Windows 2000 TCP/IP.
Czarna księga

Autor: Ian McLean
Tłumaczenie: Mateusz Izdebski, Piotr Kubiczek
ISBN: 83-7197-515-5
Tytuł oryginału: Windows 2000 TCP/IP Black Book
Format: B5, stron: około 900

TCP/IP jest tym, co wprawia w ruch sieć WWW. Jest on szeroko stosowany w intranetach i zawiera składniki obsługujące pocztę elektroniczną i grupy dyskusyjne, ale w głównej mierze jest zestawem protokołów internetowych, „rusztowaniem" Internetu. Wraz z rozwojem Internetu rozwija się TCP/IP, motor, który go napędza. Ów zestaw protokołów posiada teraz składniki obsługujące dane wideo i audio czasu rzeczywistego. Nowy protokół internetowy w wersji 6 (IPv6) daje wielki wzrost przestrzeni adresowej. Stare, wierne protokoły, takie jak TCP, zostały uaktualnione i zostały dodane do nich nowe funkcje. Udoskonalenia TCP/IP systemu Microsoft Windows 2000 stanowią znaczące usprawnienie i wr az z podstawami TCP/IP, dostarczają tematu niniejszej książce.

Wydawnictwo Helion
ul. Chopina 6
44-100 Gliwice
tel. (32)230-98-63
e-mail: helion@helion.pl

Spis treści

Rozdział 4.

Adresowanie w protokole IP

O historii krótko

Niniejszy rozdział opisuje, w jaki sposób adresy protokołu internetowego (IP) oraz maski podsieci współpracują ze sobą, aby zidentyfikować zarówno określonego hosta w danej sieci — gdzie host może być komputerem, bramą routera albo takim urządzeniem, jak drukarka sieciowa — jak i samą sieć. Opisany został zestaw możliwości protokołu IP w wersji 4 (IPv4), ponieważ IPv4 jest wersją aktualnie używaną w Internecie oraz w intranetach protokołu IP. Protokół IP wersji 6 (IPv6) jest opisany w rozdziale 18.

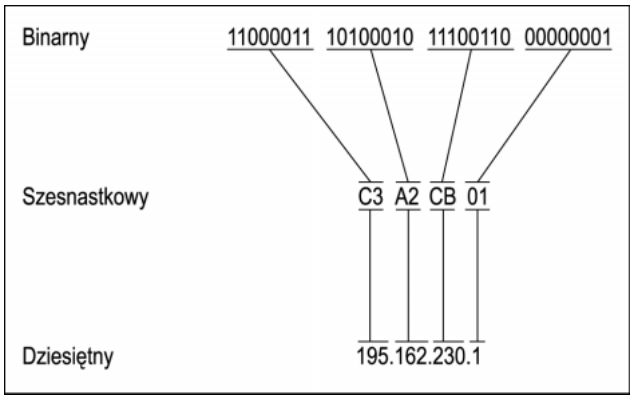
Adresy protokołu IP

W protokole IPv4 adres IP to 32-bitowa liczba binarna, która jest wykorzystywana do jednoznacznej identyfikacji danego hosta oraz jego sieci. Dwa hosty w danej sieci nie mogą mieć tego samego adresu IP. Adresy IP mogą być zapisywane w systemie binarnym (na przykład 11000011101000101110011000000001), ale jest to nieporęczne. Mogą też być zapisywane w systemie szesnastkowym (na przykład C3A2CB01). Jest to krótsze, ale i tak trudne do zapamiętania. Oczywiście mogą być również przekształcane bezpośrednio na system dziesiętny (3 282 225 921 w podanym przykładzie), ale ten format jest prawie tak trudny do zapamiętania, jak szesnastkowy. Jest on również znacznie mniej użyteczny, ponieważ wartość każdego z 4 bajtów w liczbie 32-bitowej jest ważna i nie jest łatwo do przeliczenia z wartości dziesiętnej.

W związku z tym normalną praktyką jest dzielenie danego adresu IP na 4 bajty, lub *oktety*, a następnie obliczanie wartości dziesiętnej dla każdego z oktetów. Oktety oddzielone są kropkami i stąd wywodzi się termin *kropkowa notacja dziesiętna*. Kropkowa notacja dziesiętna dla podanego przykładu to 195.162.230.1. W tym sposobie zapisu nie było nic szczególnego, kiedy go wybierano. Był to po prostu kompromis pomiędzy czytelnością a użytecznością.

Format dziesiętny kropkowy wykorzystuje się do wpisywania i wyświetlania adresów IP w szerokiej gamie graficznych interfejsów użytkownika (GUI), ale należy zawsze pamiętać, że adres IP (i w związku z tym maska podsieci, którą będziemy omawiali w dalszej części tego rozdziału) to po prostu 32-bitowa wartość binarna. Rysunek 4.1 przedstawia stosunek pomiędzy formatem binarnym, szesnastkowym oraz dziesiętnym kropkowym.

Rysunek 4.1. *Formaty adresów protokołu IP*



Klasowe adresy protokołu IP

Binarna liczba 32-bitowa daje zakres całkowity 4 294 967 296 adresów (z których nie wszystkie mogą zostać użyte). Kiedy określano przestrzeń adresową protokołu IP, adresy te zostały podzielone na grupy, czy też *klasy*. Chociaż wydaje się, że jest to jedyna rzecz, jaką można zrobić z ponad czterema milionami adresów, z dzisiejszej perspektywy był to prawdopodobnie błąd. Mimo to klasy adresów są nadal w powszechnym użyciu. Początkowe bity binarne adresu określają klasy adresów, co pokazano w tabeli 4.1. Niedozwolone są wartości pierwszego oktetu wynoszące 0; 127 oraz 255.

Tabela 4.1. *Klasy adresów protokołu IP*

Klasa	Bity początkowe	Wartość pierwszego oktetu
A	01	od 1 do 126
B	10	od 128 do 191
C	110	od 192 do 223
D	1110	od 224 do 239
E	1111	od 240 do 254

Przykładowo 195.162.230.1 to adres *klasy C*.

Sieci klasy A

W sieci *klasy A* tożsamość sieci określana jest przez wartość pierwszego oktetu, czy też ośmiu bitów. W związku z tym sieci *klasy A* są często określane jako *sieci /8*. Ponieważ zakres wartości dla pierwszego oktetu adresu *klasy A* to, z definicjistanowi 126 niepowtarzalnych sieci *klasy A*. Pozostałe 24 bity adresu identyfikują hosta. Tożsamości hostów nie mogą być wyłącznie jedynkami, ani wyłącznie zerami, więc maksymalna liczba hostów w każdej sieci *klasy A* to 2^{24}–2 lub 16 777 214.

Blok adresowy *klasy A* zawiera 2^{31} indywidualnych adresów (łącznie z zarezerwowanymi wartościami pierwszego oktetu, wynoszącymi 0 oraz 127), a przestrzeń adresowa IPv4 zawiera 2^{32} adresów. Stąd przestrzeń adresowa *klasy A* to 50% całkowitej przestrzeni adresowej IPv4.

Wszystkie adresy protokołu IP muszą być niepowtarzalne w swojej własnej sieci. Jeśli jednak dwie sieci złożone nie wiedzą o sobie nawzajem i nie mogłyby nigdy pojawić się na tej samej trasie, to ten sam adres IP mógłby pojawić się w obu z nich.

Tak więc intranet, który nigdy nie jest bezpośrednio routowany do Internetu, może korzystać z dowolnego zakresu adresów, jaki wybierze jego administrator pod warunkiem, że wszystkie adresy wewnętrzne są niepowtarzalne. Zazwyczaj do wewnętrznego adresowania w intranecie wykorzystywana jest sieć *klasy A* 10.0.0.0. Jeżeli hosty w danej sieci 10.0.0.0 mają mieć dostęp do Internetu, to musi zostać zaimplementowana usługa translacji adresów sieciowych (NAT).

Sieci klasy B

W sieci *klasy B* tożsamość sieciowa określana jest przez wartość pierwszych dwóch oktetów, czyli 16 bitów. Sieci *klasy B* są zatem czasami określane jako *sieci /16*. 2 pierwsze bity identyfikują daną sieć jako sieć *klasy B*, co pozostawia 14 bitów na określenie niepowtarzalnych tożsamości sieciowych. Stąd też można zdefiniować 2^{14}, czyli 16 384 sieci *klasy B*, przy czym każda z nich może mieć 2^{16}–2, czyli 65 534 hostów. Blok adresowy *klasy B* zawiera 2^{30} (1 073 741 824) adresów i stanowi 25% całkowitej przestrzeni adresowej IPv4.

