Agnieszka CHODOREK, Robert CHODOREK

O ZASTOSOWANIU EMULATORA NETKIT ORAZ RUTERÓW DOSTĘPOWYCH DO NAUCZANIA PROTOKOŁU OSPF

Streszczenie

Ruting wewnętrzny jest kluczowym elementem sieci danej firmy lub instytucji. Wpływa on efektywność i niezawodność sieci. W artykule przedstawiono zarys nauczania zagadnień rutingu wewnętrznego opartego na protokole OSPF przy wykorzystaniu ruterów dostępowych oraz emulatora sieci komputerowych Netkit.

WSTĘP

Sieci teleinformatyczne są jednym z istotnym elementów systemów logistycznych. Zapewniają one integrację systemów informatycznych i umożliwiają dostęp do nich z wielu lokalizacji. Integracja systemów własnych firm z publiczną siecią Internet jest ułatwiona dzięki podziałowi sieci Internet na systemy autonomiczne AS (ang. *Autonomous System*) [1][4]. System autonomiczny jest to zbiór sieci (lub prefiksów sieci) administrowanych przez jedną firmę lub instytucję. Istnienie systemów autonomicznych wymusiło podział rutingu na dwie klasy:

– ruting wewnętrzny,
– ruting zewnętrzny.

Ruting wewnętrzny realizowany jest wewnątrz systemu autonomicznego. Wykorzystuje on takie protokoły, jak RIP, RIPng, OSPFv2, OSPFv3, EIGRP czy IS-IS [8]. Szczególną uwagę należy zwrócić na protokół OSPF (*Open Shortest Path First*) [6], który jest otwartym standardem protokołu rutingu mogącym funkcjonować w zarówno w małych jak i dużych sieciach. Umożliwia on obsługę zarówno sieci z protokołem IPv4 – OSPFv2, jak i sieci z protokołem IPv6 – OSPFv3 [3].

Ruting zewnętrzny realizowany jest na zewnątrz systemów autonomicznych i współcześnie wykorzystuje tylko jeden protokół – protokół BGP (ang. *Border Gateway Protocol*) [1][7]. Ruting zewnętrzny jest rutingiem strategicznym, działającym pomiędzy systemami autonomicznymi.

W artykule zaprezentowano wybrane aspekty użycia protokołu OSPF w systemach autonomicznych, związane z nauczaniem zagadnień praktycznych rutingu wewnętrznego. Przedstawione zagadnienia mogą być realizowane zarówno na ruterach dostępowych firmy Cisco, jak i na ruterach programowych, pracujących pod kontrolą systemu operacyjnego Linux i korzystających z pakietu oprogramowania Zebra/Quagga. W przykładach pokazanych w artykule, rutery linuksowe pracowały w środowisku maszyn wirtualnych emulatora Netkit [2][5].

Dalsza część artykułu składa się z czterech rozdziałów. W rozdziale pierwszym zaprezentowano protokół OSPF. Rozdział drugi zawiera opis środowiska testowego, wykorzystywanego do realizacji procesu nauczania protokołu OSPF. Wybrane zagadnienia praktycznej realizacji rutingu OSPF na przykładzie ruterów programowych i sprzętowych przedstawiono w rozdziale trzecim. Rozdział czwarty stanowi podsumowanie artykułu.

1. PROTOKÓŁ OSPF

Protokół OSPF jest protokołem rutingu wewnętrznego, należącym do grupy protokołów stanu łącza. Obecnie wykorzystywana jest wersja 2 protokołu OSPF dla IPv4. Protokół OSPF dla IPv6 został zdefiniowany jako wersja 3 [3]. Począwszy od wersji 2, protokół OSPF pozwala na użycie masek podsieci. Umożliwia on również rozkładanie obciążeń pomiędzy równoważnymi ścieżkami. Posiada mechanizmy uwierzytelnienia informacji o rutigu, wymienianej pomiędzy ruterami.

OSPF buduje bazę danych stanów łączy, na podstawie której powstaje graf opisujący topologię sieci. Po utworzeniu grafu, każdy ruter OSPF niezależnie wyznacza w grafie trasy zgodnie z algorytmem SPF (ang. *Shortest Path First*; algorytm ten znany jest także jako algorytm Dijkstry) [6]. Algorytm SPF uwzględnia koszt każdej trasy (metrykę trasy) pomiędzy danym ruterem a siecią docelową. Typowo metryka trasy jest powiązana z przepustowością łącza (im szybsze łącze tym mniejszy koszt).

Informacja o topologii sieci, stanowiąca podstawę do wyznaczania tras w protokole OSPF, musi być identyczna we wszystkich węzłach sieci. Ponieważ każdy ruter OSPF niezależnie wyznacza trasę, różnice w budowie grafów, utworzonych przez różne rutery, mogą sprawić poważny kłopot. Dlatego w protokole OSPF szczególnie ważne jest zapewnienie odpowiedniej wymiany komunikatów rutingu. Ponieważ protokół OSPF nie przesyła danych z wykorzystaniem niezawodnego protokołu transportowego TCP (ang. *Transmission Control Protocol*), dlatego w protokole zostały zawarte mechanizmy potrzebne do niezawodnej wymiany komunikatów. Do tego celu, w protokole OSPF ustanawiane są relacje pomiędzy sąsiadami. Relacje te opisywane są poprzez stan, w jakim znajdują się sąsiedzi. W protokole OSPF zdefiniowano 7 stanów, pozwalających na niezawodne ustanowienie połączenia z sąsiadem. Są to:

– stan wyłączony (ang. *Down*),
– stan inicjalizacji (ang. *Init*),
– stan dwukierunkowy (ang. *Two-way*),
– stan gotowości (ang. *ExStart*),
– stan wymiany (ang. *Exchange*),
– stan uruchamiania (ang. *Loading*),
– stan pełnej przyległości (ang. *Full adjency*).

