Agnieszka CHODOREK, Robert CHODOREK

WYKORZYSTANIE RUTERÓW DOSTĘPOWYCH I EMULATORA NETKIT W PROCESIE NAUCZANIA WYBRANYCH PROBLEMÓW RUTINGU ZEWNĘTRZNEGO

Streszczenie

Ruting zewnętrzny stanowi tzw. ruting strategiczny w sieci Internet i sieciach poszczególnych operatorów. Odpowiada on za właściwą dystrybucję ruchu pomiędzy systemami autonomicznymi. W artykule przedstawiono zarys nauczania zagadnień rutingu zewnętrznego w na uczelni technicznej ze szczególnym uwzględnieniem procesu nauczania zagadnień praktycznych w laboratoriach. Do nauczania zagadnień praktycznych zaproponowano wykorzystanie ruterów dostępowych oraz emulatora sieci komputerowych Netkit

WSTĘP

Koncepcja rutingu w sieci Internet opiera się na podziale całej sieci na szereg systemów autonomicznych AS (ang. *Autonomous System*) [3]. System autonomiczny jest to zbiór sieci (lub prefiksów sieci) administrowanych przez jedną firmę lub instytucję. Autonomię systemów należy rozumieć dosłownie, gdyż, ogólnie rzecz biorąc, wewnętrzna organizacja tego zbioru sieci jest dosyć dowolna. Dowolność ta dotyczy nie tylko zastosowanych wewnątrz systemu technik i technologii warstw niższych niż trzecia modelu ISO/OSI, ale także użytych wewnątrz systemu autonomicznego rozwiązań rutingowych.

Ze względu na hierarchię sieci IP, uwzględniającą istnienie systemów autonomicznych jako jednostki nadrzędnej wobec sieci, rozróżnia się dwie klasy rutingu:

- ruting wewnętrzny,
- ruting zewnętrzny.

Ruting wewnętrzny realizowany jest wewnątrz systemu autonomicznego. Wykorzystywane są w tym przypadku takie protokoły, jak RIP, RIPng, OSPFv2, OSPFv3, EIGRP czy IS-IS. Ruting wewnętrzny działa między sieciami identyfikowanymi przez jeden z adresów specjalnych IP, tzw. adres własny sieci.

Ruting zewnętrzny realizowany jest na zewnątrz systemów autonomicznych i współcześnie wykorzystuje tylko jeden protokół – protokół BGP (ang. *Border Gateway Protocol*). Ruting zewnętrzny działa pomiędzy systemami autonomicznymi. Systemy autonomiczne posiadają własne identyfikatory, pełniące funkcję adresu systemu. Obecnie w użyciu są zarówno starsze identyfikatory, 16-bitowe [3], jak i nowsze, wprowadzane od 2007 roku, identyfikatory 32-bitowe [8].

W artykule zaprezentowano wybrane zagadnienia użycia protokołu BGP w systemach autonomicznych, związane z nauczaniem aspektów praktycznych rutingu zewnętrznego. Przedstawione zagadnienia mogą być demonstrowane w różnych środowiskach sieciowych, co daje dużą elastyczność w doborze pomocy dydaktycznych do nauczania rutingu. W artykule omówiono dydaktykę prowadzoną zarówno na ruterach dostępowych firmy Cisco, jak i na ruterach programowych, pracujących pod kontrolą systemu operacyjnego Linux, korzystających z pakietu oprogramowania Zebra/Quagga. Rutery linuksowe pracowały w środowisku maszyn wirtualnych emulatora Netkit [2][4], który jednocześnie emulował niezbędne połączenia sieciowe pomiędzy poszczególnymi ruterami.

Artykuł składa się z pięciu rozdziałów. W rozdziale pierwszym zaprezentowano protokół BGP. Rozdział drugi zawiera opis środowiska testowego wykorzystywanego do realizacji procesu nauczania protokołu BGP. Wybrane zagadnienia praktycznej realizacji sesji BGP na przykładzie ruterów programowych i sprzętowych przedstawiono w rozdziale trzecim.

1. PROTOKÓŁ BGP

Protokół BGP [5] został zdefiniowany jako następca protokołu EGP (ang. *Exterior Gateway Protocol*). Obecnie wykorzystywana jest wersja 4 protokołu BGP. Wersja ta, przy nie zmienionej numeracji wersji, jest znacznie rozszerzona w porównaniu z oryginalną dokumentacją BGP w wersji 4 [6].

Protokół BGP pierwotnie został zdefiniowany dla protokołu IP w wersji 4 (IPv4). Zdefiniowane w latach 90. ubiegłego wieku rozszerzenia, określane jako rozszerzenia wieloprotokołowe (ang. *Multiprotocol Extensions for BGP-4*), pozwalają na wykorzystanie go do współpracy zarówno z protokołem IPv6, jak i z innymi protokołami lub technikami sieciowymi (np. L3VPN) [1]. Rozszerzenia te nie są jednak włączane do podstawowej specyfikacji protokołu BGP, lecz funkcjonują jako samodzielne dokumenty. Pozwala to na szybsze i niezależne rozwijanie mechanizmów protokołu BGP.

Rutery BGP muszą posiadać jawnie wskazane inne rutery, z którymi będą wymieniać komunikaty rutingowe. Rutery takie nazywane są sąsiadami (ang. *neighbor*). Do komunikacji pomiędzy sąsiadami protokół BGP wykorzystuje protokół transportowy TCP, co zapewnia niezawodność dystrybucji komunikatów BGP.

Pomiędzy sąsiadami tworzona jest sesja BGP. Sesje BGP jest utrzymywana poprzez regularne wysyłanie wiadomości typu *keepalive*. W zależności od lokalizacji sąsiada, występują dwa rodzaje sesji BGP [7]:

- sesja zewnętrzna, EBGP (ang. *External BGP*),
- sesja wewnętrzna, IBGP (ang. *Interior BGP*).

Sesja EBGP jest realizowana pomiędzy ruterami brzegowymi dwóch różnych systemów autonomicznych. Domyślnie sesja EBGP jest tworzona pomiędzy ruterami bezpośrednio ze sobą połączonymi (jeśli tak nie jest, należy dodać odpowiednie wpisy do konfiguracji ruterów).

Sesja IBGP jest realizowana pomiędzy ruterami tego samego systemu autonomicznego. Zazwyczaj jest realizowana poprzez szereg ruterów wewnątrz systemu autonomicznego. Sesja IBGP musi być zestawiona pomiędzy wszystkimi ruterami BGP pracują-

cymi w danym systemie autonomicznym. Wymaga to zestawienia wielu połączeń typu każdy-z-każdym. Możliwe jest uproszczenie realizacji tego założenia poprzez zastosowanie dodatkowych mechanizmów, jak np. reflektorów tras.

W protokole BGP informacje o trasach przesyłane są jako łańcuch systemów autonomicznych. W sytuacji ustalonej, przesyłane są tylko informacje o zmianach w trasach, czyli informacje o trasach usuniętych (umieszczane jako pierwsze w komunikacie) i trasach dodanych. Pełna tablica rutingu przesyłana jest w sytuacji, gdy ruter jest uruchamiany. Protokół BGP przechowuje informacje o wszystkich możliwych ścieżkach. Spośród nich wybiera najlepszą.

