**Sprawozdanie z Listy 4 (Technologie Sieciowe)**

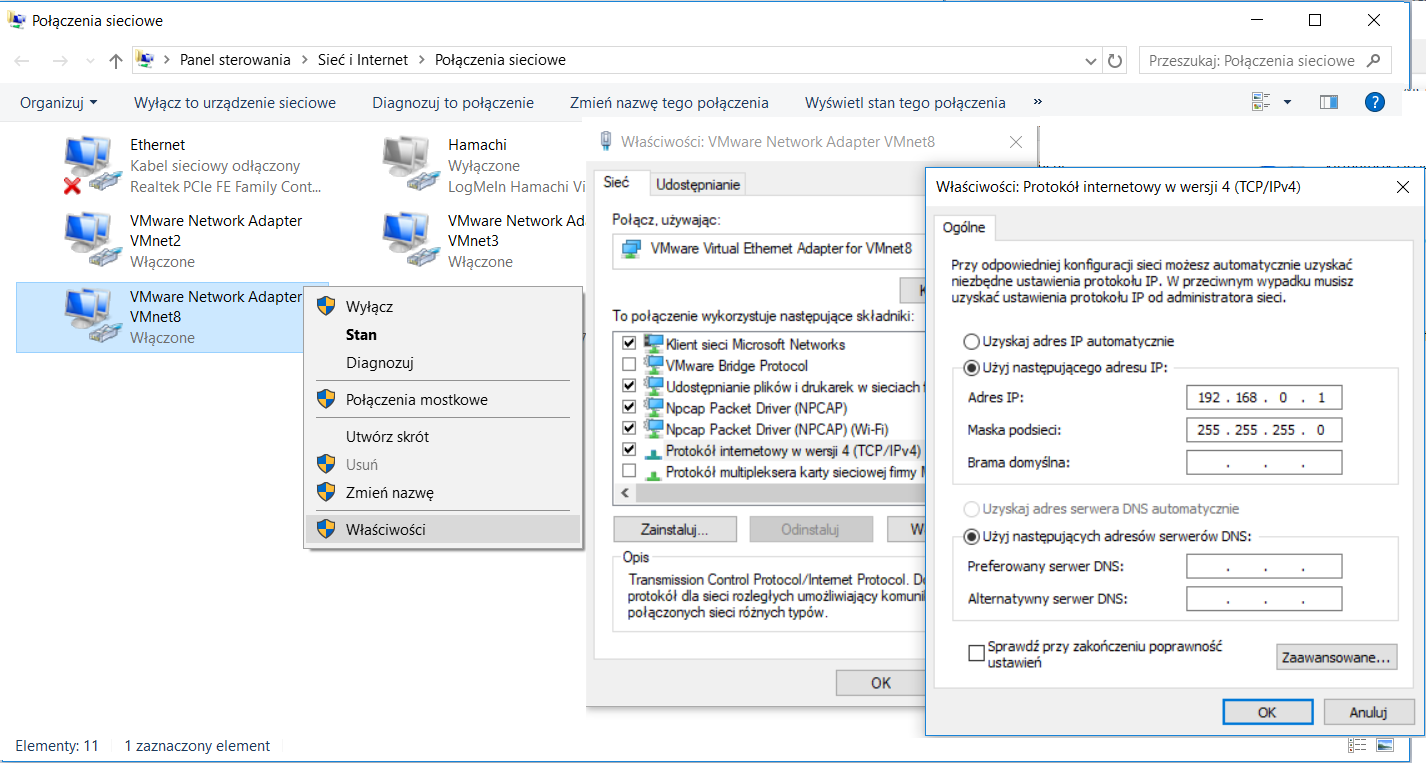
*Jakub Omieljaniuk (250090)*

Chcąc projektować i testować złożone sieci, możemy posłużyć się programem **GNS3**. Jest to narzędzie, napisane w języku Python, umożliwiające symulację realnie istniejącej sieci (bądź też taką, którą będziemy chcieli zbudować). W programie możemy zasymulować konkretne modele routerów (np. CISCO), a także komputery z wybranymi systemami operacyjnymi. GNS3 jest również dedykowane do pracy z Wiresharkiem, z którym mieliśmy przyjemność obcować w ramach Listy 1.   
Analiza zaprojektowanej w GNS3 sieci poprzez Wiresharka pozwala na wygodne analizowanie ruchu krążących w niej pakietów.

