Konfiguracja routera CISCO

Dostęp do urządzenia

Aby uzyskać dostęp do konfiguracji routera lub przełącznicy firmy Cisco, należy podłączyć port konsoli urządzenia do portu seryjnego komputera specjalnym kablem Cisco. Następnie, należy podłączyć się z routerem z użyciem programu symulującego konsolę (program PuTTY albo HyperTerminal w Windowsie, lub minicom w Linuksie). Po uzyskaniu dostępu do urządzenia Cisco, w każdej chwili można uzyskać pomoc naciskając klawisz **?**. Wyświetli się lista komend dostępnych w aktualnym trybie. Urządzenia Cisco mają też funkcję autouzupełniania, którą aktywuje się klawiszem Tab. Funkcja ta działa podobnie do funkcji autouzupełniania znanej z systemów uniksowych. Ma dodatkowo tę zaletę, że po wpisaniu komendy również można nacisnąć **?**, co spowoduje wyświetlenie wszystkich dostępnych argumentów. Klawisz **?** można naciskać po wpisaniu dowolnej ilości argumentów, zawsze uzyska się listę dostępnych opcji. Jeżeli komenda jest już kompletna, wyświetli się symbol **<cr>**.

Podstawowe komendy

System operacyjny Cisco IOS umożliwia wydawanie komend w kilkunastu trybach. Tryby różnią się uprawnieniami, jakie dają użytkownikowi oraz zakresem, w jakim możliwa jest konfiguracja. Po zalogowaniu się na urządzenie, pierwszym dostępnym trybem jest tryb użytkownika (User Exec Mode). Ten tryb nie daje wiele możliwości poza diagnostyką (np. przy użyciu komend ping, traceroute). Umożliwia tez zalogowanie się do wyższych trybów, przede wszystkim do trybu uprzywilejowanego (Privileged Exec Mode), który można porównać do roota w Linuksie lub administratora w Windowsie. Aby zalogować się do trybu uprzywilejowanego, wydajemy komendę **enable**.

```
Router> enable
```

Po wydaniu tej komendy, można przejść od razu do rzeczy:

```
Router# configure terminal

Router(config)# Interface FastEthernet 0/1

Router(config-if)# ip address 192.168.1.1 255.255.255.0
```

```
Router(config-if)# no shutdown

Router(config-if)# description Polaczenie z routerem w duzym pokoju
```

Powyższe komendy nadały adres IP 192.168.1.1 maska podsieci 255.255.255.0 drugiemu interfejsowi pierwszej karty sieciowej Fast Ethernet routera. Komenda **no shutdown** włączyła interfejs. Ostatnia z komend dodała opis interfejsu (**Polaczenie z routerem w duzym pokoju**), który będzie widoczny przy wydawaniu komend takich jak **show interfaces FastEthernet 0/1**.

```
Router# show running-config

Router# show startup-config

Router# copy running-config startup-config
```

Powyższe komendy kolejno: pokażą bieżącą konfigurację urządzenia (running-config), zapisaną konfigurację startową (startup-config) oraz zapiszą bieżąca konfigurację do pamięci tak, aby była osiągalna po restarcie urządzenia (ostatnia z komend).

```
Router# configure terminal

Router(config)# hostname RouterPoznan

RouterPoznan(config)#

RouterPoznan(config)# no hostname

Router(config)#
```

W powyższym przykładzie użytkownik był już zalogowany do trybu uprzywilejowanego. Następnie wydał komendę **configure terminal**, aby przejść do trybu konfiguracyjnego. Następnie przy pomocy komendy **hostname** nadał routerowi nazwę RouterPoznan. W przedostatniej linijce użytkownik usuwa nazwę hosta komenda **no hostname**. Komenda **no** służy do wyłączania wszystkich wcześniej wydanych komend (np. **no ip address** wydane w odpowiednim trybie osunęłoby adres IP interfejsu).

