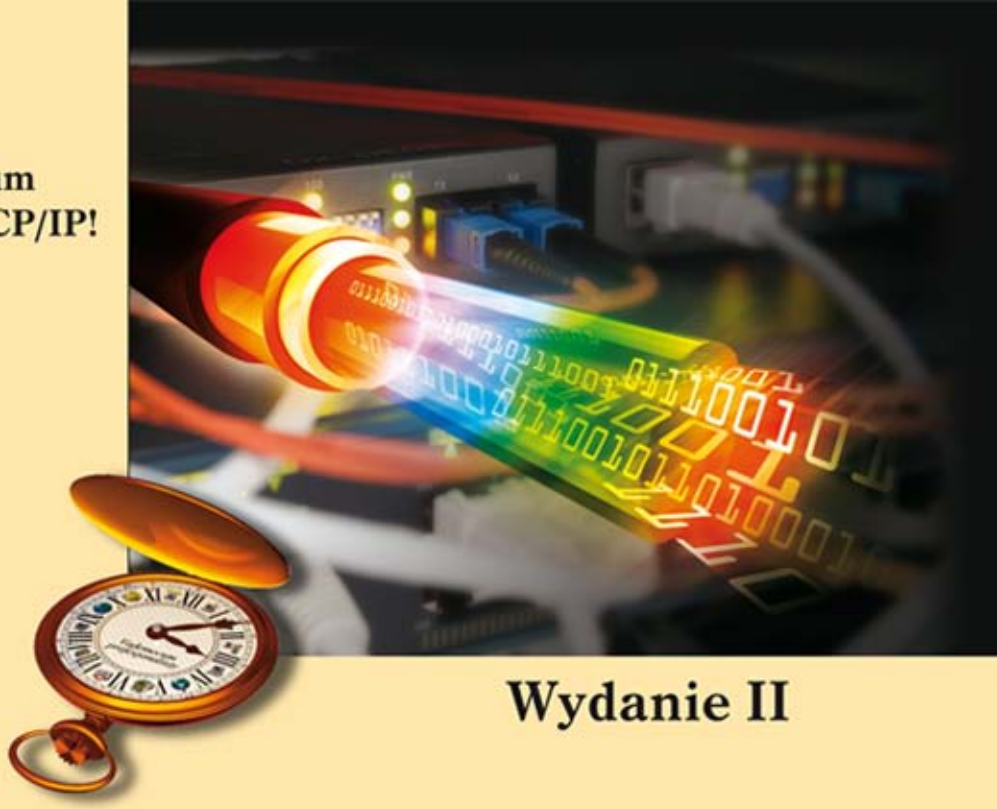
Kompendium wiedzy o TCP/IP!



Wydanie II

TCP/IP od środka Protokoły

Vademecum profesjonalisty

ADDISON-WESLEY

Helion

Kevin R. Fall
W. Richard Stevens

Tytuł oryginału: TCP/IP Illustrated, Volume 1: The Protocols (2nd Edition)

Tłumaczenie: Andrzej Grażyński (wstęp, rozdz. 1 – 9),
Marek Pałczyński (rozdz. 10 – 11, 18), Grzegorz Pawłowski (rozdz. 12 – 17, dodatek)
Projekt okładki: Maciej Pasek

Materiały graficzne na okładce zostały wykorzystane za zgodą Shutterstock Images LLC.

ISBN: 978-83-246-4815-3

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: helion@helion.pl
WWW: http://helion.pl (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie/tcppr2
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

Spis treści

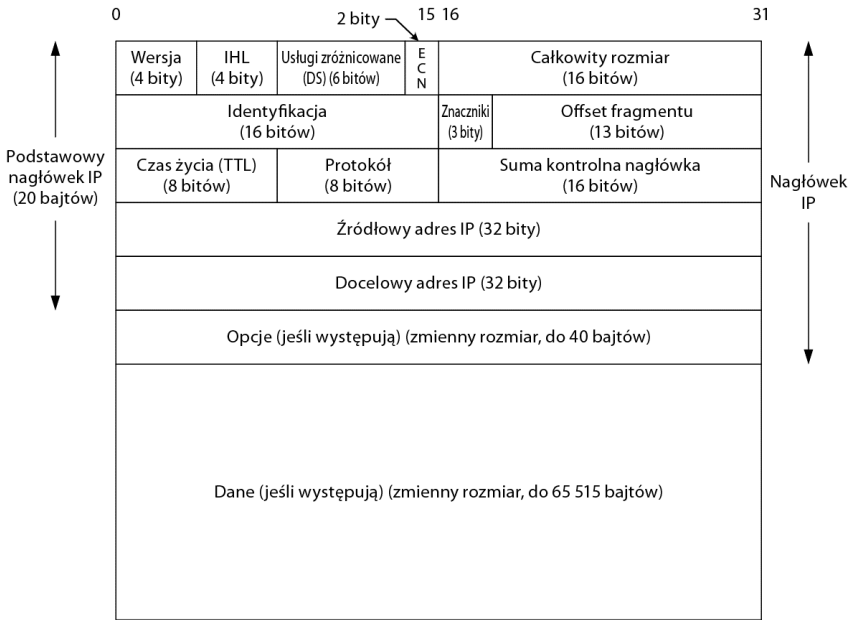
Rozdział 5.
Protokół internetowy (IP)

5.1. Wprowadzenie

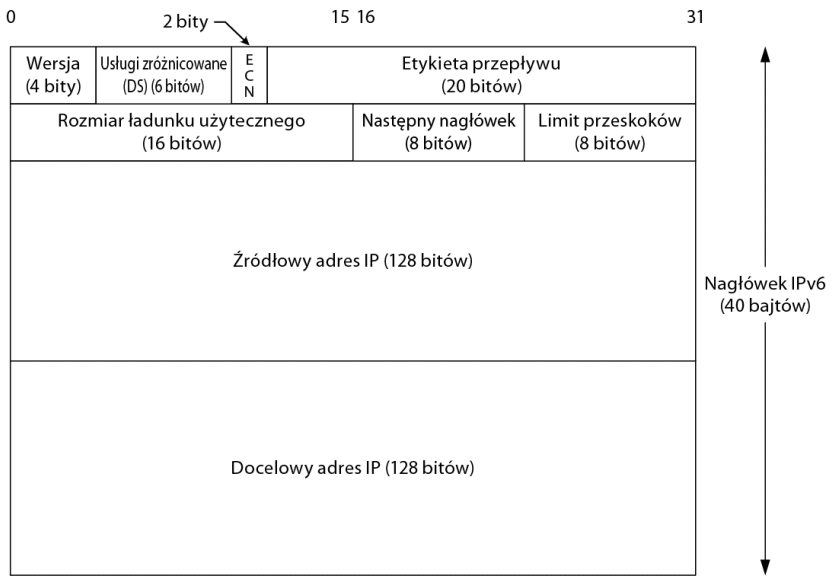
Protokół internetowy (*Internet Protocol*), znany powszechnie pod akronimem IP, jest motorem napędowym całego zestawu protokołów TCP/IP. Dane protokołów TCP, UDP, ICMP i IGMP transmitowane są właśnie w postaci datagramów IP. Protokół IP zapewnia usługę niegwarantowanego (*best-effort*), bezpołączeniowego (*connectionless*) transferu datagramów — przymiotnik „niegwarantowany" oznacza tu, że mimo „najlepszych starań" (bo tak tłumaczy się *best-effort*) wysłanie pakietu nie daje gwarancji, że pakiet ten dotrze do zamierzonego celu. Nie oznacza to, rzecz jasna, że gubienie pakietów jest naturalną funkcją protokołu IP: po prostu, po wysłaniu pakietu protokół (a właściwie jego implementacja w routerze) nie troszczy się już o jego los, a niemożność przetworzenia odebranego pakietu (np. z powodu chwilowego przepełnienia buforów routera) albo też stwierdzenie jego przekłamania „po drodze" kwitowane są banalną reakcją routera — odrzuceniem pakietu. Zasada ta wspólna jest dla obu wersji protokołu — IPv4 i IPv6. Wynika z niej natychmiast zasada pokrewna: to na (wymienionych wcześniej) protokołach warstwy wyższej spoczywa obowiązek zweryfikowania, czy zadanie zlecone protokołowi IP zostało przezeń poprawnie wykonane.

Przymiotnik „bezpołączeniowy" oznacza, że przesyłanie datagramów między dwoma węzłami sieci (routerami) nie wiąże się z uprzednim negocjowaniem jakiegoś porozumienia między tymi węzłami. Każdy docierający do routera datagram traktowany jest niezależnie od innych datagramów, a jego przetwarzanie opiera się wyłącznie na informacji zawartej wewnątrz niego, bez kontekstu jakiejkolwiek innej informacji o stanie transferu. W szczególności oznacza to, że wysyłane datagramy docierać mogą do miejsca przeznaczenia w kolejności innej niż kolejność ich wysyłania, bo podążać mogą do tegoż miejsca różnymi drogami. Możliwe jest także *powielanie* (duplikowanie) datagramów oraz *zniekształcanie* ich treści („przekłamywanie") na skutek różnych czynników fizycznych oddziałujących na nośniki transmisyjne. Z tym wszystkim liczyć się muszą protokoły zlecające protokołowi IP transfer swoich danych.

Szczegóły transferu danych enkapsulowanych w pakietach IP określone są przez *nagłówki* (*headers*) tych pakietów, różniące się istotnie w obu wersjach. Na rysunku 5.1 przedstawiono strukturę pakietu w wersji IPv4, natomiast nagłówek pakietu IPv6 widoczny jest na rysunku 5.2; oficjalna specyfikacja protokołu IPv4 zawarta jest w dokumencie [RFC0791], protokołowi IPv6 poświęcona jest natomiast cała seria dokumentów, zapoczątkowana przez [RFC2460].



Rysunek 5.1. *Datagram IPv4. Nagłówek ma zmienny rozmiar, do 60 bajtów (15 słów 32-bitowych, na tyle pozwala 4-bitowe pole IHL); w (typowym) przypadku braku opcji rozmiar ten wynosi 20 bajtów. Adresy IP źródłowy i docelowy zapisane są w standardowej postaci w słowach 32-bitowych. Pola Identyfikacja i Offset fragmentu związane są z funkcją fragmentowania pakietów IPv4. Suma kontrolna, chroniąca integralność pól nagłówka, obliczana jest wyłącznie dla obszaru nagłówka, bez związku z polem danych (których integralność nie jest w związku z tym chroniona)*



Rysunek 5.2. *Nagłówek pakietu IPv6 ma ustalony rozmiar 40 bajtów. Pole Następny nagłówek służy do wiązania w ciąg ewentualnych nagłówków rozszerzeń, które występować mogą w pakiecie po nagłówku podstawowym, zwykle bezpośrednio przed danymi protokołu warstwy wyższej. Oba adresy IP — źródłowy i docelowy — zapisane są w binarnej postaci 128-bitowej*

5.2. Nagłówki IPv4 i IPv6

Widoczny na rysunku 5.1 nagłówek datagramu IPv4 ma zasadniczo rozmiar 20 bajtów, lecz rozmiar ten może być większy, jeśli wspomniany nagłówek zawiera dodatkowe *opcje* (co zdarza się raczej rzadko). Przedstawiony na rysunku 5.2 nagłówek datagramu IPv6 ma *zawsze* rozmiar 40 bajtów; opcjonalne funkcje związane z datagramem reprezentowane są w postaci łańcucha nagłówków rozszerzeń, którymi zajmiemy się szczegółowo w dalszym ciągu tego rozdziału.

We wszystkich prezentowanych diagramach bity numerowane są w kolejności *malejącego* znaczenia: najbardziej znaczący bit ma numer 0. W 32-bitowym słowie najmniej znaczący bit ma numer 31, zaś poszczególne bajty słowa uszeregowane są w ten sposób, że najbardziej znaczący bajt znajduje się pod najmniejszym adresem, tak więc np. ciąg bajtów o wartości `0xDE`, `0xAF`, `0xC0`, `0xDA` (w kolejności wzrastających adresów) tworzy słowo o wartości `0xDEAFC0DA`. Taka konwencja traktowania słów wielobajtowych nosi nazwę *big-endian*[1] i wymagana jest dla wszystkich binarnych liczb całkowitych występujących w transmitowanych przez sieć nagłówkach protokołów TCP/IP (z tego względu konwencję tę nazywa się *sieciową kolejnością bajtów — network byte order*). Uporządkowanie odwrotne, w którym najbardziej znaczący bajt znajduje się pod najwyższym adresem, nosi nazwę *little-endian* i notabene stosowane jest przez znacznie więcej procesorów (m.in. przez procesory serii x86 Intela); cytowany ciąg bajtów `0xDE`, `0xAF`, `0xC0`, `0xDA` tworzy więc w konwencji *little-endian* słowo `0xDAC0AFDE`. W komputerach wykorzystujących procesory z uporządkowaniem *little-endian* konieczna jest więc konwersja uporządkowania bajtów nagłówka pakietu przed jego wysłaniem w sieć oraz po jego odebraniu z sieci.

5.2.1. Pola nagłówków IP

Pierwszym polem nagłówka datagramu obu wersji jest czterobitowe pole *Wersja*, zawierające wartość `4` (binarnie `0100`) dla datagramu IPv4 i wartość `6` (binarnie `0110`) dla datagramu IPv6. Pierwsze cztery bity datagramu umożliwiają więc rozpoznanie jego wersji — rozpoznanie bardzo ważne, bowiem na polu *Wersja* kończy się zgodność obu formatów i każdy z nich wymaga innego przetwarzania przez host lub router. Mimo iż opracowano i zaproponowano (w charakterze kandydatury do standardu) kilka jeszcze innych wersji protokołu IP (ich wykaz, wraz z odsyłaczami, dostępny jest na stronie [IV]), w użyciu są niemal wyłącznie wersje o numerach 4 i 6.

Pole *IHL* (od *Internet Header Length* — rozmiar nagłówka internetowego) określa rozmiar nagłówka IPv4 liczony w 32-bitowych słowach. Ponieważ na 4 bitach pola można zapisać maksymalnie liczbę 15, rozmiar nagłówka IPv4 ograniczony jest do 60 bajtów; w dalszym ciągu rozdziału pokażemy, jak ograniczenie to niweluje w praktyce użyteczność wielu opcji, m.in. opcji *Rejestracja trasy* (*Record Route*). Dla nagłówka pozbawionego opcji pole to ma wartość `5`. Ponieważ nagłówek IPv6 ma ustalony rozmiar (40 bajtów), nie występuje w nim analogiczne pole.

[1] Czytelnikom zainteresowanym genezą tej nazwy oraz innymi szczegółami uporządkowania bajtów w słowach wielobajtowych polecam lekturę strony *http://pl.wikipedia.org/wiki/Kolejność_bajtów — przyp. tłum.*

Zgodnie z oryginalną specyfikacją [RFC0791] definiującą protokół IPv4, kolejnym polem w nagłówku IPv4 jest pole definiujące *Typ usługi* (ToS — *Type of Service*); w nagłówku IPv6 analogiczne pole nosi nazwę *Klasa ruchu* (*Trafic Class*), zgodnie ze specyfikacją [RFC2460]. Ponieważ żadne z tych pól nie okazało się szczególnie użyteczne, każde z nich zostało podzielone (w jednakowy sposób w obu wersjach) na mniejsze podpola (który to podział jest przedmiotem rozważań wielu dokumentów RFC, m.in. [RFC3260], [RFC3168] i [RFC2474]). Pierwsze z podpól stanowiących wynik tego podziału zajmuje 6 bitów i nosi nazwę *Usługi Zróżnicowane* (*Differentiated Services*, w skrócie DS), pozostałe dwa bity tworzą podpole o nazwie ECN (od *Explicit Congestion Notification* — jawne powiadomienie o przeciążeniu). Rolę obu podpól w forwardowaniu datagramów omówimy szczegółowo w punkcie 5.2.3.

Pole *Całkowity rozmiar* w nagłówku IPv4 specyfikuje całkowity rozmiar datagramu w bajtach — maksymalnie 65 535, bo tyle można zapisać na 16 bitach. W połączeniu z polem *IHL* umożliwia ono określenie rozmiaru danych stanowiących ładunek użyteczny (miejsce, w którym kończy się nagłówek, a zaczynają wspomniane dane, wyznaczone jest wprost przez zawartość pola *IHL*). Konieczność jawnego wskazania rozmiaru datagramu IPv4 (w polu *Całkowity rozmiar*) wynika z faktu, że niektóre z protokołów warstw niższych stosują zabiegi prowadzące do utraty informacji o tym rozmiarze; koronnym tego przykładem jest Ethernet, wydłużający zbyt małe ramki do minimalnej wymaganej długości 64 bajtów (o czym pisaliśmy w rozdziale 3.) — tak więc ładunek użyteczny ramki w postaci 20-bajtowego datagramu IPv4 zostanie sztucznie uzupełniony do długości 64 bajtów bez wskazania miejsca, w którym rozpoczyna się owo dopełnienie, a kończy się rzeczywisty datagram IP.

Mimo iż wielkość pola *Rozmiar całkowity* (16 bitów) wyznacza 65 535 bajtów jako maksymalny rozmiar datagramu IPv4, większość technologii warstwy łącza danych (w tym Ethernet) nie radzi sobie z tak dużymi datagramami w jednym kawałku, w związku z czym datagramy te podlegają dzieleniu na mniejsze porcje, zwane **fragmentami** (proces tego podziału nosi nazwę **fragmentacji**); ponadto, zgodnie ze specyfikacją IPv4, nie można wymagać od hostów zdolności przetwarzania otrzymywanych datagramów o rozmiarze przekraczającym 576 bajtów (implementacja IPv6 musi natomiast radzić sobie z przetwarzaniem datagramów o wielkości równej MTU łącza transferującego te datagramy IP, przy czym minimalny rozmiar ramki wynosi 1280 bajtów). Wiele aplikacji wykorzystujących protokół UDP (patrz rozdział 10.) do transferu danych (m.in. DNS i DHCP) we własnym zakresie limituje datagramy transportowe do rozmiaru 512 bajtów, automatycznie czyniąc tym samym zadość wspomnianemu ograniczeniu do 576 bajtów. Protokół TCP stosuje własną politykę doboru rozmiaru datagramów IP (patrz rozdział 15.).

Gdy datagram IPv4 podzielony zostaje na kilka fragmentów, każdy z tych fragmentów jest niezależnym datagramem, wartość w polu *Rozmiar całkowity* jest więc rozmiarem fragmentu, a nie oryginalnego datagramu przed podziałem. Szczegółami fragmentacji zajmiemy się w rozdziale 10., przy okazji omawiania protokołu UDP. W datagramie IPv6 fragmentacja nie znajduje odzwierciedlenia w nagłówku; ponadto zamiast rozmiaru całego pakietu w polu *Rozmiar ładunku użytecznego* znajduje się liczba bajtów zajętych przez transmitowane dane oraz nagłówki rozszerzeń, bez uwzględniania jednakże 40-bajtowego nagłówka podstawowego. Tak jak w datagramie IPv4, wielkość pola (16 bitów) ogranicza wspomniany rozmiar do 65 535 bajtów, lecz protokół IPv6 zapewnia także opcję obsługi większych ładunków użytecznych (do 4 GB) w ramach datagramów zwanych jumbogramami (patrz podpunkt 5.3.1.2).

Pole *Identyfikacja* ułatwia protokołowi IPv4 identyfikowanie poszczególnych datagramów wysyłanych przez host. Pole to zwiększane jest systematycznie o 1 w datagramach kolejno wysyłanych przez host. Pole to ma kluczowe znaczenie w sytuacji fragmentowania datagramów, w połączeniu z polami *Znaczniki* i *Offset Fragmentu*; szczegółowo zajmiemy się tym tematem w rozdziale 10. W protokole IPv6 fragmentacja datagramu odzwierciedlana jest w jednym z nagłówków rozszerzenia — powrócimy do tej kwestii w punkcie 5.3.3.

Pole *Czas życia* w IPv4 (i jego odpowiednik *Limit przeskoków* w IPv6) zawiera liczbę przeskoków, czyli routerów, jakie datagram ma prawo jeszcze przemierzyć na swej trasie do miejsca przeznaczenia. Jest ono inicjowane przez nadawcę pewną wartością początkową — dokument [RFC1122] zaleca w tej roli wartość 64, choć często spotyka się także 128 i 255. Pole to jest dekrementowane (zmniejszane) o 1 przy przetwarzaniu w kolejnym routerze; datagram z zerową wartością tego pola jest odrzucany przez router w momencie odebrania, a nadawca powiadamiany jest o tym fakcie za pomocą odpowiedniego komunikatu ICMP (o czym piszemy w rozdziale 8.). Mechanizm ten zapobiega niepożądanym sytuacjom krążenia w nieskończoność po sieci „zapętlonych" datagramów.

W oryginalnej specyfikacji pole *Czas życia* określać miało maksymalny czas życia datagramu w sekundach (stąd nazwa); router po przetworzeniu datagramu zmniejszał to pole o wartość równą liczbie sekund, jakie poświęcił na przetworzenie pakietu; specyfikacja stawiała ponadto oczywiste wymaganie, by wspomniane pole zawsze dekrementowane było co najmniej o 1, nawet jeśli czas przetworzenia pakietu plasował się poniżej sekundy. Ponieważ we współczesnych routerach mało prawdopodobne jest przetrzymywanie datagramu dłużej niż przez sekundę, pole, które pierwotnie pełnić miało rolę czasomierza dekrementowanego zgodnie z upływem czasu, stało się tak naprawdę licznikiem, dekrementowanym przy każdym przeskoku. Ta nowa sytuacja odzwierciedlona została w nazewnictwie protokołu IPv6, zgodnie z którym wspomniane pole otrzymało nową adekwatną nazwę *Limit przeskoków*, zachowując dotychczasową funkcję.

Pole *Protokół datagramu* IPv4 zawiera liczbę identyfikującą protokół, którego dane przenoszone są jako ładunek użyteczny datagramu; najczęściej spotykanymi w tym polu wartościami są 17 (identyfikująca UDP) i 6 (identyfikująca TCP). Pole to zapewnia datagramom IPv4 dużą uniwersalność w postaci zdolności do przenoszenia danych pochodzących z różnych protokołów. Mimo iż w oryginalnej koncepcji protokołu IP ładunek użyteczny zawierał dane pochodzące z protokołu warstwy transportowej, nie jest to warunek konieczny i nic nie stoi na przeszkodzie enkapsulowaniu w datagramach IPv4 jednostek innych protokołów, np. innych datagramów IPv4 (w polu *Protokół* znajduje się wówczas wartość 4) czy IPv6 (wartość 41); oficjalna lista zdefiniowanych identyfikatorów protokołów dostępna jest na stronie [AN].

W datagramie IPv6 pole *Następny nagłówek* jest analogią (i jednocześnie uogólnieniem) pola *Protokół* z IPv4. Zapewnia ono wiązanie w łańcuch ewentualnych nagłówków rozszerzeń oraz pola ładunku użytecznego — niebawem (w podrozdziale 5.3) opiszemy szczegółowo ten mechanizm.

Zadaniem pola *Suma kontrolna nagłówka* jest (zgodnie z nazwą) kontrola integralności samego nagłówka, bez jakiegokolwiek związku z ładunkiem użytecznym. Oznacza to, że protokół IPv4 nie zapewnia sam z siebie ochrony integralności przenoszonego w datagramach ładunku użytecznego, zatem protokoły warstw wyższych, zarządzające tym

ładunkiem, muszą zapewnić własny mechanizm ochrony i weryfikacji jego integralności — i rzeczywiście, większość z tych protokołów (ICMP, IGMP, UDP i TCP) czyni to za pomocą odpowiednich sum kontrolnych, obejmujących ich własne nagłówki i dane, a być może również — uwaga! — niektóre fragmenty zewnętrznego nagłówka IP, te które z perspektywy danego protokołu wydają się szczególnie istotne (co notabene stanowi wyraźne odstępstwo od ścisłej hierarchii warstw).

Być może zaskakujący jest fakt, że nagłówek datagramu IPv6 nie posiada żadnego pola sumy kontrolnej.

Rezygnacja z sumy kontrolnej weryfikującej integralność nagłówka IPv6 była decyzją cokolwiek kontrowersyjną. Podstawową przesłanką tej rezygnacji był fakt, że te protokoły warstwy wyższej, dla których istotna jest integralność nagłówka datagramu IP, same przeprowadzają weryfikację integralności tych obszarów, które wydają się dla nich istotne. Konsekwencją przekłamania w obszarze nagłówka IP może być błędne trasowanie pakietu, błędne wskazanie jego źródła pochodzenia itp., lecz, po pierwsze, przekłamania takie zdarzają się dziś raczej rzadko (za sprawą nowoczesnych łączy światłowodowych przenoszących ruch Internetu), po drugie, jeśli już przekłamanie się zdarzy, to do jego wykrycia i zneutralizowania wymagane są mechanizmy daleko bardziej zaawansowane niż prosta suma kontrolna. Ostatecznie więc, ograniczona potencjalnie przydatność sumy kontrolnej weryfikującej integralność nagłówka IPv6 jest powodem jej nieobecności w tym nagłówku.

Algorytm obliczania wspomnianej sumy kontrolnej wykorzystywany jest w większości innych protokołów związanych z Internetem, dlatego też suma ta nazywana jest popularnie **internetową sumą kontrolną** (*Internet checksum*). Zwróćmy uwagę, że skoro po przejściu datagramu IPv4 przez router zmienia się wartość w jego polu *Czas życia*, to zmieniać się musi również suma kontrolna nagłówka. Wspomniany algorytm omówimy dokładnie w punkcie 5.2.2.

