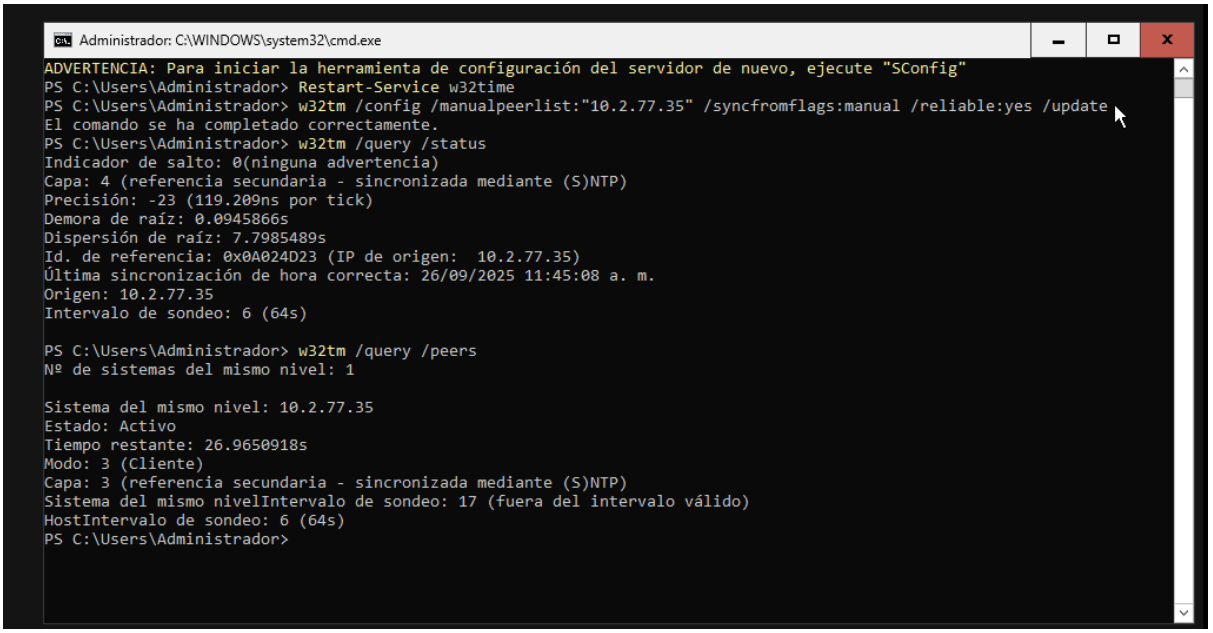
```
PS C:\Users\Administrador> Restart-Service w32time
PS C:\Users\Administrador> w32tm /config /manualpeerlist:"10.2.77.35" /syncfromflags:manual /reliable:yes /upd
ate
El comando se ha completado correctamente.
PS C:\Users\Administrador>
```

Now we verify that it has connected correctly to the server

```
PS C:\Users\Administrador> w32tm /query /status
Indicador de salto: 0(ninguna advertencia)
Capa: 4 (referencia secundaria - sincronizada mediante (S)NTP)
Precisión: -23 (119.209ns por tick)
Demora de raíz: 0.0941048s
Dispersión de raíz: 0.2918770s
Id. de referencia: 0x0A024D23 (IP de origen:  10.2.77.35)
Última sincronización de hora correcta: 26/09/2025 11:26:06 a. m.
Origen: 10.2.77.35
Intervalo de sondeo: 7 (128s)

PS C:\Users\Administrador> w32tm /query /peers
Nº de sistemas del mismo nivel: 1

Sistema del mismo nivel: 10.2.77.35
Estado: Activo
Tiempo restante: 54.6005610s
Modo: 3 (Cliente)
Capa: 3 (referencia secundaria - sincronizada mediante (S)NTP)
Sistema del mismo nivelIntervalo de sondeo: 6 (64s)
HostIntervalo de sondeo: 6 (64s)
PS C:\Users\Administrador>
```

