Creating a Simple OSPF Network with Internet Connectivity Through NAT

We will build a basic network layout with OSPF routing and connect it to the internet using a **c7200** router that performs NAT.

Please construct and label a GNS3 network topology shown in Figure 1. Please be sure to use the NAT node (not cloud). Please be sure to turn on the network adapter labels by clicking the button on the panel.

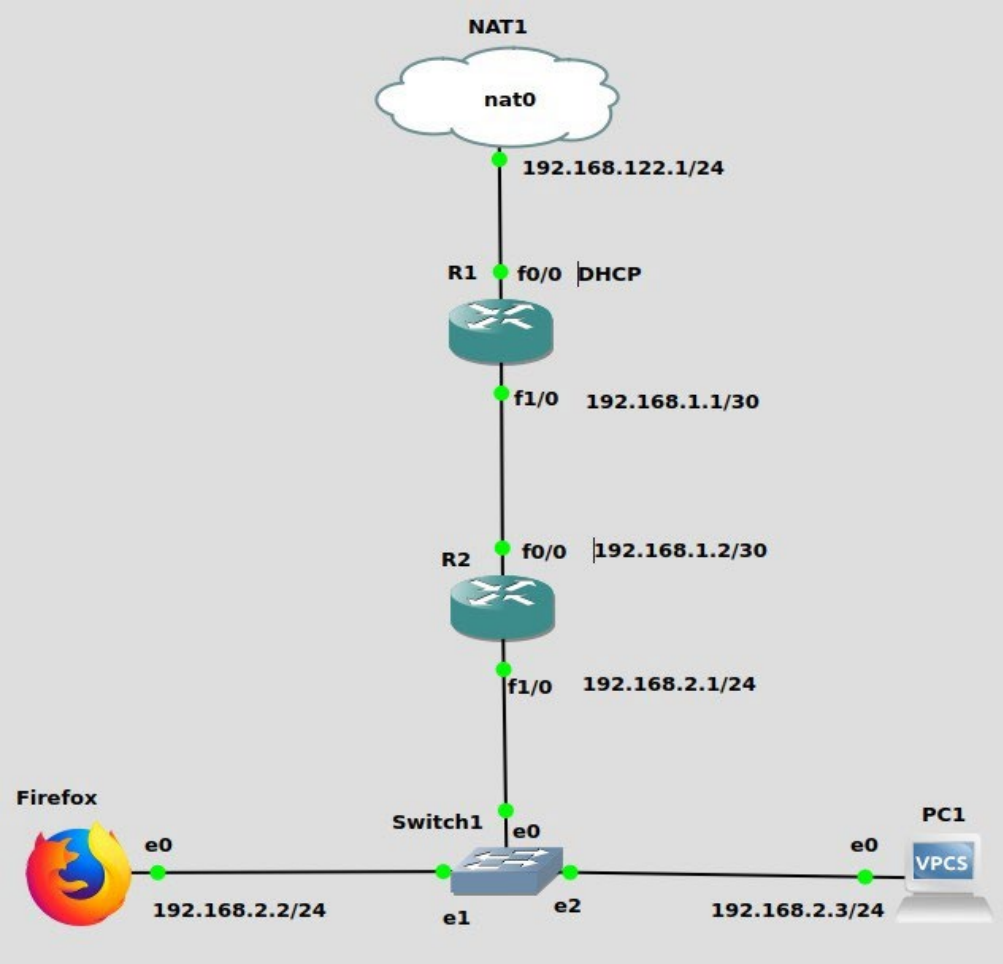

Figure 1: Network Topology

Please note: if you do not have the Firefox VM installed, please install it as discussed in the first assignment.

Router R1 will perform network address translation between the traffic inside of our network topology and the outside world. The NAT node's interface also hosts a DHCP server that will dynamically assign network addresses in the subnet of 192.168.122.0/24. This means that the f0/0 interface of router R1 will have the address assigned dynamically.

Please perform the following steps to configure our network.

