**Mateusz Lutecki**

# Konfiguracja i badanie routingu w sieci IPv6

***Fragmenty pracy dyplomowej inżynierskiej***

*Do użytku tylko jako materiał do ćwiczeń laboratoryjnych*
*na Wydziale Elektrycznym Politechniki Białostockiej*

Promotor pracy: dr inż. Andrzej Zankiewicz

Białystok, 2017

# 3. Protokoły routingu dynamicznego dla sieci IPv6.

Routing dynamiczny jest techniką, która zapewnia wymianę informacji pomiędzy routerami w danej topologii sieciowej tak, aby były w stanie wypełnić tablice routingu routerów co w efekcie doprowadzi do wyboru najlepszej trasy do sieci zdalnych. W przeciwieństwie do routingu statycznego, routing dynamiczny pozwala wybrać ścieżkę dostępną w czasie rzeczywistym przy zmianach układu sieci logicznej. W przypadku routingu dynamicznego, protokół trasowania działający na routerze jest odpowiedzialny za stworzenie, utrzymanie i aktualizację tabeli routingu dynamicznego, jest łatwy do skonfigurowania w dużych sieciach i jest mniej intuicyjny w wybraniu najlepszej trasy od routingu statycznego. W routingu statycznym, wszystkie te zadania są wykonywane ręcznie przez administratora systemu. Jednak z powodu aktualizacji tablic router, zużywa większą przepustowość łącza niż w przypadku routingu statycznego. Ostatecznie, routing dynamiczny jest mniej bezpieczny niż statyczny [4].

## 3.1 RIPnG - Routing Information Protocol

RIPng ma na celu umożliwienie routerom wymianę informacji o dostępnych trasach poprzez sieć opartą o protokół IPv6. Jest to wewnętrzny protokół routingu wektora odległości. Przeznaczony przede wszystkim do użytku jako IGP- używany w routingu wewnątrz systemu autonomicznego.

Protokół ten wykorzystuje stałe "metryki", aby porównać alternatywne trasy. Ta metoda nie jest odpowiednia dla sytuacji, w której trasy muszą być wybrane na podstawie parametrów zmieniających się w czasie rzeczywistym takich jak opóźnienie, niezawodność oraz obciążenie łącza.

Każdy router RIPng zakłada, że interfejs jest przypisany do jednej lub większej ilości sieci w przeciwnym razie urządzenie nie jest routerem. Są one określane jako bezpośrednio połączone do sieci. Protokół zakłada dostęp do informacji o każdej z tych sieci, najważniejszym jego parametrem jest metryka, czyli liczba skoków (next hop). Jeżeli liczba skoków zawiera się w przedziale od 1 do 15 to protokół RIPng odnajdzie trasy do tej sieci w przeciwnym wypadku protokół ten nie znajdzie danej trasy.

Ustawienia metryki do każdej z sieci powinien wdrażać administrator systemu. Każdy router w którym został zaimplementowany protokół posiada tablicę routingu. Tabela ta posiada jeden wpis dla każdego celu, który zostanie osiągnięty. Każdy wpis zawiera co najmniej następujące informacje:

- Prefiks IPv6 przeznaczenia.

- Metrykę, jaki jest całkowity koszt uzyskania pakietu od routera, który jest celem. Wartość ta jest sumą kosztów związanych z sieciami, które muszą wykonać ruch, aby uzyskać przeznaczenie.
- Adres IPv6 następnego routera na drodze do hosta docelowego.
- Flaga wskazująca, podająca informacje o trasie  czy się nie zmieniła.
- Licznik związany z trasą- co 30 sekund, proces RIPng wysyła wiadomości z odpowiedzią, zawierający kompletną tablicę routingu do każdego sąsiedniego routera [5].

## 3.2 OSPFv3 - Open Shortest Path First

OSPF jest wewnętrznym protokołem routingu stanu łącza. Jest on przeznaczony do uruchomienia wewnątrz jednego systemu autonomicznego. Każdy router OSPF zachowuje identyczną bazę danych opisujących topologię systemu autonomicznego. Każdy fragment tej bazy danych jest stanem lokalnym konkretnego routera (np. użyteczne interfejsy routera i osiągalni sąsiedzi). Z tej bazy danych, tabela routingu jest obliczana poprzez konstruowanie drzewa najkrótszej ścieżki. To drzewo daje drogę do każdego miejsca w systemie autonomicznym. Na drzewie pojawiają się liście reprezentujące sieci docelowe jako informacje pochodzące od zewnętrznego routingu.

OSPF przelicza trasy niezwłocznie w obliczu zmian topologicznych (takich jak awarie interfejsu routera), wykorzystując minimum ruchu protokołów routingu.

Ponadto wszystkie zmiany są uwierzytelnione. Oznacza to, że tylko zaufane routery mogą uczestniczyć w routingu dla danego systemu autonomicznego. Można stosować różne systemy uwierzytelniania: w rzeczywistości, oddzielne systemy uwierzytelniania mogą być skonfigurowane dla każdej podsieci. Trasy pakietów IP OSPF oparte są wyłącznie o docelowy adres IP znaleziony w nagłówku pakietu IP. Pakiety IP są kierowane "takie jakie są", nie są poddawane procesowi enkapsulacji w dalszych nagłówkach protokołu.

Gdy kilka tras posiada równy koszt do miejsca przeznaczenia, ruch zostaje rozłożony równo pomiędzy nimi. Koszt trasy opisany jest przez pojedynczą bezwymiarową metrykę.

OSPF zezwala na zaprojektowanie grupy sieci znajdujących się w jednym obszarze. Informacja o obszarze jest ukryta dla reszty systemu autonomicznego, zamaskowana informacja znacznie zmniejsza ruch routingu w sieci. Obszar stanowi agregację wielu podsieci IP [6].

Koszt metryki jest związany z rodzajem interfejsu routera. Koszt ścieżki można obliczyć ze wzoru:

$$koszt = \frac{10^8}{przepustowość\ [bps]}$$

Im niższa wartość kosztu, tym większe prawdopodobieństwo, że dany interfejs zostanie wybrany do przesyłania informacji do wybranej sieci. IOS automatycznie ustawia wartość kosztu bazując na przepustowości interfejsu. W tabeli 3.1 przedstawiono przykładowe wartości kosztu dla różnej przepustowości interfejsu routera.

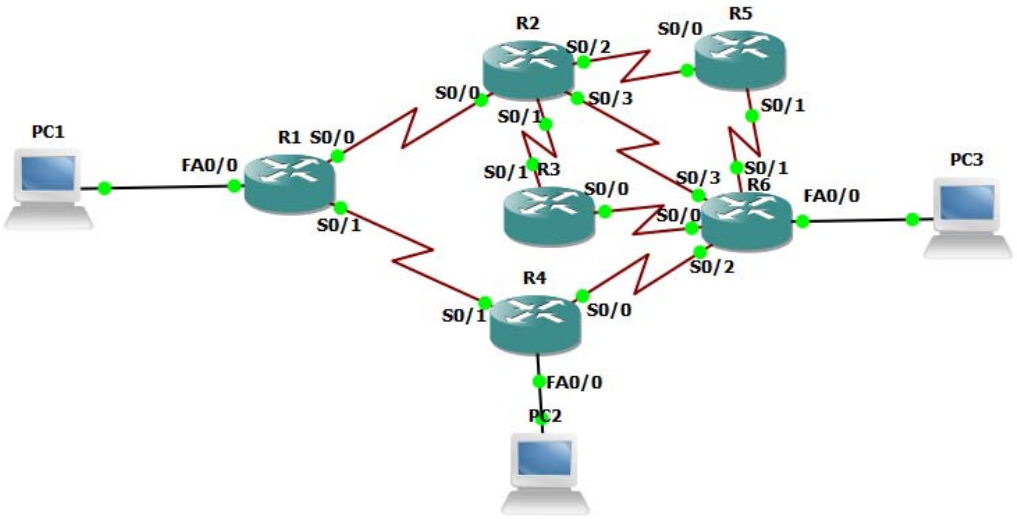
*Tabela 3.1. Koszt w zależności od przepustowości interfejsu routera.*

| Typ interfejsu i przepustowość | Koszt |
|---|---|
| Szeregowy, 56 kbps | 1785 |
| T1, 1.544 Mbps | 64 |
| 10 Mbps, Ethernet | 10 |
| 100 Mbps, Fast Ethernet | 1 |
| 1 Gbps, Gigabit Ethernet | 1 |

Dla interfejsów o przepustowości większej niż 100 Mbps koszt zawsze wynosi 1 [7].

## 4. Struktura laboratoryjna sieci do badania protokołów routingu.

Na rysunku 4.1 przedstawiono topologię sieci, czyli jakie urządzenia należy ze sobą przyłączyć za pomocą kabli szeregowych oraz hostów do routerów za pomocą kabli typu ethernet.




*Rys.4.1 Topologia badanej sieci.*

W tabeli 4.1 umieszczono tablicę adresacji poszczególnych adresów dla danego urządzenia i jego typu interfejsu:

*Tab. 4.1 Tablica adresacji.*

| Urządzenie | Interfejs | Adres IPv6/ Długość prefiksu | Brama Domyślna |
|---|---|---|---|
| R1 | FA0/0 | 2017:DB:ABCD:A::1 /64<br>FE80::1 link-local | - |
| | S0/0<br>DCE | 2017:DB:ABCD:12::1 /64<br>FE80::1 link-local | - |
| | S0/1<br>DTE | 2017:DB:ABCD:13::1 /64<br>FE80::1 link-local | - |
| R2 | S0/0<br>DTE | 2017:DB:ABCD:12::2/64 | - |
| | S0/1<br>DCE | 2017:DB:ABCD:14::2/64 | - |
| | S0/2<br>DCE | 2017:DB:ABCD:15::2/64 | - |
| | S0/3<br>DTE | 2017:DB:ABCD:16::2/64 | - |
| R3 | S0/0 | 2017:DB:ABCD:17::3/64 | - |

| | | | |
|---|---|---|---|
| | DCE | | |
| | S0/1<br>DTE | 2017:DB:ABCD:14::3/64 | - |
| R4 | FA0/0 | 2017:DB:ABCD:D::4/64<br>FE80::4 link-local | - |
| | S0/0<br>DTE | 2017:DB:ABCD:18::4/64<br>FE80::4 link-local | - |
| | S0/1<br>DCE | 2017:DB:ABCD:13::4/64<br>FE80::4 link-local | - |
| R5 | S0/0<br>DTE | 2017:DB:ABCD:15::5/64 | - |
| | S0/1<br>DCE | 2017:DB:ABCD:19::5/64 | - |
| R6 | FA0/0 | 2017:DB:ABCD:F::6/64<br>FE80::6 link-local | - |
| | S0/0<br>DTE | 2017:DB:ABCD:17::6/64<br>FE80::6 link-local | - |
| | S0/1<br>DTE | 2017:DB:ABCD:19::6/64<br>FE80::6 link-local | - |
| | S0/2<br>DCE | 2017:DB:ABCD:18::6/64<br>FE80::6 link-local | - |
| | S0/3<br>DCE | 2017:DB:ABCD:16::6/64<br>FE80::6 link-local | - |
| PC1 | Karta<br>sieciowa | 2017:DB:ABCD:A::A/64 | FE80::1 |
| PC2 | Karta<br>sieciowa | 2017:DB:ABCD:D::D/64 | FE80::4 |
| PC3 | Karta<br>sieciowa | 2017:DB:ABCD:F::F/64 | FE80::6 |

Do wykonania ćwiczenia została utworzona struktura sieciowa składająca się z 3 stacji roboczych:PC1, PC2, PC3 oraz 6 routerów połączonych według Rys. 4.1. za pomocą kabla szeregowego (serial). Każdy z routerów pracuje w innej sieci. Pary router R1 oraz PC1, router R4 oraz PC2 i router R6 i PC3 pracują w tych samych sieciach komputerowych. Można to stwierdzić na podstawie na podstawie wyznaczenia ich adresów sieci. Każdy interfejs szeregowy Routera R1, R4 oraz R6 posiada adres link-local, który jest adresem unicastowym

IPv6. Adresy te przypisane są tylko do konkretnego łącza fizycznego, wykorzystywane do porozumiewania się w jednej części sieci lokalnej do celu automatycznej konfiguracji adresu i protokołu wykrycia sąsiada. Adresy te mogą zostać użyte do komunikacji z węzłami przyłączonymi do tej samej sieci. Routery nie przekażą pakietu używając adresu link-local. Wszystkie interfejsy obsługujące protokół komunikacyjny IPv6 posiadają adres link-local unicast. W tabeli adresacji 4.1 zostało zawarte, które z połączeń serial są łączami DCE, a które DTE oraz który z portów danego routera będzie pracował w trybie master(tryb taktowania danych):łącze DCE, a który z portów drugiego routera w trybie slave(tryb odbierania danych):łącze DTE [10].

