

Network segmentation and access control implementation

In this document, I'll be going through setting up a network in **cisco packet tracer**, consisting of a router, a switch and 4 end devices.

Part 1. Switch-End Devices Connection

First step will be to add all devices to the workspace

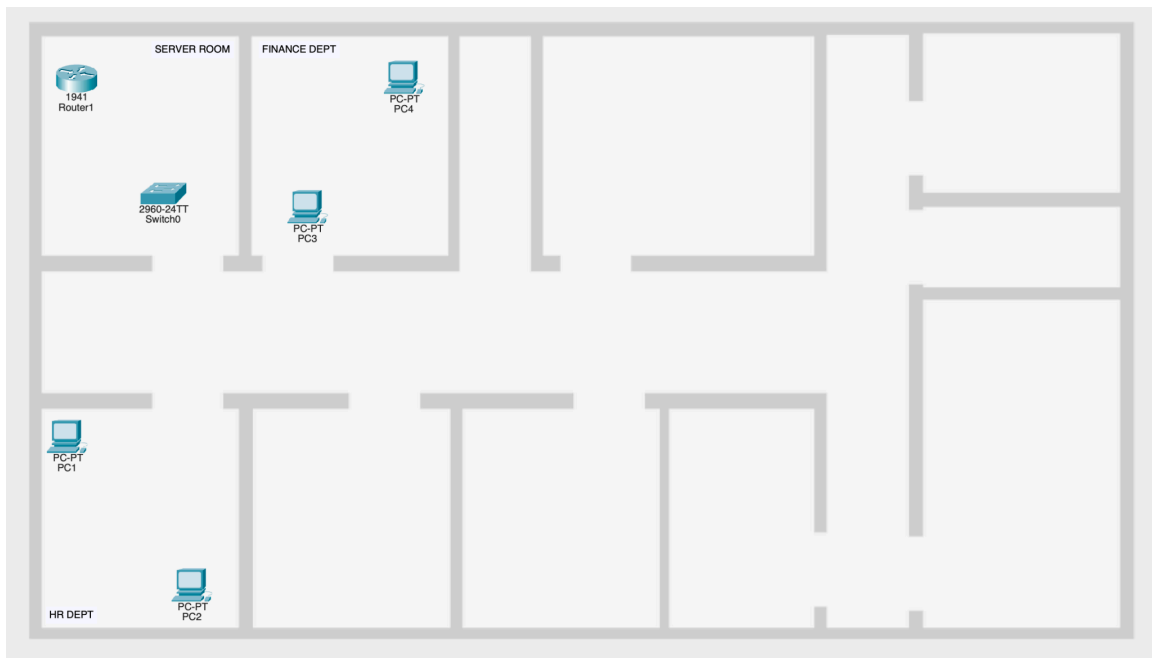


Fig 1.1 *devices placed onto logical workspace*

I made use of a background image for organisation and visualisation purposes

Next step is **connection**

- using a **copper straight through cable**:

- connect each of the PCs to the switch. Use the **FastEthernet** port i.e. FastEthernet0 (on end devices) and FastEthernet0/1,2,3&4 (on the switch).
- connect the switch to the router. Use the **GigabitEthernet** port i.e. GigabitEthernet0/0 (on the router) and GigabitEthernet0/1 (on the switch).

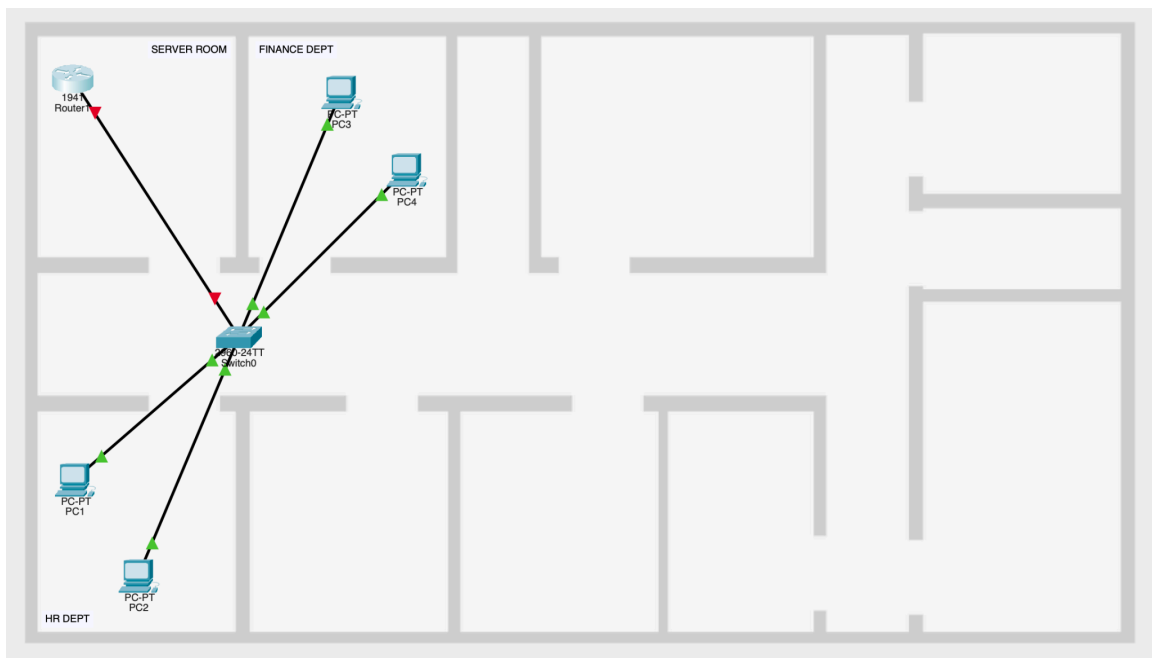


Fig 1.2 *devices connected*

Here i made some positional changes to the devices as i discovered i couldn't make the cable L-shaped/T-shaped

It is important to take note of the connection between the Switch and the Router (we will talk about this later).

