Realizacja routingu dynamicznego za pomocą protokołu OSPF.

Autorzy: Paweł Bugera, Dominik Bemer IVFDS

STRESZCZENIE

Praca jest poświęcona zagadnieniom routingu dynamicznego z wykorzystaniem protokołu OSPF. W dużych sieciach wielosegmentowych routing dynamiczny jest bowiem podstawową metodą zdobywania wiedzy, dzięki której routery poznają topologię sieci oraz budują tabele routingu. Wymiana informacji między routerami odbywa się zgodnie z określonymi algorytmami i przy wykorzystaniu protokołów routingu dynamicznego.
Protokół OSPF zaliczany jest do protokołów stanu połączenia. Protokoły stanu połączenia wymagając większej mocy obliczeniowej, zapewniają większy stopień kontroli nad procesem kierowania ruchem datagramów w sieci i szybciej dostosowują się do zmian struktury sieci. Protokół OSPF jest przystosowany do pracy w dużych systemach autonomicznych. Każdy router pracujący z protokołem OSPF musi znać strukturę sieci, w której pracuje.

Rozdział pierwszy ma na celu ogólne wprowadzenie do tematyki routingu i jego zastosowania. Omówiono w nim budowę tablic routowania, oraz ogólną klasyfikację protokołów routowania dynamicznego na którą składają się protokoły z wektorem odległości i protokoły stanu przyłączeń. Omówiono również właściwości algorytmów routowania takie jak poprawność, prostota, stabilność oraz konwergencja.

W rozdziale drugim protokół OSPF został przedstawiony od strony teoretycznej. Ukazano w nim format pakietu protokołu OSPF, jak i sam sposób konfiguracji protokołu.

Trzeci rozdział ma na celu ukazanie podstawowej konfiguracji routera. Wiążą się z tym takie zagadnienia jak: uruchomienie routera, dialog konfiguracyjny pozwalający utworzyć pierwszą, bazową konfigurację routera. W dalszej części rozdziału przedstawione zostały tryby pracy i zarządzanie skryptem konfiguracyjnym jak również konfiguracja interfejsów czyli właściwe zdefiniowanie parametrów komunikacyjnych dla poszczególnych interfejsów sieci lokalnej.

Czwarty rozdział poświęcony został praktycznej realizacji protokołu OSPF na dwóch routerach firmy CISCO 1600 dla przykładowej sieci.

SPIS TREŚCI

1. KONCEPCJA ŁĄCZENIA SIECI IP

Jedną z podstawowych funkcji protokołu IP jest routowanie. Umożliwia ono przesyłanie datagramów poprzez wiele sieci do miejsca przeznaczenia. Routowanie polega na ustaleniu ścieżki połączeń między kolejnymi routerami, do miejsca przeznaczenia pakietu. Kiedy jakieś urządzenie sieciowe ma wysłać pakiet do innego urządzenia sieciowego, wówczas mogą zajść następujące przypadki:

- Węzeł sieci, do którego skierowany jest pakiet, jest albo bezpośrednio połączony z urządzeniem sieciowym mającym wysłać ten pakiet, albo znajduje się w tej samej sieci co wspomniane urządzenie sieciowe. W obu tych sytuacjach pakiet może być bezpośrednio przesyłany do węzła docelowego.
- Węzeł sieci, do którego skierowany jest pakiet, nie znajduje się w tej samej sieci/podsieci co urządzenie sieciowe mające wysłać. W tej sytuacji urządzenie powinno podjąć decyzję o wyborze adresu urządzenia sieciowego, które przejmie odpowiedzialność za dalsze przesłanie pakietu.

Jeśli urządzeniem sieciowym wysyłający pakiet jest, np. komputer, to w sytuacji drugiej pakiet wysyłany jest do najbliższego routera, którego adres określony jest w konfiguracji interfejsu sieciowego tego komputera jako *gatweay*. Jeśli natomiast urządzeniem tym jest router, to musi on podjąć decyzję o dalszej drodze pakietu na podstawie posiadanych przez siebie informacji. Informacje te w routerze zawarte są w tablicy routowania [8].

1.1 Budowa tablic routowania

Po wykonaniu wstępnej konfiguracji routera (określenie adresów sieciowych dla poszczególnych interfejsów) jedynymi dostępnymi sieciami są te, do których ma on bezpośredni dostęp. Oznacza to, że na routerach Cisco, segmenty sieci bezpośrednio dostępne dla routera są pierwszym, najbardziej wiarygodnym źródłem informacji o strukturze sieci.

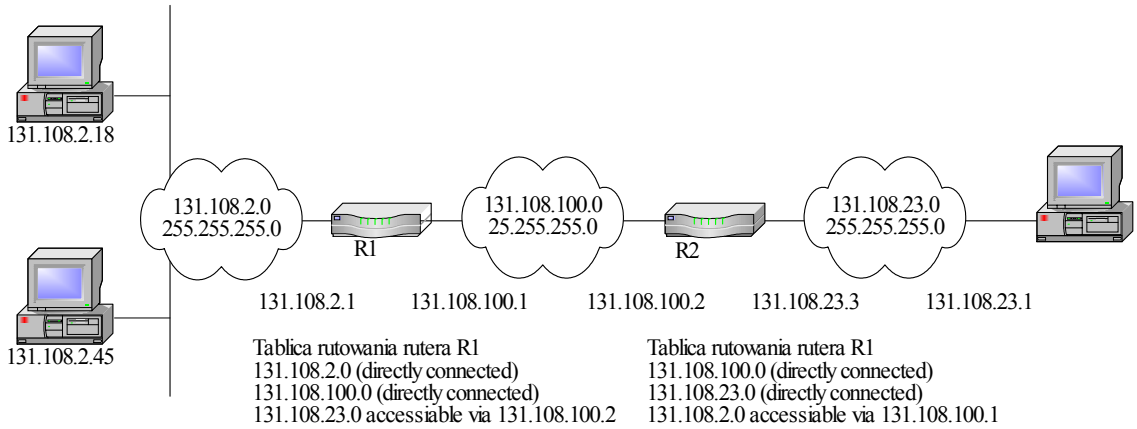
Oczywiście, nie jest to obraz całej sieci, tak więc muszą istnieć sposoby na poinformowanie routera o innych segmentach, również tych najbardziej odległych. Można skorzystać z wcześniej podanych przez administratora tras statycznych lub za pomocą protokołów routingu dynamicznego dokonać wymiany informacji przez same urządzenia sieciowe. Informacja ta musi zawierać opis dostępnych tras i pozwolić routerowi na wybór najlepszej z nich, dla pakietów kierowanych w konkretne miejsce.

Ocena składa się z dwóch elementów: metryki oraz dystansu administratorskiego. Metryka jest oceną kosztu dostępu do wybranej sieci, a sposób jej wyznaczania zależy od protokołu routingu, który ją wylicza. Trudno będzie routerowi odnieść się do metryk dotyczących tej samej sieci, a pochodzących od dwóch różnych protokołów routingu. Jednak w sytuacji, gdy protokół jest ten sam, będą one miały kluczowe znaczenie. Natomiast dystans administratorski jest oceną wiarygodności źródła, z którego pochodzi informacja. Im wyższa wartość z przedziału 0 - 255, tym mniej wiarygodne dla routera źródło informacji. Wartość dystansu administratorskiego nie jest wymieniana pomiędzy routerami, a więc ma znaczenie lokalne. Istnieją jednak pewne domyślne założenia pozwalające na wstępne oszacowanie wiarygodności źródła informacji o sieciach zdalnych - patrz tabela.

Tabela 1.

Źródło informacji o trasie	Dystans administratorski
Bezpośrednio pod-łączone	0
Statyczne	1
Protokół IGRP	100
Protokół OSPF	110
Protokół RIP	120
Nieznane	255

Spośród wszystkich informacji dostępnych dla routera, w tablicy routingu umieszczone będą tylko te, które są dla niego najbardziej wiarygodne. Tablica routowania zawiera rekordy stanowiące odwzorowanie adresów sieci docelowych w adresy interfejsów routerów bezpośrednio połączonych z danym routerem lub znajdujących się w tym samej sieci, mogących dokonać dalszego routowania pakietu odpowiadającej im sieci docelowej.



Rys 1.1

Rysunek 1.1 przedstawia przykładową sieć z trzema podsieciami i dwoma routerami łączącymi podsieci. Każdy router jest urządzeniem należącym jednocześnie do dwóch sieci oznaczonych w tablicy routowania jako directly connected. Trzecia podsieć nie jest bezpośrednio dostępna z routerów, dlatego pakiety skierowana do podsieci muszą być przesyłane na adres interfejsu następnego routera, znajdujących się w odpowiednim rekordzie tablicy routowania [6], [8].

1.2 Dynamiczne wyznaczanie tras pakietów

Jednym z zasadniczych zagadnień w sieciach WAN jest wyznaczanie optymalnych taras pakietów IP na drodze od nadawcy do odbiorcy, zwane także routowaniem. Obecnie wyróżnia się dwie podstawowe grupy protokołów wewnątrzdomenowe IGP (Interior Gateway Protocol) oraz pozadomenowe EGP (Exterior Gateway Protocol). Można powiedzieć, że protokoły te wyraźnie zmierzają w kierunku stworzenia wielopoziomowej struktury identyfikacji adresata

informacji zbliżonej do hierarchicznego systemu numeracji telefonicznej. Odbiega to znacznie od pierwotnego pomysłu na całkowicie płaski system adresacji w Internecie wielkość sieci uniemożliwiła praktyczne stosowanie tej metody.