Każda rozgłaszana trasa posiada szereg atrybutów. Atrybuty podzielone są na dwie kategorie: atrybuty wymagane i atrybuty opcjonalne. Atrybuty wykorzystywane są podczas procesu podejmowania decyzji o wyborze trasy. Służą także do kontroli poprawności działania mechanizmów protokołu BGP, jak i pozwalają na uniknięcie błędnego zadziałania pewnych mechanizmów (np. umożliwiają wykrycie i wyeliminowanie zapętlonej trasy).

Przy wyborze tras w protokole BGP uwzględnianych jest szereg kryteriów. Preferowane są:
- trasy z najwyższą wagą, ustawianą lokalnie w ruterze - atrybut `weight` (atrybut stosowany w ruterach firmy Cisco, dla innych producentów może nie występować),
- trasy z najwyższymi lokalnymi preferencjami (globalnie w danym AS) - atrybut `local preference`,
- trasy o krótszej ścieżce (liście systemów AS w łańcuchu przesyłanym jako atrybut AS_PATH),
- trasy o niższym kodzie pochodzenia trasy (atrybut ORIGIN),
- trasy o mniejszej wartości atrybutu `MED`,
- dla tras EBGP starsze, bardziej stabilne trasy,
- trasy o niższym identyfikatorze rutera `router-id`,
- trasy otrzymane od sąsiada o niższym adresie IP.

Jednym z ważniejszych problemów rutingu zewnętrznego jest skalowalność protokołu BGP. Sesja IBGP wymaga polaczenia typu mesh ze wszystkimi ruterami BGP w danym systemie autonomicznym. Dla `n` ruterów BGP, realizujących sesję IBGP, potrzeba zatem `n(n-1)/2` połączeń. I tak: dla 2 ruterów BGP potrzeba 1 połączenia, 3 rutery BGP dają 3 połączenia, ale 4 rutery BGP to już 6 połączeń, a 6 ruterów BGP to 15 połączeń. Przy 10 ruterach BGP liczba połączeń wzrasta do 45, a 150 ruterów BGP wymaga obsługi aż 11175 połączeń.

Rozwiązaniem problemu z dużą liczbą sesji IBGP może być:
- reflektor tras,
- konfederacja.

Reflektor tras (ang. *Route Reflector*, RR) jest to ruter IBGP "odbijający" trasy pozyskane od jednego rutera IBGP do innych ruterów BGP. Konfederacje zaś dzielą duży system autonomiczny na wewnętrzne systemy autonomiczne, wykorzystujące prywatną adresację systemów autonomicznych. Konfederacje dodają dodatkowe informacje do atrybutu AS_PATH.

Ze względu na sposób połączenia, systemy autonomiczne dzielimy na:
- proste (jednopunktowe nietranzytowe),
- wielokrotne (wielopunktowe nietranzytowe),
- tranzytowe (wielopunktowe tranzytowe).

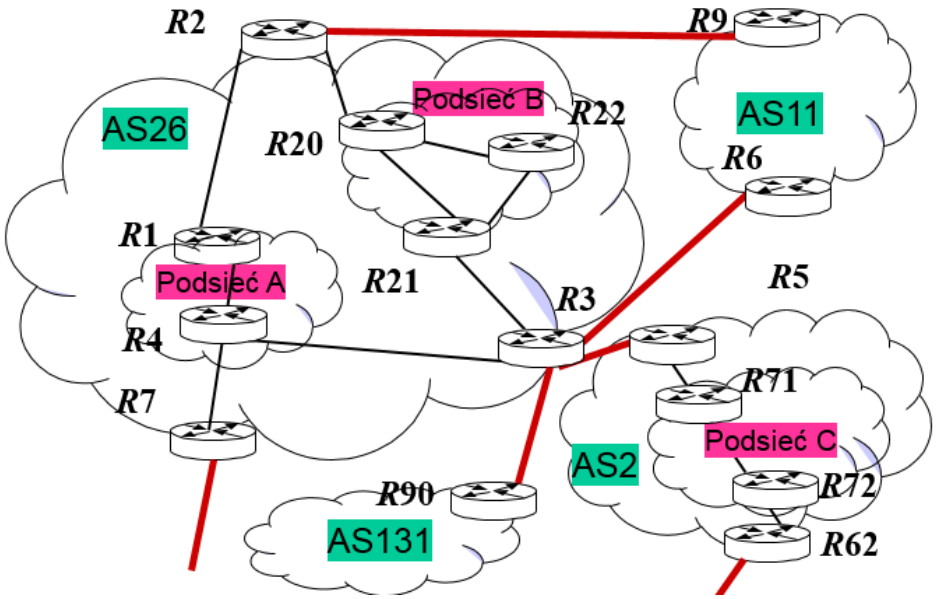
System prosty posiada tylko jedno łącze do Internetu. Systemy wielokrotne i tranzytowe są przyłączone do Internetu w wielu (przynajmniej dwóch) punktach - mają przynajmniej po dwa łącza do Internetu. Systemy jednokrotne i wielokrotne to systemy nietranzytowe. W systemie nietranzytowym dane przepływają do i z systemu. W takim systemie autonomicznym nie ma ruchu obcego, co jest charakterystyczne dla użytkowników końcowych.

W systemie autonomicznym wielopunktowym tranzytowym, dane przepływają zarówno do jaki i z systemu oraz, dodatkowo, w systemie notowany jest przepływ ruchu obcego. Obcy ruch jest ruchem tranzytowym (stąd nazwa systemu), przenoszonym do innych systemów autonomicznych. Obecność ruchu obcego jest charakterystyczna dla sieci operatorów.

Przykładowy podział sieci IP na systemy autonomiczne został zobrazowany na rysunku 1. Widoczny na rysunku system AS 131 jest przykładem systemu prostego. System AS 11 to, z kolei, przykład systemu wielokrotnego. Systemy AS 2 i AS 26 są systemami tranzytowymi.

2. ŚRODOWISKO TESTOWE

W artykule wykorzystywane są dwa środowiska testowe: sprzętowe, oparte na ruterach CISCO oraz programowe, oparte na oprogramowaniu narzędziowym Zebra/Quagga i emulatorze NetKit. Emulator NetKit [2] tworzy sieć wirtualną (domenę kolizyjną) łączącą maszyny wirtualne pracujące pod kontrolą systemu operacyjnego Linux. Maszyny wirtualne uruchamiane są w trybie UML (ang. *User*



Rys. 1. *Podział sieci Internet na systemy autonomiczne*

Mode Linux), co daje wirtualizację typu drugiego (system operacyjny gościa korzysta z zasobów sprzętowych i programowych swojego gospodarza). To, z kolei, przekłada się na niewielkie wymagania sprzętowo-programowe emulatora. Efektywność wirtualizacji odbywa się kosztem elastyczności emulacji, gdyż program Netkit może łączyć jedynie maszyny Linuxowe, przy czym wersja systemu operacyjnego Linux maszyny-gościa musi być identyczna, jak wersja systemu gospodarza.

Sieć testowa (rys. 2) składa się z trzech systemów autonomicznych AS 12, AS 56 i AS 18, o identyfikatorach, odpowiednio, 12, 56 i 18. System autonomiczny AS 12 rozgłasza cztery sieci: S1, S2, S3 i S4. System autonomiczny AS 56 rozgłasza jedynie sieć S6. System autonomiczny AS 18 rozgłasza dwie sieci: S8 i S9. Pozostałe dwie sieci, S5 i S7, nie są rozgłaszane. Przykładowe parametry konfiguracyjne sieci pokazanej na rysunku 2, w tym adresy własne sieci S1...S9 oraz maski sieci S1...S9, zostały zamieszczone w tabeli 1.

