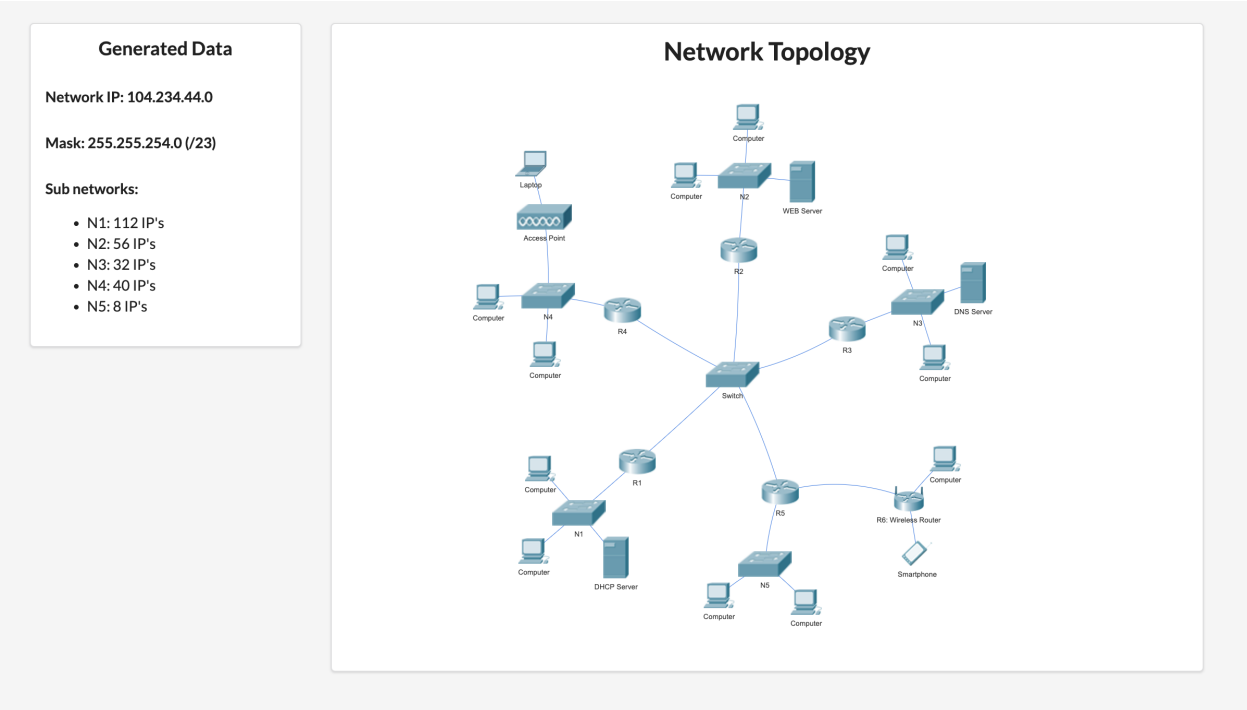
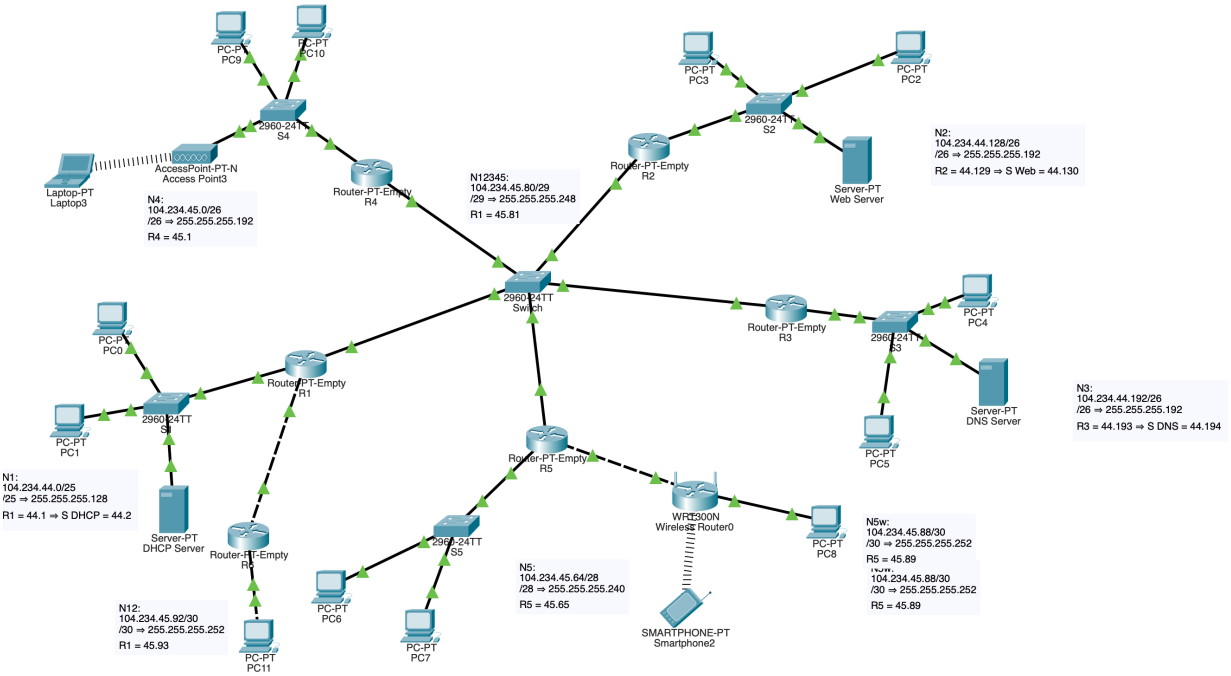
5 routers all switched up

