Lista 4

Technologie sieciowe

Patryk Majewski
250134

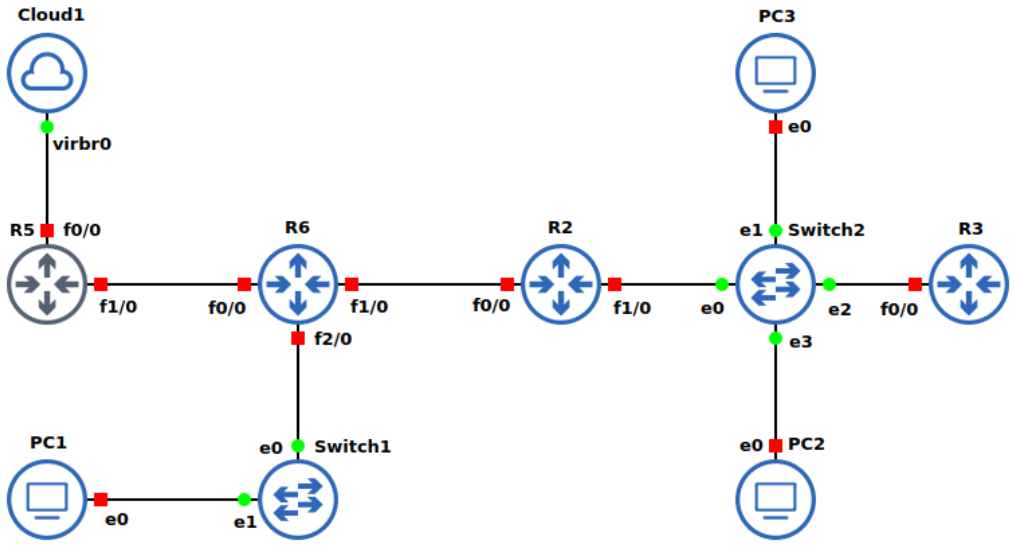
1 Opis zadania

Za pomocą programu GNS3 należy stworzyć sieć o podanej topologii, upewniając się, że:

- jest ona podłączona do prawdziwej, fizycznej sieci
- router będący bramą otrzymuje dynamicznie adres IP z tej sieci
- inne urządzenia mają statyczne adresy w swoich sieciach
- możliwe jest wysyłanie komunikatów ping pomiędzy dowolną parą urządzeń i na adres zewnętrzny

Następnie należy przeanalizować pakiety przechwycone w niektórych fragmentach sieci po wywołaniu programu ping w jednym z urządzeń.

2 Konfiguracja sieci



Rysunek 1: Zadana topologia sieci

Aby zachować konsekwencję i czytelność wyników, przyjmiemy konwencję nadawania adresów poszczególnym urządzeniom. Router $\texttt{R}_\texttt{N}$ otrzymuje w sieci `X` adres `192.168.X.N`. Komputerowi $\texttt{PC}_\texttt{N}$ w sieci `X` nadamy natomiast adres `192.168.X.(N+10)`. Maski podsieci przy nadawanych adresach ustalimy na `255.255.255.0`, ponieważ adresem podsieci w naszym modelu są trzy pierwsze oktety adresu IP. Wykorzystywanym w symulacji obrazem routera będzie Cisco 7200.

2.1 Połączenie z internetem

Nasza sieć uzyska dostęp do internetu poprzez element `Cloud`. Potrzebujemy w tym celu również routera brzegowego (R_5), który łączymy z `Cloud`, a połączenie konfigurujemy w następujący sposób:

```
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

Interface f0/0 assigned DHCP address 192.168.122.227, mask 255.255.255.0, hostname R5
```

Poleceniem `conf t` uruchamiamy tryb zmiany ustawień urządzenia, a następnie za pomocą `int` przechodzimy do konfiguracji wybranego interfejsu (w tym przypadku FastEthernet 0/0). Ustalamy, że adres IP urządzenia powinien być przydzielany naszemu routerowi przez sieć zgodnie z protokołem DHCP. Poleceniem `ip nat outside` oznaczamy interfejs jako publiczny, a następnie uruchamiamy go. Jak widać, router R_5 istotnie uzyskał swój adres z sieci.