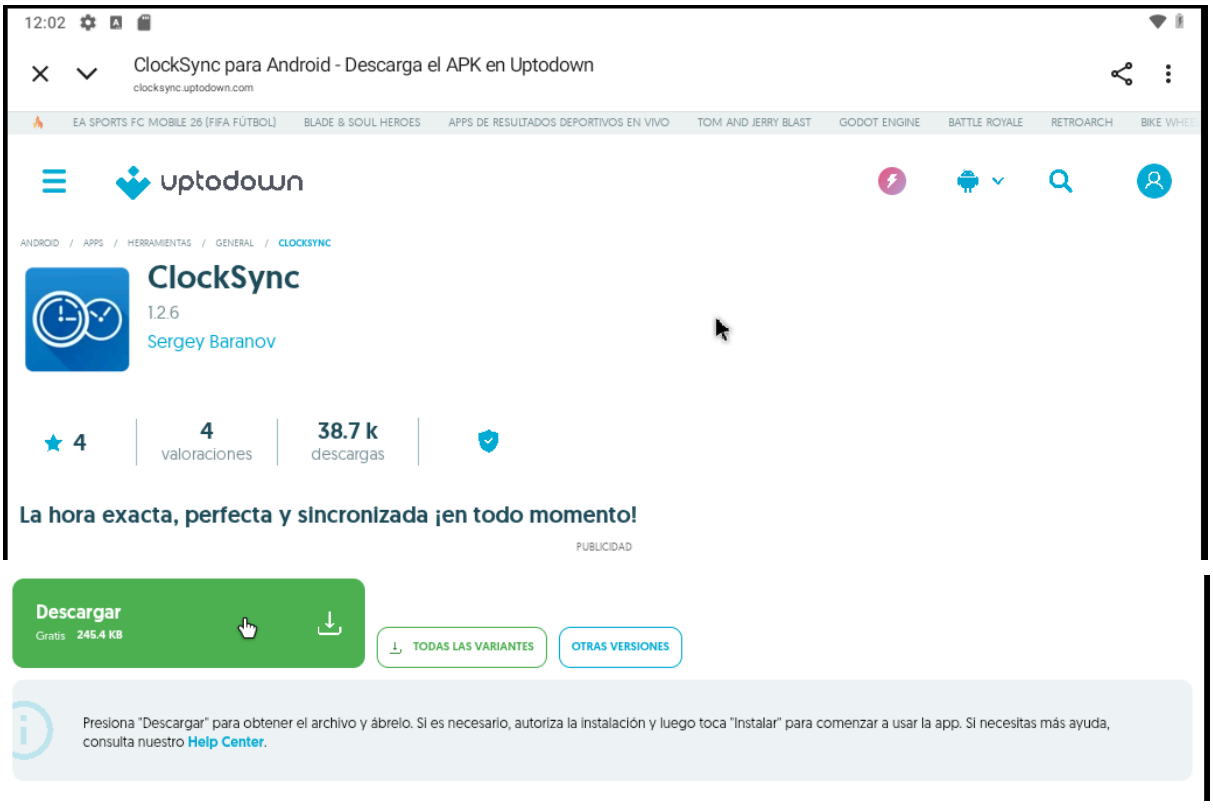
- Windows Server without GUI → NTP Client

In this machine, we follow the same steps as Windows Server GUI.