Now, creating the **VLANs**(HR & Finance departments) and assigning end devices to their respective vlans.

in the switch CLI:

Create VLANs (10 & 20)

Enter into privilege exec mode

```
enable
```

Enter into global configuration mode

```
configure terminal
```

Create Vlan 10

```
vlan 10
```

Name Vlan 10

```
name HR
```

Exit and repeat last two steps for Vlan 20(NB: name = finance)

```
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#vlan 10
Switch(config-vlan)#name HR
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#name Finance
Switch(config-vlan)#exit
Switch(config)#
```

Fig 1.3 *vlan creation*

Once done we can check to see if the vlans have been created using the command

```
show vlan brief
```

```
Switch#show vlan brief
VLAN Name                Status Ports
-----
1    default                active Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                   Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                   Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                   Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                   Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                   Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                   Gig0/1, Gig0/2
10   HR                     active
20   Finance                 active
1002 fddi-default           active
1003 token-ring-default     active
1004 fddinet-default         active
1005 trnet-default           active
Switch#
```

Fig 1.4 *vlan configuration*

Looking at the vlan configuration we can see we have vlan IDs 1, 1002, 1003, 1004 & 1005. These are all default vlans, which means we can't use them.

But we also have 10 & 20 which we created, named **HR** and **Finance** respectively.

On looking closer, we see that all the available ports are currently assigned to **vlan 1** meaning all ports can communicate freely with each other right now without any extra configuration. The aim of this task is to segment networks in order to improve security within a system. In the next steps, we would look at how we can do this.

What we need to do now is assign the devices(ports) to the appropriate vlans.

PC1(Fa0/1) & PC2(Fa0/2) -> vlan 10

PC3(Fa0/3) & PC4(Fa0/4) -> vlan 20

Still in config mode in the switch cli:

We have to select the interface(port) for configuration

```
interface fastEthernet0/1
```

This selects **Fa0/1**.

We now have to set the port to **access mode**. Here we can configure the port to carry traffic within a single vlan

```
switchport access mode
```

Next, we assign the port to the appropriate vlan

```
switchport access vlan 10
```

Run this process for all other ports(Fa0/2, Fa0/3 & Fa0/4).



Fig 1.5 assigning devices(ports) to vlans

Now we can check our vlan config to see the which ports are assigned to each vlan.

```
Switch#show vlan brief
```

VLAN	Name	Status	Ports
1	default	active	Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Fa0/24 Gig0/1, Gig0/2
10	HR	active	Fa0/1, Fa0/2
20	Finance	active	Fa0/3, Fa0/4
1002	fdci-default	active	
1003	token-ring-default	active	
1004	fdnet-default	active	
1005	trnet-default	active	

Switch#

Fig 1.6 vlan configuration with newly assigned ports

Part 2. Switch-Router connection

In **Fig 1.2** we can see that the connection signal between the switch and the router is red, while the others are green. By default the router port starts in a shutdown state, that's why a connection can't be established.

So we have to stop the port from shutting down.

In router cli:

Enter into *privilege exec mode*

Enter into *global configuration mode*

Select the port

```
interface gigabitEthernet0/0
```

Set the port to always stay on

```
no shutdown
```

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface gigabitEthernet0/0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
```

Fig 2.1 turn on router port

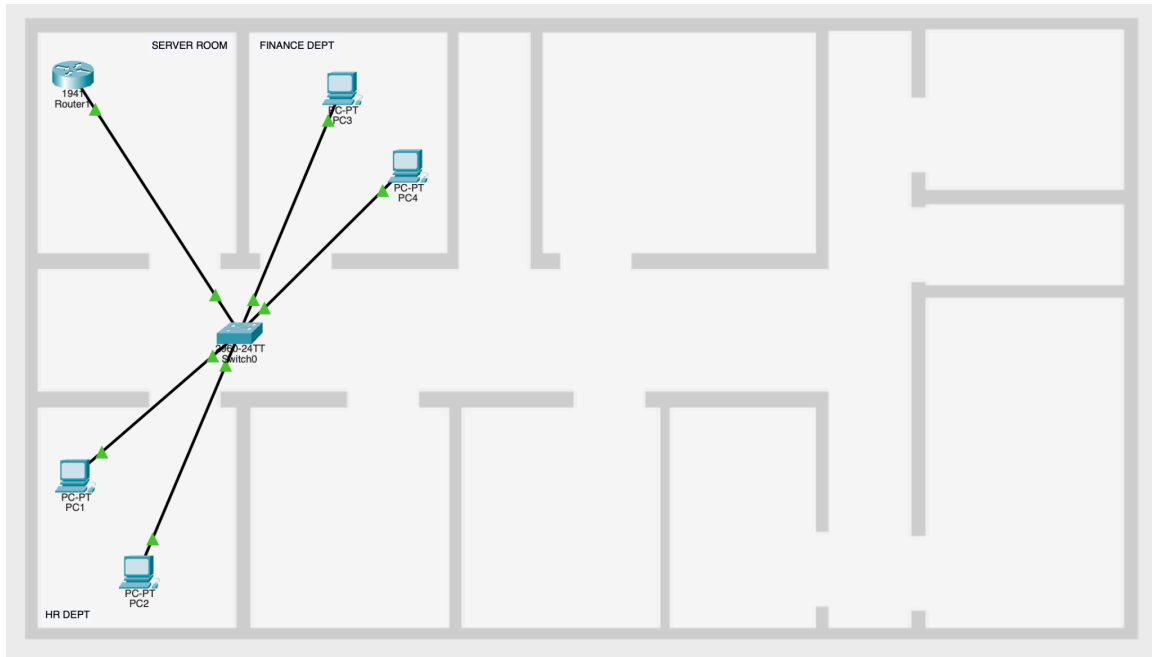


Fig 2.2 *router and switch connection active*

Now that we've initiated the connection between the switch and the router, we need to ensure that the router can perform its function properly, of routing traffic between vlans. In other words, allowing devices in different vlans to communicate (in this case).

We can do this by configuring the corresponding port on the switch (gigabitEthernet0/1) as a **trunk**.

Doing this tells the switch that the port will be used to carry traffic for multiple vlans and not just one unlike an access port which carries traffic for a single vlan.

In switch cli:

Enter into privilege exec mode

Enter into global configuration mode

Select the port

```
interface gigabitEthernet0/1
```

Configure switch port as a trunk

```
switchport mode trunk
```

```
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#interface gigabitEthernet0/1
Switch(config-if)#switchport mode trunk
Switch(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up
```

Fig 2.2 *configure switch port as a trunk*

Now we need to create & configure the sub-interfaces for both vlans on the router.

These sub interfaces are part of the traffic-handling process.

Each of the sub interfaces will handle traffic for one vlan.

In router cli:

Enter into privilege exec mode

Enter into global configuration mode

Select the port(sub-interface)

```
interface gigabitEthernet0/0.10
```

.10 refers to vlan 10

Enable **encapsulation**(this is a tagging method that keeps traffic from different vlans separate)

```
encapsulation dot1Q 10
```

Assign an IP address to the sub-interface

```
ip address <ip address> <subnet mask>
```

Repeat the process for vlan 20(...0/0.20).

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface gigabitEthernet0/0.10
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0.10, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0.10, changed state to up
Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip address 192.168.10.1 255.255.255.0
Router(config-subif)#exit
Router(config)#interface gigabitEthernet0/0.20
Router(config-subif)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0.20, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0.20, changed state to up
Router(config-subif)#encapsulation dot1Q 10
%Configuration of multiple subinterfaces of the same main
interface with the same VID (10) is not permitted.
This VID is already configured on GigabitEthernet0/0.10.
Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip address 192.168.20.1 255.255.255.0
Router(config-subif)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#write memory
Building configuration...
[OK]
Router#
```

SUB-IF FOR VLAN 10

SUB-IF FOR VLAN 10

here you see it's not possible to tag multiple vlans in the same sub-interface. Thus keeping traffic separate for each vlan

Fig 2.4 sub interfaces configured and ip addresses assigned

At this point our router is set to route traffic correctly between both vlans.

But first we have to assign ip addresses to each end device...

Part 3. Assigning ip addresses to end devices

To assign ip addresses for each end device, click on the device and head over to the desktop tab and then click on ip configuration.

Here we would assign a static ip address to our end devices as seen below:

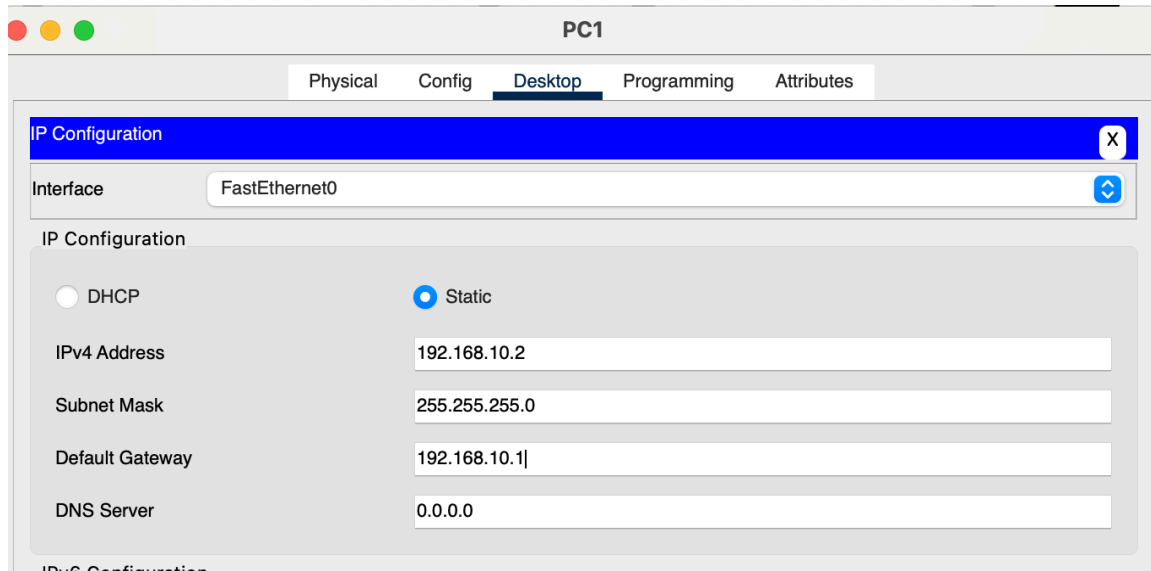


Fig 3.1 PC1 ip configuration

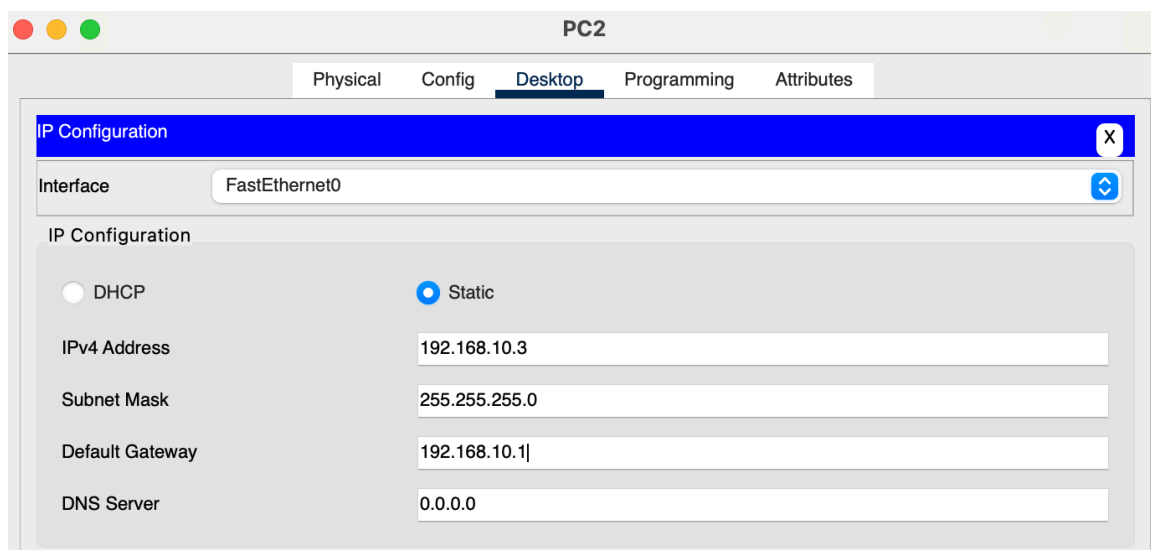


Fig 3.2 PC2 ip configuration

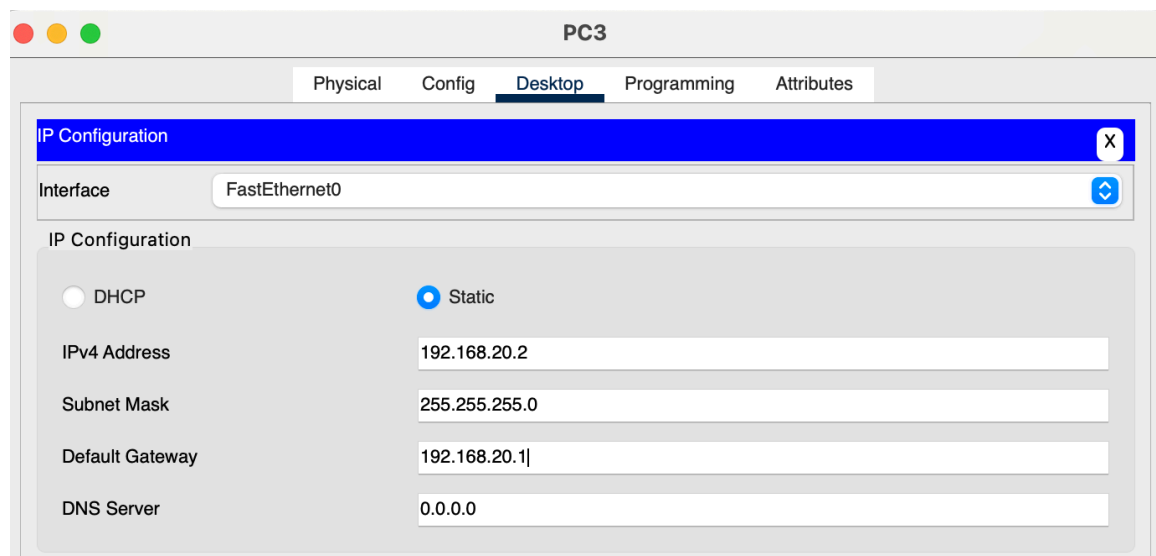


Fig 3.3 PC3 ip configuration

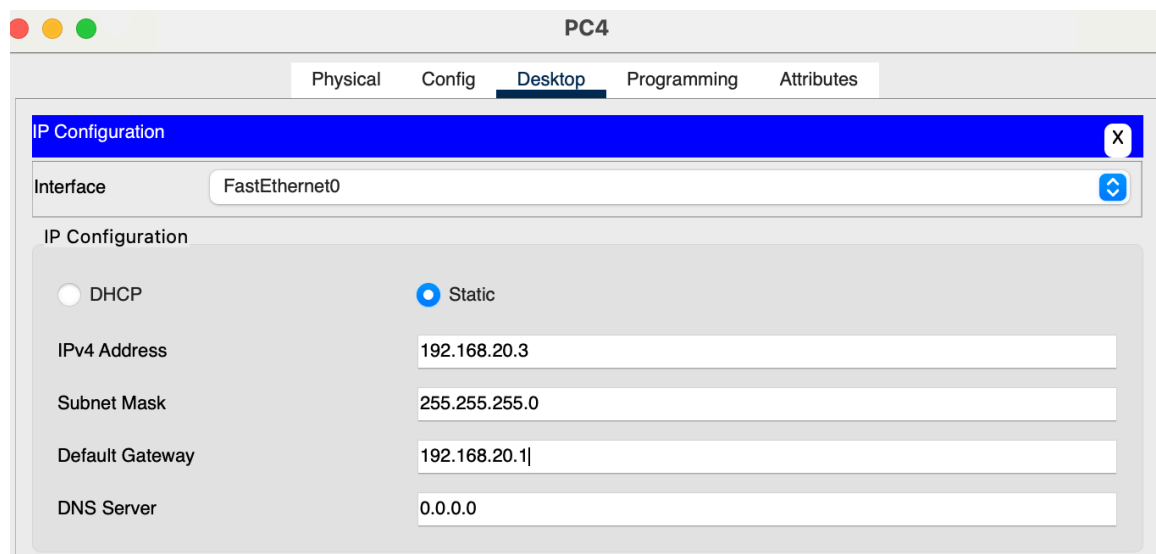


Fig 3.4 PC4 ip configuration

Part 4. Testing Connectivity

- PC1 (HR) must be able to ping PC3 (Finance) and vice versa.
- Take a screenshot of the ping results.

