LAB THREE – STATIC ROUTING

In this lab you will work with four different network topologies. The topology for Parts 1-4 is shown in Figure 3.1. These parts address router configuration on Cisco Routers (how to work with Cisco’s IOS (Internet Operating System)) and setting static routes on PCs. The topology for Part 5 is shown in Figure 3.2. This topology is used to study the role of ICMP *route redirect* message. The topology for Part 6 is shown in Figure 3.3. There, you explore the relationship between network prefixes and IP forwarding. In Part 7 we walk you through a denial of service attack.

We have added 3 appendices to this lab that will **help** you with the experiment set ups. APPENDIX A is a brief summary of the most important Cisco IOS commands. Note the usefulness of using “?” when configuring your router.

APPENDIX B shows you how you can quickly input command lines in the console window of routers, PCs and VMs.

APPENDIX C illustrates how you can save a router configuration so that when you stop a GNS3 project and come back to it later, it will have stored all the values and settings for you.

A screenshot of a cell phone

Description automatically generated

Figure 3.1 Network Topology for Parts 1-4

|  |  |  |
| --- | --- | --- |
| **PCs** | **eth0** |  |
| PC1 | 10.0.1.11 / 24 |  |
| PC4 | 10.0.3.41 / 24 |  |
| **Cisco Router** | **FastEthernet0/0** | **FastEthernet1/0** |
| R1 | 10.0.1.1 / 24 | 10.0.2.1 / 24 |
| R2 | 10.0.2.2/24 | 10.0.3.1/24 |

Table 3.1 IP addresses for Parts 1-4

# **PART 1. Configuring a CISCO Router**

The setup of the Cisco router is more involved. There are different ways to connect to a Cisco router such as by the Serial or Ethernet ports or connections. The first step is to start the router in GNS3, and then open the console window so that the configuration commands can be entered. Once in the console you have to type IOS commands using the command line interface of IOS. The network setup for this part is as shown in Figure 3.1 and Table 3.1.

### **Exercise 1(A). Switching Cisco IOS Command Modes**

This exercise demonstrates how to log into a router and how to work with the different Cisco IOS command modes. It is important to understand the different modes so you know where you are and what commands are accepted at any time.

1. Connect the Ethernet interfaces of the PCs and the Cisco router as shown in Figure 3.1.
2. Right-click on router R1 and choose Start.
3. Right-click on router R1 and choose Console. Wait a few seconds until the router is initialized. If everything is fine, you should see the prompt shown below. This is the **User EXEC mode.** If the prompt does not appear, try to restart GNS3 and repeat the setup again.

**R1>**

1. To see which commands are available in this mode, type "?":

**R1> ?**

1. To view and change system parameters of a Cisco router, you must enter the **Privileged EXEC mode** by typing:

**R1>** enable

**R1#**

1. Type the following command to disable the Privileged EXEC mode and go back to EXEC mode.

**R1#** disable

**R1>**

|  |
| --- |
| **NOTE**  The Cisco routers in GNS3 sometimes start up in Privileged instead of User EXEC mode. There is no explanation as to why that happens. |

1. To modify system wide configuration parameters, you must enter the **global configuration mode**. This mode is entered from Privileged Exec mode by typing:

**R1#** configure terminal  
**R1(config)#**

|  |
| --- |
| **Tip:**  Almost all terminal commands can be reduced to shorter commands.  **Example:** configure terminal can be reduced to conf t |

1. To make changes to a network interface, you must enter the **interface configuration mode** from Global configuration mode with the command:

**R1(config)#** interface FastEthernet0/0  
**R1(config-if)#**

The name of the interface is provided as an argument. Here, the network interface that is configured is FastEthernet0/0.

1. To return from the interface configuration to the global configuration mode, or from the global configuration mode to the Privileged EXEC mode, use the exit command:

**R1(config-if)#** exit  
**R1(config)#** exit  
**R1#**

The exit command takes you one step up in the command hierarchy. To directly return to the Privileged EXEC mode from any configuration mode, use the end command:

**R1(config-if)#** end  
**R1#**

1. To terminate the console session you have to be in User EXEC mode, type logout or exit:

**R1>** logout  
**R1 con0 is now available  
Press RETURN to get started.**

**R1>** exit  
**R1 con0 is now available  
Press RETURN to get started.**

### **Summary of Modes:**

* User EXEC Mode: **R> enable** takes you down
* Privileged EXEC Mode: **R#> disable** takes you up**, conf t** takes you down
* Global Configuration Mode: **R(config)# exit** takes you up, **int Iface** takes you down
* Interface Configuration Mode: **R(config-if)# exit** takes you up
* **Exit** moves you up one level **in “#” modes**
* **End** in **“#” modes** takes you to Privileged EXEC mode **R#**

User Exec mode => Privileged Exec mode => Global Configuration mode => Interface Configuration mode

### **Exercise 1(B). Configuring a Cisco Router via the console**

The following exercises show the basic Cisco IOS commands that are used to configure a Cisco router.

1. Right-click on R1 and choose Start.
2. Right-click on R1 and choose Console. Wait some seconds until the initial console window is set up. When the router is ready to receive commands, proceed to the next step.
3. Configure R1 and R2 with the IP addresses given in Table 3.1. Below we show how to configure R1. Follow same steps for R2 with appropriate IP addresses.

|  |
| --- |
| **IOS MODE: GLOBAL CONFIGURATION Router(config)#**  ip routing  no ip routing  Enables or disables IP forwarding. For a router to forward packets, IP forwarding has to be enabled. By default “ip routing” is disabled in a router. Has to be enabled.  no cdp run  Disables Cisoc’s discovery protocol. These packets are sent out continuously by the routers and are not needed for our labs. **Recommended to turn off.**  **IOS MODE: INTERFACE CONFIGURATION Router(config-if)#**  no shutdown  shutdown  Enables or disables, respectively, a network interface.  **NOTE:** that when you enter any commands at any level of configuration (ie Router or Interface) you have to type “end” to enable the changes/additions you made. |

**R1>** enable

**R1#** configure terminal

**R1(config)#** no cdp run

**R1(config)#** no ip routing

**R1(config)#** ip routing

**R1(config)#** interface FastEthernet0/0

**R1(config-if)#** ip address 10.0.1.1 255.255.255.0

**R1(config-if)#** no shutdown

**R1(config-if)#** interface FastEthernet1/0

**R1(config-if)#** ip address 10.0.2.1 255.255.255.0

**R1(config-if)#** no shutdown

**R1(config-if)#** end

|  |
| --- |
| **Tip:**  "no ip routing" is used to clear routing information in the configuration setup above. |

1. When you are done, use the following commands to check the changes you made to the router configuration:

**R1#** show interfaces

**R1#** show interfaces brief  
**R1#** show running-config

### **Exercise 1(C). Setting static routing table entries on a Cisco router**

In this exercise, you will add static routes to the routing table of R1. The routing table must be configured so that it conforms to the network topology shown in Figure 3.1 and Table 3.1. The routes are configured manually, which is also referred to as *static routing*.