Hasła

główne hasło

Każde urządzenie Cisco umożliwia zabezpieczenie go przy pomocy kilku haseł chroniących rożne elementy. Aby uniemożliwić dostęp do trybu uprzywilejowanego osobom nieupoważnionym, należy skonfigurować hasło jedną z dwóch poniżej przedstawionych komend:

```
Router# configure terminal

Router(config)# enable password noweHaslo

Router(config)# enable secret noweHaslo

Router(config)# service password-encryption
```

Po przełączeniu się na tryb konfiguracyjny, komenda **enable password** powoduje, że po wydaniu komendy **enable** w trybie użytkownika będziemy musieli podać hasło. Hasło to jednak nie będzie szyfrowane; każdy, kto w odpowiednim trybie wyda komendę **show running-config** lub uzyska dostęp do kopii zapasowej pliku konfiguracyjnego urządzenia, będzie mógł odczytać hasło w postaci zwykłego tekstu. Komenda **enable secret** szyfruje hasło (w pliku konfiguracyjnym w miejscu hasła pojawi się ciąg znaków reprezentujących hasło). Ostatnia z wyżej przytoczonych komend powoduje, że wszystkie hasła urządzenia nadane zarówno po jak i przed wydaniem tej komendy zostaną zaszyfrowane. Jest to jednak szyfrowanie słabe, dlatego najlepiej użyć kombinacji dwóch ostatnich komend: **enable secret** - szyfrowanie hasła dostepowego do trybu **Privileged Exec Mode** oraz **service password-encrytption** - szyfrowanie wszystkich innych hasel.

telnet

Aby móc zalogować się na router lub przełącznicę korzystając z usługi telnet, należy uprzednio skonfigurować hasło dla wirtualnych terminali urządzenia. Bez nadania hasła usługa telnet nie będzie osiągalna. Aby dodać hasło, wydaj następujące komendy (rozpoczynając w trybie konfiguracyjnym):

```
Router(config)# line vty 0 4

Router(config-line)# password noweHaslo

Router(config-line)# login

Router(config-line)# exit
```

```
Router(config)#
```

Powyższe komendy nadały hasło **noweHasło** pięciu wirtualnym terminalom (0 - 4). Komenda **login** jest konieczna, aby hasło było wymagane przy logowaniu przez telnet. Bez tej wydania tej komendy przy konfiguracji hasła, późniejsze zalogowanie przez telnet nie będzie możliwe.

konsola

Urządzenia Cisco wymagają użycia konsoli (lub programu imitującego konsolę) przy pierwotnej konfiguracji urządzenia. Oznacza to, że aby dokonać pierwszej konfiguracji użytkownik musi mieć fizyczny dostęp do urządzenia. Daje to pewien stopień zabezpieczenia przed nieautoryzowaną zmianą konfiguracji. Jako dodatkowe zabezpieczenie, warto też stosować hasło chroniące urządzenie przed nieautoryzowanymi osobami mającymi do niego fizyczny dostęp. Aby skonfigurować hasło dla konsoli, należy wydać takie komendy:

```
Router(config)# line console 0

Router(config-line)# password noweHaslo

Router(config-line)# login

Router(config-line)# end

Router#
```

Zasada używania powyższych komend jest taka sama, jak komend używanych przy konfiguracji usługi telnet (patrz wyżej).

Diagnozowanie połączenia sieciowego

Oto dwie komendy pokazujące informacje o interfejsach sieciowych. Pierwsza pokaże ogólne informacje o wszystkich interfejsach, druga pokaże szczegółowe informacje o drugim interfejsie seryjnym pierwszej seryjnej karty sieciowej:

```
Router# show ip interface brief
```

```
Router# show interfaces serial 0/1
```

Na urządzeniach Cisco można wydawać standardowe komendy diagnostyczne: **ping** i **traceroute.** Można też wydać komendę **arp**, aby sprawdzić adresy MAC skojarzone z adresami IP przechowywane aktualnie w pamięci RAM urządzenia.