Protokoły grupy IGP służą do wymiany informacji o topologii sieci w obrębie jednolitego systemu autonomicznego, w jednej domenie administracyjnej tj. zespołu sieci znajdujących się pod wspólną administracją i posługują się dwiema metodami badania sieci.

Protokoły EGP służą do wymiany informacji pomiędzy dostawcami usług internetowych i transportowych sieciowych. Można obrazowo powiedzieć, że śledzą one główne trasy, a identyfikację ostatecznego odbiorcy powierzają routerom systemów autonomicznych wykorzystujących IGP.

Statyczna informacja o trasach, zapisana przez administratora w tablicy routowania jest prostym i nie obciążającym dodatkowo sieci sposobem konfiguracji routerów. Router skonfigurowany statycznie jest pasywny. Wykorzystanie routowania statycznego ma dwie zasadnicze wady:

1. Informacje o wszystkich dostępnych sieciach powinny być ręcznie wpisane przez administratora.
2. Routery nie są zdolne do automatycznej rekonfiguracji w sytuacjach awaryjnych, np. Znalezienia alternatywnej trasy routowania w przypadku awarii łącza, Takie zmiany trzeba nanosić ręcznie.

Wad tych nie posiadają protokoły routowania dynamicznego. Protokoły te generują pewien dodatkowy ruch, ale może on być zaakceptowany ze względu na dużą przepustowość obecnie stosowanych łączy między routerami, celową optymalizację tych protokołów oraz niewielki rozmiar tych pakietów. Protokoły te umożliwiają routerom skuteczną dystrybucję informacji o dostępności odległych (tj. nie podłączonych bezpośrednio) sieci, automatyczną budowę tablic routowania z użyciem tej informacji oraz możliwość automatycznej rekonfiguracji w sytuacjach awaryjnych. Wymiana informacji odbywa się w ściśle określonych odstępach czasowych (np. co 30 sekund), a każda zmiana topologii sieci jest przez routery zauważana i prowadzi do aktualizacji tablic routingu. Ta procedura jest realizowana automatycznie bez udziału użytkowników sieci.
Protokoły routowania dynamicznego dzielą się na dwie zasadnicze klasy:

1. protokoły z wektorem odległości
2. protokoły stanu przyłączeń.

Ze względu na złożoność Internetu nie ma możliwości ręcznej konfiguracji tras. Routery robią to same, a protokoły muszą uwzględniać w przekazywanych informacjach dane, które pozwolą optymalizować i upraszczać trasy. Muszą także umieć wykryć awarie lub przeciążenia na części łączy. Pożądane jest także, aby protokół uniemożliwiał zapętlenie drogi przy automatycznej analizie tras oraz żeby wymieniane informacje o trasach nie powodowały zapchania łączy. Podstawą routowania są tablice, które w jednym routerze mogą być tworzone za pomocą kilku protokołów używanych na różnych łączach fizycznych routera.
Nie ma możliwości zaprojektowania algorytmu wyznaczania trasy optymalnego dla każdego rodzaju fizycznej topologii sieci. Należy więc pamiętać, że niektóre protokoły są efektywniejsze w ściśle określonych strukturach sieci. Analogicznie można stwierdzić, że nie ma najlepszego protokołu każdy ma dobre i złe cechy, zależnie od konkretnej sytuacji [5], [8].

1.3 Protokoły z wektorem odległości

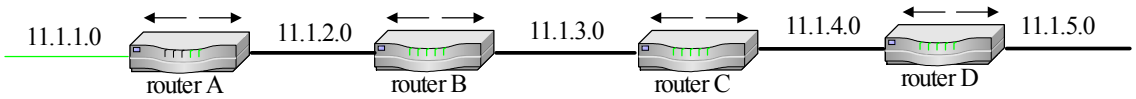
Protokoły z wektorem odległości są oparte na dwóch zasadach:

1. routery dzielą się znaną sobie informacją o całej domenie, tj. o wszystkich sieciach domeny,
2. wymiana informacji odbywa się między sąsiadami, przy czym najczęściej stosowanym typem transmisji jest rozgłoszenie.

Routery używające protokołów wektora odległości regularnie wysyłają kompletną zawartość swojej tabeli routingu do wszystkich routerów sąsiednich, a te z kolei przesyłają informacje dalej . Warto zwrócić uwagę na to, że router ogłasza nie tylko sieci, do których jest bezpośrednio podłączony, ale także te, których nauczył się od sąsiadów - protokoły tej grupy określa się też mianem "routing poprzez plotkowanie". Jako sposób wymiany danych stosowana jest najczęściej komunikacja rozgłoszeniowa (broadcast), chociaż niektóre protokoły wykorzystują multiemisję (multicast).

Nazwa "wektor odległości" pochodzi stąd, iż poszczególne trasy ogłaszane są jako wektory zawierające dwie informacje: odległość oraz kierunek. Odległość opisuje koszt danej trasy i wyrażana jest za pomocą metryki, natomiast kierunek definiowany jest poprzez adres następnego skoku. Protokoły wektora odległości są łatwe w konfiguracji i bardzo dobrze nadają się do zastosowania w małych sieciach. Niestety, jednym z ich podstawowych problemów jest tzw. zbieżność, czyli powolne reagowanie na zmiany zachodzące w topologii sieci, na przykład wyłączenie lub włączenie pewnych segmentów - zerwanie łącza zostaje odzwierciedlone w tabelach routingu poszczególnych routerów dopiero po pewnym czasie. Czas, po którym wszystkie routery mają spójne i uaktualnione tabele routingu nazywany jest czasem zbieżności. Kolejną wadą protokołów wektora odległości jest generowanie dodatkowego ruchu w sieci poprzez cykliczne rozgłaszanie pełnych tabel routingu, nawet wówczas, gdy w topologii sieci nie zachodzą żadne zmiany. Protokoły tej grupy nie są też odporne na powstawanie pętli między routerami (zarówno między bezpośrednimi sąsiadami, jak i pętli rozległych), co skutkuje wzajemnym odsyłaniem sobie pakietów z informacją o tej samej sieci.

Należy zwrócić również uwagę na problem propagacji błędnych informacji. Przykładowy Router A, otrzymujący dane od swojego sąsiada B, musi zakładać, iż są one poprawne, gdyż nie ma żadnego sposobu na ich zweryfikowanie i ewentualne wykrycie błędów w tabeli routingu routera B. To oczywiście oznacza, że router A również będzie przekazywał błędne informacje do swoich pozostałych sąsiadów .



cykle czasowe

t1 ↓

sieć	przez	skok	sieć	przez	skok	sieć	przez	skok	sieć	przez	skok
11.1.1.0	---	0	11.1.2.0	---	0	11.1.3.0	---	0	11.1.4.0	---	0
11.1.2.0	---	0	11.1.3.0	---	0	**11.1.4.0**	---	0	11.1.5.0	---	0

sieć	przez	skok	sieć	przez	skok	sieć	przez	skok	sieć	przez	skok
11.1.1.0	---	0	11.1.2.0	---	0	11.1.3.0	---	0	11.1.4.0	---	0
11.1.2.0	---	0	11.1.3.0	---	0	11.1.4.0	---	0	11.1.5.0	---	0
11.1.3.0	11.1.2.2	1	11.1.1.0	11.1.2.1	1	11.1.2.0	11.1.3.1	1	11.1.3.0	11.1.4.1	1
			11.1.4.0	11.1.3.2	1	11.1.5.0	11.1.4.2	1			

t2 ↓

sieć	przez	skok	sieć	przez	skok	sieć	przez	skok	sieć	przez	skok
11.1.1.0	---	0	11.1.2.0	---	0	11.1.3.0	---	0	11.1.4.0	---	0
11.1.2.0	---	0	11.1.3.0	---	0	11.1.4.0	---	0	11.1.5.0	---	0
11.1.3.0	11.1.2.2	1	11.1.1.0	11.1.2.1	1	11.1.2.0	11.1.3.1	1	11.1.3.0	11.1.4.1	1
11.1.4.0	11.1.2.2	1	11.1.4.0	11.1.3.2	1	11.1.5.0	11.1.4.2	1	11.1.2.0	11.1.4.1	2
			11.1.5.0	11.1.3.2	2	11.1.1.0	11.1.3.1	2			

Rys 1.3

Protokół wektora odległości - informacje o poszczególnych sieciach "propagują" się stopniowo. Na przykład router A dopiero po dwóch cyklach czasowych uzyska informacje o sieci 11.1.4.0.

Ważnym parametrem dla protokołów wektora odległości jest maksymalna rozpiętość sieci, czyli maksymalna dopuszczalna w danym protokole liczba kolejnych routerów (skoków) na ścieżce wiodącej do sieci docelowej. Sieci dostępne przez większą od dozwolonej liczbę skoków oznaczane są jako nieosiągalne. Protokoły wektora odległości pracują zgodnie z algorytmami opracowanymi przez R.E.Bellmana, L.R.Forda i D.R.Fulkersona, a typowymi przedstawicielami tej grupy są RIP oraz IGRP [3], [8].

