WAN Technologies

Lab 4 – Configuring ACLs over a small WAN and utilizing Wireshark to observe clear text and encrypted packets

Objectives

Background / Scenario

Part 1:

Step 1: Set Up the Topology and Initialize Devices

Step 2: Configure single area OSPF routing protocol across the WAN

Step 3: Configure Local LAN devices and verify remote connectivity (using Telnet and SSH protocols)

Part 2: ACLs

Task 1: Configure a standard ACL proposed by the organization

Task 2: Configure, Apply and Verify an Extended Named IPv4 ACL

Part 3:

Use Wireshark to identify traffic (both clear text and encrypted packets.)

Part 1: Identifying ICMP traffic

Part 2: Identifying Telnet traffic

Part 3: Identifying SSH traffic

Part 4:

Erase router configuration.

Background / Scenario

This is a hands-on WAN lab and your attendance is required to complete this weeks lab.

In this lab, you will work in a team of **four**. Two members of the team will work together configuring the Dublin office (Pod 1) and two members will work closely together configuring the Galway office (Pod 2).

Once both Pod's have setup their respective LAN's it is of utmost importance for all members of the team to communicate in order to setup WAN connectivity correctly, setup ACLs and observe traffic via Wireshark. The organization have some specific requirements regarding access control and you will need to implement these as a team to meet their expectations.

The organization is testing a simple site to site WAN network. They are currently going through some initial testing and require your networking and security expertise!

Grading
Depending on your individual / teams' engagement and progress will depend on the amount of marks you will score in this lab. Every member of the team needs to participate and answer the questions on Brightspace outlined below.

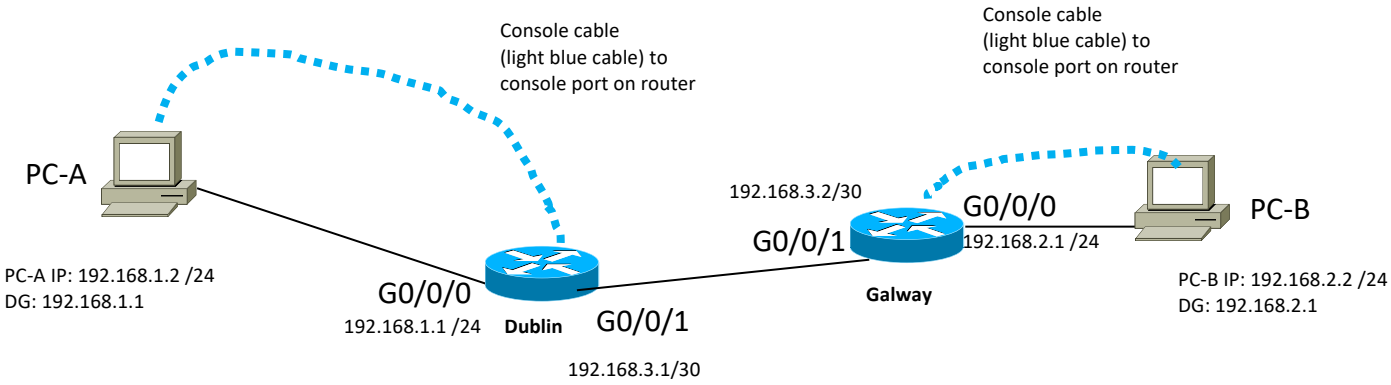
Before proceeding, open the "Lab 4 - Configuring ACLs over a small WAN - QUESTIONS -Hands-On activity -3% (2025)" quiz on the Brightspace page.

The lecturer will provide you with a code to access this quiz within your lab session.

Part 1:

Step 1: Set Up the Topology and Initialize Devices

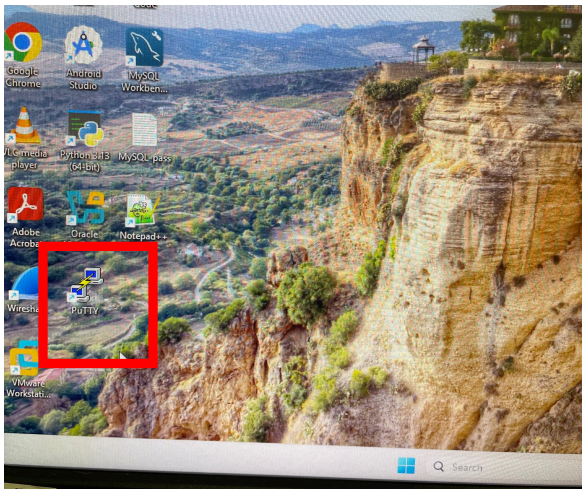
WAN Topology



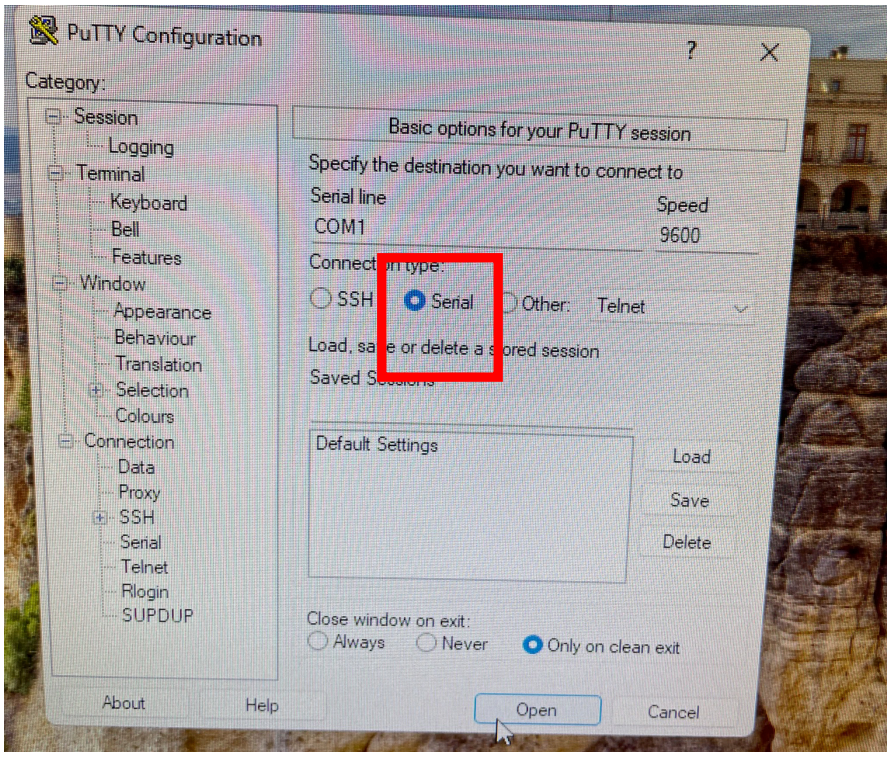
Important: Note the subnet mask /30 on the WAN link.