Stany są elementem zapewnienia niezawodności. Rutery OSPF przechodzą przez te stany, co daje efekt podobny, jak nawiązywanie połączenia w protokole TCP. Proces kończy się po ustanowieniu stanu pełnej przyległości. Następnie przesyłane są komunikaty. Komunikaty są przesyłane z potwierdzeniami odbioru i powtórzeniami w sytuacji braku potwierdzenia.

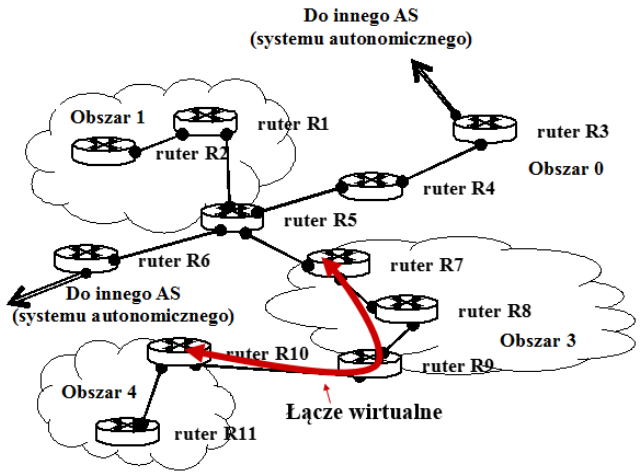
Sposób ustanawiania relacji pomiędzy ruterami jest zależny od rodzaju sieci. Rodzaje sieci wpływa też na reprezentację tej sieci w grafie tworzonym przez OSPF. W protokole OSPF zdefiniowano 4 rodzaje sieci:

– sieć rozgłoszeniowa (ang. *broadcast*),
– sieć punkt-punkt (ang. *point-to-point*),
– sieć nierozgłoszeniowa (ang. *non-broadcast multiple access network, NBMA*),
– sieć wielopunktowa (ang. *point-to-multipoint, PTMP*).

Sieć rozgłoszeniowa jest to sieć, która w sposób rozsiewczy realizuje dostarczanie komunikatu do wszystkich stacji podłączonych do łącza. Przykładem sieci rozgłoszeniowej jest sieć Ethernet. Sieć punkt-punkt to sieć naturalnie przeznaczona do pracy punkt-punkt, np. łącze szeregowe RS-232. Sieć nierozgłoszeniowa pozwala na połączenie wielu stacji, ale komunikaty są przesyłane naturalnie pomiędzy dwiema stacjami. Przykładem sieci nierozgłoszeniowej jest sieć ATM (*Asynchronous Transfer Mode*) czy *Frame Relay*. Sieć wielopunktowa jest to sieć nierozgłoszeniowa posiadająca możliwość zestawiania połączeń punkt-wielopunkt. Przykładem takiej sieci jest sieć ATM skonfigurowana przez administratora do pracy wielopunktowej.

W przypadku sieci rozgłoszeniowych, w danej podsieci może funkcjonować wiele ruterów OSPF i mogą one bezpośrednio wymieniać komunikaty rutingu. Aby usprawnić proces wymiany komunikatów, w sieci rozgłoszeniowej ograniczana jest liczba bezpośrednich przyległości pomiędzy ruterami. W tym celu, w sieci rozgłoszeniowej wyróżniane są dwa rutery: ruter DR (ang. *Designated Router*) i ruter BDR (ang. *Backup Designated Router*). Ruter DR koncentruje informacje rutingowe w danej sieci rozgłoszeniowej. Ustanawia on przyległości ze wszystkimi ruterami w danej sieci. Odbiera on zatem wszystkie informacje rutingowe od każdego ze swoich sąsiadów, a następnie redystrybuuje je do wszystkich sąsiadów. Mechanizm ten usprawnia niezawodną wymianę informacji rutingowych ale sprawia, że ruter DR staje się punktem krytycznym danej sieci. Jakakolwiek awaria tego rutera mogłaby zablokować protokół rutingu OSPF. Dlatego też wraz z powołaniem rutera DR powoływany jest automatycznie ruter zapasowy BDR, który w przypadku awarii rutera DR przejmuje jego rolę.

W pozostałych typach sieci nie występują problemy z bezpośrednim komunikowaniem się dużej liczby ruterów. W sieci punkt-punkt ruter posiada tylko jednego sąsiada. Jest nim ruter na drugim końcu łącza punkt-punkt. W sieci nierozgłoszeniowej i wielopunktowej mogą funkcjonować więcej niż dwa rutery, jednak w sieci nierozgłoszeniowej nie mogą one skomunikować się bezpośrednio (nie wszystkie na raz) i administrator nie może zestawić połączeń wielopunktowych. W sieci wielopunktowej rutery nie mogą skomunikować się automatycznie, na podobieństwo sieci rozgłoszeniowej, nawet skonfigurowanie ich przez administratora nie daje im pełnej funkcjonalności sieci rozgłoszeniowej.



Rys. 1. *Podział systemu autonomicznego na obszary*

Protokół OSPF może być stosowany zarówno w małych, jak i dużych sieciach, przy czym, z punktu widzenia OSPF, wielkość sieci zależy od liczby pracujących w niej ruterów. Posiada on stosowne rozwiązania zapewniające skalowalność protokołu nawet do obsługi dużych systemów autonomicznych. W przypadku dużych sieci (o dużej liczbie ruterów) dokonywany jest podział sieci na obszary (Rys. 1). Wyróżniany jest jeden obszar łączący wszystkie obszary – obszar szkieletowy (ang. *OSPF backbone*) oznaczany jako `OSPF Area 0` (często w konfiguracji zapisywany w postaci `Area 0.0.0.0`). Pozostałe obszary muszą być dołączone do tego obszaru bezpośrednio lub poprzez łącza wirtualne (ang. *virtual links*). Przykładem łącza wirtualnego jest łącze pomiędzy ruterami R7 i R10, zaznaczone na Rys. 1 czerwoną linią. Po podziale sieci na obszary, grafy opisujące sieć są tworzone osobno dla każdego obszaru. Informacje o sieciach z danego obszaru są przekazywane do obszaru szkieletowego, z którego są dystrybuowane do pozostałych obszarów.