Aby wyświetlić tabelę trasową, należy wydać następującą komendę:

```
Router# show ip route
```

Komendy związane z systemem operacyjnym

Komenda pokazująca wszystkie informacje dotyczące aktualnie zainstalowanego na urządzeniu systemu IOS to:

```
Router# show version
```

Poza informacjami o systemie operacyjnym (numer wersji itp.), komenda ta wyświetli także ilość zainstalowanego RAMu oraz NVRAMu, ilość pamięci Flash oraz informacje o interfejsach sieciowych. Aby wyświetlić listę plików zawartych w pamięci Flash urządzenia, należy wpisać co następuje:

```
Router# show flash:
```

Aby usunąć któryś z plików zawartych w pamięci Flash, należy wydać komendę:

```
Router# del nazwaPliku
```

Kiedy na routerze konfigurujemy interfejs i przyłączamy działający kabel sieciowy, dany interfejs zmienia stan na „up and up”, czyli: połączenie zostało wykryte i uruchomione zostały wymagane protokoły. Sieć, do której router jest przyłączony za pośrednictwem interfejsu w stanie „up and up” automatycznie pojawia się w tabeli trasowania jako "bezpośrednio przyłączona" (directly connected). Aby to sprawdzić, wydajemy następująca komendę:

```
Router# show ip route
```

Aby zaobserwować zmiany, które zachodzą w konfiguracji routera, gdy przyłączamy go do sieci, można wydać poniższa komendę:

```
Router# debug ip routing
```

Po wydaniu tej komendy, uruchamiamy interfejs sieciowy:

```
Router# configure terminal

Router(config)# interface fa0/0

Router(config-if)# ip address 192.168.1.1 255.255.255.0

Router(config-if)# no shutdown
```

Po wydaniu powyższych komend na ekranie powinniśmy zobaczyć informacje o kolejno po sobie następujących zdarzeniach (uruchomienie interfejsów, dodanie sieci do tabeli trasowych).

Ponieważ debugowanie powoduje niepotrzebne obciążenie procesora i pamięci RAM, najlepiej włączać je tylko wtedy, kiedy mamy ważne powody. Po zakończeniu debugowania należy wydać następująca komendę:

```
Router# undebug all
(ewentualnie:)
Router# undebug ip routing
```

CDP

Urządzenia Cisco używają opatentowanego protokołu CDP aby wykryć inne bezpośrednio przyłączone urządzenia tej samej firmy. Aby zobaczyć urządzenia wykryte przez nasz router lub przełącznicę, należy wydać następującą komendę:

```
Router# show cdp neighbors
```

Ze względów bezpieczeństwa, Cisco zaleca wyłączenie tego protokołu (domyślnie jest on włączony) przy użyciu następującej komendy:

```
Router(config)# no cdp run
(lub:)
Router(config-if)# no cdp enable
```

Pierwsza z powyższych komend wyłącza CDP dla całego urządzenia, a druga jedynie dla wybranego interfejsu.

Trasowanie statyczne

z użyciem adresu interfejsu routera pośredniego

Generalnie, zaletą routerów jest to, ze używają dynamicznych protokołów trasowania. Jednak w niektórych przypadkach konieczne (lub wygodne) jest skonfigurowanie statycznych tras. Kiedy połączymy dwa routery za pomocą kabla (kabel seryjny lub Ethernet) i nadamy im adresy IP z tej samej sieci - przykładowo: router **R1** 192.168.1.1/30 oraz router **R2** 192.168.1.2/30, następnym krokiem możne być skonfigurowanie statycznej trasy z **R1** do przykładowej sieci 192.168.1.128/25 przyłączonej do **R2**. Pośrednim routerem („next hop") do sieci 192.168.1.128/25 dla routera **R1** będzie interfejs 192.168.1.2 na routerze **R2.** Oto jak można to osiągnąć:

```
R1(config)# ip route 192.168.1.128 255.255.255.128 192.168.1.2
```