Assignment



Solution



Network IP

104.234.44.0

Mask

Mask no	Decimal	Binary	IPs	NOT
(/20)	255.255.**240.0**	11111111.11111111.11110000.00000000	2^12 = 4096	0.0.00001111.255
(/21)	255.255.**248.0**	11111111.11111111.11111000.00000000	2^11 = 2048	0.0.00000111.255
(/22)	255.255.**252.0**	11111111.11111111.11111100.00000000	2^10 = 1024	0.0.00000011.255
(/23)	255.255.**254.0**	11111111.11111111.11111110.00000000	2^9 = 512	0.0.00000001.255
(/24)	255.255.**255.0**	11111111.11111111.11111111.00000000	2^8 = 256	0.0.000000010.255
(/25)	255.255.255.**128**	11111111.11111111.11111111.10000000	2^7 = 128	
(/26)	255.255.255.**192**	11111111.11111111.11111111.11000000	2^6 = 64	
(/27)	255.255.255.**224**	11111111.11111111.11111111.11100000	2^5 = 32	
(/28)	255.255.255.**240**	11111111.11111111.11111111.11110000	2^4 = 16	
(/29)	255.255.255.**248**	11111111.11111111.11111111.11110000	2^3 = 8	
(/30)	255.255.255.**252**	11111111.11111111.11111111.11111100	2^2 = 4	
(/31)	255.255.255.**254**	11111111.11111111.11111111.11111110	2^1 = 2	
(/32)	255.255.255.**255**	11111111.11111111.11111111.11111111	2^0 = 1	

IP addresses

For the net:

/m ⇒ 2^(32-m) IP addresses possible

For each subnet:

n devices (IPs) + 1 IP router + 1 IP NA + 1 IP BA = n + 3

Sub networks:

- N1: 112 + 3 = 115 IP's ≤ 128 = 2^7 (/25)
- N2: 56 + 3 = 59 IP's ≤ 64 = 2^6 (/26)
- N3: 32 + 3 = 35 IP's ≤ 64 = 2^6 (/26)
- N4: 40 + 3 = 43 IP's ≤ 64 = 2^6 (/26)
- N5: 8 + 3 = 11 IP's ≤ 16 = 2^4 (/28)

Other networks:

- N12345: 5 + 2 = 7 IPs ≤ 8 = 2^3 (/29)
- N5w: 2 + 2 = 4 IPs ≤ 4 = 2^2 (/30)

Total IP addresses:

115 + 59 + 35 + 43 + 11 + 7 + 4 = 274 ≤ 512 (/23)

Start IP

`(NA) = (IP) AND (NM)`

104.234. 44.0 AND

255.255.254.0

_____

104.234. 44.0 = NA

End IP

`(BA) = (IP) OR (NOT(NM))`

104.234. 44. 0 OR

0. 0. 1.255

_____

104. 234. 45.255 = BA

Recursive network split using a binary tree:

Recursive network split using intervals:

/23 ⇒ [0…511]

/24 ⇒ [0…255] [0……255]

/25 ⇒ [0…127][128…255] [0……127][128…255]

/26 ⇒ [0…63][64……127]

/27 ⇒ [64…95][96……127]

/28 ⇒ [96…111][112……127]

/29 ⇒ [112…119][120……127]

/30 ⇒ [120…123][124…127]

/23 ⇒ [0…511]

/24 ⇒ [0…255] [256……511]

/25 ⇒ [0…127][128……255] [0……127] [128…255]

/26 ⇒ [128…… 191][192…255]

/27 ⇒ [0…63][64…127][128…191

[64…95][96……..127]

/28 ⇒ [96…111][112……127]

/29 ⇒ [112…119][120……127]

/30 ⇒ [120…123][124…127]

Enumerating the networks:

- **N1:**

  104.234.44.0/25

  /25 ⇒ 255.255.255.128

  R1 = 44.1 ⇒ S DHCP = 44.2

- **N2:**

  104.234.44.128/26

  /26 ⇒ 255.255.255.192

  R2 = 44.129 ⇒ S Web = 44.130

- **N3:**

  104.234.44.192/26

  /26 ⇒ 255.255.255.192

  R3 = 44.193 ⇒ S DNS = 44.194

- **N4:**

  104.234.45.0/26

  /26 ⇒ 255.255.255.192

  R4 = 45.1

- **N5:**

  104.234.45.64/28

  /28 ⇒ 255.255.255.240

  R5 = 45.65

- **N12345:**

  104.234.45.80/29

/29 ⇒ 255.255.255.248

R5 = 45.81

- **N5w:**

  104.234.45.88/30

  /30 ⇒ 255.255.255.252

  R5w = 45.89

Router

- add -1CFE to one
- duplicate it
- Fast Ethernet 0/0 → On → first IP address and subnet mask (IP static)
- La nivelul de #router faci `copy running-config startup-config`

DHCP on a router

In order to configure a DCHP service on a router you need to setup a dhcp pool, define its range and parameters and excluded IPs.