Podział na obszary powoduje wyodrębnienie czterech typów ruterów spełniających określone funkcje. Są to:
- rutery szkieletowe BR (ang. *Backbone Router*),
- rutery IR (ang. *Internal Routers*, dosł. rutery wewnętrzne),
- rutery brzegowe ABR (ang. *Area Border Routers*),
- rutery ASBR (ang. *AS boundary router*).

Rutery szkieletowe pracują wewnątrz obszaru szkieletowego. Ruterem BR jest, przykładowo, ruter R4 na Rys. 1. Rutery IR pracują wewnątrz obszaru nie będącego obszarem szkieletowym (np. ruter R1 na Rys. 1). Rutery brzegowe pracują na granicy dwóch obszarów: obszaru nie będącego obszarem szkieletowym i obszaru szkieletowego. Ruterem ABR jest, np., ruter R5 na Rys. 1. Rutery ASBR pracują na granicy systemów autonomicznych. Przykładowo, jeśli operator obsługuje dwa duże systemy autonomiczne, które łączy dodatkowo przez protokół OSPF. Przykładem rutera ASBR jest ruter R3 na Rys. 1.

2. ŚRODOWISKO TESTOWE

W artykule przedstawiono realizację przykładowego ćwiczenia laboratoryjnego z wykorzystaniem protokołu OSPF. Ćwiczenie zostało wykonane równolegle w dwóch środowiskach testowych:
- sprzętowym, zbudowanym w oparciu o rutery CISCO,
- programowym, zbudowanym w oparciu o oprogramowanie narzędziowe Zebra/Quagga i emulator NetKit [2].

Ćwiczenie może być realizowane równolegle w środowisku sprzętowym i programowym (w celu pokazania specyfiki konfigurowania OSPF w każdym z tych środowisk) lub tylko w jednym z nich.

Tab. 1. *Przykładowe parametry konfiguracyjne sieci o topologii przedstawionej na rysunku 2*

identyfikator rutera	adres własny sieci	maska sieci
S1	80.26.1.0	255.255.255.0
S2	80.26.2.0	255.255.255.0
S3	80.26.3.0	255.255.255.0
S4	80.26.4.0	255.255.255.0
S5	80.26.5.0	255.255.255.0

Sieć testowa łączy stacje robocze PC1 i PC2 (rys. 2). Składa się ona z pięciu sieci, S1, S2, S3, S4 i S5. Dla ułatwienia zapamiętania adresów własnych poszczególnych sieci, zawartość trzeciego oktetu adresu zawsze jest równa numerowi kolejnemu sieci. Przykładowe parametry konfiguracyjne sieci testowej pokazanej na rysunku 2 (adresy własne sieci S1...S5 oraz maski sieci S1...S5) zostały zamieszczone w tabeli 1. Stacja robocza PC1 została podłączona do sieci S1, stacja robocza PC2 do sieci S5. Adres IP stacji PC2 to 80.26.1.5 z puli adresowej sieci S1. Stacji roboczej PC2 przydzielono adres IP 80.26.5.6 z puli adresowej sieci S5.

Tab. 2. *Przykładowe parametry konfiguracyjne ruterów pokazanych na rysunku 2*

identyfikator rutera	adres IP interfejsu eth0 lub FastEthernet 0/0	adres IP interfejsu eth1 lub FastEthernet 0/1	adres IP interfejsu eth2 lub FastEthernet 1/0
R1	80.26.1.1	80.26.2.1	-
R2	80.26.2.4	80.26.3.4	-
R3	80.26.2.9	80.26.4.9	-
R4	80.26.3.6	80.26.4.6	80.26.5.6

W sieci testowej wykorzystywane są cztery rutery i jeden switch. Ruter R1 łączy sieci S1 i S2. Sieć S2 (na rysunku 2 zaznaczona chmurą) obejmuje switch SW1 i łączy ze sobą interfejsy trzech ruterów (R1, R2 i R3). Ruter R2 łączy ze sobą sieci S2 i S3, a ruter R3 sieci S2 i S4. Ruter R4 podłączony jest bezpośrednio do trzech sieci, S3, S4 i S5. Przykładowe adresy IP poszczególnych interfejsów ruterów R1...R4 zostały pokazane w tabeli 2.

W przypadku użycia środowiska sprzętowego, należy przygotować sieć o zadanej topologii i wstępnie skonfigurować rutery sprzętowe zgodnie z tabelą 2. W ruterach Cisco, wykorzystywanych na laboratorium z przedmiotu „Ruting wewnętrzny w sieciach IP", prowadzonego na krakowskiej Akademii Górniczo-Hutniczej, studenci AGH mają do dyspozycji rutery Cisco serii 2800. Są one wyposażone w dwa typy interfejsów: standardu Ethernet (Fast Ethernet) i łącza szeregowe synchroniczne.

```
r1:~# /usr/lib/quagga/zebra -d
r1:~# /usr/lib/quagga/ospfd -d
r1:~#
```

Rys. 3. *Uruchamianie modułu Zebra i demona rutingu protokołu OSPF*

W przypadku użycia środowiska programowego (ruterów linuksowych i emulatora NetKit), należy przygotować w emulatorze konfigurację sieci o zadanej topologii (konfiguracja umieszczana jest w pliku *lab.conf* w przypadku pracy w trybie "I" z tzw. I-poleceniami emulatora Netkit, przeznaczonymi do całościowego zarządzania wirtualnym laboratorium [2]) i uruchomić laboratorium (w trybie "I" jest to realizowane poleceniem `lstart` [2]). W katalogu `etc/quagga` każdego rutera należy zamieścić pliki startowe Zebry i protokołu OSPF. Następnie należy uruchomić demony rutingu z pakietu oprogramowania Zebra/Quagga (w trybie demona, opcja `-d`). Przykład uruchomienia demonów rutingu został zamieszczony na rysunku 3. W przykładzie pokazanym na rysunku 3, w każdym z ruterów uruchamiany jest demon modułu Zebra i demon procesu rutingu OSPF.