Każdy datagram IP zawiera w nagłówku dwa adresy: źródłowy, odzwierciedlający pochodzenie datagramu, i docelowy, identyfikujący miejsce przeznaczenia tegoż datagramu. Adresy te są 32-bitowe w wersji IPv4 i 128-bitowe w wersji IPv6. Zwykle każdy z nich identyfikuje konkretny interfejs, lecz adresy multicast i broadcast (patrz rozdział 2.) stanowią odstępstwo od tej zasady. Mimo iż liczba możliwych adresów IPv4 — $2^{32} \approx 4{,}5$ miliarda — wydawała się swego czasu ogromna, to ich pula jest obecnie na wyczerpaniu (pisaliśmy o tym w rozdziale 2.) i to właśnie stanowi główną motywację „przesiadki" Internetu na wersję IPv6. Zgodnie z wyliczeniami prezentowanymi w [H05], gdyby równomiernie obdzielić dostępnymi adresami IPv6 cały obszar kuli ziemskiej (z oceanami włącznie), na każdy metr kwadratowy przypadałoby tych adresów 3 911 873 538 269 506 102, co raczej powinno wystarczyć na dość długo.

5.2.2. Internetowa suma kontrolna

Internetowa suma kontrolna jest 16-bitową liczba całkowitą stanowiącą funkcję obszaru danych i wykorzystywaną do zweryfikowania — z dużym prawdopodobieństwem — autentyczności tych danych. Spełnia więc ona rolę taką samą jak nadmiarowa kontrola cykliczna (CRC — patrz rozdział 3. i [PB61]), lecz algorytm jej obliczania jest zupełnie inny — znacznie prostszy, więc zapewniający mniejszy stopień ochrony.

Zakładamy, że chroniony obszar jest ciągiem parzystej liczby bajtów, ciąg o długości nieparzystej dopełniamy bajtem zerowym. Na ciągu tym wykonujemy następnie kolejno następujące operacje.

1. Pary sąsiadujących bajtów łączymy w 16-bitowe słowa według konwencji *big-endian* — bajt o niższym adresie staje się bardziej znaczącym bajtem słowa. W przykładzie przedstawionym na rysunku 5.3 ciąg bajtów `0xE3`, `0x4F`, `0x23`, `0x96`, `0x44`, `0x27`, `0x99`, `0xF3` przekłada się w opisanej konwencji na cztery słowa `0xE34F`, `0x2396`, `0x4427` i `0x99F3`, odpowiadające liczbom dziesiętnym `58191`, `9110`, `17447` i `39411`.
2. Kolejne słowa 16-bitowe dodajemy do siebie zgodnie z regułami arytmetyki uzupełnienia do jedności (*one-complement*)[2], w skrócie U1. W przełożeniu na arytmetykę procesorów operujących w arytmetyce „uzupełnień do dwóch" U2 (czyli wszystkich używanych obecnie procesorów) polega to na zwykłym binarnym dodaniu do siebie wspomnianych słów, potraktowanie wyniku jako 32-bitowej liczby binarnej, „złamanie" jej na pół, dodanie do siebie obu połówek i zachowanie 16 najmniej znaczących bitów tak otrzymanego wyniku (to ostatnie dodawanie jest więc dodawaniem modulo `0x1000`). Zsumowanie słów, o których mowa w punkcie 1., daje w wyniku `124159` (dziesiętnie), czyli `0x0001E4FF` (w przełożeniu na liczbę 32-bitową). Dodając do siebie `0x0001` i `0xE4FF`, dostajemy `0xE500`.
3. Tworzymy negację bitową wyniku otrzymanego w punkcie 2., dostając w ten sposób wartość żądanej sumy kontrolnej. Dla przykładu z rysunku 5.3:

```
~(0xE500) = ~(1110 0101 0000 0000) = 0001 1010 1111 1111 = 0x1AFF
```

Przedstawiony algorytm wykorzystywany jest do weryfikowania nagłówka w następujący sposób: zerujemy pole *Suma kontrolna nagłówka* i obliczamy dla tego nagłówka sumę kontrolną w sposób powyżej opisany. Otrzymany wynik wpisujemy w pole *Suma kontrolna nagłówka*; gdybyśmy teraz ponownie obliczyli sumę kontrolną nagłówka, okazałoby się, że ma ona wartość `0`. Gdy wysłany datagram zostanie odebrany przez router (lub host docelowy), oblicza się sumę kontrolną jego nagłówka i sprawdza, czy faktycznie jest ona zerowa. Otrzymanie innego wyniku świadczy o zniekształceniu nagłówka w czasie transmisji datagramu — datagram jest wówczas odrzucany bez żadnego ostrzeżenia, wykrycie tego faktu i zainicjowanie retransmisji datagramu jest zadaniem protokołów warstw wyższych.

[2] W reprezentacji „uzupełnienia do jedności" liczba ujemna zapisywana jest jako bitowa negacja jej dodatniego odpowiednika. Dodawanie liczb binarnych w tej reprezentacji wykonuje się jak dodawanie zwykłych liczb binarnych (czyli traktując bit znaku na równi z pozostałymi bitami), lecz ewentualne przeniesienie z bitu znaku należy dodać na najmniej znaczącej pozycji. Przykładowo „zwykłe" dodawanie słów `0xE34F` i `0x2396` daje w wyniku `0x106E5`, czyli w arytmetyce 16-bitowej otrzymamy wynik `0x06E5` oraz przeniesienie 1, które dodane na najmłodszej pozycji daje ostatecznie `0x06E6`. Ponieważ arytmetyka uzupełnienia do jedności jest już w świecie komputerów historią — wykorzystywały ją m.in. jednostki centralne komputerów CDC i Univac — rodzi się pytanie o sens jej używania na gruncie nowoczesnych technologii. Otóż podstawową zaletą przedstawionego algorytmu jest jego *symetria względem uporządkowania bajtów*: gdybyśmy mianowicie wykonali grupowanie opisane w punkcie 1. według konwencji *little-endian*, dostalibyśmy identyczną wartość sumy kontrolnej, z zamienioną kolejnością bajtów. Oznacza to, że w implementacjach wykonujących konwersję między uporządkowaniami *big-endian* i *little-endian* obliczanie sumy kontrolnej może następować przed konwersją wysyłanego datagramu i po konwersji odebranego, albo odwrotnie, byle symetrycznie po obu stronach — *przyp. tłum.*

Przed wysłaniem

Komunikat:	E3 4F 23 96 44 27 99 F3 [00 00] ← *Pole sumy kontrolnej = 0000*
Suma w uzupełnieniu do dwóch:	1E4FF
Suma w uzupełnieniu do jedności:	1E4FF -> E34F + 2396 + 4427 + 99F3 = 1E4FF
Negacja bitowa :	~(E500) = ~(1110 0101 0000 0000) = 0001 1010 1111 1111 = 1AFF (suma kontrolna)

Po odebraniu

Komunikat + suma kontrolna:	E34F + 2396 + 4427 + 99F3 + 1AFF = E500 + 1AFF = FFFF
	~(Komunikat + suma kontrolna) = 0000

Rysunek 5.3. *Internetowa suma kontrolna jako negacja bitowa wyniku sumowania 16-bitowych słów kontrolowanego obszaru w arytmetyce U1 (w przypadku obszaru o nieparzystej liczbie bitów mniej znaczący bajt ostatniego słowa ma wartość 0). Jeżeli pole sumy kontrolnej jest częścią kontrolowanego obszaru (jak w nagłówku IPv4), do pola tego wpisuje się wartość 0, oblicza sumę kontrolną i zapisuje we wspomnianym polu. Ponowne obliczenie sumy kontrolnej dla tak zmodyfikowanego obszaru daje wartość 0, ponieważ wartość znajdująca się w polu sumy kontrolnej jest negacją sumy kontrolnej obliczanej dla pozostałej części obszaru*

5.2.2.1. Własności matematyczne internetowej sumy kontrolnej

Niech **V** będzie zbiorem nieujemnych liczb 16-bitowych, **V** = {0x0001, 0x0002, …, 0xFFFF}, niech ⊗ symbolizuje operację dodawania dwóch liczb w uzupełnieniu do jedności, opisaną w punkcie 2. scenariusza z punktu 5.2.2. Zbiór **V** w połączeniu z operacją ⊗ tworzy grupę abelową (przemienną), ponieważ:

1. Operacja ⊗ jest *wewnętrzna* w zbiorze **V**: dla dowolnych $x, y \in \mathbf{V}$, $x \otimes y \in \mathbf{V}$. Jest to oczywiste w przypadku, gdy $0x0001 \le x + y \le 0xFFFF$; w pozostałych przypadkach $0x10000 \le x + y \le 0x1FFFE$, a więc $0x0001 \le x \otimes y \le 0xFFFF$.
2. Operacja ⊗ jest *wewnętrzna* w zbiorze **V**: dla dowolnych $x, y, z \in \mathbf{V}$, $(x \otimes y) \otimes z = x \otimes (y \otimes z)$. Wynika to wprost z łączności dodawania modulo 0x10000.
3. Element $e = 0xFFFF$ jest w zbiorze **V** elementem jednostkowym: dla każdego $x \in \mathbf{V}$, $x \otimes e = e \otimes x = x$. Istotnie: dodawanie $x \otimes 0xFFFF$ zawsze powoduje powstanie przeniesienia równego 1, a więc $x \otimes 0xFFFF = x + 0xFFFF + 1 = (x + 0xFFFF) \bmod 0x10000 + 1$. Ponieważ $(x + 0xFFFF) \bmod 0x10000 < 0xFFFF$, więc $(x + 0xFFFF) \bmod 0x10000 + 1 = (x + 0xFFFF + 1) \bmod 0x10000 = x \bmod 0x10000 + (0xFFFF + 1) \bmod 0x10000 = x + (0x10000 \bmod 0x10000) = x + 0 = x$.
4. Dla każdego $x \in \mathbf{V}$ istnieje *dokładnie jeden* element odwrotny $x^{-1} \in \mathbf{V}$ taki, że $x + x^{-1} = e = 0xFFFF$. Po chwili zastanowienia staje się oczywiste, że dla $x < 0xFFFF$ element x^{-1} jest bitową negacją x. Element $x = 0xFFFF = e$ jest swoją własną odwrotnością (musi nią być jako element jednostkowy).
5. Działanie ⊗ jest przemienne: dla dowolnych $x, y \in \mathbf{V}$, $x \otimes y = y \otimes x$ — co wynika z przemienności zwykłego dodawania.

Zwróćmy uwagę, że zbiór **V** przestałby być grupą ze względu na działanie ⊗, gdybyśmy włączyli do niego element 0x0000. Dla elementu $x = 0xFFFF$ istniałyby wówczas *dwa* elementy y spełniające równanie $x \otimes y = e = 0xFFFF$: $y = 0x0000$ oraz $y = 0xFFFF$. Nie byłby więc spełniony warunek jednoznaczności elementu odwrotnego, wyartykułowany w punkcie 4.

Czytelnikom pragnącym zapoznać się z podstawami algebry abstrakcyjnej polecamy książkę [P90].

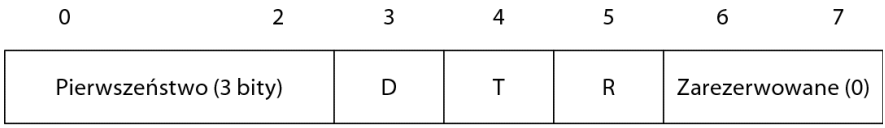
5.2.3. Pola DS i ECN (dawniej ToS i Klasa ruchu)

Pola trzecie i czwarte nagłówka IPv4 (drugie i trzecie w nagłówku IPv6), czyli *Usługi zróżnicowane* i *ECN*, związane są (ogólnie rzecz biorąc) z mechanizmami modyfikującymi standardowe forwardowanie datagramów. Pod pojęciem różnicowania usług rozumiemy świadczenie ich w standardach odbiegających poza rutynowy ruch niegwarantowany (*best-effort delivery*), głównie w oparciu o priorytety transmisji (patrz dokumenty [RFC2474], [RFC2475] i [RFC3260]). Wyższy priorytet oznacza większe uprzywilejowanie datagramu i krótsze (w stosunku do datagramów o niższym priorytecie) oczekiwanie w kolejce do przetworzenia przez router. Liczba znajdująca się w polu *Usługi zróżnicowane* nazywana bywa **punktem kodowym** (*Differentiated Services Code Point*, w skrócie DSCP). Każdy z „punktów kodowych" jest kombinacją bitów o uzgodnionym a priori znaczeniu. Zazwyczaj datagram opatrywany jest odpowiednią wartością DSCP przy wejściu do infrastruktury sieciowej, która to wartość pozostaje niezmieniona przez cały czas wędrówki datagramu przez sieć; często jednak polityka administracyjna (np. w kwestii limitowania liczby uprzywilejowanych pakietów transmitowanych w jednostce czasu) prowadzić może do modyfikowania DSCP pakietu (w kierunku zmniejszenia jego uprzywilejowania).

Para bitów tworzących pole ECN wykorzystywana jest do opatrywania datagramu tzw. **indykatorem przeciążenia** w sytuacji, gdy datagram ten przechodzi przez router, w którego kolejce czeka na przetworzenie duża liczba pakietów; routery implementujące funkcję powiadamiania o przeciążeniu ustawiają wówczas na 1 oba bity. W zamierzeniu projektantów tego mechanizmu, gdy oznakowany w ten sposób pakiet dotrze do miejsca przeznaczenia, odpowiednio inteligentny protokół (np. TCP) zostanie w ten sposób powiadomiony o trudnej sytuacji w warstwie sieciowej i zasygnalizuje nadawcy konieczność spowolnienia tempa wysyłania nowych pakietów. Jest to jeden z mechanizmów przeznaczonych do unikania przeciążeń i radzenia sobie z nimi; dokładniejszemu omówieniu tych mechanizmów poświęcamy rozdział 16.

Chociaż pola *Usługi zróżnicowane* i *ECN* nie wydają się mieć ze sobą ścisłego związku, to jednak mają wspólną genealogię, jak wcześniej wspomnieliśmy, są wynikiem podziału pola *Typ usługi* (w IPv6 pola *Klasa ruchu*). Często więc opisywane są łącznie, a wymienione oryginalne nazwy w dalszym ciągu są sankcjonowane w literaturze, także w różnych odmianach („bajt ToS", „bajt klasy ruchu" itp.). I mimo że oryginalna koncepcja pojedynczego pola *Typ usługi/Klasa ruchu* nigdy nie doczekała się pełnego wsparcia, to jednak we współczesnych implementacjach IP obsługa pola *Usługi zróżnicowane* zrealizowana została z myślą o kompatybilności wstecz (w pewnym stopniu) z pierwowzorem. Aby wyjaśnić szczegóły tej koncepcji, przyjrzyjmy się wpierw oryginalnej strukturze pola *Typ usługi*, przedstawionej na rysunku 5.4 (zaczerpniętym ze specyfikacji [RFC0791]).

Bity *D*, *T* i *R* wyrażają potrzebę szczególnego traktowania pakietu, pod względem małego opóźnienia, dużej przepustowości i wysokiej niezawodności (zgodnie z podpisem pod rysunkiem 5.4). Wartości priorytetu (pierwszeństwa) zmieniają się od 0 (rutynowy) do 7 (krytyczny z punktu widzenia zarządzania siecią). Szczegółowe znaczenie każdej



Rysunek 5.4. *Oryginalna struktura pola Typ usługi (w IPv6 Klasa ruchu). Wartość w polu Pierwszeństwo określa stopień uprzywilejowania (priorytetu) pakietu — większa wartość oznacza wyższy priorytet. Ustawienie bitów D, T lub R jest sygnałem, że dla danego pakietu szczególnie pożądane jest (odpowiednio) małe opóźnienie (delay), duża przepustowość (throughput) i niezawodność dostarczenia (reliability)*

wartości priorytetu przedstawiono w tabeli 5.1; bazuje ono na schemacie wywłaszczania zwanym *MultiLevel Precedence and Preemption* (MLPP), datującym się jeszcze z czasów systemu telefonicznego AUTOVON departamentu obrony USA (patrz [AUTOVON]), w którym rozmowa o niższym priorytecie mogła zostać przerywana („wywłaszczona" z przydziału linii telefonicznej) na rzecz połączenia o wyższym priorytecie. Widoczne w tabeli nazwy są w dalszym ciągu obowiązujące i zostały zaadaptowane na gruncie VoIP.

Tabela 5.1. *Oryginalne znaczenie podpola Pierwszeństwo w polu Typ usługi/Klasa ruchu*

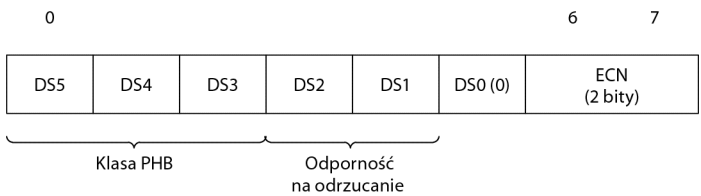
Wartość	Nazwa priorytetu
000	Zwyczajny (*routine*)
001	Nadrzędny (*priority*)
010	Natychmiastowy (*immediate*)
011	Błyskawiczny (*flash*)
100	Ponadbłyskawiczny (*flash override*)
101	Krytyczny (*critical*)
110	Sterowanie międzysiecią (*internetwork control*)
111	Sterowanie siecią (*network control*)

Z każdą wartością DSCP związana jest określona strategia routera w zakresie forwardowania pakietów, zwana *zachowaniem na przeskoku* (*Per-Hop Behavior*, w skrócie PHB). Definiując PHB dla poszczególnych DSCP, projektanci starali się zachować kompatybilność z pierwowzorem, czyli zachowaniem, jakie wynikałoby z mapowania bitowego wzorca DSCP na dawną strukturę podpól *Pierwszeństwo*, *D*, *T* i *R*. Przykładowo DSCP równe 22 — binarnie 010110 — odwzorowuje się na *Pierwszeństwo* równe 010 i ustawione bity *D* i *T*, tak więc PHB wynikające z owego DSCP powinno charakteryzować się priorytetem na poziomie „Natychmiastowy" i zapewnieniem pakietowi jak najmniejszego opóźnienia i jak największej przepustowości. Zamiar ten został jednak zrealizowany w ograniczonym zakresie, bowiem zbiór 64 możliwych wartości DSCP podzielony został na trzy grupy, identyfikowane najmniej znaczącymi bitami: numery DSCP kończące się bitem 0 przeznaczone są do standardowego użytku, te kończące się kombinacją 11 mają przeznaczenie wyłącznie eksperymentalne, pozostałe — czyli te z końcówką 01 — mają obecnie charakter eksperymentalny, lecz w przyszłości mogą stać się elementami standardu. Klasyfikację tę ilustruje tabela 5.2, zaczerpnięta z [DSCPREG].

Tabela 5.2. *Podział możliwych wartości na trzy grupy: standardowe, wyłącznie eksperymentalne i eksperymentalne z możliwością standaryzacji*

Grupa	Szablon DSCP	Przeznaczenie
1	*xxxxx*0	Użytek standardowy.
2	*xxxx*11	Wykorzystywanie eksperymentalne i w zastosowaniach lokalnych.
3	*xxxx*01	Wykorzystywanie eksperymentalne i w zastosowaniach lokalnych, z możliwością ewentualnej standaryzacji w przyszłości.

Wspomniane odwzorowanie DSCP na strukturę *Pierwszeństwo-D-T-R* zostało więc zrealizowane połowicznie, bowiem (zgodnie z tabelą 5.2) bit *R* ma w tym odwzorowaniu zawsze wartość 0. Ostatecznie trzy najbardziej znaczące bity DSCP identyfikują tzw. *klasę* PHB, pozostałe dwa natomiast odzwierciedlają *odporność pakietu na odrzucanie*: im większa wartość tej odporności, tym mniejsze prawdopodobieństwo, że w sytuacji przeciążenia router odrzuci ten właśnie pakiet[3]. Mapowanie to przedstawiono schematycznie na rysunku 5.5.



Rysunek 5.5. *Znaczenie poszczególnych grup bitów DSCP, wynikające z (częściowej) kompatybilności ze znaczeniem pola Typ usługi/Klasa ruchu. W grupie standardowych DSCP bit DS0 ma zawsze wartość 0. 2-bitowe pole ECN zawiera powiadomienie dla odbiorcy, że przetworzenie datagramu odbyło się w warunkach permanentnego przeciążenia routera, co dla tegoż odbiorcy stanowić może sygnał dla podjęcia stosownych działań, np. zażądania od nadawcy spowolnienia tempa transmisji*

Zgodnie z rysunkiem 5.5, podpole *Klasa PHB* jest odpowiednikiem podpola *Pierwszeństwo* w dawnym polu *Typ usługi*/*Klasa ruchu*. „Klasa PHB" oznacza tu odpowiedni sposób traktowania datagramów — datagramy zawierające tę samą wartość na bitach *DS5*, *DS4* i *DS3* (wartość ta nazywana jest *selektorem klasy*) traktowane są przez router według tej samej strategii. W ramach danej klasy PHB pakiety mogą być zróżnicowane pod względem preferowania przez router w sytuacji wymagającej odrzucania pakietów; jedna z czterech możliwych wartości kombinacji bitów *DS2* i *DS1* jest wskaźnikiem odporności pakietu na odrzucenie: 01 oznacza odrzucanie w pierwszej kolejności (w ramach danej klasy PHB), wartość 11 — odrzucanie tylko w ostateczności (ponownie — w ramach tej samej klasy PHB). Zerowa wartość obu bitów *DS2* i *DS1* oznacza brak jawnej preferencji dla pakietu pod względem odporności na odrzucanie — taka (domyślna) strategia nazywana jest często *zgodnym z klasami selektorów zachowaniem na przeskoku* (*class*

[3] W literaturze polskiej nazwa *Drop probability* tłumaczona jest powszechnie jako „Prawdopodobieństwo odrzucenia pakietu" i choć lingwistycznie wszystko jest w porządku, należy zdawać sobie sprawę z faktu, że związek wartości pola *Drop probability* ze wspomnianym prawdopodobieństwem jest dokładnie odwrotny — *największa* możliwa wartość tego pola (11) oznacza *najmniejsze* prawdopodobieństwo odrzucenia pakietu. Wartość ta jest zatem nie tyle *prawdopodobieństwem* odrzucenia, co raczej umownym *znacznikiem* tego prawdopodobieństwa — *przyp. tłum.*

selektor compliant PHBs); w zamierzeniu projektantów ma ona realizować (częściową) kompatybilność wstecz z oryginalną funkcją pola *Pierwszeństwo* sformułowaną w [RFC0791].

Spośród możliwych 32 wartości DSCP odpowiadających szablonowi *xxxxx*0 do standardowego użytku wybrano tylko niektóre — ich zestawienie znajduje się w tabeli 5.3. Dla każdej kombinacji podano jej oficjalne oznaczenie, definiujący ją dokument oraz wyjaśnienie znaczenia, także z nawiązaniem do dawnej struktury podpola *Pierwszeństwo*. Na początku widzimy więc selektory klas i ich odpowiedniki z tabeli 5.1, kolejne pozycje reprezentują grupy tzw. **gwarantowanego forwardowania** (*assured forwarding*): 12 grup oznaczonych AF<*x*><*y*> to rezultat możliwych kombinacji wartości selektorów klas <*x*> od 1 do 4 i wskaźników podwyższonej odporności na odrzucanie <*y*> od 1 do 3. W ramach selektora klasy 5 wyróżniono dwa poziomy preferencyjnego ruchu: **przyspieszone forwardowanie** (*expedited forwarding*) oraz forwardowanie ograniczone jedynie dostępnym pasmem (*capacity-admitted*), mające absolutne pierwszeństwo przed innymi DSCP.

Tabela 5.3. *Wykorzystywane standardowo wartości DSCP, częściowo kompatybilne z pierwowzorem w postaci pola Typ usługi/Klasa ruchu. Grupy AF i EF oznaczają strategię preferencyjną w stosunku do ruchu niegwarantowanego (best-effort delivery)*

Oznaczenie	Wartość (binarnie)	Dokument	Znaczenie
CS0	000000	[RFC2474]	Selektor klasy (ruch niegwarantowany)
CS1	001000	[RFC2474]	Selektor klasy (ruch nadrzędny)
CS2	010000	[RFC2474]	Selektor klasy (ruch natychmiastowy)
CS3	011000	[RFC2474]	Selektor klasy (ruch błyskawiczny)
CS4	100000	[RFC2474]	Selektor klasy (ruch ponadbłyskawiczny)
CS5	101000	[RFC2474]	Selektor klasy (ruch o znaczeniu krytycznym)
CS6	110000	[RFC2474]	Selektor klasy (sterowanie międzysiecią)
CS7	111000	[RFC2474]	Selektor klasy (sterowanie siecią)
AF11	001010	[RFC2597]	Gwarantowane forwardowanie (assured forwarding) (klasa 1, odporność 1)
AF12	001100	[RFC2597]	Gwarantowane forwardowanie (1,2)
AF13	001110	[RFC2597]	Gwarantowane forwardowanie (1,3)
AF21	010010	[RFC2597]	Gwarantowane forwardowanie (2,1)
AF22	010100	[RFC2597]	Gwarantowane forwardowanie (2,2)
AF23	010110	[RFC2597]	Gwarantowane forwardowanie (2,3)
AF31	011010	[RFC2597]	Gwarantowane forwardowanie (3,1)
AF32	011100	[RFC2597]	Gwarantowane forwardowanie (3,2)
AF33	011110	[RFC2597]	Gwarantowane forwardowanie (3,3)
AF41	100010	[RFC2597]	Gwarantowane forwardowanie (4,1)
AF42	100100	[RFC2597]	Gwarantowane forwardowanie (4,2)
AF43	100110	[RFC2597]	Gwarantowane forwardowanie (4,3)
EFPHB	101110	[RFC3246]	Forwardowanie przyspieszone (*expedited forwarding*)
VOICE-ADMIT	101100	[RFC5865]	Ruch uwarunkowany pasmem (*capacity-admitted traffic*)

Usługa przyspieszonego forwardowania (*expedited forwarding*) jest de facto uwolnieniem forwardowanego pakietu od kontekstu ewentualnego przeciążenia routera: jeśli pakiet taki w ogóle musi oczekiwać w kolejce, to tylko z powodu przetwarzania innego pakietu z DSCP równym EFPHB. W efekcie pakiety tej grupy rzadko są gubione, a dane przekazywane za ich pośrednictwem doświadczają małych opóźnień (i w konsekwencji małych błędów *jitter*).