Do zaprojektowania swojej sieci posłużę się dedykowaną wirtualną maszyną (korzystając   
z dodatkowego programu **VMware Workstation**). Pozwoli mi ona połączyć projektowaną sieć   
z Internetem. Skonfigurujmy zatem adres IP karty sieciowej maszyny wirtualnej.   
W systemie Windows 10 należy w tym celu wejść w panel sterowania:

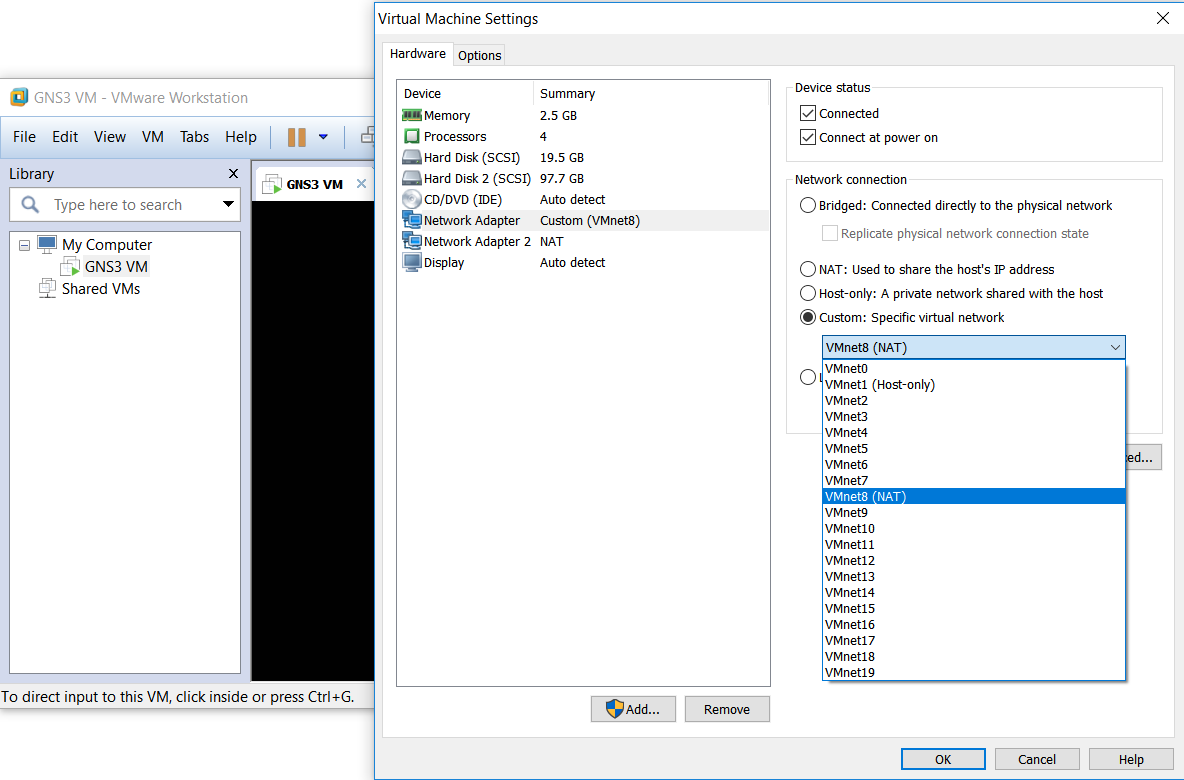
*Panel sterowania > Sieć i Internet > Połączenia sieciowe*

Następnie we „Właściwościach” karty sieciowej, którą skojarzymy z maszyną wirtualną (w moim przypadku Vmware Network Adapter VMnet8) i z listy składników wejść w *ustawienia „Protokół internetowy w wersji 4 (TCP/IPv4)”*. Tam ustawiamy IP i maskę podsieci.



1. Ustawienie adresu IP (192.168.0.1) dla karty sieciowej maszyny wirtualnej

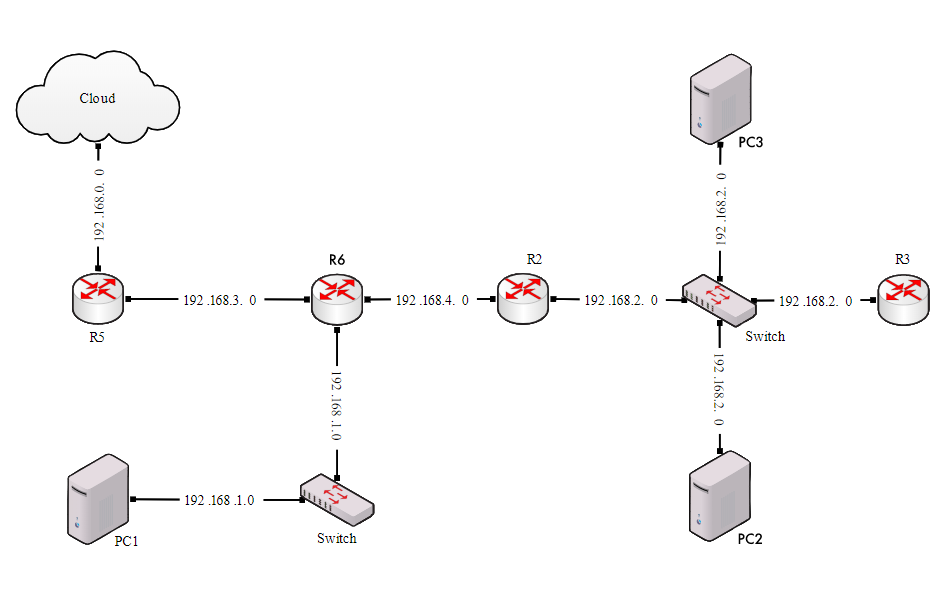
Aby wskazać maszynie wirtualnej, której karty sieciowej ma używać, wchodzimy w programie VMware Workstation w ustawienia wirtualnej maszyny i ustawiamy odpowiednie połączenie internetowe:

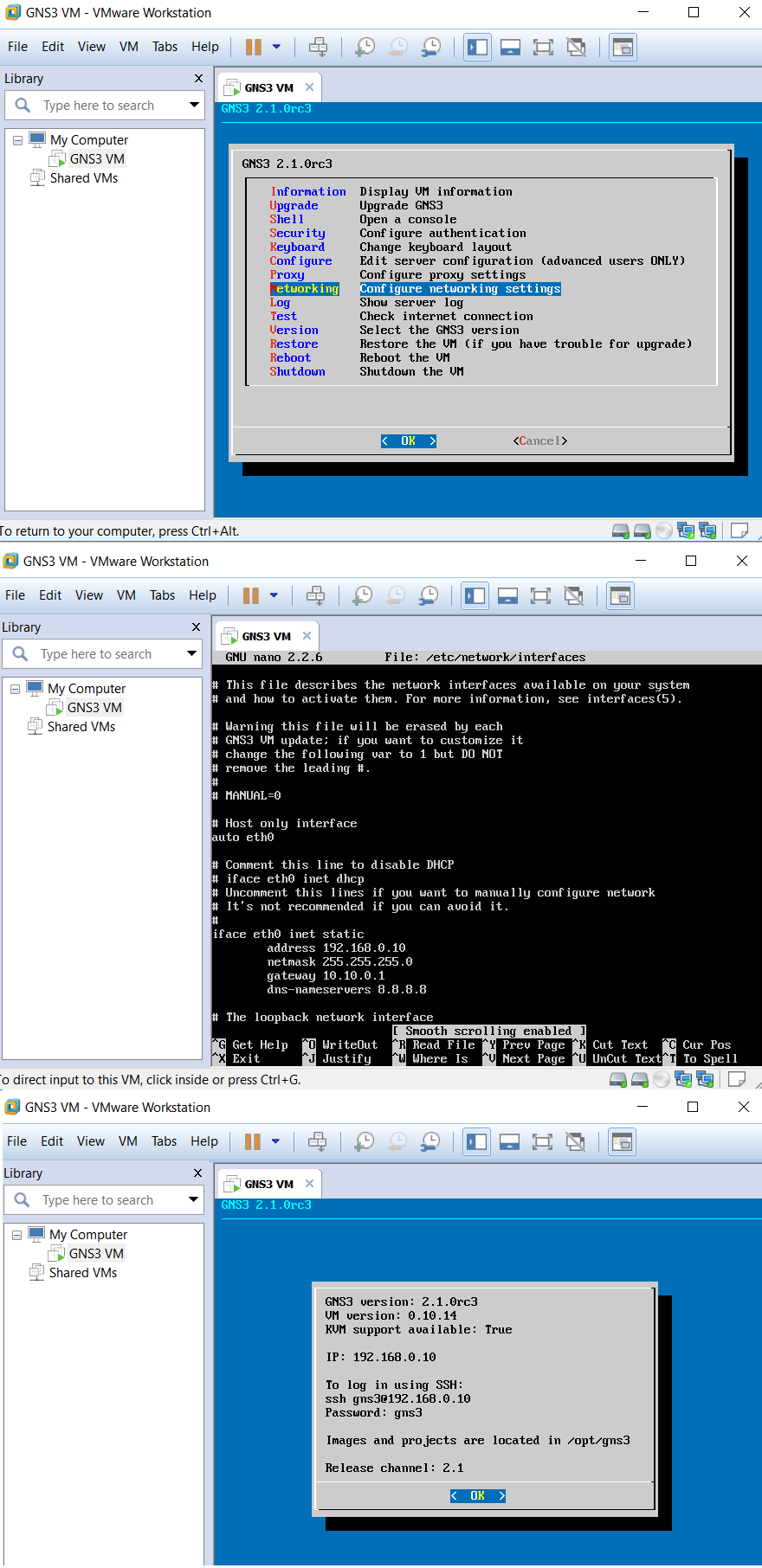


2. Ustawienie karty sieciowej dla maszyny wirtualnej

W przypadku, gdy maszyna wirtualna nie przydzieli sobie automatycznie adresu IP, należy ustawić go poprzez wybranie opcji „Configure networking settings” i edycji pliku interfaces (rysunek 3.).