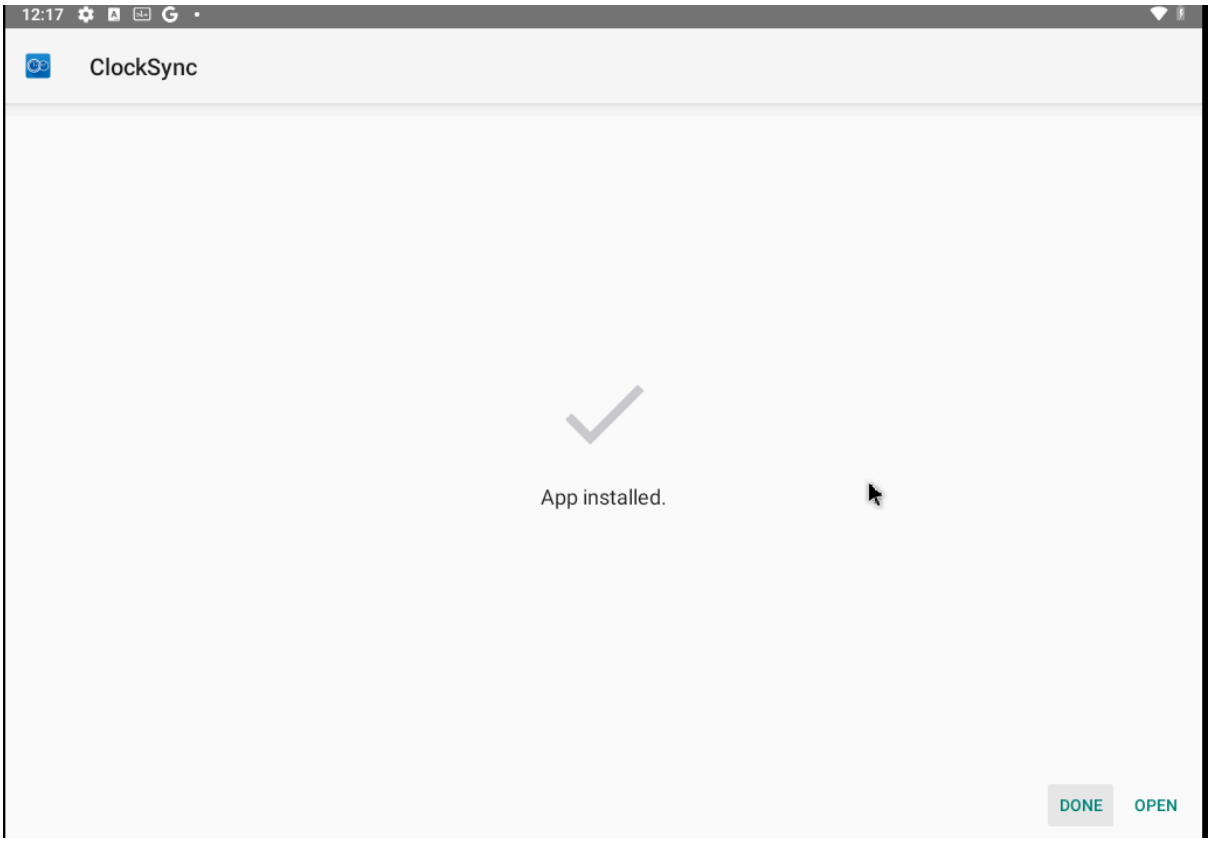


- Android → NTP Client

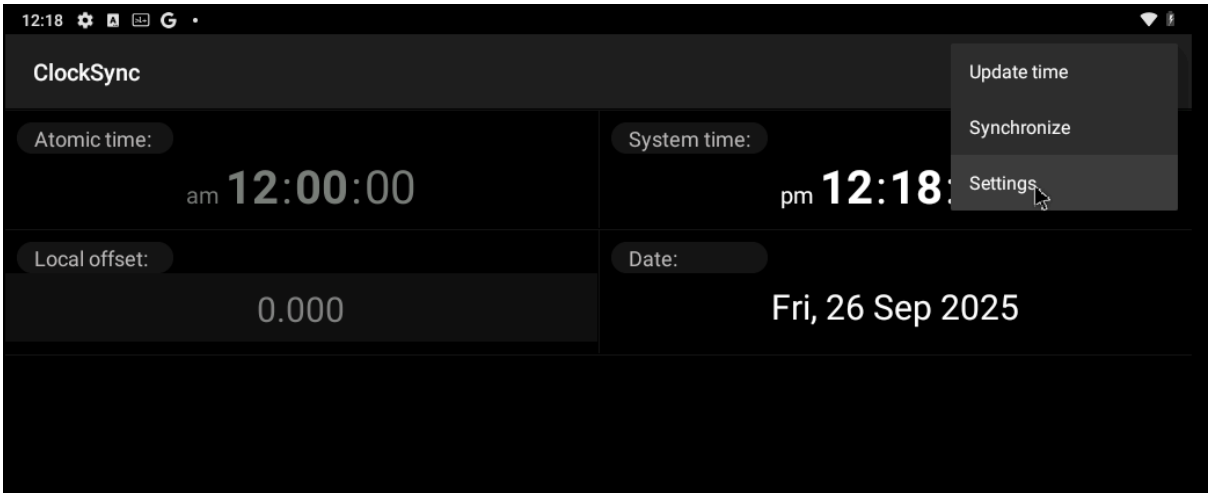
First, we go to the browser to install the ClockSync application