3. KONFIGURACJA I TESTY RUTINGU OSPF - STUDIUM PRZYPADKU

W rozdziale przedstawiono wybrane zagadnienia praktyczne, obejmujące konfigurację ruterów OSPF i analizę zawartości tablic rutingu.

Aby uruchomić ruting OSPF należy dokonać szeregu czynności konfiguracyjnych. Jest to realizowane poprzez konsolę rutera dostępowego. W przypadku rutera programowego, korzystającego z oprogramowania Zebra/Quagga, konfigurując ruter należy skorzystać z usługi zdalnego terminala, który należy uruchomić w każdym węźle sieci.

(a)

```
r1:~# telnet localhost ospfd
Trying 127.0.0.1...
Connected to r1.
Escape character is '^]'.

Hello, this is Quagga (version 0.99.10).
Copyright 1996-2005 Kunihiro Ishiguro, et al.


User Access Verification

Password:
R1_ospfd>
```

(b)

```
R1_ospfd> enable
Password:
```

(c)

```
R1_ospfd# show ip ospf
 OSPF Routing Process not enabled
R1_ospfd#
```

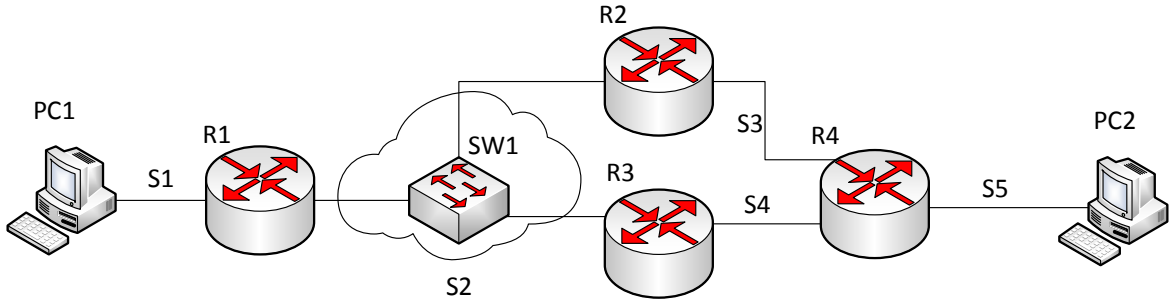
(d)

```
R1_ospfd# configure terminal
R1_ospfd(config)#
```

Rys. 4. *Czynności przygotowawcze: a) łączenie się z demonem rutingu OSPF, b) przejście do trybu uprzywilejowanego, c) sprawdzenie czy w ruterze jest uruchomiony proces rutingu, d) wejście do trybu konfiguracji demona rutingu OSPF*

Z demonem rutingu protokołu OSPF można połączyć się poprzez terminal znakowy telnet (Rys. 4a). Po podłączeniu do demona rutingu można przejść do trybu uprzywilejowanego, w którym możliwe jest zarządzanie ruterem (Rys. 4b). Po uruchomieniu demona rutingu, odczytuje on konfigurację z pliku *ospfd.conf* (zlokalizowanego w katalogu `/etc/quagga/` systemu plików każdego z ruterów). Jeżeli w plik ten zawiera jedynie podstawową konfigurację samego demona (nazwa podawana w wierszu poleceń po zalogowaniu, hasła), to proces rutingu OSPF nie jest uruchamiany, gdyż nie posiada odpowiednich parametrów konfiguracyjnych. O tym, czy proces rutingu OSPF został uruchomiony można dowiedzieć się wydając polecenie `show ip ospf` - Rys. 4c. Aby przejść do konfiguracji demona rutingu OSPF należy wydać polecenie `configure terminal` (Rys. 4d).

Uruchomienie procesu rutingu OSPF wymaga jawnego wskazania, że protokół OSPF ma zostać uruchomiony. Takim wskazaniem jest polecenie `router ospf` (Rys. 5a). Należy wówczas przystąpić do konfigurowania protokołu OSPF, m.in. określając



Rys. 2. Przykładowe środowisko testowe

topologię sieci w otoczeniu danego rutera. W praktyce, opis topologii sprowadza się do wskazania interfejsów, na których ma pracować protokół OSPF (Rys. 5b) i przypisania ich do odpowiednich obszarów. Interfejsy podaje się poprzez wskazanie sieci, do których należą. Dla wskazanych interfejsów na podstawie aktualnie ustawione przepustowości interfejsu określany jest koszt łącza. Wskazane sieci będą uwzględniane w całkowitej topologii sieci, wyznaczanej przez protokół OSPF.

(a)

```
R1_ospfd(config)# router ospf
R1_ospfd(config-router)#
```

(b)

```
R1_ospfd(config-router)# network 80.26.1.0/24 area 0
R1_ospfd(config-router)# network 80.26.2.0/24 area 0
```

(c)

```
R1_ospfd# show ip ospf
 OSPF Routing Process, Router ID: 80.26.2.1
 Supports only single TOS (TOS0) routes
 This implementation conforms to RFC2328
```

Rys. 5. *Konfiguracja protokołu OSPF w ruterze R1: a) włączenie procesu rutingu OSPF, b) zdefiniowanie interfejsów na których pracuje OSPF, c) początkowe linie informacji o procesie rutingu OSPF*

Dla wskazanych interfejsów, na podstawie aktualnie ustawionej przepustowości interfejsu określany jest koszt łącza. Użyta w niniejszym studium przypadku sieć Ethernet może pracować z jedną z 3 przepustowości: 10 Mb/s, 100 Mb/s i 1 Gb/s. Koszt ten wyliczany jest przez podzielenie bazowej przepustowości 100 Mb/s przez przepustowość interfejsu. W emulatorze NetKit standardowo ustawiana jest przepustowość 10 Mb/s. W ruterach Cisco interfejsy FastEthernet ustawiały się automatycznie na 100 Mb/s (jedynie przy bardzo starym switchu interfejs mógł mieć ustawioną przepustowość 10 Mb/s). Stąd koszty tras w analizowanych przykładach wynoszą 10 dla emulatora NetKit (10 Mb/s odniesione do referencyjnego 100 Mb/s), oraz 1 w przypadku ruterów Cisco (100 Mb/s odniesione do 100 Mb/s).

