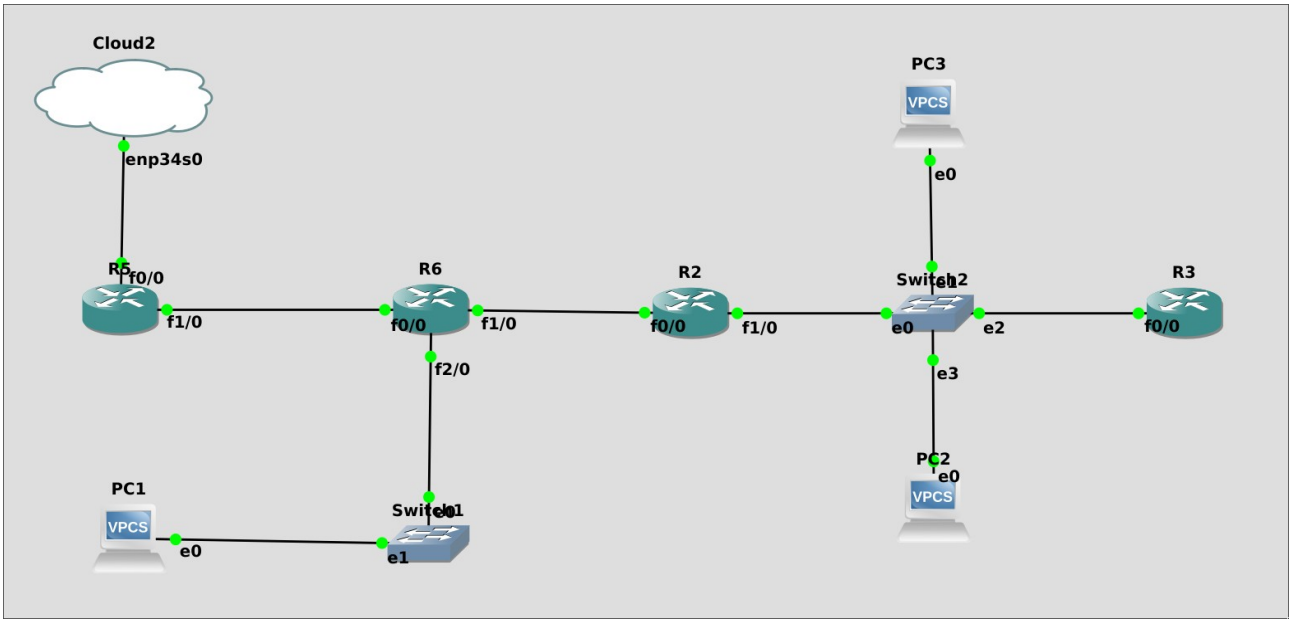
Za pomocą narzędzia GNS3 stworzyłem siec o poniższej topologii:



Przyjąłem założenie, że nadaję routerom adresy z puli 192.168.X.N, gdzie X to sieć, a N to numer routera. Maska podsieci to 155.155.155.0. Używany przeze mnie router to Cisco 7200 z portami GigabitEthernet.

Powyższa sieci wirtualna została połączona z zewnętrzną siecią Cloud poprzez router R1. Aby skonfigurować router, należy go najpierw włączyć, a następnie wejść w terminal. Przy użyciu polecenia conf t przechodzę w tryb zmiany ustawień urządzenia, a następnie za pomocą int <interfejs> w tryb konfiguracji interfejsu (np g0/0 czy g1/0).

Router R5 skonfigurowałem takimi poleceniami:

```
R5# conf t
R5(config)# int f0/0
R5(config-if)# ip address dhcp
R5(config-if)# ip nat outside
R5(config-if)# no shut
R5(config-if)# end
```

Ustawiłem tak, że router uzyskuje dynamiczny adres ip poprzez protokół DHCP. Oznaczam interfejs jako publiczny (ip nad outside).
Kolejnym krokiem było ustalenie serwera DNS, aby można było używać polecenia ping z nazwą hosta, a nie adresem IP:

```
R5# conf t
R5(config)# ip domain-lookup
R5(config)# ip name-server 8.8.8.8
R5(config)# end
```

Przechodzę teraz do konfugurowania sieci wewnętrznych. Konfiguruję interfejs łączący R5 z urządzeniami wewnętrznymi.
Poniżej przyznaję mu statyczny adres IP (wpisując ręcznie jego IP), oraz oznaczam ten interfejs jako przywatny:

```
R5# conf t
R5(config)# int f1/0
R5(config-if)# ip add 192.168.3.5 255.255.255.0
R5(config-if)# ip nat inside
R5(config-if)# no shut
R5(config-if)# end
```