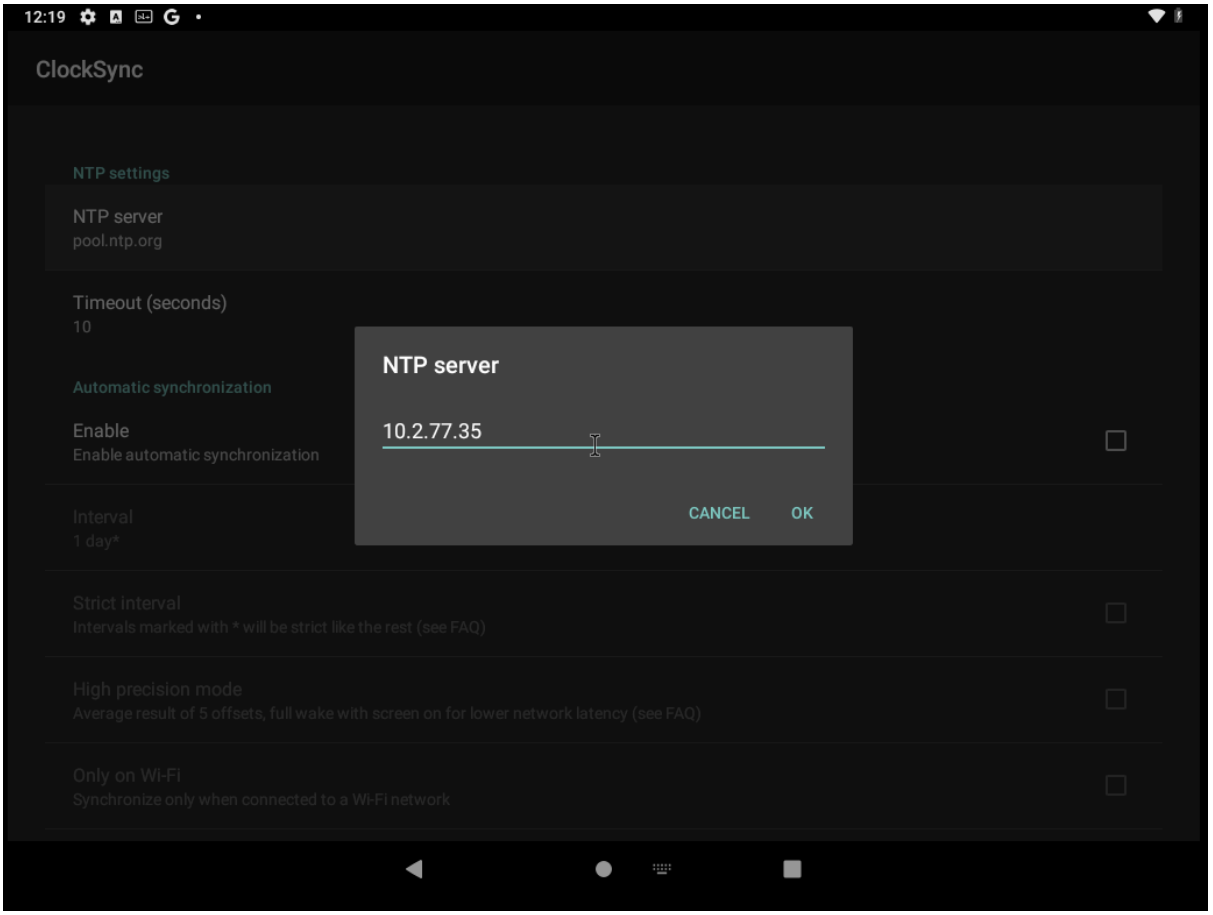
Once installed we click on open



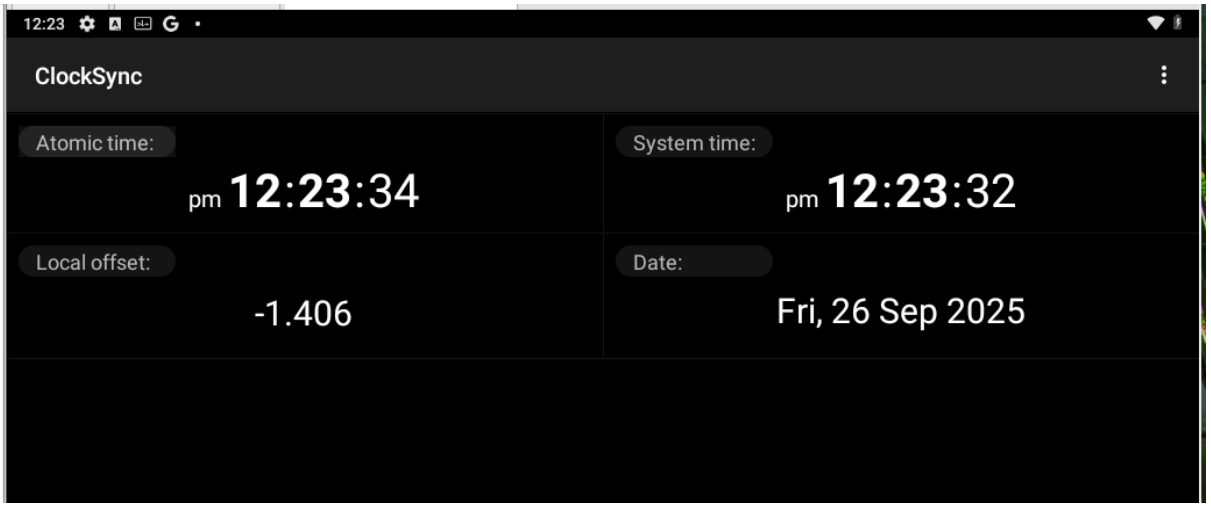
Once in the application, click on settings.



In the NTP settings section, we click on NTP Server and enter the IP of the machine where the NTP server resides.



We click OK and then check the synchronization with the time.



Finally, on Solaris we verify that the clients connected correctly to the NTP server

```
root@solaris:~# ntpq -c mrulist
Ctrl-C will stop MRU retrieval and display partial results.
Retrieved 6 unique MRU entries and 0 updates.
lstint avgint rstr r m v  count rport remote address
==============================================================================
   136     90  390 . 4 4     65   123 cronos.unad.edu.co
   166     94  390 . 4 4     62   123 0.co.ntp.edgeuno.com
   271     47  300 . 3 3      5 58761 10.2.77.33
  2381     48  300 . 3 3      4   123 10.2.77.32
  3199     57  300 . 3 3     16   123 10.2.77.34
  4930     36  300 . 3 4     21   123 10.2.77.31
root@solaris:~# Sep 26 12:28:00 solaris sendmail[1584]: unable to qualify my own
```

Team No.02:
- Linux Slackware → NTP Server

First, verify that NTP is installed

```
root@aysrSlack:~# which ntpd
/usr/sbin/ntpd
root@aysrSlack:~# ls /etc/ntp.conf
/etc/ntp.conf
```

Edit /etc/ntp.conf as follows

```