(a)

```
r1:~# telnet localhost zebra
```

(b)

```
R1> show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 80.26.1.0/24 is directly connected, eth0
C>* 80.26.2.0/24 is directly connected, eth1
C>* 127.0.0.0/8 is directly connected, lo
R1>
```

(c)

```
R1> show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

O   80.26.1.0/24 [110/10] is directly connected, eth0, 00:04:34
C>* 80.26.1.0/24 is directly connected, eth0
O   80.26.2.0/24 [110/10] is directly connected, eth1, 00:04:18
C>* 80.26.2.0/24 is directly connected, eth1
C>* 127.0.0.0/8 is directly connected, lo
R1>
```

Rys. 6. *Praca z modułem Zebra: a) łączenie się z demonem Zebra, b) wyświetlenie tablicy rutingu przed uruchomieniem demona rutingu OSPF, c) wyświetlenie tablicy rutingu po uruchomieniu demona rutingu OSPF*

Na rysunku 5c przedstawiono informację o procesie rutingu OSPF po konfiguracji rutera programowego zgodnie z rysunkiem 5b. Widoczny jest identyfikator rutera OSPF (`80.26.2.1`). Jest to najwyższy numerycznie adres IP aktywnych interfejsów rutera przed uruchomieniem procesu OSPF. Ruter R1 ma dwa aktywne interfejsy, którym przypisano adresy IP `80.26.1.1` oraz `80.26.2.1` (tab. 2).

Każdy ruter posiada co najmniej dwie tablice rutingu:

- RIB (ang. *Routing Information Base*),
- FIB (ang. *Forwarding Information Base*).

Pierwsza pracuje w tzw. płaszczyźnie sterowania (ang. *control plane*) i, w analizowanym przypadku, jest związana z modułem Zebra. Druga pracuje w tzw. płaszczyźnie przekazywania (ang. *forwarding plane*) i jest to tablica rutingu jądra systemu operacyjnego. Protokół pracujący w płaszczyźnie sterowania (analizowanym przypadku tym protokołem jest OSPF) przygotowuje tablicę RIB. Tablica RIB jest tablicą efektywną i dla samego wyboru tras ma charakter pomocniczy. Tablica FIB jest tablicą decyzyjną i trafiają do niej najlepsze trasy zgodnie z preferencjami administracyjnymi (kosztem administracyjnym).

Tablica rutingu RIB analizowanego rutera programowego jest koordynowana przez moduł Zebra. Aby zobaczyć efektywną tablicę rutingu należy połączyć się z demonem Zebra (Rys. 6a), a następnie używając polecenia `show ip route`, wyświetlić tablicę rutingu RIB (Rys. 6b). Polecenie pokazane na rysunku 6b zostało wydane przed uruchomieniem demona rutingu OSPF, d dlatego w tablicy RIB nie ma żadnych tras pochodzących od protokołu OSPF.

Po uruchomieniu, proces rutingu OSPF będzie miał wpływ na tablice rutingu RIB i FIB (Rys. 6c). Ponieważ w ruterze może funkcjonować jednocześnie wiele procesów rutingu, ich wpływ na tablice rutingu określa tzw. koszt administracyjny. W przypadku protokołu OSPF koszt administracyjny wynosi 110. Koszt administracyjny jest widoczny we wpisach o trasach pochodzących od protokołu OSPF (na Rys. 6c są to wiersze rozpoczynające się od litery „O") jako pierwszy parametr trasy umieszczony w nawiasach kwadratowych (`[110/10]`). Drugi parametr (tu: `10`), wskazuje na metrykę trasy prowadzącej przez łącze Ethernet 10 MB/s.

Jak widać na rysunku 6c, trasy OSPF (oznaczane symbolem „O") nie są uwzględniane do bieżącego rutingu (choć są widoczne w tablicy rutingu RIB), gdyż istnieją połączenia bezpośrednie do sieci docelowych (oznaczane symbolem „C"). Decyzje rutingowe (ang. *forwarding*) podejmowane są w oparciu o trasy „C" (na rysunku 6c zaznaczone symbolem gwiazdki, „*"). Trasy OSPF istnieją, ale ruter ich nie wykorzystał do rutingu. Na rysunku 6c widoczny jest też czas od ostatniej aktualizacji wpisu o obu trasach OSPF.

```
R1(config)#router ospf 40
R1(config-router)#
R1(config-router)#network 80.26.1.0 0.0.0.255 area 0
R1(config-router)#network 80.26.2.0 0.0.0.255 area 0
R1(config-router)#
```

Rys. 7. *Konfiguracja protokołu OSPF w sprzętowym ruterze R1 (Cisco)*

Korzystając z rutera programowego (linuksowego), należało najpierw skorzystać z terminala demona OSPF, a następnie z terminala modułu Zebra. W przypadku ruterów Cisco całość konfiguracji i weryfikacji protokołu OSPF jest realizowana z jednego terminala. W przypadku konfigurowania protokołu OSPF dodatkowo (w porównaniu z pakietu oprogramowania Zebra/Quagga) należy podać numer procesu OSPF. W omawianym przypadku jest to numer `40` (rys. 7). Podanie numeru pozwala na uruchamianie wielu procesów OSPF na jednym ruterze (ma to zastosowanie w zaawansowanych konfiguracjach). Każdorazowa modyfikacja konfiguracji protokołu OSPF wymaga podania numeru procesu, (tego samego, który został nadany podczas pierwszego uruchomienia procesu). Numer ten może być różny na różnych ruterach.