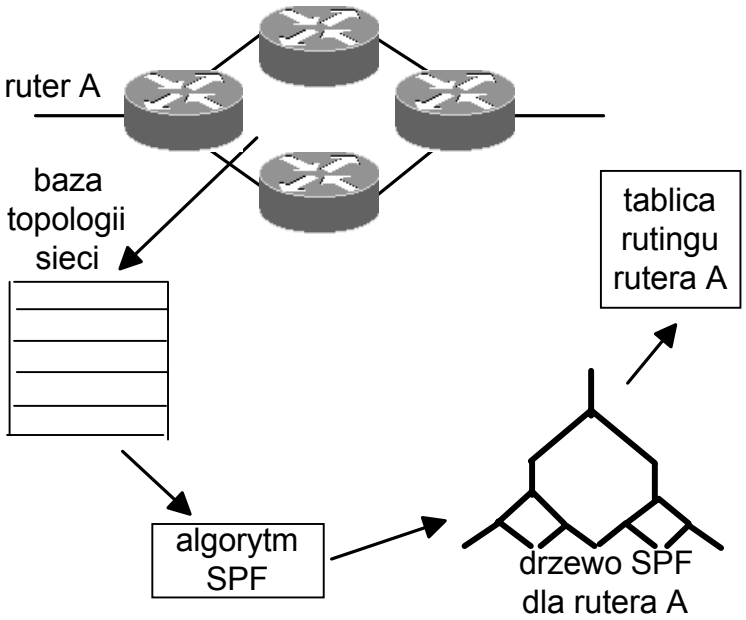
1.4 Protokoły stanu przyłączeń

Routery realizujące protokoły stanu przyłączeń działają w trzech podstawowych, cyklicznie powtarzanych krokach:

1. na bieżąco testują status wszystkich przyłączonych bezpośrednio sieci,
2. wysyłają listę sprawnych przyłączeń do wszystkich pozostałych routerów domeny,
3. na podstawie otrzymanych informacji obliczają najkrótsze ścieżki dostępu do wszystkich sieci domeny.

W protokołach stanu łącza każdy router przechowuje kompletną bazę danych o topologii sieci z informacjami o koszcie pojedynczych ścieżek w obrębie sieci oraz o stanie połączeń. Informacje te kompletowane są poprzez rozsyłanie tzw. pakietów LSA (link-state advertisement) o stanie łączy. Każdy router wysyła informację o bezpośrednio do niego podłączonych sieciach oraz o ich stanie (włączone lub wyłączone). Dane te są następnie rozsyłane od routera do routera, każdy router pośredni zapisuje u siebie kopię pakietów LSA, ale nigdy ich nie zmienia. Po pewnym czasie (czasie zbieżności) każdy router ma identyczną bazę danych o topologii (czyli mapę sieci) i na jej podstawie tworzy drzewo najkrótszych ścieżek SPF (shortest path first) do poszczególnych sieci. Router zawsze umieszcza siebie w centrum (korzeniu) tego drzewa, a ścieżka wybierana jest na podstawie kosztu dotarcia do docelowej sieci - najkrótsza trasa nie musi pokrywać się z trasą o najmniejszej liczbie skoków. Do wyznaczenia drzewa najkrótszych ścieżek stosowany jest algorytm E.W. Dijkstry. Ponieważ każdy router ma iden-

tyczną bazę danych informacji o sieci, protokoły stanu łącza są znacznie bardziej odporne na rozgłaszanie przypadkowych błędów ogłaszane przez sąsiadów niż protokoły wektora odległości. Ponadto drzewo rozpinające sieć pozbawione jest w naturalny sposób rozległych pętli łączących routery.



Rys 1.4
Protokół stanu łącza

Jedną z najważniejszych cech protokołów stanu łącza jest szybkie reagowanie na zmiany w topologii sieci. Po zmianie stanu łącza router generuje nowy pakiet LSA, który rozsyłany jest od routera do routera, a każdy router otrzymujący ten pakiet musi przeliczyć od nowa drzewo najkrótszych ścieżek i na jego podstawie zaktualizować tabelę routingu.

Protokoły stanu łącza nazywane są też protokołami "cichymi", ponieważ w przeciwieństwie do protokołów wektora odległości nie rozsyłają cyklicznych ogłoszeń, a dodatkowy ruch generują tylko przy zmianie stanu łącza. Ze względu na sposób działania i swoje cechy protokoły stanu łącza przeznaczone są do obsługi znacznie większych sieci niż protokoły wektora odległości.

Do wad protokołów stanu łącza zaliczyć można zwiększone zapotrzebowanie na pasmo transmisji w początkowej fazie ich działania (zanim "ucichną"), gdy routery rozsyłają między sobą pakiety LSA. Dodatkowo ze względu na złożoność obliczeń drzewa SPF, protokoły stanu łącza mają zwiększone wymagania dotyczące procesora i pamięci RAM routera (zwłaszcza przy większych sieciach). Typowym przedstawicielem tej grupy protokołów jest OSPF (Open Shortest Path First) [3], [8].

1.5 Właściwości algorytmów routingu

Efektywny protokół routingu (algorytm) powinien spełniać następujące, wymagania projektowe:

- poprawność
- prostota
- stabilność
- konwergencja

Poprawność to ta właściwość algorytmu routingu, która daje gwarancję, że wskazana, optymalna droga przesyłania informacji jest rzeczywiście optymalna. Należy zwrócić uwagę na fakt, iż wymagana "optymalność" jest względna i zależy od konkretnego algorytmu i tych parametrów sieci komputerowej, które dany algorytm wykorzystuje do określania metryki.

Prostota algorytmu routingu jest wymaganiem wynikającym z konieczności bardzo szybkiej obsługi pakietów przesyłanych w sieci komputerowej. Router nie może spowalniać pracy sieci w wyniku dokonywania skomplikowanych obliczeń związanych z algorytmem routingu i wyborem dalszej trasy pakietu w sieci.

Stabilność algorytmu routingu to jego odporność na przypadkowe, krótkotrwałe zmiany warunków pracy sieci. Natychmiastowa reakcja na zmianę parametrów pracy sieci, powodowałaby dezorganizację jej pracy, nieuzasadnioną z punktu widzenia czasu trwania zmiany tych parametrów. Jako przykład niech służy sytuacja wynikająca z rozpięcia sieci lokalnej na kilka sekund w celu przyłączenia do sieci nowego komputera. Kilkusekundowa przerwa pracy sieci nie może powodować natychmiastowej zmiany tablicy routingu polegającej na zamarkowaniu niepracującej sieci jako nieosiągalnej dla komputerów innych sieci, ponieważ o takiej zmianie poprzez propagacje informacji dowiedziałyby się inne routery, oznaczając rozważaną sieć także jako nieosiagalną. Taka propagacja informacji o niedostępności sieci, a zaraz potem o jej dostepności, byłaby dużo dłuższa niż rzeczywista przerwa w pracy sieci lokalnej.

Konwergencja to zdolność routerów do szybkiego, jednolitego (dokonanego przez wszystkie routery) uzgodnienia optymalnych tras przesyłania informacji w sieci. Niedocenienie wagi szybkości tego procesu może spowodować, iż pojawią się drogi alternatywne (routing loops) powodujące nieprawidłową pracę sieci [7].

2. OSPF (OPEN SHORTEST PATH FIRST)

Protokół OSPF (*Open Shortest Path First*) jest obecnie promowany przez organizacje sieciowe jako podstawowy protokół routowania dynamicznego dla sieci TCP/IP. Protokół OSPF opracowała grupa robocza IGP (*Interior Gateway Protocol*) należąca do IETF i został zdefiniowany w dokumencie RFC 1247, jego druga wersja opisana jest w RFC 1583. OSPF jest protokołem otwartym, co oznacza, że jego specyfikacja jest ogólnie dostępna. Protokół OSPF jest protokołem routingującym klasy link-state, wykorzystującym algorytm SPF (*Dijkstry*), który jest używany przez routery do konstrukcji optymalnego drzewa połączeń z innymi sieciami. Protokół OSPF został zaprojektowany w celu zwiększenia efektywności przetwarzania w sie-

ciach pracujących z protokołem IP a zatem uwzględnia adresowanie zmienną długością maski (VLSM), uwierzytelnia źródła informacji o trasach oraz szybko uaktualnia drogi routingu. Jest udoskonaleniem protokołu RIP, ponieważ pozwala na wybór ścieżki na podstawie wieloparametrowego kryterium kosztu określanego jako routing najniższego kosztu (*least-cost routing*). OSPF umożliwia m.in. strukturalizację i hierarchizację sieci w ramach domeny poprzez wprowadzenie różnych typów tzw. obszarów. Protokół ten pozwala również na sprawdzenie autentyczności pakietów oraz używa transmisji grupowej do rozsyłania pakietów.

Realizacje OSPF można podzielić na trzy fazy:

1. Faza budowy tablic LSA.

   Routery wysyłają pakiety typu „hello" do wszystkich bezpośrednich sąsiadów. Pakiety te służą do „przedstawienia się" sąsiadom, tj. do poinformowania o swojej obecności oraz stanie swoich łączy. Pakiety „hello" muszą zostać potwierdzone. Na podstawie otrzymanych od sąsiadów potwierdzeń routery budują swoje tablice LSA.
2. Faza dzielenia się informacją.

   Routery zgłaszają pakiety LSA na wszystkich interfejsach i dalej w całej domenie, używając rozsyłania grupowego. Pakiety te zawierają listę przyłączeń danego routera. OSPF dzieli przyłączenia na 7 klas, w zależności od tego, do jakich elementów struktury domeny prowadzą. Rozgłaszanie LSA jest protokołem niezawodnym, tzn. pakiety LSA muszą zostać potwierdzone.
3. Faza budowy tablic routowania.