Zagadnienie świadczenia usług zróżnicowanych było przedmiotem znaczącego wysiłku projektantów i inżynierów u schyłku XX wieku. Mimo iż zabiegi standaryzacyjne uwieńczone zostały powodzeniem jeszcze w latach 90., to dopiero po roku 2000 niektóre z mechanizmów tej kategorii doczekały się praktycznych zastosowań. Stało się to z przyczyn natury nie tyle technicznej, co raczej ekonomicznej: usługi zróżnicowane dają klientowi różnego rodzaju dodatkowe korzyści i przywileje, za które ten powinien odpowiednio płacić i właśnie kwestia sprawiedliwości systemu opłat jest tu problemem skomplikowanym, wymykającym się możliwościom dyskusji w ograniczonych ramach książki. Zainteresowani czytelnicy znajdą wiele szczegółów tego tematu w książce [MB97] i publikacji [W03].

5.2.4. Opcje IP

Proces przetwarzania datagramu IP może być modyfikowany za pomocą rozmaitych opcji o charakterze lokalnym, czyli dotyczących wyłącznie tego datagramu. Wiele tych opcji zostało zdefiniowanych wraz z oryginalną definicją protokołu IPv4 w dokumencie [RFC0791]; Internet miał wówczas rozmiary znacznie mniejsze od obecnych, mniejsze też było zagrożenie czyhające ze strony niesubordynowanych internautów. Gdy poczęły się kształtować zręby protokołu IPv6, rzeczywistość była już wyraźnie inna, wskutek czego niektóre z opcji straciły rację bytu, zaś użyteczność innych okazała się ograniczona z powodu limitowanych rozmiarów samego nagłówka IPv4, jeszcze inne okazały się wątpliwe z perspektywy bezpieczeństwa sieci. Projektanci IPv6 postanowili więc o generalnym usunięciu opcji IP z nagłówka datagramu i umieszczeniu ich w nowym elemencie — **nagłówku rozszerzeń**, umiejscowionym w pakiecie między nagłówkiem podstawowym a obszarem ładunku użytecznego. Nagłówki rozszerzeń przetwarzane są generalnie dopiero przez host końcowy, wyjątkowo jednak niektóre z nich przetwarzane są także przez routery pośredniczące. W niektórych routerach obecność nagłówków rozszerzeń powoduje spowolnienie przetwarzania datagramu, nawet jeśli nagłówki te nie są przez router przetwarzane.

Rozpoczniemy od omówienia opcji IPv4, po czym zajmiemy się opcjami i nagłówkami rozszerzeń IPv6. W tabeli 5.4 widoczny jest podsumowujący wykaz opcji IPv4, jakie z biegiem lat zyskały sobie rangę standardów. Lista opcji IP nie jest kompletna — jest okresowo uaktualniana i dostępna pod adresem [IPPARAM].

Tabela 5.4. *Opcje datagramu IPv4, jeśli występują, umiejscowione są bezpośrednio po obowiązkowych, podstawowych polach nagłówka. Każda opcja identyfikowana jest na podstawie 8-bitowego pola Typ opcji, podzielonego na trzy podpola, reprezentujące kopiowanie (1 bit), klasę (2 bity) i numer (5 bitów). Dla niektórych opcji pole to jest jedynym polem, większość opcji ma jednak strukturę bardziej rozbudowaną: po 8-bitowym polu Typ opcji występuje 8-bitowe pole wskazujące Rozmiar opcji, potem następują Dane opcji (w polu Rozmiar opcji uwzględnione są wszystkie trzy pola)*

Nazwa	Numer opcji	Typ opcji (bajt identyfikacyjny)	Rozmiar	Opis	Dokument	Komentarz
Koniec listy	0	0	1	Wskazuje koniec obszaru opcji.	[RFC0791]	Nie występuje w przypadku braku opcji.
Opcja pusta	1	1	1	Opcja niereprezentująca żadnego znaczenia.	[RFC0791]	Występuje między opcjami, np. w celu wyrównania początku następnej opcji do granicy 32-bitowego słowa.
Trasowanie źródłowe (*Source Routing*)	3 9	131 137	Zmienny	Zawiera (stworzoną przez nadawcę) listę routerów pośredniczących w forwardowaniu pakietu. Występuje w dwóch odmianach: typ 3 oznacza zezwolenie na uzupełnienie wspomnianej listy o dodatkowe pozycje, typ 9 oznacza listę ściśle ustaloną.	[RFC0791]	Rzadko używana, zwykle filtrowana.
Etykiety bezpieczeństwa i zarządzania (*Security and Handling Labels*)	2 5	130 133	11	Określa sposób dołączania etykiet bezpieczeństwa i ograniczeń w przetwarzaniu datagramów IP w zastosowaniach militarnych USA.	[RFC1108]	Obecnie niewykorzystywana.
Rejestracja trasy (*Record Route*)	7	7	Zmienny	Zawiera listę routerów, jakie dotychczas przebył pakiet na swej trasie.	[RFC0791]	Rzadko używana.
Znakowanie czasowe (*Timestamp*)	4	68	Zmienny	Zawiera listę znaczników czasowych, odzwierciedlających momenty przetwarzania przez poszczególne węzły.	[RFC0791]	Rzadko używana.

Tabela 5.4. — ciąg dalszy

Nazwa	Numer opcji	Typ opcji (bajt identyfikacyjny)	Rozmiar	Opis	Dokument	Komentarz
Identyfikator strumienia (*Stream ID*)	8	136	4	Zawiera 16-bitowy identyfikator strumienia SATNET.	[RFC0791]	Obecnie niewykorzystywana.
EIP	17	145	Zmienny	Opcja *Extended Internet Protocol* — eksperymentalnego protokołu z lat 90. ubiegłego wieku.	[RFC1385]	Obecnie niewykorzystywana.
Śledzenie trasy (*Traceroute*)	18	82	Zmienny	Powoduje śledzenie trasy i wysyłanie komunikatów ICMP — w zastosowaniach eksperymentalnych z lat 90. ubiegłego wieku.	[RFC1393]	Obecnie niewykorzystywana.
Alarm dla routera (*Router Alert*)	20	148	4	Wskazuje konieczność interpretowania przez router treści datagramu.	[RFC2113] [RFC5350]	Używana okazjonalnie.
Szybki start (*Quick-Start*)	25	25	8	Wskazuje szybki start protokołu transportowego.	[RFC4782]	Opcja eksperymentalna, rzadko używana.

Obszar opcji w nagłówku IPv4 kończy się na granicy słowa 32-bitowego (ze względu na charakter pola *IHL* długość nagłówka musi być wielokrotnością 4 bajtów). Pole *Typ opcji* jest zawsze pierwszym bajtem opcji, identyfikującym ją. Najbardziej znaczący bit tego bajta określa, czy w razie fragmentacji datagramu opcja ma być kopiowana do nagłówka każdego fragmentu (1), czy też ma pojawić się wyłącznie w nagłówku pierwszego fragmentu (0). Dwa kolejne bity określają klasę opcji; spośród opcji wymienionych w tabeli 5.4 opcje *Znakowanie czasowe* i *Śledzenie trasy* są opcjami klasy 2 (*debugging and measurement* — debugowanie i pomiary), pozostałe opcje są opcjami klasy 0. Klasy 1 i 3 są zarezerwowane do przyszłego użytku. Pięć najmniej znaczących bitów bajta identyfikacyjnego składa się na *Numer opcji* (wyjaśnia to różnicę między drugą i trzecią kolumną tabeli). Niektóre opcje reprezentowane są tylko przez swój bajt identyfikacyjny, niektóre natomiast posiadają bardziej rozbudowaną strukturę; w tym drugim przypadku bezpośrednio po bajcie identyfikacyjnym następuje bajt jawnie wskazujący *Rozmiar*, czyli liczbę bajtów owej struktury (począwszy od bajta identyfikacyjnego do ostatniego bajta danych). Przynależność opcji do jednej z tych dwóch kategorii wynika z jej numeru — spośród opcji wymienionych w tabeli 5.4 tylko opcje *Koniec listy* i *Opcja pusta* należą do kategorii pierwszej.

Większość z wymienionych opcji standardowych używana jest dziś rzadko lub wcale, także z powodu przyrodzonych ograniczeń protokołu IPv4, a konkretnie — ograniczenia rozmiaru nagłówka do 60 bajtów. Przykładowo do dyspozycji opcji *Trasowanie źródłowe* lub *Rejestracja trasy* stoi w najlepszym przypadku 40 bajtów (20 ze wspomnianego limitu zajęte jest przez obowiązkowe pola nagłówka), co jest niewystarczające w sytuacji, gdy

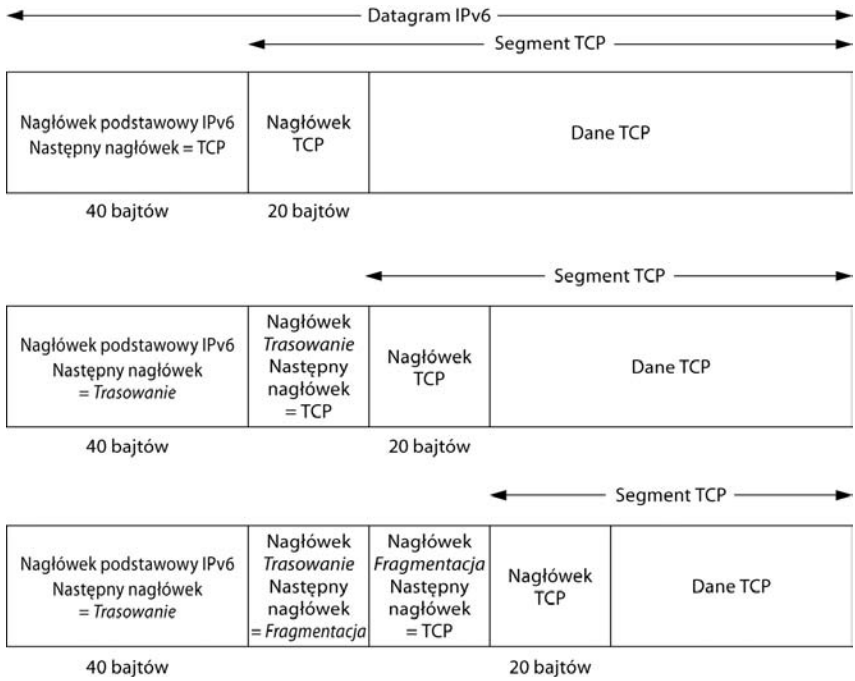
(jeśli wierzyć artykułowi [LFS07]) typowy datagram IP przebywa na swej trasie średnio 15 przeskoków. Ponadto większość z opcji ma charakter diagnostyczny i jedną z konsekwencji ich istnienia jest większa komplikacja konstrukcji firewalli i większe ryzyko związane z definiowaniem wyjątków w ich konfiguracji. Niejako wyjątkiem jest w tym względzie opcja *Alarm dla routera*, wskazująca na konieczność przetworzenia pakietu przez router w sposób wykraczający poza konwencjonalne forwardowanie. Eksperymentalną opcję *Szybki start*, wykorzystywaną zarówno w IPv4, jak i IPv6, omówimy w następnym podrozdziale, w związku z nagłówkami rozszerzeń IPv6.

5.3. Nagłówki rozszerzeń IPv6

W protokole IPv6 opcje i inne funkcje dodatkowe potraktowane zostały w sposób całkowicie odmienny w stosunku do IPv4. Ponieważ z założenia używane są opcjonalnie i wybiórczo, uznano za niecelowe lokowanie ich w obrębie nagłówka podstawowego, obarczonego dotkliwym ograniczeniem rozmiaru; znalazły swe miejsce w obszarze nagłówków rozszerzeń (*extension headers*) ułożonych w łańcuch, którego ostatnim ogniwem jest obszar danych protokołu warstwy wyższej, co (w kilku wariantach) przedstawiono na rysunku 5.6. Nagłówek podstawowy ma dzięki temu ustaloną strukturę i ustalony rozmiar 40 bajtów (patrz rysunek 5.2). Ponieważ nagłówek podstawowy jest jedynym elementem datagramu przetwarzanym przez routery pośredniczące (istnieje jeden ważny wyjątek od tej reguły), upraszcza się algorytm obsługi datagramu przez router, co w zamierzeniu powinno prowadzić do skrócenia czasu tej obsługi (choć czas ten zależny jest także od wielu innych czynników, takich jak złożoność protokołu, możliwości sprzętu, wydajność oprogramowania czy bieżące obciążenie routera).

Jak pokazano na rysunku 5.6, ciąg nagłówków rozszerzeń (o ile występują) umiejscowiony jest między podstawowym nagłówkiem IPv6 a obszarem protokołu warstwy wyższej (na rysunku jest nim protokół TCP). Elementem wiążącym nagłówki rozszerzeń w łańcuch jest pole *Następny nagłówek*, wskazujące typ następnego nagłówka, zgodnie z tabelą 5.5. Z definicji ostatnim elementem wspomnianego łańcucha jest obszar protokołu warstwy wyższej; specyfikacja IPv6 dopuszcza jednak brak tego obszaru, wówczas w polu *Następny nagłówek* ostatniego ogniwa łańcucha znajduje się wartość 59, jawnie wskazująca koniec łańcucha (patrz sekcja 4.7 dokumentu [RFC2460]). Kompletna lista wartości dopuszczalnych w polu *Następny nagłówek* dostępna jest na stronie [IP6PARAM], w tabeli 5.5 przedstawiono tylko wybrane.

Wskazywana w drugiej kolumnie kolejność występowania nagłówków rozszerzeń jest jedynie zaleceniem, a nie bezwzględnym wymogiem; wyjątkiem są opcje „Skok po skoku", które — o ile występują — muszą znaleźć się bezpośrednio za nagłówkiem podstawowym. Z perspektywy implementatora oznacza to, z jednej strony, dość dużą dowolność w zakresie wspomnianej kolejności, z drugiej natomiast, wymaga od implementacji przetworzenia dowolnego nagłówka niezależnie od pozycji, na której występuje. Każdy z typów nagłówka rozszerzenia może wystąpić co najwyżej jeden raz, wyjątkiem jest nagłówek *Opcje docelowe*, który może wystąpić dwukrotnie: pierwszy raz w związku z docelowym adresem IP zawartym w nagłówku podstawowym IPv6, drugi raz w związku z adresem IP określającym ostateczne przeznaczenie datagramu — w niektórych sytuacjach adresy te mogą się różnić, np. w przypadku użycia nagłówka *Trasowanie* docelowy adres IP w nagłówku podstawowym może zmieniać się przy przechodzeniu przez kolejne routery.



Rysunek 5.6. *Trzy przykłady umiejscowienia łańcucha nagłówków rozszerzeń w datagramie IPv6. Elementem wiążącym nagłówki w łańcuch jest pole Następny nagłówek, ostatnim elementem tego łańcucha jest obszar protokołu warstwy wyższej (tu TCP) albo nagłówek jawnie oznaczony jako ostatni, czyli zawierający wartość 59 we wspomnianym polu*

Tabela 5.5. *Wartość w polu Następny nagłówek może wskazywać na nagłówek rozszerzenia bądź obszar protokołu; większość z tych wartości rozpoznawalna jest również w polu Protokół nagłówka IPv4*

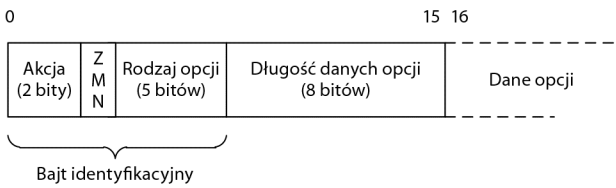
Typ nagłówka	Pozycja w łańcuchu	Wartość	Uwagi
Nagłówek podstawowy IPv6	1	41	[RFC2460] [RFC2473]
Opcje „skok po skoku” (HOPOPT)	2	0	[RFC2460] Muszą wystąpić bezpośrednio po nagłówku podstawowym.
Opcje docelowe	3,8	60	[RFC2460]
Trasowanie	4	43	[RFC2460] [RFC5095]
Fragmentacja	5	44	[RFC2460]
Nagłówek ESP (*Encapsulating Security Payload*)	7	50	Patrz rozdział 18.
Nagłówek uwierzytelniający (AH —*Authentication Header*)	6	51	Patrz rozdział 18.
Mobilne IP (MIPv6)	9	135	[RFC6275]

Tabela 5.5. *Wartość w polu Następny nagłówek może wskazywać na nagłówek rozszerzenia bądź obszar protokołu; większość z tych wartości rozpoznawalna jest również w polu Protokół nagłówka IPv4 — ciąg dalszy*

Typ nagłówka	Pozycja w łańcuchu	Wartość	Uwagi
Koniec łańcucha	Ostatnia	59	[RFC2460]
ICMPv6	Ostatnia	58	Patrz rozdział 8.
UDP	Ostatnia	17	Patrz rozdział 10.
TCP	Ostatnia	6	Patrz rozdziały 13. – 17.
Protokoły warstw wyższych	Ostatnia	—	Kompletna lista znajduje się na stronie [AN].

5.3.1. Opcje IPv6

Nowa koncepcja realizacji opcjonalnych elementów przetwarzania datagramu IP, czyli mechanizm nagłówków rozszerzeń IPv6, stwarza dla tych elementów bardziej komfortowe środowisko, wolne od skrępowania wynikającego chociażby z ograniczonego rozmiaru nagłówka IPv4. Oznacza to m.in. przywrócenie do życia niektórych opcji IPv4, które wskutek owego skrępowania stały się praktycznie bezużyteczne w obliczu nowej rzeczywistości internetowej — mowa tu m.in. o opcjach *Trasowanie źródłowe* oraz *Rejestracja trasy*, o czym wcześniej wspominaliśmy. Protokół IPv6 dzieli swe opcje na dwie podstawowe grupy: te uwzględniane jedynie przez docelowy host (*Opcje docelowe*) oraz uwzględniane przez każdy router na trasie datagramu (zwane opcjami „Skok po skoku"). Sposób kodowania opcji jest jednakowy dla obu grup, przedstawiamy go na rysunku 5.7.



Rysunek 5.7. *Schemat TLV kodowania opcji IPv6. Pierwszy bajt, identyfikujący typ opcji, dzieli się na trzy podpola: podpole Akcja określa sposób postępowania w przypadku nierozpoznania opcji przez router lub host, podpole ZMN oznacza zezwolenie (1) albo brak zezwolenia (0) na modyfikowanie opcji przez routery pośredniczące, zaś w polu Rodzaj opcji znajduje się identyfikator określający funkcję spełnianą przez opcję. W drugim bajcie wskazany jest jawnie rozmiar danych związanych z opcją*

Trzy elementy opcji określają kolejno jej identyfikację, długość i dane, co znajduje odzwierciedlenie w akronimie TLV określającym ten schemat (to skrót od *Type-Length-Value*). Pierwszy bajt — identyfikacyjny — podzielony jest na trzy podpola, określające: sposób postępowania w przypadku, gdy opcja nie zostanie rozpoznana lub nie jest implementowana w routerze (podpole *Akcja*), zezwolenie (albo jego brak) na modyfikowanie opcji przez routery pośredniczące (*ZMN*) oraz *Rodzaj opcji* (stanowiący odpowiednik numeru opcji z IPv4). Dwa bity w podpolu *Akcja* pozwalają na zakodowanie czterech różnych wariantów postępowania, zgodnie z tabelą 5.6.

Tabela 5.6. *Dwa najbardziej znaczące bity bajta Typ opcji określają sposób reakcji routera na napotkanie nierozpoznanej lub niezaimplementowanej opcji*

Wartość	Podejmowana akcja
00	Zignorowanie opcji, kontynuowanie przetwarzania datagramu.
01	Odrzucenie datagramu bez ostrzeżenia.
10	Odrzucenie datagramu i wysłanie do jego nadawcy komunikatu ICMPv6 Parameter Problem.
11	Odrzucenie datagramu i wysłanie do jego nadawcy komunikatu ICMPv6 Parameter Problem, o ile adres docelowy datagramu IP nie jest adresem multicast.

Komunikat ostrzegający o odrzuceniu datagramu stwarza sytuację bardziej klarowną niż odrzucenie bez ostrzeżenia, jednakże w sytuacji, gdy adres docelowy datagramu jest adresem multicast, jego odrzucenie mogłoby wywołać lawinę takich komunikatów. Fakt ten jest powodem rozróżnienia akcji identyfikowanych kodami 10 i 11. Tolerowanie przez router niezrozumiałych dla niego opcji jest cechą niezmiernie pożyteczną z punktu widzenia projektowania i testowania nowych opcji: routery nieimplementujące obsługi nowej opcji po prostu ją ignorują, dzięki czemu może być ona wdrażana w sposób przyrostowy, czyli implementowana selektywnie na poszczególnych routerach.

Gdy bit *ZMN* ma wartość 1, dopuszczalne jest modyfikowanie opcji przez router (w szczególności —wyzerowanie tego bitu), w przeciwnym razie opcja musi pozostać w dotychczasowej postaci.

Zestawienie zdefiniowanych obecnie opcji IPv6 znajduje się w tabeli 5.7. W pierwszej kolumnie figuruje oficjalna nazwa opcji, w drugiej wskazany jest typ nagłówka, w ramach którego opcja może wystąpić: *Skok po skoku* (H) lub *Opcje docelowe* (D). Trzy kolejne kolumny odzwierciedlają wartości podpól *Akcja*, *ZMN* i *Rodzaj* bajta identyfikacyjnego, kolejna kolumna odzwierciedla wartość znajdującą się w drugim bajcie, wskazującym długość danych opcji. Opcja Pad1 jest wyjątkowa w tym sensie, że w jej przypadku nie występuje ów drugi bajt — opcja nie zawiera żadnych danych, jedyną przekazywaną treścią jest sam fakt jej wystąpienia, sygnalizowany przez bajt identyfikacyjny o wartości 0.

Tabela 5.7. *Opcje IPv6 w podziale na dwie grupy Skok po skoku (H) oraz Opcje docelowe (D). Kolumny Akcja, ZMN i Rodzaj ukazują wartość poszczególnych podpól bajta identyfikacyjnego, w kolumnie Długość znajduje się wartość drugiego bajta, określającego długość danych (w opcji Pad1 bajt ten nie występuje)*

Nazwa opcji	Grupa	Akcja	ZMN	Rodzaj	Długość	Dokument
Pad1	HD	00	0	0	Nie dotyczy	[RFC2460]
PadN	HD	00	0	1	Zmienna	[RFC2460]
Jumbo Payload	H	11	0	194	4	[RFC2675]
Tunnel Encapsulation Limit	D	00	0	4	4	[RFC2473]
Router Alert	H	00	0	5	4	[RFC2711]
Quick-Start	H	00	1	6	8	[RFC4782]
CALIPSO	H	00	0	7	8 lub więcej	[RFC5570]
Home Address	D	11	0	201	16	[RFC6275]

5.3.1.1. Opcje Pad1 i PadN

Wymaga się, aby każda z opcji IPv6 rozpoczynała się na offsecie stanowiącym wielokrotność 8 bajtów (w stosunku do początku datagramu[4]), więc w przypadku opcji o długości niestanowiącej tej wielokrotności konieczne jest wypełnianie przestrzeni między nimi w sposób zrozumiały dla hosta przetwarzającego nagłówek rozszerzenia. Gdy przestrzeń ta ma rozmiar jednego bajta, wypełniana jest pojedynczym bajtem zerowym, co w kategoriach tabeli 5.7 oznacza wystąpienie opcji `Pad1`. Gdy przestrzeń ta jest większa, do wypełniania używana jest opcja `PadN`, rozpoczynająca się bajtem identyfikacyjnym o wartości `1`, po którym następuje bajt długości, zliczający liczbę zer wypełniających pozostałe bajty. Tak więc np. pięciobajtowy ciąg wypełniający składa się w bajtów o wartościach `1`, `3`, `0`, `0`, `0`, a wypełnienie przestrzeni dwubajtowej może mieć postać `0`, `0` (dwie opcje `Pad1`) lub `1`, `0` (opcja `PadN`).