# 5. Zestawienie i konfiguracja opracowanych sieci

W tym rozdziale zaprezentowano w jaki sposób zaimplementować w routerze podstawową konfigurację, w jaki sposób przypisać adresy IPv6 dla poszczególnych interfejsów oraz jak sprawdzić poprawność skonfigurowania danego interfejsu. Omówiono również konfigurację danego typu routingu oraz przeprowadzenie diagnostyki w celu weryfikacji połączenia pomiędzy różnymi sieciami oraz transmisji danych.

## 5.1 RIPng

Pierwsza konfiguracja zostanie przedstawiona dla protokołu routingu dynamicznego RIPng. Topologia sieci oraz tablica adresacji została zawarta w rozdziale 4. Konfiguracja zostanie podzielona na dwie części. Wpierw zostanie wprowadzona konfiguracja podstawowych właściwości urządzeń, a następnie konfiguracja danego protokołu routingu oraz sprawdzenie poprawności wykonanych czynności.

Poniżej w skrypcie 5.1 przedstawiono konfigurację podstawową, która została na początku zaimplementowana we wszystkich routerach.

*Skrypt 5.1. Konfiguracja podstawowa wprowadzona dla wszystkich routerów.*

```
Router#delete nvram:startup-config
Router#reload
Router#conf t
Router(config)#hostname R1
R1(config)#no ip domain lookup
R1(config)#enable secret password ipv6
R1(config)#banner motd l Nieautoryzowany dostep jest zabroniony l
```

Na początku usunięto bieżącą konfigurację by przywrócić router do ustawień fabrycznych. Celem tego jest upewnienie się, że router uruchomiony będzie posiadał "czystą" konfiguracją IOS. Następnie przy pomocy komendy *reload* uruchamiamy ponownie urządzenie sieciowe. Następny krok to przejście do konfiguracji globalnej, przypisanie nazw np.R1 dla routera, a także przypisanie *ipv6* jako tajne hasło dla trybu uprzywilejowanego dostępu EXEC. Dzięki wprowadzeniu hasła nikt z zewnątrz nie będzie w stanie zalogować się do danego urządzenia, a także nikt z poza otoczenia nie odczyta hasła w pliku konfiguracyjnym, gdyż komenda *enable secret password* zapewnia szyfrowanie automatyczne, przy użyciu algorytmu skrótu MD5. W celu zapobiegnięcia niepożądanym zapytaniom DNS użyto komendy *no ip domain lookup.* Skonfigurowano MOTD (Message Of The Day) mający ostrzec inne osoby przed próbą nieautoryzowanego logowania [11].

W skrypcie 5.2 przedstawiono konfigurację poszczególnych interfejsów routera R1.

*Skrypt 5.2. Konfiguracja adresów IPv6 dla poszczególnych interfejsów routera R1.*

```
R1(config)#ipv6 unicast-routing
R1(config)#int f0/0
R1(config-if)#ipv6 address 2017:db:abcd:a::1/64
```

```
R1(config-if)#ipv6 address fe80::1 link-local
R1(config-if)#ipv6 nd prefix 2017:db:abcd:a::1/64 no-advertise
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#int s0/0
R1(config-if)#ipv6 address 2017:db:abcd:12::1/64
R1(config-if)#ipv6 address fe80::1 link-local
R1(config-if)#clock rate 128000
R1(config-if)#no shut
R1(config-if)#exit
R1(config)#int s0/1
R1(config-if)#ipv6 address 2017:db:abcd:13::1/64
R1(config-if)#ipv6 address fe80::1 link-local
R1(config-if)#no shut
R1(config-if)#end
```

Aby przejść do konfiguracji interfejsów należy wejść do konfiguracji globalnej, a następnie uruchomić przesyłanie datagramów unicast IPv6 za pomocą komendy *ipv6 unicast-routing*. Kolejno przechodzimy do konfiguracji interfejsu podając nazwę tego interfejsu i jego numer portu. W tym przypadku nasz router zawiera interfejs FastEthernet 0/0 pracujący z szybkością 100 Mb/s. Po uzyskaniu dostępu do właściwości interfejsu należy przypisać adres IPv6 zgodny z tabelą adresującą, czyli *ipv6 address 2017:db:abcd:a::1/64*,a także należy wyłączyć prefiksy, które są zawarte w tym adresie IPv6 w celu wyłączenia automatycznej konfiguracji adresu przyłączonego do tego interfejsu. Nie wyłączenie tej opcji spowoduje, że dany host będzie posiadał 2 adresy IPv6: przypisany w karcie sieciowej oraz przez protokół Neighbor Discovery. Do tego interfejsu należy również przypisać *adres link-local fe80::1*, który jest ważny tylko do komunikacji w obrębie segmentu sieci [10][12].

Na listingu 5.3 zaobserwowano komunikat od routera otrzymany po użyciu komendy *no shutdown* oraz po wyjściu z konfiguracji interfejsu.

*Skrypt 5.3. Komunikat otrzymany po konfiguracji interfejs.*

```
%LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
```

Komunikat ten informuje, że interfejs FastEthernet zmienił swój status na aktywny.

Następnie przechodzimy do konfiguracji dwóch pozostałych interfejsów szeregowych. Konfigurację realizujemy w ten sam sposób co dla interfejsu FA0/0. Różnicą w konfiguracji jest fakt, że tym interfejsom należy przypisać inny adres IPv6, zgodny z tablicą adresów. Jedyną zmianą jest to, że trzeba ustawić tzw. clock rate, czyli prędkość przysyłania danych. Przypisano dla interfejsu szeregowego 0/0 routera R1 prędkość przesyłania danych dla wartości 128000 bitów na sekundę .

Ze skryptu 5.4 odczytano komunikat o włączeniu obu interfejsów szeregowych.

*Skrypt 5.4. Zmiana stanu dwóch interfejsów szeregowych na stan: włączony.*

*Line protocol on Interface Serial0/0, changed state to up*

*Line protocol on Interface Serial0/1, changed state to up*

W celu sprawdzenia poprawnej konfiguracji interfejsów routera należy użyć komendy *show ipv6 interface brief:*

*Skrypt 5.5. Sprawdzenie poprawnej konfiguracji adresów IPv6 przyporządkowanych dla odpowiednich interfejsów.*

```
R1#show ipv6 interface brief
FastEthernet0/0          [up/up]
   FE80::1
   2017:DB:ABCD:A::1
Serial0/0                [up/up]
   FE80::1
   2017:DB:ABCD:12::1
FastEthernet0/1          [administratively down/down]
   FE80::1
Serial0/1                [up/up]
   FE80::1
   2017:DB:ABCD:13::1
```

Ze skryptu 5.5 wyczytano, że adresy IPv6 przypisane dla danych interfejsów zostały przypisane zgodnie z tablicą adresacji, a także, że interfejsy są włączone oraz podłączone do odpowiedniego media transmisyjnego .

Poniżej w skryptach od 5.6 do 5.10 przedstawiono konfigurację poszczególnych interfejsów zgodnie z tablicą adresacji.

*Skrypt 5.6. Konfiguracja adresów IPv6 dla poszczególnych interfejsów routera R2.*

```
R2(config)#ipv6 unicast-routing
R2(config)#int s0/0
R2(config-if)#ipv6 address 2017:db:abcd:12::2/64
R2(config-if)#ipv6 address fe80::2 link-local
R2(config-if)#no shut
R2(config-if)#exit
R2(config)#int s0/1
R2(config-if)#ipv6 address 2017:db:abcd:14::2/64
R2(config-if)#clock rate 128000
R2(config-if)#no shut
R2(config-if)#exit
R2(config)#int s0/2
R2(config-if)#ipv6 address 2017:db:abcd:15::2/64
R2(config-if)#clock rate 128000
R2(config-if)#no shut
R2(config-if)#exit
R2(config)#int s0/3
R2(config-if)#ipv6 address 2017:db:abcd:16::2/64
R2(config-if)#no shut
```

*Skrypt 5.7. Konfiguracja adresów IPv6 dla poszczególnych interfejsów routera R3.*

```
R3(config)#ipv6 unicast-routing
R3(config)#int s0/0
R3(config-if)#ipv6 address 2017:db:abcd:17::3/64
R3(config-if)#clock rate 128000
R3(config-if)#no shut
R3(config-if)#exit
R3(config)#int s0/1
R3(config-if)#ipv6 address 2017:db:abcd:14::3/64
R3(config-if)#no shut
```

*Skrypt 5.8. Konfiguracja adresów IPv6 dla poszczególnych interfejsów routera R4.*

```
R4(config)#ipv6 unicast-routing
R4(config)#int f0/0
R4(config-if)#ipv6 address 2017:db:abcd:d::4/64
R4(config-if)#ipv6 address fe80::4 link-local
R4(config-if)#ipv6 nd prefix 2017:db:abcd:d::4/64 no-advertise
R4(config-if)#no shut
R4(config-if)#exit
R4(config)#int s0/0
R4(config-if)#ipv6 address 2017:db:abcd:18::4/64
R4(config-if)#ipv6 address fe80::4 link-local
R4(config-if)#no shut
R4(config-if)#exit
R4(config)#int s0/1
R4(config-if)#ipv6 address 2017:db:abcd:13::4/64
R4(config-if)#ipv6 address fe80::4 link-local
R4(config-if)#clock rate 128000
R4(config-if)#no shut
R4(config-if)#exit
```

*Skrypt 5.9. Konfiguracja adresów IPv6 dla poszczególnych interfejsów routera R5.*

```
R5(config)#ipv6 unicast-routing
R5(config)#int s0/0
R5(config-if)#ipv6 address 2017:db:abcd:15::5/64
R5(config-if)#no shut
R5(config-if)#exit
R5(config)#int s0/1
R5(config-if)#ipv6 address 2017:db:abcd:19::5/64
R5(config-if)#clock rate 128000
R5(config-if)#no shut
R5(config-if)#exit
```

*Skrypt 5.10. Konfiguracja adresów IPv6 dla poszczególnych interfejsów routera R6.*

```
R6(config)#hostname R6
R6(config)#ipv6 unicast-routing
R6(config)#int f0/0
R6(config-if)#ipv6 address 2017:db:abcd:f::6/64
R6(config-if)#ipv6 address fe80::6 link-local
R6(config-if)#ipv6 nd prefix 2017:db:abcd:f::6/64 no-advertise
R6(config-if)#no shut
R6(config-if)#exit
R6(config)#int s0/0
R6(config-if)#ipv6 address 2017:db:abcd:17::6/64
R6(config-if)#ipv6 address fe80::6 link-local
R6(config-if)#no shut
R6(config-if)#exit
R6(config)#int s0/1
R6(config-if)#ipv6 address 2017:db:abcd:19::6/64
R6(config-if)#ipv6 address fe80::6 link-local
R6(config-if)#no shut
R6(config-if)#exit
R6(config)#int s0/2
R6(config-if)#ipv6 address 2017:db:abcd:18::6/64
R6(config-if)#ipv6 address fe80::6 link-local
R6(config-if)#clock rate 128000
R6(config-if)#no shut
R6(config-if)#exit
R6(config)#int s0/3
R6(config-if)#ipv6 address 2017:db:abcd:16::6/64
R6(config-if)#ipv6 address fe80::6 link-local
R6(config-if)#clock rate 128000
R6(config-if)#no shut
```

*Skrypt 5.11. Konfiguracja protokołu routingu RIPng dla routera R1.*

```
R1#conf t
R1(config)#int fa0/0
R1(config-if)#ipv6 rip rip1 enable
R1(config-if)#exit
R1(config)#int s0/0
```

```
R1(config-if)#ipv6 rip rip1 enable
R1(config-if)#exit
R1(config)#int s0/1
R1(config-if)#ipv6 rip rip1 enable
R1(config-if)#end
```

Przedostatnim krokiem jest konfiguracja routingu RIPnG, która została podana powyżej w skrypcie 5.11. Pierwszy krok to wejście w tryb konfiguracji globalnej, następnie określono typ interfejsu oraz numer portu w celu przejścia do konfiguracji danego interfejsu, a na końcu uruchomiono proces routingu IPv6 RIP na danym interfejsie za pomocą komendy *ipv6 rip rip1 enable*, gdzie rip1 jest to nazwa procesu [13].