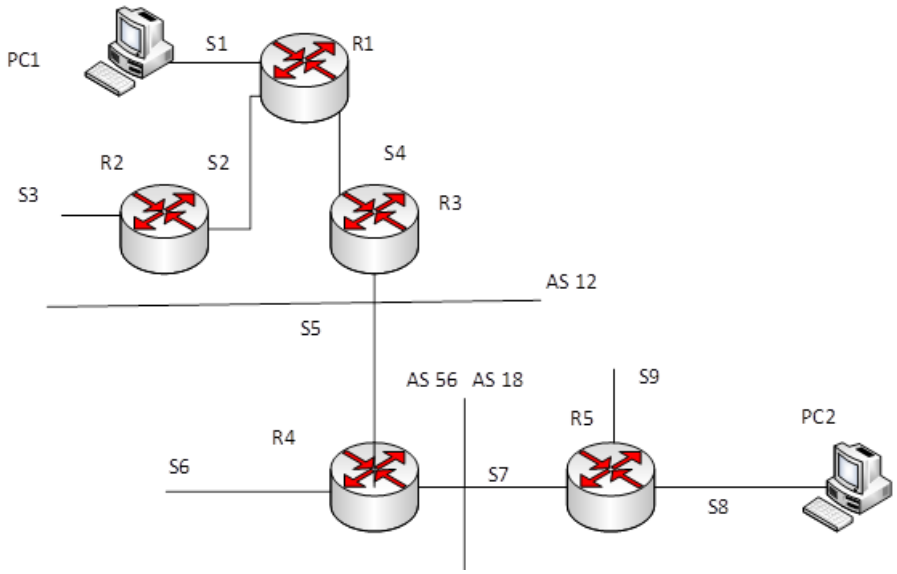
W systemie autonomicznym AS1, w sieci S1, zlokalizowany został komputer o identyfikatorze PC1. Adres IP komputera PC1 to 80.26.1.5. Drugi komputer, PC2, o adresie IP 80.26.33.7, jest podłączony do sieci S8. Rutery R1, R2 i R3 znajdują się w systemie autonomicznym AS 12, ruter R4 jest w systemie AS 56, zaś ruter R5 w systemie AS 18. Przykładowe adresy IP poszczególnych interfejsów ruterów R1...R5 zostały zamieszczone w tabeli 2.

Widoczny na rysunku 2 system autonomiczny AS 56 jest systemem wielokrotnym i tranzytowym. Przenosi on, oprócz ruchu własnego, również ruch z systemów autonomicznych AS 12 i AS 18. Systemy autonomiczne AS 12 i AS 18 są systemami prostymi.

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

Rys. 3. *Uruchamianie modułów Zebry i protokołów rutingu*

Ucząc się aspektów praktycznych rutingu zewnętrznego, studenci mają, między innymi, nabrać umiejętności konfigurowania ruterów BGP. W tym celu wykonują ćwiczenia laboratoryjne, samo-



Rys. 2. *Przykładowe środowisko testowe*

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

identyfikator sieci	adres własny sieci	maska sieci	rozgłaszana przez system autonomiczny	rozgłaszana przez ruter
S1	80.26.1.0	255.255.255.0	AS 12	R1
S2	80.26.2.0	255.255.255.0	AS 12	R1, R2
S3	80.26.3.0	255.255.255.0	AS 12	R2
S4	80.26.4.0	255.255.255.0	AS 12	R1, R3
S5	80.26.9.0	255.255.255.0	nie rozgłaszana	nie rozgłaszana
S6	80.26.19.0	255.255.255.0	AS 56	R4
S7	80.26.10.0	255.255.255.0	nie rozgłaszana	nie rozgłaszana
S8	80.26.33.0	255.255.255.0	AS 18	R5
S9	80.26.34.0	255.255.255.0	AS 18	R5

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	80.26.4.1
R2	80.26.2.4	80.26.3.4	-
R3	80.26.4.9	80.26.9.9	-
R4	80.26.9.3	80.26.10.3	80.26.19.3
R5	80.26.10.15	80.26.33.15	80.26.34.15

dzielnie zestawiając sieci testowe, konfigurując urządzenia i systemy autonomiczne.

W przypadku użycia ruterów linuksowych i emulatora NetKit, należy zestawić w emulatorze sieć o podanej konfiguracji, po czym uruchomić pakiet oprogramowania Zebra/Quagga, udostępniającego szereg protokołów rutingu (zarówno rutingu wewnętrznego, jak i protokół BGP). W tym celu, w katalogu `etc/quagga` każdego rutera należy zamieścić pliki startowe Zebry i wybranych protokołów rutingu, a następnie uruchomić moduł Zebry i obsługę niezbędnych protokołów rutingu (w trybie demona, opcja `-d`). W przykładzie pokazanym na rysunku 2, w każdym z ruterów uruchamiany jest demon modułu Zebra i demon procesu rutingu BGP. W ruterach R1, R2 i R3 uruchamiany jest dodatkowo protokół RIP (rys. 3).

Realizacja zadania na ruterach Cisco wymaga przygotowania topologii i skonfigurowania ruterów. W ruterach Cisco, wykorzystywanych na laboratorium z przedmiotu "Zaawansowane metody rutingu zewnętrznego", prowadzonego na krakowskiej Akademii Górniczo-Hutniczej, studenci AGH mają do dyspozycji dwa typy interfejsów: standardu Ethernet (Fast Ethernet) i łącza szeregowe synchroniczne. Używane łącze standardu Fast Ethernet może pracować z przepustowością 100 Mb/s lub 10 Mb/s. Łącze szeregowe pracuje z maksymalną przepustowością 8 Mb/s.

Zestawiając fizycznie sieć z wykorzystaniem ruterów dostępowych Cisco, w sieciach, które nie mają podłączenia do innego komputera lub rutera należy zaemulować takie połączenie za pomocą interfejsu *loopback*. Liczba takich interfejsów w ruterach Cisco jest bardzo duża (np. można uruchomić ich 1000) i nie stanowi ona przeszkody w realizacji zadania. Dla interfejsu należy przypisać adres interfejsu (zgodny z adresami podanymi w tabeli 2) tak, jak dla każdego innego interfejsu.

3. KONFIGUROWANIE RUTINGU BGP - STUDIUM PRZYPADKU

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

3.1. Konfigurowanie protokołu BGP

Aby uruchomić ruting BGP należy dokonać szeregu czynności konfiguracyjnych. Te same czynności wykonywane są zarówno dla rutera realizującego sesję EBGP, jak i IBGP. W przypadku rutera programowego, korzystającego z oprogramowania Zebra/Quagga, konfigurując ruter należy skorzystać z usługi zdalnego terminala.

(a)

```
r1:~# telnet localhost bgpd
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_bgpd>
```

(b)

```
R1_bgpd> enable
Password:
R1_bgpd# configure terminal
R1_bgpd(config)#
```

Rys. 4. *Czynności przygotowawcze: a) łączenie się z demonem rutingu BGP, b) wynik polecenia* `configure terminal`

Łącząc się z modułem Zebra za pomocą usługi telnet (rys. 4a) należy zalogować się do konsoli demona rutingu BGP (polecenie: `telnet localhost bgpd`), podając hasło zapisane w pliku konfiguracyjnym demona. Następnie (rys. 4b) należy przejść do uprzywilejowanego trybu wykonawczego (polecenie: `enable`) korzystając z hasła zapisanego w konfiguracji. Kolejnym krokiem jest przejście do trybu konfiguracji globalnej (polecenie: `configure terminal`).