Sieci klasy C

W sieci *klasy C* tożsamość sieciowa jest określana przez wartość pierwszych trzech oktetów, czyli 24 bity. Sieci *klasy C* są zatem czasami określane jako *sieci /24*. 3 pierwsze bity identyfikują daną sieć jako sieć *klasy C*, co pozostawia 21 bitów na określenie niepowtarzalnych tożsamości sieciowych. Stąd też można zdefiniować 2^{21}, czyli 2 097 152 sieci *klasy C*, przy czym każda z nich może mieć do 2^{8}–2, czyli 254 hostów. Blok adresowy *klasy C* zawiera 2^{29} (536 870 912) adresów i stanowi 12,5% całkowitej przestrzeni adresowej IPv4.

Klasy D i E

Sieci *klasy D* wykorzystywane są do multiemisji, gdzie pojedynczy adres sieciowy identyfikuje grupę hostów. Multiemisja została przedstawiona w rozdziale 3., a będzie dalej omawiana w rozdziale 5. Sieci *klasy E* zarezerwowane są do celów doświadczalnych. Blok *klasy D* stanowi 6,25% całkowitej przestrzeni adresowej IPv4, a blok *klasy E* nieznacznie mniejszą jej część, ponieważ wartość 255 nie jest wykorzystywana jako wartość pierwszego oktetu.

Maska podsieci

Maska podsieci, podobnie jak adres IP, jest 32-bitową liczbą binarną, ale posiada bardzo specyficzny format. Musi ona składać się z grupy jedynek poprzedzającej grupę zer — na przykład 11111111111111110000000000000000. Maski podsieci są zazwyczaj zapisywane albo przy użyciu kropkowej notacji dziesiętnej (255.255.0.0), albo w formacie *ukośnikowym*, gdzie wartość po ukośniku reprezentuje liczbę jedynek (/16).

Format ukośnikowy a format dziesiętny kropkowy

Format dziesiętny kropkowy jest opisywany jako „staroświecki" sposób określania masek podsieci od kilku lat, ale jest on wciąż — prawdopodobnie — formatem najczęściej używanym. Zgrabniej jest określić daną sieć jako 195.162.230.0/24 zamiast 195.162.230.0, maska podsieci 255.255.255.0, ale ten drugi format przekłada się bardziej na informacje, które trzeba wpisać w oknach dialogowych konfiguracji IP. System NT4 nie korzysta z formatu ukośnikowego (chyba że gdzieś go przeoczyłem), a system Windows 2000 nie korzysta z niego we wszystkich oknach dialogowych. Format dziesiętny kropkowy jest często stosowany w obliczeniach podziału na podsieci, podczas gdy bezklasowe wybieranie trasy (CIDR) i łączenie w nadsieć mogą z powodzeniem korzystać z notacji skróconej. Obie konwencje warto poznać.

Funkcją maski podsieci jest identyfikowanie, która część adresu IP określa sieć, a która część określa hosta. Jedynki określają, że odpowiadające im bity w adresie IP to bity sieci, a zera określają bity hosta. W przypadku tradycyjnego adresowania klasowego, początkowe bity adresu określają klasę adresu, która z kolei określa zakres hosta i sieci. Stąd, kiedy wprowadzano adresy IP oraz adresowanie klasowe, nie zostały zaimplementowane maski sieci.

Jednak analiza początkowych bitów adresu jest nużąca, a maski podsieci upraszczają ten proces. Binarna operacja AND sprawia, że zera w masce podsieci maskują część hosta w adresie IP, pozostawiając tylko te bity, które identyfikują sieć, albo *prefiks sieci*. Adresy *klasy A* (adresy */8*) mają domyślną maskę podsieci */8* (255.0.0.0). *Klasy B* i *C* mają domyślne maski podsieci, odpowiednio, */16* (255.255.0.0) i */24* (255.255.255.0).

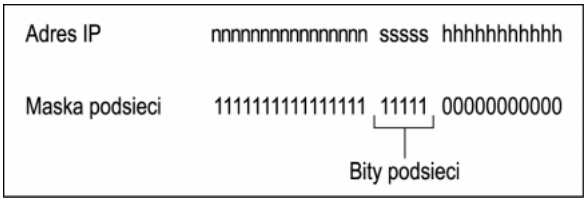
Pierwotnie maski podsieci zostały wprowadzone, aby ułatwić obliczanie adresu sieciowego. Jednak nie minęło wiele czasu, a zaczęły być wykorzystywane do innego celu — aby dzielić sieci *klasy A*, *B* oraz *C* na mniejsze części za pomocą techniki znanej jako *podział na podsieci*.

Podział na podsieci

W 1985 roku dokument RFC 950 określił standardową procedurę obsługującą podział na podsieci. Procedura została wprowadzona, ponieważ administrator lokalny, który potrzebował drugiej sieci, zmuszony był żądać innego numeru sieci, pomimo że wciąż były dostępne adresy hostów (często duża liczba adresów hostów) w sieci pierwotnie przydzielonej.

Podział na podsieci dodaje dodatkowy poziom hierarchii do struktury adresowania IP. Zamiast klasowej hierarchii dwupoziomowej, podział na podsieci realizuje hierarchię trzypoziomową. Dzieli on standardowe klasowe pole numeru hosta na dwie części — numer podsieci oraz numer hosta w tej podsieci.

W gruncie rzeczy podział na podsieci bierze bity z adresu hosta i zamiast tego przydziela te bity adresowi sieci, w ten sposób dokonując dalszego podziału sieci. Rysunek 4.2 przedstawia sieć (*/16*) *klasy B*, w której pięć *bitów podsieci* zostało wziętych z przydziału adresu hosta i dodanych do przydziału adresu sieci, dając większą liczbę sieci z mniejszą liczbą hostów w każdej z nich.



Rysunek 4.2. *Przydzielanie bitów podsieci*

Ponieważ maska podsieci przypisuje binarną jedynkę do bitu adresu sieci, a binarne zero do bitu adresu hosta, bity podsieci w masce podsieci przybierają wartość binarnej jedynki. Domyślnie dla sieci *klasy B* maska podsieci wynosi 255.255.0.0 (*/16*), ale zamienia się w 255.255.248.0 (*/21*), kiedy zostaje przydzielonych 5 bitów dla podziału na podsieci.

Można to najlepiej przedstawić za pomocą konkretnego przykładu. Przypuśćmy, że masz sieć *klasy B* 131.11.0.0. Wtedy, w formie binarnej, dowolny adres w tej sieci to:

```
10000011 00001011 hhhhhhhh hhhhhhhh,
```

gdzie `h` oznacza bit adresu hosta.

Aby dokonać dalszego podziału sieci, utrzymujemy tę samą tożsamość sieci, ale wykorzystujemy niektóre bity (w tym przykładzie 5 bitów) z tożsamości hosta (ID) do utworzenia tożsamości podsieci, w sposób następujący:

```
Adres IP             10000011 00001011 sssshhh hhhhhhhh
Maska podsieci       11111111 11111111 11111000 00000000
```

gdzie `s` oznacza bit maski podsieci.

Jeżeli dwa hosty są w tym samym segmencie, lub *podsieci*, danej sieci, to muszą one mieć taką samą tożsamość sieci i taką samą tożsamość podsieci. Jeżeli są one w różnych podsieciach, to mają identyczne tożsamości sieci, ale różne tożsamości podsieci. Na przykład adresy IP:

```
Adres IP 1       10000011 00001011 10010011 00100000 (131.11.147.64)
Adres IP 2       10000011 00001011 10010100 00110000 (131.11.148.96)
Maska podsieci   11111111 11111111 11111000 00000000 (255.255.248.0)
```

są w tej samej sieci. Jednak adresy IP:

```
Adres IP 3       10000011 00001011 10011011 00100000 (131.11.153.64)
Adres IP 2       10000011 00001011 10010101 00110000 (131.11.149.96)
Maska podsieci   11111111 11111111 11111000 00000000 (255.255.248.0)
```

są w różnych podsieciach. Innymi słowy, aby dwa adresy mogły być w tej samej sieci, bity, które odpowiadają binarnym jedynkom w masce podsieci, muszą być identyczne dla obu adresów.

Obliczanie liczby podsieci i hostów

Mając tożsamość sieci i maskę podsieci, ile podsieci możemy utworzyć i ile hostów może rezydować w każdej z podsieci?

Weźmy przykład 3 bitów podsieci. W adresie IP bity te mogą przybierać następujące wartości:

```
000
001
010
011
100
101
110
111
```

Jednak bity podsieci w adresie IP nie mogą być samymi jedynkami ani samymi zerami, więc wartości 000 oraz 111 są wykluczone. A zatem jest sześć możliwych wartości dla bitów podsieci.

Ogólnie rzecz biorąc, jest 2^x–2 możliwych podsieci, gdzie *x* stanowi liczbę bitów podsieci. W rozpatrzonym przez nas wcześniej przykładzie jest 5 bitów podsieci, a więc jest 2^5–2 (tj. 30) podsieci.

Niektóre współczesne routery przyjmują same jedynki w bitach podsieci. Jeżeli masz intranet korzystający wyłącznie z routerów, które mają tę funkcję, możesz zwiększać liczbę podsieci. Jeżeli jednak routery wymieniają informacje tablic tras z innymi, starszymi routerami w Internecie, to nie powinno być używane ustawienie „same jedynki".