Poniżej w skryptach od 5.12 do 5.16 przedstawiono konfiguracje protokołu RIPng dla pozostałych routerów.

*Skrypt nr 5.12. Konfiguracja protokołu routingu RIPng dla routera R2.*

```
R2#conf t
R2(config)#int s0/0
R2(config-if)#ipv6 rip rip2 enable
R2(config-if)#exit
R2(config)#int s0/1
R2(config-if)#ipv6 rip rip2 enable
R2(config-if)#exit
R2(config)#int s0/2
R2(config-if)#ipv6 rip rip2 enable
R2(config-if)#exit
R2(config)#int s0/3
R2(config-if)#ipv6 rip rip2 enable
```

*Skrypt nr 5.13. Konfiguracja protokołu routingu RIPng dla routera R3.*

```
R3#conf t
R3(config)#int s0/0
R3(config-if)#ipv6 rip rip3 enable
R3(config-if)#exit
R3(config)#int s0/1
R3(config-if)#ipv6 rip rip3 enable
R3(config-if)#end
```

*Skrypt nr 5.14. Konfiguracja protokołu routingu RIPng dla routera R4.*

```
R4#conf t
R4(config)#int fa0/0
R4(config-if)#ipv6 rip rip4 enable
R4(config-if)#exit
R4(config)#int s0/0
R4(config-if)#ipv6 rip rip4 enable
R4(config-if)#exit
R4(config)#int s0/1
R4(config-if)#ipv6 rip rip4 enable
```

*Skrypt nr 5.15. Konfiguracja protokołu routingu RIPng dla routera R5.*

```
R5#conf t
R5(config)#int s0/0
R5(config-if)#ipv6 rip rip5 enable
R5(config-if)#exit
R5(config)#int s0/1
R5(config-if)#ipv6 rip rip5 enable
R5(config-if)#end
```

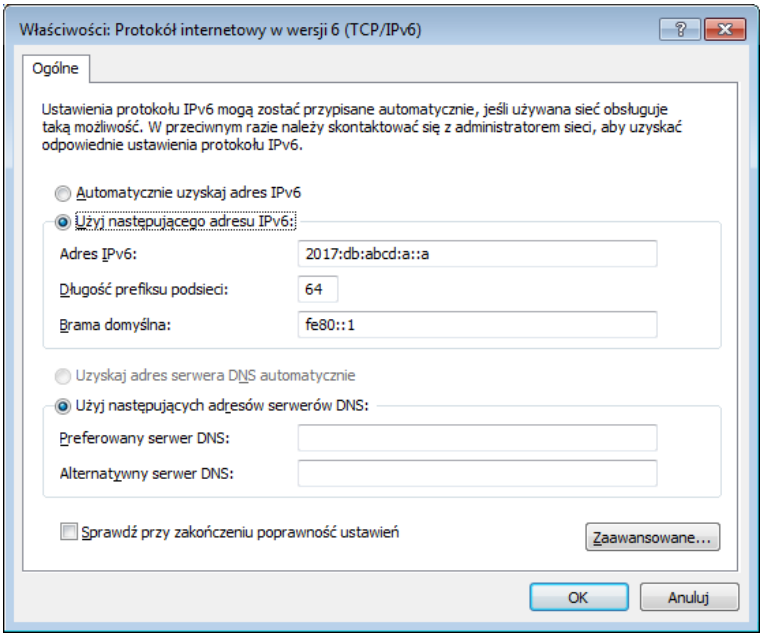
*Skrypt nr 5.16. Konfiguracja protokołu routingu RIPng dla routera R6.*

```
R6#conf t
R6(config)#int fa0/0
R6(config-if)#ipv6 rip rip6 enable
R6(config-if)#exit
```

```
R6(config)#int s0/0
R6(config-if)#ipv6 rip rip6 enable
R6(config-if)#exit
R6(config)#int s0/1
R6(config-if)#ipv6 rip rip6 enable
R6(config-if)#exit
R6(config)#int s0/2
R6(config-if)#ipv6 rip rip6 enable
R6(config-if)#exit
R6(config)#int s0/3
R6(config-if)#ipv6 rip rip6 enable
R6(config-if)#end
```

Różnicą pomiędzy konfiguracją protokołu dynamicznego RIPnG dla wszystkich routerów jest to, że każdy z nich posiada inną nazwę procesu.

Ostatnim krokiem jest konfiguracja adresów IPv6 dla 3 hostów. Na rysunku 5.1 przedstawiono w jaki sposób skonfigurować ustawienia karty sieciowej. Pierwszym krokiem jest wejście do panelu sterowania, wybranie opcji: Sieć i Internet, następnie Centrum sieci i udostępniania, a na koniec opcji Zmień ustawienia karty sieciowej. Następny krok to przejście do właściwości karty sieciowej za pomocą kliknięcia prawym przyciskiem myszy na ikonę karty sieciowej. Kolejnym krokiem było przejście do właściwości protokołu internetowego w wersji 6 za pomocą dwukrotnego kliknięcia lewym przyciskiem myszy. Na końcu zmieniono ustawienia adresacji IPv6 zgodnie z tablicą adresacji.




*Rys. 5.1. Konfiguracja adresu IPv6 dla hosta PC1.*

W celu sprawdzenia poprawności skonfigurowania karty sieciowej należy uruchomić wiersz poleceń i użyć komendy *ipconfig*.

*Skrypt 5.17. Sprawdzenie poprawności konfiguracji karty sieciowej dla hosta PC1.*

```
C:\Users\PC1>ipconfig

Konfiguracja IP systemu Windows

Karta Ethernet Połączenie lokalne:

   Sufiks DNS konkretnego połączenia :
   Adres IPv6. . . . . . . . . . . . : 2017:db:abcd:a::a
   Adres IPv6 połączenia lokalnego . : fe80::28:2dbc:39d0:81c2%11
   Brama domyślna. . . . . . . . . . : fe80::1%11
```

*Skrypt 5.18. Sprawdzenie poprawności konfiguracji karty sieciowej dla hosta PC2.*

```
C:\Users\PC2>ipconfig

Konfiguracja IP systemu Windows

Karta Ethernet Połączenie lokalne:

   Sufiks DNS konkretnego połączenia :
   Adres IPv6. . . . . . . . . . . . : 2017:db:abcd:d::d
   Adres IPv6 połączenia lokalnego . : fe80::f8c6:917a:29e7:88bc%11
   Brama domyślna. . . . . . . . . . : fe80::4%11
```

*Skrypt 5.19 Sprawdzenie poprawności konfiguracji karty sieciowej dla hosta PC3.*

```
C:\Users\PC3>ipconfig

Konfiguracja IP systemu Windows

Karta Ethernet Połączenie lokalne:

   Sufiks DNS konkretnego połączenia :
   Adres IPv6. . . . . . . . . . . . : 2017:db:abcd:f::f
   Adres IPv6 połączenia lokalnego . : fe80::84b3:62a9:6378:7fdb%11
   Brama domyślna. . . . . . . . . . : fe80::6%11
```

W skryptach 5.17, 5.18 i 5.19 przedstawiono sprawdzenie poprawności skonfigurowania Karty sieciowej dla PC1, PC2 oraz PC3 korzystających z protokołu komunikacyjnego IPv6. Podane adresy IPv6 oraz bramy domyślne mają zgodne adresy z tablicą adresacji 4.1. Dla bramy domyślnej został przydzielony domyślny prefiks o wartości 11.

W celu sprawdzenia czy proces routingu RIPnG działa oraz na jakim interfejsach jest uruchomiony należy użyć polecenia *show ipv6 protocols.*

*Skrypt 5.20. Przyłączone interfejsy do protokołu RIPng w wynikach polecenia* ***show ipv6 protocols****.*

```
R1#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "rip rip1"
  Interfaces:
    Serial0/1
    Serial0/0
    FastEthernet0/0
  Redistribution:
    None
```

Ze skryptu 5.20 odczytano, że procesem routingu uruchomionym na routerze R1 jest rip, którego nazwa procesu to "rip1". Na Interfejsach szeregowych 0/0, 0/1 i FastEthernet 0/0 skonfigurowano protokół routingu RIPnG .Świadczy to o poprawnym skonfigurowaniu procesu routingu.

W celu sprawdzenia poprawnego skonfigurowania protokołu routingu RIPnG należy zweryfikować czy każde urządzenie sieciowe znajdujące się w danej topologii posiada połączenie z innym urządzeniem np. PC1 z PC2, PC3 z R2 oraz PC1 z bramą domyślną. Wpierw wyłączono zaporę systemu Windows na każdym PC, gdyż może ona uniemożliwiać komunikację pomiędzy hostami.

W celu diagnozowania połączeń należy użyto komendy *ping*.