(a)

```
R2_ospfd(config)# router ospf
R2_ospfd(config-router)# network 80.26.2.0/24 area 0
R2_ospfd(config-router)# network 80.26.3.0/24 area 0
R2_ospfd(config-router)#
```

(b)

```
R3_ospfd(config)# router ospf
R3_ospfd(config-router)# network 80.26.2.0/24 area 0
R3_ospfd(config-router)# network 80.26.4.0/24 area 0
R3_ospfd(config-router)#
```

(c)

```
R4_ospfd(config)# router ospf
R4_ospfd(config-router)# network 80.26.3.0/24 area 0
R4_ospfd(config-router)# network 80.26.4.0/24 area 0
R4_ospfd(config-router)# network 80.26.5.0/24 area 0
R4_ospfd(config-router)#
```

Rys. 8. *Konfiguracja protokołu OSPF w ruterze: a) R2, b) R3, c) R4*

Analogicznie do konfiguracji rutera R1, konfigurowane są rutery R2, R3 i R4 (Rys. 8). W przypadku rutera R4 (Rys. 8c) w konfiguracji widoczne są 3 sieci (w pozostałych ruterach dwie sieci) gdyż ruter ten, jako jedyny w analizowanej topologii, podłączony jest do trzech sieci (S3, S4, S5).

(a)

```
R1_ospfd> show ip ospf neighbor

    Neighbor ID Pri State           Dead Time Address         Interface
   RXmtL RqstL DBsmL
80.26.3.4         1 Full/Backup       31.727s 80.26.2.4       eth1:80.26.2.1
      0     0     0
R1_ospfd>
```

(b)

```
R2_ospfd# show ip ospf neighbor

    Neighbor ID Pri State           Dead Time Address         Interface
   RXmtL RqstL DBsmL
80.26.2.1         1 Full/DR           36.782s 80.26.2.1       eth0:80.26.2.4
      0     0     0
R2_ospfd#
```

(c)

```
R1_ospfd> show ip ospf neighbor

    Neighbor ID Pri State           Dead Time Address         Interface
   RXmtL RqstL DBsmL
80.26.3.4         1 Full/Backup       30.979s 80.26.2.4       eth1:80.26.2.1
      0     0     0
80.26.4.9         1 Full/DROther      30.990s 80.26.2.9       eth1:80.26.2.1
      0     0     0
R1_ospfd>
```

(d)

```
R1#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
80.26.3.4         1   FULL/BDR        00:00:35    80.26.2.4       FastEthernet0/1
80.26.4.9         1   FULL/DROTHER    00:00:36    80.26.2.9       FastEthernet0/1
R1#
```

Rys. 9. *Informacje o sąsiadach w protokole OSPF obserwowane w: a) ruterze programowym R1 bezpośrednio po podłączeniu R2, b) ruterze programowym R2 po jego uruchomieniu, c) ruterze programowym R1 po uruchomieniu R2 i R3, d) ruterze Cisco R1 po uruchomieniu R2 i R3*

Po uruchomieniu ruterów R1 i R2 można zaobserwować utworzenie pomiędzy nimi relacji sąsiedzkich. Jest to stan pełnej przyległości. Ruter R1, uruchomiony jako pierwszy, jest ruterem DR w sieci s2 (Rys. 9a). Ruter R2 jest w sieci s2 ruterem BDR (Rys 9b). Kolejny ruter, R3, uruchomiony po ruterach R1 i R2, nie jest ani ruterem DR, ani ruterem BDR (Rys. 9c). Na rysunkach 9a, 9b i 9c widoczne są też informacje o: liczbie retransmisji pakietów LS przenoszących dane o stanie łącza (`RXmtL`), liczbie żądań retransmisji stanu łącza (`RqstL`) i liczbie aktualizacji bazy topologii (`DBsmL`). W analizowanym przypadku wszystkie te wielkości przyjmują wartość zero. Widoczny jest też zegar *Dead Timer*, który upłynął od chwili odebrania wiadomości OSPF od sąsiada.

(a)

```
R3_ospfd# show ip ospf neighbor

    Neighbor ID Pri State           Dead Time Address         Interface
   RXmtL RqstL DBsmL
80.26.2.1         1 Full/DR           30.074s 80.26.2.1       eth0:80.26.2.9
      0     0     0
80.26.3.4         1 Full/Backup       30.072s 80.26.2.4       eth0:80.26.2.9
      0     0     0
80.26.5.6         1 Full/Backup       38.477s 80.26.4.6       eth1:80.26.4.9
      0     0     0
R3_ospfd#
```

(b)

```
R4_ospfd# show ip ospf neighbor

    Neighbor ID Pri State           Dead Time Address         Interface
   RXmtL RqstL DBsmL
80.26.3.4         1 Full/DR           31.332s 80.26.3.4       eth0:80.26.3.6
      0     0     0
80.26.4.9         1 Full/DR           35.458s 80.26.4.9       eth1:80.26.4.6
      0     0     0
R4_ospfd#
```

Rys. 10. *Informacje o sąsiadach w protokole OSPF po uruchomieniu wszystkich ruterów obserwowane w ruterze: a) R3, b) R4*

W przypadku ruterów Cisco (Rys. 9d) wyświetlane są takie same podstawowe informacje, jak w przypadku ruterów linuksowych. Brak jest szczegółowych statystyk odbioru komunikatów.

Po uruchomieniu wszystkich ruterów w każdej sieci są ustanowione relacje sąsiedzkie. W ruterze R3 widoczne są relacje w sieci S2 i S4 (Rys. 10a). W sieci S2 ruter R1 (o identyfikatorze `80.26.2.1`) jest ruterem DR, a ruter R2 (o identyfikatorze `80.26.3.4`) ruterem BDR. W sieci S4 widoczny jest ruter R4 (o identyfikatorze `80.26.5.6`), który jest tam ruterem BDR. W ruterze R4 widoczne są, z kolei, rutery R2 i R3 (Rys. 10b). Są one ruterami DR w sieciach, odpowiednio, S3 i S4.