   Obliczanie najbardziej optymalnych tras routowania odbywa się za pomocą algorytmu Dijkstry. OSPF dopuszcza używanie kilku algorytmów wyznaczania metryk, określających koszt połączenia. Lista wartości wraz z ich typami jest przesyłana w pakiecie LSA.

Nowo włączone do sieci routery po potwierdzeniu pakiety „hello" otrzymują specjalną wiadomość, zawierającą tablicę stanów przyłączeń, by mogły szybciej zbudować pełną tablicę routowania [1], [5], [8].

2.1 Hierarchia routingu

W odróżnieniu od protokołu RIP protokół OSPF może działać w układzie hierarchicznym. Największą jednostką w hierarchii jest system autonomiczny AS (*Autonomous System*), który jest zbiorem sieci pod wspólną administracją, a które mają wspólną strategię routingu. OSPF jest protokołem routingu wewnętrznego systemów AS (wewnętrzna brama), może jednak przyjmować i wysyłać trasy do innych systemów AS.

System AS można podzielić na pewną liczbę obszarów (*areas*), które są grupami sąsiednich sieci i przyłączonych hostów. Poszczególne obszary sprzęgają routery graniczne obszaru (*area border routers*). Router graniczny utrzymuje oddzielną dla każdego obszaru bazę danych o topologii (*topological database*).

Baza danych o topologii jest obrazem sieci wyrażonym w powiązaniach między routerami. Zawiera zbiór zgłoszeń LSA pochodzących od wszystkich routerów w danym obszarze. Ponieważ routery w jednym obszarze otrzymują tę samą informacje, to ich bazy dot. topologii są identyczne.

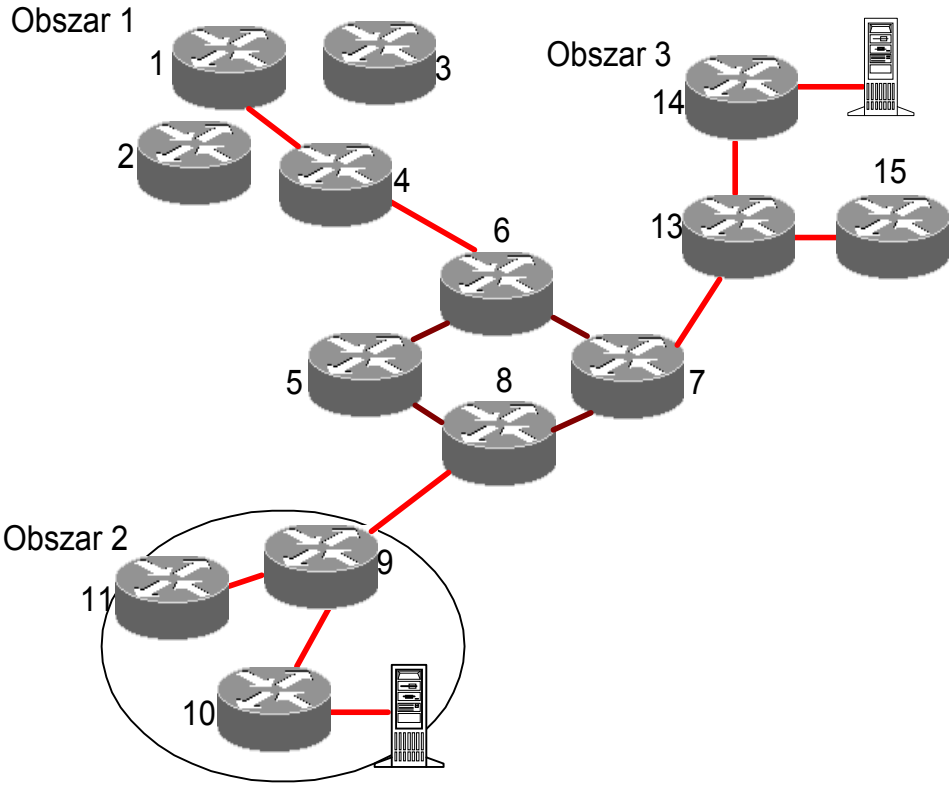
Topologia obszaru jest niewidoczna dla urządzeń znajdujących się poza nim. Podział systemów AS na obszary przyczynia się do zmniejszenia ruchu związanego z routingiem.

Wydzielenie obszarów stworzyło dwa typy routingu OSPF: wewnętrzny, jeśli źródło i miejsce przeznaczenia znajdują się w tym samym obszarze, oraz zewnętrzny, jeśli znajdują się one w

dwu różnych obszarach.

Za dystrybucję informacji pomiędzy obszarami jest odpowiedzialna sieć szkieletowa OSPF (*OSPF backbone*). Składa się ona ze wszystkich routerów granicznych, linii, które nie łączą routerów wewnątrz obszaru, oraz przyłączonych do nich routerów.

Szkielet jest również obszarem OSPF, stąd wynika, że routery szkieletu używają takich samych procedur i algorytmów do utrzymania informacji routingu w szkielecie, jak każdy inny router w obszarach sprzężonych ze szkieletem. Topologia szkieletu jest niewidoczna dla routerów wewnątrz obszarów, ponieważ nie należy do topologii obszarów.



Rys 2.1
System autonomiczny składający się z wielu obszarów

Szkielet tworzą routery 5,6,7 i 8. Jeśli host A zlokalizowany w obszarze 2 chce wysłać pakiet do hosta B w obszarze 3, to pakiet jest przekazywany kolejno do routera 10 i9 (routery wewnętrzne obszaru 2), następnie do routerów 8 i 7 (routery szkieletu) i do routera 13 w obszarze 3. Następnie pakiet jest wysyłany do routera 14 i stąd do hosta B.

Routery brzegowe systemów autonomicznych pracujące z protokołem OSPF dowiadują się o zewnętrznych trasach przez zewnętrzne protokoły bramowe, takie jak protokół EGP (*Exterior Gateway Protocol*) lub protokół BGP (*Border Gateway Protocol*) [2].

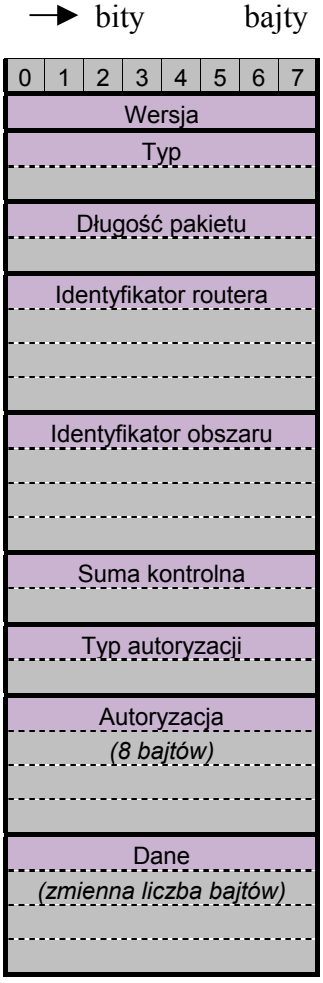
2.2 Dodatkowe właściwości protokołu OSPF

Wśród dodatkowych właściwości protokołu OSPF można wymienić: jednakowy koszt (*equal cost*), routing wielościeżkowy (*multipath routing*) i routing oparty na żądaniach TOS (*type-of-service*) wyższej warstwy. Routing oparty na żądaniach TOS wspomaga te protokoły warstwy wyższej, które mogą określić szczególne typy usług. Na przykład aplikacja może określić pewne dane jako pilne. Jeśli protokół OSPF dysponuje szybkimi łączami, to może ich użyć do przekazywania pilnych danych.

Protokół OSPF może posługiwać się jedną lub wieloma miarami. W przypadku użycia jednej miary jest ona przyjmowana arbitralnie i nie zachodzi potrzeba obsługi TOS. W przypadku użycia większej liczby miar TOS jest wspomagany oddzielnie każdą z nich i związanymi z nimi tablicami routingu. Ponieważ TOS protokołu IP zawiera trzy bity - opóźnienie (*delay*), przepustowość (*throughput*) i niezawodność (*reliability*) - to do dyspozycji jest osiem kombinacji. Jeśli przykładowo trzy bity TOS określają małe opóźnienie, niską przepustowość i wysoką niezawodność, to protokół OSPF wylicza trasy do wszystkich miejsc przeznaczenia opierając się na tym wyznaczniku TOS.

Do zgłoszenia każdego miejsca przeznaczenia są dołączane maski podsieci IP, dające możliwość użycia opcji zmiennej długości maski podsieci (*variable-length subnet mask*). Dysponując tą opcją, sieć IP można podzielić na wiele podsieci o różnych rozmiarach, dzięki czemu administrator może bardzo elastycznie konfigurować sieć [2].

Format pakietu protokołu OSPF



Wszystkie pakiety protokołu OSPF mają 24 bitowy nagłówek.

Numer wersji (*Version Number*) – identyfikuje użytą wersję protokołu OSPF.