*Skrypt 5.21. Sprawdzenie połączenia pomiędzy hostem PC1, a hostem PC2.*

```
C:\Users\PC1>ping 2017:db:abcd:d::d

Badanie 2017:db:abcd:d::d z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:d::d: czas=277ms
Odpowiedź z 2017:db:abcd:d::d: czas=187ms
Odpowiedź z 2017:db:abcd:d::d: czas=203ms
Odpowiedź z 2017:db:abcd:d::d: czas=172ms

Statystyka badania ping dla 2017:db:abcd:d::d:
   Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
         (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
   Minimum = 172 ms, Maksimum = 277 ms, Czas średni = 209 ms
```

*Skrypt 5.22 Sprawdzenie połączenia pomiędzy hostem PC1, a hostem PC2.*

```
C:\Users\PC1>ping 2017:db:abcd:f::f

Badanie 2017:db:abcd:f::f z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:f::f: czas=138ms
Odpowiedź z 2017:db:abcd:f::f: czas=225ms
Odpowiedź z 2017:db:abcd:f::f: czas=421ms
Odpowiedź z 2017:db:abcd:f::f: czas=281ms

Statystyka badania ping dla 2017:db:abcd:f::f:
   Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
         (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
   Minimum = 138 ms, Maksimum = 421 ms, Czas średni = 266 ms
```

*Skrypt 5.23. Sprawdzenie połączenia pomiędzy hostem PC1, a jego bramą domyślną.*

```
C:\Users\PC1>ping fe80::1

Badanie fe80::1 z 32 bajtami danych:
Odpowiedź z fe80::1: czas=22ms
Odpowiedź z fe80::1: czas=624ms
Odpowiedź z fe80::1: czas=608ms
Odpowiedź z fe80::1: czas=156ms

Statystyka badania ping dla fe80::1:
   Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
         (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
   Minimum = 22 ms, Maksimum = 624 ms, Czas średni = 352 ms
```

*Skrypt 5.24. Sprawdzenie połączenia pomiędzy hostem PC1, a routerem R1.*

```
C:\Users\PC1>ping 2017:db:abcd:12::1

Badanie 2017:db:abcd:12::1 z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:12::1: czas=494ms
Odpowiedź z 2017:db:abcd:12::1: czas=320ms
Odpowiedź z 2017:db:abcd:12::1: czas=561ms
Odpowiedź z 2017:db:abcd:12::1: czas=280ms

Statystyka badania ping dla 2017:db:abcd:12::1:
```

```
    Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
          (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
    Minimum = 280 ms, Maksimum = 561 ms, Czas średni = 413 ms
```

*Skrypt 5.25. Sprawdzenie połączenia pomiędzy hostem PC1, a routerem R2.*

```
C:\Users\PC1>ping 2017:db:abcd:12::2

Badanie 2017:db:abcd:12::2 z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:12::2: czas=74ms
Odpowiedź z 2017:db:abcd:12::2: czas=87ms
Odpowiedź z 2017:db:abcd:12::2: czas=16ms
Odpowiedź z 2017:db:abcd:12::2: czas=62ms

Statystyka badania ping dla 2017:db:abcd:12::2:
    Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
          (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
    Minimum = 16 ms, Maksimum = 87 ms, Czas średni = 59 ms
```

*Skrypt 5.26. Sprawdzenie połączenia pomiędzy hostem PC1, a routerem R3.*

```
C:\Users\PC1>ping 2017:db:abcd:17::3

Badanie 2017:db:abcd:17::3 z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:17::3: czas=51ms
Odpowiedź z 2017:db:abcd:17::3: czas=112ms
Odpowiedź z 2017:db:abcd:17::3: czas=127ms
Odpowiedź z 2017:db:abcd:17::3: czas=144ms

Statystyka badania ping dla 2017:db:abcd:17::3:
    Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
          (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
    Minimum = 51 ms, Maksimum = 144 ms, Czas średni = 108 ms
```

*Skrypt 5.27. Sprawdzenie połączenia pomiędzy hostem PC1, a routerem R4.*

```
C:\Users\PC1>ping 2017:db:abcd:18::4

Badanie 2017:db:abcd:18::4 z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:18::4: czas=502ms
Odpowiedź z 2017:db:abcd:18::4: czas=515ms
Odpowiedź z 2017:db:abcd:18::4: czas=235ms
Odpowiedź z 2017:db:abcd:18::4: czas=455ms

Statystyka badania ping dla 2017:db:abcd:18::4:
    Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
          (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
    Minimum = 235 ms, Maksimum = 515 ms, Czas średni = 426 ms
```

*Skrypt 5.28. Sprawdzenie połączenia pomiędzy hostem PC1, a routerem R5.*

```
Badanie 2017:db:abcd:15::5 z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:15::5: czas=100ms
Odpowiedź z 2017:db:abcd:15::5: czas=125ms
Odpowiedź z 2017:db:abcd:15::5: czas=68ms
Odpowiedź z 2017:db:abcd:15::5: czas=101ms

Statystyka badania ping dla 2017:db:abcd:15::5:
    Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
          (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
    Minimum = 68 ms, Maksimum = 125 ms, Czas średni = 98 ms
```

*Skrypt 5.29. Sprawdzenie połączenia pomiędzy hostem PC1, a routerem R6.*

```
C:\Users\PC1>ping 2017:db:abcd:17::6

Badanie 2017:db:abcd:17::6 z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:17::6: czas=339ms
Odpowiedź z 2017:db:abcd:17::6: czas=78ms
```

```
Odpowiedź z 2017:db:abcd:17::6: czas=77ms
Odpowiedź z 2017:db:abcd:17::6: czas=15ms

Statystyka badania ping dla 2017:db:abcd:17::6:
   Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
         (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
   Minimum = 15 ms, Maksimum = 339 ms, Czas średni = 127 ms
```

*Skrypt 5.30. Sprawdzenie połączenia pomiędzy hostem PC2, a hostem PC3.*

```
C:\Users\PC2>ping 2017:db:abcd:f::f

Badanie 2017:db:abcd:f::f z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:f::f: czas=420ms
Odpowiedź z 2017:db:abcd:f::f: czas=453ms
Odpowiedź z 2017:db:abcd:f::f: czas=281ms
Odpowiedź z 2017:db:abcd:f::f: czas=453ms

Statystyka badania ping dla 2017:db:abcd:f::f:
   Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
         (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
   Minimum = 281 ms, Maksimum = 453 ms, Czas średni = 401 ms
```

*Skrypt 5.31. Sprawdzenie połączenia pomiędzy hostem PC2, a jego bramą domyślną.*

```
C:\Users\PC2>ping fe80::4

Badanie fe80::4 z 32 bajtami danych:
Odpowiedź z fe80::4: czas=39ms
Odpowiedź z fe80::4: czas=86ms
Odpowiedź z fe80::4: czas=47ms
Odpowiedź z fe80::4: czas=46ms

Statystyka badania ping dla fe80::4:
   Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
         (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
   Minimum = 39 ms, Maksimum = 86 ms, Czas średni = 54 ms
```

*Skrypt 5.32. Sprawdzenie połączenia pomiędzy hostem PC3, a jego bramą domyślną.*

```
C:\Users\PC3>ping fe80::6

Badanie fe80::6 z 32 bajtami danych:
Odpowiedź z fe80::6: czas=17ms
Odpowiedź z fe80::6: czas=116ms
Odpowiedź z fe80::6: czas=31ms
Odpowiedź z fe80::6: czas=47ms

Statystyka badania ping dla fe80::6:
   Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
         (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
   Minimum = 17 ms, Maksimum = 116 ms, Czas średni = 52 ms
```

Ze skryptów od 5.21 do 5.32 została przeprowadzona diagnostyka połączeń. Świadczy ona o poprawnym skonfigurowaniu sieci zgodnym z tablicą adresacji. Możliwość komunikacji pomiędzy hostami świadczy o poprawnym działaniu routingu dynamicznego (przesyłanie pakietów pomiędzy hostami znajdującymi się w innych sieciach). Duże opóźnienia podczas diagnostyki są spowodowane bardzo dużym obciążeniem sieci. Aby poprawnie skonfigurować sieć należy dokładnie sprawdzić czy poprawnie skonfigurowaliśmy dane interfejsy zgodnie z tablicą adresacji 4.1, a także adresy IPv6 hostów. Nieuruchomienie na którymś z portów FastEthernet protokołu routingu powodowało brak komunikacji pomiędzy hostami.

## 5.2 OSPFv3

Kolejnym protokołem w którym zostanie zaimplementowany proces routingu jest OSPF. Konfigurację podstawową i interfejsów urządzeń sieciowych należy wykonać zgodnie z skryptami od 5.1 do 5.2 oraz 5.6 do 5.10 z rozdziału o protokole routingu RIPnG oraz zgodnie z Rys.5.1 dla hostów. Sprawdzenie poprawności skonfigurowania routerów i hostów również zostały zweryfikowane zgodnie ze skryptami od 5.17 do 5.19 oraz 5.5. Ostatnim zadaniem jest skonfigurowanie routingu na wszystkich routerach. W skrypcie 5.33 zaprezentowano konfigurację routingu OSPF dla routera R1.

*Skrypt 5.33. Konfiguracja routingu OSPFv3 dla routera R1.*

```
R1#conf t
R1(config)#ipv6 router ospf 1
R1(config-rtr)#router-id 1.1.1.1
R1(config-rtr)#exit
R1(config)#int fa0/0
R1(config-if)#ipv6 ospf 1 area 0
R1(config-if)#exit
R1(config)#int s0/0
R1(config-if)#ipv6 ospf 1 area 0
R1(config-if)#exit
R1(config)#int s0/1
R1(config-if)#ipv6 ospf 1 area 0
R1(config-if)#exit
```

Pierwszym krokiem jest przejście do konfiguracji globalnej routera, wydanie polecenie *ipv6 router ospf 1,* aby móc rozpocząć proces OSPFv3 do routera, a następnie przypisanie identyfikator routera OSPFv3 *1.1.1.1* do R1. Następny krok to przejście do konfiguracji interfejsów danego routera, które mają uczestniczyć w procesie routingu OSPFv3 za pomocą komendy *ipv6 ospf 1 area 0,*gdzie "1" odpowiada za numer identyfikatora procesu, a "0" za numer obszaru OSPF. Aby proces routingu działał poprawnie w sieci należy uruchomić ten proces na wszystkich interfejsach [14].

W skryptach 5.34-5.38 przedstawiono poprawną konfigurację routingu dla R2,R3,R4,R5 i R6. Na każdym z routerów został przypisany inny identyfikator routera.

*Skrypt 5.34. Konfiguracja routingu OSPFv3 dla routera R2.*

```
R2(config)#ipv6 router ospf 1
R2(config-rtr)#router-id 2.2.2.2
R2(config-rtr)#exit
R2(config)#int s0/0
R2(config-if)#ipv6 ospf 1 area 0
R2(config-if)#exit
R2(config)#int s0/1
R2(config-if)#ipv6 ospf 1 area 0
R2(config-if)#exit
R2(config)#int s0/2
R2(config-if)#ipv6 ospf 1 area 0
R2(config-if)#exit
R2(config)#int s0/3
R2(config-if)#ipv6 ospf 1 area 0
R2(config-if)#exit
```

*Skrypt 5.35. Konfiguracja routingu OSPFv3 dla routera R3.*

```
R3(config)#ipv6 router ospf 1
R3(config-rtr)#router-id 3.3.3.3
R3(config-rtr)#exit
R3(config)#int s0/0
R3(config-if)#ipv6 ospf 1 area 0
R3(config-if)#exit
R3(config)#int s0/1
R3(config-if)#ipv6 ospf 1 area 0
R3(config-if)#exit
```

*Skrypt 5.36. Konfiguracja routingu OSPFv3 dla routera R4.*

```
R4(config)#ipv6 router ospf 1
R4(config-rtr)#router-id 4.4.4.4
R4(config-rtr)#exit
R4(config)#int fa0/0
R4(config-if)#ipv6 ospf 1 area 0
R4(config-if)#exit
R4(config)#int s0/0
R4(config-if)#ipv6 ospf 1 area 0
R4(config-if)#exit
R4(config)#int s0/1
R4(config-if)#ipv6 ospf 1 area 0
R4(config-if)#exit
```

*Skrypt 5.37. Konfiguracja routingu OSPFv3 dla routera R5.*

```
R5(config)#ipv6 router ospf 1
R5(config-rtr)#router-id 5.5.5.5
R5(config-rtr)#exit
R5(config)#int s0/0
R5(config-if)#ipv6 ospf 1 area 0
R5(config-if)#exit
R5(config)#int s0/1
R5(config-if)#ipv6 ospf 1 area 0
R5(config-if)#exit
```

*Skrypt 5.38. Konfiguracja routingu OSPFv3 dla routera R6.*

```
R6(config)#ipv6 router ospf 1
R6(config-rtr)#router-id 6.6.6.6
R6(config-rtr)#exit
R6(config)#int fa0/0
R6(config-if)#ipv6 ospf 1 area 0
R6(config-if)#exit
R6(config)#int s0/0
R6(config-if)#ipv6 ospf 1 area 0
R6(config-if)#exit
R6(config)#int s0/1
R6(config-if)#ipv6 ospf 1 area 0
R6(config-if)#exit
R6(config)#int s0/2
R6(config-if)#ipv6 ospf 1 area 0
R6(config-if)#exit
R6(config)#int s0/3
R6(config-if)#ipv6 ospf 1 area 0
R6(config-if)#exit
```

W Skrypcie 5.39 przedstawiono wiadomości wyświetlane po dodaniu interfejsów do strefy "0", w którym można odczytać numer procesu, identyfikator danego routera oraz interfejs dzięki któremu zostało wykryte "sąsiedztwo".