(a)

```
R1_bgpd(config)# router bgp 12
R1_bgpd(config-router)#
```

(b)

```
R1_bgpd(config-router)# neighbor 80.26.2.4 remote-as 12
R1_bgpd(config-router)# neighbor 80.26.4.9 remote-as 12
```

(c)

```
R1_bgpd(config-router)# network 80.26.1.0/24
R1_bgpd(config-router)# network 80.26.2.0/24
R1_bgpd(config-router)# network 80.26.4.0/24
```

Rys. 5. *Konfigurowanie protokołu BGP: a) wynik polecenia* `router bgp`*, b) wynik polecenia* `neighbor`*, c) wynik polecenia* `network`

Na rysunku 5 zaprezentowano przykład konfigurowania protokołu BGP dla rutera R1. Aby skonfigurować protokół BGP, w kolejnych krokach należy:

- podać numer systemu autonomicznego,
- skonfigurować sąsiadów,
- wskazać sieci rozgłaszane przez konfigurowany ruter.

Podawanie numeru systemu autonomicznego przedstawiono na rysunku 5a (polecenie `router bgp 12`). Numer systemu autonomicznego to 12.

Konfigurowanie sąsiadów rutera R1 (rys. 5b) realizowane jest za pomocą polecenia `neighbor`. Jako sąsiedzi wskazani zostali: interfejs eth0 rutera R2 o adresie `80.26.2.4` (polecenie: `neighbor 80.26.2.4 remote-as 12`) oraz interfejs eth0 rutera R4 o adresie `80.26.4.9` (polecenie: `neighbor 80.26.4.9 remote-as 12`).

Wskazywanie sieci rozgłaszanych przez ruter R1 zobrazowano na rysunku 5c. Realizowane jest ono za pomocą polecenia `network`, którego parametrem jest adres własny sieci rozgłaszanej i maska sieci rozgłaszanej (podana w formacie skróconym). Jak widać na rysunku, ruter R1 rozgłasza sieć S1 (polecenie: `network 80.26.1.0/24`), sieć S2 (polecenie: `network 80.26.2.0/24`) i sieć S4 (polecenie: `network 80.26.4.0/24`).

```
R1>
R1>enable
R1#configure terminal
Enter configuration commands, one per line.  End with C
R1(config)#router bgp 12
R1(config-router)#neighbor 80.26.2.4 remote-as 12
R1(config-router)#neighbor 80.26.4.9 remote-as 12
R1(config-router)#
R1(config-router)#network 80.26.1.0 mask 255.255.255.0
R1(config-router)#network 80.26.2.0 mask 255.255.255.0
R1(config-router)#network 80.26.4.0 mask 255.255.255.0
R1(config-router)#
R1(config-router)#exit
R1(config)#
```

Rys. 6. *Konfigurowanie protokołu BGP w ruterze R1 - ruter dostępowy firmy Cisco.*

W przypadku ruterów dostępowych firmy Cisco, konfiguracja rutera i protokołów rutingu odbywa się z jednej konsoli tekstowej, dołączonej do rutera. Podstawowe komendy są identyczne, jak w module Zebra, choć mogą różnić się nieco składnią (rys. 6). Z trybu konfiguracji można od razu wejść do konfiguracji interfejsów (w module Zebra należało skorzystać z terminala związanego z demonem zebra) lub do konfiguracji któregoś z protokołów rutingu (w module Zebra są to osobne okna terminali).

Przykład konfigurowania protokołu BGP dla rutera R1 pokazano na rysunku 6. Rysunek ten odpowiada funkcjonalnie rysunkom 4 i 5. Brak zabezpieczenia hasłem dostępu do rutera wynika ze specyfiki tego dostępu. W przypadku linuksowego, programowego rutera, dostęp zapewniała usługa zdalnego terminala (telnet), standardowo korzystająca z ochrony hasłem. Rutery sprzętowe zarządzane są lokalnie poprzez fizycznie dołączoną konsolę, komunikującą się z ruterem przez łącze szeregowe. Nie ma tu specjalnego zagrożenia, a rutery wykorzystywane są tylko do celów dydaktycznych i, dla wygody, nie mają ustawionych haseł dostępu na żaden poziom zarządzania ruterem.

Po połączeniu poprzez łącze szeregowe zostaje udostępniony podstawowy, nieuprzywilejowany dostęp do rutera (znak zachęty ">"). Polecenie `enable` powoduje przejście do trybu uprzywilejowanego, a polecenie `configure terminal` przejście do trybu konfiguracji globalnej. W trybie tym zostaje ustawiony numer systemu autonomicznego oraz wskazani sąsiedzi rutera R1. Polecenia network na rysunkach 5 i 6 różnią się składnią. Ruter linuksowy posługiwał się maską w formacie skróconym, ruter sprzętowy wymaga maski w formacie dziesiętnym, poprzedzonej słowem kluczowym `mask`.

(a)

```
R2_bgpd# configure t
R2_bgpd(config)# router bgp 12
R2_bgpd(config-router)# neighbor 80.26.2.1 remote-as 12
R2_bgpd(config-router)# neighbor 80.26.4.9 remote-as 12
R2_bgpd(config-router)#
R2_bgpd(config-router)# network 80.26.2.0/24
R2_bgpd(config-router)# network 80.26.3.0/24
R2_bgpd(config-router)#
R2_bgpd(config-router)#
```

(b)

```
R3_bgpd# configure t
R3_bgpd(config)# router bgp 12
R3_bgpd(config-router)# neighbor 80.26.2.4 remote-as 12
R3_bgpd(config-router)# neighbor 80.26.4.1 remote-as 12
R3_bgpd(config-router)# neighbor 80.26.9.3 remote-as 56
R3_bgpd(config-router)#
R3_bgpd(config-router)# network 80.26.4.0/24
R3_bgpd(config-router)#
R3_bgpd(config-router)#
```

(c)

```
R4_bgpd# configure t
R4_bgpd(config)# router bgp 56
R4_bgpd(config-router)# neighbor 80.26.9.9 remote-as 12
R4_bgpd(config-router)# neighbor 80.26.10.15 remote-as 18
R4_bgpd(config-router)#
R4_bgpd(config-router)# network 80.26.19.0/24
R4_bgpd(config-router)#
R4_bgpd(config-router)#
```

(d)

```
R5_bgpd# configure terminal
R5_bgpd(config)# router bgp 18
R5_bgpd(config-router)# neighbor 80.26.10.3 remote-as 56
R5_bgpd(config-router)#
R5_bgpd(config-router)# network 80.26.33.0/24
R5_bgpd(config-router)# network 80.26.34.0/24
R5_bgpd(config-router)#
```

Rys. 7. *Konfigurowanie protokołu BGP w: a) ruterze R2, b) ruterze R3, c) ruterze R4, d) ruterze R5.*

Konfigurację protokołu BGP należy przeprowadzić dla wszystkich pozostałych ruterów sieci testowej (rys. 5, rys. 7). Wykonując czynności konfiguracyjne, niektóre polecenia można wydawać w postaci pełnej lub skróconej (polecenia: `configure terminal` i `configure t` na rysunku 7).