This next section provides an overview of connecting up a PC to a router in the lab. The example used is from E202 but the same concepts apply in any networking lab in TU Dublin (Blanchardstown).

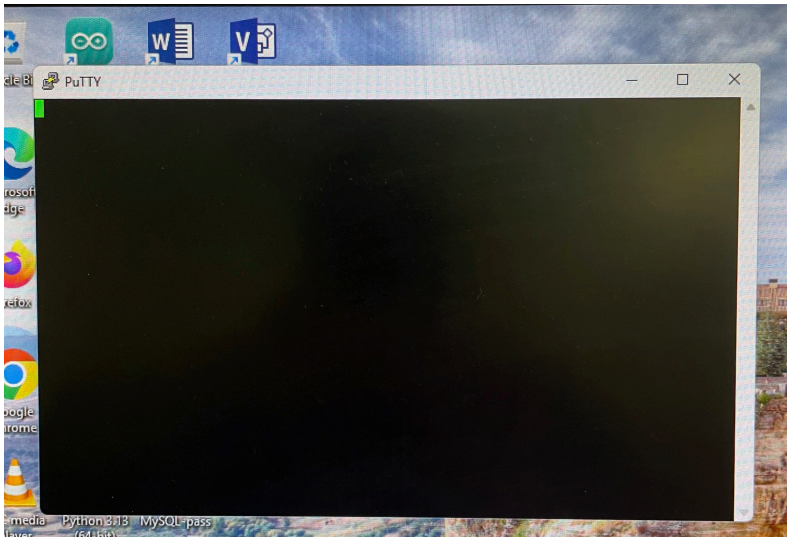
Use **Putty** software to connect to physical equipment (router).



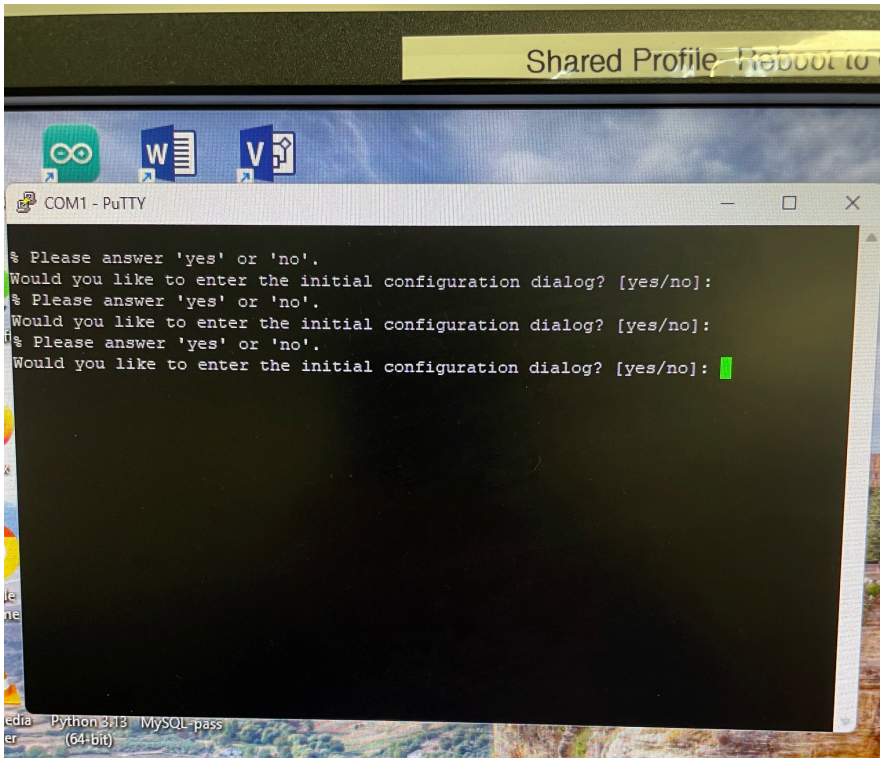
Note: this needs to be changed to Serial.



Note, Serial is now highlighted and we are connecting to **COM1.**
When Open is selected, the following window appears empty.



NOTE: If you have a console cable attached to your router, press Enter.