The IOS command to configure static routing is **ip route**. The command can be used to show, clear, add, or delete entries in the routing table. The commands are summarized in the list below.

|  |
| --- |
| **IOS MODE: PREVILEGED EXEC**  show ip route  Displays the contents of the routing table  show ip route IPAddress  Displays the information in the routing table for a specific route given by IPAddress  clear ip route \*  Deletes all routing table entries  clear ip route IPAddress  Deletes IPAddress from routing table  clear ip route IPAddress/prefix  Deletes network given by IPAddress/prefix from routing table  show ip cache  Displays the routing cache  show ip cache IPAddress mask  Displays the routing cache entry for IPAddress/prefix  show ip cache Iface  Displays the routing cache entry for Iface |

|  |
| --- |
| **IOS MODE: GLOBAL CONFIGURATION**  ip route **destination mask gw\_address**  no ip route **destination mask gw\_address**  Adds or deletes a static routing table entry to **destination** with netmask **mask**. The argument **gw\_address** is the IP address of the next-hop router.  ip route **0.0.0.0 0.0.0.0 gw\_address**  no ip route **0.0.0.0 0.0.0.0 gw\_address**  Adds or deletes a default routing table entry to a gateway where **gw\_address** is the IP address of the next-hop router  ip route **destination mask Iface**  no ip route **destination mask Iface**  Adds or deletes a static routing table entry to **destination** with netmask **mask**. Here, the next-hop information is the name of a network interface (e.g. FastEthernet0/0). |

Next we show some examples for adding and deleting routing table entries in Cisco IOS. Note that whenever an IP address is configured for a network interface on a router, routing table entries for the directly connected network are added automatically.

The command for adding a route on R1 for the network address 10.0.1.0/24 with 10.0.2.22 as the next-hop gateway IP address is

**R1(config)#** ip route 10.0.1.0 255.255.255.0 10.0.2.22

|  |
| --- |
| **NOTE**  This is very important because if you do not set up the IP routes between the routers, the routers will never be able to ping each other from remote networks. |

The command below shows you how to add a host route to a host with IP address 10.0.2.65 with next-hop (gateway) set to 10.0.1.21. In IOS, a host route is identified by a 32bit prefix.

**R1(config)#** ip route 10.0.2.65 255.255.255.255 10.0.1.21

The command to add e.g. the IP address 10.0.4.4 as the default gateway is done with the command

**R1(config)#** ip route 0.0.0.0 0.0.0.0 10.0.4.4

Finally, to delete any specific entry use the no ip route command. For example:

**R1(config)#** no ip route 10.0.1.0 255.255.255.0 10.0.2.22  
**R1(config)#** no ip route 10.0.2.65 255.255.255.255 10.0.1.21  
**R1(config)#** no ip route 0.0.0.0 0.0.0.0 10.0.4.4

1. Display the content of the routing table with show ip route. Note the routing entries that are already present. Save the output.
2. Add routing entries to R1 and R2, so that the routers forward datagrams and operate correctly for the configuration shown in Figure 3.1. Routing entries should exist for the following networks in each router (either directly connected or via a nexthop/gateway).
3. 10.0.1.0/24
4. 10.0.2.0/24
5. 10.0.3.0/24
6. Display the routing table again with show ip route and save the output.

### **Lab Questions**

* Explain the fields of the routing table entries of the Cisco router.

**PART 2. Configuring a PC with static routes**

### **Exercise 2(A). Network setup**

1. Start all the PCs on GNS3. Then, configure the IP addresses of the interfaces as given in Table 3.1.
2. Start Wireshark to capture traffic on PC1 link.
   1. Before you send ping commands, let’s save a filter to the Wireshark capture window as discussed in Part 3 of Lab 1.
   2. Open the add filter toolbar to add the expression “!(loop || cdp)” and label it something simple like “No cisco data” or something along those lines. Save the expression.

A screenshot of a computer

Description automatically generated

* 1. This will filter out the loop and cdp packets that the Cisco routers send periodically and can fill up your Wireshark capture data easily. Anytime we use routers in any of the labs be sure to add this command in Wireshark (if you disabled “cdp” when setting up the router, then you only need to filter out LOOP packets. But this expression will work even if no cdp packets are being transmitted).

1. Issue a ping command from PC1 to R1, R2 and PC4, respectively.

**PC1%** ping 10.0.1.1 –c 5 **PC1%** ping 10.0.2.2 –c 5  
**PC1%** ping 10.0.3.41 –c 5

1. Save the captured Wireshark output.

### **Lab Questions**

Use the saved data to answer the following questions:

* What is the output on PC1 when the ping commands are issued?
* Which packets, if any, are captured by Wireshark?
* Do you observe any ARP or ICMP packets? If so, what do they indicate?
* Are some of the destinations not reachable? If yes, which ones?

### **Exercise 2(B). Setting static routing table entries for a PC**

Next, you will set up the routing tables of the PCs. The routing tables are configured so that they conform to the network topology shown in Figure 3.1 and Table 3.1.

Configuring static routes in Linux is done with the command route, which has numerous options for viewing, adding, deleting, or modifying routing entries. The various uses of the route command are summarized in the list below.

In Linux, there is no simple way to delete all entries in the routing table. When the commands are issued interactively in a Linux shell, the added entries are valid until Linux is rebooted. To make static routes permanent, the routes need to be entered in the configuration file /etc/sysconfig/static-routes, which is read each time Linux is started.