Typ (*Type*) – określa typ pakietu protokołu OSPF. Może to być jeden z następujących typów:

- *Hello*: ustala i utrzymuje powiązania z sąsiadem;
- *Database Description*: opisuje zawartość bazy danych o topologii. Komunikaty te są wymieniane po zainicjowaniu powiązania miedzy sprzężonymi routerami;
- *Link-state Request:* żadanie od sąsiada przekazania fragmentu bazy danych o topologii. Komunikat ten jest wymieniany w przypadku, gdy router zorientuje się, że część jego bazy danych o topologii straciła aktualność;
- *Link-state Update*: odpowiedź na żądanie 1 Link-state Request. Pakiet jest używany do regularnego rozsyłania zgłoszeń LSA. Jeden pakiet Link-state Update może zawierać kilka tego typu zgłoszeń;
- *Link-state Acknowledgment:* potwierdzenie pakietu Link-state Update.

Długość pakietu (*Packet Length*) – wskazuje całkowitą długość komunikatu w bajtach.

Identyfikator routera (*Router ID)* – identyfikuje źródło pakietu.

Identyfikator obszaru (*Area ID*) – identyfikuje obszar, do którego należy pakiet. Wszystkie pakiety OSPF są skojarzone z jednym obszarem.

Suma kontrolna (*Checksum*) – kontroluje zawartość pakietu w celu wykrycia ewentualnych przekłamań.

Typ autoryzacji (*Authentication Type*) – wskazuje na typ autoryzacji. Wszystkie informacje wymieniane za pomocą protokołu OSPF są autoryzowane.

Autoryzacja (*Authentication*) – zawiera informację autoryzującą.

Dane (*Data*) – zawiera odbudowaną informację dla warstwy wyższej [2].

2.3 Konfiguracja OSPF

1. Udostępnienie OSPF dla danego routera:

Protokół OSPF włączany jest głównym poleceniem konfiguracyjnym:

router(config)# router ospf id-procesu

2. id-procesu jest wewnętrznym numerem, stosowanym dla pojedynczego routera, w którym uruchomionych jest wiele procesów OSPF. Numer procesu nie musi pasować do numerów procesów w innych routerach. Nie zaleca się uruchamianie wielu procesów OSPF w tym samym routerze, ponieważ tworzy to wiele baz danych i powoduje dodatkowe obciążenie sieci.

3. Zidentyfikuj sieci IP w routerze. Dla każdej sieci należy określić strefę do której ta sieć należy. Wartość sieci może się zmieniać, w zależności od tego, czy adres sieci jest obsługiwany przez router, czy też przez specjalnie skonfigurowany interfejs. Router może rozróżniać adresy poprzez porównanie maski wzorca. Służy do tego następująca funkcja:

Router(config-router)# network adres maska-wzorca area id-strefy

Tabela 2.

Polecenie network area	Opis
adres	Może być adresem sieci, podsieci lub interfejsu. Dostarcza routerowi informacji o łączach, które mają być rozgłaszane.
maska-wzorca	Maska używana do odczytywania adresu. Maska ma wzorcowe bity, gdzie 0 oznacza dopasowanie, a 1 – „bez znaczenia”. Na przykład 0.0.255.255 wskazuje, że dopasowanie zachodzi dla dwóch pierwszych bitów. Do określenia adresu interfejsu stosuje się maskę 0.0.0.0
id-sterfy	Wskazuje strefę związaną z danym adresem. Może być liczbą lub mieć postać A.B.C.D, co przypomina adres IP. Numer IP pojedyńczej sterfy powinien być równy 0

Modyfikacja kosztu łącza – unieważnia koszt przypisany interfejsowi OSPF:

Router(config-if)# ip ospf cost *koszt*

koszt – liczba z zakresu od 1 do 65535, która wskazuje metrykę przyporządkowaną do interfejsu. Koszt ścieżki jest sumą kosztów przypisanych do wszystkich interfejsów, które przekazują ruch do miejsca przeznaczenia.
Koszt ścieżki obliczany jest ze wzoru 10^8 / *pasmo*. Stosując ten wzór można wyznaczyć koszt przykładowych łączy:

- łącze szeregowe 56kbit/s, domyślny koszt 1785
- łącze szeregowe T1 (1,544Mbit/s), domyślny koszt wynosi 128
- Ethernet, domyślny koszt wynosi 10
- Łącze Token Ring 16Mbit/s, domyślny koszt wynosi 6

3. PODSTAWOWA KONFIGURACJA ROUTERA CISCO

3.1 Uruchomienie routera

Pierwszą czynnością jest ustanowienie połączenia z routerem poprzez port konsoli. Każdy router Cisco wyposażony jest w jeden taki port (interfejs RS-232 lub RJ-45), do którego podłączyć można terminal znakowy lub komputer z emulatorem terminala (np. HyperTerminal w systemach Windows). Za pomocą terminala można przeprowadzić proces konfiguracji routera. Poprawna komunikacja z routerem wymaga ustawienia odpowiednich parametrów transmisyjnych terminala - zwykle stosuje się: terminal typu VT100, prędkość 9600 (chociaż w rejestr routera można wpisać inną wartość), 8 bitów danych, 1 bit stopu, transmisję bez parzystości.

Po włączeniu routera w oknie terminala pojawi się zestaw komunikatów związanych ze startem routera. Proces uruchamiania routera składa się z kilku etapów i jest inicjowany przez program rozruchowy (bootstrap), znajdujący się w pamięci ROM. Po przeprowadzeniu testów diagnostycznych sprzętu w ramach procedury POST, w której sprawdza się m.in. działanie procesora, pamięci i interfejsów, poszukiwany jest i ładowany obraz systemu operacyjnego IOS - zgodnie z ustawieniami w rejestrze routera oraz poleceniami zawartymi w skrypcie konfiguracyjnym.

Większość routerów zawiera pamięć Flash. Jest to pamięć typu EEPROM, jej zawartość może być wielokrotnie usuwana i zapisywana ponownie. Zawartość pamięci Flash nie ginie po wyłączeniu routera, dlatego przeznaczona jest przede wszystkim do przechowywania wielu kopii systemu operacyjnego IOS. Zwykle początkowo w pamięci Flash znajduje się tylko jeden obraz systemu operacyjnego (zwany domyślnym plikiem systemu operacyjnego) i właśnie on zostanie załadowany po pierwszym włączeniu routera. Pamiętać jednak należy, że niektóre routery (Cisco 2500, 4000, 4500) przechowują minimalną wersję systemu operacyjnego bezpośrednio w pamięci ROM. Inne, np. routery serii 7000 i 7500, wczytują pełen obraz systemu operacyjnego z pamięci ROM.

Po załadowaniu systemu operacyjnego poszukiwany jest skrypt konfiguracyjny, zawierający parametry definiujące pracę routera (np. hasło dla trybu uprzywilejowanego) oraz poszczególnych jego części (np. interfejsów). Skrypt konfiguracyjny zapisywany jest w nieulotnej pamięci NVRAM, skąd przy każdym ponownym uruchomieniu routera może być odczytany i załadowany do pamięci operacyjnej RAM. Aktualna konfiguracja oraz wszelkie dokonywane w niej zmiany przechowywane są tylko w pamięci RAM, aby więc utrwalić wprowadzane przez administratora modyfikacje, należy ręcznie zapamiętać tę konfigurację w pamięci NVRAM jako konfigurację startową. Przy pierwszym uruchomieniu routera skrypt konfiguracyjny w pamięci NVRAM nie istnieje, co powoduje automatyczne uruchomienie dialogu konfiguracyjnego [4].

3.2 Dialog konfiguracyjny

Dialog konfiguracyjny to interaktywna sekwencja pytań i odpowiedzi, pozwalających utworzyć pierwszą, bazową konfigurację routera. Dialog wywoływany jest również w przypadku usunięcia zawartości pamięci NVRAM lub po uruchomieniu routera w specjalnym trybie naprawczym z pominięciem odczytywania pamięci NVRAM. Pracujący w trybie uprzywilejowanym może także w dowolnej chwili uruchomić dialog konfiguracyjny poleceniem SETUP. Zbiór parametrów, jakie można ustawić bezpośrednio w dialogu konfiguracyjnym, zależy od modelu routera i wersji systemu operacyjnego.

Przykładowy dialog konfiguracyjny dla routera 2600 z systemem 11.3.

Dialog konfiguracyjny – ustawienie haseł dostępu

```
Configuring global parameters:

Enter host name [Router]: C2600

The enable secret is a password used

  to protect access to privileged EXEC

  and configuration modes. This password,

  after entered, becomes encrypted

  in the configuration.

Enter enable secret: haslo1

The enable password is used when you

  do not specify an enable secret password,

  with some older software versions, and

  some boot images.

Enter enable password: haslo2

The virtual terminal password is used

  to protect access to the router over

  a network interface.

Enter virtual terminal password: haslo3
```

Po wyświetleniu pierwszego pytania wciskamy klawisz Enter, aby wejść do trybu interaktywnego. Niewątpliwie warto wyświetlić na ekranie podsumowanie dotyczące aktualnej konfiguracji interfejsów, w tym celu w odpowiedzi na drugie pytanie wciskamy ponownie Enter, zatwierdzając proponowaną domyślną wartość podaną w nawiasach kwadratowych. W pierwszej kolumnie wyświetlonego zestawienia sprawdzić można, jak oznaczane są w danym routerze poszczególne interfejsy. Nazwa interfejsu składa się z typu (np. Ethernet lub Serial) oraz numeru. W routerach niemodularnych (poniżej rodziny 2600) numer interfejsu jest pojedynczą liczbą (np. Serial 0, Ethernet 1), natomiast w routerach modularnych, które mogą być rozbudowywane o kolejne karty interfejsów, stosuje się zestaw dwu liczb w notacji nr_karty/nr_portu (np. Serial 0/1 oznacza drugi port szeregowy na pierwszej karcie). W routerach serii 7000 i 7500, wyposażonych w złącza (slot) dla kart VIP, oznaczenie interfejsu złożone będzie z trzech liczb, zgodnie z konwencją nr_karty_VIP/nr_karty/nr_portu (np. Ethernet 1/0/1).