Oto projekt sieci, którą będziemy chcieli stworzyć:

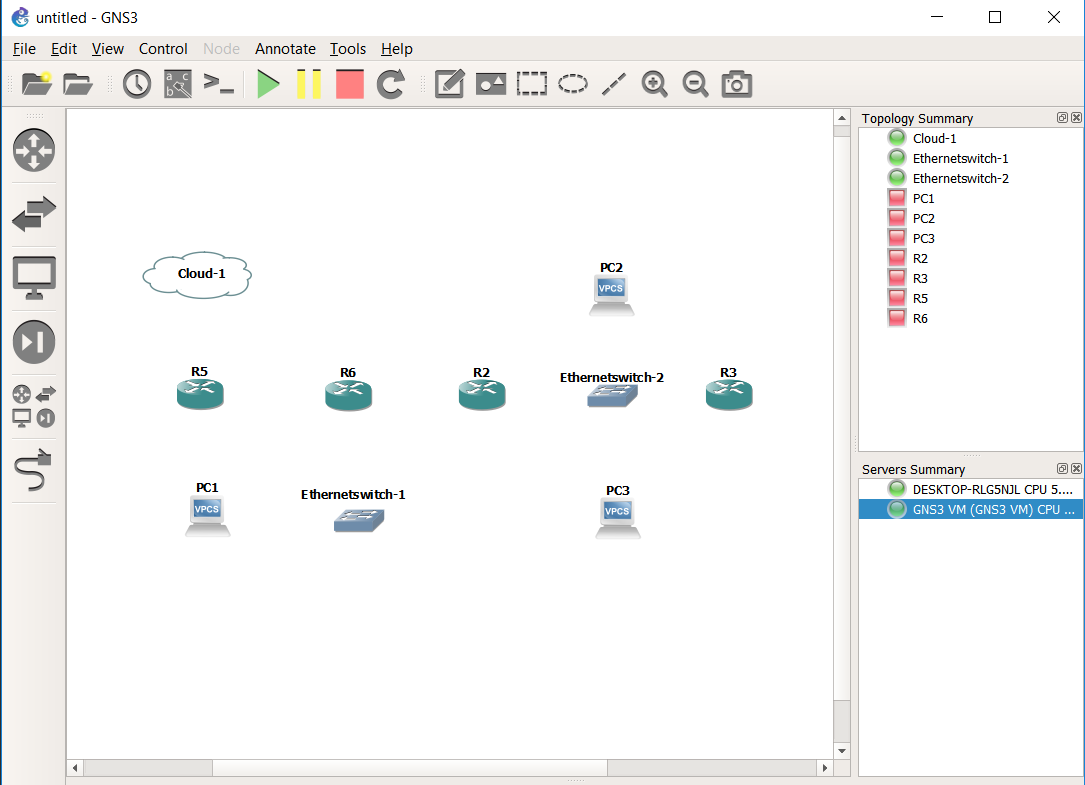




3. Ustawienie adresu IP dla maszyny wirtualnej

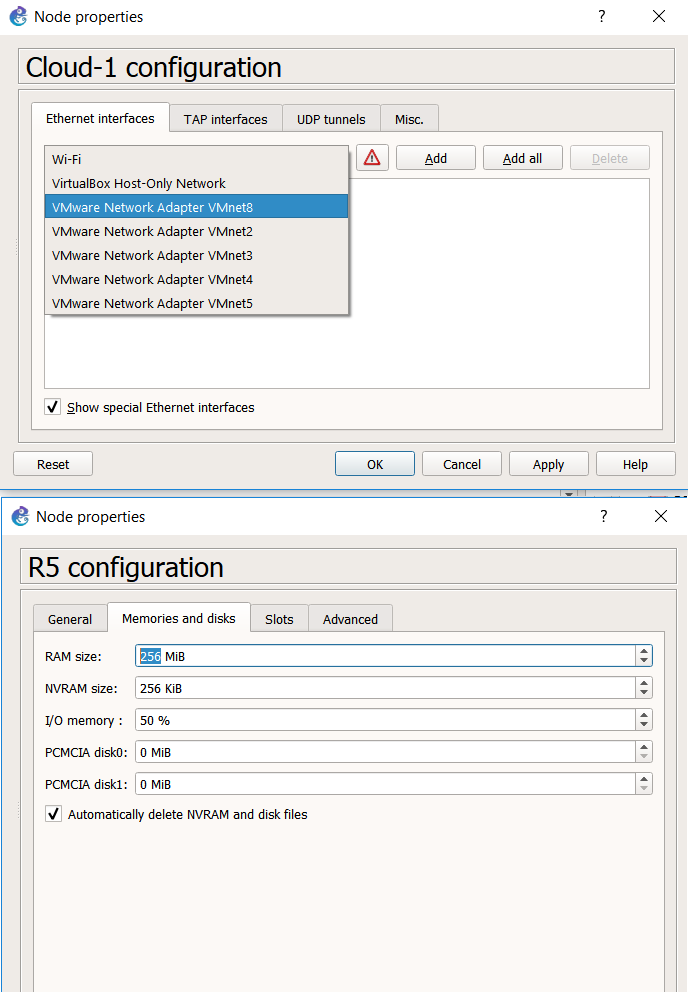
Budowę sieci rozpoczniemy od dodania wszystkich urządzeń występujących w projekcie:

1. Z pola „End devices” (ikonki po lewej stronie) wybieramy „Cloud”.
2. Z pola „Routers” wybieramy pobrany z Internetu obraz routera CISCO „c3725”. Dodajemy 4 takie routery.
3. Z pola „Switches” dodajemy 2 razy „Ethernet switch”.
4. Z pola „End devices” dodajemy 3 razy „VPCS”.