|  |
| --- |
| route add –net **netaddress** netmask **mask** gw **gw\_address**  route add –net **netaddress** netmask **mask** dev **iface**  Adds a routing table entry for the network prefix identified by IP address **netaddress** and netmask **mask**. The next-hop is identified by IP address **gw\_address** or by interface **iface**.  **Example:** The command for adding a route for the network address 10.21.0.0/16 with next-hop address 10.11.1.4 is:  route add –net **10.21.0.0** netmask **255.255.0.0** gw **10.11.1.4**  route add –host **hostaddress** gw **gw\_address**  route add –host **hostaddress** dev **iface**  Adds a host route entry for IP address **hostaddress** with the next-hop identified by IP address **gw\_address** or by interface **iface**.  route add default gw **gw\_address**  Sets the default route to IP address **gw\_address**.  route del –net **netaddress** netmask **mask** gw **gw\_address**  route del –host **hostaddress** gw **gw\_address**  route del default gw **gw\_address**  Deletes an existing route from the routing table with specific arguments.  route -e  Displays the current routing table with extended fields. The command is identical to the netstat –r command.  ip route flush table main  deletes all entries in the routing table on a PC  ip route flush cache  deletes all entries in a routing cache on a PC  Please note that when you flush the routing table, you cannot add any entries to the routing table again. You need to STOP the PC and then restart it. It will need to be reconfigured.  ip route get **IPAddress**  displays the cached route for **IPAddress**  ip route flush cache **IPAddress**  flushes the cached route entry for **IPAddress** |

|  |
| --- |
| **Tip:** The following two commands are helpful to debug mistakes in configuration setup and get information on routing. The ping command tests whether **IPaddr** can be reached or not, and the traceroute command displays the route to an **IPaddr**.  ping **IPaddr** traceroute **IPaddr** |

1. Configure the routing table entries of PC1 and PC4. You can either specify a default route or you can insert separate routing entries for each remote network. For this exercise, add a route for each individual remote network. Below we show you how to set up the routing configuration for PC4. Follow similar steps to setup the static routes on PC1.

**PC4%** route add –net 10.0.2.0 netmask 255.255.255.0 gw 10.0.3.1  
**PC4%** route add –net 10.0.1.0 netmask 255.255.255.0 gw 10.0.3.1

1. Display the routing table of PC1 and PC4 with route -e and screenshot the output.

### **Lab Questions**

* Explain the entries in the routing table and discuss the values of the fields for each entry.

# **PART 3. More on ROUTER Configuration**

If the configuration of PC2 and R1 was done correctly, it is now possible to send IP datagrams between any two machines in the network shown in Figure 3.1. In most real network configurations, the network configuration requires additional changes before all hosts and routers can send and receive IP datagrams. However, if the network is not configured properly, you will need to debug and test your setup. The table below illustrates several common problems that may arise. Since it is impossible to cover all scenarios, network debugging is a crucial skill that you need to attain for your lab experiments to work well.

### **Exercise 3(A). Testing the network setup**

1. Test the network configuration by issuing ping commands from each host and router to every other host and router. If some ping commands do not work, you need to modify the software configuration of routers and hosts. If all ping commands are successful, the network configuration is correct, and you can proceed to the next step.
2. Start Wireshark on PC1 link.
3. Execute a traceroute command from PC1 to PC4, and save the output.

**PC1%** traceroute 10.0.3.41

1. Execute a trace command from R1 to PC4, and save the output.

**R1#** trace 10.0.3.41

1. Stop Wireshark and save the captured traffic. Observe how traceroute commands gather route information.
2. Save the routing table of PC1, PC4, R1 and R2.

### **Lab Question**

* Using the Wireshark output and the previously saved routing tables, explain the operation of traceroute command.

### **Exercise 3(B). Observe MAC addresses at a router**

When a router forwards an IP datagram from one Ethernet segment to another, it does not modify the IP destination address. However, it modifies the destination address in the Ethernet header.

This exercise requires manipulations to the ARP cache. The ARP command in Linux was covered in Lab 2. The list shows corresponding IOS commands for Cisco routers.

|  |
| --- |
| **IOS MODE: PRIVILEGED EXEC**  show arp  Displays the contents of the ARP cache  clear arp-cache  Deletes the entire ARP cache, but sends multiple gratuitous arp to deleted entries  clear ip arp  Deletes the entire ARP cache, but sends ONE gratuitous arp to deleted entries  clear ip arp force-delete  Deletes the entire ARP cache with no refresh (i.e., no gratuitous arp to deleted entries)  **IOS MODE: GLOBAL CONFIGURATION**  arp **IPaddress H/WAddress** arpa  no arp **IPaddress H/WAddress** arpa  Adds or deletes a **static** ARP entry for IPaddress with MAC in the ARP cache.  These entries need to be removed individually if ARP cache is to be cleared. |

1. Erase all ARP entries on PC1, PC4, and R1, R2.
2. Run Wireshark on PC1 and PC4 links.
3. Issue a ping command on PC1 to PC4.

**PC1%** ping 10.0.3.41 –c 5

1. Stop and save Wireshark output.

### **Lab Questions**

* Determine the source and destination addresses in the Ethernet and IP headers, for the ICMP Echo Request messages that were captured at PC1.
* Determine the source and destination addresses in the Ethernet and IP headers, for the ICMP Echo Request messages that were captured at PC4.
* Use your answers above to explain how the source and destination Ethernet and IP addresses are changed when a datagram is forwarded by a router.

### **Exercise 3(C). Order of the routing table lookup**

A router or host uses a routing table to determine the next hop of the path of an IP datagram. Generally, routing table entries are sorted in the order of decreasing prefix length, and are read from top to bottom. In this exercise, you determine how an IP router or PC resolves multiple matching entries in a routing table.

1. Add the following routes to the routing table of PC1:

**PC1%** route add –net 10.0.0.0 netmask 255.255.0.0 gw 10.0.1.71  
**PC1%** route add –host 10.0.3.9 gw 10.0.1.81

From Exercise 2(B), there should be a network route for the network prefix 10.0.3.0/24 in PC1 routing table. If there is no such route, then add the following entry:

**PC1%** route add –net 10.0.3.0 netmask 255.255.255.0 gw 10.0.1.61

1. Referring to the routing table, determine how many matches exist for the following IP addresses:
2. 10.0.3.9
3. 10.0.3.14
4. 10.0.4.1
5. Start a Wireshark session on PC1, and issue the following ping commands from PC1:

**PC1%** ping 10.0.3.9 –c 5  
**PC1%** ping 10.0.3.14 –c 5  
**PC1%** ping 10.0.4.1 –c 5

Note that gateways with IP addresses 10.0.1.61, 10.0.1.71, and 10.0.1.81 do not exist.

1. Show PC1’s routing table with the command route -e. Screenshot and save.
2. Stop and save the output of Wireshark and the PC1's routing table.

### **Lab Question**

* Use the saved output to indicate the number of matches for each of the IP addresses above. Based upon what you have seen, explain how PC1 resolves multiple matches in the routing table.