Step 1: *Start all devices on the network and configure the IP and subnet mask of R1 interface f 0/1:*

1a. Open the console for router R1 (right-click → Console). In the console that opens, please run the following commands:

```
R1(config-if)#ip addr 192.168.1.1 255.255.255.252
R1(config-if)#no shut
R1(config-if)#ex
R1(config)#int f0/0
R1(config-if)#ip addr dhcp
R1(config-if)#no shut
```

1b. Next, let's enable DHCP on interface of f0/0 of router R1:

```
R1(config-if)#interface f0/0        // Configure interface f0/0
R1(config-if)#ip address dhcp       // Enable DHCP client on this interface
R1(config-if)#no shut               // Enable the interface
```

The above message indicates that the interface has dynamically obtained the IP address of 192.168.122.7/24 from the NAT node's DHCP server.

```
R1(config)#do sh ip int br
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        192.168.122.78  YES DHCP   up                    up
FastEthernet1/0        192.168.1.1     YES manual up                    up
```

Step 2: *Configure OSPF on router R1:*

2a. Set the router id to 1.1.1.1

```
R1(config-if)#router ospf 1  // Start the OSPF process with a process ID of 1

// Add R1 interfaces part of the subnet 192.168.1.0/30 to OSPF area 0.
// Please note: 0.0.0.3 is the wildcard mask.  It is obtained by taking the
// subnet mask (255.255.255.252) and inverting all of the bits (flipping 1's to 0's and 0's to 1's).
```

2b. R1(config-router)#network 192.168.1.0 0.0.0.3 area 0

Step 3: *Enable NAT on router R1:*

R1(config-router)#interface f0/0 // Enter configuration of interface f0/0
R1(config-if)#ip nat outside // Let the interface represent the public side of NAT
R1(config-if)#interface f1/0 // Switch to interface 1/0 configuration
R1(config-if)#ip nat inside // Let this interface represent the internal network side of NAT

```
R1(config)#router ospf 1
R1(config-router)#network 192.168.1.0 0.0.0.3 area 0
R1(config-router)#int f0/0
R1(config-if)#ip nat outside
R1(config-if)#int f1/0
R1(config-if)#ip nat inside
```

//This command that follows is used to configure Network Address Translation (NAT) on a router that
// has an inside interface connected to a local network and an outside interface connected to the
// internet. NAT allows the router to translate the private IP addresses of the local network devices into // public IP addresses that can be routed on the internet. // The command has the following components:

// ip nat inside source: This enables NAT of the inside source address and specifies that the translation // will be done on the source IP address of the packets coming from the inside network.

// list 1: This refers to an access list that defines which IP addresses should be translated. The access list // number or name must match the one defined by the ip access-list command.

// interface f 0/0: This specifies the interface that connects to the outside network and
// provides the public IP address for the translation. The interface must be configured with the ip nat // outside command.

// overload: This enables Port Address Translation (PAT), which allows multiple inside devices to share // a single public IP address by using different port numbers. PAT helps conserve public IP addresses // and increases security by hiding the internal network structure.
R1(config-if)#ip nat inside source list 1 interface f 0/0 overload

// Translate internal network address within the subnet of 192.168.0.0/16 //
(0.0.255.255 is the wildcard mask). This ACL is allowing communication from all
devices in the 192.168.x.x subnets.
R1(config)#access-list 1 permit 192.168.0.0 0.0.255.255

```
R1(config-if)#ip nat inside source list 1 interface f0/0 overload
R1(config)#access-list 1 permit 192.168.0.0 0.0.255.255
```

Allow domain-lookup to ping www.google.com

```
R1(config)#ip name-server 192.168.122.1
```

```
R1(config)#ip domain-lookup
R1(config)#do ping www.google.com
Translating "www.google.com"...domain server (192.168.122.1) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 142.251.32.36, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/28/40 ms
```

Step 4: *Add a static default route to R1 to route all traffic that has no matches in the router's routing table to the gateway address of 192.168.122.1 (the address of the NAT node interface).*

R1(config)#ip route 0.0.0.0 0.0.0.0 192.168.122.1 // Set the static default route

R1(config)#router ospf 1 // Configure OSPF on this router
R1(config-router)#default-information originate // **advertise the default route** (0.0.0.0) to its OSPF neighbors

```
R1(config)#ip route 0.0.0.0 0.0.0.0 192.168.122.1
R1(config)#router ospf 1
R1(config-router)#default-information originate
```

Step 5: *Configure IP addresses of interfaces on Router R2:*

5a. Configure the IP and subnet mask of interface f 0/0
R2#configure terminal // Enter the configuration terminal
R2(config)#interface f0/0 // Configure interface f0/0
R2(config-if)#ip address 192.168.1.2 255.255.255.252 // Set the IP address and subnet mask

R2(config-if)#no shut // Enable the interface

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#int f0/0
R2(config-if)#ip addr 192.168.1.2 255.255.255.252
R2(config-if)#no shut
```

5b. Configure the IP address and subnet mask of interface f 1/0:

R2(config-if)#interface f1/0 // Configure interface f0/0
R2(config-if)#ip address 192.168.2.1 255.255.255.0 // Set the IP address and subnet mask
R2(config-if)#no shut // Enable the interface
R2(config-if)#end // Exit the configuration terminal