W powyższym przykładzie 192.168.1.128 255.255.255.128 to adres docelowej podsieci, a 192.168.1.2 to „next hop", czyli adres routera do którego należy przesyłać pakiety przeznaczone dla urządzeń podłączonych do owej podsieci.

z użyciem numeru lokalnego interfejsu

Drugim sposobem na skonfigurowanie trasy do odległej sieci jest użycie nazwy i numeru lokalnego interfejsu, przez który prowadzi trasa do sieci docelowej:

```
R1(config)# ip route 192.168.1.128 255.255.255.128 serial 0/0
```

W tym przykładzie, podobnie jak w poprzednim, adres 192.168.1.128/25 (=192.168.1.128 255.255.255.128) reprezentuje sieć docelową. Jednak zamiast adresu IP sąsiedniego routera, użyliśmy nazwy i numeru interfejsu wyjściowego, przez który należy wysyłać pakiety przeznaczone dla sieci 192.168.1.128/25.

W sytuacji, kiedy dwa routery połączone są siecią Ethernet, nie wystarczy skonfigurowanie interfejsu wyjściowego. Aby pakiety dotarły do celu, trasa powinna być skonfigurowana z uwzględnieniem zarówno interfejsu wyjściowego jak i adresu IP najbliższego routera. Konfiguruje się to następująco:

```
R1(config)# ip route 192.168.1.128 255.255.255.128 fastethernet 0/1 192.168.1.2
```

Daje to pewność, że pakiety wysłane przez interfejs **fastethernet 0/1** dotrą do routera o adresie 192.168.1.2.

Aby usunąć powyższą trasę, używamy komendy **no**:

```
R1(config)# no ip route 192.168.1.128 255.255.255.128 192.168.1.2
(lub:)
R1(config)# no ip route 192.168.1.128 255.255.255.128 serial 0/0
```

I jeszcze jedna ważna do skonfigurowania trasa: trasa do bramy domyślnej (default route, default gateway). Konfiguracja jest prosta:

```
R1(config)# ip route 0.0.0.0 0.0.0.0 serial 0/0
```

RIP

RIP i RIPv2 to jedne z najstarszych protokołów trasowania. Są ciągle używane ze względu na ich prostotę, popularność a także kompatybilność pomiędzy różnymi producentami sprzętu i oprogramowania. Obydwa te protokoły będą w moich notatkach omówione w jednym artykule, choć podręcznik „Routing Protocols and Concepts” wydany przez wydawnictwo Cisco Press poświęca im dwa osobne rozdziały.

Wspólne komendy RIPv1 i RIPv2

Aby uruchomić RIP i dodać do niego kilka przykładowych sieci, wpisz:

```
R1# configure terminal
R1(config)# router rip
```

```
R1(config-router)# network 192.168.1.0
R2(config-router)# network 192.168.2.0
```

Komendy te uruchomiły proces RIP (**router rip**) oraz dodały do niego sieci **192.168.1.0** oraz **192.168.2.0**. Sieci te będą uwzględnione w pakietach wysyłanych przez proces RIP do sąsiednich routerów, jak również RIP będzie nasłuchiwał na interfejsach należących do tych sieci aby wyłapać pakiety procesu RIP wysłane przez sąsiednie routery.

Kiedy polaczymy ze sobą przynajmniej dwa routery używające protokołu RIP, routery wymienia miedzy sobą informacje na temat posiadanych tras (uwzględnianych przez proces RIP). Aby zweryfikować czy nasze trasy pojawiły się w tabeli routera, można wydać komendę:

```
R1# show ip route
```

Trasy oznaczone literką **R** to trasy, o których router dowiedział się dzięki protokołowi **RIP.** Trasy oznaczone literką **C** są trasami do sieci bezpośrednio przyłączonych do routera.