### **Exercise 3(D). Default routes**

1. Delete the routing table entries added to PC1 in Step 1 of Exercise 3(C) above using the "route del" command. (Otherwise, the entries will interfere with the remaining exercises in this lab.)
2. Add default routes on PC1 an PC4.
3. On PC1, add a default route for interface FastEthernet0/0 on R1 as the default gateway.
4. On PC4, add a default route for interface FastEthernet1/0 of R2 as the default gateway.
5. Start Wireshark to capture traffic on PC1 link.
6. Issue a ping command from PC1 to a host on a network that does not exist, e.g.:

**PC1%** ping 10.0.10.110 –c 5

1. Save the Wireshark output.

### **Lab Questions**

Use the saved output to answer the following questions.

* What is the output on PC1 when the ping command is issued?
* Determine how far the ICMP Echo Request message travels.
* Which, if any, ICMP Echo Reply message returns to PC1?

# **PART 4. PROXY ARP**

Proxy Address Resolution Protocol (Proxy ARP) enables a host to send packets to remote subnets without using its routing table. Proxy ARP is a configuration option when an IP router responds to ARP Requests that arrive from one of its connected subnets for a host that is on another of its connected subnets. Without Proxy ARP enabled, an ARP Request for a host on a different network would be unsuccessful, since routers do not forward ARP packets to another subnet.

In this part, you explore how Proxy ARP enables routers to forward an IP datagram even though the sender of the datagram is not aware that the IP datagram should be forwarded to a router. Continue with the network configuration from Figure 3.1, and with IP addresses as shown in Table 3.1. The commands to enable and disable Proxy ARP in IOS are listed below.

|  |
| --- |
| **IOS MODE: INTERFACE CONFIGURATION**  ip proxy-arp  no ip proxy-arp  Proxy ARP is enabled and disabled separately on each interface. In IOS, proxy ARP is enabled by default. Note that when configuring the interface ip proxy-arp and no ip proxy-arp you have to **terminate** the interface configuration mode with “end” to **enable** the changes. E.g.:  **Router(config-if)#** no ip proxy-arp  **Router(config-if)#** end |

### **Exercise 4. Observing Proxy ARP**

1. Stop PC4, then restart it.
2. Configure PC4 with the IP address given in Table 3.1 BUT with a netmask of 255.0.0.0, so that PC4 assumes it belongs to subnet 10.0.0.0/8, not 10.0.3.0/24.
3. Start Wireshark capture on PC4 and PC1 links. Set a display or capture filter for ICMP and ARP packets only (icmp || arp).
4. Issue a ping from PC4 to PC1:

**PC4%** ping 10.0.1.11 –c 5

Even though PC4 had no default routing entry in its table for R2, it was still able to connect to PC1, i.e., you should not observe a “network unreachable” response.

1. Stop Wireshark capture on both links and save. Show and save the ARP table of PC4.
2. Now, disable Proxy ARP on both interfaces of R2. Is it still feasible to issue a ping from PC4 to PC1?
3. Reset the subnet mask of PC4 to its original value of 255.255.255.0. Re-enable Proxy ARP on R2.

### **Lab Question**

* Use the captured data to explain the outcome of the different scenarios in the exercise above.
* Use the data to explain how Proxy ARP allowed PC4 to communicate with PC1.

# **PART 5. ICMP ROUTE REDIRECT**

ICMP route redirect messages are sent from a router to a host, when a datagram should have been forwarded to a different router or interface. In Linux, an ICMP route redirect message updates the *routing cache*, but not the *routing table*.

Both the routing cache and the routing table contain information for forwarding traffic. Before a Linux system performs a routing table lookup, it first inspects the routing cache. If no matching entry is round in the cache, Linux performs a lookup in the routing table. After each routing table lookup, an entry is added to the routing cache. The routing cache does not aggregate table entries, and there is a separate entry for each destination IP address. As a consequence, a lookup in the routing cache does not require a longest prefix match. An entry in the routing cache is deleted if it has not been used for some time, usually after 10 minutes. When an ICMP Redirect message arrives, an entry is added to the routing cache, but no update is performed to the routing table.

Recall the following commands to display the contents of the routing cache in Linux:

ip route get **IPAddress**

To clear the route cache in Linux:

ip route flush cache **IPAddress**

or

ip route flush cache

Similarly, for IOS the commands are:

show ip cache

clear ip cache

In this part of the lab, you will use three Routers. Figure 3.2 and Table 3.2 describe the network configuration for the exercises below.

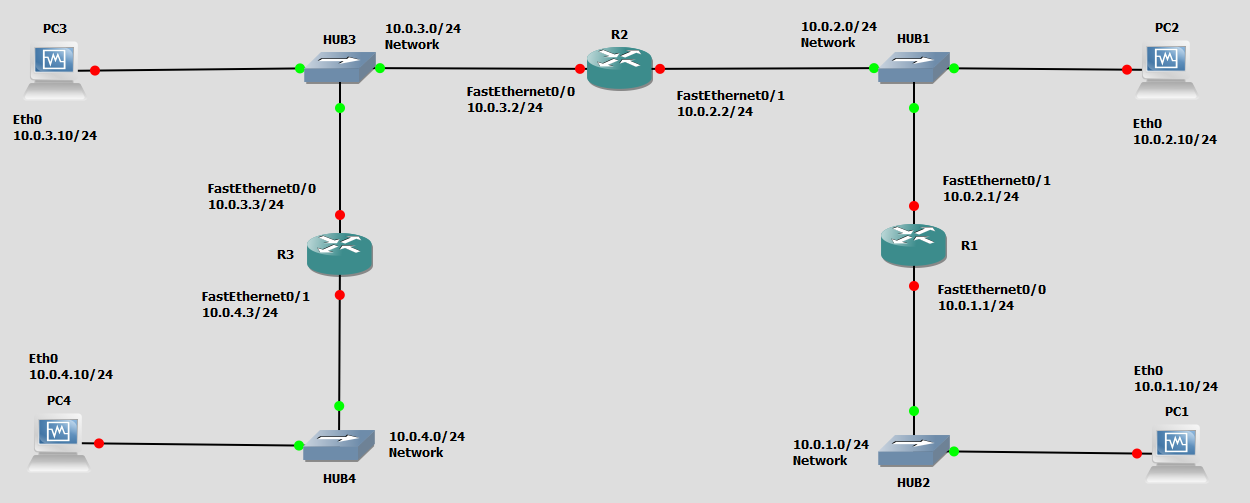


Figure 3.2 Network topology for Part 5

|  |  |  |
| --- | --- | --- |
| **Cisco Routers** | **FastEthernet0/0** | **FastEthernet1/0** |
| R1 | 10.0.1.1 / 24 | 10.0.2.1 / 24 |
| R2 | 10.0.3.2 / 24 | 10.0.2.2 / 24 |
| R3 | 10.0.3.3 / 24 | 10.0.4.3 / 24 |
| **PC** | **Ethernet Interface eth0** |  |
| PC1 | 10.0.1.10 / 24 |  |
| PC2 | 10.0.2.10 / 24 |  |
| PC3 | 10.0.3.10 / 24 |  |
| PC4 | 10.0.4.10 / 24 |  |

Table 3.2 IP addresses for Part 5

**Exercise 5. Observing ICMP Redirect**

In the network shown in Figure 3.2, when PC2 sends datagrams with destination 10.0.3.10 (PC3) to 10.0.2.1 (R1), as opposed to 10.0.2.2 (R2), then R1 sends an ICMP route redirect message to PC2. The ICMP route redirect informs PC2 that it should send datagrams with destination 10.0.3.10 to R2 instead.