5.3.1.2. Opcja Jumbo Payload

W niektórych sieciach TCP/IP, m.in. w sieciach łączących superkomputery, używanie datagramów o rozmiarze 64 kB oznaczałoby zbyt duży narzut sieciowy w przypadku przesyłania masywnych porcji danych. Rozwiązaniem tego problemu jest użycie datagramów o większych rozmiarach (do 4 GB), zwanych jumbogramami. Gdy obecna jest opcja *Jumbo Payload*, rozmiar ładunku użytecznego datagramu odczytywany jest z 32-bitowego pola danych tej opcji, a nie jak zwykle z pola *Rozmiar ładunku użytecznego* w nagłówku podstawowym IPv6 — dla jumbogramu pole to powinno mieć wartość `0`.

Opcja *Jumbo Payload* może nie być implementowana przez węzły operujące na bazie łączy, których MTU nie przekracza wartości 64 kB; napotkanie jej w takiej sytuacji powoduje odrzucenie datagramu z ostrzeżeniem (vide wartość 11 w podpolu *Akcja*). Jej wystąpienie jest także nieobojętne dla protokołu TCP. Jak pokażemy w dalszym ciągu książki, integralność jego pakietów kontrolowana jest za pomocą sumy kontrolnej, uwzględniającej również rozmiar ładunku użytecznego w datagramie enkapsulującym segment TCP, więc w tym celu konieczne jest odwołanie się do właściwego pola (tego w opcji, nie tego w nagłówku podstawowym). Należy ponadto pamiętać, że większy rozmiar pakietu oznacza większe prawdopodobieństwo wystąpienia *niewykrytego* błędu (patrz [RFC2675]).

5.3.1.3. Opcja Tunnel Encapsulation Limit

Jak wyjaśnialiśmy w rozdziale 3., **tunelowanie** to enkapsulowanie pakietów jednego protokołu w pakietach innego, z naruszeniem hierarchicznej zależności warstw wynikającej z modelu odniesienia TCP/IP; przykładowo datagramy IP mogą być enkapsulowane w innych pakietach IP. Tunelowanie wykorzystywane jest do konstrukcji wirtualnych sieci nakładkowanych, w ramach których jedna z sieci (np. Internet) pełni rolę niezawodnej warstwy łącza danych dla innej warstwy IP (patrz [TWEF03]). Nic nie stoi na przeszkodzie tunelowaniu *wielopoziomowemu* — czyli sytuacji, w której pakiet enkapsulujący staje się również pakietem enkapsulowanym.

[4] Dokładniej mówiąc: w stosunku do początku nagłówka rozszerzenia. Ponieważ jednak nagłówek podstawowy ma rozmiar 40 bajtów, czyli wielokrotność 8 bajtów, oba stwierdzenia są równoważne — *przyp. tłum.*

Generalnie host lub router wysyłające pakiet IP nie mają możliwości kontroli nad liczbą poziomów tunelowania zagnieżdżonych w tym pakiecie; opisywana opcja daje im możliwość określenia limitu tych poziomów. Router zamierzający wykonać enkapsulację pakietu IPv6 w tunelu sprawdza najpierw, czy limit ów został zdefiniowany: jeśli wynosi zero, pakiet jest odrzucany, a do nadawcy (czyli punktu wejścia do aktualnego tunelu) wysyłane jest ostrzeżenie ICMPv6 (patrz rozdział 8.); jeżeli jest niezerowy, tunelowanie jest wykonywane, a w nowo utworzonym datagramie IPv6 wartość tego limitu (w polu danych opcji) zmniejszana jest o 1. W rezultacie wspomniany limit zachowuje się podobnie jak pola *Czas życia* (w nagłówku IPv4) i *Limit przeskoków* (w nagłówku IPv6), tyle że w odniesieniu do innej własności datagramu. Oczywiście, nieobecność opcji *Tunnel Encapsulation Limit* oznacza brak ograniczenia poziomu zagnieżdżenia tunelu.

5.3.1.4. Opcja Router Alert

Wystąpienie tej opcji oznacza, że datagram zawiera informacje wymagające przetwarzania przez router — podobnie jak w przypadku datagramu IPv4. Szczegółowy powód alarmowania routera zakodowany jest w 4-bajtowym polu danych opcji; znaczenie poszczególnych kodów opisane zostało na stronie [RTAOPTS].

5.3.1.5. Opcja Quick-Start

Opcja ta (w skrócie QS) wykorzystywana jest w połączeniu z eksperymentalną procedurą tzw. szybkiego startu (*Quick Start Procedure*) opisywaną w dokumencie [RFC4782]. Obsługiwana jest zarówno w ramach IPv4, jak i IPv6, jednak zaleca się ograniczenie jej stosowania do sieci prywatnych, poza kontekstem globalnego Internetu. Dane opcji zawierają zakodowaną szybkość transmisji wymaganą przez nadawcę, wartość QS TTL i inne informacje pomocnicze. Każdy router na trasie pakietu, który implementuje obsługę opisywanej opcji, zmniejsza o 1 wartość pola QS TTL i ewentualnie koryguje (zmniejsza) podaną wartość transmisji; routery nieimplementujące opcji zwyczajnie ją ignorują (vide wartość `00` w podpolu *Akcja*). Host docelowy po odebraniu datagramu wysyła do nadawcy komunikat zawierający m.in. różnicę między wartością w polu *Limit przeskoków* (*Czas życia* w IPv4) a QS TTL oraz skorygowaną ostatecznie wartość szybkości transmisji. Jeśli wspomniana różnica jest niezerowa, oznacza to istnienie na trasie datagramu routerów, które nie implementują opcji QS, w przeciwnym razie skorygowana wartość transmisji przyjmowana jest przez nadawcę jako obowiązująca na całej trasie (może być ona wyższa od tej, którą przyjąłby protokół TCP bez wykonywania procedury *Quick Start*).

5.3.1.6. Opcja CALIPSO

CALIPSO to skrót od *Common Architecture Label IPv6 Security Option* — „opcja wspólnej architektury oznaczania [datagramów] IPv6 etykietami bezpieczeństwa”. Opcja ta, opisana w dokumencie [RFC5570], wykorzystywana jest w niektórych sieciach prywatnych do etykietowania datagramów IP wskaźnikami poziomu bezpieczeństwa, wraz z informacjami towarzyszącymi. Przeznaczona jest głównie do użytku w sieciach z wielopoziomową strukturą zabezpieczeń, wykorzystywanych w krytycznych zastosowaniach bankowych, militarnych i rządowych, gdzie poziom bezpieczeństwa danych musi być jawnie wskazany za pomocą specjalnej etykiety.

5.3.1.7. Opcja Home Address

Opcja ta związana jest z mechanizmem „mobilnego IP" (o którym piszemy w podrozdziale 5.5) obejmującym zestaw procedur umożliwiających dynamiczną zmianę adresów przyporządkowywanych urządzeniom mobilnym, bez przerywania trwających połączeń w warstwach wyższych. Jedną z koncepcji mobilnego IP jest „adres domowy" (*home address*) urządzenia, czyli adres IP stanowiący pochodną prefiksu sieciowego typowej lokalizacji tego urządzenia; gdy urządzenie się przemieszcza, może ono dynamicznie otrzymywać inne, tymczasowe adresy IP. Adres domowy urządzenia, utrzymywany w datagramach jako wartość opisywanej opcji, wykorzystywany jest w kontekście tego urządzenia jako swoista identyfikacja z innymi urządzeniami mobilnymi.

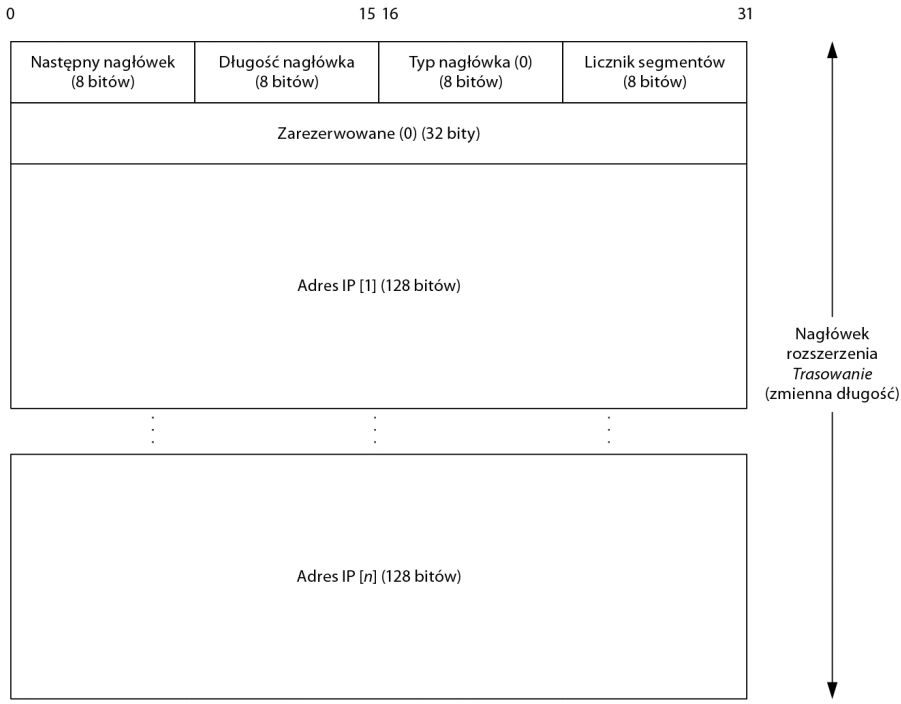
Jeśli opcja *Home address* jest używana, zawierający ją nagłówek rozszerzenia *Opcje docelowe* musi pojawić się po nagłówku *Trasowanie*, a przed nagłówkami *Fragmentacja*, *Uwierzytelnianie* i *Nagłówek ESP* (patrz rozdział 18.), o ile — oczywiście — wspomniane nagłówki w ogóle występują. Powrócimy do tej kwestii przy okazji szczegółowego omawiania mobilnego IP.

5.3.2. Nagłówek trasowania

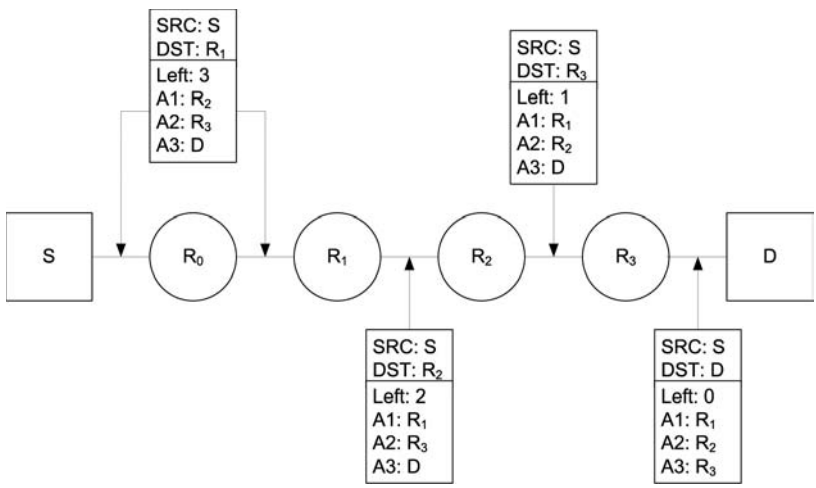
Nagłówek *Trasowanie* umożliwia nadawcy datagramu określenie listy (przynajmniej częściowej) adresów IP routerów, przez które datagram ten ma przejść na drodze do celu. Pierwotna wersja tego nagłówka, oznaczana umownie RH0, definiowana w sekcji 4.4 dokumentu [RFC2460], została na mocy dokumentu [RFC5095] wycofana z użycia jako wątpliwa pod względem bezpieczeństwa i ustąpiła miejsca wersji zwanej RH2, zdefiniowanej w związku z mobilnym IP. Rozpoczniemy od omówienia wersji RH0, wyjaśnimy przyczyny jej zarzucenia i pokażemy, czym różni się od niej wersja RH2. Format nagłówka RH0 widoczny jest na rysunku 5.8.

Centralną częścią nagłówka RH0 jest lista pozycji reprezentujących węzły, które odwiedzić musi datagram na swej trasie do celu. Standardowo pozycje te są adresami IP typu unicast rzeczonych węzłów; teoretycznie możliwe jest identyfikowanie węzłów w inny sposób, jest to jednak możliwość nieudokumentowana w standardzie IPv6 i nie będziemy jej rozważać. W polu *Typ nagłówka* znajduje się wartość `0` jako identyfikator wersji RH0 (wersja RH2 identyfikowana jest przez wartość `2`). Wartość pola *Licznik segmentów* równa jest liczbie nieodwiedzonych jeszcze pozycji z listy. Liczba właściwych pozycji poprzedzona jest polem 32 zerowych bitów, ignorowanym przez routery.

W czasie wędrówki datagramu nagłówek *Trasowanie* nie jest przetwarzany do momentu osiągnięcia węzła, którego adres równy jest zawartości pola *Docelowy adres IP* w nagłówku podstawowym. Wówczas pole *Licznik segmentów* wykorzystywane jest do określenia pozycji (na liście adresów IP) zawierającej następny żądany adres na trasie. Następuje zamiana miejscami obu adresów oraz zmniejszenie o 1 pola *Licznik segmentów*. Przykład przedstawiony na rysunku 5.9 pozwoli lepiej zrozumieć ten mechanizm.



Rysunek 5.8. *Zdeprecjonowana wersja RH0 nagłówka Trasowanie jest uogólnieniem opcji Trasowanie źródłowe i Rejestracja trasy. Nadawca datagramu umieszcza w nagłówku listę adresów węzłów, które odwiedzić musi datagram na swej trasie do węzła docelowego. Lista może mieć charakter zamknięty (tzw. ścisłe trasowanie) lub elastyczny (tzw. luźne trasowanie) — w tym drugim przypadku trasa między sąsiednimi pozycjami nie musi być pokonana w postaci pojedynczego przeskoku. Pole Docelowy adres IP w nagłówku podstawowym modyfikowane jest przy każdym przeskoku tak, by wskazywać następny węzeł na trasie*



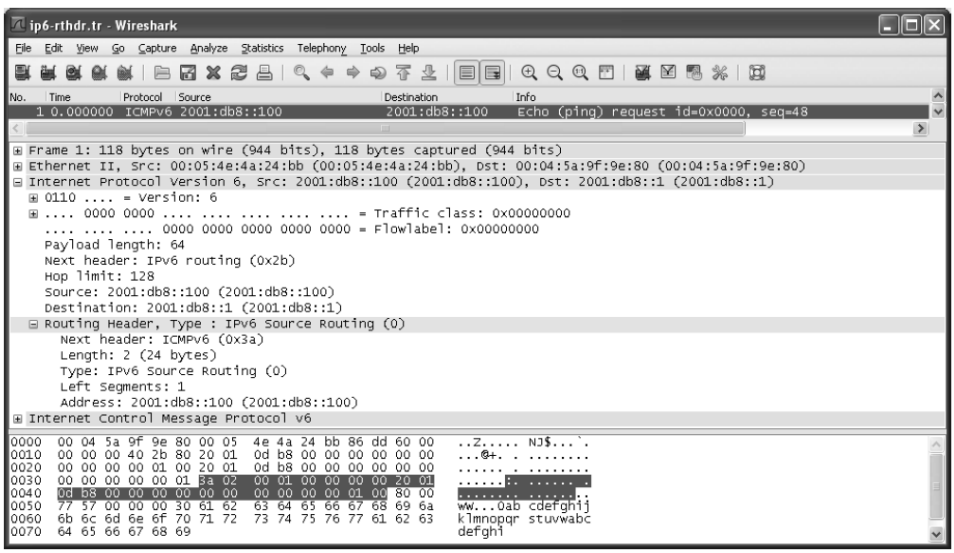
Rysunek 5.9. *Przykład użycia nagłówka trasowania w wersji RH0. Węzeł S wysyła datagram IPv6 do węzła D, wytyczając trasę prowadzącą przez węzły pośrednie R1, R2 i R3. Zauważmy, że adresy tych węzłów pojawiają się kolejno w polu Docelowy adres IP nagłówka podstawowego*

Na rysunku tym widzimy węzeł nadawczy **S** oraz datagram z nagłówkiem *Trasowanie*, w którym określono ciąg adresów $\mathbf{R}_1$, $\mathbf{R}_2$, $\mathbf{R}_3$ i **D**. W pole adresu docelowego (*DST*) wpisany zostaje adres R1, w pole *Licznik segmentów* — wartość `3` i datagram rusza w drogę. Router $\mathbf{R}_0$ nie napotyka swego adresu w polu *DST*, więc nagłówek *Trasowanie* pozostaje nienaruszony. Datagram dociera następnie do routera $\mathbf{R}_1$, pole *DST* zamienia się zawartością z pierwszą pozycją na liście, *Licznik segmentów* zostaje zmniejszony do `2` i datagram wędruje do routera $\mathbf{R}_2$. Tu sytuacja się powtarza: pole *DST* i przedostatnia pozycja zamieniają się miejscami, licznik segmentów przyjmuje wartość `1`. Po dotarciu datagramu do $\mathbf{R}_3$ następuje kolejne, ostatnie powtórzenie scenariusza. Pole *DST* odzyskuje pierwotną zawartość, licznik segmentów zostaje wyzerowany, datagram zmierza ku węzłowi docelowemu **D**.

Opisany scenariusz możemy zaobserwować za pomocą programu Wireshark uwidaczniającego efekt „pingowania" wywołany za pomocą polecenia `ping6` w Windows XP; w Windows Vista i Windows 7 dostępne jest tylko standardowe polecenie `ping`, zapewniające obsługę IPv6.

```
C:\> ping6 -r -s 2001:db8::100 2001:db8::1
```

Powyższe polecenie powoduje wysyłanie żądań `ping` z adresu źródłowego `2001:db8::100` do adresu docelowego `2001:db8::1`; parametr `-r` powoduje dołączenie nagłówka RH0. Wychodzący pakiet żądania widoczny jest w oknie programu Wireshark (patrz rysunek 5.10).



Rysunek 5.10. *Polecenie ping powoduje pojawienie się pakietu żądania Echo protokołu ICMPv6. W polu Następny nagłówek nagłówka podstawowego IPv6 znajduje się wartość 0x2b (dziesiętnie 43), co zgodnie z tabelą 5.5 identyfikuje nagłówek Trasowanie; licznik segmentów równy jest 1, lista adresów zawiera tylko jedną pozycję 2001:db8::100. Następnym w kolejności nagłówkiem jest ICMPv6 (wartość 0x3a, dziesiętnie 58)*

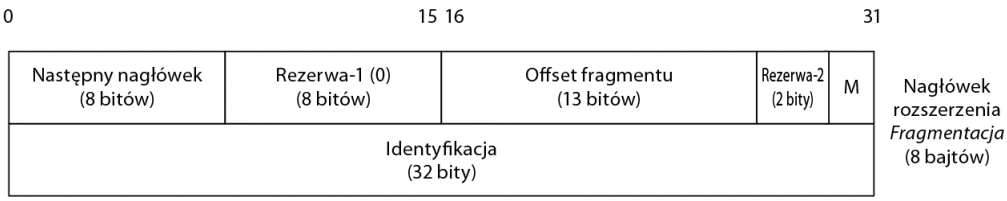
Komunikat `ping` ma postać pakietu żądania *Echo* protokołu ICMPv6 (patrz rozdział 8.). W datagramie IP komunikat ten poprzedzony jest nagłówkiem *Trasowanie* z wartością `0` w polu *Typ nagłówka*, mamy więc do czynienia z wersją RH0. Na liście adresów znajduje się tylko jedna pozycja — `2001:db8::100`.

Wspominaliśmy wcześniej, że wersja RH0 wycofana została z użytku jako niebezpieczna. Zauważmy otóż, że obecna w nagłówku lista adresów IP może być kształtowana w zasadzie dowolnie, m.in. nic nie stoi na przeszkodzie *wielokrotnemu użyciu tych samych adresów*. Aranżując odpowiednio owe powtórzenia, możemy doprowadzić do narastającego ruchu oscylacyjnego, co efektywnie równa się atakowi przeciążeniowemu (DoS). W wersji RH2 wspomniana lista adresów może zawierać *co najwyżej jedną pozycję*, a *Licznik segmentów* musi mieć wartość `0` — i jest to jedyna różnica w stosunku do wersji RH0, oprócz — oczywiście — wartości `2` w polu *Typ nagłówka*.

5.3.3. Nagłówek fragmentacji

Gdy rozmiar datagramu przekracza wartość PMTU dla ścieżki od nadawcy do hosta docelowego, datagram ten dzielony jest na **fragmenty**. O znaczeniu PMTU wspominaliśmy już w rozdziale 3., szczegółowo omawiamy ją w rozdziale 13., tu ograniczymy się tylko do przypomnienia, że protokół IPv6 narzuca dolne ograniczenie 1280 bajtów na jej wartość (zgodnie z sekcją 5. dokumentu [RFC2460]). W ramach protokołu IPv4 fragmentacji datagramu dokonywać mógł każdy host lub router, jeśli datagram ten był zbyt duży w kontekście MTU łącza prowadzącego do następnego węzła; informacja związana z fragmentacją obecna była w drugim 32-bitowym słowie nagłówka IPv4, obecnego w każdym fragmencie. Protokół IPv6 ogranicza poniekąd tę swobodę — fragmentowanie datagramu wykonywać może wyłącznie jego nadawca, a informacja opisująca szczegóły fragmentacji umiejscowiona jest w ramach odpowiedniego nagłówka rozszerzenia.

Nagłówek rozszerzenia *Fragmentacja* zawiera te same informacje o fragmentacji, które znajdowały się w nagłówku IPv4, lecz pole *Identyfikacja* rozszerzone zostało do 32 bitów, co umożliwia współegzystencję w sieci większej liczby pofragmentowanych pakietów. Format nagłówka *Fragmentacja* przedstawiono na rysunku 5.11.



Rysunek 5.11. *Nagłówek rozszerzenia Fragmentacja. Pole identyfikacyjne jest dwukrotnie większe niż w nagłówku IPv4, offset fragmentu oznacza przesunięcie fragmentu, liczone w jednostkach 8-bajtowych, względem początku części fragmentowalnej oryginalnego pakietu. Bit M jest wyzerowany w ostatnim fragmencie pakietu i ustawiony na 1 w pozostałych fragmentach*

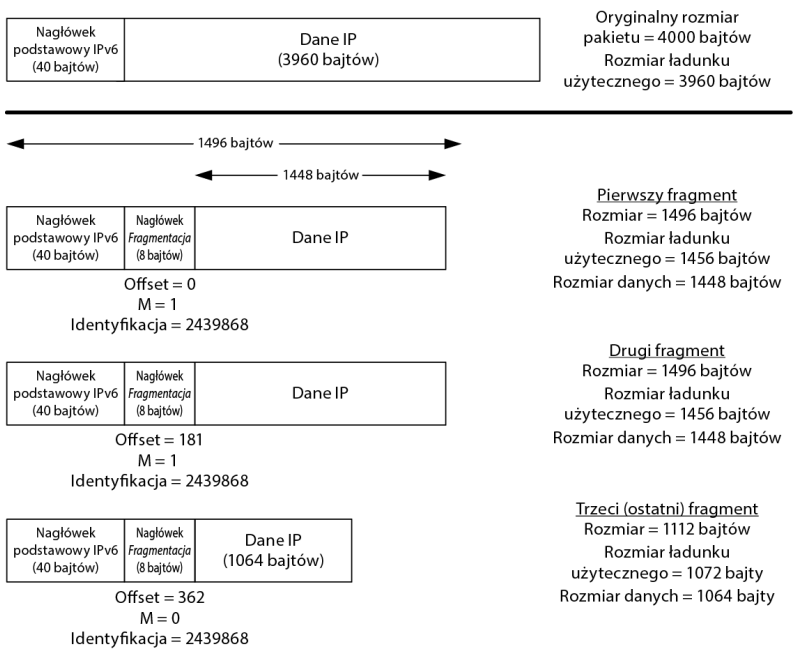
Dwa pola rezerwy zawierają zera i są ignorowane przez odbiorcę pakietu. *Offset fragmentu* określa umiejscowienie fragmentu w oryginalnym pakiecie, a konkretnie — przesunięcie tego fragmentu względem początku tzw. części fragmentowalnej, liczone w jednostkach 8-bajtowych. Zerowa wartość bitu *M* oznacza ostatni fragment pakietu.

W datagramie poddawanym fragmentacji (zwanym „oryginalnym datagramem") wyróżniamy dwie części: początkowa, *niefragmentowalna* część obejmuje nagłówek podstawowy oraz wszelkie nagłówki rozszerzeń podlegające przetwarzaniu przez węzły pośrednie (czyli wszystkie nagłówki do nagłówka *Trasowanie* włącznie albo nagłówek

opcji *Skok po skoku*, jeśli tylko on występuje w pakiecie). Pozostała, *fragmentowalna* część pakietu obejmuje nagłówek *Opcje docelowe*, nagłówki protokołów warstw wyższych i ładunek użyteczny.