Oczywiście, zawsze można wydać komendę **show running config**, aby sprawdzić bieżącą konfigurację routera. Aby dowiedzieć się więcej o protokołach trasowania aktywnych na danym urządzeniu, należy wydać następującą komendę:

```
R1# show ip protocols
```

Aby dowiedzieć się więcej o procesach zachodzących na urządzeniu (otrzymywanie pakietów RIP z innych urządzeń, wysyłanie pakietów, dodawanie nowych tras do tabeli routera) można włączyć debugowanie. Odpowiednią komendę umieszczam poniżej. Pod komendą włączającą debugowanie procesu RIP, pokazuję też komendę wyłączającą debugowanie. Ponieważ debugowanie zwiększa użycie procesora i RAM-u, należy je wyłączyć kiedy tylko zbierze się wszystkie potrzebne informacje.

```
R1# debug ip rip
(aby wyłączyć:)
R1# undebug all
```

Kiedy wydajemy komendę **network** przy konfiguracji protokołu **RIP**, sieć, którą wpiszemy po tej komendzie, zostaje dodana do procesu RIP i zostaje rozreklamowana pośród sąsiadujących routerów. Przez interfejs należący do tej sieci również wysyłane są pakiety protokołu RIP. Jeżeli na danej sieci

nie znajduje się żaden inny router, wysyłanie takich pakietów jest zbędne i niepotrzebnie zwiększa natężenie ruchu. Aby tego uniknąć, należy interfejs należący do danej sieci skonfigurować jako pasywny. Robi się to za pomocą tej komendy (przykładowy interfejs to FastEthernet 0/0):

```
R1(config)# router rip
R1(config-router)# passiwe-interface FastEthernet 0/0
R1(config-router)# end
```

W poprzedniej części moich notatek pokazywałem jak dodać bramę domyślną (na przyklad **ip route 0.0.0.0 0.0.0.0 serial 0/0**). Aby brama domyślna została rozpropagowana przy użyciu protokołu RIP, należy wydać komendę:

```
R1(config)# router rip
R1(config-router)# default-information originate
R1(config-router)# end
```

Komenda powyższa jest także używana przy konfiguracji bramy domyślnej w protokole **OSPF.** Natomiast protokół **EIGRP** wymaga innej komendy: **redistribute static.**

Podstawową wadą RIPv1 jest to, że nie daje możliwości stosowania CIDR (Classless Inter-Domain Routing), czyli innymi słowy poszczególne sieci muszą należeć do klasy A, B lub C i używać domyślnej maski podsieci odpowiedniej dla swojej klasy. We współczesnych sieciach jest to dużą wadą protokołu RIPv1. Na szczęście protokół RIPv2 daje możliwość stosowania CIDR.

RIPv2

Przejście z wersji RIP na RIPv2 jest bardzo proste:

```
R1(config)# router rip
R1(config-router)# version 2
```

Aby powrócić do wersji RIPv1, należy wydać tą samą komendę zamieniając **version 2** na **version1** .

RIPv2 w kwestii CIDR zachowuje się podobnie do RIPv1 dopóki nie wydamy następującej komendy:

```
R1(config)# router rip
R1(config-router)# no auto-summary
```

Jest to bardzo ważna komenda, jeśli nasze sieci używają niestandardowych masek podsieci (co jest bardzo częste). Bez niej routery dokonają automatycznego podsumowania sieci, co prowadzić będzie do błędów.

I jeszcze jeden drobiazg, który może okazać się bardzo ważny. Kiedyś adresy IP podzielone były na klasy: A, B lub C (oraz D i E). W klasie A było mniej sieci, ale za to każda z nich mogła pomieścić więcej hostów. W klasie B było trochę więcej sieci, ale sieci mieściły mniej hostów. W klasie C było bardzo wiele sieci, ale każda mogła mieć najwyżej 254 hosty. Dzisiaj podział ten nie jest tak istotny dzięki stosowaniu maski podsieci o zmiennej długości. Routery mogą zachowywać się na dwa sposoby: z uwzględnieniem klas (classful routing behaviour) lub pozwalając na dowolną długość maski podsieci (classless routing behaviour). Od dłuższego już czasu routery domyślnie stosują to drugie podejście (classless routing behaviour). Jednak ciągle istnieje komenda, którą to zachowanie można modyfikować. Oto ta komenda:

```
R1(config)# no ip classless
(powyższa komenda powoduje, że router uwzględnia klasy sieci)
R1(config)# ip classless
(powyższa komenda powoduje, że router nie uwzględnia klas sieci; jest to ustawienie
domyślne
nowszych urządzeń)
```