# Sample /etc/ntp.conf:  Configuration file for ntpd.

#
# Undisciplined Local Clock. This is a fake driver intended for backup
# and when no outside source of synchronized time is available. The
# default stratum is usually 3, but in this case we elect to use stratum
# 0. Since the server line does not have the prefer keyword, this driver
# is never used for synchronization, unless no other other
# synchronization source is available. In case the local host is
# controlled by some external source, such as an external oscillator or
# another protocol, the prefer keyword would cause the local host to
# disregard all other synchronization sources, unless the kernel
# modifications are in use and declare an unsynchronized condition.
#
# server        127.127.1.0     # local clock
# fudge 127.127.1.0 stratum 10

#
# NTP server (list one or more) to synchronize with:
server 0.pool.ntp.org iburst
server 1.pool.ntp.org iburst
server 2.pool.ntp.org iburst
server 3.pool.ntp.org iburst

# Allow access to LAN clients
restrict 10.2.77.0 mask 255.255.0.0 nomodify notrap_
#
# Full path of a directory where statistics files should be created
#
statsdir /var/lib/ntp/stats

#
# Location of an alternate log file to be used instead of the default system syslog(3) facility
#
logfile /var/log/ntp
```

```
#
# Trust ourselves.  :-)
restrict 127.0.0.1
restrict ::1
```

We give execution permissions and start the service

```
root@aysrSlack:~# chmod +x /etc/rc.d/rc.ntpd
root@aysrSlack:~# /etc/rc.d/rc.ntpd start
Starting NTP daemon:  /usr/sbin/ntpd -g -u ntp:ntp
root@aysrSlack:~# _
```