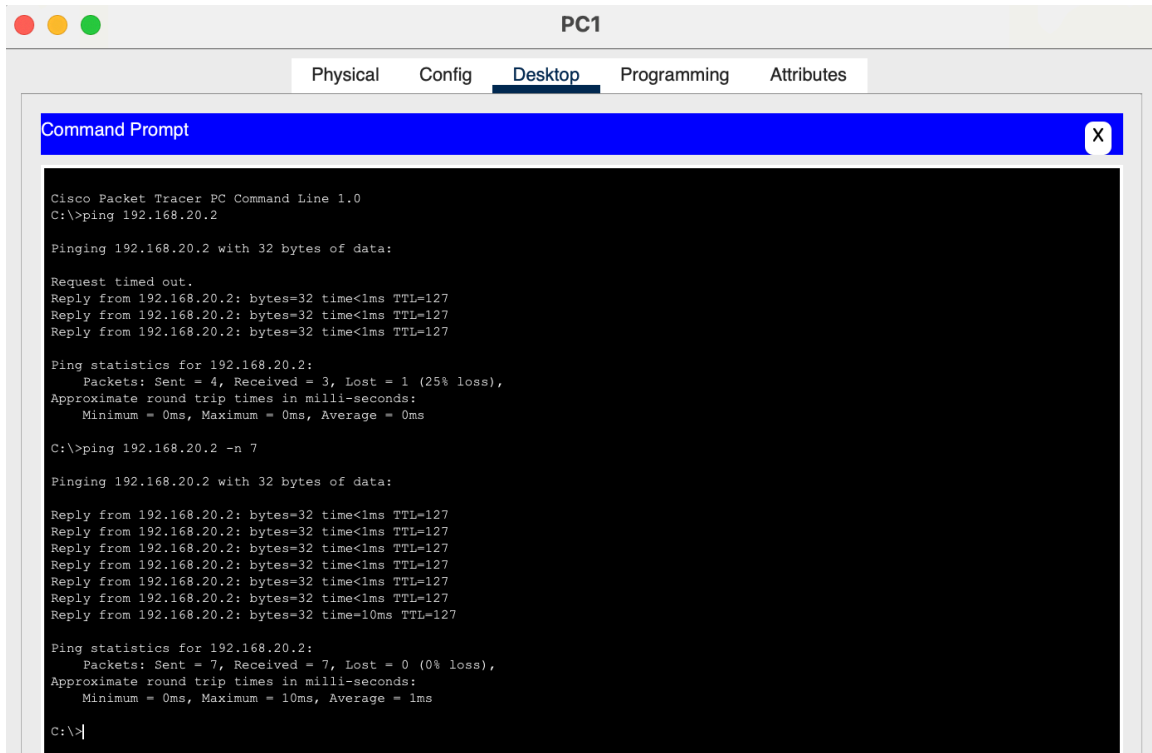


Fig 4.1 PC1 pinging PC3

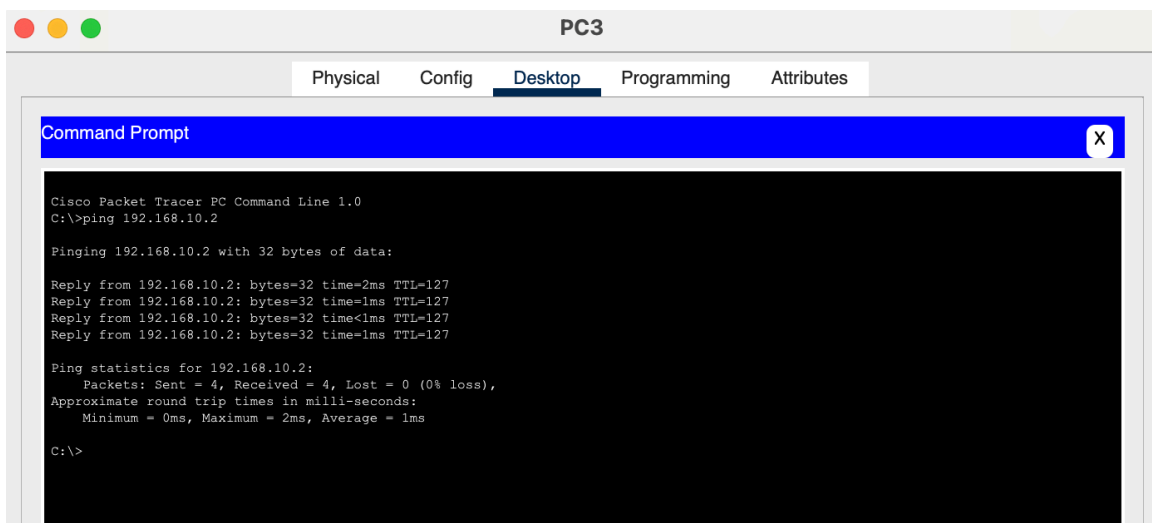


Fig 4.2 PC3 pinging PC1

Visualisation of the process

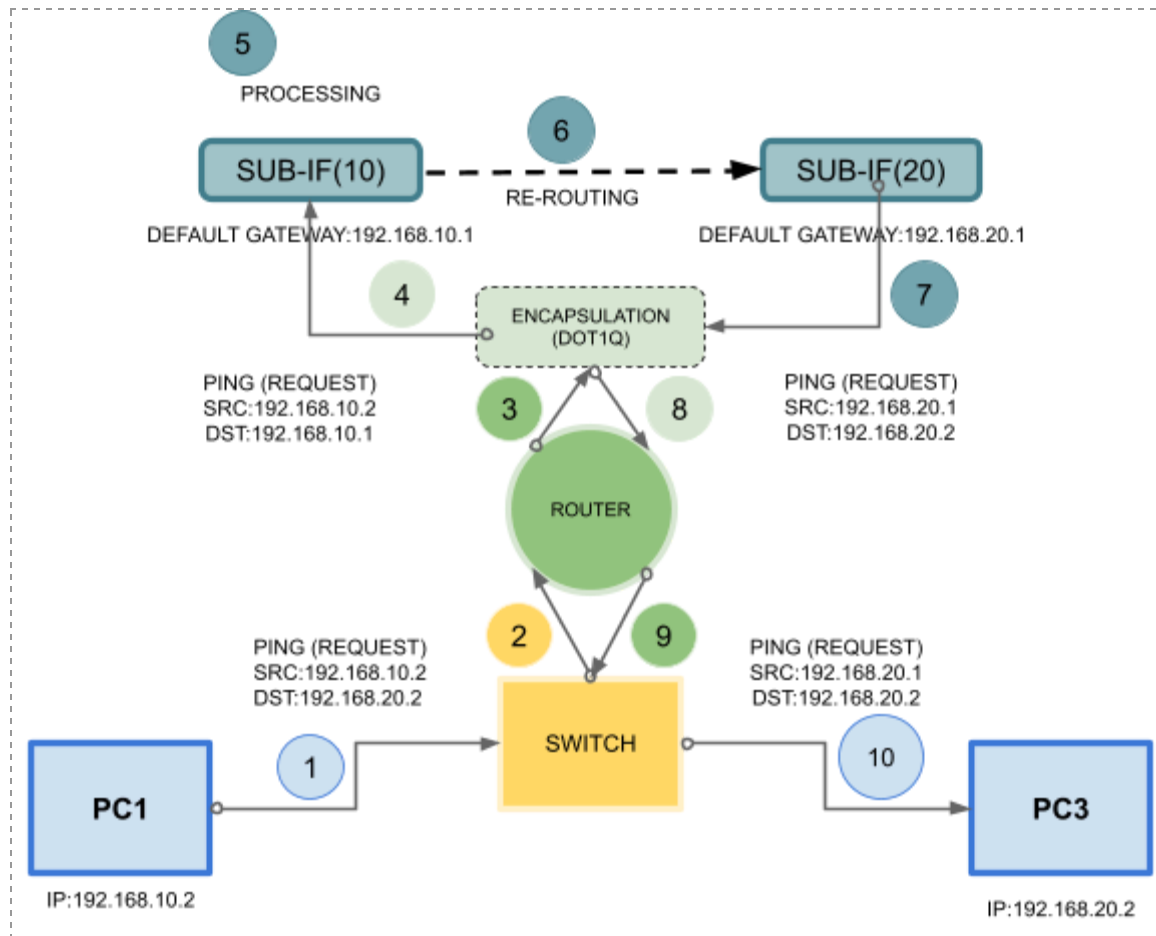


Fig 4.3 ping request visualisation

Please note that **encapsulation** and **re-routing** from vlan 10 sub-interface to vlan 20 sub-interface happens inside a router. I only have them separate here for illustration purposes

Part 5. ACL Rule

- Create an ACL rule on the router that blocks all HTTP traffic (port 80) from Finance (VLAN 20) to HR (VLAN 10).