Kilka dodatkowych informacji

Protokół RIPv1 może współdziałać z protokołem RIPv2. W przypadku, gdy router używający protokołu RIPv1 dowie się o jakiejś sieci dzięki aktualizacji otrzymanej od routera używającego protokołu RIPv2, nie uwzględni on maski podsieci otrzymanej wraz z tą aktualizacją. Zamiast tego, zastosuje maskę domyślną lub maskę podsieci interfejsu, na którym dana aktualizacja została odebrana (co jest zgodne z normalnym zachowaniem RIPv1). Natomiast protokół RIPv2 domyślnie ignoruje aktualizacje wysyłane zgodnie z protokołem RIPv1.

Aby sprawdzić, której wersji RIP używa router, należy wpisać:

```
R1# show ip protocols
```

Aktualizacje w obydwu wersjach protokołu RIP wysyłane są co 30 sekund. Protokoły RIP używają też kilku innych timerów, które tutaj tylko wymienię:

- Invalid (180 sekund)
- Flush (240 sekund)
- hold-down (180 sekund)

EIGRP

Protokół trasowania EIGRP jest to opatentowany protokół firmy Cisco, działający wyłącznie na urządzeniach tej firmy. W związku z tym, że firma Cisco zdominowała rynek routerów, jest to istotny i często używany protokół. Poniżej przeanalizujemy listę komend potrzebnych, aby ten protokół skonfigurować na routerze.

Wybór trasy

Protokół RIP, który konfigurowaliśmy ówcześnie wybiera najlepszą trasę na podstawie ilości routerów pomiędzy źródłem a celem wysyłanych pakietów. Za najlepszą zostaje uznana trasa z najmniejszą ilością routerów pośrednich. Maksymalna ilość routerów na trasie pakietu nie może przekroczyć 15. Jest to rozwiązanie proste i sprawne. Niestety problem pojawia się, kiedy najkrótsze fizycznie trasy mają niską przepustowość danych. Czasami pakiety dotarłyby do celu szybciej pokonując fizycznie dłuższą trasę, jeżeli przepustowość połączenia pomiędzy routerami na trasie dłuższej byłaby lepsza niż pomiędzy routerami na trasie krótszej. Problem ten rozwiązuje protokół EIGRP (ulepszona wersja wycofanego z użycia protokołu IGRP). W przypadku tego protokołu, podstawą do ustalenia najlepszej trasy jest przepustowość połączenia oraz opóźnienie, z jakim pakiety docierają do celu (**bandwidth** i **delay**).

Aktualizacje

Inną zaletą protokołu EIGRP w porównaniu z RIP jest to, że aktualizacje nie są wysyłane tak jak w przypadku RIP co 30 sekund (co niepotrzebnie zwiększa natężenie ruchu w sieci). Aktualizacje w EIGRP wysyłane są tylko wtedy, gdy nastąpi zmiana w konfiguracji sieci (dodanie nowego urządzenia, nowej sieci itp.). Zamiast aktualizacji co 30 sekund, wysyłane są bardzo niewielkie pakiety **hello**, które informują sąsiednie routery o obecności naszego urządzenia. Pakiety te ze względu na swój niewielki rozmiar nie zwiększają natężenia ruchu w sieci, w związku z tym wysyłane są dość często, co 5 sekund (w przypadku sieci o przepustowości mniejszej niż 1.544 Mbps jest to 60 sekund). Protokół EIGRP używa także innego timera, tzw. **hold time**, który decyduje jak długo informacja o sąsiednich routerach przechowywana jest w pamięci w przypadku nieotrzymania pakietów **hello. Hold time** w EIGRP wynosi 15 sekund (180 sekund w przypadku sieci o przepustowości mniejszej niż 1.544 Mbps).