Następne kolumny podsumowania dotyczącego interfejsów zawierają informacje o przypisanych adresach IP, aktualnym statusie pracy interfejsu i wybranym protokole warstwy łącza danych. Domyślnie wszystkie interfejsy są wyłączone (status oznaczony jako down), nie mają adresów IP ani określonego protokołu warstwy łącza danych. W kolejnych etapach dialogu konfiguracyjnego zdefiniować należy parametry globalne, w tym logiczną nazwę urządzenia wykorzystywaną w różnych procesach identyfikacyjnych oraz trzy hasła dostępowe wykorzystywane na routerze.

Pierwsze hasło, oznaczone jako **enable secret**, chroni dostępu do trybu uprzywilejowanego, w którym administrator może uruchamiać wszystkie polecenia, a także przeprowadzać dowolne zmiany konfiguracyjne. Hasło enable secret przechowywane jest w postaci zaszyfrowanej. Aby zapewnić zgodność z wcześniejszymi wersjami systemu operacyjnego, w dialogu konfiguracyjnym pozostawiono możliwość zdefiniowania również hasła **enable password.** Hasło to także chroni dostępu do trybu uprzywilejowanego, ale jest wykorzystywane tylko w starszych wersjach systemu oraz wtedy, gdy hasło enable secret nie jest zdefiniowane. Ponieważ enable password przechowywane jest w postaci niezaszyfrowanej, zalecane jest stosowanie enable secret. Trzecim wymagane hasło chroni dostępu do routera poprzez linie terminali wirtualnych VTY, zwykle są to połączenia z wykorzystaniem protokołu telnet. Standardowo router udostępnia pięć linii wirtualnych VTY. Domyślnie dostęp do routera poprzez linię konsoli nie jest zabezpieczany żadnym hasłem.

Po określeniu haseł, w dialogu konfiguracyjnym pojawia się możliwość zdefiniowania społeczności protokołu SNMP, w której pracować będzie router. Domyślnie proponowana jest społeczność Public i początkowo można tę nazwę pozostawić bez zmiany. Właściwe zdefiniowanie społeczności może mieć duże znaczenie dla pracujących w trybie graficznym programów do zdalnego zarządzania routerem, które działanie opierają na protokole SNMP. Kolejne pytania dialogu konfiguracyjnego dotyczą protokołów routingu dynamicznego, takich jak RIP czy IGRP. Można początkowo pozostawić proponowane, domyślne ustawienia lub wyłączyć routing dynamiczny.

Ostatnia sekcja dialogu konfiguracyjnego pozwala w pętli zdefiniować parametry dotyczące poszczególnych interfejsów routera, np.: adres IP czy maska podsieci. Po udzieleniu odpowiedzi na wszystkie pytania pojawia się możliwość przejrzenia zdefiniowanych ustawień oraz zapamiętania konfiguracji startowej w pamięci NVRAM. Odpowiednia opcja w menu wyboru pozwala opuścić dialog konfiguracyjny bez zapamiętywania zmian. Z trybu dialogu można także wyjść w dowolnej chwili, wybierając kombinację Ctrl_C [4].

Tabela 3.

Skróty klawiszowe	
Kombinacja	**Działanie**
strzałka w górę lub Ctrl_P	poprzednie polecenie w historii poleceń
strzałka w dół lub Ctrl_N	następne polecenie w historii poleceń
Ctrl_A	przejście na początek linii
Ctrl_E	przejście na koniec linii
Tab lub Ctrl_I	dokończenie polecenia
Ctrl_C	wyjście z trybu interaktywnego
Ctrl_Z (polecenie End)	wyjście z trybu konfiguracyjnego
Ctrl_^ (Ctrl_Shift_6)	przerwanie wykonywanego polecenia
Ctrl_Shift_6+x	chwilowe opuszczenie zdalnej sesji telnetowej
? (polecenie Help)	system pomocy
Enter	następny wiersz w trybie "-More-"
Odstęp	następna strona w trybie "-More-"
Q	wyjście z trybu "-More-"
Ctrl_Break	wywołanie z konsoli trybu monitora pamięci ROM

3.3 Tryby pracy i zarządzanie skryptem konfiguracyjnym

Po zapamiętaniu konfiguracji startowej oraz po ponownym uruchomieniu routera administrator podłączony do routera poprzez port konsoli automatycznie uzyskuje dostęp do trybu wykonywania poleceń, zwanego trybem EXEC. Tryb EXEC pozwala na pracę na szesnastu poziomach uprzywilejowania, choć zwykle wykorzystywane są tylko dwa: poziom użytkownika (poziom 1) oraz poziom uprzywilejowany (poziom 15). Poziomem domyślnym - oznaczanym przez znak zachęty zakończony symbolem ">" - jest poziom użytkownika, na którym dostępne są tylko niektóre polecenia sprawdzające status routera oraz definiujące pracę terminala.

Listę dostępnych poleceń w dowolnym trybie pracy routera wyświetlić można przez wciśnięcie znaku "?". W trakcie wpisywania poleceń o złożonej składni wciśnięty znak "?" przywołuje kontekstową pomoc z informacjami o kolejnych parametrach czy słowach kluczowych wymaganych w danym poleceniu. Bardzo użyteczną cechą systemu operacyjnego jest rozróżnianie poleceń na podstawie wpisanych początkowych znaków nazwy. Wpisana część nazwy komendy musi jednoznacznie identyfikować polecenie, np. słowo en oznaczać będzie w praktyce polecenie enable. System operacyjny pamięta również historię ostatnio wykonywanych poleceń, po której w większości terminali poruszać można się za pomocą klawiszy kierunkowych w górę i w dół.

Pełen zestaw poleceń łącznie z trybem konfiguracyjnym przypisany jest do poziomu uprzywilejowanego oznaczanego znakiem zachęty zakończonym symbolem "#" (poziomy 2 - 14 też oznaczane są symbolem "#"). Aby przejść na poziom 15, należy wykonać polecenie enable, pamiętając o tym, że dostęp do poziomu uprzywilejowanego chroniony jest hasłem enable secret, zdefiniowanym w dialogu konfiguracyjnym (jeżeli zdefiniowane jest hasło enable secret, nie można wykorzystać hasła enable password do przejścia na poziom 15). Powrót na poziom domyślny (poziom 1) realizowany jest poleceniem disable.

Tabela 4.

Tryby pracy routera	
Tryb pracy	**Działanie**
Tryb użytkownika C2600>	Ograniczony zestaw poleceń "nieniszczących"; definiowanie ustawień terminala; wyświetlanie statusu routera.
Tryb uprzywilejowany C2600#	Pełen zestaw poleceń; tryb konfiguracyjny; śledzenie pracy routera poprzez polecenie debug.
Tryb konfiguracyjny C2600(config)#	Globalne i główne polecenia konfiguracyjne; wywoływany z trybu uprzywilejowanego.
Tryb konfiguracyjny procesu C2600	Konfiguracja specyficznego procesu lub interfejsu routera;
(config-proces)#	realizacja podpoleceń; wywoływany z trybu konfiguracyjnego.
Dialog konfiguracyjny	Konfiguracja routera w trybie ineraktywnym; wywoływany poleceniem setup lub automatycznie przy braku konfiguracji startowej.
Monitor pamięci ROM rommon>	Procesy naprawcze (hasła lub pamięci Flash); modyfikowanie rejestru; wywoływany ręcznie odpowiednią kombinacją (zwykle Ctrl_Break) lub automatycznie przy braku poprawnego systemu operacyjnego.

Ponieważ interaktywny dialog konfiguracyjny nie pozwala na zdefiniowanie wszystkich parametrów pracy routera, administrator będzie musiał dokończyć proces konfiguracji ręcznie z wykorzystaniem specjalnego trybu pracy routera, zwanego trybem konfiguracyjnym. Tryb ten (podobnie jak tryb śledzenia, wywoływany poleceniem debug) zarezerwowany jest dla poziomu uprzywilejowanego, a wchodzi się do niego komendą configure - pozwala ona skonfigurować router trzema różnymi metodami:

- Terminal (metoda domyślna) - konfiguracja ręczna poprzez wykonywanie poszczególnych poleceń z poziomu terminala,
- Memory - wczytanie pełnej konfiguracji z pamięci NVRAM (konfiguracja startowa) do pamięci RAM,
- Network - wczytanie skryptu konfiguracyjnego z serwera sieciowego TFTP.

Po wejściu do trybu konfiguracyjnego z opcją domyślną zmienia się odpowiednio znak zachęty, zgodnie z notacją: Nazwa_routera(config)#. Wyróżniamy trzy rodzaje poleceń konfiguracyjnych: globalne, główne i podpolecenia. Komendy globalne, zapisywane w pojedynczej linii, definiują parametry dotyczące pracy routera jako całości.