Rutery R2 i R3 należą do systemu autonomicznego AS 12 - tego samego, co ruter R1. Ruter R2 (rys. 7a) sąsiaduje z interfejsem eth1 rutera R1 o adresie `80.26.2.1` i interfejsem eth0 rutera R3 o adresie `80.26.4.9`. Obaj sąsiedzi rutera R2 należą do systemu autonomicznego AS 12 i ruter zestawia z nimi sesje IBGP. Ruter R3 (rys. 7b) sąsiaduje z interfejsem eth2 rutera R1 o adresie `80.26.4.1` i interfejsem eth0 rutera R4 o adresie `80.26.9.3`. Ruter R1 należy do AS 12 (sesja IBGP), natomiast ruter R4 do AS 56 (sesja EBGP). Ponieważ sesja IBGP tworzona jest na zasadzie "każdy z każdym", w konfiguracji rutera R3 musi znaleźć się jeszcze wpis dla sesji IBGP z ruterem R2. Mimo, iż R3 nie sąsiaduje fizycznie z R2, z punktu widzenia sesji IBGP rutery te są sąsiadami (polecenie `neighbor 80.26.2.4 remote-as 12` na rysunku 7).

Sąsiadami rutera R4 (rys. 7c) z systemu autonomicznego AS 56 są ruter R3 należący do AS 12 (interfejs eth1 o adresie `80.26.9.9`) oraz ruter R5 należący do AS 18 (interfejs eth0 o adresie `80.26.10.15`). Ruter R4 tworzy zatem dwie sesje EBGP (i ani jednej sesji IBGP). Ruter R5 (rys. 7d) ma tylko jednego sąsiada, czyli R4 (interfejs eth1 o adresie IP `80.26.10.3`).

Ruter R2 (rys. 7a) rozgłasza sieci S2 (`80.26.2.0/24`) i S3 (`80.26.3.0/24`). Ruter R3 (rys. 7b) rozgłasza tylko sieć S4 (`80.26.4.0/24`), a ruter R4 (rys. 7c) tylko sieć S6 (`80.26.19.0`). Ruter R5 (rys. 7d), podobnie jak ruter R2, rozgłasza dwie sieci. Są to: sieć S8 (`80.26.33.0/24`) i sieć S9 (`80.26.34.0/24`).

```
pc2:~# tracert 80.26.1.5
traceroute to 80.26.1.5 (80.26.1.5), 30 hops max, 40 byte packets
 1  (80.26.33.15)  0.227 ms  0.063 ms  0.057 ms
 2  (80.26.10.3)  0.225 ms  0.114 ms  0.107 ms
 3  (80.26.9.9)  0.296 ms  0.170 ms  0.247 ms
 4  (80.26.4.1)  0.363 ms  0.216 ms  1.465 ms
 5  (80.26.1.5)  0.321 ms  0.261 ms  0.317 ms
pc2:~#
```

Rys. 8. *Wynik działania programu traceroute*

Poprawność działania rutingu (a zatem i poprawność konfiguracji) można sprawdzić m.in. za pomocą programu traceroute (polecenie: `tracert`). Na rysunku 8 został przedstawiony wynik działania traceroute dla sieci o topologii pokazanej na rysunku 2. Kontrola była przeprowadzona z komputera PC2, stacją docelową był komputer PC1. Komunikaty traceroute przechodziły, kolejno, przez interfejs eth1 rutera R5 (`80.26.33.15`), interfejs eth1 rutera R4 (`80.26.10.3`), interfejs eth1 rutera R3 (`80.26.9.9`) i interfejs eth2 rutera R1 (`80.26.4.1`). Ostatnim osiągniętym węzłem był węzeł docelowy PC1 (`80.26.1.5`).

3.2. Analiza tablic rutingu

Oprogramowanie Zebra/Quagga pozwala na obserwację tablic rutingu na trzech poziomach:

- poziomie jądra systemu operacyjnego Linux,
- poziomie modułu Zebra,
- poziomie protokołu rutingu (tu: poziomie BGP).

Tablica rutingu jądra systemu operacyjnego (ang. *Forwarding Information Base, FIB*) jest tablicą decyzyjną. Na jej podstawie jądro podejmuje decyzje o rutingu datagramów IP. Tablica FIB zawiera tylko jedną, optymalną (pod kątem ustalonej funkcji celu) trasę

prowadzącą do danej sieci docelowej. Pozostałe tablice mają charakter pomocniczy i zawierają bardziej szczegółowe informacje o trasach oraz informacje o trasach alternatywnych. Tablica rutingu modułu Zebra (ang. *Routing Information Base, RIB*) zawiera trasy zebrane od różnych protokołów rutingu. Protokoły rutingu utrzymują własne tablice rutingu (ang. L*ocal Routing Information Base, Loc-RIB*), z których wpisy są eksportowane do tablicy RIB.

(a)

```
r1:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
80.26.34.0      80.26.4.9       255.255.255.0   UG    0      0        0 eth2
80.26.19.0      80.26.4.9       255.255.255.0   UG    0      0        0 eth2
80.26.3.0       80.26.2.4       255.255.255.0   UG    2      0        0 eth1
80.26.2.0       *               255.255.255.0   U     0      0        0 eth1
80.26.1.0       *               255.255.255.0   U     0      0        0 eth0
80.26.33.0      80.26.4.9       255.255.255.0   UG    0      0        0 eth2
80.26.4.0       *               255.255.255.0   U     0      0        0 eth2
80.26.9.0       80.26.4.9       255.255.255.0   UG    2      0        0 eth2
r1:~#
```

(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
B   80.26.3.0/24 [200/0] via 80.26.2.4, eth1, 04:26:54
R>* 80.26.3.0/24 [120/2] via 80.26.2.4, eth1, 04:34:14
C>* 80.26.4.0/24 is directly connected, eth2
R>* 80.26.9.0/24 [120/2] via 80.26.4.9, eth2, 00:01:25
B>* 80.26.19.0/24 [200/0] via 80.26.9.3 (recursive via 80.26.4.9), 00:00:28
B>* 80.26.33.0/24 [200/0] via 80.26.9.3 (recursive via 80.26.4.9), 00:00:28
B>* 80.26.34.0/24 [200/0] via 80.26.9.3 (recursive via 80.26.4.9), 00:00:28
C>* 127.0.0.0/8 is directly connected, lo
R1#
```

(c)

```
R1_bgpd(config-router)# end
R1_bgpd# show ip bgp
BGP table version is 0, local router ID is 80.26.4.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 80.26.1.0/24     0.0.0.0                  0         32768 i
* i80.26.2.0/24     80.26.2.4                0    100      0 i
*>                  0.0.0.0                  0         32768 i
*>i80.26.3.0/24     80.26.2.4                0    100      0 i
* i80.26.4.0/24     80.26.4.9                0    100      0 i
*>                  0.0.0.0                  0         32768 i
* i80.26.19.0/24    80.26.9.3                0    100      0 56 i
* i80.26.33.0/24    80.26.9.3                     100      0 56 18 i
* i80.26.34.0/24    80.26.9.3                     100      0 56 18 i

Total number of prefixes 7
```

(d)