*Skrypt 5.39. Komunikaty wyświetlane przez router R1 po przypisaniu interfejsów do obszaru "0".*

```
*Mar  1 00:06:20.915: %OSPFv3-5-ADJCHG: Process 1, Nbr 4.4.4.4 on Serial0/1 from LOADING to FULL, Loading Done
*Mar  1 00:06:20.915: %OSPFv3-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial0/0 from LOADING to FULL, Loading Done
```

W celu sprawdzenia czy proces routingu OSPFv3 działa oraz na jakim interfejsach jest uruchomiony należy użyć polecenia *show ipv6 protocols*.

*Skrypt 5.40. Sprawdzenie działania procesu routingu na routerze R1.*

```
R1#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "ospf 1"
  Interfaces (Area 0):
    Serial0/1
    Serial0/0
    FastEthernet0/0
  Redistribution:
    None
```

Ze skryptu 5.40 dokonano następującej analizy: procesem routingu uruchomionym na R1 jest OSPF, którego numer procesu to "1". Interfejsy szeregowe 0/0, 0/1 i FastEthernet 0/0 są przypisane do obszaru "0". Świadczy to o poprawnym skonfigurowaniu procesu routingu.

Ostatnim krokiem jest sprawdzenie czy każde urządzenie sieciowe znajdujące się w topologii posiada połączenie z innym urządzeniem np. PC1 z PC2, PC3 z R2 oraz PC1 z bramą domyślną. W tym celu wykorzystano wiersz poleceń i użyto komendy *ping*.

*Skrypt 5.41. Sprawdzenie połączenia pomiędzy hostem PC2, a hostem PC1.*

```
C:\Users\PC2>ping 2017:db:abcd:a::a

Badanie 2017:db:abcd:a::a z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:a::a: czas=384ms
Odpowiedź z 2017:db:abcd:a::a: czas=94ms
Odpowiedź z 2017:db:abcd:a::a: czas=94ms
Odpowiedź z 2017:db:abcd:a::a: czas=101ms

Statystyka badania ping dla 2017:db:abcd:a::a:
    Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
            (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
    Minimum = 94 ms, Maksimum = 384 ms, Czas średni = 168 ms
```

*Skrypt 5.42. Sprawdzenie połączenia pomiędzy hostem PC2, a hostem PC3.*

```
C:\Users\PC2>ping 2017:db:abcd:f::f
Badanie 2017:db:abcd:f::f z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:f::f: czas=480ms
Odpowiedź z 2017:db:abcd:f::f: czas=255ms
Odpowiedź z 2017:db:abcd:f::f: czas=226ms
Odpowiedź z 2017:db:abcd:f::f: czas=90ms

Statystyka badania ping dla 2017:db:abcd:f::f:
    Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
            (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
    Minimum = 90 ms, Maksimum = 480 ms, Czas średni = 262 ms
```

*Skrypt 5.43. Sprawdzenie połączenia pomiędzy hostem PC3, a hostem PC1.*

```
C:\Users\PC3>ping 2017:db:abcd:a::a

Badanie 2017:db:abcd:a::a z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:a::a: czas=294ms
Odpowiedź z 2017:db:abcd:a::a: czas=144ms
Odpowiedź z 2017:db:abcd:a::a: czas=301ms
Odpowiedź z 2017:db:abcd:a::a: czas=211ms

Statystyka badania ping dla 2017:db:abcd:a::a:
```

```
    Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
          (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
    Minimum = 144 ms, Maksimum = 301 ms, Czas średni = 237 ms
```

*Skrypt 5.44. Sprawdzenie połączenia pomiędzy hostem PC2, a routerem R1.*

```
C:\Users\PC2>ping 2017:db:abcd:13::1

Badanie 2017:db:abcd:13::1 z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:13::1: czas=179ms
Odpowiedź z 2017:db:abcd:13::1: czas=109ms
Odpowiedź z 2017:db:abcd:13::1: czas=31ms
Odpowiedź z 2017:db:abcd:13::1: czas=78ms

Statystyka badania ping dla 2017:db:abcd:13::1:
    Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
          (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
    Minimum = 31 ms, Maksimum = 179 ms, Czas średni = 99 ms
```

*Skrypt 5.45. Sprawdzenie połączenia pomiędzy hostem PC2, a routerem R2.*

```
C:\Users\PC2>ping 2017:db:abcd:14::2

Badanie 2017:db:abcd:14::2 z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:14::2: czas=76ms
Odpowiedź z 2017:db:abcd:14::2: czas=125ms
Odpowiedź z 2017:db:abcd:14::2: czas=124ms
Odpowiedź z 2017:db:abcd:14::2: czas=109ms

Statystyka badania ping dla 2017:db:abcd:14::2:
    Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
          (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
    Minimum = 76 ms, Maksimum = 125 ms, Czas średni = 108 ms
```

*Skrypt 5.46. Sprawdzenie połączenia pomiędzy hostem PC2, a routerem R3.*

```
C:\Users\PC2>ping 2017:db:abcd:14::3

Badanie 2017:db:abcd:14::3 z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:14::3: czas=124ms
Odpowiedź z 2017:db:abcd:14::3: czas=47ms
Odpowiedź z 2017:db:abcd:14::3: czas=172ms
Odpowiedź z 2017:db:abcd:14::3: czas=45ms

Statystyka badania ping dla 2017:db:abcd:14::3:
    Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
          (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
    Minimum = 45 ms, Maksimum = 172 ms, Czas średni = 97 ms
```

*Skrypt 5.47. Sprawdzenie połączenia pomiędzy hostem PC2, a routerem R4.*

```
C:\Users\PC2>ping 2017:db:abcd:13::4

Badanie 2017:db:abcd:13::4 z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:13::4: czas=53ms
Odpowiedź z 2017:db:abcd:13::4: czas=57ms
Odpowiedź z 2017:db:abcd:13::4: czas=63ms
Odpowiedź z 2017:db:abcd:13::4: czas=15ms

Statystyka badania ping dla 2017:db:abcd:13::4:
    Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
          (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
    Minimum = 15 ms, Maksimum = 63 ms, Czas średni = 47 ms
```

*Skrypt 5.48. Sprawdzenie połączenia pomiędzy hostem PC2, a routerem R5.*

```
Badanie 2017:db:abcd:19::5 z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:19::5: czas=154ms
Odpowiedź z 2017:db:abcd:19::5: czas=115ms
```

```
Odpowiedź z 2017:db:abcd:19::5: czas=243ms
Odpowiedź z 2017:db:abcd:19::5: czas=125ms

Statystyka badania ping dla 2017:db:abcd:19::5:
   Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
         (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
   Minimum = 115 ms, Maksimum = 243 ms, Czas średni = 159 ms
```

*Skrypt 5.49. Sprawdzenie połączenia pomiędzy hostem PC2, a routerem R6.*

```
C:\Users\PC2>ping 2017:db:abcd:19::6

Badanie 2017:db:abcd:19::6 z 32 bajtami danych:
Odpowiedź z 2017:db:abcd:19::6: czas=387ms
Odpowiedź z 2017:db:abcd:19::6: czas=46ms
Odpowiedź z 2017:db:abcd:19::6: czas=102ms
Odpowiedź z 2017:db:abcd:19::6: czas=165ms

Statystyka badania ping dla 2017:db:abcd:19::6:
   Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
         (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
   Minimum = 46 ms, Maksimum = 387 ms, Czas średni = 175 ms
```

Ze skryptów 5.41 do 5.49 ustalono, że przeprowadzona diagnostyka świadczy o poprawnym skonfigurowaniu sieci zgodnym z tablicą adresacji oraz poprawnym skonfigurowaniu procesu routingu. Wyniki diagnostyki są takie same jak w przypadku badanego routingu RIPnG. Zarówno przy konfiguracji RIPng jak i OSPFv3 nie była wymagana znajomość przyłączonych interfejsów do sieci (administrator sieci w takim przypadku nie musi znać wcześniejsze topologii do której przyłączone są pozostałe routery).

# 6. Praktyczna analiza pracy wybranych protokołów routingu IPv6

W tym rozdziale zostało przedstawione badanie tras pomiędzy hostami PC1, PC2 oraz PC3 dla trzech różnych protokołów routingu zaimplementowanych dla sieci. Trasy oraz tablice routingu zostały zbadane przy różnych ustawieniach konfiguracji sieci (zmiana metryki dla danego interfejsu oraz przy wyłączonym interfejsie).

## 6.1 Badanie RIPng

Dla sieci ze skonfigurowanym protokołem routingu RIPnG zostaną przeprowadzone badania trasy pakietów w sieci IP. Pierwsze badanie odbyło się bez żadnych zmian w konfiguracji całej sieci.

Aby przeprowadzić badanie należy użyć komendy *tracert* w wierszu poleceń z hosta, z którego zostaną wysłane pakiety, a następnie podać adres IP sieci docelowej.

*Skrypt 6.1. Badanie trasy pakietów w sieci IP z hosta PC1 do hosta PC2.*

```
C:\Users\PC1>tracert 2017:db:abcd:d::d

Śledzenie trasy do 2017:db:abcd:d::d z maksymalną liczbą 30 przeskoków.

  1    44 ms    29 ms    30 ms  2017:db:abcd:a::1
  2   240 ms    79 ms    91 ms  2017:db:abcd:13::4
  3   121 ms    93 ms   182 ms  2017:db:abcd:d::d

Śledzenie zakończone.
```

Z powyższego skryptu odczytano ,że trasa przebiega przez Router R1, następnie R4, ostatecznie pakiet zostaje przesłany do interfejsu ethernet hosta PC2. Dzieje się tak, ponieważ wartość metryki z routera R1 do sieci 2017:db:abcd:d::/64 poprzez interfejs szeregowy 0/1 wynosi 2, a poprzez interfejs szeregowy wynosi 4. Trasa alternatywna została wpisana do tablicy routingu taka, która posiada mniejszą wartość metryki.

*Skrypt 6.2. Badanie trasy pakietów w sieci IP z hosta PC1 do hosta PC3*

```
C:\Users\PC1>tracert 2017:db:abcd:f::f

Śledzenie trasy do 2017:db:abcd:f::f z maksymalną liczbą 30 przeskoków.

  1    44 ms    62 ms    46 ms  2017:db:abcd:a::1
  2    47 ms    30 ms    45 ms  2017:db:abcd:12::2
  3    68 ms   128 ms    69 ms  2017:db:abcd:18::6
  4   656 ms   235 ms   276 ms  2017:db:abcd:f::f

Śledzenie zakończone.
```

Z powyższego skryptu odczytano ,że trasa przebiega przez Router R1, następnie R2 i R6, w rezultacie pakiet zostaje przesłany do interfejsu ethernet hosta PC3. Jednakże, po ponownym sprawdzeniu przebiegu trasy, pakiety zostały przesłane przez router R4 co świadczy o zmianie drogi pakietu. Dzieje się tak ponieważ, pomiędzy hostami PC1 oraz PC3 istnieją dwie alternatywne trasy (przez R2 oraz R4). Pakiety, które będą miały trafić do sieci

2017:db:abcd:f::/64 będą wysyłane do routera R2 jak i R4, gdyż posiadają one taką samą ilość przeskoków. Następnie sprawdzono tablicę routingu w celu sprawdzenia ilości tras alternatywnych do sieci 2017:db:abcd:f::/64. Na routerze R1 użyć polecenia *show ipv6 route*

*Skrypt 6.3. Sprawdzenie tablicy routingu na routerze R1 w celu weryfikacji trasy do sieci 2017:DB:ABCD:F::/64 oraz 2017:DB:ABCD:D::/64.*

```
R1#sh ipv6 route
C   2017:DB:ABCD:A::/64 [0/0]
     via ::, FastEthernet0/0
L   2017:DB:ABCD:A::1/128 [0/0]
     via ::, FastEthernet0/0
R   2017:DB:ABCD:D::/64 [120/2]
     via FE80::4, Serial0/1
R   2017:DB:ABCD:F::/64 [120/3]
     via FE80::4, Serial0/1
     via FE80::2, Serial0/0
```

Z powyższej tablicy routingu odczytano, że w tablicy są zawarte dwie alternatywne trasy przez interfejsy szeregowe 0/0 routera R2 oraz 0/1 routera R4. W rzeczywistości pakiet zostaje przesłany raz przez interfejs szeregowy 0/0, a kolejny raz przez interfejs 0/1 (pakiety będą przesyłane naprzemiennie). Wynika to z takiej samej wartości metryki przez oba interfejsy (druga wartość w nawiasie kwadratowym), czyli ilości przeskoków do docelowej sieci. Z tablicy routingu możemy również odczytano odległość administracyjną (pierwsza wartość w nawiasie kwadratowym). Dla sieci przyłączonych bezpośrednio do routera wartość wynosi 0. Natomiast dla procesu routingu RIPng wartość odległości administracyjnej wynosi 120.