Przykładami trzech poleceń globalnych, definiującymi odpowiednio: logiczną nazwę routera, hasło chroniące dostęp do trybu uprzywilejowanego (przechowywane w postaci zaszyfrowanej) i routing dla protokołu IP, są:

```
Router2600(config)#hostname C2600

C2600(config)#enable secret password

C2600(config)#ip routing
```

Polecenia główne nie definiują bezpośrednio żadnych parametrów routera, lecz wyróżniają konkretny proces lub interfejs, który ma podlegać dalszej konfiguracji. Dostępnych jest ponad 17 specyficznych trybów konfiguracyjnych, wybieranych poleceniami głównymi. Poniższe dwa przykładowe polecenia główne wybierają odpowiednio interfejs Ethernet 0/1 oraz protokół routingu dynamicznego IGRP. Wykonanie polecenia głównego, poza zmianą znaku zachęty wskazującego wybrany proces, nie powoduje praktycznych zmian w konfiguracji:

```
C2600(config)#interface Ethernet 0/1

C2600(config-if)#

C2600(config)#router IGRP 10

C2600(config-router)#
```

Właściwą konfigurację procesu czy interfejsu wybranego poleceniem głównym przeprowadza się, podając w kolejnych liniach podpolecenia. Polecenie główne musi mieć przynajmniej jedno podpolecenie. Listę specyficznych dla danego trybu podpoleceń można wyświetlić, wciskając znak "?". Na przykład podpolecenie definiujące tekstowy opis dla interfejsu Ethernet 0/1 wygląda następująco:

```
C2600(config)#interface Ethernet 0/1

C2600(config-if)#description Drugi segment sieci lokalnej
```

Zmiany przeprowadzane w trybie konfiguracyjnym dotyczą zawsze konfiguracji aktualnej, przechowywanej w pamięci RAM. Aby zmiany te utrwalić, należy nagrać konfigurację aktual-

ną w pamięci nieulotnej NVRAM jako konfigurację startową. W tym celu wykonujemy polecenie:

```
C2600#copy running-config startup-config
```

Zarówno konfigurację aktualną, jak i startową można w dowolnej chwili wyświetlić na ekranie za pomocą odpowiedniej składni polecenia show. W poniższych przykładach wyświetlana jest konfiguracja aktualna i startowa, zwana też czasami konfiguracją zapasową. Warto zwrócić uwagę na skrótowy zapis w drugim przykładzie [4]:

```
C2600#show running-config

C2600#sh start
```

Tabela 5.

Rodzaje pamięci routera CISCO	
Pamięć	**Zawartość**
ROM	Inicjujący program ładujący (bootstrap) odpowiedzialny za znalezienie i wczytanie systemu operacyjnego oraz program monitora wykorzystywany w procedurach naprawczych. Pamięć ROM może zawierać także minimalny lub kompletny system operacyjny.
RAM	Aktualna konfiguracja, zwykle obraz systemu operacyjnego oraz inne dynamiczne struktury związane z bieżącą pracą routera.
NVRAM	Konfiguracja startowa (zapasowa).
Flash	Obrazy systemu operacyjnego.

3.4 Konfigurowanie interfejsów

Jednym z pierwszych zadań konfiguracyjnych, jakie wykonać musi administrator nowego routera, będzie właściwe zdefiniowanie parametrów komunikacyjnych dla poszczególnych interfejsów - zarówno tych dotyczących segmentów sieci lokalnej, jak i interfejsów szeregowych, wykorzystywanych najczęściej do połączeń w sieci WAN. Dla interfejsów sieci LAN, takich jak Ethernet, zwykle wystarczające jest zdefiniowanie parametrów dotyczących adresowania w protokole warstwy sieciowej (np. IP) oraz odwołanie domyślnie włączonego polecenia shutdown, które blokuje pracę interfejsu. Czynności te mogą być niepotrzebne, jeśli interfejs skonfigurowano z poziomu dialogu konfiguracyjnego.

Poniższa sekwencja poleceń pokazuje wywołanie trybu konfiguracyjnego, wybór właściwego interfejsu, przypisanie adresu IP i maski podsieci do interfejsu Ethernet 0/0 oraz wyłączenie polecenia shutdown blokującego interfejs. Na przykładzie polecenia shutdown warto zwrócić uwagę na sposób odwoływania poleceń przez wykorzystanie komendy no, dopisywanej na początku oryginalnej linii.

```
C2600#configure terminal

C2600(config)#interface Ethernet 0/0

C2600(config-if)#ip address 131.108.1.1 255.255.255.0

C2600(config-if)#no shutdown
```

W niektórych sytuacjach może okazać się konieczne przypisanie do jednego interfejsu więcej niż jednego adresu IP. Dzieje się tak na przykład wtedy, gdy router obsługuje kilka wirtualnych sieci IP w jednym segmencie fizycznym. Polecenie dodające do interfejsu kolejny adres IP (drugi, trzeci itd.) ma składnię:

```
C2600(config-if)#ip address 212.1.1.1 255.255.255.0 secondary
```

W przypadku interfejsów szeregowych konfiguracja jest bardziej złożona, bowiem oprócz parametrów warstwy sieciowej (adres IP czy maska podsieci) określić należy również ustawienia dla warstwy łącza danych oraz warstwy fizycznej .

W przypadku komunikacji synchronicznej, typu punkt-punkt z wykorzystaniem interfejsów szeregowych, jedno urządzenie w parze pełni rolę urządzenia biernego typu DTE, zaś drugie jest urządzeniem aktywnym DCE, definiującym parametry transmisyjne, np. parametr zegara transmisji. W typowej sytuacji, gdy router podłącza się do sieci WAN, rolę DCE pełni urządzenie brzegowe dostawcy, a DTE - interfejs szeregowy routera oraz odwołanie domyślnie włączonego polecenia shutdown, które blokuje pracę interfejsu. Czynności te mogą być niepotrzebne, jeśli interfejs skonfigurowano z poziomu dialogu konfiguracyjnego. W warstwie łącza danych, jako typ hermetyzacji (encapsulation) dla przesyłanych danych, wybierany jest automatycznie i domyślnie protokół HDLC. W zależności od potrzeb protokół ten można zmienić.

Jeżeli interfejs szeregowy routera pracuje jako urządzenie DCE, obowiązkowo dla tego interfejsu zdefiniować należy parametr zegara transmisji. W tym celu wykonujemy następujące polecenie w ramach konfiguracji interfejsu, podając jako parametr jedną z dozwolonych wartości (wyrażoną w bps):

```
C2600(config-if)#clock rate 128000
```

Dla wszystkich interfejsów szeregowych można dodatkowo skonfigurować przepustowość oraz opóźnienie wprowadzane przez dany interfejs. Trzeba jednak pamiętać, że obydwa te parametry są statycznie wpisywane przez administratora (początkowo mają wartości domyślne, wynikające z typu interfejsu), mają znaczenie etykietowe i nie odzwierciedlają w żadnym wypadku faktycznej komunikacji przez konkretny interfejs. Modyfikuje się je w celu zmiany środowiska pracy protokołów routingu dynamicznego, takich jak IGRP czy OSPF.

W poniższym przykładzie polecenia definiują przepustowość i opóźnienie dla interfejsu szeregowego. Parametr dla polecenia bandwidth wyrażany jest w kbps, natomiast opóźnienie podaje się w dziesiątkach mikrosekund:

```
C2600(config-if)#bandwidth 128

C2600(config-if)#delay 2000
```

Zdefiniowane dla interfejsów parametry oraz stan ich pracy można w dowolnej chwili obejrzeć poleceniem show interfaces - wyświetla ono m.in. następujące komunikaty dla konkretnego interfejsu Serial 0/0:

```
C2600#show interfaces serial 0/0

Serial0/0 is up, line protocol is up

Hardware is PowerQUICC Serial
```

```
Internet address is 131.107.11.1/24

MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,


reliability 255/255, txload 1/255, rxload 1/255

Encapsulation HDLC, loopback not set, keepalive set (10 sec)
```

Informacja typu "Serial0/0 is up" oznacza poprawne działanie interfejsu w warstwie fizycznej (status down sygnalizuje na przykład brak częstotliwości nośnej lub niepodłączony kabel). Komunikat "line protocol is up" opisuje tu poprawne działanie protokołu warstwy łącza danych, czyli otrzymywanie pakietów keepalive (status down może oznaczać na przykład niezgodność protokołu warstwy drugiej bądź niezdefiniowany zegar (clock rate) w urządzeniu pracującym jako DCE). W pewnych sytuacjach stan interfejsu wyświetlany jest jako "administratively down", co oznacza, że w konfiguracji interfejsu włączono polecenie shutdown, blokujące pracę interfejsu.