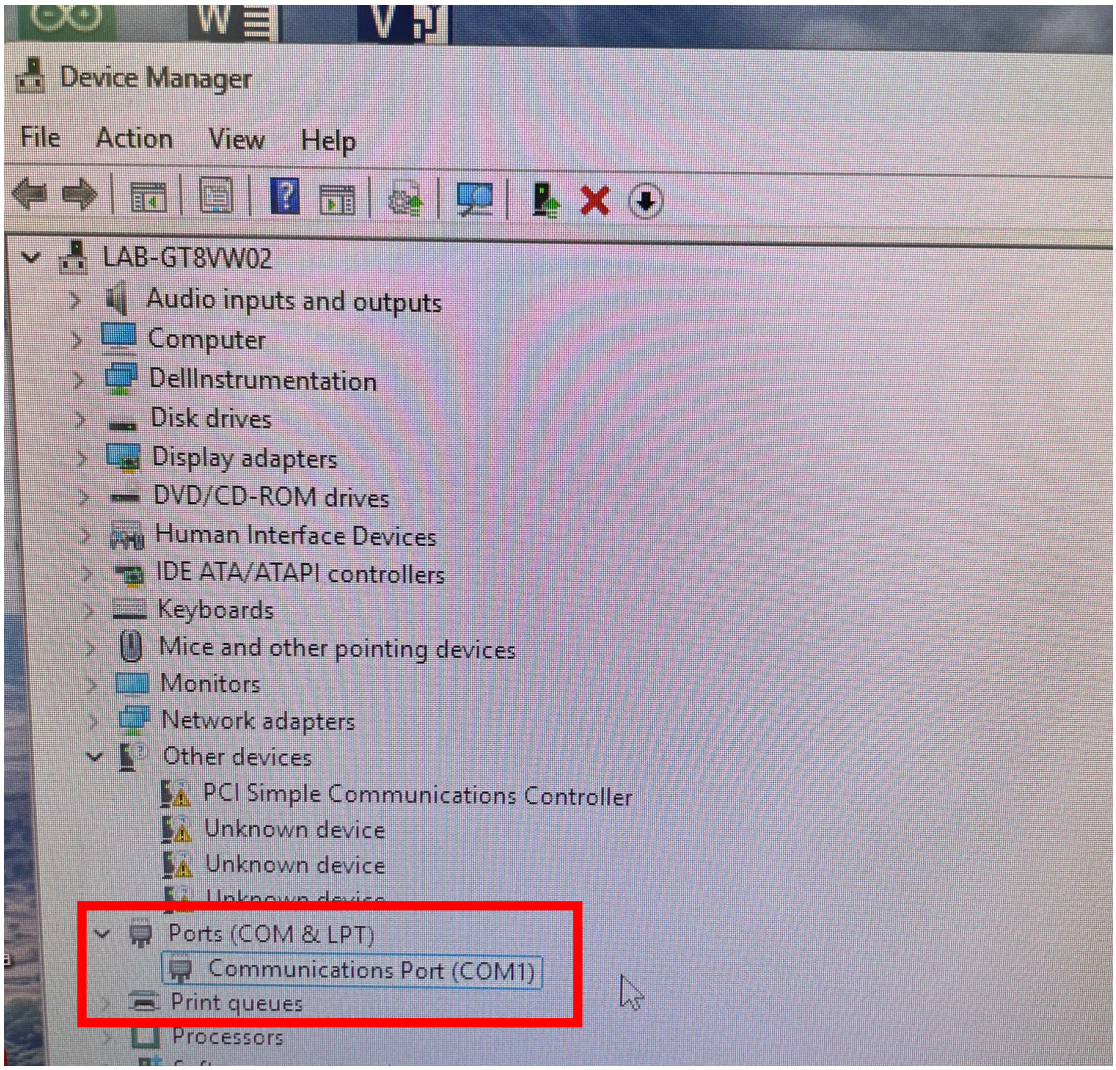
You are now ready to begin configuration on your router. Enter no to exit initial configuration dialog.

TROUBLESHOOTING (Software): If you cannot see any prompts from your router on screen, check your cabling.

Also, you may also check **Device Manager** to ensure that console is connected to **COM1** port. This may be different for your PC in the lab.

To enter device manager, right click on Windows icon on taskbar and select Device Manager. Check to ensure Com port is available and which port it is. In the below, COM1 is shown but this may be different for your lab PC.

.



If you are having further issues connecting to your router, please ask the lab lecturer for help.
Note, the PC may be attached to another COM port e.g. COM5, if this is the case, you will need to change the COM port to this number in Putty.

Configuration

> **NOTE: If you need help with configuration, please refer to the PDF on Brightspace "Routing Concepts, Static Routing and Dynamic Routing" where lots of different configuration examples are provided. Please note this is a very similar topology as WAN LAB 1. Further note: please refer back to previous labs and lecture notes to help with configuration activities.**

First, your instructor will allocate you to a group and designate your team with equipment.

You will cable the equipment as shown in the topology diagram.

Please note: ensure to add the appropriate cables per device.

IMPORTANT: These instructions are general guidelines ONLY as the equipment will differ depending on the lab (which may mean ports/commands may change. For example: below in the topology diagram it shows both G0/0/0 and G0/0/1– one or both of these device names may differ depending on the lab/room and particular device). Ensure to check the equipment.

Run a **'show ip int brief'** to check your interfaces on the router.

Set the IPv4 addresses on the Gigabit port 0/0/0 and G0/0/1 ports*
Ensure to enable the interfaces.

Note: *These interfaces will depend on the equipment you are using and the port that you plugged the cable into! Ensure to communicate between your Dublin and Galway Pods.

Configure IPv4 addresses on Dublin PC and Galway PC
You many need to refer back to Week 1 Lab – Appendix section if you cannot remember how to Add a static IPv4 address (on Windows 11).
Remember you need to have an Ethernet connection from PC-A to Dublin router and also another Ethernet cable from PC-B to Galway router.

Testing; You should now be able to ping your respective default gateway

e.g. from PC-A to Dublin Router and from Galway's PC-B to Galway's default gateway.

Before proceeding any further with configuration, open the "Lab 4 - Configuring ACLs over a small WAN - QUESTIONS -Hands-On activity -3% (2025)" quiz on the Brightspace page and enter your answer for question 1.

QUESTION 1 At this point, after configuring all PC's and appropriate IP addresses on all interfaces on the router, is it possible to ping from PC-A to PC-B across the WAN (and get successful reply packets)?

True-Yes it's possible to ping

False – No, it's not possible to ping (pings will fail)

Part 1:

Step 2: Configure single area OSPF routing protocol across the WAN

- Ensure you've connected the Dublin to Galway offices using an Ethernet cable (red cable is allowed here).
- Configure OSPF on both Dublin and Galway routers so that they both advertise their respective local LANs.

Now test your connections as before.

Tip: Temporarily Turn Off Windows 11 Host Firewalls
Test to see if you can ping from PC-A Dublin office to the Galway PC-B office.

Look at your routing table of both routers and see what networks your router knows about.

What are the responses to the following commands?

- show ip route
- show ip protocols

You should see 'O' – OSPF learned routes.

TESTING; You should be now able to ping from Dublin's PC-A to Galway's PC-B.

QUESTION 2 What wildcard mask for the WAN links is most suitable based on the choices below?