Then, we prove that the NTP server is working

```
root@aysrSlack:~# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*0.co.ntp.edgeun 129.6.15.28      2 u    1   64    1    2.410   +1.545   2.939
 saturno.unad.ed 200.25.3.11      3 u    1   64    1    4.021   +2.177   3.211
 0.cl.ntp.edgeun 129.6.15.28      2 u    1   64    1   68.522   +1.571   6.982
root@aysrSlack:~#
```

- Solaris → NTP Client

Using vi or nano edit /etc/inet/ntp.conf, use cat to see the file.

```
root@solaris:~# cat /etc/inet/ntp.conf




server 10.2.77.31 iburst



root@solaris:~#
```

Now restart the ntp service and check the status

```
root@solaris:~# svcadm enable ntp
root@solaris:~# svcs -xv ntp
svc:/network/ntp:default (Network Time Protocol (NTP) Version 4)
 Estado: online desde 30 de septiembre de 2025,  4:25:22 -05
   Consulte: man -M /usr/share/man -s 4 ntp.conf
   Consulte: man -M /usr/share/man -s 8 ntpd
   Consulte: man -M /usr/share/man -s 8 ntpq
   Consulte: /var/svc/log/network-ntp:default.log
Impacto: ninguno.
root@solaris:~#
```

Then, we prove that the NTP server is working

```
root@solaris:~# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 10.2.77.31      23.155.40.38     2 u    -   64    1    0.807   -1.077   0.017
root@solaris:~#
```

- Windows Server with GUI → NTP Client

We proceed to configure this machine in the same way that we configured Windows Server Desktop Experience for the Lab01 machines, the only thing that varies here is that in this case we must write the IP of the Linux Slackware machine

```
PS C:\Users\Administrador> w32tm /config /manualpeerlist:"10.2.77.31" /syncfromflags:manual /reliable:yes /upd
ate
El comando se ha completado correctamente.
PS C:\Users\Administrador> w32tm /query /status
Indicador de salto: 0(ninguna advertencia)
Capa: 3 (referencia secundaria - sincronizada mediante (S)NTP)
Precisión: -23 (119.209ns por tick)
Demora de raíz: 0.0968698s
Dispersión de raíz: 7.7775713s
Id. de referencia: 0x0A024D1F (IP de origen:  10.2.77.31)
Última sincronización de hora correcta: 30/9/2025 19:19:18
Origen: 10.2.77.31
Intervalo de sondeo: 6 (64s)

PS C:\Users\Administrador> w32tm /query /peers
Nº de sistemas del mismo nivel: 1

Sistema del mismo nivel: 10.2.77.31
Estado: Activo
Tiempo restante: 32.8086158s
Modo: 3 (Cliente)
Capa: 2 (referencia secundaria - sincronizada mediante (S)NTP)
Sistema del mismo nivelIntervalo de sondeo: 17 (fuera del intervalo válido)
HostIntervalo de sondeo: 6 (64s)
PS C:\Users\Administrador>
```

4. Structured Cabling and Cable Construction

To build a technological infrastructure, elements are required to connect computing devices. The structured cabling standards are used to connect elements, maintain order, facilitate growth, and promote the management of network physical elements. The following activities are focused on understanding this structure.

a. Patch Cord Construction

Individually:

- Following the professor's instructions and the presentation posted in the classroom, crimp two RJ45-RJ45 cables, one straight and one crossover.

- What is the purpose of each one?

A straight-through cable is mainly used to connect devices of different types, such as a computer to a switch or a switch to a router, while a crossover cable is used to connect devices of the same type, such as two computers directly or two switches, without needing an intermediate device.

- Use the cable tester to check that the cable was correctly made.

- Document the process and include photos proving you made it.

Straight-Through Cable

A straight-through cable is used to connect devices that work on different layers of the network, such as a computer to a switch, or a switch to a router.