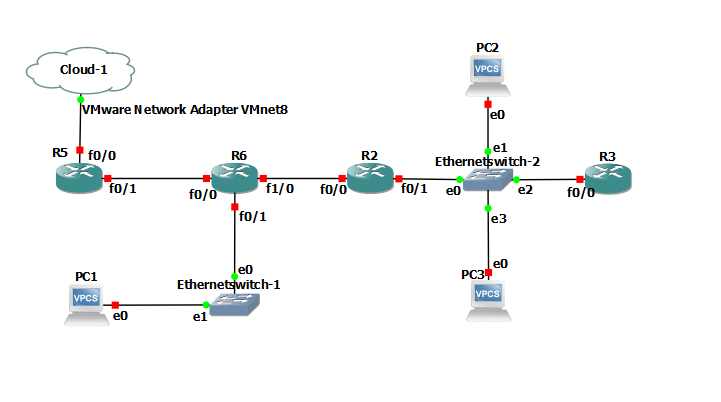
4. Dodanie urządzeń do sieci

W konfiguracji „Cloud-1” dodajmy wcześniej spreparowany interfejs karty sieciowej, przez którą nasza sieć będzie się łączyć z Internetem. Natomiast w routerze 5 zwiększamy dostępną pamięć, aby router był wstanie obsłużyć NAT1:



5. Konfiguracja ustawień chmury i routera R5

W następnej kolejności dodajemy połączenia pomiędzy urządzeniami. Za pomocą ostatniej ikonki po lewej stronie „Add a link”, wybieramy odpowiednie interfejsy sieciowe:



Klikając PPM na R5, możemy najpierw uruchomić router, a następnie włączyć jego konsolę, gdzie dokonamy jego konfiguracji. W tym celu będziemy wpisywać następujące komendy:

R5# configuration terminal (skrócona forma*: conf t*)

R5(config)# interface FastEthernet 0/0 (*int f0/0*)

R5(config-if)# ip address dhcp (*ip add dhcp*)

R5(config-if)# no shutdown (*no shut*)

R5(config-if)# end

W odpowiedzi powinniśmy otrzymać informację o przydzielonym adresie IP:

**R5#**

**\*Mar 1 00:01:01.715: %DHCP-6-ADDRESS\_ASSIGN: Interface FastEthernet0/0 assigned DHCP address 192.168.72.132, mask 255.255.255.0, hostname R5**

Teraz dodamy do routera adres domyślnego serwera DNS, do którego będzie on wysyłał zapytania o zamianę adresów mnemonicznych (np. google.com) na adresy IP (216.58.215.110).

R5(config)# ip domain-lookup

R5(config)# ip name-server 8.8.8.8

**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 172.217.16.14, timeout is 2 seconds:**

**!!!!!**

**Success rate is 100 percent (5/5), round-trip min/avg/max = 20/32/44 ms**

Dostaliśmy odpowiedź na wszystkie 5 pakietów ICMP wysłanych do serwera google.com, zatem nasz router został poprawnie skonfigurowany dla interfejsu FastEthernet 0/0. Teraz skonfigurujemy interfejs f0/1, przez który router R5 będzie komunikować się z routerem R6:

R5# conf t

R5(config)# int f0/1

R5(config-if)# ip address 192.168.3.1 255.255.255.0

W konsoli routera R6 nadajemy mu statycznie IP i przypisujemy bramę sieciową na adres interfejsu f0/1 routera R5:

R6# conf t

R6(config)# int f0/0

R6(config-if)# ip add 192.168.3.2 255.255.255.0

R6(config-if)# no shut

R6(config-if)# end

R6(config)# ip route 0.0.0.0 0.0.0.0 192.168.3.1

Konfigurujemy routing pomiędzy R5 i R6:

R6(config-router)# network 192.168.0.0 0.255.255.255 area 0

R5(config-router)# network 192.168.0.0 0.255.255.255 area 0

R5(config-router)# default-information originate

Powinniśmy otrzymać komunikat o nawiązaniu połączenia w obu konsolach:

**R5# %OSPF-5-ADJCHG: Process 1, Nbr 192.168.3.2 on FastEthernet0/1 from LOADING to FULL, Loading Done**

**R6# %OSPF-5-ADJCHG: Process 1, Nbr 192.168.3.1 on FastEthernet0/0 from LOADING to FULL, Loading Done**

Aby router R6 miał dostęp do Internetu, musimy skonfigurować NAT1 dla interfejsów routera R5 i dodać stworzoną sieć lokalną do listy dozwolonych sieci:

R5(config)# int f0/0

R5(config-if)# ip nat outside

R5(config)# int f0/1

R5(config-if)# ip nat inside

R5(config)# ip nat inside source list 1 int f0/0 overload

R5(config)# access-list 1 permit 192.168.0.0 0.255.255.255

***1NAT*** *– ang. Network Address Translation, technika umożliwiająca dostęp do Internetu wielu hostom w prywatnej sieci lokalnej, poprzez zamianę ich adresów prywatnych na jeden adres publiczny IP (brama sieciowa).*