Komendy

Aby rozpocząć proces EIGRP na routerze, należy wydać następującą komendę:

```
Router(config)# router eigrp 1
```

Ponieważ na routerze można jednocześnie uruchomić więcej niż jeden proces EIGRP (każdy dla jednego autonomicznego systemu), komenda **router eigrp** musi zawierać numer, jakim chcemy opatrzyć nasz proces EIGRP (porcess ID). Numer ten musi być taki sam na każdym routerze należącym do tego samego autonomicznego systemu co nasz, a więc na każdym routerze, z którym chcemy utrzymywać wymianę informacji na temat tras.

Oto jak dodaje się sieć do procesu EIGRP:

```
Router(config)# router eigrp 1
Router(config-router)# network 172.16.0.0
```

Ponieważ w powyższej komendzie nie została podana maska podsieci, router uwzględni w procesie EIGRP (a więc także w pakietach aktualizacyjnych wysyłanych do innych routerów) wszystkie znane mu podsieci mieszczące się w sieci klasy B 172.16.0.0/16. Aby zadeklarować wyłącznie wybranej sieci, potrzebna jest odwrócona maska podsieci (**wildcard mask**) określająca dokładnie, która sieć ma być użyta:

```
Router(config-router)# network 172.16.0.8 0.0.0.3
```

Powyższa komenda dodała sieć 172.16.0.8 maska podsieci 255.255.255.252 do procesu EIGRP. Aby obliczyć odwróconą maskę podsieci (**wildcard mask**), należy od adresu 255.255.255.255 odjąć maskę podsieci:

```
  255. 255. 255. 255
— 255. 255. 255. 252
-------------------
    0.   0.   0.   3
```

Weryfikacja

Aby router mógł wysyłać i odbierać pakiety aktualizujące informacje o trasach do innych sieci, najpierw musi ustanowić zależność przyległości z routerem po drugiej stronie. Do tego celu używane są pakiety **hallo**. Do weryfikacji, czy taka zależność została utworzona, służy następująca komenda:

```
Router# show ip eigrp neighbors
```

Informacja uzyskana po wydaniu tej komendy będzie zawierała m.in. adres IP przyległych routerów, lokalny interfejs, na którym dany router został wykryty, czas trwania zależności przyległości i licznik timera **hold**.

W przypadku protokołu EIGRP należy zwrócić uwagę, aby nie wydawać komendy **passive-interface** na interfejsach łączących z innymi routerami, gdyż uniemożliwi to routerom wymianę informacji na temat tras. Natomiast tak samo jak w innych protokołach, ogólne informacje o procesie EIGRP na lokalnym routerze można uzyskać wydając komendę **show ip protocols**. Kiedy zbudujemy sieć składającą się z więcej niż jednego routera, możemy zweryfikować tablicę trasowania na naszym routerze następującą komendą:

```
Router# show ip route
```

Komenda ta pokaże wszystkie trasy do sieci znane naszemu routerowi. Przy każdej sieci widoczna będzie symbol reprezentujący protokół, dzięki któremu router dowiedział się o danej trasie. Protokół EIGRP oznaczony jest literką **D**.

Inne ważne cechy EIGRP

Jak już pisałem wcześniej, protokół EIGRP ustala, która trasa jest używana do przesyłania danych na podstawie przepustowości (bandwidth) i opóźnienia (delay) danego połączenia. Wartości te są ustalane przez administratora sieci i przypisywane do interfejsów na odpowiednich routerach; mogą być zmieniane w zależności od potrzeb. Istnieje również możliwość uwzględniania innych wartości: skuteczność (reliability) i obciążenie (load). Te wartości obliczane są co pięć minut automatycznie przez router i reprezentowane w wartości od 0 do 255 (im wyższa wartość tym lepsze połączenie). Należy pamiętać, że domyślnie protokół EIGRP używa tylko dwóch pierwszych z wymienionych tu wartości (przepustowość i opóźnienie).