Każdym fragment powstający z podziału datagramu oryginalnego rozpoczyna się kopią jego części niefragmentowalnej z tą różnicą, że w polu *Rozmiar ładunku użytecznego* znajduje się rozmiar fragmentu, nie rozmiar oryginalnego ładunku użytecznego[5]. Bezpośrednio za tą kopią znajduje się nagłówek *Fragmentacja*, z odpowiednio wypełnionym polem *Offset fragmentu* (równym zero w pierwszym fragmencie) i odpowiednio ustawionym bitem *M* (wyzerowanym w ostatnim fragmencie). Pole identyfikacyjne jest kopią pierwowzoru z części niefragmentowalnej.

Proces fragmentowania przykładowego datagramu przedstawiono poglądowo na rysunku 5.12. Datagram zawierający ładunek użyteczny o rozmiarze 3960 bajtów zostaje pofragmentowany w celu przetransmitowania przez ścieżkę, której PMTU wynosi 1500 (to wartość typowa dla Ethernetu). Rozmiar fragmentu nie może więc przekraczać 1500 bajtów, przy czym rozmiar danych fragmentu musi być wielokrotnością 8 bajtów. Odliczając 40 bajtów na nagłówek podstawowy i 8 bajtów na nagłówek *Fragmentacja* dostajemy limit 1452 bajtów na rozmiar danych, co po zaokrągleniu w dół do wielokrotności 8 daje 1448 bajtów. Ostatecznie maksymalny rozmiar pakietu równy jest 40 + 8 + 1448 = 1496 bajtów.



Rysunek 5.12. *Przykład podziału pakietu z ładunkiem użytecznym 3960 bajtów na trzy fragmenty o rozmiarze danych nie większym niż 1448 bajtów. We wszystkich fragmentach oprócz ostatniego bit M ma wartość 1. Offset fragmentu wyrażony jest w jednostkach 8-bajtowych, co dla drugiego i trzeciego fragmentu daje (odpowiednio) 181 · 8 = 1448 oraz 362 · 8 = 2896 bajtów. Schemat fragmentacji jest więc podobny do tego z IPv4, choć różni się od niego obecnością nagłówka rozszerzenia*

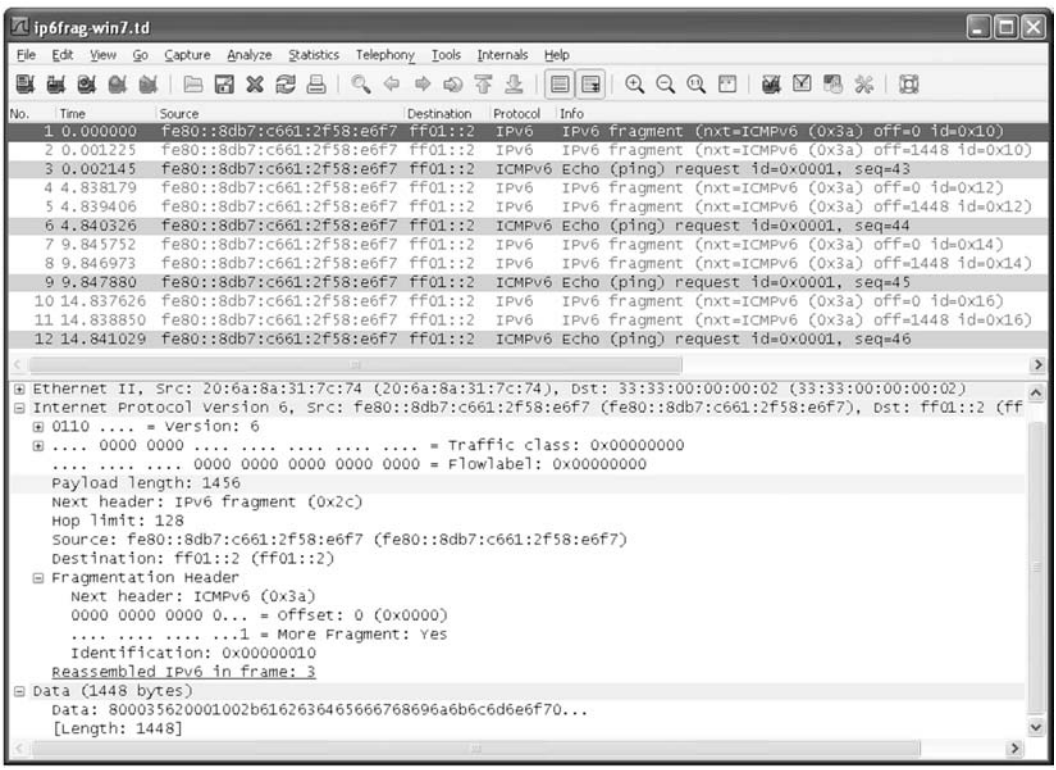
[5] Rozmiar ładunku użytecznego plus 8 bajtów nagłówka *Fragmentacja — przyp. tłum.*

Proces odwrotny do fragmentacji, czyli rekonstrukcja oryginalnego datagramu na podstawie fragmentów, nosi nazwę **defragmentacji** lub **reasemblacji**. Host docelowy musi — oczywiście — uprzednio otrzymać wszystkie fragmenty; podobnie jak w protokole IPv4, mogą one nadchodzić w dowolnej kolejności, niekoniecznie naturalnej, czyli według wzrastających offsetów (patrz rozdział 10.); do rozpoczęcia deasemblacji jest więc konieczne otrzymanie ostatniego fragmentu (rozpoznawanego na podstawie zerowej wartości bitu *M*) i wszystkich fragmentów poprzednich.

Fragmentację możemy zaobserwować w praktyce, inicjując wysyłanie datagramów IP za pomocą polecenia `ping` z odpowiednimi parametrami i podglądając transmitowane dane za pomocą programu Wireshark. W systemie Windows 7 wspomniane polecenie ma postać:

```
C:\> ping -l 3952 ff01::2
```

a jego konsekwencje wyglądają w oknie programu Wireshark podobnie do tych z rysunku 5.13.

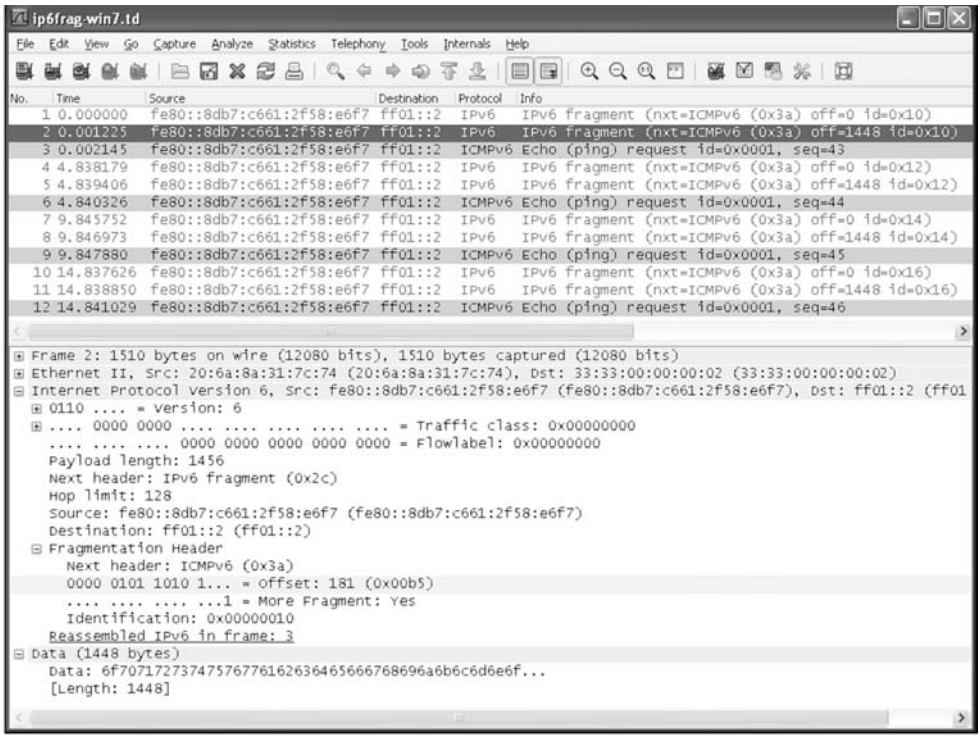


Rysunek 5.13. *Program ping generuje pakiety ICMPv6 (patrz rozdział 8.) zawierające 3960-bajtowy ładunek użyteczny IPv6. Pakiety te są fragmentowane w celu zdolności ich przetransmitowania przez łącze ethernetowe o wartości MTU 1500 bajtów*

Na rysunku widzimy pofragmentowany komunikat żądania Echo ICMPv6, wysyłany na adres docelowy multicast `ff01::2`. Fragmentacja jest konieczna, ponieważ parametr `-l 3952` powoduje generowanie pakietów ICMPv6 niosących ładunek użyteczny o rozmiarze

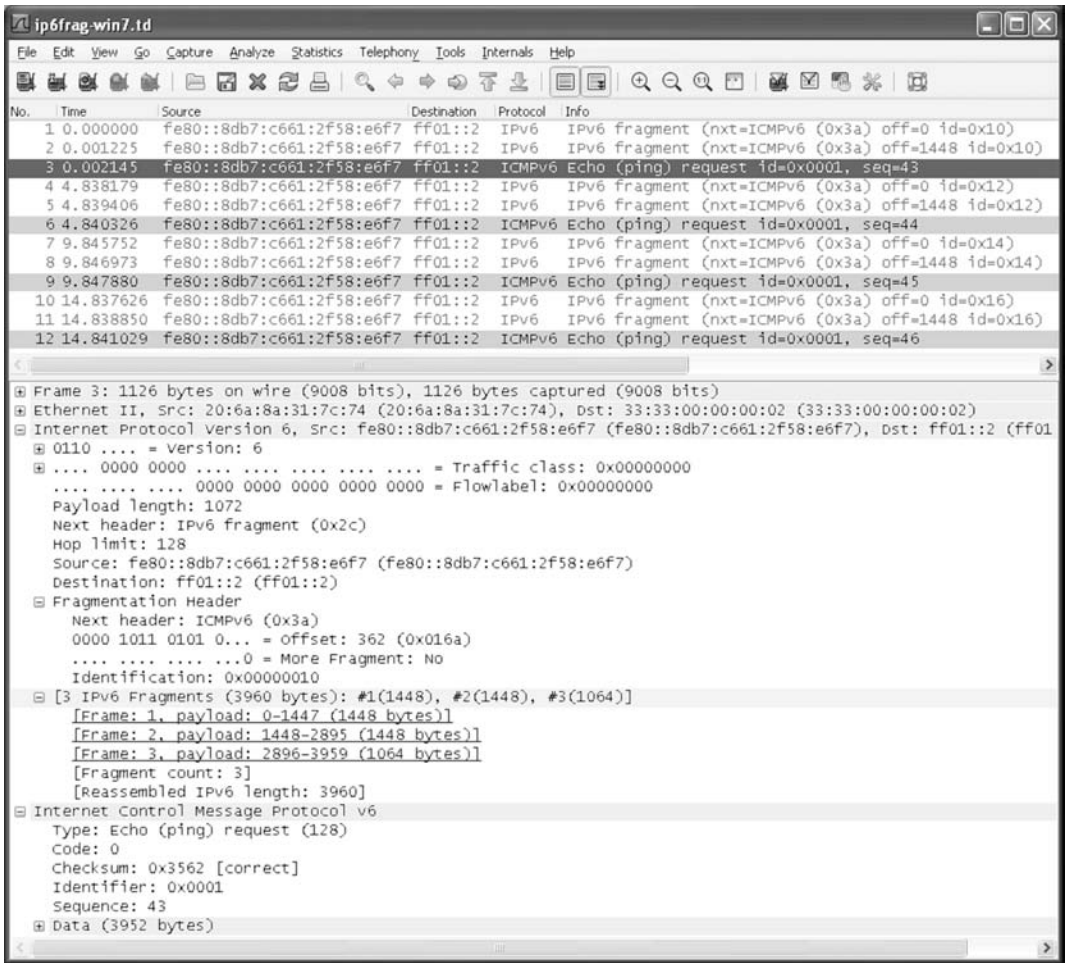
3952 bajtów. Razem z 8-bajtowym nagłówkiem ICMPv6 daje to pakiet ICMPv6 o wielkości 3960 bajtów. Pakiet ten staje się następnie ładunkiem użytecznym datagramu IPv6. Widoczny w nagłówku podstawowym IPv6 adres źródłowy jest adresem IPv6 lokalnym dla łącza. Dla określenia docelowego adresu multicast warstwy łącza danych wykorzystywana jest specyficzna dla IPv6 procedura mapująca, opisywana w rozdziale 9. Gdy wszystkie fragmenty danego pakietu dotrą do odbiornika, jakim jest program Wireshark, zostaną poskładane w całość, jaką jest oryginalny datagram IPv6.

Na rysunku 5.14 zobaczyć możemy szczegóły drugiego fragmentu. Zgodnie z oczekiwaniami, rozpoczyna się on podstawowym nagłówkiem IPv6, w którym wykazany jest ładunek użyteczny o rozmiarze 1448 bajtów, zaś pole *Następny nagłówek* zawiera kod 44 (0x2c) identyfikujący (zgodnie z tabelą 5.5) nagłówek rozszerzenia *Fragmentacja*. Kolejnym nagłówkiem w łańcuchu jest nagłówek protokołu ICMPv6, nie ma więc dalszych nagłówków rozszerzeń. Wartość 181 w polu *Offset fragmentu* umiejscawia początek tego fragmentu na bajcie przesuniętym o 181 · 8 = 1448 bajtów w stosunku do początku części fragmentowalnej oryginalnego datagramu. Ustawienie na 1 bitu *M* (sygnalizowane przez Wireshark jako wartość *Yes* zatytułowana *More Fragment*) oznacza, że wyświetlany fragment nie jest ostatnim fragmentem oryginalnego datagramu.



Rysunek 5.14. *Zaznaczony drugi fragment datagramu IPv6, zawierający 1448 bajtów ładunku użytecznego, w tym 8-bajtowy nagłówek rozszerzenia Fragmentacja. Obecność tego nagłówka jest świadectwem pofragmentowania datagramu przez nadawcę, wartość 181 w polu offsetu oznacza natomiast, że niniejszy fragment odzwierciedla część oryginalnego datagramu rozpoczynającą się na przesunięciu 181 · 8 = 1448 bajtów względem początku części fragmentowalnej. Wartość 1 bitu M oznacza, niniejszy fragment nie jest ostatni. Wszystkie fragmenty tego samego pakietu mają identyczną wartość w polu Identyfikacja, w tym przypadku 2*

Analogiczne obserwacje poczynić możemy w odniesieniu do trzeciego fragmentu (patrz rysunek 5.15). Z pola *Offset fragmentu* odczytujemy 362 · 8 = 2896 jako przesunięcie tego fragmentu względem początku części fragmentowalnej. W polu *Rozmiar ładunku użytecznego* widnieje wartość 1072; jeśli odejmiemy od niej 8 jako rozmiar nagłówka *Fragmentacja*, otrzymamy 1064 jako rozmiar „netto" tegoż ładunku. Na podstawie offsetów i rozmiarów poszczególnych fragmentów Wireshark odtwarza ich chronologiczną aranżację, w sposób czytelny dla użytkownika.



Rysunek 5.15. *Zaznaczony ostatni fragment datagramu IPv6, zawierający 1072 bajty ładunku użytecznego, w tym 8-bajtowy nagłówek rozszerzenia Fragmentacja. Wartość 0 bitu M kwalifikuje niniejszy fragment jako ostatni, ponadto sumując jego offset (362 · 8 = 2896) i rozmiar „netto" ładunku użytecznego (bez uwzględniania nagłówka Fragmentacja) równy 1064 otrzymamy wartość 2896 + 1064 = 3960 jako rozmiar oryginalnego ładunku użytecznego (złożonego z 8-bajtowego nagłówka pakietu ICMPv6 i 3956 bajtów danych tego pakietu)*

Na podstawie dotychczasowego opisu fragmentacji datagramu IPv6 i na podstawie analizy prezentowanego przykładu możemy ocenić narzut — w postaci dodatkowych danych wymagających transmitowania — jaki wprowadza fragmentacja do protokołu IPv6. Po

pierwsze, w każdym fragmencie znajduje się kopia podstawowego nagłówka IPv6, zamiast więc 40 bajtów jednego egzemplarza tegoż nagłówka mamy teraz trzy egzemplarze, co oznacza narzut w ilości (3–1)*40 = 80 bajtów. W każdym fragmencie obecny jest ponadto 8-bajtowy nagłówek rozszerzenia *Fragmentacja*, co oznacza narzut 3*8 = 24 dodatkowych bajtów. Ponadto, ponieważ do przeniesienia fragmentów potrzebne będą trzy ramki ethernetowe (zamiast jednej), musimy uwzględnić 18-bajtowy narzut (14-bajtowy nagłówek + 4-bajtowa suma FCS) na każdą z dwóch dodatkowych ramek. Łącznie zatem warstwa łącza danych otrzyma dodatkowo 80+24+2*18 = 140 bajtów do przetransferowania.

5.4. Forwardowanie datagramów IP

Forwardowanie datagramów IP jest pod względem koncepcyjnym mało skomplikowane, szczególnie z perspektywy hosta. Jeśli węzeł docelowy przyłączony jest do hosta bezpośrednio (np. za pomocą łącza punkt-punkt) lub za pośrednictwem współdzielonej sieci (np. Ethernetu), datagram przesyłany jest bezpośrednio, bez użycia routera. W przeciwnym razie host wysyła datagram do **routera domyślnego** (*default router*), powierzając temu routerowi zadanie dostarczenia datagramu do celu. Ten prosty schemat funkcjonuje w przeważającej większości hostów.

W tym podrozdziale przeanalizujemy szczegóły tej prostej operacji, by następnie przyjrzeć się operacji, która już tak prosta nie jest. Zauważmy na wstępie, że większość współczesnych komputerów ma dwojakie oblicze, każdy z nich funkcjonować może zarówno jak typowy host, jak i w charakterze routera — wiele z domowych sieci wykorzystuje komputery PC przyłączone do Internetu właśnie jako routery (a także jako firewalle, którym poświęcamy rozdział 7.). Zasadniczą cechą odróżniającą (w kontekście IP) host od routera jest traktowanie „obcych" datagramów: host, w przeciwieństwie do routera, nigdy nie zajmuje się forwardowaniem datagramów, których sam nie wytworzył. W naszym ogólnym schemacie protokół IP otrzymywać może datagramy od innych protokołów funkcjonujących w tym samym komputerze (m.in. TCP i UDP) bądź z interfejsu sieciowego.

Wykonywane przez warstwę IP forwardowanie datagramów wiąże się z utrzymywaniem w pamięci hosta lub routera pomocniczych danych, zwanych popularnie **tablicą trasowania** (*routing table*) lub **tablicą forwardowania** (*forwarding table*). Gdy datagram zostaje odebrany przez interfejs sieciowy, protokół IP sprawdza najpierw, czy zawarty w tym datagramie docelowy adres IP jest adresem własnym węzła, w którym protokół ten funkcjonuje (czyli jednym z adresów przypisanych interfejsom sieciowym tegoż węzła), bądź „pasującym" do węzła adresem multicast lub broadcast. Jeśli tak, datagram dostarczany jest do modułu implementującego protokół wskazywany w polu *Protokół* (w IPv6 w polu *Następny nagłówek*) w nagłówku tegoż datagramu. W przeciwnym razie datagram potraktowany zostać może w dwojaki sposób:

- jeśli warstwa IP skonfigurowana jest do pełnienia funkcji routera, datagram jest forwardowany, czyli przetwarzany według scenariusza opisywanego w punkcie 5.4.2,
- jeśli warstwa IP nie implementuje funkcji routera, datagram jest odrzucany; w niektórych sytuacjach do nadawcy wysyłany jest komunikat protokołu ICMP z informacją o wystąpieniu błędu.

5.4.1. Tablica forwardowania

Specyfikacja protokołu IP nie precyzuje szczegółów danych składających się na tablicę forwardowania, pozostawiając w tym zakresie swobodę autorom konkretnych implementacji. Reguły protokołu IP nieuchronnie prowadzą jednak do struktury, która (przynajmniej koncepcyjnie) powinna być wektorem pozycji składających się z czterech następujących pól. Oto one.

- **Przeznaczenie**, czyli 32-bitowa (dla IPv4) lub 128-bitowa (dla IPv6) wartość spełniająca wraz z polem maski (patrz następna pozycja) funkcje *dopasowywania* adresu docelowego (szczegóły już za chwilę). Może to być np. wartość zerowa, odpowiadająca „trasie domyślnej" reprezentującej wszystkie możliwe miejsca przeznaczenia, albo pełny adres IP w przypadku „trasy hosta" reprezentującej pojedyncze miejsce przeznaczenia.
- **Maska**, czyli ciąg bitów o rozmiarze takim samym jak pole **Przeznaczenie**. Zostaje pomnożona bitowo (AND) z docelowym adresem IP zawartym w datagramie, po czym następuje próba dopasowania wyniku tego mnożenia do któregoś z pól **Przeznaczenie** w tablicy forwardowania.
- **Adres następnego przeskoku**, czyli adres IP następnego węzła (routera lub hosta), do którego przekazany zostanie datagram.
- **Identyfikator interfejsu** wykorzystywany przez warstwę IP do wyboru interfejsu, który powinien dostarczyć datagram do następnego przeskoku; może to być np. karta ethernetowa (tradycyjna lub bezprzewodowa) albo interfejs PPP skojarzony z portem szeregowym. Gdy węzeł wysyłający datagram jest jednocześnie jego twórcą, pole to może posłużyć do określenia, który z adresów IP powinien zostać wpisany do nagłówka datagramu jako adres źródłowy (do tej kwestii powrócimy w podpunkcie 5.6.2.1).

Forwardowanie pakietów IP odbywa się na zasadzie „skok po skoku": jak można się zorientować z dotychczasowego opisu, host ani router nie zawierają informacji o kompletnej trasie datagramu, aż do węzła docelowego, lecz jedynie informację konieczną do właściwego wyboru następnego węzła na tej trasie (wyjątkiem jest — oczywiście — sytuacja, gdy następny węzeł jest jednocześnie docelowym, bezpośrednio połączonym z bieżącym). Węzeł „właściwie wybrany" to węzeł z założenia znajdujący się „bliżej" węzła docelowego na wspomnianej trasie i połączony z węzłem bieżącym w sposób umożliwiający bezpośrednie przekazanie datagramu. Od trasy datagramu wymaga się natomiast, by była pozbawiona cykli, w przeciwnym razie datagram, zamiast dotrzeć do celu, mógłby krążyć po sieci, aż do wyczerpania się limitu ustalonego przez pole *Czas życia* (w IPv6 pole *Limit przeskoków*). Zapewnienie poprawnej postaci tablic forwardowania (m.in. we wspomnianych aspektach) jest zadaniem **protokołów trasowania** (*routing protocols*) — RIP, OSPF, BGP czy IS-IS, że ograniczymy się do wymienienia tylko kilku najbardziej znanych (więcej informacji na temat protokołów tej kategorii znajdą czytelnicy np. w publikacji [DC05]).

5.4.2. Szczegóły forwardowania

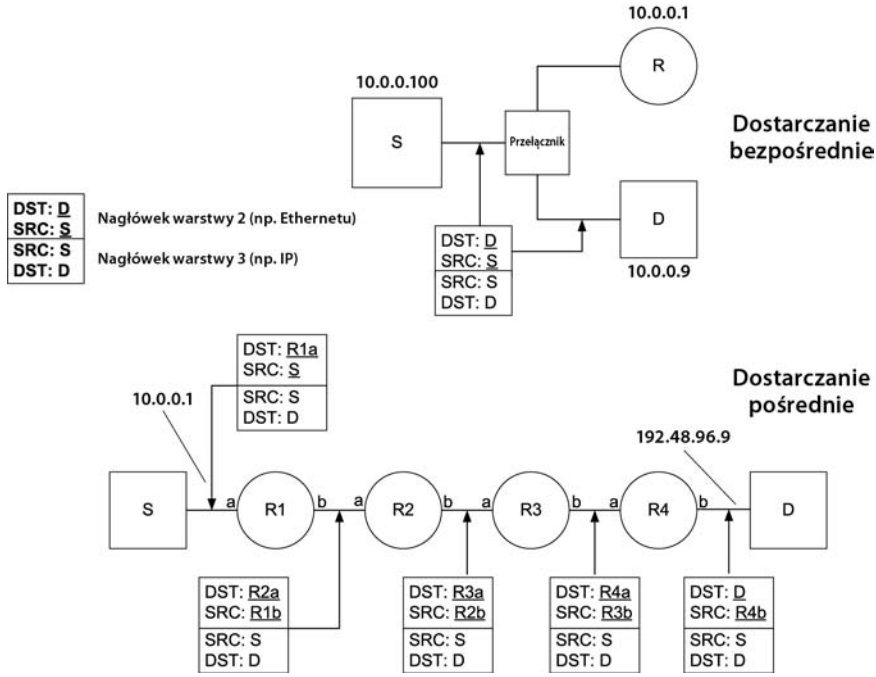
Gdy warstwa IP implementowana przez host lub router zamierza wysłać datagram IP do następnego przeskoku, rozpoczyna od konfrontacji adresu docelowego tegoż datagramu (oznaczmy ten adres przez *D*) z kolejnymi pozycjami swej tablicy forwardowania. Pozycję o numerze *j* we wspomnianej tablicy uznaje się za *pasującą* do adresu *D*, jeśli koniunkcja bitowa (AND) adresu *D* i pola *Maska* (m_j) na tej pozycji da wynik równy zawartości pola *Przeznaczenie* (d_j), czyli gdy $D \wedge m_j = d_j$. Spośród wszystkich pozycji pasujących do adresu *D* wybrana zostaje ostatecznie ta, której pole m_j zawiera najwięcej bitów jedynkowych. Powyższy algorytm nosi nazwę *dopasowania do najdłuższego prefiksu* (*longest prefix match*), bo maski w tablicy forwardowania są maskami podsieci, a maska podsieci stanowi (jak wyjaśnialiśmy w rozdziale 2.) konkatenację ciągu (być może pustego) bitów jedynkowych z ciągiem (być może pustym) bitów zerowych. Wyjaśnia to związek z długością prefiksu i liczbą „jedynek" w masce.