Konfiguracja komputerów polega na przypisaniu im adresu IP, bramy sieciowej i serwera DNS.   
Ze względu na późniejsze problemy z połączeniem z Internetem przez VPCS, zgodnie z zaleceniem wykładowcy zamieniłem je na routery. Konfiguracja zatem przebiega analogicznie jak przy połączeniu routerów R5 i R6. Zmieniają się adresy kolejnych sieci, które później dodajemy również do routera R5:

R5(config)# access-list 1 permit 192.168.1.0 0.255.254.255

R5(config)# access-list 1 permit 192.168.2.0 0.255.253.255

R5(config)# access-list 1 permit 192.168.3.0 0.255.252.255

R5(config)# access-list 1 permit 192.168.4.0 0.255.251.255

R5(config)# ip nat inside source list 10 interface f0/0 overload

R5(config)# end

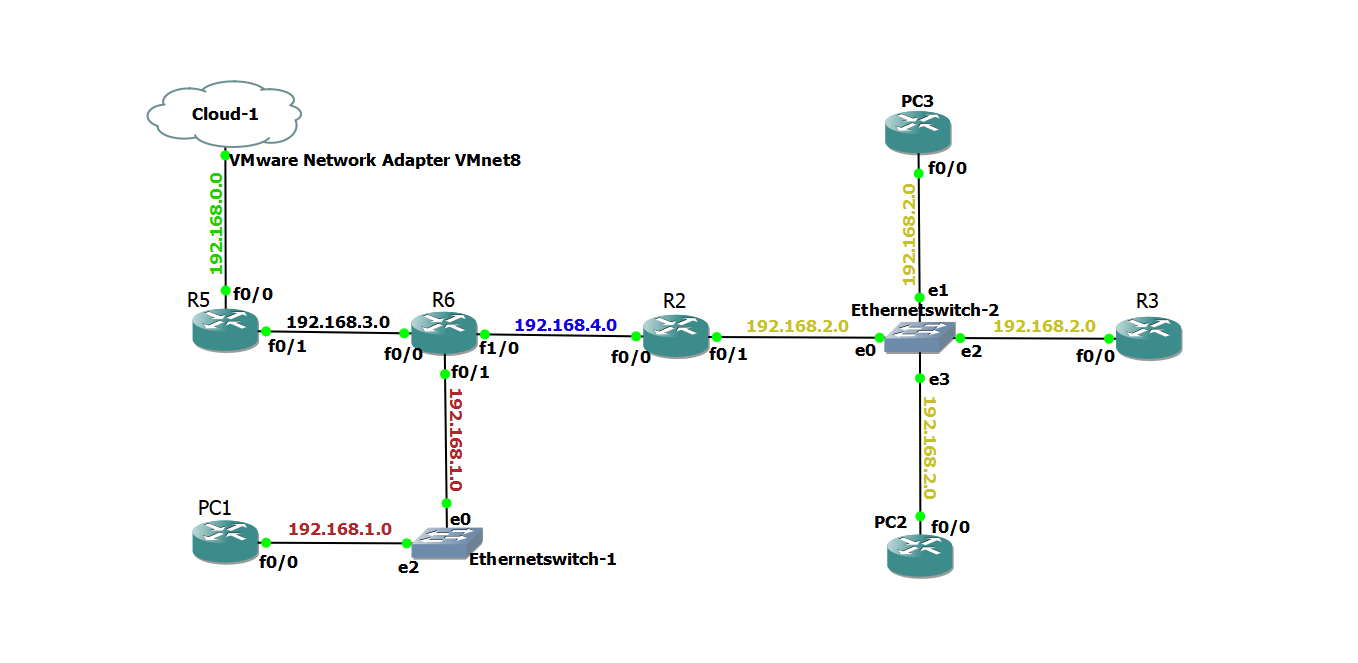
R5# write

Część konfiguracji PC2 (najbardziej istotne komendy):

PC2(config-if)# ip address 192.168.2.27 255.255.255.0

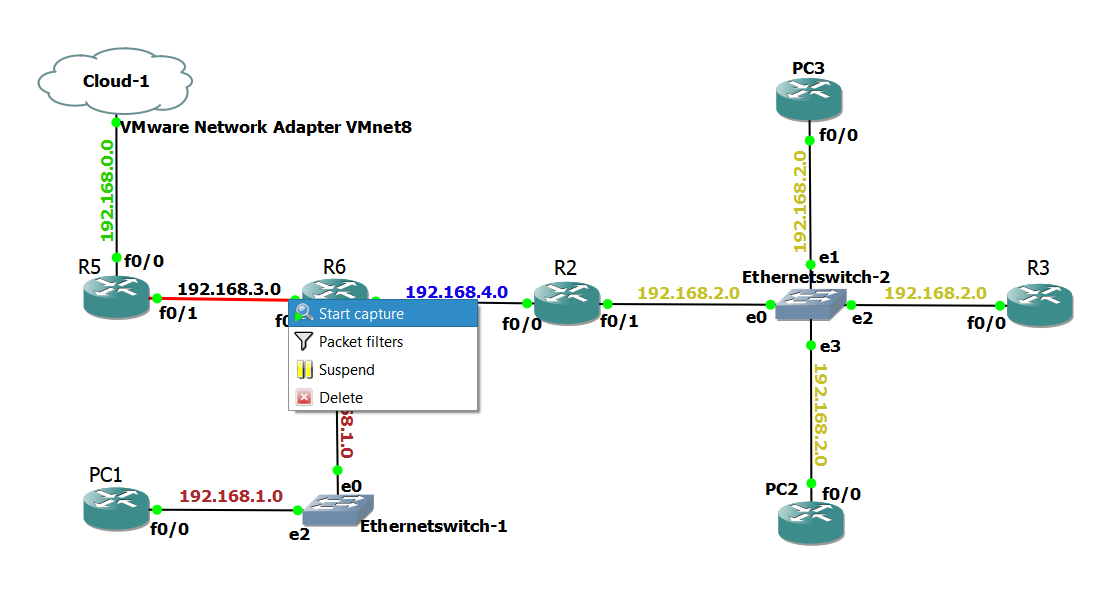
PC2(config)# ip route 0.0.0.0 0.0.0.0 192.168.2.1