Materials Needed:

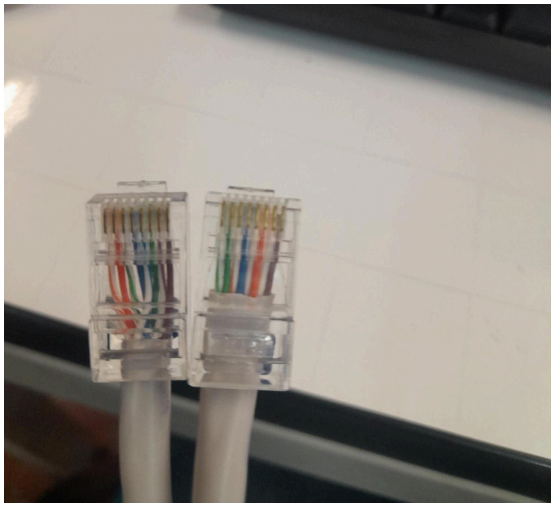
- UTP cable
- RJ45 connectors (for UTP cables)
- Crimping tool
- Cable stripper or scissors

Steps:

1. **Prepare the Cable**:
    - Use the cable stripper or scissors to strip about 1 inch (2.5 cm) of the outer jacket from both ends of the UTP cable.
    - After stripping the jacket, you will expose the 4 twisted pairs of wires (typically colored as blue, orange, green, and brown, with solid and striped versions of each color).

2. **Untwist the Pairs**:
    - Untwist the pairs of wires gently, and align them in the proper order for a straight-through connection:
        - **Pinout for Straight-Through (T568B standard)**:
            - Pin 1: White with orange stripe
            - Pin 2: Orange
            - Pin 3: White with green stripe
            - Pin 4: Blue
            - Pin 5: White with blue stripe
            - Pin 6: Green
            - Pin 7: White with brown stripe
            - Pin 8: Brown
    - Ensure the wires are flat and in the correct order.
3. **Trim the Wires**:
    - Once the wires are in the correct order, trim them so they are even and about 1/2 inch (1.25 cm) long.
4. **Insert the Wires into the RJ45 Connector**:
    - Hold the RJ45 connector with the clip facing down. Insert the wires into the connector, making sure that each wire goes into its corresponding pin slot in the connector. The wires should go all the way to the front of the connector.
5. **Crimp the Connector**:
    - Place the RJ45 connector into the crimping tool and squeeze the handle firmly. This will push the metal pins inside the connector into the wires, securing them in place.
    - Repeat the process for the other end of the cable.

6. **Test the Cable**:
    - After both ends are crimped, use a cable tester to ensure the cable is wired correctly and that all connections are secure. (The test was proved in the lab session)



Crossover Cable

A crossover cable is typically used to connect devices like two computers, two switches, or two routers without a hub or switch in between.

Pinout for Crossover Cable (T568A to T568B):

- On one end, use the **T568A standard**:
    - Pin 1: White with green stripe
    - Pin 2: Green
    - Pin 3: White with orange stripe
    - Pin 4: Blue
    - Pin 5: White with blue stripe
    - Pin 6: Orange
    - Pin 7: White with brown stripe

    - Pin 8: Brown

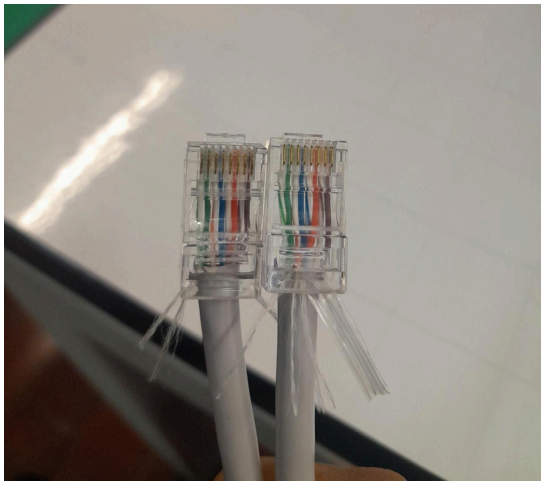
- On the other end, use the **T568B standard**:
    - Pin 1: White with orange stripe
    - Pin 2: Orange
    - Pin 3: White with green stripe
    - Pin 4: Blue
    - Pin 5: White with blue stripe
    - Pin 6: Green
    - Pin 7: White with brown stripe
    - Pin 8: Brown

Steps for Crimping a Crossover Cable:

1. **Prepare the Cable**:
    - Same as in the straight-through process, strip about 1 inch (2.5 cm) of the outer jacket from both ends of the UTP cable.
2. **Untwist the Pairs**:
    - Untwist the wire pairs and arrange them in the proper pinout order:
        - **T568A for one end**
        - **T568B for the other end**
3. **Trim the Wires**:
    - Trim the wires to about 1/2 inch (1.25 cm).
4. **Insert the Wires into the RJ45 Connector**:
    - Insert the wires into the RJ45 connector in the proper order for each end:

- **One end: T568A**
- **Other end: T568B**

5. **Crimp the Connector**:
    - Use the crimping tool to secure the wires by pressing the metal pins into the wires.
6. **Test the Cable**:
    - As with the straight-through cable, use a cable tester to verify the proper connections and ensure the cable functions correctly.