Konfiguruję jeszcze routowanie zgodnie z protokołem RIP (wersja 2). Ustawiam tutaj, że R5 ma bezpośredni dostęp do sieci o podanych adresach:

```
R5# conf t
R5(config)# router rip
R5(config-router)# version 2
R5(config-router)# network 192.168.122.0
R5(config-router)# network 192.168.3.0
R5(config-router)# default-information originate
R5(config-router)# end
```

Sprawdzę teraz, czy router może pingować zewnętrzne adresy:

```
R5#ping google.com
Translating "google.com"...domain server (8.8.8.8) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 142.250.203.142, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/84/92 ms
```

Tabela routingu dla tego routera:

```
R5#show ip rip database
0.0.0.0/0    auto-summary
0.0.0.0/0    redistributed
    [1] via 0.0.0.0,
192.168.1.0/24    auto-summary
192.168.1.0/24
    [1] via 192.168.3.6, 00:00:49, GigabitEthernet1/0
192.168.2.0/24    auto-summary
192.168.2.0/24
    [2] via 192.168.3.6, 00:00:49, GigabitEthernet1/0
192.168.3.0/24    auto-summary
192.168.3.0/24    directly connected, GigabitEthernet1/0
192.168.4.0/24    auto-summary
192.168.4.0/24
    [1] via 192.168.3.6, 00:00:49, GigabitEthernet1/0
```

Na koniec jeszcze podaję routerowi R% listę sieci, z których pakiety może wysyłać do zewnętrznej sieci. Podaję tutaj adres sieci oraz maskę (w “odwróconej” postaci). Jeśli n-ty bit z takiej maski ma wartość 0,. to adres zostanie zaakceptowany tylko wtedy, gdy jego n-ty bit zgadza się z adresem podsieci.

```
R5# conf t
R5(config)# access-list 10 permit 192.168.1.0 0.0.0.255
R5(config)# access-list 10 permit 192.168.2.0 0.0.0.255
R5(config)# access-list 10 permit 192.168.3.0 0.0.0.255
R5(config)# access-list 10 permit 192.168.4.0 0.0.0.255
R5(config)# ip nat inside source list 10 interface f0/0 overload
R5(config)# end
```

Dla pozostałych routerów konfiguracja przebiega bardzo podobnie, chociaż oczywiście nie musimy ich łączyć poprzez dhcp z cloud oraz ustawiać access-listy. Przykładowa konfiguracja jednego z routerów (tutaj R2):

```
R2(config)# int f0/0
R2(config-if)# ip add 192.168.4.2 255.255.255.0
R2(config-if)# no shut
R2(config-if)# end

R2(config)# int f1/0
R2(config-if)# ip add 192.168.2.2 255.255.255.0
R2(config-if)# no shut
R2(config-if)# end

R2(config)# ip domain lookup source-interface f0/0
R2(config)# ip name-server 8.8.8.8

R2(config)# router rip
R2(config-router)# version 2
R2(config-router)# network 192.168.4.0
R2(config-router)# network 192.168.2.0
R2(config-router)# end
```

Switche nie wymagały żadnej konfiguracji. W komputerach konfiguracja polegała na dopisaniu inpormacji o adresie ip oraz masce, a także ustawienie serwera DNS. Np dla komputera PC3:

```
set pcname PC3
ip 192.168.2.33 192.168.2.3 24
ip dns 8.8.8.8
```

Sprobuję pingować poszczególne urządzenia:

pinguję z routera R3 do PC1

```
R3#ping 192.168.1.11
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.11, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/28/40 ms
R3#
```

do google.com:

```
R3#ping google.com
Translating "google.com"...domain server (8.8.8.8) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 142.250.203.206, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/40/40 ms
R3#
```