```
     80.0.0.0/24 is subnetted, 7 subnets
C       80.26.2.0 is directly connected, FastEthernet0/1
B       80.26.3.0 [200/0] via 80.26.2.4, 00:37:11
C       80.26.1.0 is directly connected, FastEthernet0/0
C       80.26.4.0 is directly connected, FastEthernet1/0
B       80.26.19.0 [200/0] via 80.26.4.9, 00:00:38
B       80.26.34.0 [200/0] via 80.26.4.9, 00:00:38
B       80.26.33.0 [200/0] via 80.26.4.9, 00:00:38
```

(e)

```
R1#show ip bgp
BGP table version is 10, local router ID is 80.26.4.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 80.26.1.0/24     0.0.0.0                  0         32768 i
* i80.26.2.0/24     80.26.2.4                0    100      0 i
*>                  0.0.0.0                  0         32768 i
*>i80.26.3.0/24     80.26.2.4                0    100      0 i
* i80.26.4.0/24     80.26.4.9                0    100      0 i
*>                  0.0.0.0                  0         32768 i
*>i80.26.19.0/24    80.26.4.9                0    100      0 56 i
*>i80.26.33.0/24    80.26.4.9                0    100      0 56 18 i
*>i80.26.34.0/24    80.26.4.9                0    100      0 56 18 i
```

Rys. 9. *Tablica rutingu: a) FIB, b, d) RIB, c, e) Loc-RIB rutera R1 zrealizowanego: a, b, c) programowo, d, e) sprzętowo*

Struktura tablicy RIB modułu Zebra (rys. 9b) jest identyczna, jak tablicy rutingu rutera Cisco (rys. 9d). Podobnie Loc-RIB protokołu BGP funkcjonuje zarówno w Zebrze (rys. 9c), jak i w Cisco (rys. 9e). Tablica FIB w ruterach Cisco również występuje, aczkolwiek może być ona scentralizowana lub (w gigabitowych ruterach) rozproszona na poszczególne karty liniowe (sieciowe) rutera. FIB pełni rolę także swoistej pamięci podręcznej dla tablicy rutingu RIB, pozwalającej podejmować szybkie, gotowe decyzje.

Przykładowa tablica rutingu, widziana z poziomu systemu operacyjnego Linux, została zaprezentowana na rysunku 9a. Tablica ta została wyświetlona w wyniku wykonania polecenia `route` przez ruter R1. Tablica ta zawiera osiem wpisów, z których trzy dotyczą tras do sieci miejscowych rutera R1:

– S2, o adresie `80.26.2.0`, dołączona do interfejsu eth1,
– S1, o adresie `80.26.1.0`, dołączona do interfejsu eth0,
– S4, o adresie `80.26.4.0`, dołączona do interfejsu eth2.

Sieci te są bezpośrednio dołączone do rutera R1, zatem druga kolumna nie zawiera informacji o adresie następnego rutera na trasie (adres interfejsu rutera zastąpiony został znakiem "`*`"), a metryka jest równa 0 (na poziomie tablicy FIB - tablice Loc-RIB mogą stanowić inaczej). Znaczniki trasy wskazują, że są to trasy aktywne (znacznik `U` - z ang. *route is Up*).

Dalsze pięć wpisów do tablicy FIB rutera R1 wskazuje trasy do sieci nie podłączonych bezpośrednio do tego rutera. Cztery z tras prowadzą przez ruter R3:

– trasa do sieci S9 o adresie własnym `80.26.34.0`,
– trasa do sieci S6 o adresie własnym `80.26.19.0`,
– trasa do sieci S8 o adresie własnym `80.26.33.0`,
– trasa do sieci S5 o adresie własnym `80.26.9.0`.

Pakiety adresowane do tych sieci będą wysyłane przez interfejs eth2 rutera R1 do interfejsu eth0 rutera R3 (adres IP: `80.26.4.9`). Znaczniki wskazują, że wszystkie te trasy są trasami aktywnymi (znacznik `U`), prowadzącymi przez inny ruter (znacznik `G` - z ang. *Gateway*, dosł. brama). Metryki tras prowadzących do S6, S8 i S9 są zerowe, metryka trasy do sieci S5 wynosi 2.

Jedna trasa, czyli:

– trasa do sieci S3 (adres własny: `80.26.3.0`),

prowadzi przez ruter R2 (adres IP `80.26.2.4` to adres interfejsu eth0 tego rutera). Jest to trasa aktywna (znacznik `U`), przechodząca przez inny ruter niż R1 (znacznik `G`), a jej metryka jest równa 2.

Tablica rutingu FIB nie zawiera wpisu o trasie do sieci S7, która jest siecią nierozgłaszaną przez BGP, choć zawiera wpis o trasie do sieci S5, również nierozgłaszanej. Można to wyjaśnić na podstawie analizy tablic RIB. Zawartość tablicy RIB jest wyświetlana po wydaniu polecenia `show ip route` (rys. 9b,d).

Na rysunkach 9b,d widoczne są trzy (rys. 9d) lub cztery (rys. 9b) wpisy o trasach do sieci miejscowych (znacznik źródła wpisu: C). Trzy wpisy, obecne na obu rysunkach, dotyczą tras do sieci S1, S2 i S4, wymienionych w tablicy FIB. Czwarta trasa, widoczna tylko na rysunku tablicy RIB modułu Zebra, to trasa do sieci wirtualnej (adres własny i maska sieci: `127.0.0.0/8`), utworzonej na bazie adresu specjalnego IP (adresu zwrotnego rutera R1).

Trasy do sieci miejscowych zostały opatrzone napisem w języku angielskim, informującym, że dana sieć jest bezpośrednio dołączona (tu: do rutera R1) wraz z podaniem, do którego jest dołączona interfejsu. Skrót "`lo`" (rys. 9b) oznacza interfejs zwrotny *loopback*. Trasy te są ważne bezterminowo.

Na rysunkach 9b,d widoczne są trzy pozyskane za pośrednictwem rutera R3 (adres IP `80.26.4.9` to adres interfejsu eth0 rutera R3) trasy pozyskane za pomocą protokołu BGP (znacznik źródła B) za pośrednictwem rutera R3 (adres IP `80.26.4.9` to adres interfejsu eth0 rutera R3). Są to trasy do sieci S6 (adres własny: `80.26.19.0`), sieciS8 (adres własny: `80.26.33.0`) i sieci S9 (adres własny: `80.26.34.0`). Tablica RIB modułu

Zebra/Quagga (rys. 9b) uściśla, że trasy te pozyskano od rutera R4 (adres IP interfejsu: 80.26.9.3) będącego ruterem brzegowym systemu autonomicznego AS56, sąsiadującego z systemem AS12. Ostatnim polem jest pole czasu, wskazujące, kiedy dany wpis pojawił się w tablicy rutingu RIB.

Jak widać na rysunkach 9b i 9d, ani trasa do sieci S5, ani trasa do sieci S7 nie są rozgłaszane przez protokół BGP. Żadna z tych tras nie jest widoczna na rysunku 9d, obrazującym sytuację, gdy w systemie autonomicznym nie działa żaden inny ruting poza rutingiem BGP. Na rysunku 9b, obrazującym sytuację, gdy w systemie autonomicznym oprócz rutingu BGP działa jeszcze ruting wewnętrzny (protokół RIP) nie ma wpisu o trasie do sieci S7, ale istnieje wpis o trasie do sieci S5 (o adresie własnym 80.26.9.0). Trasa ta dotarła do rutera R1 za pośrednictwem protokołu RIP (znacznik źródła R). Wprawdzie sieć S5 nie należy do systemu autonomicznego AS12, ale jest siecią miejscową rutera brzegowego systemu AS12. Protokół rutingu wewnętrznego RIP, operujący wewnątrz systemu autonomicznego AS12, przekazał tę informację jako informację o sieci bezpośrednio podłączonej do rutera R3.