Jeżeli w tablicy forwardowania nie zostanie odnaleziona przynajmniej jedna pozycja pasująca do adresu *D*, datagram uznawany jest za *niedostarczalny* (*undeliverable*). Jeśli sytuacja taka ma charakter lokalny — niemożliwość forwardowania datagramu stwierdzona została przez host, który datagram ów wygenerował — odnośna aplikacja otrzymuje standardowy komunikat w rodzaju „Host nieosiągalny". Gdy opisana sytuacja wydarzy się w obrębie routera, zostaje wysłany odpowiedni komunikat ICMP do hosta będącego nadawcą datagramu.

Powróćmy jeszcze do dopasowywania adresu *D* do pozycji tablicy forwardowania: może się zdarzyć, że na kilku pozycjach stwierdzona zostanie największa liczba bitów jedynkowych w masce i żadna z nich nie zasługuje tym samym na miano „najlepszej". Może się tak zdarzyć, jeśli dla hosta określono kilka domyślnych routerów, bo w sieci zastosowano multihoming, czyli przyłączenie do kilku dostawców Internetu. Standard IP nie określa żadnej szczególnej zasady rozstrzygania takich remisów, najczęściej więc implementacje wybierają pierwszą z kandydatur. Bardziej inteligentna strategia wyboru mogłaby realizować podział ruchu między kilka tras, być może z uwzględnieniem ich oczekiwanego obciążenia. Jak dowodzą analizy przedstawione w publikacji [THL06], multihoming może być korzystny nie tylko dla dużych przedsiębiorstw, lecz także dla użytkowników domowych.

5.4.3. Przykłady

Zilustrujmy konkretnymi przykładami opisane reguły forwardowania, a szczególnie dopasowywanie do najdłuższego prefiksu w tablicach trasowania. Przeanalizujemy dwa przypadki dostarczania datagramu: lokalne — gdzie host wysyłający i docelowy znajdują się w tej samej sieci (tzw. **dostarczanie bezpośrednie** — *direct delivery*) — oraz zdalne, angażujące serię routerów (zwane **dostarczaniem pośrednim** — *indirect delivery*). Oba schematy zilustrowano na rysunku 5.16.



Rysunek 5.16. *Dostarczanie bezpośrednie nie wymaga użycia routera — datagram IP enkapsulowany jest w ramce warstwy łącza danych, do której wprost wpisywane są adresy sprzętowe źródłowy i docelowy. Dostarczanie pośrednie wiąże się z wykorzystaniem routera — dane forwardowane są do routera przy użyciu jego adresu sprzętowego jako docelowego w ramce. Adres IP routera nie pojawia się w datagramie IP (chyba że router jest węzłem nadawcą lub węzłem docelowym albo użyta została opcja Trasowanie źródłowe*

5.4.3.1. Dostarczanie bezpośrednie

Rozpoczniemy od przykładu dostarczania bezpośredniego, zilustrowanego w górnej części rysunku 5.16. Nasz host **S**, zarządzany przez system Windows XP, posiada adres IPv4 `10.0.0.100` oznaczony tu *S*, a jego karta sieciowa posiada adres MAC, który oznaczymy *S* (podkreślone); host ten wysyła właśnie datagram IP do hosta linuksowego **D** o adresie IP równym `10.0.0.9`, który oznaczamy przez *D*, i adresie MAC, który oznaczamy przez *D* (podkreślone). Oba hosty połączone są za pomocą przełącznika. Tablica trasowania utrzymywana przez warstwę IP hosta **S** ma postać przedstawioną w tabeli 5.8.

Tabela 5.8. *Tablica trasowania w hoście S zawiera dwie pozycje. Hostowi S przydzielony został adres 10.0.0.100/25. Adresy docelowe z przedziału od 10.0.0.1 do 10.0.0.126 dopasowywane są do drugiej pozycji (jako zawierającej dłuższy prefiks) i stanowią podstawę dostarczania bezpośredniego datagramów. Pozostałe adresy docelowe pasują wyłącznie do pierwszej pozycji, prowadzącej do routera R o adresie 10.0.0.1*

Przeznaczenie (di)	Maska (mi)	Następny przeskok	Interfejs
`0.0.0.0`	`0.0.0.0`	`10.0.0.1`	`10.0.0.100`
`10.0.0.0`	`255.255.255.128`	`10.0.0.100`	`10.0.0.100`

Dopasowywanie adresu *D* równego 10.0.0.9 daje wynik pozytywny dla obu pozycji tablicy; w pierwszej masce liczba bitów jedynkowych wynosi 0, w drugiej 25, zdecydowanie więc wybrana zostaje druga pozycja, w której adres następnego przeskoku 10.0.0.100 równy jest adresowi hosta **S**; jest to jednocześnie sygnał dla hosta **S**, że datagram powinien zostać dostarczony w sposób bezpośredni, bez używania jakichkolwiek routerów.

Datagram zostaje więc poddany przez host **S** enkapsulacji w ramce warstwy łącza danych, przeznaczonej dla hosta **D**. Jeśli adres *D* jest dla hosta **S** nieznany, musi zostać rozpoznany przez odpowiedni protokół — w wersji IPv4 jest to protokół ARP, opisywany w rozdziale 4., w wersji IPv6 protokół ICMPv6, a dokładniej jego komunikaty *Neighbor Solicitation,* którymi zajmiemy się w rozdziale 8. Adres *D* wpisany zostaje następnie w pole *DST* wspomnianej ramki — w procesie jej dostarczania przełącznik wykorzystuje wyłącznie adresy warstwy łącza danych, abstrahując zupełnie od adresów IP w enkapsulowanym datagramie.

5.4.3.2. Dostarczanie pośrednie

Przeanalizujmy teraz przypadek przedstawiony w dolnej części rysunku 5.16. Z naszego hosta wysyłany jest datagram IP na adres 192.48.96.9, który na swej trasie napotka cztery routery pośredniczące. Jedynie pierwsza pozycja w tablicy trasowania (ta z zerową maską) pasuje do wspomnianego adresu docelowego, jako następny przeskok wykazywany jest tu adres 10.0.0.1, przyporządkowany „stronie a" routera R1. Sytuacja taka jest typowa dla większości sieci domowych.

Jak pamiętamy, w przypadku dostarczania bezpośredniego adresy IP w nagłówku datagramu — źródłowy i docelowy — odpowiadają adresom (odpowiednio) węzła nadawcy i węzła docelowego i podobnie jest z adresami sprzętowymi (np. ethernetowymi). W przypadku dostarczania pośredniego tak samo ma się rzecz z adresami IP, lecz już nie z adresami sprzętowymi: adresy źródłowy i docelowy w ramce odnoszą się do węzłów wyznaczających kolejny odcinek jej trasy; docelowy adres ethernetowy w ramce wychodzącej z hosta S wskazuje interfejs sieciowy po „stronie a" routera **R1**, ten opatrzony adresem IP 10.0.0.1. „Przeliczeniem" adresu IP na adres sprzętowy zajmuje się jeden z wymienionych wcześniej protokołów (ARP albo ICMPv6) działających w sieci łączącej host S z routerem **R1**. Gdy tylko host S otrzyma odpowiedź od routera **R1**, wyśle do niego ramkę enkapsulującą datagram IP; przetworzenie tej ramki odbędzie się wyłącznie na podstawie adresów warstwy łącza danych (a konkretnie — docelowego adresu ethernetowego).

Router **R1** po otrzymaniu wspomnianej ramki wyłuskuje z niej rzeczony datagram IP i przystępuje do przeglądania swej tablicy trasowania w celu dopasowania adresu docelowego 192.48.96.9 odczytanego z nagłówka datagramu. Zakładamy, że tablica trasowania zawiera dwie pozycje, jak w tabeli 5.9.

Router **R1** po stwierdzeniu, że adres docelowy datagramu (192.48.96.9) nie jest jego własnym adresem IP, przystępuje do forwardowania otrzymanego datagramu. Pole *Następny przeskok* dopasowanej pozycji wskazuje na adres 70.231.159.254, przypisany do „strony a" routera **R2** — jest to adres sieci o (cokolwiek skomplikowanej) nazwie adsl-70-231-159-254.dsl.snfc21.sbcglobal.net. Ponieważ jest to globalny adres internetowy, a zawarty w datagramie adres źródłowy IP jest lokalnym adresem jego własnej

Tabela 5.9. *Tabela forwardowania w routerze R1 wskazuje na potrzebę wykonania translacji adresów (NAT). W wyniku tej translacji prywatny, lokalny adres routera 10.0.0.1 odwzorowany zostaje na globalny adres 70.231.132.85, w wyniku czego datagramy generowane w sieci 10.0.0.0/25 widoczne są w Internecie jako wysyłane z adresu 70.231.132.85*

Przeznaczenie (di)	Maska (mi)	Następny przeskok	Interfejs	Komentarz
0.0.0.0	0.0.0.0	70.231.159.254	70.231.132.85	NAT
10.0.0.0	255.255.255.128	10.0.0.100	10.0.0.1	NAT

sieci, router **R1** musi zastąpić go adresem globalnym, wykonując procedurę tłumaczenia adresów sieciowych (*Network Address Translation*, w skrócie NAT) opisywaną szczegółowo w rozdziale 7. W wyniku wykonania tej procedury adres lokalny `10.0.0.100` zastąpiony zostaje przez globalny adres `70.231.132.85`, reprezentujący odtąd „stronę b" routera R1.

Datagram IP po dotarciu do routera **R2** poddawany jest przetwarzaniu podobnemu do tego z **R1**, z wyjątkiem procedury NAT, która jest zbędna, jako że oba adresy IP w nagłówku datagramu są adresami globalnymi. Adres docelowy w nagłówku datagramu nie jest adresem własnym routera **R2**, więc datagram jest forwardowany; tym razem jednak router dysponuje większą liczbą możliwych tras (zamiast jednej trasy domyślnej), zależnie od sposobu połączenia routera z Internetem i lokalnych założeń polityki bezpieczeństwa.

W protokole IPv6 forwardowanie datagramów odbywa się podobnie, z dwiema istotnymi różnicami. Po pierwsze, przeliczanie adresów IP na adresy fizyczne jest zadaniem komunikatów *Neighbor Solicitation*, które (jako część protokołu ICMPv6) opiszemy w rozdziale 8. Po drugie, protokół IPv6 definiuje nową kategorię adresów — adresy lokalne dla łącza, rozpoczynające się od prefiksu `f380::/10` (patrz rozdział 2.); fakt ten może wymagać interwencji użytkownika (administratora) w przypadku hostów *multihomed*, w celu wskazania konkretnego interfejsu, jaki ma być używany do wysyłania datagramów kierowanych na adresy tej grupy.

Poniższy przykład ilustruje wykorzystywanie adresów lokalnych dla łącza w systemie Windows XP; zakładamy, że protokół IPv6 został w tym systemie zainstalowany i działa prawidłowo:

```
C:> ping6 fe80::204:5aff:fe9f:9e80

Badanie fe80::204:5aff:fe9f:9e80 z użyciem 32 bajtów danych:

Brak trasy do miejsca docelowego.
  Określ poprawny identyfikator zakresu lub użyj parametru -s do określenia adre
su źródłowego.

...

Statystyka badania dla fe80::204:5aff:fe9f:9e80:
    Pakiety: Wysłane = 4, Odebrane = 0, Utracone = 4 (100% utraconych),
```

Niepowodzenie wynika z konieczności wskazania konkretnego interfejsu, przeznaczonego do wysyłania datagramów produkowanych przez program `ping6`; wskazanie to

może mieć formę konkretnego adresu IP lub zakresu. Skorzystamy z drugiej możliwości, wskazując jawnie numer interfejsu jako przyrostek `%6` dodany do zasadniczego adresu:

```
C:> ping6 fe80::204:5aff:fe9f:9e80%6

Badanie fe80::204:5aff:fe9f:9e80%2
z fe80::5efe:192.168.131.67%2 z użyciem 32 bajtów danych:

Odpowiedź z fe80::5efe:192.168.131.67%6: bajtów=32 czas=1 ms
Odpowiedź z fe80::5efe:192.168.131.67%6: bajtów=32 czas=1 ms
Odpowiedź z fe80::5efe:192.168.131.67%6: bajtów=32 czas=1 ms
Odpowiedź z fe80::5efe:192.168.131.67%6: bajtów=32 czas=1 ms

Statystyka badania dla fe80::204:5aff:fe9f:9e80%2:
    Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0 (0% straty),
Szacunkowy czas błądzenia pakietów w millisekundach:
    Minimum = 1 ms, Maksimum = 1 ms, Czas średni = 1 ms
```

Trasę datagramu IP prześledzić można za pomocą linuksowego programu `traceroute` (jego windowsowy odpowiednik ma nazwę `tracert` i nieco inny zestaw opcji); użycie parametru `-n` blokuje wyświetlanie nazw DNS przyporządkowanych wyświetlanym adresom:

```
Linux% traceroute -n ftp.uu.net
traceroute to ftp.uu.net (192.48.96.9), 30 hops max, 38 byte packets
 1  70.231.159.254 9.285 ms 8.404 ms 8.887 ms
 2  206.171.134.131 8.412 ms 8.764 ms 8.661 ms
 3  216.102.176.226 8.502 ms 8.995 ms 8.644 ms
 4  151.164.190.185 8.705 ms 8.673 ms 9.014 ms
 5  151.164.92.181 9.149 ms 9.057 ms 9.537 ms
 6  151.164.240.134 9.680 ms 10.389 ms 11.003 ms
 7  151.164.41.10 11.605 ms 37.699 ms 11.374 ms
 8  12.122.79.97 13.449 ms 12.804 ms 13.126 ms
 9  12.122.85.134 15.114 ms 15.020 ms 13.654 ms
     MPLS Label=32307 CoS=5 TTL=1 S=0
10  12.123.12.18 16.011 ms 13.555 ms 13.167 ms
11  192.205.33.198 15.594 ms 15.497 ms 16.093 ms
12  152.63.57.102 15.103 ms 14.769 ms 15.128 ms
13  152.63.34.133 77.501 ms 77.593 ms 76.974 ms
14  152.63.38.1 77.906 ms 78.101 ms 78.398 ms
15  207.18.173.162 81.146 ms 81.281 ms 80.918 ms
16  198.5.240.36 77.988 ms 78.007 ms 77.947 ms
17  198.5.241.101 81.912 ms 82.231 ms 83.115 ms
```

Widzimy listę przeskoków na trasie od komputera wykonującego program `traceroute` do węzła docelowego *ftp.uu.net* (`192.48.96.9`). Program `traceroute` opiera swe działanie na generowaniu kombinacji datagramów UDP z sukcesywnie zwiększaną wartością pola *Czas życia* (w IPv6 *Limit przeskoków*) i komunikatów ICMP (wykorzystywanych do wykrywania sytuacji, gdy w danym węźle wyczerpuje się limit czasu życia (przeskoków) datagramu — w ten właśnie sposób wykrywane jest istnienie węzła na trasie). Dla każdej wartości *Czasu życia*/*Limitu przeskoków* wysyłane są trzy datagramy UDP, w celu trzykrotnego pomiaru czasu wędrówki datagramu od nadawcy do każdego z węzłów.

Standardowo program `traceroute` przetwarza tylko informację związaną z protokołem IP, na listingu widzimy jednak wiersz

```
MPLS Label=32307 CoS=5 TTL=1 S=0
```

Jest on konsekwencją zastosowania techniki MPLS (*MultiProtocol Label Switching* — wieloprotokołowe przełączanie etykiet) opisywanej w dokumencie [RFC3031] i wiążącej z pakietami specjalne etykiety, stanowiące podstawę forwardowania tych pakietów (przez routery obsługujące MPLS). Etykieta sygnalizowana w powyższym wierszu posiada identyfikator 32307, została zakwalifikowana do klasy usług 5, limit przeskoków określono na 1, a komunikat reprezentowany przez etykietę nie znajduje się na dnie stosu MPLS (S=0, patrz [RFC4950]). Współdziałanie MPLS z protokołem ICMP opisane jest w dokumencie [RFC4950], natomiast ich łączenie z pakietami IPv4 (na zasadzie opcji IP) opisano w [RFC6178]. Wielu operatorów sieci wykorzystuje MPLS na potrzeby kontrolowania dróg przepływu poszczególnych części ruchu.

5.4.4. Dyskusja

Przedstawione typowe przykłady pozwalają na sformułowanie kilku spostrzeżeń związanych z forwardowaniem datagramów IP opatrzonych adresami docelowymi typu unicast.

1. Większość hostów i routerów posiada w swych tablicach trasowania pozycję o zerowej masce i zerowym adresie wynikowym, reprezentującą kolejny przeskok na domyślnej trasie pakietu. Istotnie, typowy host lub router znajdujący się na styku Internetu z siecią lokalną kieruje na domyślną trasę wszystkie pakiety nieprzeznaczone dla własnej sieci, posiada bowiem tylko jeden interfejs łączący go z Internetem.
2. Adresy IP źródłowy i docelowy, zawarte w nagłówku datagramu, nie ulegają zmianie na trasie pakietu, z wyjątkiem stosowania opcji *Trasowanie źródłowe* lub procedury NAT. Decyzje dotyczące trasowania datagramu podejmowane są przez hosty i routery na podstawie docelowego adresu IP.
3. Datagramy IP enkapsulowane są w ramkach warstwy łącza danych, w których adresy docelowe (o ile występują) identyfikują zawsze najbliższy przeskok na trasie, zmieniają się więc przy kolejnych przeskokach. W przedstawionych przykładach adresy warstwy łącza danych obecne były w nagłówkach ramek ethernetowych, lecz już nie w ramkach łącza DSL. W protokole IPv4 przeliczanie adresów IP na adresy sprzętowe jest zadaniem protokołu ARP, w protokole IPv6 zadanie to spełniają komunikaty *Neighbor Solicitation* protokołu ICMPv6.

5.5. Mobilny IP

Opisywany dotychczas model forwardowania datagramów IP posiada ważną cechę charakterystyczną: jest nią niezmienność wzajemnej lokalizacji węzłów sieci — każdy host współdzieli ten sam prefiks z najbliższymi hostami i routerami. Zmieniając niespodziewanie adres IP węzła i jednocześnie pozostawiając ów węzeł przyłączony do sieci na poziomie warstwy łącza danych, ryzykujemy załamanie się połączeń nawiązanych w warstwach wyższych (np. połączeń TCP), bo połączenia te wciąż bazują na starym adresie IP wspomnianego węzła i poprawne trasowanie związanych z nimi datagramów może okazać się niemożliwe. Problem ten nabiera poważnego wymiaru użytkowego w przypadku węzłów mobilnych, a wieloletnie (teraz już — kilkudziesięcioletnie) wysiłki projektantów zmierzające do jego rozwiązania zaowocowały skonstruowaniem mechanizmu

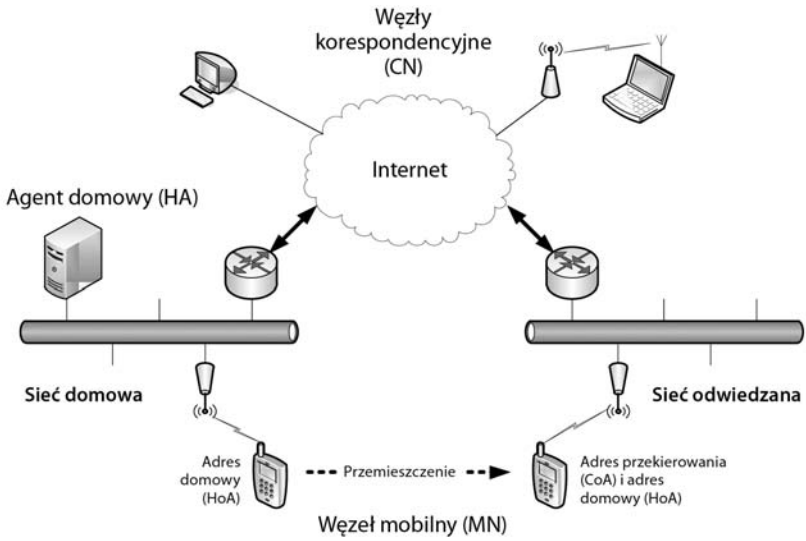
mobilnego IP (*Mobile IP*) (propozycje innych protokołów tego rodzaju przedstawione są w dokumencie [RFC6301]). Mobilny IP istnieje w wersjach dla obu protokołów IPv4 ([RFC5944]) oraz IPv6 ([RFC6275]), skoncentrujemy się jednak wyłącznie na wersji dla IPv6 (oznaczanej dalej skrótowo MIPv6) jako bardziej elastycznej, łatwiejszej do objaśnienia i bardziej rozpowszechnionej, zwłaszcza w nowszych urządzeniach mobilnych, głównie smartfonach. Oczywiście, nie sposób w tak zwięzłych ramach przedstawić wszystkich istotnych jej cech, dlatego ograniczymy się do przedstawienia jedynie podstawowych koncepcji i zasad, polecając zainteresowanym czytelnikom literaturę dedykowaną tematowi, np. publikację [RC05].

Mobilny IP opiera się na koncepcji **sieci domowej** hosta (*home network*) — jest to sieć, w ramach której host przebywa przez większość czasu i którą może opuszczać, przemieszczając się do innych sieci. Komunikacja hosta z jego siecią domową odbywa się zgodnie z opisanymi wcześniej algorytmami. Gdy host opuszcza swą sieć domową, zapamiętuje swój „domowy" adres IP i (przy użyciu różnych trików i specjalnych algorytmów forwardowania) prezentuje się pod tym adresem nowej sieci i wszystkim hostom, z którymi nawiązuje komunikację — tak jakby nadal znajdował się w swej domowej sieci. Podstawowym elementem tego schematu jest specjalny rodzaj routera, zwany **agentem domowym** (*home agent*), wspomagający trasowanie związane z mobilnymi węzłami.

Stopień złożoności MIPv6 uprawnia do uznania go za odrębny protokół. Złożoność ta objawia się głównie w postaci komplikacji mechanizmu komunikatów sygnalizacyjnych i ich zabezpieczania. Komunikaty te wykorzystują różne formy nagłówka rozszerzenia *Mobilność*, identyfikowanego (zgodnie z tabelą 5.5) wartością 135 w polu *Następny nagłówek*. Występuje on w wielu (obecnie 17) odmianach, szczegółowo opisanych na stronie [MP]. Omawiając model komunikacji mobilnego IP, skoncentrujemy się na podstawowych komunikatach opisywanych w [RFC6275]; inne komunikaty wykorzystywane są m.in. do implementowania „szybkich urządzeń podręcznych" (*fast handovers*) ([RFC5568]), do zmiany agenta domowego ([RFC5142]) i do celów eksperymentalnych ([RFC5096]). Rozpoczniemy od przedstawienia ogólnego modelu komunikacji MIPv6 i związanej z nim terminologii.

5.5.1. Model podstawowy — tunelowanie dwukierunkowe

Na rysunku 5.17 widoczne są podstawowe elementy infrastruktury MIPv6; większość terminologii odnosi się również do MIPv4 ([RFC5944]). I tak, host który może się przemieszczać, nazywamy **węzłem mobilnym** (*Mobile Node* — MN), zaś węzły, z którymi może się on komunikować, określane są mianem **węzłów korespondencyjnych** (*Correspondent Nodes* — CN). MN opatrzony zostaje adresem IP utworzonym na bazie prefiksu IP jego sieci domowej — jest to jego **adres domowy** (*Home Address* — HoA). Gdy MN przemieści się do innej sieci, otrzymuje dodatkowy adres, zwany **adresem przekierowania** (*Care-of Address* — CoA). W opisywanym modelu podstawowym komunikacja CN z MN trasowana jest za pomocą **agenta domowego** (*Home Agent* — HA) — to specjalny rodzaj routera, równorzędny w infrastrukturze sieciowej z innymi ważnymi systemami (np. routerami i serwerami WWW). Skojarzenie adresów HoA i CoA węzła mobilnego nazywamy **wiązaniem** (*binding*) tego węzła.