An **ACL rule** acts as a filter that accepts or denies traffic based on rules we set.

The rule takes parameters like acl number, protocol type, source ip address, destination ip address, port number.

This rule will be applied to the vlan 20 sub-interface on the router.

In router cli:

Enter into *privilege exec mode*

Enter into *global configuration mode*

Define ACL Rule

```
access-list 100 deny tcp 192.168.20.0 0.0.0.255 192.168.10.0  
0.0.0.255 eq 80
```

access-list 100 - ACL number for extended ACLs that allows source ip and destination ip parameters.

deny - sets the rule to deny.

tcp - protocol type to deny.

192.168.20.0 - source ip.

192.168.10.0 - destination ip.

0.0.0.255 - wildcard mask. This refers to all ips on the vlane.g. from

192.168.10.1-192.168.10.254 (vlan 10).

eq 80 - port number (http).

Then we have to permit all other ip addresses because there is an implicit deny all rule at the end of every ACL.

There's actually two ways we can go about this according to the question

- We can permit all ip traffic e.g tcp, icmp, udp OR
- We can permit just icmp traffic as that is what we need to perform ping requests (let's go with this).

```
access-list permit icmp any any
```

```
Router>enable
Router#configure terminal
^
% Invalid input detected at '^' marker.
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#access-list 100 deny tcp 192.168.20.0 0.0.0.255 192.168.10.0 0.0.0.255 eq 80
Router(config)#access-list permit icmp any any
^
% Invalid input detected at '^' marker.
Router(config)#access-list 100 permit icmp any any
Router(config)#
```

Fig 5.1 *acl creation*

If we check our access-lists we can see the rules we've just specified being applied to access-list 100.

```
Router#show access-lists
Extended IP access list 100
 10 deny tcp 192.168.20.0 0.0.0.255 192.168.10.0 0.0.0.255 eq www
 20 permit icmp any any
Router#
```

Fig 5.2 *show access-list*

Now, we have to apply this ACL rule to the sub-interface for vlan 20.

Select the port(sub-interface) for vlan 20

```
interface gigabitEthernet0/0.20
```

Apply the rule to outgoing traffic

```
ip access-group 100 out
```

```
Router(config)#interface gigabitEthernet0/0.20
Router(config-subif)#ip access-group 100 out
Router(config-subif)#
```

Fig 5.3 *apply ACL rule*

Part 6. Testing Firewall Rules

- Try accessing a web page from PC3 (Finance) to PC1 (HR) and ensure the request is blocked.