Korzystając ponownie z przykładu, którego użyliśmy poprzednio: jeżeli weźmiemy 5 bitów z zakresu adresów hosta, zostaje nam jeszcze 11 bitów dla adresów hostów. Adres hosta nie może składać się z samych jedynek, ani z samych zer, więc w każdej z podsieci może rezydować maksymalnie 2^{11}–2 (tj. 2 046) hostów. Gdybyśmy, z drugiej strony, przydzielili tylko 3 bity podsieci, to pozostałoby nam 13 bitów na adresy hostów, co daje 2^{13}–2 (tj. 8 190) tożsamości hostów w każdej z naszych sześciu podsieci.

Przedstawiony przykład to podzielona na podsieci sieć *klasy B*. Dokładnie te same zasady można zastosować wobec sieci *klasy A* i *klasy C*. Procedury do przeprowadzania tych obliczeń są podane w podrozdziale „Rozwiązania natychmiastowe".

Obliczanie zakresu adresów IP dla podsieci

Po obliczeniu liczby podsieci oraz liczby hostów na podsieć dla pary typu adres IP — maska podsieci, następny krok to rozpracowanie zakresu adresów IP dla każdej z podsieci. Aby zilustrować tę technikę wykorzystamy przykład, który już rozważaliśmy: tożsamość sieci o wartości 131.11.0.0 z maską podsieci o wartości 255.255.248.0 (czasami zapisywaną 131.11.0.0/21).

Stosujemy trzy reguły:

- bity maski podsieci nie mogą być samymi zerami,
- bity tożsamości hosta nie mogą być samymi zerami,
- bity tożsamości hosta nie mogą być samymi jedynkami.

Zatem pierwsza wartość podsieci, jakiej możemy użyć, to 0001, pierwsza tożsamość hosta, jaką możemy określić to 0000000001, a ostatnia tożsamość hosta, jaką możemy określić to 11111111110. Dla pierwszej podsieci daje to wartości:

```
Tożsamość sieci            10000011 00001011 00000000 00000000 (131.11.0.0)
Maska podsieci             11111111 11111111 11111000 00000000 (255.255.248.0)
Pierwszy adres IP          10000011 00001011 00001000 00000001 (131.11.8.1)
Ostatni adres IP           10000011 00001011 00001111 11111110 (131.11.15.254)
```

A zatem, w podanym przykładzie, zakres adresów IP dla pierwszej podsieci przyjmuje wartości od 131.11.8.1 do 131.11.15.254. Zastosowanie tych samych obliczeń do drugiej podsieci daje zakres od 131.11.16.1 do 131.11.23.254. Tę samą technikę można zastosować wobec dowolnej pary typu tożsamość sieci — maska podsieci; można też wyprowadzić tablicę zakresów podobną do tabeli 4.2.

Tabela 4.2. *Zakresy adresów podsieci przypadku sieci 131.11.0.0/21*

Podsieć	Zakres adresów
1	131.11.8.1 do 131.11.15.254
2	131.11.16.1 do 131.11.23.254
3	131.11.24.1 do 131.11.31.254
-	- - - - - - - - - - - - - - - - - - - -
-	- - - - - - - - - - - - - - - - - - - -
30	131.11.240.1 do 131.11.247.254

W tej sekcji wyprowadziliśmy liczbę podsieci, liczbę hostów na podsieć oraz zakresy adresów dla każdej z podsieci przy użyciu arytmetyki binarnej. Wykonanie tych czynności jest potrzebne do zrozumienia, w jaki sposób dokonuje się podziału na podsieci i w jaki sposób są obliczane numery. Jednak byłoby rzeczą skrajnie nużącą przeprowadzanie pełnych obliczeń binarnych ilekroć chcielibyśmy dokonać podziału na podsieci. W podrozdziale „Rozwiązania natychmiastowe" zobaczymy, w jaki sposób budować tablicę podsieci, która zdejmie z nas ciężar dokonywania obliczeń i umożliwi nam obliczanie optymalnej struktury podziału na podsieci, wziąwszy pod uwagę wymagania związane z liczbą podsieci oraz liczbą hostów przypadających na podsieć.

Bez względu na to jak jesteś biegły w korzystaniu z tabeli podsieci, zawsze upewniaj się, czy potrafisz rozpracować podział na podsieci i czy rozumiesz, w jaki sposób wyprowadzane są numery. Skróty są świetne, kiedy wszystko się udaje.

Maski podsieci o zmiennej długości

Czasem bywają mylone pojęcia podziału na podsieci i masek podsieci o zmiennej długości (VLSM). Jest to zrozumiałe — sedno techniki podziału na podsieci polega na zmianie długości maski podsieci. Jednakże kiedy dzielisz sieć na podsieci, rozbijasz ją na segmenty, z których wszystkie są tej samej wielkości. Pojedynczą maskę podsieci, aczkolwiek nie domyślną maskę podsieci, stosuje się wobec całej sieci.

W 1987 roku dokument RFC 1009 określił, w jaki sposób sieć może wykorzystywać więcej niż jedną maskę podsieci, aby implementować segmenty różnej długości. VLSM umożliwia przypisanie danej sieci więcej niż jednej maski, w związku z czym rozszerzone prefiksy sieci różnych segmentów sieci mają różne długości.

Niestety niektóre protokoły routingu, takie jak protokół routingu internetowego w wersji 1 (RIPv1), wymagają jednolitych masek podsieci w obrębie całego prefiksu sieci. RIPv1 pozwala na użycie tylko pojedynczej maski podsieci z każdym z numerów sieci, ponieważ nie zapewnia on informacji o maskach podsieci w ramach swoich komunikatów uaktualnień tablicy tras.

Jednakże protokoły bardziej elastyczne, takie jak RIPv2 i protokół otwierania najkrótszej ścieżki w pierwszej kolejności (OSPF), dopuszczają VLSM. Jest kilka korzyści płynących z przydzielania wielu masek podsieci danemu numerowi IP sieci:

- umożliwiają one bardziej wydajne wykorzystanie przydzielonej danemu przedsiębiorstwu przestrzeni adresów IP;
- umożliwiają one *zespalanie tras*, co może znacząco ograniczyć ilość informacji dotyczących routingu w obrębie domeny routingu danej organizacji.

Wydajne wykorzystanie przydzielonej przestrzeni adresów IP

Jednym z ważniejszych problemów związanych z wcześniejszymi ograniczeniami obsługiwania tylko jednej maski podsieci w obrębie danego prefiksu sieci było to, że kiedy została wybrana maska, to zamykała ona przedsiębiorstwo w stałej liczbie równych rozmiarami podsieci. Biorąc przykład, który rozpracowaliśmy we wcześniejszej części tego rozdziału, sieć 131.11.0.0/21 zapewniała 30 podsieci, przy czym każda z nich miała 2 046 hostów. Ale wyobraźmy sobie, że podsieć *klasy B* została przydzielona przedsiębiorstwu posiadającemu dwa duże zakłady, z których każdy wymaga około 5 000 adresów IP. Ponadto przedsiębiorstwo ma 25 filii, z których każda wymaga najwyżej 200, a często znacznie mniej, adresów IP.

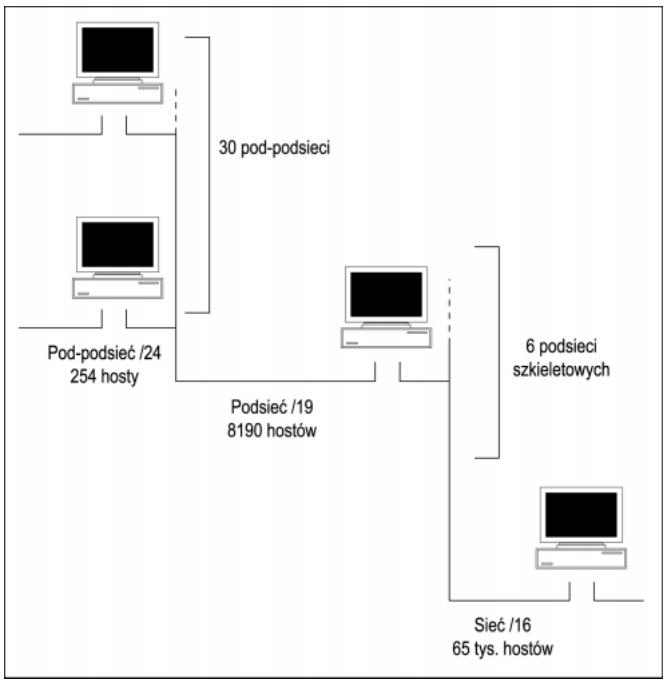
Oba z tych dużych zakładów potrzebowałyby co najmniej trzech podsieci, a przydzielono by im prawdopodobnie cztery. Oznacza to poważną i być może niepotrzebną, inwestycję w routery. Mogą być inne powody segmentowania sieci liczącej 8 000 użytkowników (jak na przykład ograniczanie ruchu emisji), ale konstruktor sieci powinien mieć wybór określenia najbardziej wydajnej segmentacji, a nie powinien być zmuszony do zastosowania segmentów liczących 2 000 hostów.