*Skrypt 6.4. Badanie trasy pakietów w sieci IP z hosta PC2 do hosta PC3.*

```
C:\Users\PC2>tracert 2017:db:abcd:f::f

Śledzenie trasy do 2017:db:abcd:f::f z maksymalną liczbą 30 przeskoków.

  1    59 ms    80 ms    37 ms  2017:db:abcd:d::4
  2    58 ms    80 ms   100 ms  2017:db:abcd:18::6
  3   246 ms   487 ms    98 ms  2017:db:abcd:f::f

Śledzenie zakończone.
```

Z powyższego skryptu odczytano ,że trasa przebiega przez router R4, następnie R6,a ostatecznie pakiet zostaje przesłany do interfejsu ethernet hosta PC3- jest to najkrótsza trasa tego pakietu.

Następnie obserwowano komunikaty protokoły routingu przy pomocy komendy debug oraz przy pomocy analizatora Wireshark. Na początku użyto komendy *debug ipv6 rip* na routerze R4.

*Skrypt 6.5. Obserwacja komunikatów protokołów routingu na routerze R4.*

```
*Mar  1 00:13:28.243: RIPng: Sending multicast update on Serial0/0 for rip4
```

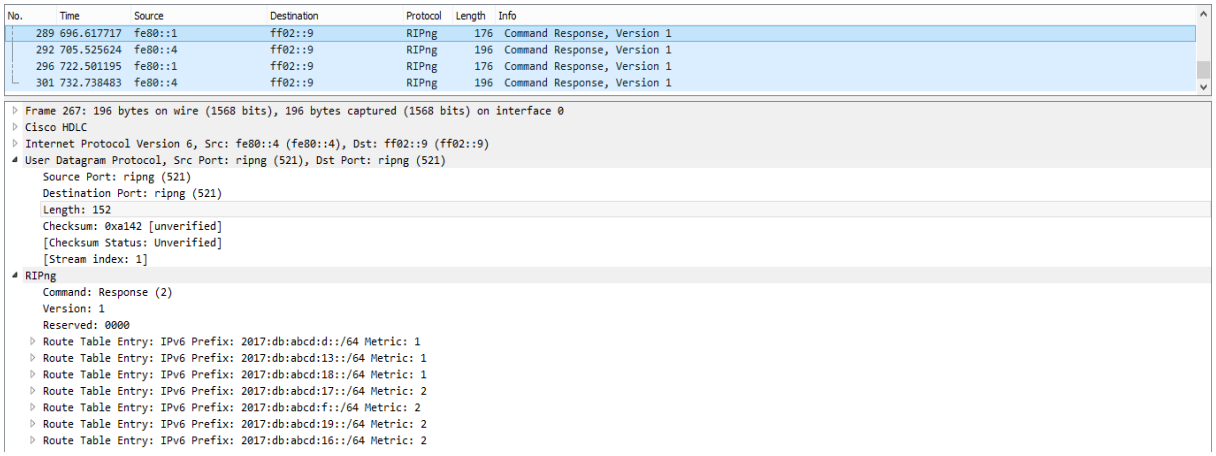
```
*Mar  1 00:13:28.243:       src=FE80::4
*Mar  1 00:13:28.243:       dst=FF02::9 (Serial0/0)
*Mar  1 00:13:28.243:       sport=521, dport=521, length=112
*Mar  1 00:13:28.243:       command=2, version=1, mbz=0, #rte=5
*Mar  1 00:13:28.243:       tag=0, metric=1, prefix=2017:DB:ABCD:D::/64
*Mar  1 00:13:28.243:       tag=0, metric=1, prefix=2017:DB:ABCD:18::/64
*Mar  1 00:13:28.243:       tag=0, metric=1, prefix=2017:DB:ABCD:13::/64
*Mar  1 00:13:28.243:       tag=0, metric=2, prefix=2017:DB:ABCD:A::/64
*Mar  1 00:13:28.243:       tag=0, metric=2, prefix=2017:DB:ABCD:12::/64 (...)
*Mar  1 00:13:30.915: RIPng: response received from FE80::6 on Serial0/0 for rip4
*Mar  1 00:13:30.915:       src=FE80::6 (Serial0/0)
*Mar  1 00:13:30.915:       dst=FF02::9
*Mar  1 00:13:30.919:       sport=521, dport=521, length=172
*Mar  1 00:13:30.919:       command=2, version=1, mbz=0, #rte=8
*Mar  1 00:13:30.919:       tag=0, metric=1, prefix=2017:DB:ABCD:F::/64
*Mar  1 00:13:30.919:       tag=0, metric=1, prefix=2017:DB:ABCD:18::/64
*Mar  1 00:13:30.919:       tag=0, metric=1, prefix=2017:DB:ABCD:19::/64
*Mar  1 00:13:30.923:       tag=0, metric=1, prefix=2017:DB:ABCD:17::/64
*Mar  1 00:13:30.923:       tag=0, metric=1, prefix=2017:DB:ABCD:16::/64
*Mar  1 00:13:30.923:       tag=0, metric=2, prefix=2017:DB:ABCD:12::/64
*Mar  1 00:13:30.923:       tag=0, metric=2, prefix=2017:DB:ABCD:15::/64
*Mar  1 00:13:30.927:       tag=0, metric=2, prefix=2017:DB:ABCD:14::/64R4#
```

W skrypcie 6.5 przedstawiono wiadomości wyświetlane podczas obserwacji komunikatów protokołu routingu RIPng. Z komunikatów odczytano: aktualizacje rip wysyłane są na wszystkie interfejsy przyłączone do routera to jest interfejsy szeregowe 0/0, 0/1, a także interfejs FastEthernet 0/0. Aktualizacje przenoszone są na porcie 521. Adresem docelowym jest adres FF02::9, który jest adresem multicast. Router wysyła i odbiera informacje od routerów, które są do niego przyłączone. Informacjami wysyłanymi i odbieranymi przez router są: adres sieci, maska sieci, interfejsy oraz metryka.

Do interfejsu FastEthernet 0/0 jest podłączony tylko host, więc wysyłanie aktualizacji RIPng poprzez ten interfejs jest zbędne, ponieważ aktualizacje te może przetworzyć tylko router.

Nie powinny być wysyłane aktualizacje poprzez interfejs, do którego nie jest podłączony router, gdyż może to zaburzyć ciągłość transmisji.

Następnie obserwowano pakiety przy pomocy analizatora Wireshark. Na rysunku 6.1 przedstawiono przesyłane komunikaty przez router R1 oraz R4.



*Rys.6.1. Przesyłane komunikaty przez router R1 oraz R4.*

Routery R1 oraz R4 wysyłają swoje wpisy tras sieci, które są osiągalne (mogą wysyłać informacje także o sieciach nieosiągalnych, wtedy wartość metryki dla sieci jest równa 16). We wpisach są zawarte informacje o adresie sieci, masce sieci oraz metryce. Aktualizacje przesyłane są na adres multicast ff02::9. W celu zapewnienia komunikacji używany jest protokół UDP (User Datagram Protocol), który wykorzystywany jest do przesyłania danych. Jest to protokół bezpołączeniowy, który w porównaniu do protokołu TCP umożliwia szybszą transmisję danych, ale nie gwarantuje niezawodności przesyłania danych.

Następnie w celu zmiany tras przebiegu pakietu użyto polecenia *shutdown* we właściwościach interfejsu szeregowego 0/0 routera R4, a później sprawdzono przebieg trasy pakietów z PC1 do PC3 i PC1 do PC2.

*Skrypt 6.6. Badanie trasy pakietów w sieci IP z hosta PC2 do hosta PC3.*

```
C:\Users\PC1>tracert 2017:db:abcd:f::f

Śledzenie trasy do 2017:db:abcd:f::f z maksymalną liczbą 30 przeskoków.

  1    19 ms    32 ms    43 ms  2017:db:abcd:a::1
  2    57 ms    66 ms    33 ms  2017:db:abcd:12::2
  3    61 ms   103 ms    16 ms  2017:db:abcd:16::6
  4   127 ms   131 ms    70 ms  2017:db:abcd:f::f

Śledzenie zakończone.
```

*Skrypt 6.7. Badanie trasy pakietów w sieci IP z hosta PC2 do hosta PC3.*

```
C:\Users\PC2>tracert 2017:db:abcd:f::f

Śledzenie trasy do 2017:db:abcd:f::f z maksymalną liczbą 30 przeskoków.

  1    52 ms    68 ms    32 ms  2017:db:abcd:d::4
  2    16 ms    43 ms    10 ms  2017:db:abcd:13::1
  3    59 ms    88 ms    34 ms  2017:db:abcd:12::2
  4    92 ms    43 ms    43 ms  2017:db:abcd:16::6
  5   219 ms   132 ms    43 ms  2017:db:abcd:f::f

Śledzenie zakończone.
```

Wyłączenie interfejsu lub uszkodzenie nie spowodowało przerwania odpowiedniego działania sieci. Zaobserwowano zmianę przebiegu trasy między PC2 a PC3. Wcześniejsza trasa przebiegała przez interfejs podłączony do R6, jednakże jego wyłączenie spowodowało, że trasa przebiega teraz przez interfejsy wyjściowe R1,R2, R6, a następnie do hosta PC3. Trasa pomiędzy PC1 oraz PC3 nie uległa zmianie, ale spowodowało to usunięcie jednej z alternatywnych tras w tablicy routingu.

Następnie sprawdzono czy przy awarii interfejsu połączonego routera R2 z R6, zostanie przerwana komunikacja, jeżeli nie, sprawdzono jaki przebieg trasy będą posiadały pakiety. W tym celu wyłączono interfejs szeregowy 0/3 na routerze R6 za pomocą komendy *shutdown*.