From the **command prompt (not from Putty)** on PC-A, you should be able to ping Galway's PC-B (192.168.2.2)
Also, from the **command prompt (not from Putty)** PC-B should be able to ping Dublin PC-A (192.168.1.2).

Part 1:

Step 3: Configure Local LAN devices and verify remote connectivity

Please note, the following shows sample Dublin configuration **ONLY** - E.g. for Dublin Router – Galway will also have similar configuration.

If you haven't done so previously it is recommended you configure the following:

- Set the hostname for the respective router
  Router(config)#hostname Dublin

- Configure the enable secret password to be "class".
  Dublin(config)#enable secret class

Configure Remote Access on both Dublin and Galway routers:

Configure all network devices for telnet and SSH support.

- Create a local user with the username **admin** and the encrypted password **cisco**
  Dublin(config)# **username admin secret cisco**

- Use **ccna-lab.com** as the domain name.
  Dublin(config)# **ip domain name ccna-lab.com**

- Generate crypto keys using a 2048-bit modulus.
  Dublin(config)# **crypto key generate rsa general-keys modulus 2048**

  This will take a moment to complete – press enter to continue

Configure the first five VTY lines on each device to support SSH and telnet connections and to authenticate to the local user database.

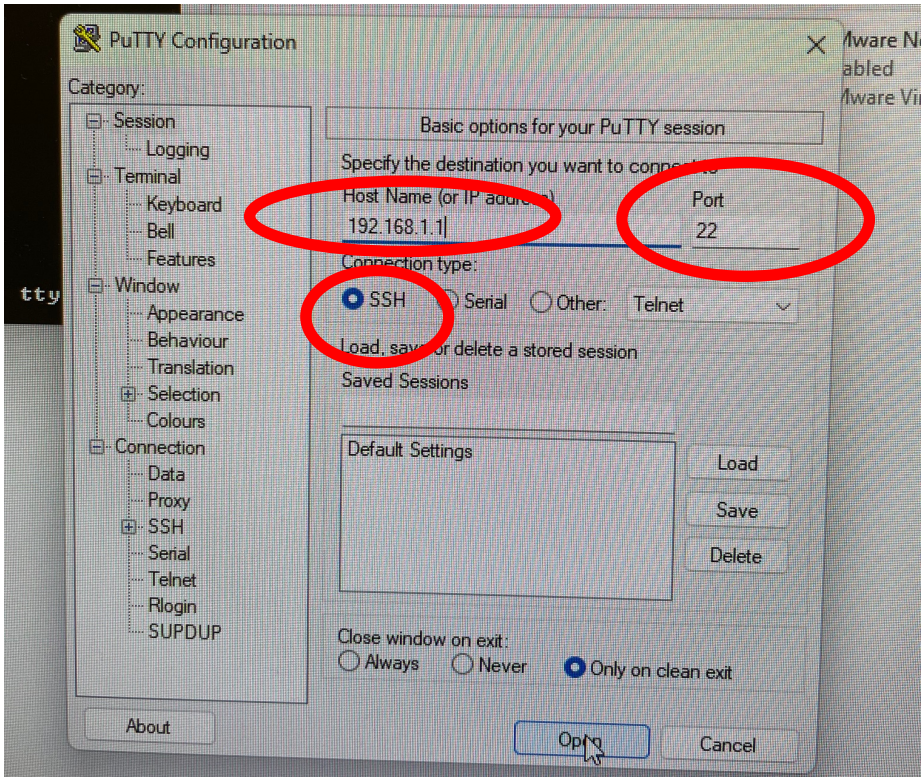
Dublin(config)# **line vty 0 4**

Dublin(config-line)# **transport input ssh telnet**
Dublin(config-line)# **login local**
Dublin(config-line)# **exit**

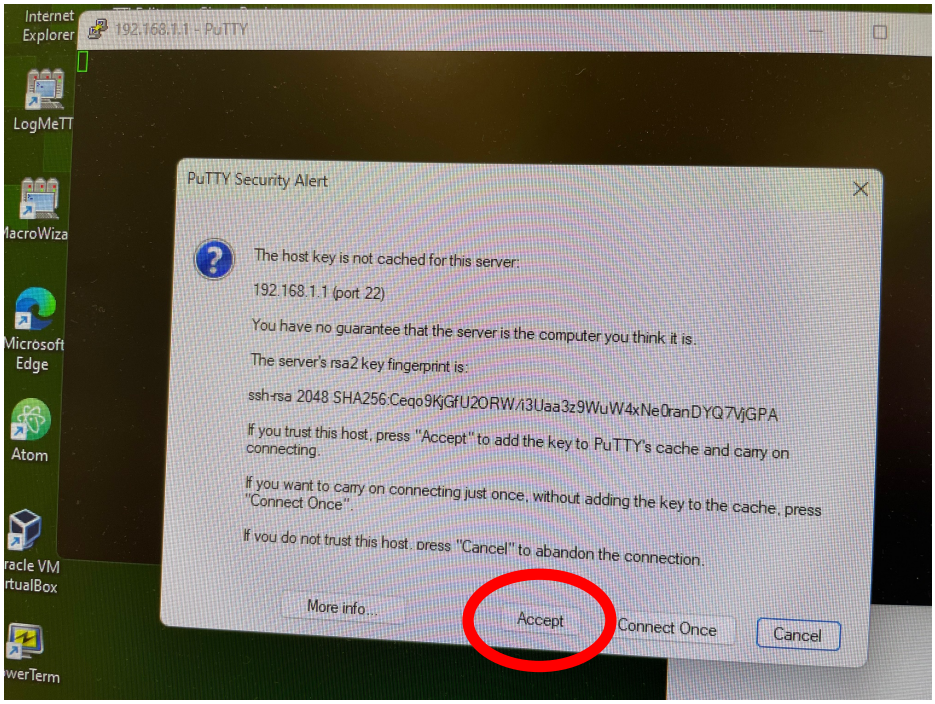
Galway Pod: Do a similar configuration on Galway's Router.

Use a new Putty window to test connectivity from PC-A to the default gateway of Dublin router.