Tym, co stanowi jednak dużo większy problem, jest fakt, że każda z filii, liczących 200 użytkowników, musi korzystać z podsieci liczących 2 000 hostów. Oznacza to poważne marnotrawstwo przestrzeni adresów IP. W rzeczywistości przy ośmiu podsieciach już przydzielonych dużym zakładom przedsiębiorstwu nie pozostaje wystarczająco dużo podsieci, aby przydzielić jedną każdej z filii. Dlatego też potrzebuje ono albo drugiej sieci, pomimo że wykorzystuje o wiele mniej adresów IP, niż 65 000, które (teoretycznie) zapewnia jego sieć *klasy B*, albo też musi implementować maskę podsieci */22* (62 podsieci). To drugie rozwiązanie prowadziłoby do jeszcze większej liczby routerów w dużych zakładach oraz do dwukrotnego wzrostu ogłaszanych tras.

Nie omówiłem matematyki binarnej, aby wytłumaczyć liczby, które podaję w tym przykładzie. Jest to całkowicie zamierzone. Jeżeli nie rozumiesz skąd pochodzą te liczby, rozpracuj je przy użyciu techniki przedstawionej w jednym z poprzednich podrozdziałów („Podział na podsieci").

Rozwiązanie VLSM polega na określeniu sześciu podsieci */19* o pojemności 2^{13}–2 (tj. 8 190) adresów hostów każda. Dwie z nich mogą zostać przydzielone dużym zakładom, a trzecia może zostać bardziej podzielona przy użyciu maski podsieci */24* — co daje 30 podsieci liczących 254 hostów. Przedsiębiorstwu pozostają jeszcze trzy podsieci */19*, lub połowa przydzielonej mu przestrzeni adresowej, na przyszły rozwój. Rysunek 4.3 przedstawia tę strategię podziału na podsieci.

Rysunek 4.3. *Wykorzystywanie VLSM do implementowania wydajnej segmentacji sieci*



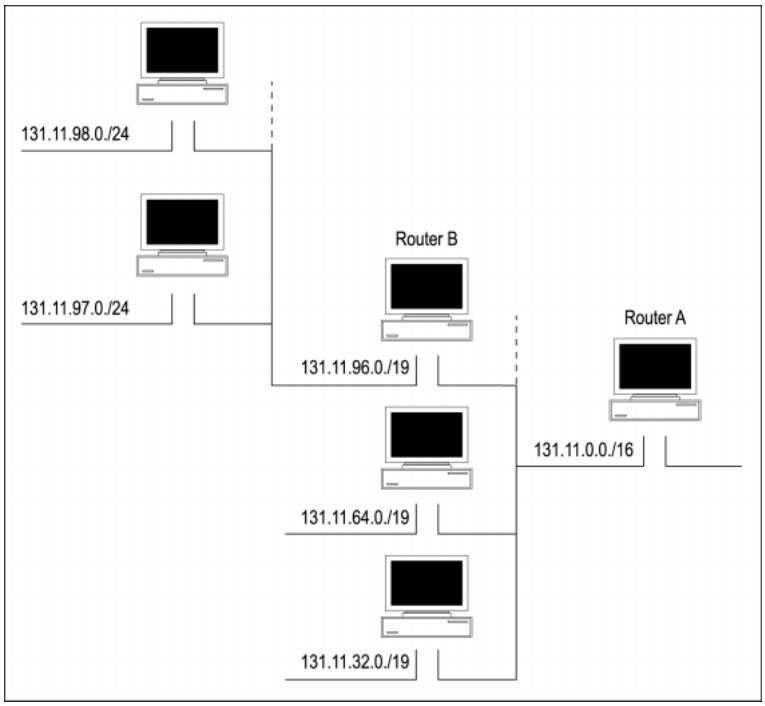
Zespalanie tras

VLSM działa poprzez dzielenie danej sieci na podsieci mające największe wymagane rozmiary (podsieci *szkieletowe*), a następnie dokonanie dalszego podziału tych dużych podsieci według potrzeby. Ten *rekurencyjny* podział umożliwia ponowne zebranie i zespolenie przestrzeni adresowej, co z kolei ogranicza ilość informacji dotyczących routingu na najwyższym poziomie i pozwala na ukrycie szczegółowej struktury informacji routingu dla jednej z grup podsieci przed inną grupą podsieci.

Dyskusja ta zakłada, że podział na podsieci jest jedyną wykorzystywaną techniką. W praktyce można by skorzystać z serwera translacji adresów sieciowych (NAT), aby ograniczyć liczbę tras, które dane przedsiębiorstwo ogłasza w Internecie, a także by chronić wewnętrzne adresy IP przedsiębiorstwa.

Na przykład, w omówionym wcześniej przykładzie podziału na podsieci, wszystkich 30 podsieci */21* byłoby ogłaszanych, tak wewnętrznie, jak i zewnętrznie, przez tablice tras przedsiębiorstwa. Jeżeli jednak zastosuje się rozwiązanie VLSM, jak na rysunku 4.4, to *Router A* ogłasza w Internecie tylko jedną pozycję sieciową tablicy tras (131.11.0.0/16). *Router B* zespala wszystkie podsieci */24* w jedną tożsamość podsieci */19*, którą ogłasza w sieci szkieletowej organizacji. Prowadzi to powstania mniejszych tablic tras i zmniejszenia się ruchu ogłoszeń routingu.

Rysunek 4.4. *Zespalanie tras przy użyciu VLSM*

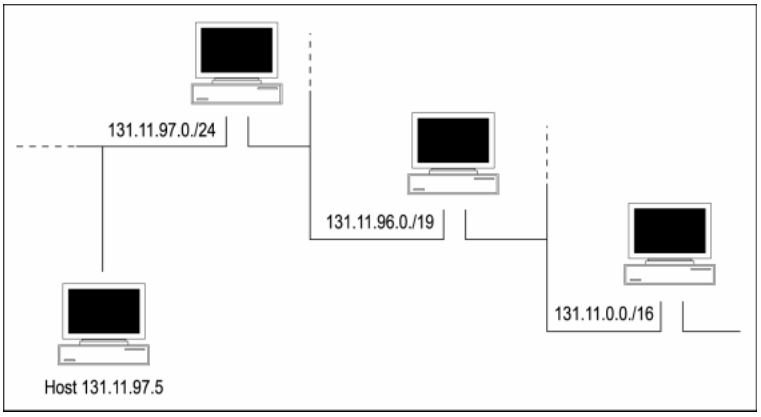


Algorytm najdłuższego dopasowania

Routery implementują spójny algorytm przekazywania oparty na algorytmie *najdłuższego dopasowania*. Jeżeli wykorzystywany jest VLSM, to większe podsieci (z mniejszymi prefiksami sieci) zostają bardziej podzielone, tworząc mniejsze pod-podsieci (z większymi prefiksami sieci). Mówi się, że pod-podsieci są *bardziej określone*, ponieważ dłuższy prefiks sieci lepiej określa lokalizację danego hosta w sieci.

Na przykład na rysunku 4.5 trasa sieciowa do hosta 131.11.97.5 może być określona jako 131.11.0.0/16, 131.11.96.0/19 lub 131.11.97.0/24. Ponieważ te bardziej określone segmenty sieci są podsieciami tych mniej określonych segmentów, host jest na wszystkich trzech trasach.

Przy użyciu algorytmu najdłuższego dopasowania router przekazujący będzie routował do najbardziej określonej sieci to jest 131.11.97.0/24. Oznacza to, że host 131.11.97.5 musi być zainstalowany w podsieci 131.11.97.0/24. Gdyby, przez pomyłkę, host ten został podłączony do sieci szkieletowej 131.11.96.0/19, nie udałoby się go nigdy osiągnąć.



Rysunek 4.5. *Algorytm najdłuższego dopasowania*

Wdrażanie VLSM

Wdrażanie hierarchicznego schematu podziału na podsieci zapewnionego przez VLSM wymaga starannego planowania. Musisz brnąć w dół poprzez plan adresów, aż dotrzesz do najgłębszego poziomu, gdzie musisz się upewnić, czy najmniejsze podsieci, albo podsieci *liścia*, są wystarczająco duże, aby obsługiwać wymaganą liczbę hostów. Jeżeli VLSM jest wdrażany przy użyciu logicznej struktury hierarchicznej — tak aby plan adresów odzwierciedlał strukturę albo *topologię* sieci — to adresy każdej z podsieci mogą być zespalane w pojedynczy blok adresowy, który nie dopuszcza, by tablice tras sieci szkieletowej zbytnio się rozrosły.