In this exercise, you will create the above scenario. You will trigger the transmission of an ICMP Route Redirect message and subsequently observe a change to the routing cache.

1. Connect the Ethernet interfaces of the routers and the hosts to the hubs as shown in Figure 3.2.
2. Delete the routing table entries, route caches and ARP caches on all PCs and on all Routers.
3. Build a new static routing entry on R1 for network 10.0.3.0/24 to R2 (FastEthernet1/0)
4. ICMP redirect messages can be used to attack a network. For this reason, hosts by default ignore ICMP redirect messages. On a Linux system, the accept\_redirects variable controls whether the host can accept or not a redirect ICMP message.
5. Use sysctl command to verify the current ICMP redirect status on PC2

**PC2%** sysctl net.ipv4.conf.all.accept\_redirects

1. If the response is “0”, then you need to enable it. E.g., enable PC2 to accept ICMP redirect messages.

**PC2%** echo 1 | tee /proc/sys/net/ipv4/conf/\*/accept\_redirects

1. Use the sysctl command again to ensure that the parameter change occurred.
2. Set up the routing table of PC2 in such a way that it provokes the transmission of an ICMP route redirect message as discussed above, i.e., make R1 the default router for PC2. In other words, force it to send the packet to a router other than the one you would expect it to use to get to PC3.
3. Save the contents of the routing table and the routing cache on each of R1, R2, and PC2.
4. Set up the routing table of PC3 and PC4 so that they can reach PC2.
5. Start Wireshark on PC2 link to capture ICMP messages and issue a ping -c 5 from PC2 to PC3. Repeat for a ping –c 5 from PC2 to PC4.
6. Stop Wireshark, save the capture data and the contents of the routing table and the routing cache of PC2, and routers R1, R2 after the ICMP redirect messages.
7. Wait a few minutes with no transmissions occurring and check the contents of the routing caches again. Save the output.

### **Lab Questions**

* Is there a difference between the contents of the routing table and the routing cache immediately after the ICMP route redirect message?
* When you viewed the cache a few minutes later, what did you observe?
* Describe how the ICMP route redirect works using the outputs you saved. Include only relevant data from your saved output to support your explanations.
* Explain how R1, in the above example, knows that datagrams destined to network 10.0.3.10 should be forwarded to 10.0.2.2?

# **PART 6. NETWORK PREFIXES and ROUTING**

In this exercise you study how the network prefixes (netmasks) play a role when hosts determine if a datagram can be directly delivered or if it must be sent to a router.

This part uses the network setup shown in Figure 3.3. The network includes one router, four hosts and two hubs. The IP addresses of all devices are given in Table 3.3. Here, each host has only a default route. In other words, the routing table at a host only knows about the directly connected networks and the default gateway.

A picture containing sky, screenshot, indoor

Description automatically generated

Figure 3.3 Network topology for Part 6.

|  |  |  |
| --- | --- | --- |
| **PCs** | **Ethernet Interface eth0** |  |
| PC1 | 10.0.1.10 / 24 |  |
| PC2 | 10.0.2.10 / 24 |  |
| PC3 | 10.0.2.137 / 29 |  |
| PC4 | 10.0.2.139 / 24 |  |
| **Cisco Routers** | **FastEthernet0/0 (F0/0)** | **FastEthernet1/0 (F1/0)** |
| R1 | 10.0.1.1 / 24 | 10.0.2.138 / 24 |

Table 3.3. IP addresses for Part 6.

### **Exercise 6. Exploring the role of prefixes at hosts**

In this exercise, you explore how hosts that are connected to the same local area network, but that have different netmasks, communicate or fail to communicate.

1. Configure the hosts and the router to conform to the topology shown in Figure 3.3, using the IP addresses as given in Table 3.3. Note that PC2, PC3, and PC4 have different netmasks.
2. Add R1 as default gateway on all hosts. (PC1, PC2, PC3, and PC4.)
3. **Issue ping command from PC1 to PC3**
4. Clear the ARP table on all PCs.
5. Start Wireshark on links connected to PC1 and PC3, and set the capture filter to capture ICMP and ARP packets only (ICMP or ARP).
6. Issue a ping command from PC1 to PC3 for at least 5 sends (-c 5).
7. Save the output of the ping command at PC1. Stop and save the output of Wireshark on PC1 and PC3 links.
8. Show the ARP tables, routing tables, and routing caches of each PC. Screenshot and save.
9. **Issue ping command from PC3 to PC4**
10. Clear the ARP table on all PCs.
11. Start Wireshark on PC3 and set the capture filter to capture ICMP and ARP packets only.
12. Issue a ping command from PC3 to PC4 for at least 5 sends (-c 5).
13. Save the output of the ping command at PC3. Stop and save the output of Wireshark on PC3.
14. Show the ARP table, routing table, and routing cache of PC3. Screenshot and save.
15. **Repeat Step 4, but this time issue a ping from PC3 to PC2**.

Note that once an entry is made in the routing cache, you cannot repeat the previous experiment to obtain the same results. You have to wait a few seconds until the routing cache is reset (ICMP re-direct entry is deleted) or you can delete the routing caches on all devices.

### **Lab Questions**

* Explain what you observed in steps 3, 4 and 5. Use the saved data to support your answers. Provide explanations of the observations. Try to explain each observed phenomenon, e.g., if you observe more ICMP Echo Requests than Echo Replies, try to explain the reason.
* If PC3 had no default entry in its routing table, would you have seen the same results? Explain for each of the pings above what would have been different

# **PART 7. ICMP Denial of Service (DoS) Attack**

In this part of the lab, you will be using the Mallory as the attacker to attack a local area network by actively broadcasting ICMP requests to every single host in the network. This will overload the network and cause a denial of service for several time critical services running within the network.

hping3 is a command for sending (almost) arbitrary TCP/IP packets to network hosts via a broadcast. You can also specify UDP and ICMP packets.