(a)

```
R1> show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

O   80.26.1.0/24 [110/10] is directly connected, eth0, 01:01:20
C>* 80.26.1.0/24 is directly connected, eth0
O   80.26.2.0/24 [110/10] is directly connected, eth1, 01:01:04
C>* 80.26.2.0/24 is directly connected, eth1
O>* 80.26.3.0/24 [110/20] via 80.26.2.4, eth1, 00:52:11
O>* 80.26.4.0/24 [110/20] via 80.26.2.9, eth1, 00:46:20
O>* 80.26.5.0/24 [110/30] via 80.26.2.4, eth1, 00:43:36
  *                       via 80.26.2.9, eth1, 00:43:36
C>* 127.0.0.0/8 is directly connected, lo
R1>
```

(b)

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter a
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-I
       ia - IS-IS inter area, * - candidate default, U - per-user s
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     80.0.0.0/24 is subnetted, 5 subnets
C       80.26.2.0 is directly connected, FastEthernet0/1
O       80.26.3.0 [110/2] via 80.26.2.4, 00:56:10, FastEthernet0/1
C       80.26.1.0 is directly connected, FastEthernet0/0
O       80.26.4.0 [110/2] via 80.26.2.9, 00:06:20, FastEthernet0/1
O       80.26.5.0 [110/3] via 80.26.2.9, 00:04:23, FastEthernet0/1
                  [110/3] via 80.26.2.4, 00:04:23, FastEthernet0/1
R1#
```

Rys. 11. *Tablica rutingu rutera R1: a) w przypadku rutera programowego, b) dla rutera Cisco*

Ostateczna tablica rutingu protokołu OSPF pokazana dla rutera R1 została przedstawiona na rysunku 11. Widać w niej wszystkie sieci zamieszczone w tabeli 1. Są też widoczne dwie trasy do sieci s5 (`80.26.5.0`). W tablicach rutera Cisco (Rys. 11b) widać historyczny wpływ rutingu klasowego na sposób zapisu informacji o sieciach docelowych – sieć `80.0.0.0` została podzielona na 5 podsieci z maską 24-bitową.

```
R1_ospfd> show ip ospf database

       OSPF Router with ID (80.26.2.1)

                Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
80.26.2.1       80.26.2.1       1195 0x80000007 0x12f6 2
80.26.3.4       80.26.3.4       1032 0x80000007 0x91f5 2
80.26.4.9       80.26.4.9       1028 0x80000006 0x96d4 2
80.26.5.6       80.26.5.6        984 0x80000005 0x4e99 3

                Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
80.26.2.1       80.26.2.1       1195 0x80000003 0xc85a
80.26.3.4       80.26.3.4       1032 0x80000002 0xa0ee
80.26.4.9       80.26.4.9       1028 0x80000002 0x81fb


R1_ospfd>
```

Rys. 12. *Baza danych stanów łącz – podstawa grafu OSPF – wyświetlona w ruterze R1*

Na rysunku 12 została przedstawiona baza danych protokołu OSPF w ruterze R1. Widoczne są wszystkie rutery od R1 do R4 (wraz z liczbą interfejsów) oraz łącza bezpośrednio przyłączone do rutera R1.

(a)

```
PC1> ping 80.26.5.7
80.26.5.7 icmp_seq=1 timeout
80.26.5.7 icmp_seq=2 timeout
84 bytes from 80.26.5.7 icmp_seq=3 ttl=61 time=39.003 ms
84 bytes from 80.26.5.7 icmp_seq=4 ttl=61 time=40.002 ms
84 bytes from 80.26.5.7 icmp_seq=5 ttl=61 time=32.001 ms

PC1>
```

(b)

```
PC1> trace 80.26.5.7
trace to 80.26.5.7, 8 hops max, press Ctrl+C to stop
 1   80.26.1.1   9.000 ms  10.001 ms  10.001 ms
 2   80.26.2.4   19.001 ms  19.001 ms  19.001 ms
 3   80.26.3.6   29.002 ms  29.001 ms  29.002 ms
 4   *80.26.5.7   39.002 ms (ICMP type:3, code:3, Destination port unreachable)

PC1> trace 80.26.5.7
trace to 80.26.5.7, 8 hops max, press Ctrl+C to stop
 1   80.26.1.1   7.001 ms  9.000 ms  9.001 ms
 2   80.26.2.4   19.001 ms  19.001 ms  20.001 ms
 3   80.26.3.6   29.002 ms  29.002 ms  29.001 ms
 4   *80.26.5.7   39.003 ms (ICMP type:3, code:3, Destination port unreachable)

PC1> trace 80.26.5.7
trace to 80.26.5.7, 8 hops max, press Ctrl+C to stop
 1   80.26.1.1   4.000 ms  9.001 ms  9.000 ms
 2   80.26.2.4   21.001 ms  18.001 ms  19.002 ms
 3   80.26.3.6   29.001 ms  29.002 ms  29.002 ms
 4   *80.26.5.7   39.002 ms (ICMP type:3, code:3, Destination port unreachable)

PC1>
```

(c)

```
PC1> trace 80.26.5.7
trace to 80.26.5.7, 8 hops max, press Ctrl+C to stop
 1   80.26.1.1   7.000 ms  9.000 ms  9.000 ms
 2   80.26.2.4   19.001 ms  19.001 ms  19.001 ms
 3   80.26.4.6   29.001 ms  39.002 ms  29.001 ms
 4   *80.26.5.7   49.003 ms (ICMP type:3, code:3, Destination port unreachable)

PC1> trace 80.26.5.7
trace to 80.26.5.7, 8 hops max, press Ctrl+C to stop
 1   80.26.1.1   8.000 ms  9.001 ms  9.001 ms
 2   80.26.2.9   20.001 ms  19.001 ms  19.001 ms
 3   80.26.3.6   39.002 ms  31.002 ms  29.002 ms
 4   *80.26.5.7   39.002 ms (ICMP type:3, code:3, Destination port unreachable)