```
R2(config-if)#int f1/0
R2(config-if)#ip ad
*Sep 17 00:31:52.103: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.122.78 on FastEther
net0/0 from LOADING to FULL, Loading Done
R2(config-if)#ip address 192.168.2.1 255.255.255.0
R2(config-if)#no shut
```

Step 6: *Enable OSPF and set the router ID of router R2:*

Input these commands in the figure below:

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#router ospf 1
R2(config-router)#
*Sep 17 00:25:25.703: %OSPF-4-NORTRID: OSPF process 1 failed to allocate unique
router-id and cannot start
R2(config-router)#router-id 2.2.2.2
R2(config-router)#network 192.168.1.0 0.0.0.3 area 0
R2(config-router)#network 192.168.2.0 0.0.0.255 area 0
```

This line: `R2(config-router)#network 192.168.1.0 0.0.0.3 area 0` adds interfaces in the subnet of 192.168.1.0/30 to area 0

[110/1]: These numbers represent the OSPF **administrative distance** and the **cost (metric)** of the route:

110: The administrative distance (AD) for OSPF, which is used to rank the reliability of routes from different routing protocols. OSPF has an AD of 110 by default.

1: This is the metric (cost) for the OSPF external type-2 route. E2 routes have a fixed cost, in this case, 1.

```
R2#show ip route ospf
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 192.168.1.1 to network 0.0.0.0

O*E2  0.0.0.0/0 [110/1] via 192.168.1.1, 00:02:23, FastEthernet0/0
```

Note: **R1** has a static default route pointing to **192.168.122.1** and is advertising it to its OSPF neighbors using the `default-information originate` command. **R2**, as an OSPF neighbor of **R1**, learns the default route and installs **192.168.1.1** (R1's IP address) as its default gateway for any unknown destinations.

```
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#ip name-server 192.168.122.1
R2(config)#do ping 8.8.8.8
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/22/28 ms
R2(config)#do ping www.google.com
Translating "www.google.com"...domain server (8.8.8.8) (192.168.122.1) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 142.250.189.164, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/37/60 ms
```

Step 6: *Save the configuration on both routers:*

6a. Save the configuration in router R1:

R2#write // Be sure to press Enter to confirm

6b. Save the configuration in router R2:

R2#write // Be sure to press Enter to confirm

Step 7: *Configure the IP address, gateway, and DNS server of the virtual PC:*

// Set the IP address of the PC1 to 192.168.2.2/24 with the default gateway of 192.168.2.1 (router R1 // interface f 0/1).

PC1> ip 192.168.2.2 255.255.255.0 192.168.2.1

PC1 : 192.168.2.2 255.255.255.0 gateway 192.168.2.1

// Set the default DNS server address to 192.168.122.1

PC1> ip dns 192.168.122.1

```
PC1> ping 1.1.1.1

84 bytes from 1.1.1.1 icmp_seq=1 ttl=52 time=32.594 ms
84 bytes from 1.1.1.1 icmp_seq=2 ttl=52 time=22.977 ms
84 bytes from 1.1.1.1 icmp_seq=3 ttl=52 time=36.271 ms
84 bytes from 1.1.1.1 icmp_seq=4 ttl=52 time=28.100 ms
84 bytes from 1.1.1.1 icmp_seq=5 ttl=52 time=39.139 ms
```

```
PC1> ip dns 192.168.122.1

PC1> show ip

NAME        : PC1[1]
IP/MASK     : 192.168.2.3/24
GATEWAY     : 192.168.2.1
DNS         : 192.168.122.1
MAC         : 00:50:79:66:68:01
LPORT       : 20018
RHOST:PORT  : 127.0.0.1:20019
MTU         : 1500

PC1> ping www.google.com
www.google.com resolved to 142.250.189.164

84 bytes from 142.250.189.164 icmp_seq=1 ttl=51 time=49.536 ms
84 bytes from 142.250.189.164 icmp_seq=2 ttl=51 time=47.376 ms
84 bytes from 142.250.189.164 icmp_seq=3 ttl=51 time=46.915 ms
84 bytes from 142.250.189.164 icmp_seq=4 ttl=51 time=32.422 ms
84 bytes from 142.250.189.164 icmp_seq=5 ttl=51 time=47.579 ms
```