Rysunek 5.17. *Mobilny IP umożliwia utrzymywanie operatywności węzłów, mimo ich przemieszczania się między sieciami. Agent domowy węzła pośredniczy w jego komunikacji z innymi węzłami (w wariancie podstawowym), a w wersji zoptymalizowanej organizuje bezpośrednie skojarzenie węzła mobilnego z węzłem korespondentem*

W podstawowym modelu komunikacji węzły CN nie angażują protokołu MIPv6. Odmianą tego modelu jest sytuacja, gdy mobilna jest całą sieć, wraz ze swym routerem (mechanizm taki, o sugestywnej nazwie „NEMO" — od *Network Mobility* — opisywany jest w dokumencie [RFC3963]). Gdy MN (lub mobilny router sieci) przyłączony zostaje do nowej sieci, otrzymuje CoA i komunikuje go HA za pośrednictwem komunikatu **aktualizacja wiązania** (*binding update*); w odpowiedzi HA przesyła do MN komunikat **potwierdzenie wiązania** (*binding acknowledgment*). Odtąd ruch między MN a jego CN odbywać się będzie za pośrednictwem wspomnianego HA, przy użyciu dwustronnej formy tunelowania pakietów IPv6 (patrz [RFC2473]), zwanej **tunelowaniem dwukierunkowym** (*bidirectional tunneling*). Wymieniane w związku z tym komunikaty chronione są za pomocą protokołu IPsec w wersji z nagłówkiem ESP (*Encapsulating Security Payload* — patrz rozdział 18.). Stanowi to zabezpieczenie przed możliwością nawiązania skojarzenia z HA przez intruzywną stację, podszywającą się pod legalny MN.

5.5.2. Optymalizacja trasy (RO)

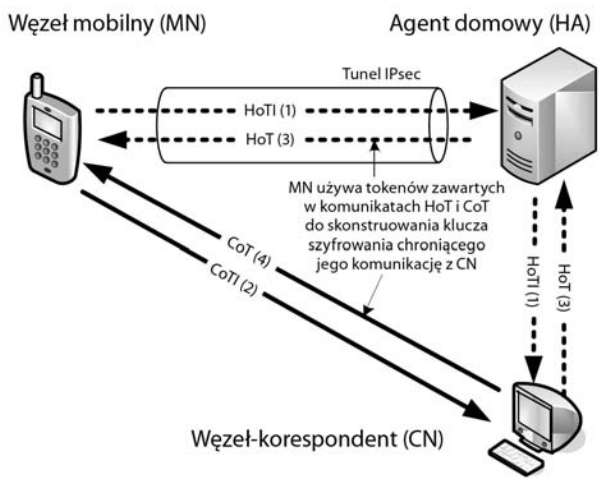
Tunelowanie dwukierunkowe jest stosunkowo nieskomplikowaną techniką realizacji MIPv6 w komunikacji z węzłami CN, które mogą nawet nie implementować MIP, owa prostota może jednak przekładać się na znaczną nieefektywność komunikacji: szczególnie kuriozalna jest sytuacja, gdy MN utrzymuje komunikację z pobliskimi CN, lecz znajduje się w znacznym oddaleniu od swego HA, który *musi* w tej komunikacji pośredniczyć. Wyeliminowanie tego mankamentu jest zadaniem procesu zwanego **optymalizacją trasy** (*route optimization*, w skrócie RO); proces ten musi być wspierany zarówno przez MN, jak i wszystkie CN, z którymi ten się komunikuje. Jak za chwilę pokażemy, metody zapewniające użyteczność i bezpieczeństwo RO są dość skomplikowane, ograniczymy się

więc tylko do naszkicowania podstawowych operacji RO, odsyłając czytelników zainteresowanych szczegółami do lektury dokumentów [RFC6275] i [RFC4866]; racjonalizacja RO — czyli uzasadnienie wyboru takich, a nie innych rozwiązań szczegółowych — zawarta jest w dokumencie [RFC4225].

Podstawową ideą RO jest **rejestracja korespondencyjna**, czyli proces, w ramach którego MN uwierzytelnia się wobec CN, jednocześnie informując je o swym aktualnym adresie CoA — to wszystko w celu nawiązania komunikacji bezpośredniej, nieangażującej HA. Z tego punktu widzenia, funkcjonowanie RO można podzielić na dwa podprocesy: jeden z nich zajmuje się wspomnianą rejestracją korespondencyjną i w efekcie nawiązaniem skojarzenia między MN a CN, drugi natomiast zajmuje się wymianą datagramów między MN a skojarzonymi z nim CN.

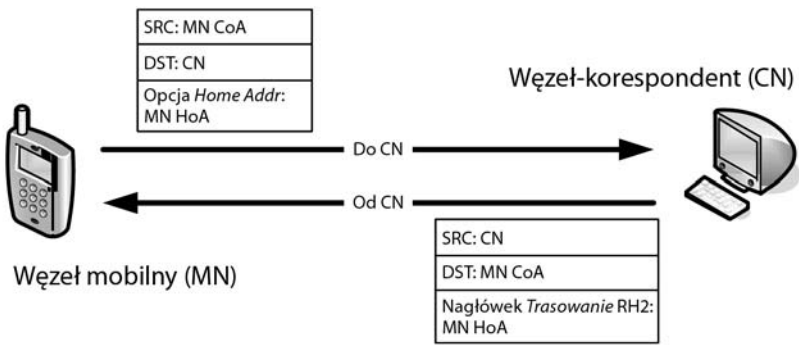
W ramach rejestracji korespondencyjnej MN uwiarygodnia się najpierw wobec każdego CN, wykonując **procedurę trasowalności powrotnej** (*Return Routability Procedure*, w skrócie RRP). Komunikaty wymieniane w ramach RRP nie są chronione przez IPsec (ochrona taka miała miejsce w przypadku komunikacji MN ze swym HA), ponieważ funkcjonowanie IPsec w tym kontekście byłoby (zdaniem projektantów) dość zawodne — co prawda, od każdej implementacji protokołu IPv6 wymaga się zaimplementowania IPsec, jednak nie ma obowiązku jego używania. I choć RRP nie zapewnia ochrony tak solidnej jak IPsec, to jednak jest od niego znacznie prostsza i wychodzi naprzeciw większości problemów bezpieczeństwa, jakie rozważali projektanci MIP.

Procedura RRP obejmuje wymianę następujących *komunikatów mobilności*: `HoTI` (*Home Test Init*), `HoT` (*Home Test*), `CoTI` (*Care-of Test Init*) oraz `CoT` (*Care-of Test*). Ich zadaniem jest weryfikacja przez CN osiągalności MN zarówno na podstawie adresu domowego (komunikaty `HoTI` i `HoT`), jak i adresu CoA (komunikaty `CoTI` i `CoT`). Schemat działania RRP przedstawiono na rysunku 5.18.



Rysunek 5.18. *Wymiana komunikatów w ramach procedury trasowalności powrotnej (RRP) przygotowującej zoptymalizowaną, bezpośrednią komunikację między MN a CN; CN zostaje poinformowany o dostępności MN pod obydwoma adresami: domowym i przekierowania. Linie przerywane reprezentują komunikaty wymieniane za pośrednictwem agenta domowego; cyfry oznaczają kolejność wysyłania komunikatów, choć komunikaty (1) i (2) mogą zostać wysłane przez MN równocześnie*

Na rysunku tym widzimy banalny przypadek jednego MN i jednego CN. Procedurę RRP rozpoczyna MN, wysyłając do CN dwa komunikaty: `HoTI` (za pośrednictwem HA) oraz `CoTI` (bezpośrednio do CN). CN po otrzymaniu obu komunikatów (w dowolnej kolejności) odpowiada wysłaniem do MN dwóch komunikatów: `HoT` (za pośrednictwem wspomnianego HA) oraz `CoT` (bezpośrednio do MN). Komunikaty te niosą losowe ciągi znaków, zwane **tokenami**, służące do skonstruowania klucza kryptograficznego używanego do uwierzytelniania MN wobec CN w ramach procedury ich kojarzenia. Po zakończeniu procedury węzły MN i CN mogą komunikować się w sposób bezpośredni, jak na rysunku 5.19 — trasa wymienianych między nimi datagramów została zoptymalizowana. RO ukazuje swe drugie oblicze, jakim jest metoda bezpośredniego komunikowania się MN i CN, zilustrowana w uproszczeniu na rysunku 5.19.



Rysunek 5.19. *Po nawiązaniu skojarzenia MN i CN wymieniają dane w sposób bezpośredni. Pakiety wędrujące od MN do CN wykorzystują opcję docelową Adres domowy, pakiety wędrujące w kierunku przeciwnym posiadają nagłówek Trasowanie w wersji RH2*

Po pomyślnym przeprowadzeniu kojarzenia dane między MN i CN przepływać mogą w sposób bezpośredni, wolne od nieefektywności właściwej tunelowaniu dwukierunkowemu. Komunikacja ta organizowana jest z użyciem *Opcji docelowych* IPv6 dla datagramów przepływających od MN do CN oraz nagłówka *Trasowanie* RH2 dla datagramów płynących w kierunku przeciwnym; w nagłówku tym adres HoA węzła MN wskazywany jest jawnie jako następny (jedyny) przeskok.

Pakiety wysyłane przez CN zawierają w polu *Adres źródłowy IP* adres CoA węzła MN, co pozwala na uniknięcie problemu polegającego na odrzucaniu pakietów w ramach tzw. **filtrowania wprowadzającego** (*ingress filtering*) (patrz [RFC2827]) — odrzucanie takie mogłoby następować, gdyby w roli adresu źródłowego datagramów wykazywany był adres HoA węzła MN. Notabene adres ten figuruje w pakiecie, lecz znajduje się w polu danych opcji *Home Address* i jako taki nie jest interpretowany (ani modyfikowany) przez routery.

Węzeł CN wysyła więc pakiety przeznaczone dla MN na jego adres CoA (bo taki odczytał z pola *Adres źródłowy IP* w nagłówku podstawowym). Węzeł MN po pomyślnym odebraniu pakietu od CN wykonuje ciekawy trik, polegający na wpisaniu w pole *Adres docelowy IP* tegoż pakietu swego adresu HoA, odczytanego z nagłówka RH2. Tak spreparowany pakiet przekazywany jest pozostałym protokołom stosu TCP/IP implementowanym w węźle MN, wskutek czego protokoły traktują ów pakiet tak, jakby wysłany został na adres HoA, nie CoA, węzła MN.

5.5.3. Dyskusja

Mobilny IP zaprojektowany został z myślą o obsłudze (do pewnego stopnia) łączności mobilnej, w warunkach której adres urządzenia może się dynamicznie zmieniać, lecz zarazem urządzenie to musi pozostać połączone z warstwą łącza danych. Problem ten w mniejszym stopniu dotyczy komputerów przenośnych, które przed przemieszczeniem są zwykle wyłączane, usypiane lub hibernowane[6], jest natomiast typowy dla urządzeń podręcznych (głównie smartfonów), na których uruchamiane są aplikacje czasu rzeczywistego (np. VoIP). Podejmowane są więc różne wysiłki zmierzające do minimalizowania czasu kojarzenia węzłów, odnotować na tym polu należy przede wszystkim technologię **szybkich urządzeń podręcznych** (*fast handovers*) opisywaną w [RFC5568], rozszerzenie MIPv6 nazywane **hierarchicznym MIPv6** (HMIPv6) i opisywane w [RFC5380], a także użycie serwera proxy do odciążenia MN od przetwarzania komunikatów sygnalizacyjnych (technika ta, nazywana po prostu „proxy MIPv6" lub „PMIPv6", opisywana jest w [RFC5213]).

5.6. Przetwarzanie datagramów IP przez host

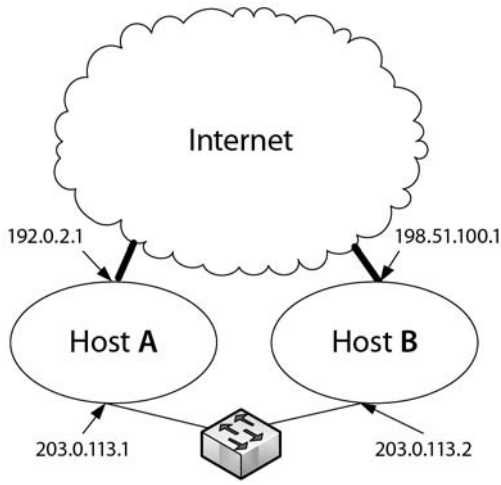
Chociaż routery zasadniczo wolne są od obowiązku wpisywania źródłowego i docelowego adresu IP do przetwarzanych pakietów, host (zarówno ten wysyłający, jak i odbierający pakiety) musi troszczyć się o zawartość tych pól. Przykładowo aplikacja w rodzaju przeglądarki WWW może łączyć się z serwerem posiadającym kilka adresów IP, podobnie komputer kliencki uruchamiający tę przeglądarkę może być także identyfikowany przez wiele adresów IP. Rodzi to m.in. naturalny problem określenia, którego z dostępnych adresów IP (czy nawet — której wersji protokołu IP) należy użyć do wysłania konkretnego datagramu; podobnym problemem jest rozstrzyganie o legalności pakietu, który dotarł do hosta przez „nie ten" interfejs, czyli interfejs nieprzeznaczony do obsługi adresu docelowego figurującego w nagłówku datagramu.

5.6.1. Modele hosta

Gdy do hosta dociera datagram opatrzony adresem docelowym unicast identycznym z którymś z adresów IP tegoż hosta, decyzja o podjęciu przetwarzania odebranego datagramu wydaje się być oczywista; w rzeczywistości jest ona uzależniona od *modelu hosta* przyjętego w systemie odbierającym wspomniany datagram (patrz [RFC1122]) i wbrew pozorom oczywista często nie jest, szczególnie w systemach *multihomed*. Istnieją dwa modele hostów: **silny** (*strong*) i **słaby** (*weak*) — jak można oczekiwać, bardziej restrykcyjny jest model silny. Istotnie, w modelu silnym protokoły stosu TCP/IP otrzymują datagram odebrany przez host tylko wtedy, gdy zawarty w tym datagramie adres docelowy zgodny jest z którymś z adresów IP przypisanych do interfejsu, za pośrednictwem którego ów datagram do hosta dotarł. W modelu słabym kontrola ta jest mniej rygorystyczna, obejmuje jedynie zgodność adresu docelowego datagramu z którymś z lokalnych adresów hosta, kwestia interfejsu dostarczającego datagram jest nieistotna. Analogicznie ma się rzecz z wysyłaniem datagramów: w modelu silnym adres źródłowy w wysyłanym datagramie musi zgadzać się z którymś z adresów przypisanych do interfejsu realizującego wysyłanie, w modelu słabym nie ma takiego wymogu.

[6] Niekoniecznie, coraz więcej komputerów wykorzystuje szerokopasmowe modemy komórkowe — *przyp. tłum.*

Na rysunku 5.20 przedstawiono jedną z typowych sytuacji, w której wybór odpowiedniego modelu hosta staje się kwestią bardzo istotną. Dwa hosty — **A** i **B** — połączone są w dwojaki sposób: za pomocą Internetu oraz za pośrednictwem sieci lokalnej. Jeśli host **A** skonfigurowany jest do pracy według modelu silnego, przychodzące doń z Internetu datagramy o adresie docelowym `203.0.113.1` będą odrzucane, taki sam los spotykać będzie pakiety o adresie docelowym `192.0.2.1` przychodzące do hosta **A** przez sieć lokalną. Jeśli host **B** skonfigurowany jest na pracę według modelu słabego, może zdecydować o wysłaniu datagramu na adres `192.0.2.1` przez interfejs prowadzący do sieci lokalnej, jako zapewniający bardziej efektywną trasę (w porównaniu z trasą przez Internet). W rezultacie pakiet, legalnie wysłany przez host **B** do hosta **A**, zostanie przez ten ostatni odrzucony — ze względu na stosowany model operacyjny. Nie sposób w tym kontekście nie zadać fundamentalnego pytania — pytania o sam sens istnienia modelu silnego.



Rysunek 5.20. *Hosty mogą być połączone w różny sposób, przy użyciu różnych interfejsów, konieczne jest więc dla każdego z hostów właściwe określenie adresów źródłowego i docelowego w wysyłanych datagramach. Wybór wspomnianych adresów dokonywany jest przez host na podstawie jego tablicy forwardowania, algorytmu rankingowego (patrz [RFC3484]) i stosowanego modelu (silnego albo słabego)*

Ów sens da się sprowadzić do wiele mówiącego, magicznego słowa — bezpieczeństwo. Załóżmy, że złośliwy internauta wprowadził do Internetu pakiet przeznaczony dla adresu `203.0.113.2` i zawierający wartość (oczywiście, fikcyjną) `203.0.113.1` w polu adresu źródłowego. Działająca w hoście **B** aplikacja odbiera ów pakiet, wierząc, że utworzony i nadany został przez host **A**. Jeśli w gestii wspomnianej aplikacji spoczywają decyzje w kwestii np. kontroli dostępu, podejmowane na podstawie adresu źródłowego pakietu, fatalne konsekwencje podstępu zastosowanego przez intruza nietrudno sobie wyobrazić. Gdyby host **B** funkcjonował w trybie modelu silnego, nie przyjąłby rzeczonego pakietu na interfejsie prowadzącym do Internetu.

Większość systemów operacyjnych umożliwia jawny wybór modelu hosta; w systemie Windows (w wersji Vista i 7) model silny jest domyślny dla odbierania i wysyłania datagramów, w obu wersjach IPv4 i IPv6. Użytkownik może jednak zmieniać ten domyślny stan, włączając lub wyłączając model słaby za pomocą polecenia o postaci

```
C:\> netsh interface ipv<x> set interface <nazwa> <kierunek> <stan>
```

Parametr <*stan*> określa włączenie (`enabled`) albo wyłączenie (`disabled`) słabego modelu, <*x*> jest numerem wersji protokołu IP (`4` albo `6`), <*nazwa*> jest nazwą interfejsu, <*kierunek*> określa ustawianie słabego modelu dla wysyłania (`weakhostsend`) albo odbierania (`weakhostreceived`) datagramów.

W Linuksie domyślnym modelem jest model słaby, w systemach linii BSD (włącznie z Mac OS X) — model silny.

5.6.2. Selekcja adresów

Nieodłączną czynnością towarzyszącą wysyłaniu przez host datagramu IP jest właściwe ustawienie pól *Adres źródłowy* i *Adres docelowy* w nagłówku tego datagramu. W niektórych sytuacjach adres źródłowy określony jest jednoznacznie bądź to przez aplikację, bądź też przez fakt, że wysyłany datagram jest odpowiedzią na inny, wcześniej odebrany w ramach tego samego połączenia (np. połączenia TCP — w rozdziale 13. zajmiemy się problematyką zarządzania adresami przez protokół TCP).

W nowoczesnych implementacjach protokołu IP wybór wspomnianych adresów jest zadaniem zaawansowanych **procedur selekcyjnych**. Dawno temu, u zarania Internetu, typowy host wykorzystywał tylko jeden adres IP na potrzeby zewnętrznej komunikacji, więc procedury te miały charakter oczywisty. Sytuacja zmieniła się diametralnie w związku z powszechnym wykorzystywaniem wielu adresów IP dla jednego interfejsu, no i — oczywiście — w związku z pojawieniem się IPv6, zgodnie z regułami którego równoczesne wykorzystywanie, dla danego interfejsu, wielu adresów o zróżnicowanym zakresie jest na porządku dziennym. Sytuacja komplikuje się dodatkowo w przypadku hostów implementujących tzw. dualny stos TCP/IP, czyli oba protokoły IPv4 i IPv6 (patrz [RFC4213]); niewłaściwy wybór adresów może prowadzić do różnych niepożądanych efektów ubocznych: asymetrycznego trasowania, nieuzasadnionego filtrowania lub odrzucania pakietów z innych przyczyn. Skuteczne rozwiązanie tych problemów stanowi nie lada wyzwanie dla projektantów.

W przypadku hostów implementujących wyłącznie IPv4 sprawy nie są aż tak skomplikowane. Dla implementacji IPv6 reguły domyślnego wyboru adresów zawarte są w dokumencie [RFC3484], aplikacje powinny mieć możliwość zmiany domyślnych ustawień za pomocą odpowiednich funkcji API (czyhające w związku z tym pułapki są przedmiotem rozważań dokumentu [RFC5220]). Generalnie rzecz biorąc, wspomniane reguły domyślnie zalecają wybór pary adresów „źródłowy-docelowy" o tym samym zakresie, preferowanie węższych zakresów kosztem szerszych dla adresów docelowych, zaś dla adresów źródłowych preferowanie adresów publicznych kosztem tymczasowych; w komunikacji mobilnej adresy domowe preferowane są kosztem adresów przekierowania. Ostatnie słowo w tej kwestii należy do administratora, który ma możliwość zastępowania reguł domyślnych własną polityką selekcyjną — nie będziemy się jednak zajmować tą kwestią, jako specyficzną dla konkretnego środowiska.

Działanie procedury selekcyjnej sterowane jest przez **tablicę założeń**, obecną (przynajmniej koncepcyjnie) w każdym hoście i konfigurowaną na podstawie parametrów określonych przez administratora. Tablica założeń sugerowana przez dokument [RFC3484] — o ile jej zawartość nie stoi w sprzeczności z założeniami administracyjnymi — widoczna jest jako tabela 5.10. Podobnie jak w przypadku tablicy trasowania (patrz punkt 5.4.2),

wybór jednej z pozycji odbywa się na zasadzie najdłuższego pasującego prefiksu. Dla każdego adresu (prefiksu) *x* określone jest pierwszeństwo wyboru P(*x*) — im większa jego wartość, tym większa preferencja w wyborze. Przyporządkowywane poszczególnym pozycjom etykiety L(*x*) służą do grupowania adresów o podobnym typie: jeśli dla pary adresów *x* i *y* spełniony jest warunek L(*x*) = L(*y*), para ta jest preferowana w wyborze do roli adresów „źródłowy-docelowy".

Tabela 5.10. *Domyślna tablica założeń, sugerowana przez [RFC3484]. Większa wartość pierwszeństwa oznacza większą preferencję przy wyborze*

Prefiks	Pierwszeństwo P(x)	Etykieta L(x)
::1/128	50	0
::/0	40	1
2002::/16	30	2
::/96	20	3
::ffff:0:0/96	10	4

Procedura selekcyjna wykorzystuje ponadto następujące własności określone dla danego adresu lub pary adresów.

- CPL(*A*, *B*) to długość (w bitach) wspólnego prefiksu (*common prefix length*), czyli najdłuższego ciągu identycznych bitów lewostronnych, adresów IPv6 *A* i *B*.
- S(*A*) to zakres (*scope*) adresu *A* odwzorowany na wartość numeryczną; większa wartość odpowiada szerszemu zakresowi, jeśli więc *A* jest adresem lokalnym dla łącza, a *B* jest adresem globalnym, to S(*A*) < S(*B*).
- M(*A*) to wynik mapowania adresu IPv4 A na adres IPv6. Ponieważ zakres adresu IPv4 jest funkcją jego wartości, więc konieczne jest honorowanie przez implementację następującej relacji:

```
S(M(169.254.x.x)) = S(M(127.x.x.x)) < S(M(<jakikolwiek adres prywatny IPv4>))
↪< S(M(jakikolwiek inny adres IPv4))
```

- Λ(*A*) jest czasem życia adresu *A*; jeśli *A* jest adresem przestarzałym, niezalecanym do użytku (*deprecated*), a *B* adresem zalecanym (*preferred*), to Λ(*A*) < Λ(*B*),
- W kontekście mobilnego IP: predykat H(*A*) ma wartość `true`, jeśli *A* jest adresem domowym, i `false` w przeciwnym razie, natomiast predykat C(*A*) ma wartość `true`, jeśli *A* jest adresem przekierowania, i `false` w przeciwnym razie.

5.6.2.1. Algorytm selekcji adresu źródłowego

Oznaczmy przez CS(*D*) zbiór *adresów kandydackich*, czyli potencjalnie możliwych adresów IP bazujących na określonym przeznaczeniu *D*. Z definicji do zbioru tego nie należą adresy anycast, multicast i adres nieokreślony. W zbiorze CS(*D*) definiuje się następujące funkcje.

- $R_D(A)$ oznacza *ranking* adresu A w zbiorze CS(*D*): jeśli $R_D(A) > R_D(B)$, to *ranking* adresu *A* jest większy niż ranking adresu *B*, co zapisujemy jako $R_D(A)$ *> $R_D(B)$, a co oznacza, że *A* preferowany jest względem *B* jako adres źródłowy dla osiągnięcia hosta o adresie *D*.

- I(D) oznacza interfejs wybrany dla forwardowania datagramu w kierunku przeznaczenia D, na zasadzie dopasowania względem najdłuższego prefiksu (o czym pisaliśmy w punkcie 5.4.2).
- @(I) jest zbiorem adresów IP przypisanych do interfejsu I.
- T(A) jest predykatem przyjmującym wartość `true`, jeśli A jest adresem tymczasowym (patrz rozdział 6.), i `false` w przeciwnym razie.

W oparciu o powyższe funkcje w zbiorze CS(D) definiowana jest relacja częściowego porządku, w kolejności stosowania następujących reguł dla dwóch dowolnych adresów A i B.