- Show your professor the result of your work.

b. Patch Panel Crimping

c. Knowledge of the University's Structured Cabling

Observe the structured cabling in Building I at the School and identify the components of the structured cabling in the building. Include photos related to the topic (which prove that you took them).


In the images provided, we can observe a well-organized structured cabling system inside a network rack. The components of the structured cabling system in this building include:

1. **Patch Panels**: These are the horizontal black components visible in the rack. They are used for organizing and managing network cables. Patch panels allow cables from various areas of the building to be neatly routed and connected to the network.

2. **Cables**: The cables visible in the image are mainly Ethernet cables (likely Cat 5e, Cat 6, or Cat 6a), organized into bundles for ease of management. The white cables are most likely the standard network cables used for data transmission, while the red cables might be used for specific connections or to identify critical circuits.

3. **Cable Management**: There are cable organizers and management systems (such as cable ties and trays) ensuring the cables are kept tidy and preventing tangling. Proper cable management is important to maintain airflow and reduce the risk of overheating and damage.

4. **Rack Enclosure**: The entire setup is housed within a network rack or cabinet, which serves to keep the equipment organized, secure, and protected from environmental factors.

This structured cabling setup allows for easy management of the building's network infrastructure, ensuring efficient data flow, scalability, and the ability to troubleshoot or reconfigure the network when needed.

Conclusions

This laboratory allowed students to integrate theoretical knowledge with practical skills by working with both simulated and real network environments. Through the configuration of DNS, HTTP, FTP, and email services in Cisco Packet Tracer, the activities demonstrated how application and transport layer protocols function in real communication scenarios. Students were able to observe how ports, addresses, and services interact to ensure that information is delivered accurately across the network. The use of Wireshark reinforced this understanding by providing visibility into packet structures, headers, and payloads, making it possible to analyze and compare different protocols and their behavior in detail.

The exercises on structured cabling highlighted the critical role of physical infrastructure in networking. Constructing straight-through and crossover cables, working with patch panels, and identifying cabling components emphasized the importance of following standards to maintain order, reliability, and scalability in network design. This hands-on experience complemented the logical configuration tasks, showing how both software and hardware aspects must work together for networks to function properly.

In conclusion, the lab provided a comprehensive view of networking by combining service configuration, packet analysis, and cabling practices. Students not only learned how to configure and test essential network services but also gained an appreciation of the underlying mechanisms that support modern communication systems. This integration of theory and practice strengthens their ability to design, implement, and troubleshoot networks effectively, which is a fundamental skill for any networking professional.

Bibliography

Cisco Networking Academy. *Packet Tracer Resources and Tutorials*. Available at: https://www.netacad.com/

Cisco. *Understand the OSI Model and TCP/IP Model*. Available at: https://www.cisco.com/c/en/us/solutions/enterprise-networks/what-is-computer-networking.html

Wireshark Foundation. *Wireshark Documentation*. Available at: https://www.wireshark.org/docs/
Internet Assigned Numbers Authority (IANA). *Service Name and Transport Protocol Port Number Registry*. Available at: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml
Cloudflare. *What is DNS?*. Available at: https://www.cloudflare.com/learning/dns/what-is-dns/
Mozilla Developer Network (MDN). *HTTP Overview*. Available at: https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview
Internet Engineering Task Force (IETF). *Simple Mail Transfer Protocol (SMTP) RFC 5321*. Available at: https://datatracker.ietf.org/doc/html/rfc5321
Internet Engineering Task Force (IETF). *Post Office Protocol (POP3) RFC 1939*. Available at: https://datatracker.ietf.org/doc/html/rfc1939
W3Schools. *FTP Tutorial*. Available at: https://www.w3schools.com/ftp/
ISO/IEC. *Structured Cabling Standards*. Summary available at: https://www.iso.org/standard/63532.html
Network Encyclopedia. *Straight-through and Crossover Cables*. Available at: https://networkencyclopedia.com/