*Skrypt 6.8. Badanie trasy pakietów w sieci IP z hosta PC2 do hosta PC3.*

```
C:\Users\PC1>tracert 2017:db:abcd:f::f

Śledzenie trasy do 2017:db:abcd:f::f z maksymalną liczbą 30 przeskoków.

  1    55 ms    34 ms    34 ms  2017:db:abcd:a::1
  2    38 ms    71 ms    43 ms  2017:db:abcd:12::2
  3    42 ms    65 ms    70 ms  2017:db:abcd:14::3
  4    80 ms    65 ms    66 ms  2017:db:abcd:19::6
  5   143 ms   105 ms    69 ms  2017:db:abcd:f::f

Śledzenie zakończone.
```

*Skrypt 6.9. Badanie trasy pakietów w sieci IP z hosta PC2 do hosta PC3.*

```
C:\Users\PC2>tracert 2017:db:abcd:f::f

Śledzenie trasy do 2017:db:abcd:f::f z maksymalną liczbą 30 przeskoków.

  1    42 ms    32 ms    32 ms  2017:db:abcd:d::4
  2    82 ms    33 ms    33 ms  2017:db:abcd:13::1
  3    19 ms    21 ms    32 ms  2017:db:abcd:12::2
  4    43 ms    99 ms    67 ms  2017:db:abcd:14::3
  5    85 ms    65 ms   112 ms  2017:db:abcd:19::6
  6   314 ms   307 ms   138 ms  2017:db:abcd:f::f

Śledzenie zakończone.
```

Ze skryptów 6.8 oraz 6.9 trasa pakietów przebiega przez Router R3 i R6, jednakże pakiet posiada dwie trasy: R3 do R6 oraz R5 do R6, oba te routery również posiadają tą samą metrykę.

Trasa przebiega przez R3 oraz R5 (dwie alternatywne trasy). Na proces trasowania nie ma wpływu jaki numer procesu routingu został uruchomiony na danym interfejsie. Następnie sprawdzono tablicę routingu routera R2 w celu weryfikacji.

*Skrypt 6.10. Badanie trasy pakietów w sieci IP z hosta PC2 do hosta PC3.*

```
R2#sh ipv6 route
R   2017:DB:ABCD:F::/64 [120/3]
     via FE80::5, Serial0/2
     via FE80::3, Serial0/1
```

Z powyższego skryptu odczytano, że w tablicy są zawarte dwie alternatywne trasy przez interfejsy szeregowe 0/0 oraz 0/1 do sieci *2017:DB:ABCD:F::/64*. Metryka dla sieci *2017:db:abcd:a::/64* wynosi 4, a dla sieci *2017:db:abcd:d::/64* wynosi 5. Zmiana trasy spowodowała wzrost kosztu metryk do sieci *2017:DB:ABCD:F::/64*, jednakże koszt ten nie przekroczył wartości 15. Gdyby wartość ta została przekroczona nie możliwa była by komunikacja.

Aby wprowadzić zmianę przebiegu trasy należy zwiększyć metrykę interfejsu szeregowego 0/1 routera R3 o 1, a później sprawdzić czy trasa pakietu uległa zmianie.

Najpierw sprawdzono jaka jest ustawiona domyślnie metryka dla danego procesu routingu. W tym celu użyto polecenia *show ipv6 rip database*.

*Skrypt 6.11. Badanie trasy pakietów w sieci IP z hosta PC2 do hosta PC3.*

```
R2#show ipv6 rip database

 2017:DB:ABCD:F::/64, metric 3, installed
```

```
    Serial0/1/FE80::3, expires in 178 secs
    Serial0/2/FE80::5, expires in 165 secs
```

Z powyższego skryptu odczytano: domyślna metryka dla interfejsu szeregowego 0/1 jest równa 3 oraz wpisy w bazie danych są zapisane na 3 minuty. Przez ten czas ta tablica będzie aktualna i jego zawartość nie zmieni się po upływie tego czasu. Jedynie użycie odpowiedniej komendy usunie aktualną bazę danych. Następnie zmieniono wartość metryki, tak aby trasa przebiegała przez interfejs routera R5. Pierwszy krokiem w celu zmiany wartości metryki jest wejście do konfiguracji globalnej routera R2, wpisanie komendy *ipv6 rip rip2 metric-offset 4*, aby zwiększyć metrykę o wartość 4. W celu sprawdzenia poprawnej zmiany metryki użyto komendy *show ipv6 rip database*.

*Skrypt 6.12. Badanie trasy pakietów w sieci IP z hosta PC2 do hosta PC3.*

```
R2#show ipv6 rip database

2017:DB:ABCD:F::/64, metric 3, installed
    Serial0/2/FE80::5, expires in 172 secs
```

Z powyższej bazy danych zaobserwowano, że przez interfejs szeregowy 0/1 nie istnieje trasa do sieci *2017:DB:ABCD:F::/64*, gdyż wartości metryki jest większa od wartości metryki interfejsu szeregowego 0/2 (po zwiększeniu wartości metryki łącza szeregowego 0/1 jego wartość metryki wynosi 7). Po zmianie metryki sprawdzono tablicę routingu. Jeżeli tablica posiadałaby wpis z interfejsem szeregowym 0/1 jako alternatywną trasą należałoby użyć polecenia *clear ipv6 route* * i *clear ipv6 rip* na danym routerze. Następnie sprawdzono trasę przebiegu pakietów z PC1 do PC3 oraz PC2 oraz PC 3

*Skrypt 6.13. Badanie trasy pakietów w sieci IP z hosta PC2 do hosta PC3.*

```
C:\Users\PC1>tracert 2017:db:abcd:f::f

Śledzenie trasy do 2017:db:abcd:f::f z maksymalną liczbą 30 przeskoków.

  1    42 ms   155 ms    78 ms  2017:db:abcd:a::1
  2    65 ms    77 ms    78 ms  2017:db:abcd:12::2
  3   123 ms    46 ms   274 ms  2017:db:abcd:15::5
  4   213 ms    91 ms    77 ms  2017:db:abcd:17::6
  5   315 ms   258 ms   552 ms  2017:db:abcd:f::f

Śledzenie zakończone.
```

*Skrypt 6.14. Badanie trasy pakietów w sieci IP z hosta PC2 do hosta PC3.*

```
C:\Users\PC2>tracert 2017:db:Abcd:f::f

Śledzenie trasy do 2017:db:abcd:f::f z maksymalną liczbą 30 przeskoków.

  1    60 ms   101 ms    84 ms  2017:db:abcd:d::4
  2   194 ms    83 ms    72 ms  2017:db:abcd:13::1
  3   258 ms    90 ms    79 ms  2017:db:abcd:12::2
  4   569 ms    45 ms    77 ms  2017:db:abcd:15::5
  5   156 ms   139 ms   125 ms  2017:db:abcd:17::6
  6   156 ms  1004 ms   989 ms  2017:db:abcd:f::f

Śledzenie zakończone.
```

Zmiana metryki na interfejsie szeregowym 0/1 routerze R2 spowodowała to, że istnieje tylko 1 alternatywna trasy pakietów z PC1 oraz PC2 do PC3. Dzięki takiej operacji można

narzucić zmianę przepływu danych pomiędzy hostami w celu ustawienia, przez który interfejs mają być przesyłane pakiety, aby nie przeciążać obu interfejsów routera. Mimo iż pakiet musi przebyć dłuższą drogę to opóźnienie czasowe jest nieznaczne ( w tym przypadku opóźnienia są wywołane zbyt dużym obciążeniem sieci).

## 6.2 Badanie OSPFv3

Kolejnym badanym protokołem routingu dynamicznego jest OSPF. Na początku badań sprawdzono przebieg tras pakietów z PC1 do PC2 i PC3 oraz PC2 i PC3 bez żadnych zmian w konfiguracji sieci. W tym celu użyto polecenia służącego do sprawdzenia przebiegu trasy *tracert*.

*Skrypt 6.15. Trasa przebiegu pakietu od hosta PC1 do hosta PC2.*

```
C:\Users\PC1>tracert 2017:db:abcd:d::d

Śledzenie trasy do 2017:db:abcd:d::d z maksymalną liczbą 30 przeskoków.

  1   201 ms    51 ms   763 ms  2017:db:abcd:a::1
  2   311 ms   265 ms   248 ms  2017:db:abcd:13::4
  3   187 ms  1142 ms   424 ms  2017:db:abcd:d::d

Śledzenie zakończone.
```

*Skrypt 6.16. Trasa przebiegu pakietu od hosta PC1 do hosta PC3.*

```
C:\Users\PC1>tracert 2017:db:abcd:f::f

Śledzenie trasy do 2017:db:abcd:f::f z maksymalną liczbą 30 przeskoków.

  1    59 ms    46 ms    62 ms  2017:db:abcd:a::1
  2    31 ms    30 ms    46 ms  2017:db:abcd:12::2
  3    59 ms   331 ms    93 ms  2017:db:abcd:18::6
  4   557 ms    55 ms   814 ms  2017:db:abcd:f::f

Śledzenie zakończone
```

*Skrypt 6.17. Trasa przebiegu pakietu od hosta PC1 do hosta PC2.*

```
C:\Users\PC2>tracert 2017:db:abcd:f::f

Śledzenie trasy do 2017:db:abcd:f::f z maksymalną liczbą 30 przeskoków.

  1    99 ms   272 ms   136 ms  2017:db:abcd:d::4
  2    78 ms    92 ms    93 ms  2017:db:abcd:18::6
  3   268 ms   825 ms   139 ms  2017:db:abcd:f::f

Śledzenie zakończone.
```

Z otrzymanych powyżej skryptów odczytano następujące wyniki: trasa od PC1 do PC2 przebiega przez R1, następnie R4, finalnie pakiet dociera do punktu docelowego. Trasa od PC1 do PC3 przebiega przez R1, R2, R6, a ostatecznie pakiet dociera do adresu docelowego. Trasa od PC2 do PC 3 przebiega przez R4 oraz R6.

Jednakże, po ponownym sprawdzeniu przebiegu trasy, pakiety zostały przesłane przez router R2 co świadczy o zmianie drogi pakietu. Dzieje się tak ponieważ, pomiędzy hostami PC1 oraz PC3 istnieją dwie alternatywne trasy (przez R2 oraz R4). Pakiety, które będą miały trafić do sieci *2017:db:abcd:f::/64* będą wysyłane do routera R2 jak i R4, gdyż posiadają

metrykę, która jest zależna od ustawienia kosztu dla danego interfejsu jak i jego przepustowości. Sprawdzono tablicę routingu w celu weryfikacji liczby tras alternatywnych do sieci *2017:db:abcd:f::/64*. Na routerze R1 użyto polecenia *show ipv6 route.*

*Skrypt 6.18. Sprawdzenie tablicy routingu na tablicy R1 w celu weryfikacji trasy do sieci 2017:DB:ABCD:F::/64 oraz 2017:DB:ABCD:D::/64.*

```
R1#sh ipv6 route
C   2017:DB:ABCD:A::/64 [0/0]
     via ::, FastEthernet0/0
L   2017:DB:ABCD:A::1/128 [0/0]
     via ::, FastEthernet0/0
O   2017:DB:ABCD:D::/64 [110/74]
     via FE80::4, Serial0/1
O   2017:DB:ABCD:F::/64 [110/138]
     via FE80::2, Serial0/0
     via FE80::4, Serial0/1
```

Z powyższej tablicy routingu odczytano, że w tablicy są zawarte dwie alternatywne trasy przez interfejsy szeregowe 0/0 routera R2 oraz 0/1 routera R4. W rzeczywistości pakiet zostaje przesłany raz przez interfejs szeregowy 0/0, a kolejny raz przez interfejs 0/1 (pakiety będą przesyłane naprzemiennie). Wynika to z takiej samej wartości metryki przez oba interfejsy. Dla sieci przyłączonych bezpośrednio do routera wartość administracyjna wynosi 0 , a dla procesu routingu OSPFv3 wynosi 110.

Następnie obserwowano komunikaty protokoły routingu za pomocą polecenia debug oraz przy użyciu analizatora Wireshark. Na początku użyto komendy *debug ipv6 ospf packet* na routerze R4.