As shown in the diagram below, add the IPv4 address and ensure the well-known port of 22 (SSH) is selected.



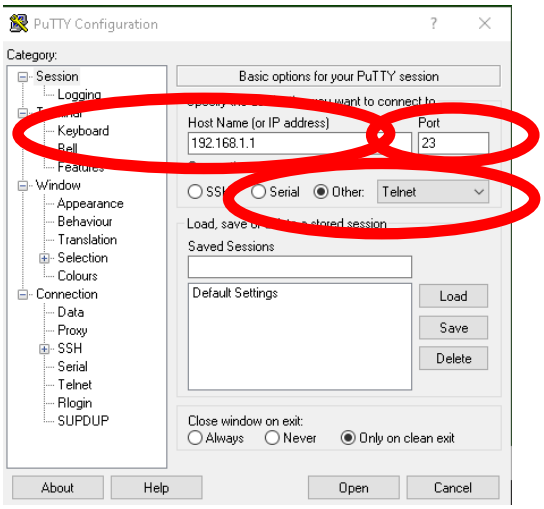
Note: when you connect using **SSH** you will likely receive something similar to the screenshot (below) the first time you connect. This shows that the host key is not cached for this server. Proceed by clicking the Accept button.



You should now be prompted for both a username and password.
Fill in the credentials and you should now see you are at the Dublin prompt.
Well done you have **securely** logged in over the network via the **SSH** protocol.

You now need to attempt to **Telnet** from PC-A to the Dublin router.

For Dublin Pod: Now you will **telnet** from the PC to your local default gateway(Dublin router). On the Desktop PC, open a new **Putty** window. Type in the address of the default gateway (e.g. diagram shows connecting to Dublin router). Ensure to select **Other** radio button and **Telnet**. Ensure that the port is **23**. This is the well-known port for telnet traffic.



This should then prompt you to login with the username and password you set previously. Note: you should be logged in over an **insecure** method – telnet.

TESTING:
The Dublin Pod should now be able to login BOTH using **Telnet and SSH** from PC-A to the Dublin router.

Also, the Galway Pod need to run the following tests:

Using Putty: SSH from PC-B to Galway router (note the Galway team would need to have configured remote access Part 1 step 3 for this to be successful). Only change to make is to configure the hostname as **Galway**.

Also:
Using Putty: Telnet from from PC-B to Galway router (again, note the Galway team would need to have configured remote access Part 1 step 3 for this to be successful).

The Galway POD team, can now using Putty, attempt to telnet (and SSH) from PC-B (Galway's PC) to the Dublin router – this should also be successful.

The Dublin POD team, can now using Putty, attempt to telnet (and SSH) from PC-A (Dublin's PC) to Galway router – this should also be successful.
This is IMPORTANT to test that Galway Pod can access the Dublin router remotely and Dublin Pod can access the Galway router remotely. Run tests using Putty to

connect over telnet and another test using SSH. Once you have connected successfully, disconnect (close the connection) to the remote sites router.

QUESTION 3 After completing all of section Part 1: Step 3: were you able to connect via Telnet and SSH to your respective Dublin and Galway routers?

True – Yes, we were able to telnet and SSH successfully

No – Unable to get to this point in the lab activity

Part 2: ACLs

Task 1: Configure a standard ACL proposed by the organization

Restrict remote access to Dublin and Galway's routers to only local PC's by configuring TWO standard named IPv4 ACLs.
Tip: You will need to create TWO standard named IPv4 ACLs (one on Dublin Router and one on Galway router)

Task 1
The organization has recently decided to restrict traffic using **standard named IPv4 ACLs.** As the networking engineering team, it is your responsibility to configure standard named IPv4 ACLs to restrict remote access to Dublin's router from PC-A only.

In addition, Galway's router should only be remotely available via Galway's PC-B only. (see Network Topology Diagram).

Specific requirements

Dublin POD: You need to configure and apply a **named standard IPv4 ACL** to **restrict remote access** (192.168.1.2 only) **to the Dublin router**. Permit only PC-A to access Dublin router remotely using 192.168.1.2 ONLY. All other remote access attempts from other PC's should be denied.

Galway POD: You need to configure and apply a named standard IPv4 ACL to restrict remote access (192.168.2.2 only) to Galway's router. Permit only Galway's PC-B to access R2 remotely using 192.168.2.2 only. All other remote access attempts from other PC's should be denied.

Further instruction
Use the access-list name **REMOTE-ACCESS** to restrict traffic.
Note: Before you attempt the task, a recommended approach is the following:
Test Telnet and SSH access on both Galway and Dublin routers from PC-A and PC-B.

You should be able to login in remotely (from both PC's to each router).
Note: these tests should have been successful from Part 1: step 3 above.

IMPORTANT:
Once you have created your ACL on Dublin, ensure to apply it on the appropriate interface.

Note: the Galway Pod will also need to configure ACL and apply it on the appropriate interface.

TESTING:
Use Putty:
Test remote access from PC-A, Dublin PC to Dublin Router (192.168.1.1)– **this should be successful.**

Test remote access from PC-B, Galway PC to Dublin Router (192.168.1.1)– **this should be unsuccessful** – this is how you can verify the correct operation of the ACL.

Use a similar process for Galway's router and ACL.

Use Putty:
Test remote access from PC-B, Galway PC to Galway's router (192.168.2.1)– **this should be successful.**
Test remote access from PC-A, Dublin PC to Galway's router – **this should be unsuccessful.**

QUESTION 4 Select the correct statement you configured in your router under the line vty 0 4 section.

Task 2 - Configure, Apply and Verify an Extended Named IPv4 ACL

All four members are expected to work together on this challenge

The Chief Technology Officer has asked you to use an **extended ACL** as they provide a greater degree of control.

They have the following requirements which they would like implemented in **one** ACL only.

For testing, they wish to deny all ping (ICMP) traffic from the **Dublin PC to the Galway PC-B only**.

However, they wish to allow all other traffic from the Dublin LAN to the Galway PC.

Create an Extended Named IPv4 ACL called BLOCK_ ICMP_ONLY and place it closest to the source of the traffic.