Aby wdrażanie VLSM w pełni się powiodło muszą być spełnione trzy wymogi:

- Protokoły routingu muszą nieść rozszerzone informacje o prefiksach sieci wraz z każdym ogłoszeniem tras. Takie protokoły, jak RIPv2 i OSPF mają tę funkcję.
- Routery muszą implementować algorytm najdłuższego dopasowania.
- Adresy muszą być przydzielone tak, aby miały znaczenie topologiczne, umożliwiając w ten sposób zespalanie tras.

Bezklasowy routing międzydomenowy

Bezklasowy routing międzydomenowy (CIDR), opisany w dokumentach RFC 1518 i 1519, usuwa pojęcie *klasy* z przydzielania i zarządzania adresami IP. Zamiast wstępnie zdefiniowanych klas (*A*, *B* oraz *C*) alokacje CIDR są określane za pomocą adresu początkowego i zakresu. Zakres (w praktyce maska podsieci) określa sieciową część adresu.

Pozwala to na wydajniejsze wykorzystanie dostępnej przestrzeni. Na przykład dostawca usług internetowych (ISP) może przydzielić klientowi 151.26.2.128/25. Klient może następnie korzystać z adresów IP od 151.26.2.129 do 151.26.2.254. Historycznie rzecz ujmując, przedsiębiorstwo zatrudniające (powiedzmy) 10 tys. pracowników prosiło o (i otrzymywało) pełną sieć *klasy B*, która zawierała ponad 65 000 adresów IP. Wraz z innymi czynnikami omówionymi w dalszej części niniejszego rozdziału, doprowadziło to prawie do wyczerpania przestrzeni adresowej IPv4. CIDR umożliwia elastyczną alokację numerów adresowych bardziej współmierną do potrzeb danego przedsiębiorstwa.

Dokument RFC 1917 uprasza społeczność internetową o zwracanie nieużywanych alokacji adresów do organizacji przydzielania adresów internetowych (IANA) w celu dokonania redystrybucji. Alokacje te zawierają nieużywane numery sieciowe, adresy dla sieci, które nigdy nie będą połączone z Internetem z przyczyn bezpieczeństwa oraz alokacje z ośrodków, które wykorzystują jedynie mały odsetek swojej przestrzeni adresowej. W miarę zwracania tych adresów IP, będą one ponownie przydzielane przy użyciu bezklasowych technik CIDR. Niestety wiele przedsiębiorstw, które posiadają nieużywane adresy, nie chce ich zwracać, ponieważ są one postrzegane jako cenne.

Wdrażanie CIDR

Zarówno CIDR, jak i VLSM umożliwiają częściowy podział przestrzeni adresów IP na mniejsze kawałki. Różnica jest taka, że przy zastosowaniu VLSM, segmentacja jest przeprowadzana na przestrzeni adresowej uprzednio przydzielonej danej organizacji i jest ona niewidoczna dla Internetu. Z kolei CIDR pozwala na przydzielanie bloku adresowego przez dany rejestr internetowy usługodawcy internetowemu (ISP) wyższego poziomu, który przydzieli segmenty ISP pośredniego poziomu. Ten ISP przeprowadzi dalszą segmentację swojej alokacji, aby dostarczyć blok adresowy ISP niskopoziomowemu, który następnie przydzieli adresy przedsiębiorstwu prywatnemu.

W związku z podobieństwami pomiędzy tymi technikami warunki wstępne dla pomyślnego wdrażania CIDR są takie same, jak dla VLSM, a mianowicie:

- Protokoły routingu muszą nieść rozszerzone informacje o prefiksach sieci wraz z każdym ogłoszeniem tras. Takie protokoły, jak RIPv2 i OSPF mają tę funkcję.
- Routery muszą implementować algorytm najdłuższego dopasowania.
- Adresy muszą być przydzielone tak, aby miały znaczenie topologiczne, umożliwiając w ten sposób zespalanie tras.

Dodatkowo, routery oraz zainstalowany system operacyjny (OS) muszą obsługiwać CIDR. W praktyce muszą być obsługiwane maski podsieci każdej długości. Niektóre stare systemy żądają klasowych domyślnych masek podsieci, podczas gdy inne przyjmują maski dłuższe niż domyślne (umożliwiając podział na podsieci oraz VLSM), ale nie przyjmują masek, które są krótsze (uniemożliwiając pełną implementację CIDR). TCP/IP systemu Windows 2000 zawiera obsługę podsieci zerowych i jedynkowych, tak jak opisano w dokumencie RFC 1878, i umożliwia pełną implementację CIDR.

Łączenie w nadsieć

Gdyby CIDR był implementowany od samego początku Internetu, prawdopodobnie nie stawalibyśmy przed takimi problemami związanymi z przestrzenią adresów IP, jakie teraz mamy. Jednak, kiedy CIDR był wprowadzany, istniała już duża, zainstalowana baza systemów klasowych. Dlatego też początkowym zastosowaniem CIDR stało się sklejanie kawałków przestrzeni *klasy C*, przy użyciu *łączenia w nadsieć*.

Łączenia w nadsieć można dokonywać w celu konsolidacji kilku sieci *klasy C* w jedną sieć logiczną. Technika ta nie musi być koniecznie ograniczona do adresów *klasy C*; sieci *klasy A* oraz *B* również można łączyć w nadsieć. Jednak przeprowadzanie łączenia w nadsieć *klasy B* jest rzadko wymagane, a jest już wysoce nieprawdopodobne, że kiedykolwiek zostaniesz wezwany do połączenia w nadsieć sieci *klasy A*.

Adresy sieciowe, które mają być połączone przy użyciu łączenia w nadsieć muszą dzielić ze sobą te same bity wysokiego poziomu. Oznacza to, że muszą one być przyległe — nie mógłbyś, na przykład łączyć 172.168.5.0 i 210.23.56.0. Przy łączeniu w nadsieć maska podsieci zostaje skrócona, aby zabrać bity z sieciowej części adresu i w zamian za to przydzielić je części *hosta*. Najlepiej ilustruje to przykład.

Przypuśćmy, że przedsiębiorstwu zostały przydzielone dwie sieci *klasy C*, 195.162.230.0/24 i 195.162.231.0/24. Dla wygody i dla zaoszczędzenia kosztów routera chciałbyś te sieci skleić, aby powstała z nich pojedyncza sieć mająca 510 zdatnych do użytku adresów. Rozwiązaniem w tym przypadku jest skrócenie maski podsieci o 1 bit, w taki sposób, że definicja CIDR twojej sieci staje się 195.162.230.0/23. Zobaczmy jak to wygląda w systemie binarnym:

```
ID pierwszej sieci     11000011 10100010 11100100 00000000 (195.162.228.0)
ID drugiej sieci       11000011 10100010 11100101 00000000 (195.162.229.0)
Maska podsieci         11111111 11111111 11111110 00000000 (255.255.254.0 [/23])
```

Bity sieciowe identyfikowane przez maskę sieci są identyczne. Zatem warunek dla zaistnienia sieci został spełniony i licząca 510 hostów sieć 195.162.228.0/23 jest określona zakresem adresów IP od 195.162.228.1 do 195.162.229.254. W obrębie tego zakresu zarówno 195.162.228.255, jak i 195.162.229.0 są ważnymi, zdatnymi do użytku adresami IP. (Muszę się przyznać do osobistej niechęci do ich stosowania, ale to pewnie z powodu mojego podeszłego wieku!)

Podobne obliczenia binarne powinny Cię przekonać, że 195.162.228.0/22 łączy cztery sieci *klasy C*, co daje sieć liczącą 1 022 hostów, mającą zakres adresów od 195.162.228.1 do 195.162.231.254. Podobnie 195.162.228.0/21 łączy osiem sieci *klasy C*, dając sieć liczącą 2 046 hostów, która ma zakres adresów od 195.162.228.1 do 195.162.235.254.

Ograniczenia łączenia w nadsieć związane z granicami

Jeżeli pomyślałeś, że opisany powyżej przykład był troszeczkę za prosty, aby mógł być prawdziwy albo przynajmniej dawał się powszechnie zastosować, to miałeś rację. Wartość w trzecim oktecie została starannie wybrana, tak aby wszystko działało. Rozważmy co by się stało, gdyby zamiast 195.162.228.0/24 i 195.162.229.0/24 przykładowemu przedsiębiorstwu przypisano 195.162.229.0/24 i 195.162.230.0/24. Jeśli podejmiemy próbę zastosowania podsieci */23*, otrzymamy następujący wynik:

```
ID pierwszej sieci     11000011 10100010 11100101 00000000 (195.162.229.0)
ID drugiej sieci       11000011 10100010 11100110 00000000 (195.162.230.0)
Maska podsieci         11111111 11111111 11111110 00000000 (255.255.254.0 [/23])
```

W tym przypadku bity sieci określane przez maskę podsieci */23* nie są identyczne, w związku z czym warunek spójnej sieci został naruszony. 195.162.229.0/23 nie jest więc ważną specyfikacją sieci.