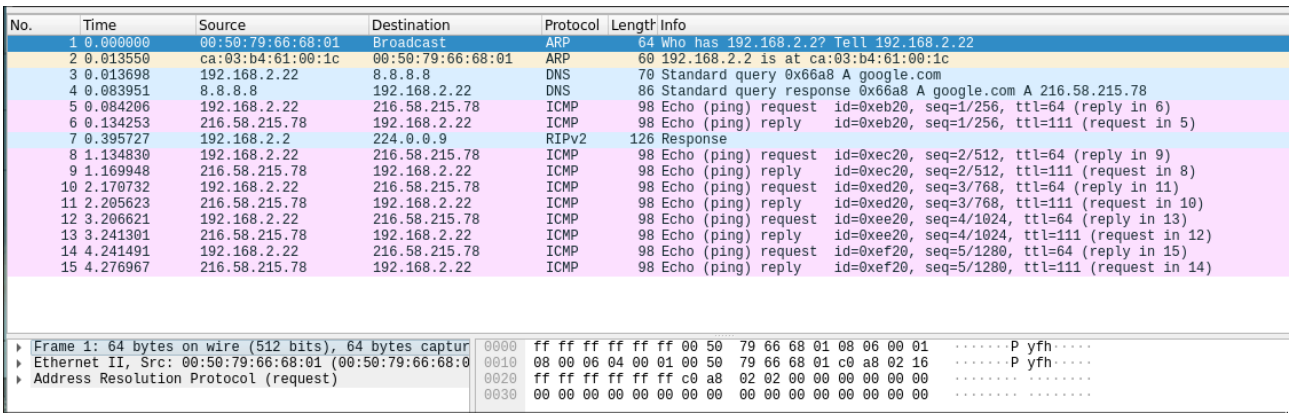
z komputera PC3 do PC1:

```
PC3> ping 192.168.1.11

Redirect Network, gateway 192.168.2.3 -> 192.168.2.2
84 bytes from 192.168.1.11 icmp_seq=1 ttl=62 time=28.006 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=62 time=25.011 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=62 time=25.254 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=62 time=25.490 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=62 time=24.581 ms
```

I do google.com:

```
PC3> ping google.com
google.com resolved to 142.250.203.206

84 bytes from 142.250.203.206 icmp_seq=1 ttl=111 time=39.970 ms
Redirect Network, gateway 192.168.2.3 -> 192.168.2.2
84 bytes from 142.250.203.206 icmp_seq=1 ttl=111 time=37.687 ms
```

Używając wireshark, ustawiłem przechwytywanie pakietów. Początkowo dla PC2:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	00:50:79:66:68:01	Broadcast	ARP	64	Who has 192.168.2.2? Tell 192.168.2.22
2	0.013550	ca:03:b4:61:00:1c	00:50:79:66:68:01	ARP	60	192.168.2.2 is at ca:03:b4:61:00:1c
3	0.013698	192.168.2.22	8.8.8.8	DNS	70	Standard query 0x66a8 A google.com
4	0.083951	8.8.8.8	192.168.2.22	DNS	86	Standard query response 0x66a8 A google.com A 216.58.215.78
5	0.084206	192.168.2.22	216.58.215.78	ICMP	98	Echo (ping) request id=0xeb20, seq=1/256, ttl=64 (reply in 6)
6	0.134253	216.58.215.78	192.168.2.22	ICMP	98	Echo (ping) reply id=0xeb20, seq=1/256, ttl=111 (request in 5)
7	0.395727	192.168.2.2	224.0.0.9	RIPv2	126	Response
8	1.134830	192.168.2.22	216.58.215.78	ICMP	98	Echo (ping) request id=0xec20, seq=2/512, ttl=64 (reply in 9)
9	1.169948	216.58.215.78	192.168.2.22	ICMP	98	Echo (ping) reply id=0xec20, seq=2/512, ttl=111 (request in 8)
10	2.170732	192.168.2.22	216.58.215.78	ICMP	98	Echo (ping) request id=0xed20, seq=3/768, ttl=64 (reply in 11)
11	2.205623	216.58.215.78	192.168.2.22	ICMP	98	Echo (ping) reply id=0xed20, seq=3/768, ttl=111 (request in 10)
12	3.206621	192.168.2.22	216.58.215.78	ICMP	98	Echo (ping) request id=0xee20, seq=4/1024, ttl=64 (reply in 13)
13	3.241301	216.58.215.78	192.168.2.22	ICMP	98	Echo (ping) reply id=0xee20, seq=4/1024, ttl=111 (request in 12)
14	4.241491	192.168.2.22	216.58.215.78	ICMP	98	Echo (ping) request id=0xef20, seq=5/1280, ttl=64 (reply in 15)
15	4.276967	216.58.215.78	192.168.2.22	ICMP	98	Echo (ping) reply id=0xef20, seq=5/1280, ttl=111 (request in 14)