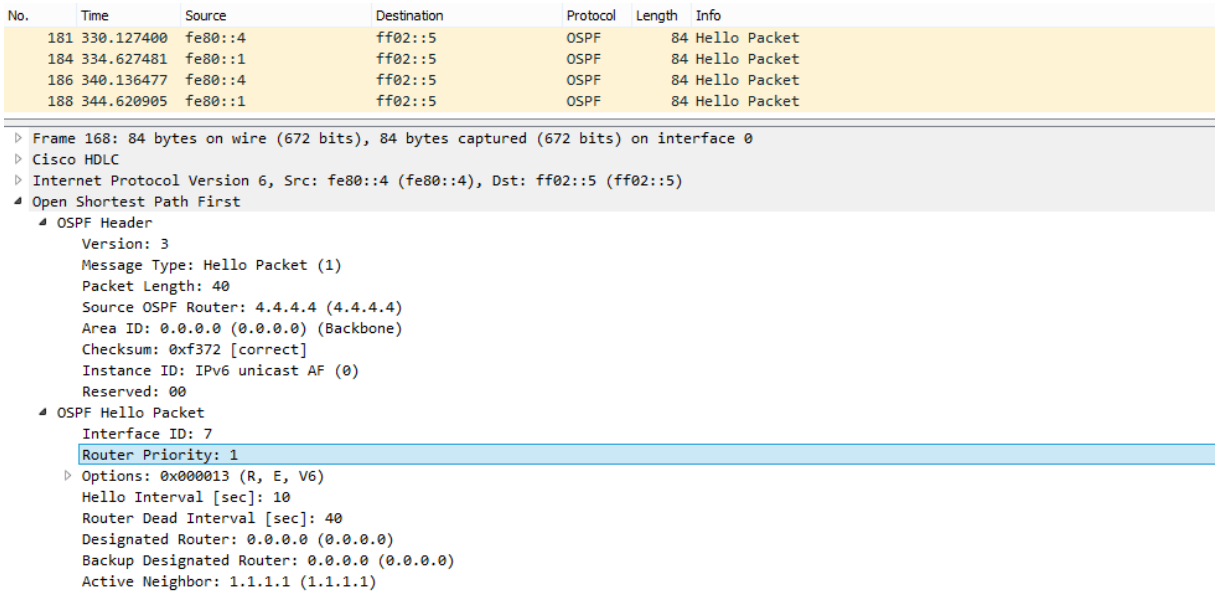
*Skrypt 6.19. Obserwacja komunikatów protokołu routingu OSPF.*

```
R4#
*Mar  1 00:12:35.091: OSPFv3: rcv. v:3 t:1 l:40 rid:1.1.1.1
     aid:0.0.0.0 chk:F375 inst:0 from Serial0/1
R4#
R4#
*Mar  1 00:12:37.135: OSPFv3: rcv. v:3 t:1 l:40 rid:6.6.6.6
     aid:0.0.0.0 chk:E965 inst:0 from Serial0/0
```

Jedynymi komunikatami routera są otrzymane informacje od routerów sąsiadujących, czyli routera R1 oraz R6. Otrzymywane są pakiety typu hello o długości 40 bajtów. W komunikacie jest zawarty numer identyfikacyjny routera, suma kontrolna oraz od jakiego interfejsu został otrzymany komunikat [18].

Komunikaty OSPF w bardzo małym stopniu zużywają przepustowość łącza w porównaniu do komunikatów RIPng.

Na koniec zbadano komunikaty przy pomocy analizatora Wireshark. Na rysunku 6.2 przedstawiono przesyłane komunikaty przez router R1 oraz R4.




*Rys.6.2. Przesyłane komunikat OSPF routera R1 oraz R4.*

Pakiety hello wysyłane przez routery R1 oraz R4 przesyłane są na adres ff02::5, który jest adresem multicast. W nagłówku OSPF zawarte są informacje o wersji protokołu OSPF (wersja 3), adresie źródłowy routera OSPF, a także wartości sumy kontrolnej. W pakiecie hello zawarte są informacje co jaki okres czasu wysyłane są pakiety (10 sekund), po jakim czasie zostanie zerwane połączenie z powodu braku komunikacji (40 sekund), a także z jaką siecią OSPF jest nawiązane aktywne połączenie.

Kolejnym krokiem było zmiana przebiegu trasy pakietów. W celu zmiany przebiegu trasy pakietów wyłączono interfejs szeregowy 0/0 na routerze R4 oraz interfejsie szeregowym 0/3 na routerze R2. Użyto polecenia *shutdown* we właściwościach danego interfejsu.

*Skrypt 6.20. Potwierdzenie wyłączenia interfejsów.*

```
%OSPFv3-5-ADJCHG: Process 1, Nbr 6.6.6.6 on Serial0/0 from FULL to DOWN, Neighbor Down: Interface down or
detached
*Mar  1 00:34:49.455: %SYS-5-CONFIG_I: Configured from console by console
*Mar  1 00:34:50.547: %LINK-5-CHANGED: Interface Serial0/0, changed state to administratively down
*Mar  1 00:34:51.547: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state to down
*Mar  1 00:35:48.427: %OSPFv3-5-ADJCHG: Process 1, Nbr 6.6.6.6 on Serial0/3 from FULL to DOWN, Neighbor Down:
Interface down or detached
*Mar  1 00:35:50.423: %LINK-5-CHANGED: Interface Serial0/3, changed state to administratively down
*Mar  1 00:35:51.423: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/3, changed state to down
```

Powyższe komunikaty od routerów potwierdzają wyłączenie interfejsów. Wyłączenie ich spowodowało, że proces routingu przeszedł ze stanu FULL do stanu DOWN, innemu słowy nie będą wysyłane i odbierane aktualizacje OSPF przez dany interfejs.

Następnie sprawdzono przebieg trasy między hostami PC1 oraz PC3, a także PC2 i PC3. Użyto polecenia *tracert* w wierszu poleceń.

*Skrypt 6.21. Trasa przebiegu pakietu od hosta PC1 do hosta PC3.*

```
C:\Users\PC1>tracert 2017:db:abcd:f::f

Śledzenie trasy do 2017:db:abcd:f::f z maksymalną liczbą 30 przeskoków.
```

```
  1    58 ms    46 ms    47 ms  2017:db:abcd:a::1
  2   169 ms    38 ms    34 ms  2017:db:abcd:12::2
  3    54 ms    68 ms    31 ms  2017:db:abcd:14::3
  4   499 ms   124 ms   438 ms  2017:db:abcd:19::6
  5   859 ms   180 ms   443 ms  2017:db:abcd:f::f

Śledzenie zakończone.
```

*Skrypt 6.22. Trasa przebiegu pakietu od hosta PC2 do hosta PC3.*

```
C:\Users\PC2>tracert 2017:db:abcd:f::f

Śledzenie trasy do 2017:db:abcd:f::f z maksymalną liczbą 30 przeskoków.

  1   219 ms   233 ms   114 ms  2017:db:abcd:d::4
  2    78 ms    46 ms    62 ms  2017:db:abcd:13::1
  3    52 ms    79 ms   133 ms  2017:db:abcd:12::2
  4   204 ms   130 ms   100 ms  2017:db:abcd:14::3
  5    46 ms   108 ms    77 ms  2017:db:abcd:19::6
  6   204 ms   263 ms   221 ms  2017:db:abcd:f::f

Śledzenie zakończone.
```

W obu skryptach zarejestrowano zmianę przebiegu trasy pakietów. Wyłączenie interfejsów nie spowodowało awarii sieci. W pierwszym przypadku trasa przebiega przez R1, R2, R3, R6, a następnie pakiet trafił do adresu docelowego. W drugim przypadku trasa przebiegała przez R4, a następnie trasa jest identyczna jak w pierwszym przypadku trasy. Tablica routingu routera R2 posiada dwie alternatywne trasy (router R3 lub router R5), dlatego też pakiety są przesyłane naprzemiennie przez oba interfejsy routerów. Następnie sprawdzono tablicę routingu na routerze R1 w celu weryfikacji wartości metryki.

*Skrypt 6.23. Sprawdzenie tablicy routingu routera R1 po wyłączeniu interfejsów.*

```
 R1#sh ipv6 route
O   2017:DB:ABCD:D::/64 [110/74]
     via FE80::4, Serial0/1
O   2017:DB:ABCD:F::/64 [110/202]
     via FE80::2, Serial0/0
```

Ze skryptu można dokonano analizy: wartość metryki wzrosła dla sieci 2017:DB:ABCD:F::/64. Przyczyną tego jest to, że pakiety będą musiały zostać przesłane przez dodatkowo jeden interfejs, a koszt jednego interfejsu jest równy 64 (wynika to z przepustowości interfejsu routera), co nam daje 138+64=202- nowa wartość metryki.

Następnie sprawdzono koszt metryki interfejsu szeregowego 0/1 na routerze R2 oraz zmieniono metrykę interfejsu szeregowego 0/1 routera R2 tak ,aby trasa prowadziła przez router R5.

*Skrypt 6.24. Sprawdzenie kosztu interfejsu szeregowego 0/1 routera R2.*

```
R2#show ipv6 ospf interface serial 0/1
Serial0/1 is up, line protocol is up
  Link Local Address FE80::2, Interface ID 7
  Area 0, Process ID 1, Instance ID 0, Router ID 2.2.2.2
  Network Type POINT_TO_POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:02
  Index 1/2/2, flood queue length 0
```

```
 Next 0x0(0)/0x0(0)/0x0(0)
 Last flood scan length is 2, maximum is 4
 Last flood scan time is 0 msec, maximum is 4 msec
 Neighbor Count is 1, Adjacent neighbor count is 1
   Adjacent with neighbor 3.3.3.3
 Suppress hello for 0 neighbor(s)
```

Pierwszym krokiem w celu zmiany metryki jest wejście we właściwości danego interfejsu, a na koniec użycie komendy: *ipv6 ospf cost 65*, gdzie 65 to nowy koszt trasy danego interfejsu. Następnie sprawdzono czy nastąpiła zmiana metryki za pomocą komendy *show ipv6 ospf interface s0/1*.

*Skrypt 6.25. Sprawdzenie poprawności wprowadzonych zmian.*

```
R2#show ipv6 ospf interface serial 0/1
Serial0/1 is up, line protocol is up
  Link Local Address FE80::2, Interface ID 7
  Area 0, Process ID 1, Instance ID 0, Router ID 2.2.2.2
  Network Type POINT_TO_POINT, Cost: 65
(...)
```

Zmiana metryki danego interfejsu przebiegła pomyślnie. Nowy koszt trasy z routera R1 do routera R6 poprzez R3 wynosi: 138+65=203. Następnie sprawdzono czy zmieniła się trasa pomiędzy hostem PC2, a hostem PC3.

*Skrypt 6.26. Sprawdzenie trasy pomiędzy hostem PC2 oraz PC3.*

```
C:\Users\PC2>tracert 2017:db:abcd:f::f

Śledzenie trasy do 2017:db:abcd:f::f z maksymalną liczbą 30 przeskoków.

  1   195 ms    56 ms    27 ms  2017:db:abcd:d::4
  2    46 ms    62 ms    30 ms  2017:db:abcd:13::1
  3    93 ms    62 ms    93 ms  2017:db:abcd:12::2
  4   134 ms   521 ms    79 ms  2017:db:abcd:15::5
  5   171 ms   109 ms   141 ms  2017:db:abcd:19::6
  6   284 ms   262 ms   201 ms  2017:db:abcd:f::f

Śledzenie zakończone.
```

Zmiana metryki na interfejsie szeregowym 0/1 w routerze R2 spowodowała zmianę trasy poprzez interfejs routera R5, gdyż jego koszt będzie mniejszy w porównaniu do trasy przez interfejs szeregowy routera R3. Manipulacja kosztem łącza za pomocą polecenia *ipv6 ospf cost* jest bardzo prostą metodą zmiany trasy. Wyłączenie jednego interfejsu szeregowego spowodowało aktualizację tablicy routingu dzięki czemu zostały zainstalowane nowe trasy do sieci będących przyłączony do hostów PC1, PC2 oraz PC2.