PC1> trace 80.26.5.7
trace to 80.26.5.7, 8 hops max, press Ctrl+C to stop
 1   80.26.1.1   10.001 ms  9.000 ms  9.001 ms
 2   80.26.2.4   19.001 ms  19.001 ms  19.001 ms
 3   80.26.4.6   30.002 ms  39.002 ms  29.002 ms
 4   *80.26.5.7   49.003 ms (ICMP type:3, code:3, Destination port unreachable)
```

Rys. 13. *Test trasy z sieci S1 do S5 w sieci z ruterami Cisco: a) polecenie ping, b) polecenie trace bez równoważenia obciążeń c) polecenie trace z równoważeniem obciążeń*

Najprostszy test prawidłowego funkcjonowania rutingu obejmuje sprawdzenie osiągalności wszystkich sieci docelowych. Na rysunku 13 przedstawiono test osiągalności sieci S5 z sieci S1 z wykorzystaniem polecenia `ping` (Rys. 13a) lub polecenia `trace` (Rys. 13b). Oba testy przeprowadzone zostały ze stacji roboczej PC1. Transmisja była realizowana do stacji roboczej PC2. W przypadku polecenia `trace` (Rys. 13b i 13c) widoczna jest pełna trasa, którą pokonują pakiety przesyłane pomiędzy siecią S1 i S5. Po trzykrotnym powtórzeniu polecenia `trace` (Rys. 13b) widoczna jest ciągle ta sama trasa. Wynika to z faktu, że w protokole OSPF standardowo nie jest włączane równoważenie obciążeń, wobec czego w procesie decyzyjnym wykorzystywana jest zawsze jedna, najlepsza ścieżka. W analizowanym przykładzie pomiędzy sieciami S1 i S5 istnieją dwie równoważne trasy. Rutery wybierają pierwszą najlepszą trasę i nie zmieniają jej dopóty, dopóki nie zmieni się topologia sieci lub nie nastąpią awarie.

(a)

```
R1(config)#interface fastEthernet 0/1
R1(config-if)#ip load-sharing per-destination
R1(config-if)#
```

(b)

```
R1(config-if)#ip load-sharing per-packet
R1(config-if)#
```

Rys. 14. *Włączenie równoważenia obciążeń w interfejsie FastEthernet 0/1 rutera R1 a) dla połączenia, b) dla pakietu*

Aby wykorzystać wiele równoważnych ścieżek, w interfejsie, w którym następuje rozgałęzienie trasy, (w przykładzie z Rys. 2a jest to interfejs `FastEthernet 0/1` rutera R1) należy włączyć równoważenie obciążeń (Rys. 14). Opcja ta jest dostępna jedynie w ruterach Cisco (niedostępna w Zebrze). Typowo używane jest takie równoważenie obciążeń, w którym dla danego połączenia wybierana jest zawsze ta sama trasa (Rys. 14a). Wówczas, w ramach jednego połączenia (opisanego przez te same adresy IP, te same porty, ten sam protokół) nie ma równoważenia obciążeń. Równoważenie obciążeń odbywa się dopiero pomiędzy różnymi połączeniami. Jest to rozwiązanie zalecane ze względu na stabilność połączeń TCP (stabilna trasa, stabilne RTT) i stabilność transmisji UDP czasu rzeczywistego (stała trasa daje małą zmienność opóźnień). Gdy chcemy, aby każdy pakiet (niezależnie od połączenia) przechodził inną, równoważną trasą, włączamy równoważenie obciążeń indywidualne dla każdego pakietu (Rys. 14b). W efekcie następuje losowy wybór jednej z równoważnych tras dla każdego pakietu z osobna (Rys. 13c).

PODSUMOWANIE

Efektywne działanie współczesnego systemu informatycznego zależy w dużym stopniu od dobrze funkcjonującej sieci. Istotne jest zatem dobre poznanie rozwiązań wykorzystywanych we współczesnych sieciach i sprawne ich wykorzystanie w praktyce. W artykule przedstawiono wybrane zagadnienia związane z nauczanie protokołu OSPF, który jest podstawowym protokołem rutingu stosowanym w sieciach wielu firm. Zaprezentowane w artykule zagadnienia mogą być realizowane zarówno na ruterach dostępowych firmy Cisco, jak i na ruterach programowych korzystających z pakietu oprogramowania Zebra/Quagga.

BIBLIOGRAFIA

1. Chodorek R.R., Chodorek A., *Wybrane aspekty selekcji tras BGP*. Studia Informatica 2016, nr 2.
2. Chodorek A., Chodorek R., *Możliwości zastosowania emulatora Netkit w badaniach naukowych i dydaktyce*. Logistyka 2014, nr 6.
3. Coltun R., Ferguson D., Moy J., Lindem A., *OSPF for IPv6*. RFC 5340, July 2008.

4. Hawkinson J., Bates T., *Guidelines for creation, selection, and registration of an Autonomous System (AS)*. RFC 1930 (BCP 6), March 1996.
5. Pizzonia M., Rimondini M., *Netkit: network emulation for education*. Software: Practice and Experience, 2014.
6. Moy J., *OSPF Version 2*. RFC 2328, April 1998.
7. Rekhter Y. (Ed.), Li T. (Ed.), Hares S. (Ed.): *A Border Gateway Protocol 4 (BGP-4)*. RFC 4271. January 2006..
8. Wendell O., *CCNP Route: oficjalny przewodnik certyfikacji*. Wydawnictwo Naukowe PWN, 2012.

ON THE USAGE OF THE NETKIT EMULATOR AND ACCESS ROUTERS FOR TEACHING THE OSPF PROTOCOL

Abstract

Internal routing is a key element of IP networking in companies and institutions. It affects the efficiency and the reliability of a network. This paper outlines the teaching of the internal routing based on the OSPF protocol with usage of the access routers and the NetKit computer network emulator.

Autorzy:
dr inż. **Agnieszka Chodorek** – Politechnika Świętokrzyska, Wydział Elektrotechniki, Automatyki i Informatyki, Katedra Systemów Informatycznych; 25-314 Kielce; al. Tysiąclecia Państwa Polskiego 7.
E-mail: a.chodorek@tu.kielce.pl
dr inż. **Robert Chodorek** – AGH Akademia Górniczo-Hutnicza, Wydział Informatyki, Elektroniki i Telekomunikacji, Katedra Telekomunikacji; 30-059 Kraków; Al. A. Mickiewicza 30.
E-mail: chodorek@agh.edu.pl