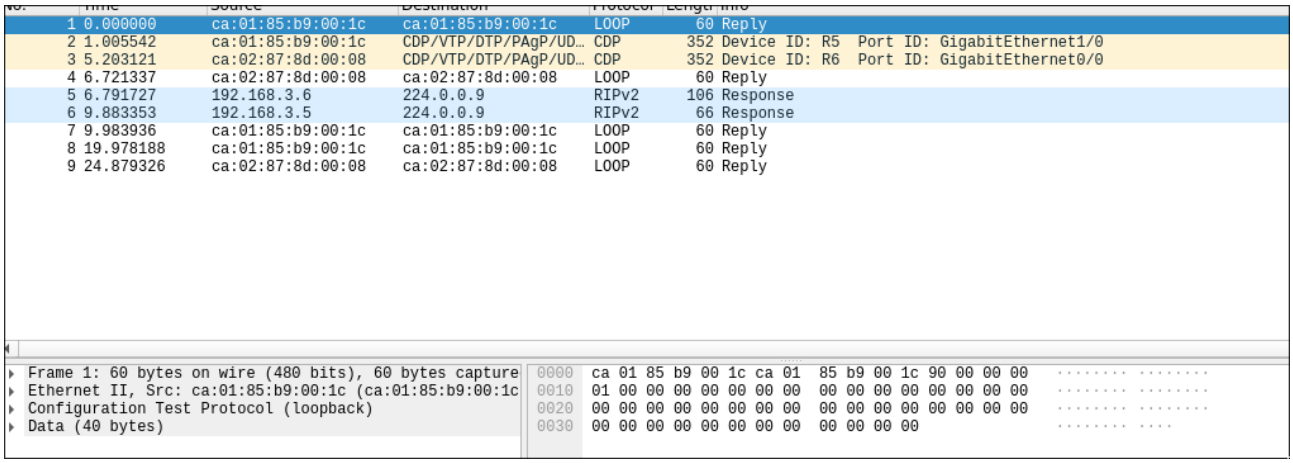
```
▸ Frame 1: 64 bytes on wire (512 bits), 64 bytes captur
▸ Ethernet II, Src: 00:50:79:66:68:01 (00:50:79:66:68:0
▸ Address Resolution Protocol (request)
```

```
0000  ff ff ff ff ff ff 00 50  79 66 68 01 08 06 00 01   .......P yfh.....
0010  08 00 06 04 00 01 00 50  79 66 68 01 c0 a8 02 16   .......P yfh.....
0020  ff ff ff ff ff ff c0 a8  02 02 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

```
PC2> ping google.com
google.com resolved to 216.58.215.78

84 bytes from 216.58.215.78 icmp_seq=1 ttl=111 time=50.138 ms
84 bytes from 216.58.215.78 icmp_seq=2 ttl=111 time=35.234 ms
84 bytes from 216.58.215.78 icmp_seq=3 ttl=111 time=34.988 ms
84 bytes from 216.58.215.78 icmp_seq=4 ttl=111 time=34.800 ms
84 bytes from 216.58.215.78 icmp_seq=5 ttl=111 time=35.589 ms
^C
```

Widać, że pakiety wysłane oraz otrzymane przez PC2 realizują protokół DNS oraz ICMP.

Przechwytywanie ustawione na 192.168.3.5:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	ca:01:85:b9:00:1c	ca:01:85:b9:00:1c	LOOP	60	Reply
2	1.005542	ca:01:85:b9:00:1c	CDP/VTP/DTP/PAgP/UD…	CDP	352	Device ID: R5 Port ID: GigabitEthernet1/0
3	5.203121	ca:02:87:8d:00:08	CDP/VTP/DTP/PAgP/UD…	CDP	352	Device ID: R6 Port ID: GigabitEthernet0/0
4	6.721337	ca:02:87:8d:00:08	ca:02:87:8d:00:08	LOOP	60	Reply
5	6.791727	192.168.3.6	224.0.0.9	RIPv2	106	Response
6	9.883353	192.168.3.5	224.0.0.9	RIPv2	66	Response
7	9.983936	ca:01:85:b9:00:1c	ca:01:85:b9:00:1c	LOOP	60	Reply
8	19.978188	ca:01:85:b9:00:1c	ca:01:85:b9:00:1c	LOOP	60	Reply
9	24.879326	ca:02:87:8d:00:08	ca:02:87:8d:00:08	LOOP	60	Reply

```
▸ Frame 1: 60 bytes on wire (480 bits), 60 bytes capture
▸ Ethernet II, Src: ca:01:85:b9:00:1c (ca:01:85:b9:00:1c
▸ Configuration Test Protocol (loopback)
▸ Data (40 bytes)
```

```
0000  ca 01 85 b9 00 1c ca 01  85 b9 00 1c 90 00 00 00   ........ ........
0010  01 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00              ........ ....
```