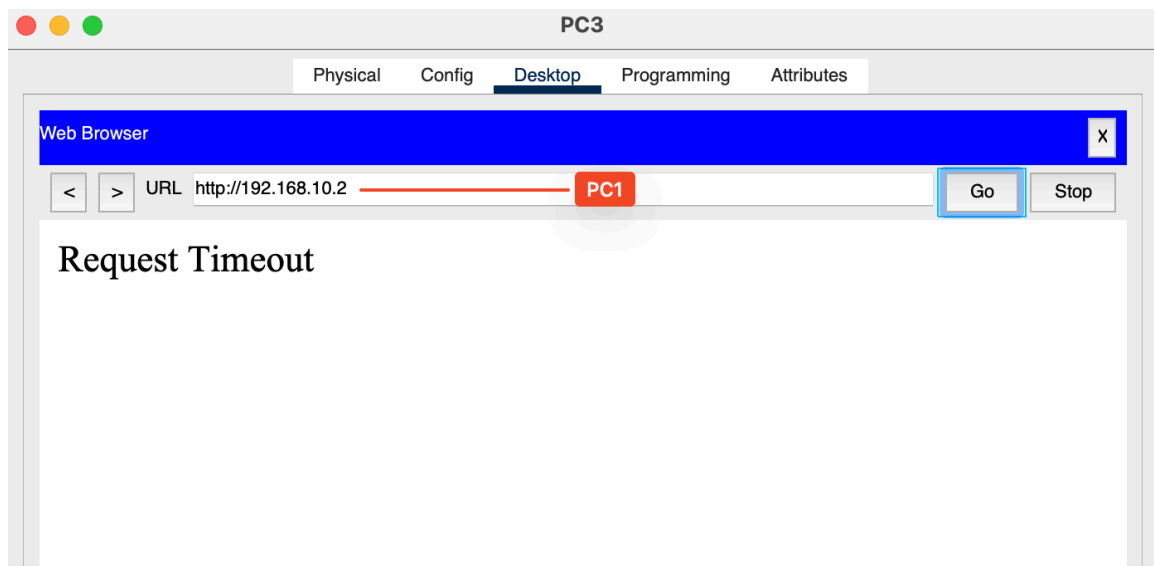
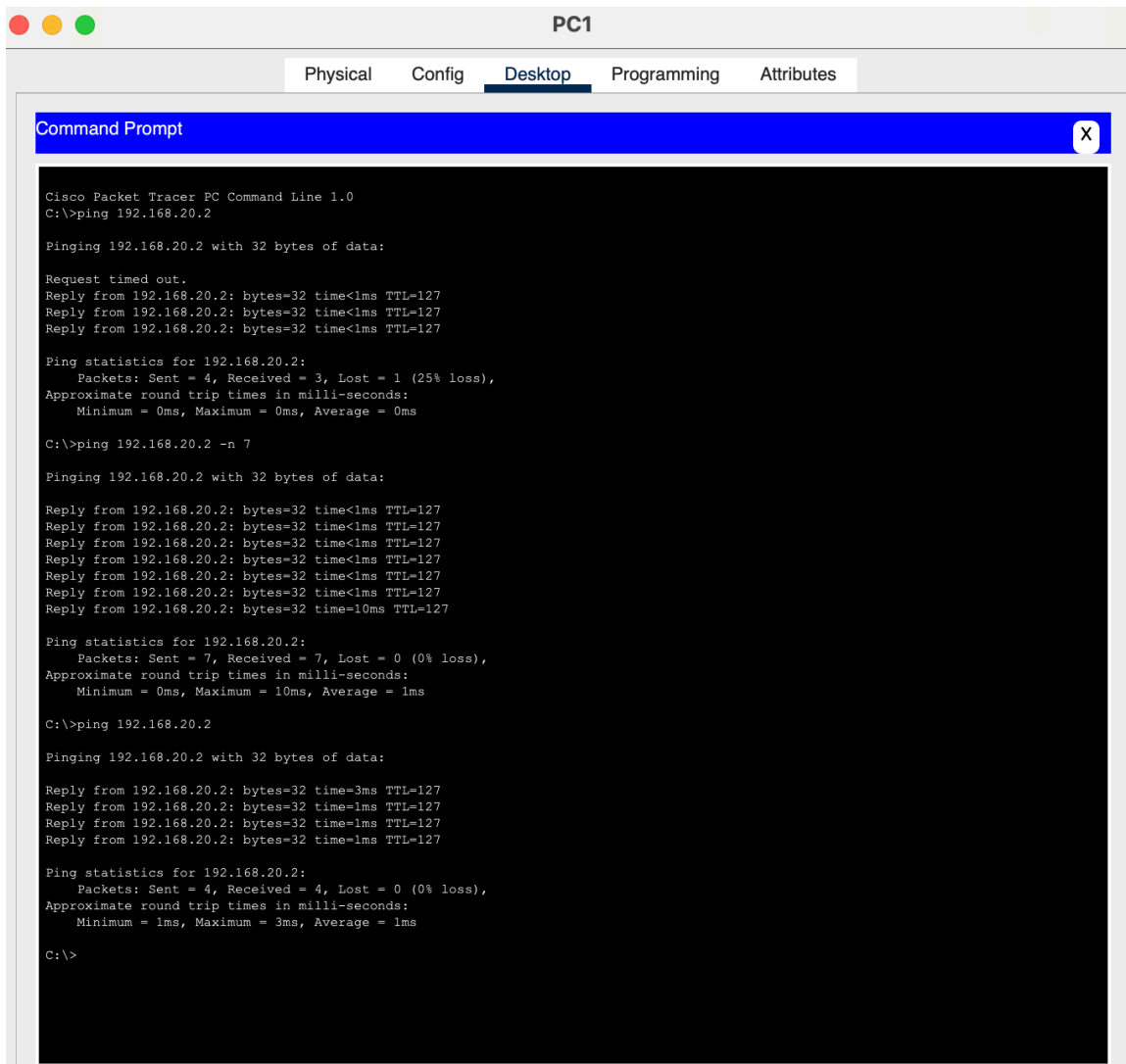


Fig 6.1 *PC3-to-PC1 web page test*

- Try pinging PC1 from PC3 and ensure ICMP is still allowed.



The screenshot shows a Cisco Packet Tracer PC interface for PC1. The 'Desktop' tab is selected, displaying a 'Command Prompt' window. The command prompt shows the execution of three ping commands to the IP address 192.168.20.2. The first command shows a 25% loss (1 out of 4 packets received). The second command, using the -n 7 flag, shows 0% loss (7 out of 7 packets received). The third command shows 0% loss (4 out of 4 packets received).

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.20.2

Pinging 192.168.20.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.20.2: bytes=32 time<1ms TTL=127
Reply from 192.168.20.2: bytes=32 time<1ms TTL=127
Reply from 192.168.20.2: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.20.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 192.168.20.2 -n 7

Pinging 192.168.20.2 with 32 bytes of data:

Reply from 192.168.20.2: bytes=32 time<1ms TTL=127
Reply from 192.168.20.2: bytes=32 time<1ms TTL=127
Reply from 192.168.20.2: bytes=32 time<1ms TTL=127
Reply from 192.168.20.2: bytes=32 time<1ms TTL=127
Reply from 192.168.20.2: bytes=32 time<1ms TTL=127
Reply from 192.168.20.2: bytes=32 time<1ms TTL=127
Reply from 192.168.20.2: bytes=32 time=10ms TTL=127

Ping statistics for 192.168.20.2:
    Packets: Sent = 7, Received = 7, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 10ms, Average = 1ms

C:\>ping 192.168.20.2

Pinging 192.168.20.2 with 32 bytes of data:

Reply from 192.168.20.2: bytes=32 time=3ms TTL=127
Reply from 192.168.20.2: bytes=32 time=1ms TTL=127
Reply from 192.168.20.2: bytes=32 time=1ms TTL=127
Reply from 192.168.20.2: bytes=32 time=1ms TTL=127

Ping statistics for 192.168.20.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 3ms, Average = 1ms

C:\>
```

Fig 6.1 PC1-to-PC3 ping test