Ogólnie rzecz biorąc, jeżeli chcesz połączyć dwie sieci *klasy C* z zastosowaniem łączenia w nadsieć, to sieci te muszą być przyległe, a wartość w trzecim oktecie pierwszej sieci musi być podzielna przez dwa. Jeżeli chcesz połączyć cztery sieci *klasy C*, to sieci te muszą być przyległe, a wartość w trzecim oktecie pierwszej sieci musi być podzielna przez cztery — i tak dalej. Podobnie, jeśli chcesz połączyć dwie sieci *klasy B*, to sieci te muszą być przyległe, a wartość w drugim oktecie pierwszej sieci musi być podzielna przez dwa.

Alokacja adresów w przypadku prywatnych intranetów

Dokument RFC 1918 zaleca, aby w przypadku hostów, które wymagają łączności IP, ale nie wymagają zewnętrznych połączeń z Internetem, przedsiębiorstwa korzystały z przestrzeni adresowej dla prywatnych intranetów przy użyciu określonych wewnętrznych adresów IP. IANA zarezerwowała następujące bloki adresów dla prywatnych intranetów:

- 10.0.0.0/8 (10.0.0.1 do 10.255.255.254)
- 172.16.0.0/12 (172.16.0.1 do 172.32.255.254)
- 192.168.0.0/16 (192.168.0.1 do 192.168.255.254)

Każde przedsiębiorstwo, które zdecyduje się na korzystanie z adresów z tych zarezerwowanych bloków, może to zrobić bez kontaktowania się z IANA, czy danym rejestrem internetowym. Ponieważ adresy te nie są nigdy routowane do Internetu, ta przestrzeń adresowa może być wykorzystywana jednocześnie przez wiele różnych przedsiębiorstw. Oczywiście nic nie może przeszkodzić danej organizacji w używaniu dowolnie wybranego systemu adresowego w swoim intranecie pod warunkiem, że nie ma możliwości, aby te wewnętrzne adresy były routowane do Internetu. Zaletą bloków zarezerwowanych jest to, iż zostaną one automatycznie odrzucone przez każdy router internetowy, w związku z czym nigdy nie będą przypadkowo routowane do Internetu.

Ten schemat adresowy (albo każdy inny schemat adresowania w prywatnym intranecie) wymaga, aby dana organizacja używała przy dostępie do Internetu serwera NAT. Jednak korzystanie z prywatnej przestrzeni adresowej oraz z serwera NAT ułatwia klientom zmianę swojego ISP bez potrzeby zmiany swojego adresu IP. Ponadto w obrębie dużego przedsiębiorstwa tylko pewna część (czasem mała część) pracowników potrzebuje stałego dostępu do Internetu. A zatem, pomimo że każdy host w przedsiębiorstwie ma swój własny adres IP pochodzący z prywatnego przydziału, określona musi być mniejsza liczba globalnych adresów internetowych, co zmniejsza popyt na przestrzeń adresową IPv4.

Wyczerpanie przestrzeni adresowej IPv4

Problem obecnego niedoboru oraz nieuchronnego wyczerpania przestrzeni adresowej IPv4 był poruszany na przestrzeni tego rozdziału. Z alokacji adresów IPv4 można wyciągnąć wiele wniosków, które miejmy nadzieję, zapobiegną przyszłemu marnotrawstwu brakujących zasobów (jeżeli takie wnioski w ogóle bywają wyciągane).

Po pierwsze, każdy z zasobów jest skończony i cenny. U zarania Internetu, przy istnieniu zaledwie kilku wojskowych i edukacyjnych sieci, ponad cztery miliony adresów internetowych musiało się wydawać źródłem prawie nieskończonym, ostatecznie odpornym

na przyszłość. Wskutek tego, jak również na skutek braku elastyczności adresowania klasowego, dokonywano alokacji sieci IP na podstawie żądań, a nie potrzeb. Przedsiębiorstwo zatrudniające parę tysięcy pracowników nie chciało kłopotu związanego z implementowaniem (powiedzmy) 10 sieci *klasy C* (szczególnie w czasach przed łączeniem w nadsieć), i dlatego prosiło i dostawało pełną sieć *klasy B*. Całe sieci */8*, takie jak 0.0.0.0, 127.0.0.0 i 255.0.0.0 nie nadają się do użytku z powodu sposobu, w jaki implementowane są funkcje domyślne, funkcje sprzężenia zwrotnego i funkcje emisji.

Przestrzeń adresowa 64.0.0.0/2 pozostaje nieprzydzielona w momencie pisania tej książki. W kwestii szczegółów odwołaj się do dokumentu RFC 1817.

Łączenie w nadsieć dawało większą elastyczność w przypadku wewnętrznej alokacji sieciowej, ale (być może) pogarszało sprawę, jeżeli chodzi o marnotrawstwo adresów. Łączenie w nadsieć może być bardzo rozrzutne. Weźmy przykład sieci *klasy B* (powiedzmy 154.12.0.0) z maską podsieci */19*. Daje to 3 bity podsieci, lub teoretycznie, osiem podsieci. Jednakże, jak widzieliśmy wcześniej, dwie z tych podsieci (same jedynki i same zera) nie mogą być wykorzystane. Dlatego też pierwszym adresem w sieci nadającym się do użytku jest 154.12.32.1, a ostatnim 154.12.223.254. Innymi słowy, jedna czwarta całkowitej puli adresowej sieci *klasy B* (ponad 16 000 adresów) nie może być wykorzystana. Jeżeli są wykorzystywane 2 bity podsieci (maska podsieci */18*), to połowa puli adresów *klasy B* staje się niezdatna do użycia.

Istnieje kilka inicjatyw mających na celu odzyskanie i ponowną alokację internetowej przestrzeni adresowej. Jak wspominałem we wcześniejszych częściach tego rozdziału, IANA uprasza o dobrowolny zwrot nie używanej przestrzeni adresowej w celu ponownej ich alokacji za pomocą CIDR. Inne grupy, takie jak zespół roboczy ds. procedur przenumerowywania Internetu/przedsiębiorstw (PIER) grupy roboczej do spraw sieci Internet (IETF), zajmują się takimi sprawami, jak prawo własności adresów a dzierżawa adresów. Grupa PIER jest również odpowiedzialna za zadanie opracowania strategii przenumerowywania.

W końcu jednak przestrzeń adresowa IPv4 się wyczerpie. Internet nie przestanie działać — po prostu za dużo zainwestowano już w e-gospodarkę, aby do tego dopuścić. Nie zniknie też IPv4. Zamiast tego będzie połączenie pomiędzy przestrzenią adresową IPv4, a przestrzenią adresową IPv6.

Adresy IPv6 są 128-bitowymi liczbami binarnymi. Teoretyczny rozmiar przestrzeni adresowej IPv6 to 2^{128}. Podanie tej liczby w systemie dziesiętnym byłoby bez sensu, ponieważ jest ona zbyt wielka, by ją pojąć.

Mówiono mi, że przestrzeń adresowa IPv6 jest zasobem prawie nieskończonym, który nigdy nie może się wyczerpać. Mówiono mi, że IPv6 jest całkowicie odporny na przyszłość. Mówiono mi, że są wnioski, których rodzaj ludzki nigdy nie wyciąga. Wierzę tylko w jedno z powyższych stwierdzeń.

Rozwiązania natychmiastowe

Budowanie wykresu podsieci

Obliczenia podziału na podsieci można wykonywać z zasad pierwszych, przy użyciu arytmetyki binarnej. Obliczenia te nie są szczególnie trudne, ale są one nużące i czasochłonne. Wielu fachowców od tworzenia sieci woli wygenerować schemat podsieci, który mogą potem wykorzystywać jako źródło odniesienia i zaoszczędzić zarówno na czasie, jak i na wysiłku wiążącym się z ciągłym powtarzaniem tych samych lub podobnych obliczeń.

Zauważysz, że powiedziałem „wygenerować" wykres, a nie „nauczyć się na pamięć". Jeśli nauczysz się, jak generować wykres i poznasz zasady, które leżą u podstaw jego budowy, to kilka minut pracy przyniesie upragnioną pomoc. Jeżeli spróbujesz nauczyć się go na pamięć, to będziesz miał trudności z przypominaniem, a twoje projekty sieciowe nie będą działały.

Opracowywanie maski podsieci

Przy podziale na podsieci wszystkie obliczenia biorą się z liczby bitów podsieci. Normalnie istnieje maksimum wynoszące 8 bitów podsieci. Może być więcej — sieć *klasy B* mogłaby być, przykładowo, podzielona na 510 podsieci liczących po 126 hostów — ale taki poziom podziału jest rzeczą niezwykłą. Bity podsieci nie mogą być samymi jedynkami, ani samymi zerami. Dlatego też może być tylko 1 bit podsieci. Zakres bitów podsieci z praktycznego punktu widzenia wynosi zatem 2 do 8.