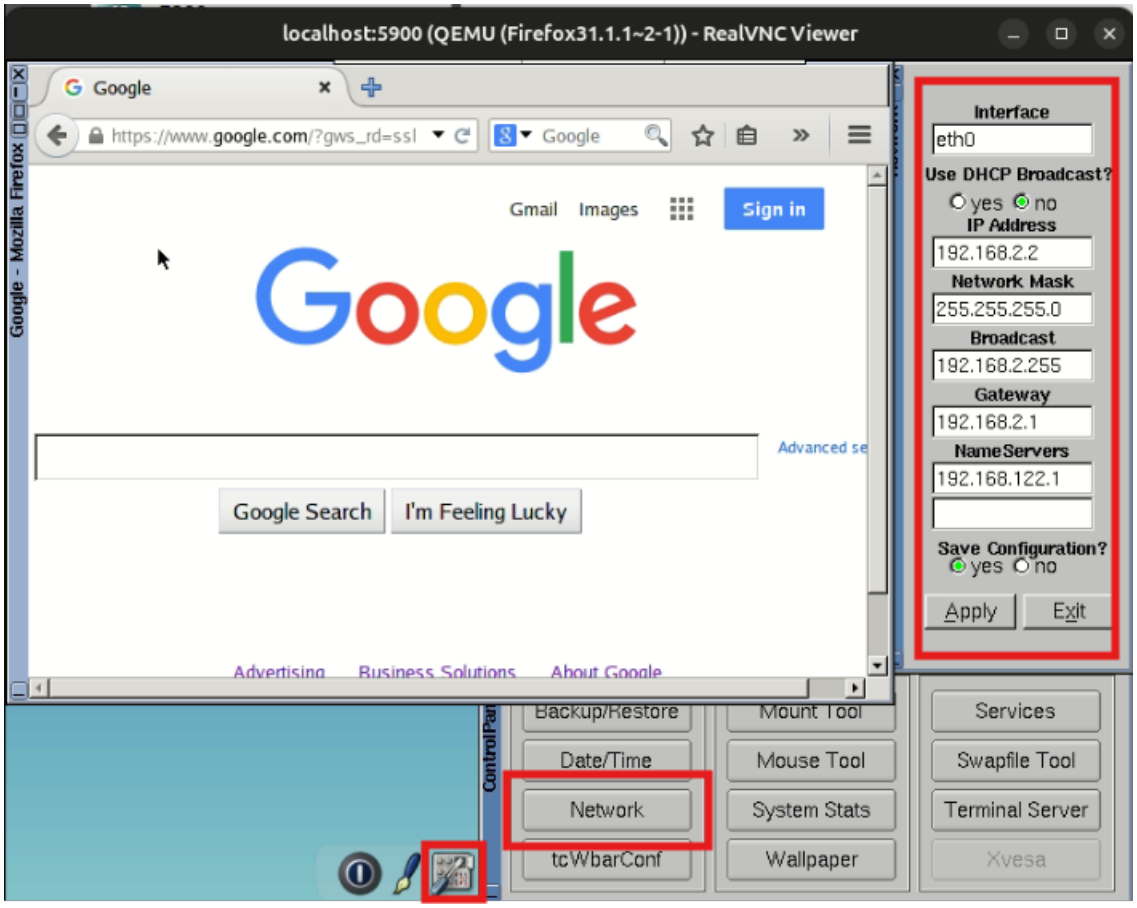

Figure 2: Configuring the Firefox VM

Step 8: *Configure the IP and default gateway of the Firefox VM as shown in Figure 5. Similarly, configure the IP and gateway of the Virtual PC. Try pinging server 8.8.8.8 (Google's DNS server). If the ping succeeds, you now have an OSPF network behind NAT that can communicate with the outside world.*

Why 192.168.122.1 Works as the DNS:

- **192.168.122.1** (the NAT router) is acting as a **DNS forwarder**, correctly routing DNS queries from your internal network to external DNS servers and sending the responses back. Since it's also handling NAT, it tracks and manages the DNS requests properly.

Why 1.1.1.1 or 8.8.8.8 Doesn't Work:

- When using public DNS servers directly, the NAT router likely isn't tracking or handling DNS queries correctly, possibly due to misconfigured NAT rules or missing routes. As a result, the DNS responses don't reach your internal network, causing resolution to fail.

- The NAT router may only be translating traffic that it directly manages (like queries to itself, **192.168.122.1**), and is not forwarding traffic correctly to external DNS servers like **8.8.8.8**.