Between both pods, collaborate with each other to abide by best practices on where to implement this ACL.

TESTING
Once implemented:
ICMP traffic should be **unsuccessful** from PC-A to Galway's PC-B, however, ICMP traffic to other destinations such as from PC-A to Galway's router e.g. 192.168.2.1 should still work.
There should be a dedicated rule to allow all other traffic (HINT: IP traffic).
In addition, you could change the IP address of the Dublin PC to another address e.g. 192.168.1.3/24 and ping the Galway PC and this should be **successful**. Change the IP address back to 192.168.1.2 when finished testing.

Verify this with your lecturer.

QUESTION 5 Your lecturer will provide you with a code for successfully completing this Extended Named ACL and verifying it works as expected.

Please enter this code exactly as provided. Failure to enter the code exactly as given to you will result in 0%. Please note, this code is case sensitive.

Part 3:

Use Wireshark to identify traffic (both cleartext and encrypted protocols)

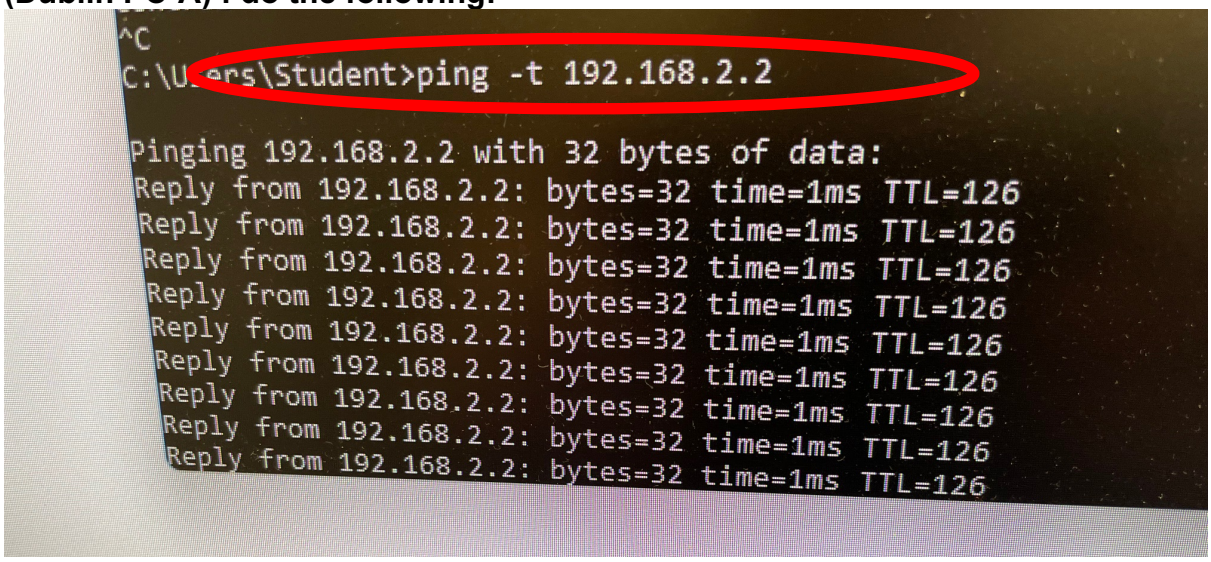
Background / Scenario
Wireshark is a software protocol analyzer, or "packet sniffer" application, used for network troubleshooting, analysis, software and protocol development, and education. As data streams travel back and forth over the network, the sniffer "captures" each protocol data unit (PDU) and can decode and analyze its content according to the appropriate RFC or other specifications.
Wireshark is a useful tool for anyone working with networks and can be used for data analysis and troubleshooting. In this part, you will use Wireshark to capture ICMP packets and Telnet and SSH traffic.

Remove the ACL you created in Task 2.

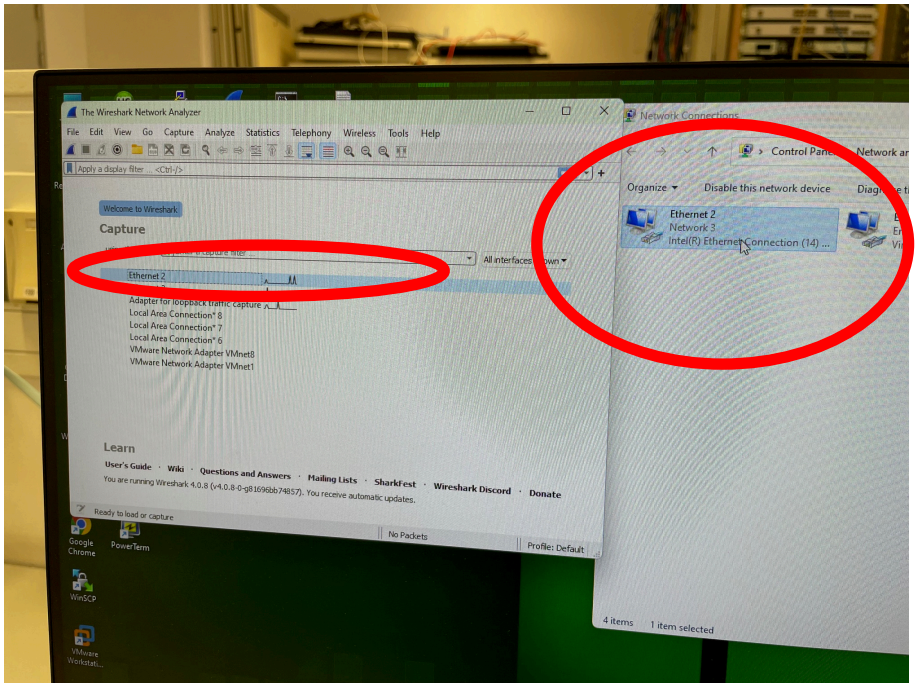
Part 1: Identifying ICMP traffic

Dublin Pod: **Setup a continuous ping from Dublin PC-A to Galway PC-B. These pings should now succeed.**

From command prompt on PC-A:
(Dublin PC-A) : do the following:

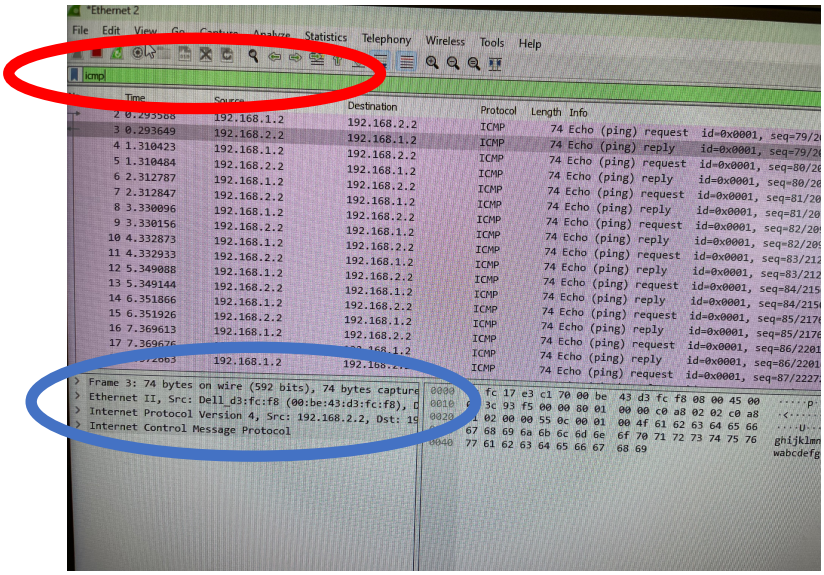


Open Wireshark on Galway's PC. Select the appropriate interface to sniff traffic (based on your PC setup).
Note, in my case below this interface is Ethernet 2.
Important: it will NOT be any Virtual interface.



To open Wireshark and to start sniffing on Ethernet 2, double click.

Use Wireshark to **filter** and to observe the **ICMP traffic**. Type **icmp** and press **enter** to filter for ICMP traffic only.
Example output shown. Notice how I've filtered for ICMP traffic only. If you wish to view other traffic delete the icmp filter.
Note how we can see ping traffic (requests and replies.)



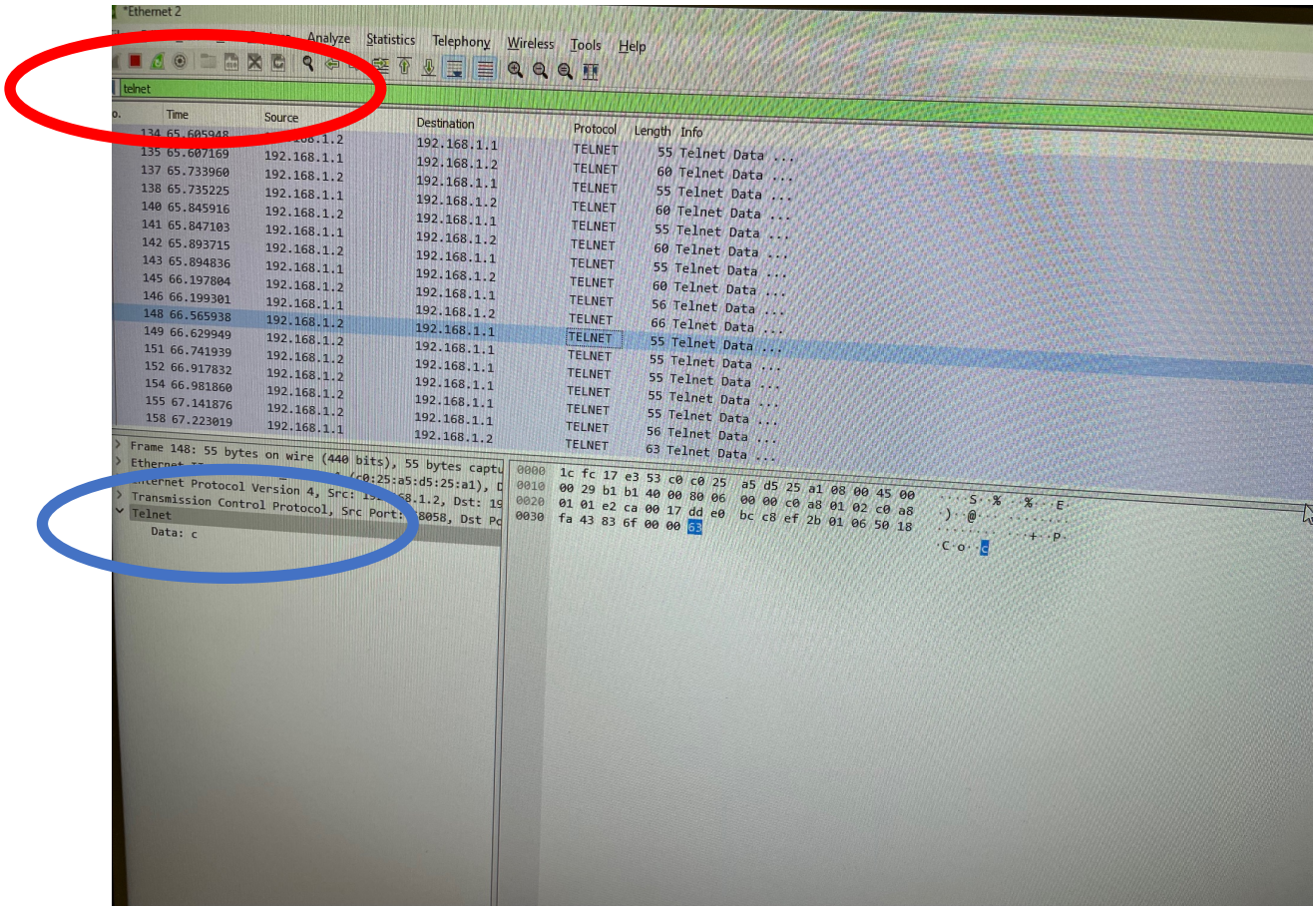
At the bottom of the output, note the TCP/IP layers (indicated with blue outline above). **Observe each layer – are you able to explain what is happening at each?**

To end the ICMP continuous ping on the Dublin PC, press Ctrl + C in the command prompt window.