Musimy jeszcze ustalić serwer DNS, dzięki któremu możliwe będzie pingowanie serwerów po ich domenach, a nie adresach. Uruchamiamy usługę wyszukiwania nazw poleceniem `ip domain-lookup`, a następnie ustawiamy adres DNS na publiczny serwer Google:

```
R5# conf t
R5(config)# ip domain-lookup
R5(config)# ip name-server 8.8.8.8
R5(config)# end
R5# ping cs.pwr.edu.pl
Translating "cs.pwr.edu.pl"...domain server (192.168.122.1) [OK]

Sending 5, 100-byte ICMP Echos to 156.17.7.22, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/52/72 ms
```

2.2 Wewnętrzne sieci

Kontynuujemy konfigurację routera R_5, przygotowując interfejs łączący go z wewnętrznymi urządzeniami naszej sieci. Od tej strony nadajemy mu statyczny adres, a następnie oznaczamy interfejs jako prywatny (`ip nat inside`).

```
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
```

Musimy jeszcze skonfigurować routowanie zgodnie z protokołem RIP w wersji 2. Informujemy R_5, że ma bezpośredni dostęp do sieci o adresach `192.168.122.0` oraz `192.168.3.0`.

```
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
```

Po skonfigurowaniu całej sieci routery rozgłaszają swoje informacje dotyczące ścieżek. Wówczas tabela routingu dla R_5 prezentuje się następująco:

```
R5# show ip rip database

0.0.0.0/0    redistributed
    [1] via 0.0.0.0
192.168.1.0/24
    [1] via 192.168.3.6, 00:00:20, FastEthernet1/0
192.168.2.0/24
    [2] via 192.168.3.6, 00:00:20, FastEthernet1/0
192.168.3.0/24
    directly connected, FastEthernet1/0
192.168.4.0/24
    [1] via 192.168.3.6, 00:00:20, FastEthernet1/0
192.168.122.0/24
    directly connected, FastEthernet0/0
```

Jak widać, mamy dostęp do każdej z utworzonych podsieci. Podsieci 3 i 122 są bezpośrednio połączone z R_5. Do podsieci 1 i 4 dostaniemy się już po skoku do R_6. Aby dotrzeć do podsieci 2, po skoku do R_6 będziemy musieli wykonać kolejny.

Polecenie `default-information originate` powoduje, że wraz z innymi informacjami, R_5 będzie rozgłaszać domyślną ścieżkę do zewnętrznej sieci. Dzięki temu na przykład dla R_2 mamy:

```
R2# show ip route
Codes: R  - RIP, * - candidate default

R*    0.0.0.0/0 [120/2] via 192.168.4.6, 00:00:17, FastEthernet0/0
```

Ostatnim krokiem inicjalizacji routera R_5 jest wskazanie mu listy sieci, z których pakiety ma przepuszczać na zewnątrz. Podajemy bazowy adres akceptowanej podsieci oraz "odwróconą maskę". Jeśli n-ty bit takiej maski ma wartość 0, adres zostanie zaakceptowany tylko wtedy, gdy jego n-ty bit zgadza się z adresem podsieci. Jeśli bit ten ma wartość 1, n-ty bit adresu nie wpływa na jego akceptowalność. W routerze R_5 akceptujemy zatem adresy z zakresu `192.168.X.0 - 192.168.X.255`, gdzie $X \in \{1, 2, 3, 4\}$.

```
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
```

Pozostałe routery inicjalizujemy podobnie, z pominięciem niektórych elementów obecnych przy R_5. Trochę inaczej wygląda ustawienie DNS, tutaj informujemy router, który z jego interfejsów będzie źródłem informacji o domenach. Konfigurację wszystkich pozostałych routerów zaprezentujemy na przykładzie R_2:

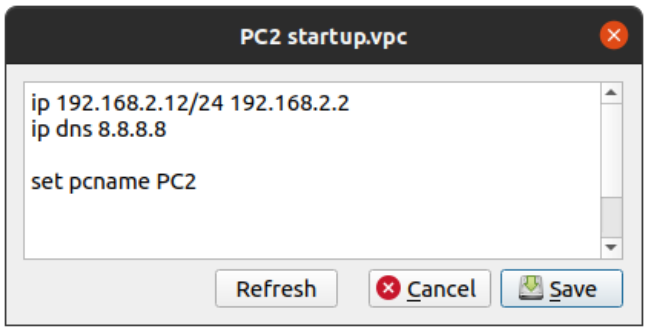
```
R2(config)# int f0/0
R2(config-if)# ip add 192.168.4.2 255.255.255.0
R2(config-if)# no shut
R2(config-if)# end

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
```