The necessary commands are (from config mode):

```
enable
config terminal or conf t
# in configure mode:
ip route ….

#define a dhcp pool of addresses to be delivered
ip dhcp pool <name_of_pool>

# e.g. ip dhcp pool lan

# define the network range
network 192.168.0.0 255.255.255.0

# define the default gateway (if any) that should be passed to the clients
default-router 192.168.0.1

# define the DNS server (if any) that should be passed to the clients
dns-server 192.168.0.3

# exit dhcp pool configuration
exit

# If there any IPs in that range that you do not want to be served to PCs - add them to the excluded range:
ip dhcp excluded-address 192.168.0.1      (for a single IP)

# or (for a range of IPs)
ip dhcp excluded–address 192.168.0.1 192.168.0.10
```

Server

- IP configuration → IP static → second IP address, subnet mask, default gateaway = router, DNS server = IP of dns in network
- no Server ⇒ static IP

DHCP

- on DCHP service
- default gateaway = router, DNS server = IP of dns in network
- punem de la ce IP uri sa inceapa sa dea automat
- salvam!!

Web Server

- il putem face DHCP
- on HTTP service → ON → editam index.html → putem accesa la adresa IP a web serverului

DNS

- il putem face DHCP
- on DNS service → ON → give it a name like exam.com → connect it to the IP of the web server → SAVE

Access Point

- Setezi pe port 1 la ssid un nume oarecare
- La laptop bagi placa de wireless (prima) setezi pe port wireless la ssid numele dat mai sus

Wireless Router

- connectam pe portul de INTERNET al routerului (CROPPER)
- Settings → Internet → Static

Routing

Static:

Easy, just connect everything in the middle network.

RIP:

In each Router's CLI run:

```
exit
exit
exit
enable
configure terminal or configure t
router rip
?
version 2 or ver 2
no auto-summary
network [na of N1]
network [na of N12]
network [na of N12345]
```

```
command z
show ip route
```

Router - Router - Network:

- leftmost router has a static route

  0.0.0.0

  0.0.0.0

  [ip next router from shared network]

NAT

In order to config NAT on a router you have to specify:

- **one or multiple *inside (local LAN)* interfaces**
- **one or multiple *outside (WAN)* interfaces**

After setting up NAT all packets travelling from an inside interface to an outside interface are NAT-ted (their IP addresses are changed according to the NAT policy in place).

Suppose in our case that FastEthernet 0/0 (192.168.0.0/24 range) is inside and Serial 0/0/0 (193.226.40.1) is outside.

In order to accomplish NAT we do the following:

Router:

```
enable
conf t

# specify that this is an inside interface, the interface needs to have an IP Address
interface FastEthernet 0/0
ip nat inside

exit

# define Serial 0/0/0 as WAN (outside) interface
interface Serial 0/0/0
ip nat outside

exit
```

Define an Access list with the addresses from the inside that can be nat-ted.

The 0.0.0.31 specify the masks of bits from the IP Address that can vary. In our example bellow all addresses between 192.168.0.1 – 192.168.0.31 would pass!

```
# these are simple one liner lists

access-list 1 permit 192.168.0.1 0.0.0.31

# or extended lists that are defined as lists of rules – these allow the actions where they are going to be applied from source

# (192.168.0.0 0.0.0.255 -equivalent to 192.168.0.0/24 to destination 193.231.20.0/24)

ip access-list extended nat-internet

permit ip 192.168.0.0 0.0.0.255 193.231.20.0 0.0.0.255

permit ip 192.168.1.0 0.0.0.255 193.231.20.0 0.0.0.255
```

```
permit ip 192.168.2.0 0.0.0.255 193.231.20.0 0.0.0.255

permit ip 192.168.3.0 0.0.0.255 193.231.20.0 0.0.0.255

# define a pool of addresses to be allocated to the clients when NAT-ted. First IP – last IP netmask for those IPs

ip nat pool ISP 193.226.40.1 193.226.40.1 netmask 255.255.255.252
```

Define the NAT policy. The NAT policy applies NAT by selecting a source and a NAT pool or single IP (which replace the private range)

Overload allows to use a single outside IP from the defined pool for multiple clients – by altering the port.

One port is allocated on that IP for each outgoing client.

Overload allows this behaviour.

ip nat inside source list 1 pool ISP overload

or

#choose an interface that will provide the public IP and you do not need to define a **pool**:

ip nat inside source list 1 interface Serial0/1/1 overload