Part 2: Identifying Telnet traffic

Dublin Pod: On PC-A, use Putty to telnet into Dublin Router.
Galway Pod: On PC-B, use Putty to telnet into Galway Router.

Dublin Pod: Open Wireshark and ensure to select the Ethernet adapter to sniff traffic. Create a filter for **telnet** traffic only.



Galway Pod: you will just need to change the filter from icmp to telnet.

Select one of the Telnet packets in the top pane, then in the bottom pane (indicated with blue outline) collapse the **Telnet** option to display the **Data**:
View different packets using your cursor keys. Currently I'm selecting 148, by moving down I will select packet 149 etc. (this number will likely be different on your output!)
Can you find the credentials sent between PC and router?

QUESTION 6 Observing Telnet traffic – filtering the output with telnet, from PC to router in Wireshark, does this show the packet data (text) in clear text?

True- Yes , the password is visible in clear text

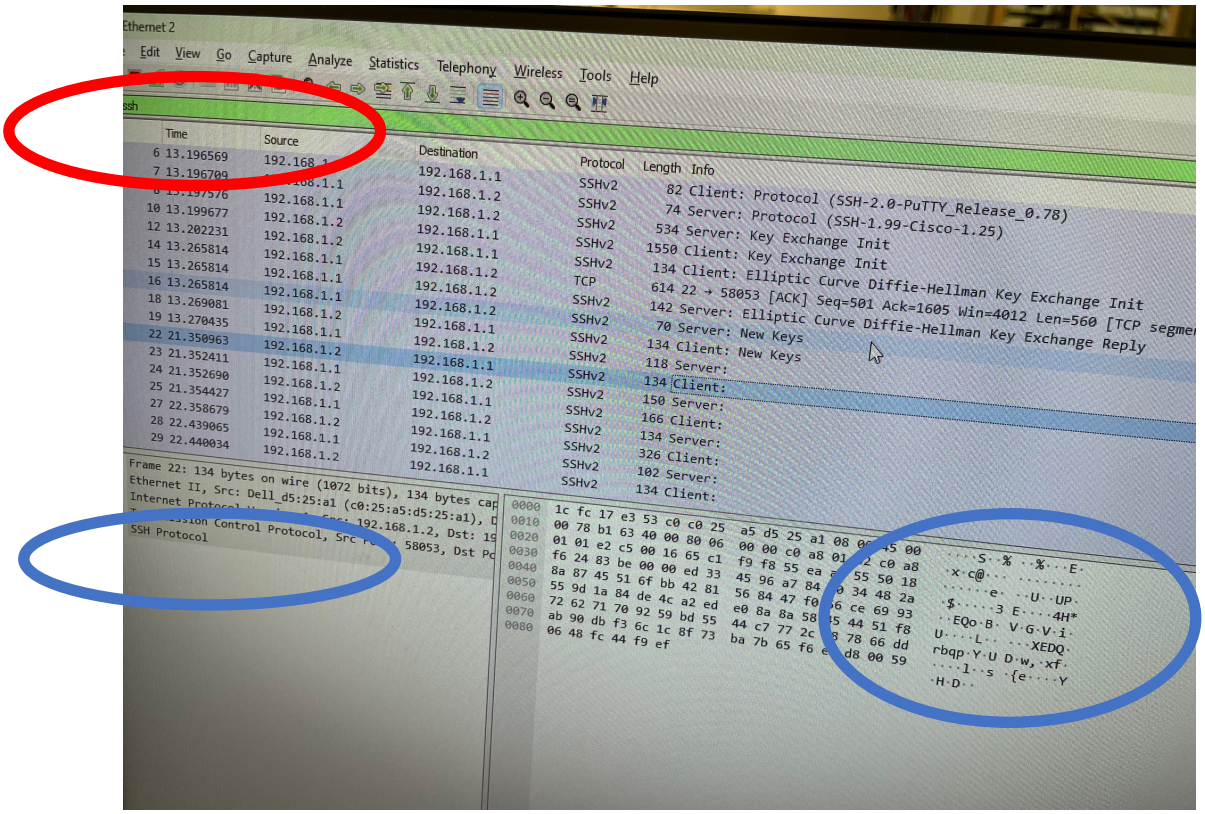
False – no, password is not visible in clear text

Close the Putty Telnet session.

Part 3: Identifying SSH traffic

Both Galway and Dublin Pods:
Create a **new Putty session** but this time create an **SSH** connection.
Dublin Pod: open up Putty and create a SSH connection to Dublin's router.
Galway Pod: open up Putty and create a SSH connection to Galway's router.

Filter on Wireshark for SSH traffic.
Use the **ssh** keyword in the filter toolbar on Wireshark. **Select an SSH packet and check to observe in the SSH part to see if any clear text passwords are visible.**



QUESTION 7 Now observing SSH traffic, filtering the output with ssh from PC to router, does Wireshark output show packet data (text) in clear text?

True- All passwords are visible in clear text

False – Passwords are not visible in clear text – they are encrypted

Once complete, submit your quiz on Brightspace.

After successful completion of Part 3, at this point have completed this weeks WAN lab. Well done! Please ensure to do the final important step below – erase your router.

Part 4:

Erase router configuration & revert PC Ethernet adapter settings back to default (enable DHCP)

Final task – All pods
Disconnect all cables connected to routers.

Clear configuration:
Erase your router configuration from the lab
To clear the configuration, issue the erase start command. Confirm your intentions when prompted, and answer "no" if you are asked to save changes. The result should look something like this:

- Dublin>enable
- Dublin#**erase startup-config**

Erasing the nvram filesystem will remove all files! Continue?
[confirm]
[OK]
Erase of nvram: complete
When the prompt returns, issue the reload command. Confirm your intentions when prompted.

- Dublin#**reload**

If you get the following:
System config has been modified. Save? **No**
Proceed with reload [Press enter]

Once the router boots, it may show the following prompt:

Would you like to enter the initial configuration dialog? NO

You should now have the following prompt,
Router>

At this point you know that you have cleared the NVRAM of the router – Well done!