Aby opracować maskę podsieci dla danej liczby bitów podsieci, wykonaj następujące czynności:

1. Określ, czy sieć jest siecią *klasy A*, *B*, czy *C*.
2. Weź domyślną maskę podsieci (odpowiednio */8*, */16*, lub */24*) i dodaj liczbę bitów podsieci. W ten sposób sieć *klasy B* (*/16*) mająca 3 bity podsieci ma maskę podsieci */19*.
3. Aby obliczyć maskę podsieci w kropkowej notacji dziesiętnej, weź pierwszy oktet zerowy domyślnej maski podsieci. W przypadku *klasy B* (255.255.0.0) jest to trzeci oktet.
4. Przekształć najbardziej znaczące bity tego oktetu na jedynki, aby pasowały do bitów maski podsieci. To znaczy, jeżeli są 3 bity maski podsieci, to przekształć 3 pierwsze bity oktetu na jedynki.
5. Oblicz dziesiętną wartość oktetu, zważywszy że binarne 10000000 równa się 128, 01000000 równa się 64 i tak dalej.
6. Z tych obliczeń wygeneruj tabelę 4.3.

Tabela 4.3. *Opracowywanie maski podsieci*

Bity podsieci	Maska
2	192
3	224
4	240
5	248
6	252
7	254
8	255

Większość ludzi uważa, że łatwiej jest zacząć od dołu tej tabeli i pracować w górę.

Obliczanie liczby podsieci

Liczbę podsieci można obliczyć bezpośrednio z liczby bitów podsieci. Wszystko, o czym musisz tutaj pamiętać, to żeby odjąć 2, ponieważ bity nie mogą być samymi jedynkami albo samymi zerami. Aby obliczyć liczbę podsieci, wykonaj następujące działania:

1. Oblicz 2^x, gdzie *x* stanowi liczbę bitów podsieci (2^2=4, 2^3=8, 2^4=16 i tak dalej).
2. Odejmij 2 od każdej z tych liczb.
3. Dołącz wyniki do tabeli 4.3, aby wygenerować tabelę 4.4.

Tabela 4.4. *Dodawanie liczby podsieci*

Bity podsieci	Maska	Podsieci
2	192	2
3	224	6
4	240	14
5	248	30
6	252	62
7	254	126
8	255	254

Obliczanie przyrostu

Przyrost jest wartością wykorzystywaną do obliczania zakresu adresów w każdej z podsieci. Reprezentuje ona różnicę, albo *skok*, w obrębie odpowiedniego oktetu, (drugiego w przypadku *klasy A*, trzeciego w przypadku *klasy B*, czwartego w przypadku *klasy C*) pomiędzy adresami początkowymi dla każdej z podsieci. W przykładzie, który wypracowywaliśmy w pierwszym podrozdziale w dzielonej na podsieci sieci *klasy B* 131.11.0.0/21 (5 bitów podsieci), pierwszy adres w pierwszej podsieci to 131.11.8.1, a pierwszy adres w drugiej podsieci to 131.11.16.1. A zatem przyrost wynosi 8.

Jeżeli ta sama sieć zostanie podzielona na podsieci z 3 bitami podsieci (*/19*), to pierwszy adres w pierwszej podsieci to 131.11.64.1, pierwszy adres w drugiej podsieci to 131.11.128.1, a zatem przyrost wynosi 64.

Jest to dość złożone i czasochłonne obliczenie binarne. Na szczęście jest bardzo prosta metoda obliczania przyrostu:

1. Weź uprzednio obliczoną wartość oktetu z maski podsieci.
2. Odejmij tę wartość od 256.
3. Dodaj wartości przyrostu do tabeli 4.4, aby wygenerować tabelę 4.5.

Tabela 4.5. *Dodawanie wartości przyrostu*

Bity podsieci	Maska	Podsieci	Przyrost
2	192	2	64
3	224	6	32
4	240	14	16
5	248	30	8
6	252	62	4
7	254	126	2
8	255	254	1

Obliczanie liczby hostów na podsieć

Obliczanie liczby hostów na podsieć jest proste, nawet w systemie binarnym. Aby obliczyć liczbę hostów, wykonaj następujące kroki:

1. Weź liczbę bitów domyślnie przydzielonych tożsamościom hostów (24 dla *klasy A*, 16 dla *klasy B*, 8 dla *klasy C*).
2. Odejmij liczbę bitów podsieci, aby otrzymać wartość *y*.
3. Oblicz 2^y dla każdego rzędu w tabeli.
4. Odejmij 2 od każdej wartości (ponieważ adresem hosta nie mogą być same jedynki ani same zera).
5. Dodaj uzyskane liczby hostów do tabeli 4.5, aby uzyskać wykres podsieci przedstawiony w tabeli 4.6 Zazwyczaj nie ma potrzeby dokładnego obliczania liczby hostów powyżej 510; dlatego też stosuje się przybliżenia.

Jest mało prawdopodobne, żebyś wypracowywał każdą wartość liczby hostów od zera. Kiedy już wypracujesz jedną, albo co najwyżej jedną z każdej klasy, zasada staje się dość oczywista (dodać 2, podwoić, odjąć 2). Ja zaczynam od *klasy B*, liczby bitów podsieci równej osiem. Ponieważ *klasa B* z 8 bitami podsieci ma taką samą maskę sieci, jak domyślna *klasa C*, wiem, że ma 254 hosty. Zazwyczaj jestem w stanie obliczyć resztę od tego miejsca.

Tabela 4.6. *Wykres podsieci*

Bity podsieci	Maska	Podsieci	Przyrost	Hosty klasy A	Hosty klasy B	Hosty klasy C
2	192	2	64	4M	16K	62
3	224	6	32	2M	8K	30
4	240	14	16	1M	4K	14
5	248	30	8	500K	2K	6
6	252	62	4	250K	1K	2
7	254	126	2	130K	510	–
8	255	254	1	65K	254	–

Dzielenie sieci klasy A na podsieci

Duże przedsiębiorstwa czasem używają sieci *klasy A* (szczególnie 10.0.0.0) w intranetach firmowych. Dlatego też może się zdarzyć, że zostaniesz wezwany do wykonania podziału sieci *klasy A* na podsieci, choć jest mało prawdopodobne, że będzie to jakaś globalna sieć internetowa.

Zadaniem tej procedury jest opisanie techniki, a nie odzwierciedlenie sytuacji z życia wziętej. W praktyce jest bardzo prawdopodobne, że nawet największe przedsiębiorstwo będzie wykorzystywało tylko pewną część przestrzeni adresowej 10.0.0.0/8 i będzie wykorzystywało VLSM do dalszej segmentacji swoich podsieci szkieletowych. Kiedy zaznajomisz się z tą techniką, możesz stosować ją wobec dowolnej sieci IP, łącznie z podsiecią, która wymaga dodatkowej segmentacji.

Pewne międzynarodowe przedsiębiorstwo wymaga ogólnej liczby 70 podsieci. Chociaż większość z nich będzie względnie małych, dyrekcja przewiduje zapotrzebowanie w wysokości 80 tys. hostów w jednej z nich. Korzystasz ze specyfikacji wewnętrznego adresu intranetowego 10.0.0.0/8 (RFC 1918). Kierownik techniczny chce wiedzieć, czy hosty 10.2.4.213 i 10.6.1.14 będą w tej samej podsieci. Aby zaimplementować wymaganą strukturę podsieci, podejmij następujące działania:

1. Wybierz liczbę bitów podsieci. Według tabeli 4.6 wybór 7 bitów podsieci daje 126 sieci, co spełnia wymogi i pozostawia miejsce na rozbudowę.
2. Sprawdź liczbę hostów na podsieć. Sieć *klasy A*, która ma 7 bitów podsieci, dopuszcza 130 tys. hostów na sieć. Spokojnie mieści się to w granicach wymogów.
3. Oblicz maskę podsieci. Według tabeli 4.6, wartość drugiego oktetu (jako że jest to sieć *klasy A*) wynosi 254. Zatem maska podsieci to 255.254.0.0 (lub */15*).
4. Zastosuj przyrost. Według tabeli 4.6 wynosi on 2. Zatem podsieci to 10.2.0.0/15, 10.4.0.0/15, 10.6.0.0/15 i tak dalej.
5. Dodaj zakresy adresów hostów. Adresy hostów nie mogą być samymi jedynkami, ani samymi zerami, więc zakresy adresów to 10.2.0.1 do 10.3.255.254, 10.4.0.1 do 10.5.255.254, 10.6.0.1 do 10.7.255.254 i tak dalej.
6. Skontroluj strukturę sieci, którą uzyskałeś. Host 10.2.4.213 jest w sieci 10.2.0.0, a host 10.6.1.14 jest w sieci 10.6.0.0. A zatem nie są one w tej samej podsieci.

Dzielenie sieci klasy B na podsieci

Zazwyczaj przedsiębiorstwo, któremu została przydzielona sieć *klasy B* lub zaimplementowało prywatną sieć wewnętrzną *klasy B* w swoim intranecie, potrzebuje podziału na podsieci.