Administrator sieci ma wpływ na to, które połączenie protokół EIGRP uzna za najlepsze (czyli którego będą używały routery). Można to regulować zmieniając przepustowość danego interfejsu następującą komendą w trybie konfiguracyjnym interfejsu (w tym przykładzie Serial 0/0):

```
Router# interface serial 0/0
Router(config-if)# bandwidth 64
```

Jednostka stosowana w tej komendzie to kbps (kilobity na sekundę). Komenda ta nie zmienia rzeczywistej przepustowości danego interfejsu. Jest to jedynie wartość umowna używana przez niektóre protokoły (np. EIGRP i OSPF).

DUAL

DUAL to nazwa algorytmu używanego przez protokół EIGRP. Dzięki temu algorytmowi, protokół EIGRP jest szybszy niż protokół RIP. Protokół EIGRP przechowuje w pamięci routera nie tylko najkrótszą trasę do odległej sieci, ale także trasę rezerwową. Najkrótsza trasa do odległej sieci określana jest jako **`Successor`**, a trasa zapasowa określana jest jako **`Feasible Sucessor`**. Jeżeli trasa określana jako **`Successor`** przestanie być osiągalna, router nie musi wykonywać żadnych kalkulacji. Może od razu zacząć używać trasy zapasowej, **`Feasible Successor`**. Aby zobaczyć obie te trasy przechowywane w tablicy topologicznej routera, użyj następującej komendy:

```
Router# show ip eigrp topology
```

Wyświetli się tabela podająca informacje o aktualnych trasach jak również o trasach zapasowych. Aby wyświetlić wszystkie trasy do odległych sieci - nie tylko najlepszą bądź zapasową - wydaj następującą komendę:

```
Router# show ip eigrp topology all-links
```

Powyższa komenda wyświetli wszystkie trasy do odległej sieci. Nie ma jednak gwarancji, że trasy te nie zapętlają się. Dla tego jeśli zaistnieje sytuacja, że którakolwiek z tych tras będzie miała być użyta, algorytm DUAL będzie użyty przez router do dokonania odpowiednich wyliczeń. To zabierze więcej czasu i zakłóci na jakiś czas komunikację na sieci.

Automatyczne podsumowywanie

Protokół EIGRP automatycznie podsumowuje sieci z tego samego zakresu i tej samej klasy. W nowoczesnych sieciach, gdzie nie stosuje się klas i używa się masek podsieci o zmiennej długości może to prowadzić do błędów. Aby tego uniknąć, podobnie jak w protokole RIPv2, należy wydać poniższą komendę:

```
Router(config)# router eigrp 1
Rotuer(config-router)# no auto-summary
```

Ręczne podsumowywanie

Dla przykładowych sieci: **192.168.1.0/24, 192.168.2.0/24 oraz 192.168.3.0/24** routery mogą utrzymywać osobne wpisy w tabelach trasowych. Jednak po dokonaniu odpowiednich obliczeń, można również dokonać podsumowania tych trzech sieci jako **192.168.0/22** wydając taką komendę w trybie konfiguracyjnym interfejsu:

```
Router(config)# interface serial 0/0
Router(config-if)# ip summary-address eigrp 1 192.168.0.0 255.255.252.0
Router(config-if)# interface serial 0/1
Router(config-if)# ip summary-address eigrp 1 192.168.0.0 255.255.252.0
```

Komendę tę należy wydać dla każdego interfejsu, który uczestniczy w procesie EIGRP (w powyższym przykładzie interfejsy seryjne 0/0 i 0/1).

Brama domyślna

Brama domyślna w protokole EIGRP dodawana jest przy użyciu następującej komendy:

```
Router(config)# ip route 0.0.0.0 0.0.0.0 serial 0/0
Router(config)# router eigrp 1
Router(config-router)# redistribute static
```

Komenda ta różni się od komendy używanej w protokołach RIP i OSPF.