Urządzenia typu switch służą wyłącznie do unikania kolizji podczas przesyłania pakietów, zatem nie wymagają dodatkowej konfiguracji, pozostaje nam więc tylko przygotowanie komputerów. Ich inicjalizacja jest jednak o wiele mniej skomplikowana. Podajemy urządzeniu nadany mu adres wraz z maską i adresem routera, który łączy je z resztą sieci, jak również adres serwera DNS. Konfiguracja na przykładzie $\texttt{PC}_2$ została przedstawiona na rysunku 2.



Rysunek 2: Konfiguracja komputera na przykładzie $\texttt{PC}_2$

Na koniec sprawdzimy jeszcze, czy możemy pingować urządzenia znajdujące się w naszej sieci oraz zewnętrzne serwery – wykorzystamy w tym celu $\texttt{PC}_3$.

```
PC3> ping google.com
google.com resolved to 216.58.208.206
84 bytes from 216.58.208.206 icmp_seq=1 ttl=50 time=79.639 ms
84 bytes from 216.58.208.206 icmp_seq=2 ttl=50 time=75.320 ms
84 bytes from 216.58.208.206 icmp_seq=3 ttl=50 time=75.728 ms
84 bytes from 216.58.208.206 icmp_seq=4 ttl=50 time=85.654 ms
84 bytes from 216.58.208.206 icmp_seq=5 ttl=50 time=74.995 ms

PC3> ping 192.168.1.11
84 bytes from 192.168.1.11 icmp_seq=1 ttl=62 time=26.461 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=62 time=24.543 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=62 time=24.955 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=62 time=25.283 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=62 time=24.540 ms
```

3 Analiza pakietów

Zgodnie z poleceniem, ustawimy nasłuchiwanie na połączeniach $\texttt{R}_5 \leftrightarrow \texttt{Cloud}$, $\texttt{R}_5 \leftrightarrow \texttt{R}_6$ oraz $\texttt{R}_2 \leftrightarrow \texttt{Switch}_2$. Jeśli zrobimy to wystarczająco wcześnie, możemy zauważyć, jak routery wymieniają się między innymi pierwszymi informacjami dotyczącymi tablic routingu. Na przykład komputer $\texttt{PC}_1$ wysyła zapytanie RIP, a następnie otrzymuje kolejno:

```
IP Address: 192.168.2.0, Metric: 2
IP Address: 192.168.3.0, Metric: 1
IP Address: 192.168.4.0, Metric: 1
```

oraz

```
IP Address: 0.0.0.0, Metric: 2
IP Address: 192.168.122.0, Metric: 2
```

Przejdziemy teraz do właściwej części zadania: przeanalizujemy pakiety przemieszczające się przez nasłuchiwane połączenia w wyniku wydania polecenia `ping google.com` z komputera $\texttt{PC}_2$. Na wypadek, gdyby w sieci coś się zmieniło, urządzenia co jakiś czas retransmitują swoje informacje o routingu. Celem zwiększenia czytelności, te i inne sygnały niezwiązane z naszym eksperymentem odfiltrujemy w Wiresharku z użyciem polecenia `not (rip or loop or cdp or stp or ssdp)`. Przeanalizujemy podróżujące pakiety w kolejności chronologicznej.

Segment	Source	Destination	Prot.	Info
$\texttt{R}_2 \leftrightarrow \texttt{Sw}_2$	Private_66:68:01	Broadcast	ARP	Who has 192.168.2.2? Tell 192.168.2.12
$\texttt{R}_2 \leftrightarrow \texttt{Sw}_2$	ca:02:1c:5b:00:1c	Private_66:68:01	ARP	192.168.2.2 is at ca:02:1c:5b:00:1c

Na początku komputer PC_2 poszukuje urządzenia, które zostało mu przydzielone jako brama wyjścia z lokalnej sieci. Rozprowadza po całej podsieci zapytanie zgodne z protokołem ARP, który umożliwia zamianę adresów warstwy sieci (IP) na adresy fizycznych urządzeń z warstwy łącza danych. W końcu odpowiada mu R_2, transmitując do niego swój adres MAC.