Pewne przedsiębiorstwo aktualnie wymaga 28 podsieci w swojej sieci *klasy B*, 155.62.0.0. Obecnie maksymalna liczba hostów w każdej z podsieci wynosi 250 i jest mało prawdopodobne, aby liczba ta miała przekroczyć 500 w najbliższej przyszłości. Istnieje wymóg, aby hosty 155.62.10.6 i 155.62.15.230 nie dzieliły ze sobą tej samej podsieci. Aby implementować wymaganą strukturę podsieci, wykonaj następujące czynności:

1. Wybierz liczbę bitów podsieci. Według tabeli 4.6 wybór zarówno 5 bitów podsieci (30 podsieci), jak i 6 bitów sieci (62 podsieci) spełnia wymogi, przy czym druga z opcji daje więcej miejsca na przyszłą rozbudowę.
2. Oblicz liczbę hostów przypadających na podsieć. Jeżeli wybierzesz 5 bitów podsieci, to każda z podsieci będzie w stanie pomieścić w przybliżeniu 2 000 hostów. Wybór 6 bitów podsieci ogranicza maksymalną liczbę hostów na podsieć do około 1 000. Obydwie liczby spokojnie mieszczą się w granicach wymogów.
3. Zastosuj przyrost. Dla 5 bitów podsieci jest to 8, dla 6 bitów podsieci — 4. Stąd też wybór podsieci to:
   - *5 bitów podsieci* — 155.62.8.0/21, 155.62.16.0/21, 155.62.24.0/21 i tak dalej;
   - *6 bitów podsieci* — 155.62.4.0/22, 155.62.8.0/22, 155.62.12.0/22 i tak dalej.
4. Zastosuj wymóg sformułowany w specyfikacji. Jeżeli wybierzesz 5 bitów podsieci, to hosty 155.62.10.6 i 155.62.15.230 będą razem w sieci 155.62.8.0/21. Jeżeli jednak wybierzesz 6 bitów podsieci, to będą one — odpowiednio — w podsieciach 155.62.8.0/22 i 155.62.12.0/22. Dlatego też wybór padnie na 6 bitów podsieci.
5. Uzyskaj maskę podsieci. Według tabeli 4.6, wartość trzeciego oktetu (jako że jest to sieć *klasy B*) wynosi 252. A zatem maska sieci to 255.255.254.0.0 (lub */22*).
6. Dodaj zakresy adresów hostów. Adresy hostów nie mogą być samymi jedynkami, ani samymi zerami, więc zakresy adresów to 155.62.4.1 do 155.62.7.254, 155.62.8.1 do 155.62.11.254, 155.62.12.1 do 155.62.15.254 i tak dalej.

Dzielenie sieci klasy C na podsieci

Zazwyczaj bardziej prawdopodobne jest, że sieć *klasy C* poddana zostanie łączeniu w nadsieć, a nie podziałowi na podsieci. Jednak małe przedsiębiorstwo może być podzielone na kilka grup, z których każda wymaga swojej własnej sieci.

Pewna mała firma wymaga ogólnej liczby czterech sieci. W żadnej z tych podsieci nigdy nie będzie więcej, niż 20 hostów. Została im przydzielona sieć *klasy C* 195.162.230.0/24. Aby implementować wymaganą strukturę sieciową, wykonaj następujące kroki:

1. Wybierz liczbę bitów podsieci. Według tabeli 4.6 wybór 3 bitów podsieci daje 6 sieci liczących maksymalnie po 30 hostów. Spełni to wymogi.
2. Uzyskaj maskę podsieci. Według tabeli 4.6, wartość czwartego oktetu (jako że jest to sieć *klasy C*) wynosi 224. Zatem maska podsieci to 255.255.255.224 (lub */27*).
3. Zastosuj przyrost. Według tabeli 4.6 wynosi on 32. A zatem podsieci to 195.162.230.32/27, 195.162.230.64/27, 195.162.230.97/27 i tak dalej.
4. Dodaj zakresy adresów hostów. Adresy hostów nie mogą być samymi jedynkami, ani samymi zerami, więc zakresy adresów to 195.162.230.33 do 195.162.230.62, 195.162.230.65 do 195.162.230.94, 195.162.230.97 do 195.162.230.126 i tak dalej.

Wartości ostatniego oktetu w przypadku dzielenia na podsieci w *klasie C* mogą czasem powodować zamieszanie, ponieważ jednocześnie zastosowuje się przyrost oraz adres hosta (przy ograniczeniach związanych z samymi jedynkami i samymi zerami) wobec tego samego oktetu. Jeżeli sprawia ci to kłopot, zapisz to sobie w systemie binarnym. Zajmujesz się tylko 8 bitami, więc zapis binarny nie będzie wyglądał zniechęcająco.

Dzielenie segmentu VLSM na podsieci

Podział na podsieci w środowisku VLSM rządzi się tymi samymi zasadami, co zwyczajny podział na podsieci. Niniejsza procedura pokazuje zarówno podział na podsieci segmentu VLSM, jak i podział na podsieci poprzez granice klas.

Pewne przedsiębiorstwo zaimplementowało podział na podsieci w sieci *klasy B*, tak jak opisano w poprzedniej procedurze. Informatycy chcą dokonać dalszego podziału podsieci 155.62.12.0/22 na największą możliwą liczbę pod-podsieci, biorąc pod uwagę wymóg, że w każdej z podsieci może być maksymalnie 40 hostów. Wewnętrzny podział podsieci szkieletowej wymaga, aby został wdrożony VLSM. Sprawdzono, że przedsiębiorstwo korzysta z protokołu routingu, który niesie rozszerzone informacje o prefiksie sieci wraz z każdym ogłoszeniem trasy, i że routery sieci implementują algorytm najdłuższego dopasowania.

Aby dokonać dalszej segmentacji podsieci szkieletowej 155.62.12.0/22, wykonaj następujące czynności:

1. Według tabeli 4.6 określ podsieć, która spełnia wymóg ograniczenia do 40 hostów. Jest to sieć *klasy C* posiadająca 2 bity podsieci (maksymalna liczba 62 hostów).
2. Uzyskaj maskę podsieci dla tej podsieci. Zgodnie z zasadami podziału na podsieci wykorzystywanymi we wszystkich pozostałych procedurach, maska ta określona jest jako 255.255.255.192, lub */26*.
3. Wyznacz przyrost. Jako że przekroczyliśmy granicę klas na rzecz *klasy C*, przyrost stosuje się do czwartego oktetu adresu. Według tabeli 4.6 wynosi on 64.
4. Zastosuj przyrost. Podsieci to 155.62.12.64/26, 155.62.12.128/26, 155.62.12.192/26, 155.62.13.0/26 i tak dalej, aż do 155.62.15.128/26.

5. Dodaj tożsamości hostów. Daje to zakresy adresów 155.62.12.65 do 155.62.12.126, 155.62.12.129 do 155.62.12.190, 155.62.12.193 do 155.62.12.254, 155.62.13.1 do 155.62.13.62 i tak dalej.
6. Aby obliczyć maksymalną liczbę podpodsieci, odejmij maskę podsieci szkieletowej (*/22*) od maski podpodsieci (*/26*). W dłuższym z tych prefiksów są cztery dodatkowe bity podsieci. Liczba podpodsieci wynosi zatem 2^4–2, czyli 14.

Łączenie sieci klasy C w nadsieć

Obliczenia związane z łączeniem w nadsieć są proste. Jedynym limitem, o którym musisz pamiętać jest limit granicy. Jeżeli chcesz połączyć dwie sieci *klasy C* w nadsieć, to wartość trzeciego oktetu niższego adresu musi być podzielna przez 2. Jeżeli chcesz połączyć w nadsieć cztery sieci, to wartość ta musi być podzielna przez 4 i tak dalej. Sieci muszą być przyległe i poddaje się je łączeniu w nadsieć w grupach po 2, 4, 8, 16 i tak dalej (potęgi liczby dwa).

Pewnemu przedsiębiorstwu przydzielone zostały cztery sieci *klasy C*, 207.23.68.0 do 207.23.71.0, i pragnęłoby ono połączyć je w pojedynczą sieć. Sprawdź, czy to jest możliwe i oblicz maskę podsieci oraz zakres adresów.

1. Sprawdź, czy sieci są przyległe (są przyległe) i czy wartość trzeciego oktetu najniższej sieci (68) jest podzielna przez 4 (jest podzielna). A zatem sieci te mogą zostać połączone.
2. Weź domyślną maskę podsieci *klasy B* (*/24*) i skróć ją o odpowiednią liczbę bitów. Aby połączyć dwie sieci — skróć ją o jeden; aby połączyć cztery — skróć ją o dwa; aby połączyć osiem — skróć ją o trzy, i tak dalej. W naszym przypadku skracamy ją o dwa. Stąd maska podsieci to */22*, lub 255.255.252.0.
3. A zatem połączona sieć to 207.23.68.0/22. Dodaj tożsamości hostów, aby otrzymać zakres adresów od 207.23.68.1 do 207.23.71.254.