This part of the Lab utilizes this command to send spoofed ICMP requests to flood a targeted host on a network. It can also send messages to a network broadcast address. Note that to make that work, the PCs have to be able to accept broadcast ICMP messages. By default they don’t. We show you how to enable that feature and observe the DoS attack.

hping3 -1 --flood **IPAddress**

Send ICMP packets flooding the target **IPAddress**, -1 flag here indicates that packets are sent using the ICMP protocol. The *flood* option here tells the command to send packets as fast as possible, ignoring incoming replies.

hping3 -1 --flood --rand-source **IPAddress**

This command does the same thing as the previous command, except sending the packets from a random source address instead of your address.

hping3 -1 --flood -a **TargetIPAddress DestinationIPAddress**

This command will send packets from **TargetIPAddress** to the **DestinationIPAddress** using the ICMP protocol, as well as sending as fast as possible.

For more info about the hping3, look up it’s MAN page in the Linux manual:

<https://linux.die.net/man/8/hping3>

**Exercise 7. Using the hping3 for Denial of Service Attack**

* 1. For this exercise we will use the network configuration shown below in Figure 3.4, and Table 3.4. You will login to Mallory, and fire up the terminal. Remember to turn “Wired Connection” off. Now Mallory is ready to be configured. Always ‘su root’ for VMs.

A screenshot of a video game

Description automatically generated

Figure 3.4

* 1. Start GNS3. Wait a few minutes for Mallory to start.
  2. Assign IP addresses to the PCs and Mallory as shown in Table 3.4:

|  |  |
| --- | --- |
| **PCs** | **IP Address of eth0** |
| PC1 | 10.0.1.11 / 24 |
| PC2 | 10.0.1.12 / 24 |
| PC3 | 10.0.1.13 / 24 |
| Mallory | 10.0.1.44 / 24 |

Table 3.4

* 1. Start Wireshark on the link between PC1 and the Hub.
  2. Issue a ping command from PC2 to PC1, DO NOT terminate the PING command.

**PC2%** ping 10.0.1.11

* 1. Now issue the following command on Mallory to flood ICMP messages to PC1:

**Mallory%** hping3 --icmp --flood 10.0.1.11

* 1. Do not stop the hping3 command on Mallory.
  2. Now switch to PC2 and observe the incremental time taken for each ping request.
  3. Stop the ping command on PC2 (^C). Screenshot PC2’s console window showing the ping output.
  4. Stop the hping3 command on Mallory (^C).
  5. Examine the packets captured in Wireshark.
  6. By default the PCs ignore broadcast ICMP requests. To disable “ignore ICMP broadcast requests” execute the following command on PC1, PC2 and PC3

echo 0 > /proc/sys/net/ipv4/icmp\_echo\_ignore\_broadcasts

* 1. Use sysctl command to verify the current ICMP redirect status on PC1, PC2, and PC3. It should be 0.

sysctl net.ipv4.icmp\_echo\_ignore\_broadcasts

* 1. Issue a ping command from PC2 to PC1, DO NOT terminate the PING command.

**PC2%** ping 10.0.1.11

* 1. Now execute the following command on Mallory to spoof your IP address as PC1’s IP address and send ICMP request to the broadcast address of the subnet.

**Mallory%** hping3 --icmp --flood -a 10.0.1.11 10.0.1.255

* 1. Switch to PC2 and observe the output of the ping command. You should see a diference from previous output.
  2. Stop hping3 on Mallory (^C) and the ping command on PC2(^C).
  3. Stop Wireshark capture. Save the output.
  4. Stop GNS3, quit VMWare and quit GNS3.

### **Lab Questions**

* What difference did you notice when looking at the screen shots of the ping command output on PC2? Is there a huge difference between the time taken for each ping request? How would this impact the service provided by PC1?
* What difference did you notice in the output when doing step 5-8 compared to the packets captured from the first run of steps 14 – 16?
* Use the packets captured in Wireshark for the ping issued in step 14 to explain why all the PCs on the network sent ICMP replies back to PC1 even though it was not sending any packets.

**APPENDIX A. Summary of Cisco Commands w Brief Explanations**

**#1: The "?"**

It may seem entirely too obvious that you should know how to type  *?* to ask for help when using the Cisco IOS. However, the Cisco IOS is completely different from other operating systems when it comes to using the question mark (help key). As the IOS is a command-line operating system with thousands of possible commands and parameters, using the *?* can save your day.

You can use the command in many ways. First, use it when you don't know what command to type. For example, type  *?* at the command line for a list of all possible commands. You can also use *?* when you don't know what a command's next parameter should be. For example, you might type *show ip ?* If the router requires no other parameters for the command, the router will offer <CR> as the only option.

Finally, use  *?* to see all commands that start with a particular letter. For example, *show c?* will return a list of commands that start with the letter "c".

**#2: show running-configuration**

The  *show running-config* command shows the router, switch, or firewall's current configuration. The running-configuration is the config that is in the router's memory. You change this config when you make changes to the router. Keep in mind that config is not saved until you do a *copy running-configuration startup-configuration*. This command can be abbreviated *sh run*.

**#3: copy running-configuration startup-configuration**

This command will save the configuration that is currently being modified (in RAM), also known as the running-configuration, to the nonvolatile RAM (NVRAM). If the power is lost, the NVRAM will preserve this configuration. In other words, if you edit the router's configuration, don't use this command and reboot the router--those changes will be lost. This command can be abbreviated *copy run start*. The *copy* command can also be used to copy the running or startup configuration from the router to a TFTP server in case something happens to the router.

**#4: show interface**

The *show interface* command displays the status of the router's interfaces. Among other things, this output provides the following:

* Interface status (up/down)
* Protocol status on the interface
* Utilization
* Errors
* MTU

This command is essential for troubleshooting a router or switch. It can also be used by specifying a certain interface, like *sh int g0/0*.

**#5: show ip interface**

Even more popular than show interface are *show ip interface* and *show ip interface brief*. The *show ip interface* command provides tons of useful information about the configuration and status of the IP protocol and its services, on all interfaces. The *show ip interface brief* command provides a quick status of the interfaces on the router, including their IP address, Layer 2 status, and Layer 3 status.

**#6: config terminal, enable, interface, and router**

Cisco routers have different modes where only certain things can be shown or certain things can be changed. Being able to move between these modes is critical to successfully configuring the router.

For example, when logging in, you start off at the user mode (where the prompt looks like >). From there, you type enable to move to privileged mode (where the prompt looks like #). In privileged mode, you can show anything but not make changes. Next, type *config terminal* (or *config t*) to go to global configuration mode (where the prompt looks like router(config)# ). From here, you can change global parameters. To change a parameter on an interface (like the IP address), go to interface configuration mode with the *interface* command (where the prompt looks like router(config-if)#). Also from the global configuration mode, you can go into router configuration using the *router {protocol}* command. To exit from a mode, type *exit*.