Adres ten był potrzebny komputerowi do wysłania zapytania DNS w celu rozpoznania adresu `google.com`. Zobaczymy, jak to zapytanie jest propagowane przez naszą sieć.

Segment	Source	Destination	Protocol	Info
$R_2 \leftrightarrow Sw_2$	192.168.2.12	8.8.8.8	DNS	Standard query 0xe039 google.com
$R_5 \leftrightarrow R_6$	192.168.2.12	8.8.8.8	DNS	Standard query 0xe039 google.com
$R_5 \leftrightarrow$ Cloud	192.168.2.12	8.8.8.8	DNS	Standard query 0xe039 google.com
$R_5 \leftrightarrow$ Cloud	8.8.8.8	192.168.2.12	DNS	Response 0xe039 google.com 216.58.209.14
$R_5 \leftrightarrow R_6$	8.8.8.8	192.168.2.12	DNS	Response 0xe039 google.com 216.58.209.14
$R_2 \leftrightarrow Sw_2$	8.8.8.8	192.168.2.12	DNS	Response 0xe039 google.com 216.58.209.14

Przyjrzymy się teraz dokładniej, jak wyglądają te pakiety na każdym z odcinków. Pomimo, że adresy IP źródła i celu pozostają bez zmian na każdym z badanych odcinków sieci, pakiety różnią się nagłówkami nadanymi przez warstwę łącza danych, na przykład dla sygnału związanego z zapytaniem:

$R_2 \leftrightarrow Sw_2$:
Destination: ca:02:1c:5b:00:1c (router R_2)
Source: 00:50:79:66:68:01 (komputer PC_2)

$R_5 \leftrightarrow R_6$:
Destination: ca:01:2d:4d:00:1c (router R_5)
Source: ca:04:1c:7b:00:00 (router R_6)

$R_5 \leftrightarrow$ Cloud:
Destination: 52:54:00:2f:b2:0a (interfejs mojej karty sieciowej reprezentowany przez Cloud)
Source: ca:01:2d:4d:00:00 (router R_5)

Po ustaleniu adresu serwera, do którego chce dotrzeć, PC_2 rozpoczyna emisję pakietów z programu ping.

Segment	Source	Destination	Protocol	Info
$R_2 \leftrightarrow Sw_2$	192.168.2.12	216.58.209.14	ICMP	Echo (ping) request id=0xb5a9, seq=1/256, ttl=64 (reply in 326)
$R_5 \leftrightarrow R_6$	192.168.2.12	216.58.209.14	ICMP	Echo (ping) request id=0xb5a9, seq=1/256, ttl=62 (reply in 581)
$R_5 \leftrightarrow$ Cloud	192.168.2.12	216.58.209.14	ICMP	Echo (ping) request id=0xb5a9, seq=1/256, ttl=61 (reply in 1237)
$R_5 \leftrightarrow$ Cloud	216.58.209.14	192.168.2.12	ICMP	Echo (ping) reply id=0x0400, seq=1/256, ttl=54 (request in 1236)
$R_5 \leftrightarrow R_6$	216.58.209.14	192.168.2.12	ICMP	Echo (ping) reply id=0xb5a9, seq=1/256, ttl=53 (request in 580)
$R_2 \leftrightarrow Sw_2$	216.58.209.14	192.168.2.12	ICMP	Echo (ping) reply id=0xb5a9, seq=1/256, ttl=51 (request in 325)

Ramki warstwy łącza danych przepinane są podobnie, jak miało to miejsce przy pakietach DNS. Możemy zauważyć, że zgodnie z obserwacjami z listy 1, licznik `ttl` naszych pakietów dekrementowany jest przez każdy skok.

4 Wnioski

Konfiguracja nawet prostej sieci może okazać się dosyć skomplikowana. Sprawne posługiwanie się dokumentacją dostarczoną przez Cisco wymaga od nas przynajmniej podstawowej wiedzy na temat protokołów wykorzystywanych w różnych warstwach. Poprzez eksperymenty z narzędziami takimi jak GNS3 i Wireshark możemy jednak wyrobić sobie pewną intuicję dotyczącą zachodzących mechanizmów i łatwo korygować ewentualne błędy w naszych ustawieniach.