Wśród innych ciekawych informacji wyświetlanych poleceniem show znaleźć można: adres sprzętowy MAC (dla interfejsów typu Ethernet), przypisany adres IP, parametr MTU (maksymalny rozmiar pola danych transmitowanej ramki), przepustowość (BW), opóźnienie (DLY), niezawodność i obciążenie interfejsu (dwa parametry rzeczywistej transmisji) oraz włączony protokół warstwy łącza danych (np. HDLC dla interfejsu szeregowego lub ARPA, czyli Ethernet II dla interfejsu typu Ethernet). Więcej parametrów dotyczących tylko protokołu IP dla wybranego interfejsu zobaczyć można po wykonaniu następującego polecenia:

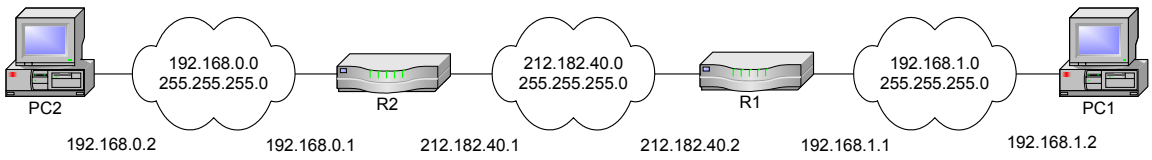
```
C2600#show ip interface Serial 0/0
```

Jeżeli administrator routera chce sprawdzić, które interfejsy szeregowe pracują jako urządzenia DTE, a które jako DCE, jaki został zdefiniowany zegar oraz jaki jest stosowany (zwykle zależny od wybranego kabla) protokół warstwy fizycznej, może wykonać komendę show controllers interfejs, które wyświetla parametry fizyczne interfejsu [4]:

```
C2600#show controllers serial 0/0

Interface Serial0/0

Hardware is PowerQUICC MPC860

DCE V.35, clock rate 56000
```

4. REALIZACJA PRAKTYCZNA PROTOKOŁU OSPF

Ćwiczenie zostało zrealizowane na dwóch routerach firmy CISCO 1600, dla przykładowej sieci:



Rys 4

Pierwszą czynnością było ustanowienie połączenia z routerem poprzez port konsoli - interfejs RS-232 lub RJ-45, do którego podłączyliśmy komputer z emulatorem terminala (np. HyperTerminal w systemach Windows). Za pomocą terminala przeprowadziliśmy proces konfiguracji routera. Poprawna komunikacja z routerem wymagała ustawienia odpowiednich parametrów transmisyjnych terminala:

- terminal typu VT100
- prędkość 9600 (liczba bitów na sekundę)
- 8 bitów danych
- 1 bit stopu
- transmisję bez parzystości.

Powyższa konfiguracja została przeprowadzona zarówno dla routera R1 jak i R2 w sposób identyczny.

Konfiguracja routera R1:

Pierwszym zadaniem konfiguracyjnym, było właściwe zdefiniowanie parametrów komunikacyjnych dla poszczególnych interfejsów dotyczących segmentów sieci lokalnej. Dla interfejsów sieci LAN, takich jak Ethernet, zwykle wystarczające jest zdefiniowanie parametrów dotyczących adresowania w protokole warstwy sieciowej (np. IP) oraz odwołanie domyślnie włączonego polecenia shutdown, które blokuje pracę interfejsu.

Poniższa sekwencja poleceń pokazuje wywołanie trybu konfiguracyjnego, wybór właściwego interfejsu, przypisanie adresu IP i maski podsieci do interfejsu Ethernet0 oraz wyłączenie polecenia shutdown blokującego interfejs.

1) konfiguracja interfejsów

```
Router>enable
Router#config

Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)#interface Ethernet0
Router(config-if)#ip address 212.182.40.2 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit

Router(config)#interface Ethernet1
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#exit
Router#

Router#show interfaces
Ethernet0 is up, line protocol is up
  Hardware is QUICC Ethernet, address is 0002.4b5b.4a6e (bia 0002.4b5b.4a6e)
  Internet address is 212.182.40.2/24
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:07, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
```

```
  5 minute output rate 0 bits/sec, 0 packets/sec
     247 packets input, 36719 bytes, 0 no buffer
     Received 197 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     1358 packets output, 114343 bytes, 0 underruns
     463 output errors, 0 collisions, 5 interface resets
     0 babbles, 0 late collision, 2 deferred
     463 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Ethernet1 is up, line protocol is up
  Hardware is QUICC Ethernet, address is 0002.4b5b.4a6f (bia 0002.4b5b.4a6f)
  Internet address is 192.168.1.1/24
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:01:12, output 00:00:09, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     282 packets input, 50196 bytes, 0 no buffer
     Received 224 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     1356 packets output, 118060 bytes, 0 underruns
     324 output errors, 0 collisions, 4 interface resets
     0 babbles, 0 late collision, 0 deferred
     324 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Serial0 is administratively down, line protocol is down
  Hardware is QUICC Serial
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters 03:01:58
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
     DCD=up  DSR=down  DTR=down  RTS=down  CTS=down

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

Gateway of last resort is not set

C    212.182.40.0/24 is directly connected, Ethernet0
C    192.168.1.0/24 is directly connected, Ethernet1
```

2) włączenie protokołu OSPF:

```
Router#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)#router ospf 1

Router(config-router)#network 192.168.1.0 255.255.255.0 area 0
Router(config-router)#network 212.182.40.0 255.255.255.0 area 1

Router(config-router)#exit
Router(config)#exit
Router#

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

Gateway of last resort is not set

C    212.182.40.0/24 is directly connected, Ethernet0
C    192.168.1.0/24 is directly connected, Ethernet1
```

3) Sprawdzenie połączenia z routerem R2

```
C:\>ping 192.168.1.1

Badanie 192.168.1.1 z użyciem 32 bajtów danych:

Odpowiedź z 192.168.1.1: bajtów=32 czas<10ms TTL=255
Odpowiedź z 192.168.1.1: bajtów=32 czas<10ms TTL=255
Odpowiedź z 192.168.1.1: bajtów=32 czas<10ms TTL=255
Odpowiedź z 192.168.1.1: bajtów=32 czas<10ms TTL=255

Statystyka badania dla 192.168.1.1:
    Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0 (0% utraconych),
Szacunkowy czas błądzenia pakietów w millisekundach:
    Minimum = 0ms, Maksimum =  0ms, Średnia =  0ms
```

Konfiguracja routera R2:

1) konfiguracja interfejsów:

```
Router>enable
Router#config

Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)#interface Ethernet0
Router(config-if)#ip address 212.182.40.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit

Router(config)#interface Ethernet1
Router(config-if)#ip address 192.168.0.1  255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
Router(config)#exit
```

```
Router#

Router#show interfaces
Ethernet0 is up, line protocol is up
  Hardware is QUICC Ethernet, address is 0004.c1c7.cb88 (bia 0004.c1c7.cb88)
  Internet address is 212.182.40.1/24
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:21, output 00:00:06, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     53 packets input, 4632 bytes, 0 no buffer
     Received 44 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 input packets with dribble condition detected
     1006 packets output, 159096 bytes, 0 underruns
     5 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 0 deferred
     5 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Ethernet1 is up, line protocol is up
  Hardware is QUICC Ethernet, address is 0004.c1c7.cb89 (bia 0004.c1c7.cb89)
  Internet address is 192.168.0.1/24
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:07:53, output 00:00:03, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     161 packets input, 22269 bytes, 0 no buffer
     Received 161 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 input packets with dribble condition detected
     917 packets output, 132753 bytes, 0 underruns
     95 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 0 deferred
     95 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Serial0 is administratively down, line protocol is down
  Hardware is QUICC Serial
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 16 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
     DCD=up  DSR=down  DTR=down  RTS=down  CTS=down
```

2) włączenie protokołu OSPF:

```
Router#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Router(config)#router ospf 1
Router(config-router)#network 192.168.0.0 255.255.255.0 area 0
Router(config-router)#network 212.182.40.0 255.255.255.0 area 1
Router(config-router)#exit
Router(config)#exit
Router#

Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

C    212.182.40.0/24 is directly connected, Ethernet0
C    192.168.0.0/24 is directly connected, Ethernet1
O IA 192.168.1.0/24 [110/20] via 212.182.40.2, 00:00:54, Ethernet0
```

3)

Sprawdzenie połączenia z routerem R2

```
C:\>ping 192.168.0.1

Badanie 192.168.0.1 z użyciem 32 bajtów danych:

Odpowiedź z 192.168.0.1: bajtów=32 czas=10ms TTL=255
Odpowiedź z 192.168.0.1: bajtów=32 czas<10ms TTL=255
Odpowiedź z 192.168.0.1: bajtów=32 czas<10ms TTL=255
Odpowiedź z 192.168.0.1: bajtów=32 czas<10ms TTL=255

Statystyka badania dla 192.168.0.1:
    Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0 (0% utraconych),
Szacunkowy czas błądzenia pakietów w millisekundach:
    Minimum = 0ms, Maksimum =  10ms, Średnia =  2ms
```

//

Sprawdzenie połączenia z komputerem PC1

```
C:\>ping 192.168.1.2

Badanie 192.168.1.2 z użyciem 32 bajtów danych:

Odpowiedź z 192.168.1.2: bajtów=32 czas=10ms TTL=126
Odpowiedź z 192.168.1.2: bajtów=32 czas<10ms TTL=126
Odpowiedź z 192.168.1.2: bajtów=32 czas<10ms TTL=126
Odpowiedź z 192.168.1.2: bajtów=32 czas<10ms TTL=126

Statystyka badania dla 192.168.1.2:
    Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0 (0% utraconych),
Szacunkowy czas błądzenia pakietów w millisekundach:
  Minimum = 0ms, Maksimum =  10ms, Średnia =  2ms
```

LITERATURA

[1] http://www.networld.pl/artykuly/20910_3.html
[2] http://www.networld.pl/artykuly/20910_4.html
[3] http://www.pckurier.pl/archiwum/art0.asp?ID=5011
[4] http://www.pckurier.pl/archiwum/art0.asp?ID=4734
[5] http://www.pckurier.pl/archiwum/art0.asp?ID=4340
[6] http://www.pckurier.pl/archiwum/art0.asp?ID=4904
[7] http://integrator.solidex.pl/index.phtml?php_wid=4&php_aid=2
[8] materiały udostępnione na serwerze Politechniki Rzeszowskiej.