```
R3# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

B   80.26.1.0/24 [200/0] via 80.26.4.1, eth0, 06:11:03
R>* 80.26.1.0/24 [120/2] via 80.26.4.1, eth0, 06:30:05
B   80.26.2.0/24 [200/0] via 80.26.4.1, eth0, 06:11:03
R>* 80.26.2.0/24 [120/2] via 80.26.4.1, eth0, 06:30:05
B   80.26.3.0/24 [200/0] via 80.26.2.4, 06:09:37
R>* 80.26.3.0/24 [120/3] via 80.26.4.1, eth0, 06:30:05
C>* 80.26.4.0/24 is directly connected, eth0
C>* 80.26.9.0/24 is directly connected, eth1
B>* 80.26.19.0/24 [20/0] via 80.26.9.3, eth1, 03:53:04
B>* 80.26.33.0/24 [20/0] via 80.26.9.3, eth1, 03:51:03
B>* 80.26.34.0/24 [20/0] via 80.26.9.3, eth1, 03:51:03
C>* 127.0.0.0/8 is directly connected, lo
R3#
```

Rys. 10. *Tablica rutingu rutera R3 widziana z poziomu modułu Zebra*

Tab. 3. *Wartości wskaźnika DAD dla różnych typów tras*

Wskaźnik DAD	Opis
0	bezpośrednio dołączona sieć
1	trasa statyczna
20	trasa pozyskana z EBGP
110	trasa pozyskana z OSPF
115	trasa pozyskana z ISIS
120	trasa pozyskana z RIP
200	trasa pozyskana z IBGP

Protokół BGP, w przeciwieństwie do protokołów rutingu wewnętrznego, nie podaje metryki. Dlatego kolumna metryki w tablicy rutingu FIB dla wpisów pochodzących z BGP przyjmuje wartość zero (rys. 9a). Ponieważ różne protokoły rutingu mogą korzystać (i często korzystają) z różnych metryk, do porównywania kosztów tras pochodzących z danej trasy używa się wskaźnika DAD (ang. *Default Administrative Distances*, dosł. domyślna odległość administracyjna, przydzielana administracyjnie).

Wskaźnik DAD wskazuje na hierarchię wpisów pochodzących od różnych protokołów rutingu i ma sens zryczałtowanego kosztu trasy (tab. 3). Im mniejsza wartość DAD, tym trasa jest lepsza. Jeżeli w tablicy RIB znajdują się dwie i więcej informacji o trasach, pochodzących od różnych protokołów, do tablicy FIB przekazana zostanie ta trasa, która ma najmniejszy DAD.

Tablica RIB zobrazowana na rysunku 9b posiada dwa wpisy dotyczące tras do sieci S3 (80.26.3.0/24). Obie kierują datagramy IP przez interfejs eth1 rutera R1 na interfejs eth0 rutera R2 (80.26.2.4). Pierwsza z nich jest trasą pozyskaną za pośrednictwem protokołu BGP i jej wartość wskaźnika DAD wynosi 200, metryki nie podano (wpis [200/0]). Druga trasa pozyskana została przez protokół RIP i jej wartość DAD wynosi 120, a metryka 2 (wpis [120/2]). Na podstawie kryterium DAD do użytku wybrana została (znak ">") druga w kolejności wpisów trasa, czyli trasa pozyskana przez RIP. Trasa ta została również przekazana z tablicy RIB do tablicy FIB (ustawiony znacznik FIB, znak "*").

Typowo trasy pozyskane przez protokół rutingu wewnętrznego mają pierwszeństwo przed trasami pozyskanymi z sesji IBGP, podczas gdy trasy pozyskane z sesji EBGP mają pierwszeństwo przed trasami przekazanymi przez protokoły rutingu wewnętrznego. W tablicy rutingu modułu Zebra trasy pochodzące z sesji IBGP i EBGP nie są rozróżniane na poziomie znacznika źródła (oznaczane są tym samym znacznikiem B), choć różni je inna wartość DAD (rys.10).

W tablicach rutingu RIB przedstawionych na rysunkach 9b i 9d informacja o wskaźniku DAD i metryce trasy (w formacie: DAD/metryka) zamieszczona jest w nawiasach kwadratowych przy każdej trasie pozyskanej za pośrednictwem danego protokołu. Metryka jest umieszczana wtedy, gdy jest podawana przez protokół rutingu. W przeciwnym wypadku jest wyzerowana. Metryka podawana przez RIP, wedle której odległość od danej sieci mierzona jest liczbą ruterów pośredniczących, na poziomie jądra linuxa jest ignorowana, ale mogą z niej korzystać protokoły rutingu.

Tablica rutingu rutera R1 widziana z poziomu protokołu BGP została przedstawiona na rysunkach 9c i 9e. Rysunki zawierają obraz fragmentu ekranu konsoli rutera R1, po połączeniu się za pomocą usługi telnet z protokołem BGP i wydaniu polecenia show ip bgp. Jako pierwsza pojawia się informacja o wersji tablicy (tu: 0). Wersja powinna się zmieniać po zmianie najlepszej trasy do danej sieci. W przypadku ruterów Cisco wersja zmienia się, w analizowanym module Zebra/Quagga wersja pozostaje niezmieniona, najprawdopodobniej zmiana wersji nie została zaimplementowana. Następnie wyświetlony został identyfikator rutera. W przykładowym eksperymencie identyfikator został wybrany automatycznie (jest to najwyższy numer interfejsu rutera).

Pierwsza kolumna tablicy Loc-RIB protokołu BGP, pokazanej na rysunkach 9c i 9e, zawiera znaczniki stanu trasy w postaci trójznakowego ciągu oznaczeń kodowych. Na pierwszym znaku ciągu kodowany jest stan trasy BGP sensu stricte. I tak znacznik s (z ang. *suppressed*, od: *supress*, dosł. zakazywać, powstrzymywać) oznacza, że BGP zna trasę do danej sieci, ale jej nie rozgłasza. Znacznik d (z ang. *damped*, tłumiony) wskazuje, że trasa nie jest rozgłaszana, gdyż jest niestabilna (w stanie „*flaps*", dosł. trzepotania, - tj. gdy przechodzi naprzemiennie pomiędzy stanem aktywnym i nieaktywnym. Znacznik h (z ang. *history*, historia) wskazuje, iż BGP zna sieć, ale aktualnie nie ma do niej ważnej trasy. Znacznik * (ang. *valid*, ważny) symbolizuje trasę aktualną. Znacznik r (z ang. *RIB-failure*, tablica RIB jest niesprawna) opisuje sytuację, gdy trasa jest rozgłaszana przez BGP i funkcjonuje na poziomie BGP, ale ze względów technicznych nie może być zainstalowana w tablicy rutingu RIB rutera. Znacznik S (ang. *Stale*, dosł. nieświeży) oznacza trasę wymagającą odświeżenia. Trasy są oznaczane jako nieświeże, gdy, przykładowo, sąsiedni ruter nie odpowiada, bo się restartuje. Znacznik R (ang. *Removed*, usunięty) oznacza trasę usuniętą.

Znacznik ">", występujący jako drugi w kolejności znacznik stanu, oznacza najlepszą (ang. *best*) trasę spośród dwóch lub więcej tras alternatywnych. Jeżeli trasa nie zostanie uznana za najlepszą, znacznik jest znacznikiem pustym, a na ekranie wyświetlony zostanie znak spacji.