**#7: no shutdown**

The *no shutdown* command enables an interface (brings it up). This command must be used in interface configuration mode. It is useful for new interfaces and for troubleshooting. When you're having trouble with an interface, you may want to try a *shut* and *no shut*. Of course, to bring the interface down, reverse the command and just say *shutdown*. This command can be abbreviated *no shut*.

**#8: show ip route**

The *show ip route* command is used to show the router's routing table. This is the list of all networks that the router can reach, their metric (the router's preference for them), and how to get there. This command can be abbreviated *sh ip ro* and can have parameters after it, like *sh ip ro ospf* for all OSPF routes. To clear the routing table of all routes, you do *clear ip route*. To clear it of just one route, issue the command *clear ip route x.x.x.x* (where *x.x.x.x* is the network you want to clear).

# **APPENDIX B. A TIP TO SPEED UP the SETUP of EXPERIMENTS**

To configure the cisco routers and PCs, you are required to enter the commands manually in the console window. By now you will probably have realized that many of the commands are repeated. Also, you probably noticed that GNS3 resets PC and Cisco router configurations when you stop a GNS3 project. Saving the commands in a file and then using copy and paste of the command set will save you a lot of time.

Below we show you how you can save configuration commands in a Microsoft Excel Spread Sheet, for use at a later time.

1. Open an Excel spread sheet. Then, add a command in each row as shown in Figure A.1. The columns represent the different devices you are saving commands for.

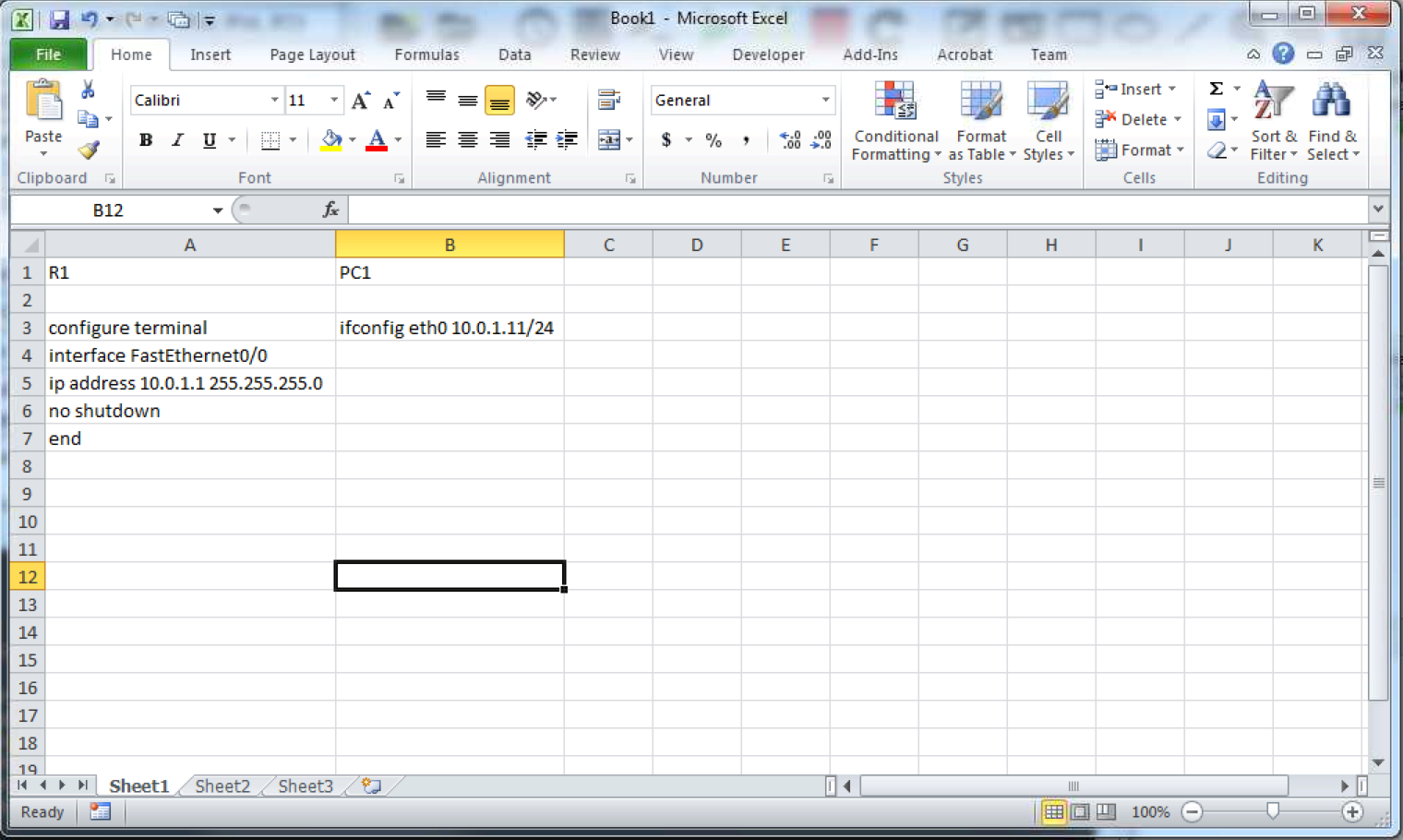


Figure A.1

1. Select a command set from a column and copy the block of commands by pressing the shortcut key “Ctrl + C” for Windows and “Command + C” for Mac. The command set will be highlighted as shown in Figure A.2.