PC2(config)# ip name-server 8.8.8.8

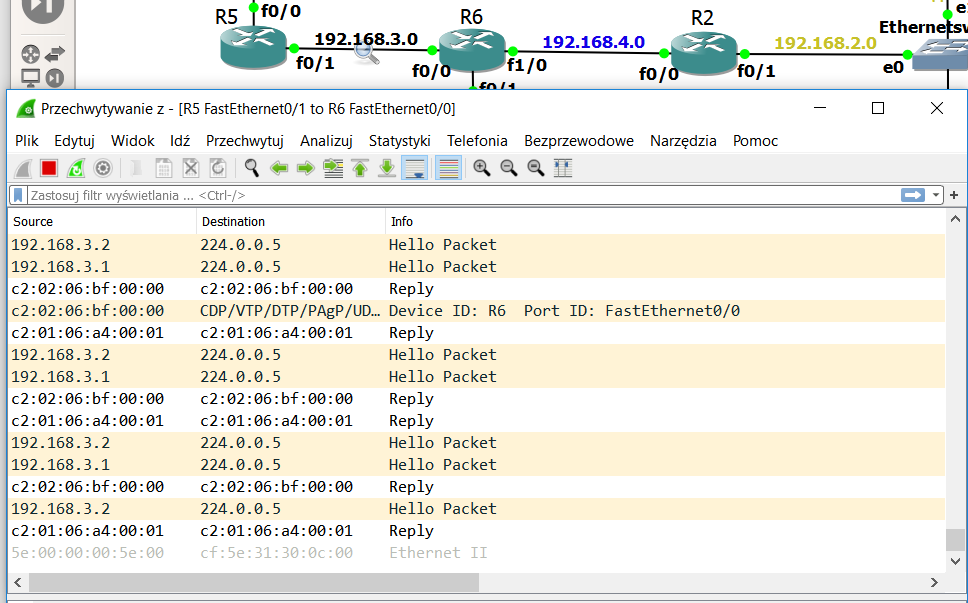
Schemat gotowej, skonfigurowanej sieci:

6. Schemat składający się z 5 sieci

Aby zbadać ruch pakietów w wybranej sieci, klikamy PPM na wybrane połączenie i wybieramy opcję „Start capture”:

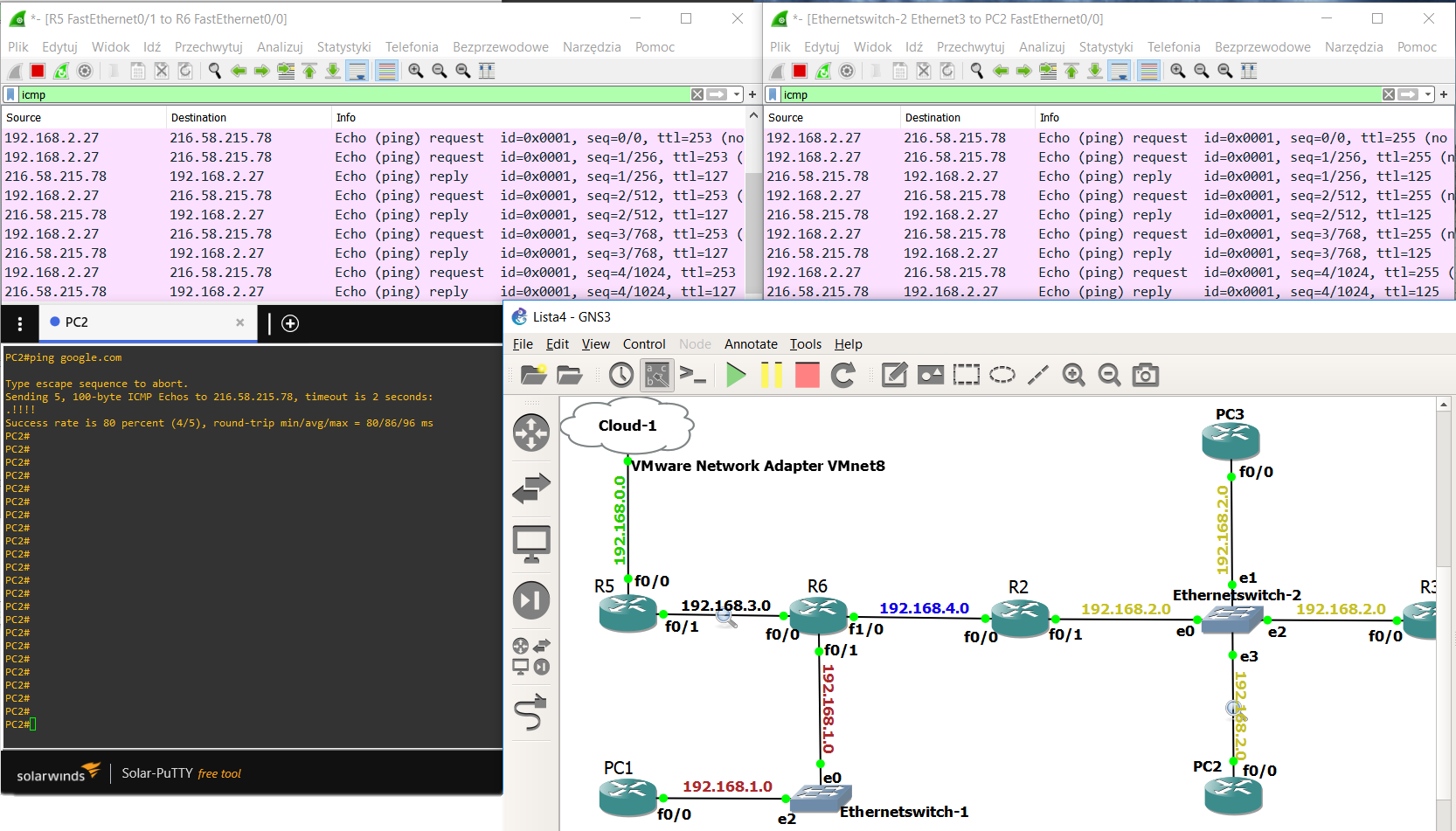


Po chwili uruchomi się nam program Wireshark z ustawionym przechwytywaniem pakietów   
w sieci 192.168.3.0:



7. Program Wireshark przechwytujący pakiety w wybranej sieci

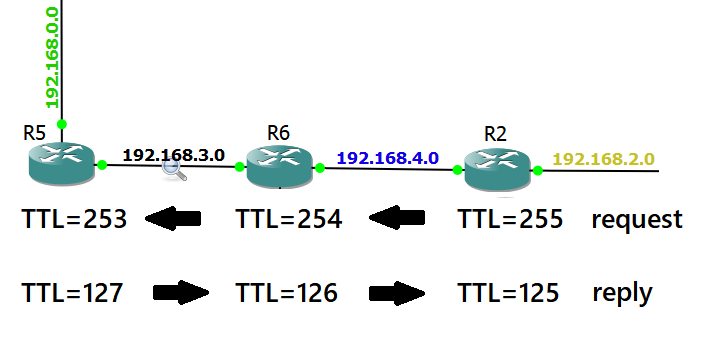
Zbadajmy zapytanie ping google.com wysłane z PC2 (192.168.2.27) w sieciach 192.168.3.0   
i 192.168.2.0. W programie Wireshark wprowadziłem filtr wyświetlający tylko pakiety ICMP:



8. Śledzenie pakietów ICMP

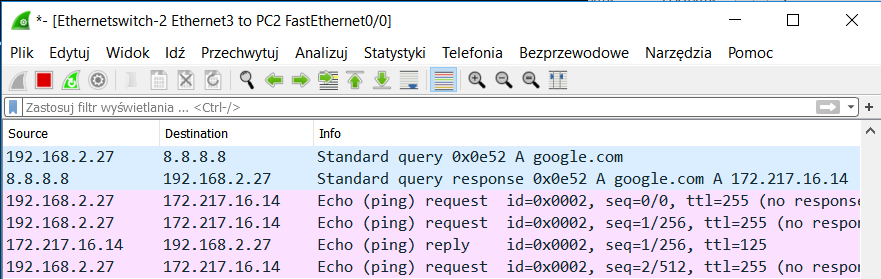
W konsoli widać, że nie dostaliśmy odpowiedzi na pierwszy z wysłanych pakietów, co pokrywa się z wykrytymi przez Wiresharka pakietami IMCP Echo Reply. W obu sieciach wykryto ich 4 na 5 pakietów ICMP Echo Request.

Dodatkowo widać także, że TTL w sieci 192.168.3.0 jest o 2 jednostki mniejszy w przypadku pakietów wychodzących od PC2   
i 2 jednostki większy w przypadku pakietów zaadresowanych do PC2. Jest to spowodowane tym, że pomiędzy obiema sieciami mamy routery R5, R6 i R2:



9. Zmiana wartości TTL dla pakietów ICMP

Gdy w Wiresharku usuniemy filtr na komunikaty ICMP i ponownie wywołamy ping, zobaczymy również jak router uzyskuje adres IP serwera Googla (172.217.16.14) poprzez zapytanie do serwera DNS (8.8.8.8):



10. ping google.com