Trzeci znacznik stanu wskazuje, czy trasa pozyskana została z sesji IBGP. Jeżeli tak, to trzeci znacznik stanu przyjmuje wartość i (z ang. *internal*). W przeciwnym wypadku znacznik jest pusty (znak spacji). Dotyczy to również tras pozyskanych z sesji EBGP. Jak widać na rysunkach 9c i 9e, wszystkie trasy są trasami aktualnymi,

choć nie wszystkie zostały uznane za najlepsze. Sześć z nich (do sieci S1...S4, S6, S8 i S9) zostało pozyskanych z sesji IBGP.

Druga, po znacznikach stanu, kolumna tablicy Loc-RIB protokołu BGP, zawiera sieć docelową (adres własny sieci docelowej wraz z maską sieci podaną w formacie skróconym). Trzecia kolumna zawiera informację o najbliższym ruterze na trasie. Jak widać na rysunku, sieć S1 (sieć, dla której ruter R1 jest jedynym ruterem) została wskazana jako podłączona bezpośrednio i jest to trasa uznana za najlepszą dla tej sieci. Pozostałe dwie trasy do sieci bezpośrednio dołączonych, czyli trasy do sieci S2 i S4, prowadzą przez, odpowiednio, rutery R2 i R9. Trasy te zostały pozyskane od sąsiadów, którzy te sieci ogłaszali i nie zostały uznane za najlepsze. Wpisy dla sieci S2 i S4 są podwójne - drugi wpis występuje bezpośrednio pod pierwszym wpisem, adres własny sieci i maska nie są w nim podane, co sygnalizuje, że odnosi się on do sieci umieszczonej powyżej w tablicy RIB. Drugi wpis jest poczyniony z punktu widzenia R1. Pole następnego rutera jest w nim wyzerowane, a trasa wskazana została jako najlepsza.

Następna, czwarta kolumna, zawiera metrykę (ang. *metric*) trasy. Metryka jest podawana, gdy ruter odbierze atrybut MED (ang. *Multi-Exit Discriminator*). Im mniejsza wartość metryki, tym trasa ma wyższy priorytet.

Piąta kolumna (oznaczona LocPrf) zawiera wartość lokalnych preferencji (lokalnych dla danego systemu autonomicznego). Im większa wartość lokalnych preferencji, tym dany wpis ma wyższy priorytet. Wartość domyślna parametru LocPrf to 100.

Szósta kolumna zawiera wagę (ang. *weight*), będącą kolejnym parametrem lokalnym dla rutera. Im większa wartość wagi, tym wyższy priorytet wpisu.

Wartość 32768 to wartość maksymalna dla parametrów Metric, LocPrf i Weight. Taka wartość *weight* została nadana trasom do sieci bezpośrednio podłączonych do rutera R1.

Przedostatnia, siódma kolumna zawiera ścieżkę (ang. *path*), czyli listę obcych systemów autonomicznych, przez które prowadzi trasa do danej sieci. System autonomiczny danego rutera (tu: system AS 12) nie jest uwzględniany. Jak widać na rysunku, trasa do sieci S6 prowadzi przez system autonomiczny AS 56, a trasa do sieci S8 i S9 przez systemy autonomiczne AS 56 i AS 18.

Ostatnia, ósma kolumna zawiera kod pochodzenia wpisu w tablicy Loc-RIB protokołu BGP (ang. *origin codes*), wskazującym, z jakiego źródła BGP wziął daną trasę. Kod `i` (z ang. *IGP*, *interior gateway protocol*, protokół rutingu wewnętrznego) wskazuje, że wpis ma oparcie w tablicach rutingu. Innymi słowy, dana sieć jest widziana przez jakiś protokół rutingu wewnętrznego i (lub) konfigurację rutera. Kod `e` (ang. *EGP*, *Exterior Gateway Protocol*) oznacza, że wpis został otrzymany z protokołu EGP (obecnie nie używanego). Kod `?` (ang. *incomplete*) wskazuje na informację niekompletną.

Trasa musi nie mieć oparcia w tablicach rutingu wewnętrznego. Wszystkie trasy pokazane na rysunkach 9c i 9e mają oparcie w tablicach rutingu.

Opracowując założenia do rutingu zewnętrznego przyjęto, że rozgłaszane są sieci w systemach:

- AS 12: sieci S1, S2, S3 i S4,
- AS 56: sieć S6,
- AS 18: sieci S8 i S9.

Sieci S5 i S7 nie są rozgłaszane.

Jak widać na rysunkach 9c i 9e, sieci są rozgłaszane zgodnie z przyjętym założeniem. W tablicy znajdują się wpisy o trasach do sieci S1...S4, S6, S8 i S9. Sieci S5 i S7 nie ma w tablicy rutingu BGP. Trasa do sieci S5 (`80.26.9.0`) widoczna na poziomie systemu operacyjnego (rys. 9a) i modułu Zebra (rys. 9b), jak już wcześniej wspomniano, nie pochodzi z BGP ale z protokołu RIP pracującego w systemie AS 12.

PODSUMOWANIE

W artykule przedstawiono zarys nauczania rutingu zewnętrznego na uczelni technicznej. Wskazano jak można rozwiązać nauczanie kwestii praktycznych rutingu zewnętrznego na laboratoriach. Zaproponowano wykorzystanie ruterów dostępowych oraz emulatora sieci komputerowych Netkit.

BIBLIOGRAFIA

1. Bates T., Chandra R., Katz D., Rekhter Y., *Multiprotocol Extensions for BGP-4*. RFC 4760, January 2007.
2. Chodorek A., Chodorek R., *Możliwości zastosowania emulatora Netkit w badaniach naukowych i dydaktyce*. Logistyka 2014, nr 6.
3. Hawkinson J., Bates T., *Guidelines for creation, selection, and registration of an Autonomous System (AS)*. RFC 1930 (BCP 6), March 1996.
4. Pizzonia M., Rimondini M., *Netkit: network emulation for education*. Software: Practice and Experience, 2014.
5. Rekhter Y. (Ed.), Li T. (Ed.), Hares S. (Ed.): *A Border Gateway Protocol 4 (BGP-4)*. RFC 4271. January 2006.
6. Rekhter Y., Li T., *A Border Gateway Protocol 4 (BGP 4)*. RFC 1771, March 1995.
7. Rekhter Y., Gross P., *Application of the Border Gateway Protocol in the Internet*. RFC 1772, March 1995.
8. Vohra Q., Chen E., *BGP Support for Four-octet AS Number Space*. RFC 4893, May 2007.

THE USAGE OF ACCESS ROUTERS AND THE NETKIT EMULATOR IN THE TEACHING PROCESS OF SELECTED EXTERNAL ROUTING PROBLEMS

Abstract

The external routing is so called the strategic routing in the Internet. It is responsible for the proper distribution of traffic between autonomous systems. The paper outlines the teaching of external routing issues at a technical university with special emphasis on practical aspects of the learning process in laboratories. For teaching of practical issues of external routing usage of the access routers and the NetKit computer network emulator is proposed.

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