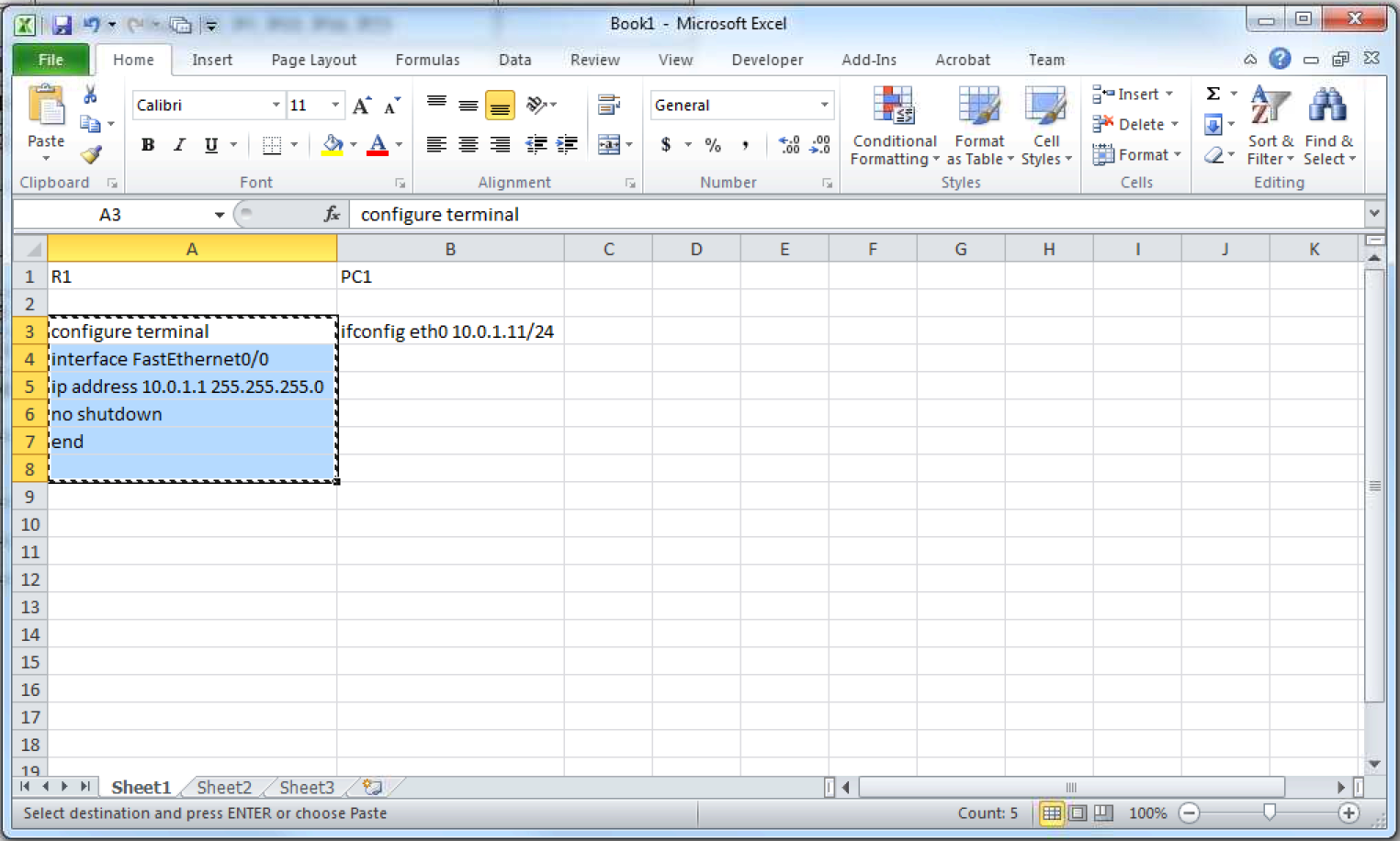


Figure A.2

1. Start a Cisco router R1 and a PC1 in GNS3.
2. Open console windows for each of R1 and PC1.
3. Now you paste the commands by right click on the console windows for Windows users. For Mac users, paste the commands with the shortcut key “Command + V”. You will see that the commands are executing as shown in Figure A.3 for R1.

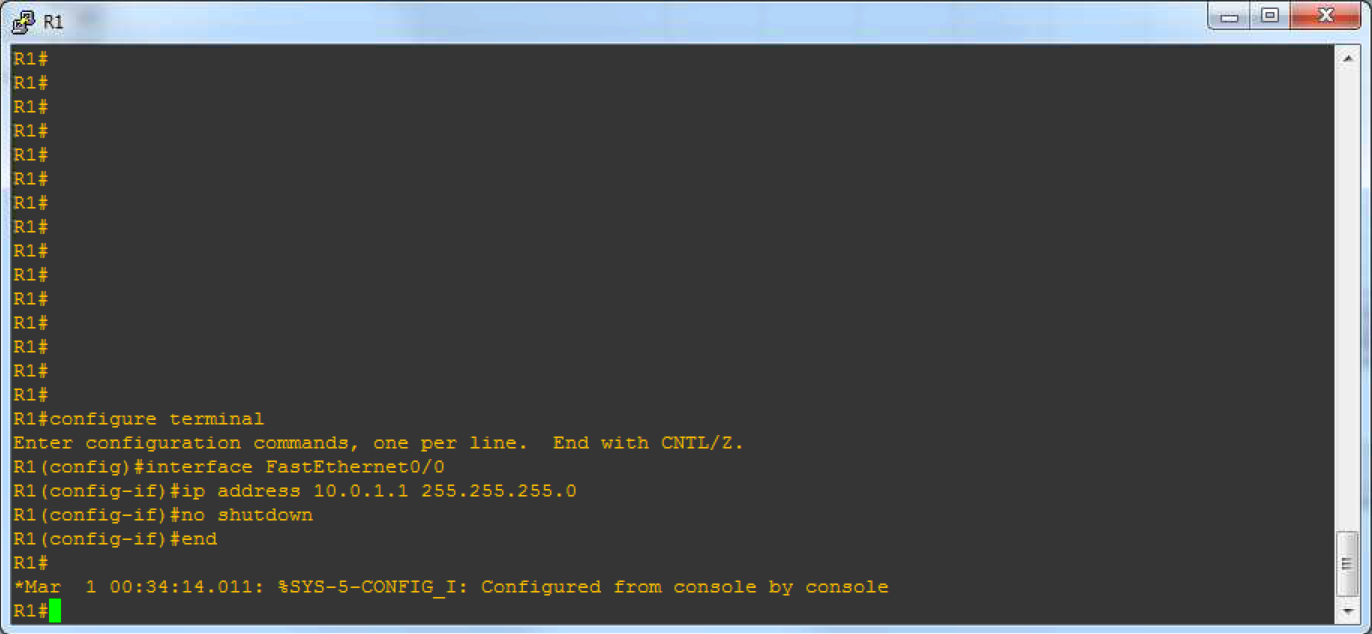


Figure A.3

1. Repeat the above step to configure PC1.
2. Please make sure that the commands do not contain any typos.
3. Save the excel spread sheet. You can use for the next experiment if the configuration is similar. If not you can edit the saved file, e.g., IP address may have changed. And repeat the copy-paste routing to configure your devices.

**APPENDIX B. Saving a Router Configuration in GNS3 Project**

For routers, you can use the command ***# copy running-config startup-config***to save the configurations, and then save your GNS3 project. This will ensure your router configurations don't get erased when you restart GNS3 and open that project.