1. Preferowanie tożsamości adresów: jeśli $A = D$ i $B \neq D$, to $R_D(A)$ *> $R_D(B)$, i vice versa: jeśli $A \neq D$ i $B = D$, to $R_D(B)$ *> $R_D(A)$.
2. Preferowanie szerszego zakresu: jeśli $S(A) < S(B)$ i $S(A) < S(D)$, to $R_D(B)$ *> $R_D(A)$, w przeciwnym razie $R_D(A)$ *> $R_D(B)$. I vice versa: jeśli $S(B) < S(A)$ i $S(B) < S(D)$, to $R_D(A)$ *> $R_D(B)$, w przeciwnym razie $R_D(B)$ *> $R_D(A)$.
3. Unikanie adresów przestarzałych — w sytuacji gdy $S(A) = S(B)$: jeśli $\Lambda(A) < \Lambda(B)$, to $R_D(A)$ *> $R_D(B)$, w przeciwnym razie $R_D(A)$ *> $R_D(B)$.
4. Preferowanie adresów domowych ($\wedge$ oznacza koniunkcję, $\neg$ oznacza zaprzeczenie):

   a) jeśli $H(A) \wedge C(A) \wedge \neg (H(B) \wedge C(B))$, to $R_D(A)$ *> $R_D(B)$

   b) jeśli $H(B) \wedge C(B) \wedge \neg (H(A) \wedge C(A))$, to $R_D(B)$ *> $R_D(A)$

   c) jeśli $H(A) \wedge \neg C(A) \wedge \neg H(B) \wedge C(B)$, to $R_D(A)$ *> $R_D(B)$

   d) jeśli $H(B) \wedge \neg C(B) \wedge \neg H(A) \wedge C(A)$, to $R_D(B)$ *> $R_D(A)$
5. Preferowanie interfejsów dla ruchu wychodzącego: jeśli $A \in @(I(D))$ i $B \notin @(I(D))$, to $R_D(A)$ *> $R_D(B)$. I vice versa: jeśli $A \notin @(I(D))$ i $B \in @(I(D))$, to $R_D(B)$ *> $R_D(A)$
6. Preferowanie identyczności etykiet: jeśli $L(A) = L(D)$ i $L(B) \neq L(D)$, to $R_D(A)$ *> $R_D(B)$. I vice versa: jeśli $L(A) \neq L(D)$ i $L(B) = L(D)$, to $R_D(B)$ *> $R_D(A)$
7. Preferowanie adresów trwałych: jeśli $T(A)$ i $\neg T(B)$, to $R_D(B)$ *> $R_D(A)$. I vice versa: jeśli $T(B)$ i $\neg T(A)$, to $R_D(A)$ *> $R_D(B)$.
8. Preferowanie dłuższego wspólnego prefiksu: jeśli $CPL(A, D) > CPL(B, D)$, to $R_D(A)$ *> $R_D(B)$. I vice versa: jeśli $CPL(B, D) > CPL(A, D)$, to $R_D(B)$ *> $R_D(A)$.

Stosowanie powyższych reguł — w wymienionej kolejności — powinno (choć niekoniecznie) doprowadzić ostatecznie do wyboru adresu maksymalnego względem relacji *>, czyli takiego adresu Ψ, że $R_D(\Psi)$ *> $R_D(x)$ dla każdego $x \in CS(D) - \{\Psi\}$. Adres ten tworzy jednoelementowy zbiór oznaczany przez Q(D) i stanowiący podstawę dla algorytmu wyboru adresu docelowego. Jeśli zaprezentowany ciąg reguł nie doprowadza do wybrania adresu o maksymalnej randze, zbiór Q(D) jest zbiorem pustym.

5.6.2.2. Algorytm selekcji adresu docelowego

W podobny sposób przeprowadzana jest selekcja adresu docelowego, identyfikującego w datagramie jego przeznaczenie. Niech Q(x) oznacza rezultat wyboru adresu docelo-

wego dla przeznaczenia *x*, zgodnie z algorytmem opisanym w poprzednim podpunkcie — czyli zbiór jednoelementowy albo zbiór pusty. Na użytek algorytmu wyboru adresu docelowego wprowadzamy kolejne oznaczenia:

- U(*B*) jest predykatem przyjmującym wartość `true`, jeśli adres *B* jest nieosiągalny z poziomu bieżącego hosta;
- E(*B*) jest predykatem przyjmującym wartość `true`, jeśli adres *A* jest osiągalny za pomocą pewnego „enkapsulowanego transportu" (np. tunelowanego trasowania).

Analogicznie do przypadku selekcji adresu źródłowego, poniższy ciąg reguł, definiujących częściowy porządek między adresami kandydackimi tworzącymi zbiór SD(*S*), zdefiniowano z intencją wyłonienia adresu o największym rankingu R_S.

1. Unikanie bezużytecznych adresów: jeśli U(*B*) ma wartość `true` lub Q(*B*) jest zbiorem pustym, to $R_S(A)$ *> $R_S(B)$; analogicznie, jeśli U(*A*) ma wartość `true` lub Q(*A*) jest zbiorem pustym, to $R_S(B)$ *> $R_S(A)$.
2. Preferowanie zgodnych zakresów: jeśli $S(A) = S(Q(A))$ i $S(B) \neq S(Q(B))$, to $R_S(A)$ *> $R_S(B)$; i vice versa — jeśli $S(A) \neq S(Q(A))$ i $S(B) = S(Q(B))$, to $R_S(B)$ *> $R_S(A)$.
3. Unikanie adresów przestarzałych: jeśli $\Lambda(Q(A)) < \Lambda(Q(B))$, to $R_S(B)$ *> $R_S(A)$. I vice versa: jeśli $\Lambda(Q(A)) > \Lambda(Q(B))$, to $R_S(A)$ *> $R_S(B)$.
4. Preferowanie adresów domowych:

   a) jeśli $H(Q(A)) \wedge C(Q(A)) \wedge \neg (C(Q(B)) \wedge H(Q(B)))$, to $R_S(A)$ *> $R_S(B)$

   b) jeśli $H(Q(B)) \wedge C(Q(B)) \wedge \neg (C(Q(A)) \wedge H(Q(B)))$, to $R_S(B)$ *> $R_S(A)$

   c) jeśli $H(Q(A)) \wedge C(Q(A)) \wedge \neg H(Q(B)) \wedge H(Q(B))$, to $R_S(A)$ *> $R_S(B)$

   d) jeśli $H(Q(B)) \wedge C(Q(\boldsymbol{A})) \wedge \neg H(Q(A)) \wedge H(Q(A))$, to $R_S(B)$ *> $R_S(A)$
5. Preferowanie identyczności etykiet: jeśli $L(Q(A)) = L(A)$ i $L(Q(B)) \neq L(B)$, to $R_S(A)$ *> $R_S(B)$. I vice versa: jeśli $L(Q(A)) \neq L(A)$ i $L(Q(B)) = L(B)$, to $R_S(B)$ *> $R_S(A)$.
6. Preferowanie większego pierwszeństwa: jeśli $P(A) > P(B)$, to $R_S(A)$ *> $R_S(B)$. I vice versa: jeśli $P(A) < P(B)$, to $R_S(B)$ *> $R_S(A)$.
7. Preferowanie natywnego transportu: jeśli $E(A) \wedge \neg E(B)$, to $R_S(B)$ *> $R_S(A)$; analogicznie, jeśli $\neg E(A) \wedge E(B)$, to $R_S(A)$ *> $R_S(B)$.
8. Preferowanie węższego zakresu: jeśli $S(A) < S(B)$, to $R_S(A)$ *> $R_S(B)$, w przeciwnym razie $R_S(B)$ *> $R_S(A)$.
9. Preferowanie dłuższego wspólnego prefiksu: jeśli $CPL(A, Q(A)) > CPL(B, Q(B))$, to $R_S(A)$ *> $R_S(B)$. I vice versa: jeśli $CPL(A, Q(A)) < CPL(B, Q(B))$, to $R_S(B)$ *> $R_S(A)$.
10. Jeżeli nie ma zastosowania żadna z powyższych reguł, to o względnym porządku adresów decyduje ich kolejność na oryginalnej liście: jeśli *A* występuje na pozycji wcześniejszej niż *B*, to $R_S(A)$ *> $R_S(B)$, w przeciwnym razie $R_S(B)$ *> $R_S(A)$.

Tak jak przy wyborze adresu źródłowego, tak i tym razem celem powyższych reguł jest wyłonienie kandydata o największym rankingu spośród kandydatur składających się na zbiór SD(*S*) dla danego źródła *S*. W odróżnieniu jednak od wyboru adresu źródłowego,

tym razem przedstawiona procedura zawsze kończy się powodzeniem, gdy bowiem zawiodą wszystkie inne kryteria, ostatecznym kryterium wyboru jest oryginalna pozycja kandydata na liście. Niektóre z operacji przedstawionej procedury kryją pewne pułapki, przykładowo w kroku 9. może wystąpić zjawisko „karuzeli DNS" (*DNS round-robin*) opisywane w rozdziale 11. Pułapki te stały się przesłanką do zrewidowania oryginalnej specyfikacji RFC3484 — poprawka [RFC3484-revise] wprowadza kilka zmian i nowości, m.in. uwzględnia w przedstawionej procedurze rankingowej tzw. unikatowe lokalne adresy IPv6 unicast (*Unique Local IPv6 Unicast Addresses*, w skrócie ULA), opisywane w [RFC4193], rozpoczynające się od prefiksu `fc00::/7`. Są one globalnie rozpoznawalne, lecz ich używanie ograniczone jest do wybranej (prywatnej) sieci lub miejsca sieciowego.

5.7. Ataki wykorzystujące protokół IP

W ciągu swego dość długiego już żywota Internet doświadczył wielu ataków dokonywanych za pośrednictwem protokołu IP; większość z nich sprowadzała się do nadużywania opcji IPv4 lub eksploatacji błędów tkwiących w implementacjach specjalizowanych operacji, m.in. reasemblacji fragmentów datagramu. Niesolidność wielu implementacji protokołu IP we wczesnych routerach powodowała, że bardzo łatwo można było paraliżować działanie tych routerów przez generowanie „bezsensownych" datagramów, np. zawierających kuriozalne wartości w polach *Wersja* czy *IHL*. Na szczęście, współczesne implementacje cechują się wystarczającym stopniem „idiotoodporności", by radzić sobie niezawodnie z próbami podstępów tego rodzaju, poza tym współczesne routery wykorzystywane w Internecie generalnie ignorują obecność opcji IP w datagramie. I choć nie sposób wymagać absolutnej bezbłędności od jakiegokolwiek oprogramowania, wskutek czego w oprogramowaniu routerów także pojawiają się luki zabezpieczeń, to jednak są one sukcesywnie eliminowane, m.in. w dokumentach [RFC1858] i [RFC3128] opisywane są techniki przeciwdziałania atakom wykonywanym przy użyciu mechanizmów reasemblacji datagramów. Twórcy exploitów mają więc coraz bardziej pod przysłowiową górkę.

Wiele nadużyć odbywa się obecnie za pomocą fabrykowania (*spoofing*) adresów IP w sytuacji, gdy datagramy nie są chronione przez szyfrowanie ani uwierzytelnianie. Ponieważ wiele wczesnych mechanizmów kontroli dostępu opierało się na adresach źródłowych IP, które, jak wiadomo, można fabrykować bez większego wysiłku, więc furtka nieuprawnionego dostępu w wielu systemach stała otworem dla intruzów, szczególnie gdy ci łączyli proste „podrabianie" adresów z innymi technikami, np. opcją *Trasowanie źródłowe*. W rezultacie zdalny komputer, próbujący wtargnąć do prywatnej sieci, postrzegany był jako uczestnik tejże sieci (a często nawet utożsamiany był z hostem, którego usługi zamierzał wyłudzać). I chociaż podrabianie adresów IP wciąż jest źródłem potencjalnych zagrożeń, opracowano wiele skutecznych mechanizmów poważnie ograniczających te zagrożenia, m.in. **filtrowanie wprowadzające** (*ingress filtering*), w ramach którego dostawcy Internetu weryfikują adresy źródłowe w datagramach przychodzących od klientów pod kątem zgodności z przydzielonymi tym klientom prefiksami IP.

Ponieważ IPv6 i mobilny IP są technologiami relatywnie młodymi w porównaniu z IPv4, trudno jeszcze mówić o jakimś systematycznym rozpoznaniu ich podatności na ataki wynikające ze specyfiki ich konstrukcji czy implementacji. Trzeba przyznać, że w obli-

czu nowego mechanizmu nagłówków rozszerzeń, znacznie bardziej elastycznego od opcji IPv4, potencjalni intruzi zyskują nowe pole do popisu. Wspominaliśmy już o ataku DoS realizowanym za pomocą oscylacji datagramów wywołanej odpowiednią konstrukcją nagłówka RH0 (patrz punkt 5.3.2), co stało się powodem zarzucenia jego obsługi w nowych implementacjach na rzecz poprawionej wersji RH2. Wersja ta jest jednak nieodporna (podobnie jak jej poprzednik) na podrabianie adresu IP, co stawia szczególne wyzwanie przed konstruktorami firewalli i konfigurującymi je administratorami. Zauważmy, że banalne odrzucanie datagramów zawierających nagłówki rozszerzeń jest pomysłem chybionym, bo uniemożliwiałoby funkcjonowanie mobilnego IP, a poważnie ograniczało wiele innych funkcji protokołu IPv6.

5.8. Podsumowanie

Rozdział ten poświęciliśmy protokołowi IP w dwóch najczęściej używanych wersjach — IPv4 i IPv6. Rozpoczęliśmy od opisu nagłówków w datagramach obu wersji; nagłówki te są całkowicie różne od siebie i muszą być przetwarzane w zupełnie inny sposób, jedynym ich wspólnym elementem jest pole *Wersja*, zajmujące pierwsze 4 bity i służące (zgodnie z nazwą) do rozróżniania obu wersji. Struktura nagłówka IPv6 jest wynikiem optymalizacji podyktowanych spostrzeżeniami nabytymi w związku z wieloletnim stosowaniem wersji IPv4: usunięto z nagłówka opcje, wykorzystywane selektywnie i raczej rzadko, dzięki czemu nagłówek zyskał ustaloną strukturę i ustalony rozmiar. Mimo czterokrotnego wydłużenia adresu IP, rozmiar nagłówka zwiększył się tylko dwukrotnie.

Nagłówek IP (w obu wersjach) zawierał pole charakteryzujące typ ruchu (w IPv6 nazywany klasą usługi) związany z danym datagramem. Wobec nikłej jego użyteczności, przedefiniowano po kilku latach jego znaczenie, dedykując je różnicowaniu usług świadczonych przez Internet, czyli zapewnieniu niektórym rodzajom usług większej wydajności w porównaniu z wariantem standardowym. Stopień wykorzystywania tej możliwości uwarunkowany jest aspektami nie tyle technicznymi, ile odpowiednim modelem biznesowym, obejmującym m.in. rozsądny i sprawiedliwy system opłat za uzyskiwane korzyści.

Forwardowanie IP to proces transportu datagramów przez sieć — prostą lub wieloskokową. Z wyjątkiem przypadków specjalnych, forwardowanie realizowane jest „skok po skoku". Docelowy adres IP w datagramie pozostaje niezmienny na całej jego trasie, zmienia się natomiast na poszczególnych przeskokach docelowy adres warstwy łącza danych w ramkach enkapsulujących datagram. Wybór następnego przeskoku dokonywany jest na bazie tablic trasowania (zwanych także tablicami forwardowania) i procedury dopasowania zgodnie z regułą „najdłuższego pasującego prefiksu". W najprostszym przypadku tablica taka zawiera tylko jedną pozycję, określającą domyślną trasę dla wszystkich forwardowanych pakietów.

Na bazie zestawu specjalnych protokołów sygnalizacyjnych oraz zabezpieczających zaprojektowano i zrealizowano odmianę protokołu IP przeznaczoną dla urządzeń mobilnych, zwaną mobilnym IP. Z mobilnym węzłem skojarzone są dwa adresy: domowy, wywodzący się z jego macierzystej sieci, oraz adres przekierowania, wywodzący się z sieci, w obrębie której węzeł ten aktualnie przebywa. W wersji podstawowej komunikacja węzła mobilnego z innymi węzłami prowadzi zawsze przez agenta domowego, rezydującego w macierzystej sieci tegoż węzła; ze względu na być może znaczne ich oddalenie

komunikacja ta może przybrać wariant bardzo nieefektywny, w związku z czym opracowano (w wersji MIPv6) opcję jej usprawnienia, zwaną popularnie optymalizacją trasy: rola agenta domowego ogranicza się wówczas do ustanowienia skojarzenia węzła mobilnego z węzłem-korespondentem, w ramach którego dalsza komunikacja odbywa się już w sposób bezpośredni.

Kolejnym niebanalnym zagadnieniem związanym z protokołem IP jest sposób przetwarzania datagramów przez hosty, a raczej stopień rygoryzmu kontroli związanej z tym przetwarzaniem. Rygoryzm ten odzwierciedlony jest przez dwa tzw. modele hosta — silny i słaby; ogólnie rzecz biorąc, w warunkach modelu silnego kwestia legalności adresu IP powiązana jest ściśle z interfejsami, przez które wysyłane są i odbierane datagramy, natomiast model słaby jest w tej kwestii znacznie bardziej liberalny, co jednak czyni host mniej odpornym na ewentualne ataki. Ponadto w sytuacji, gdy z danym hostem związanych jest kilka adresów IP (co w IPv6 jest sytuacją normalną, a IPv4 również jest możliwe), powstaje problem odpowiedniej strategii wyboru adresu źródłowego i adresu docelowego zapisywanych w nagłówku datagramu. Strategia ta była oczywista w czasach, gdy host łączony był z Internetem za pośrednictwem pojedynczego interfejsu i opatrzony pojedynczym adresem IP, dziś w obliczu hostów *multihomed* (czyli hostów połączonych z kilkoma dostawcami za pośrednictwem różnych interfejsów) odpowiednie albo kiepskie zaprojektowanie tej strategii przekłada się w konsekwencji na efektywne albo kiepskie trasowanie. Standardowy ciąg reguł składający się na domyślną postać tej strategii preferuje adresy trwałe, o ograniczonym zakresie, kosztem adresów ogólnych i tymczasowych.

Na koniec zajęliśmy się problematyką ataków internetowych, możliwych do realizacji za pośrednictwem mechanizmów protokołu IP. Większość z tych ataków sprowadza się do podrabiania lub fałszowania adresów IP w nagłówkach i opcjach, z fatalną często konsekwencją dla poprawności trasowania datagramów; intruzi mogą też wykorzystywać („eksploatować") rozmaite błędy i luki tkwiące w konkretnych implementacjach protokołu. Obecnie większość routerów ignoruje opcje specyfikowane w datagramach (a routery brzegowe często usuwają je z datagramów na styku z Internetem). Chociaż podrabianie adresów wciąż jest potencjalnym źródłem zagrożeń, jego konsekwencje są w dużej części niwelowane przez rozmaite mechanizmy filtrowania, m.in. filtrowanie wprowadzające.

5.9. Bibliografia

[AN] *http://www.iana.org/assignments/protocol-numbers*

[AUTOVON] http://en.wikipedia.org/wiki/Autovon

[DC05] J. Doyle, J. Carroll, *Routing TCP/IP, Volume 1, Second Edition* (Cisco Press, 2005).

[DSCPREG] *http://www.iana.org/assignments/dscp-registry/dscp-registry.xml*

[H05] G. Huston, *Just How Big Is IPv6? — or Where Did All Those Addresses Go?0*, „The ISP Column", lipiec 2005, *http://www.potaroo.net/papers/isoc/2005-07/ipv6size.html*

[IP6PARAM] *http://www.iana.org/assignments/ipv6-parameters*

[IPPARAM] *http://www.iana.org/assignments/ip-parameters*

[IV] *http://www.iana.org/assignments/version-numbers*

[LFS07] J. Leguay, T. Friedman, K. Salamatian, *Describing and Simulating Internet Routes*, „Computer Networks", 51(8), czerwiec 2007.

[MB97] L. McKnight, J. Bailey(red.), *Internet Economics* (MIT Press, 1997).

[MP] *http://www.iana.org/assignments/mobility-parameters*

[P90] C. Pinter, *A Book of Abstract Algebra, Second Edition* (Dover, 2010; reprint wydania z 1990 roku).

[PB61] W. Peterson, D. Brown, *Cyclic Codes for Error Detection*, Proc. IRE, 49(228), styczeń 1961.

[RC05] S. Raab, M. Chandra, *Mobile IP Technology and Applications* (Cisco Press, 2005).

[RFC0791] J. Postel, *Internet Protocol*, Internet RFC 0791/STD 0005, wrzesień 1981.

[RFC1108] S. Kent, *U.S. Department of Defense Security Options for the Internet Protocol*, Internet RFC 1108 (historical), listopad 1991.

[RFC1122] R. Braden (red.), *Requirements for Internet Hosts — Communication Layers*, Internet RFC 1122/STD 0003, październik 1989.

[RFC1385] Z. Wang, *EIP: The Extended Internet Protocol*, Internet RFC 1385 (informational), listopad 1992.

[RFC1393] G. Malkin, *Traceroute Using an IP Option*, Internet RFC 1393 (experimental), styczeń 1993.

[RFC1858] G. Ziemba, D. Reed, P. Traina, *Security Consideration for IP Fragment Filtering*, Internet RFC 1858 (informational), październik 1995.

[RFC2113] D. Katz, *IP Router Alert Option*, Internet RFC 2113, luty 1997.

[RFC2460] S. Deering, R. Hinden, *Internet Protocol, Version 6 (IPv6)*, Internet RFC 2460, grudzień 1998.

[RFC2473] A. Conta, S. Deering, *Generic Packet Tunneling in IPv6 Specification*, Internet RFC 2473, grudzień 1998.

[RFC2474] K. Nichols, S. Blake, F. Baker, D. Black, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, Internet RFC 2474, grudzień 1998.

[RFC2475] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, *An Architecture for Differentiated Services*, Internet RFC 2475 (informational), grudzień 1998.

[RFC2597] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, *Assured Forwarding PHB Group*, Internet RFC 2597, czerwiec 1999.

[RFC2675] D. Borman, S. Deering, R. Hinden, *IPv6 Jumbograms*, Internet RFC 2675, sierpień 1999.

[RFC2711] C. Partridge, A. Jackson, *IPv6 Router Alert Option*, Internet RFC 2711, październik 1999.

[RFC2827] P. Ferguson, D. Senie, *Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing*, Internet RFC 2827/BCP 0038, maj 2000.

[RFC3031] E. Rosen, A. Viswanathan, R. Callon, *Multiprotocol Label Switching Architecture*, Internet RFC 3031, styczeń 2001.

[RFC3128] I. Miller, *Protection Against a Variant of the Tiny Fragment Attack*, Internet RFC 3128 (informational), czerwiec 2001.

[RFC3168] K. Ramakrishnan, S. Floyd, D. Black, *The Addition of Explicit Congestion Notification (ECN) to IP*, Internet RFC 3168, wrzesień 2001.

[RFC3246] B. Davie, A. Charny, J. C. R. Bennett, K. Benson, J. Y. Le Boudec, W.Courtney, S. Davari, V. Firoiu, D. Stiliadis, *An Expedited Forwarding PHB (Per-Hop Behavior)*, Internet RFC 3246, marzec 2002.

[RFC3260] D. Grossman, *New Terminology and Clarifications for Diffserv*, Internet RFC 3260 (informational), kwiecień 2002.

[RFC3484] R. Draves, *Default Address Selection for Internet Protocol Version 6 (IPv6)*", Internet RFC 3484, luty 2003.

[RFC3484-revise] A. Matsumoto, J. Kato, T. Fujisaki, T. Chown, *Update to RFC 3484 Default Address Selection for IPv6*, Internet draft-ietf-6man-rfc3484-revise (w przygotowaniu), lipiec 2011. *http://tools.ietf.org/agenda/79/slides/6man-6.pdf*

[RFC3704] F. Baker, P. Savola, *Ingress Filtering for Multihomed Hosts*, Internet RFC 3704/BCP 0084, maj 2004.

[RFC3963] V. Devarapalli, R. Wakikawa, A. Petrescu, P. Thubert, *Network Mobility (NEMO) Basic Support Protocol*, Internet RFC 3963, styczeń 2005.

[RFC4193] R. Hinden, B. Haberman, *Unique Local IPv6 Unicast Addresses*, Internet RFC 4193, październik 2005.

[RFC4213] E. Nordmark, R. Gilligan, *Basic Transition Mechanisms for IPv6 Hosts and Routers*, Internet RFC 4213, październik 2005.

[RFC4225] P. Nikander, J. Arkko, T. Aura, G. Montenegro, E. Nordmark, *Mobile IP Version 6 Route Optimization Security Design Background*, Internet RFC 4225 (informational), grudzień 2005.

[RFC4594] J. Babiarz, K. Chan, F. Baker, *Configuration Guidelines for Diffserv Service Classes*, Internet RFC 4594 (informational), sierpień 2006.

[RFC4782] S. Floyd, M. Allman, A. Jain, P. Sarolahti, *Quick-Start for TCP and IP*, Internet RFC 4782 (experimental), styczeń 2007.

[RFC4866] J. Arkko, C. Vogt, W. Haddad, *Enhanced Route Optimization for Mobile IPv6*, Internet RFC 4866, maj 2007.

[RFC4950] R. Bonica, D. Gan, D. Tappan, C. Pignataro, *ICMP Extensions for Multiprotocol Label Switching*, Internet RFC 4950, sierpień 2007.

[RFC5095] J. Abley, P. Savola, G. Neville-Neil, *Deprecation of Type 0 Routing Headers in IPv6*, Internet RFC 5095, grudzień 2007.

[RFC5096] V. Devarapalli, *Mobile IPv6 Experimental Messages*, Internet RFC 5094, grudzień 2007.

[RFC5142] B. Haley, V. Devarapalli, H. Deng, J. Kempf, *Mobility Header Home Agent Switch Message*, Internet RFC 5142, styczeń 2008.

[RFC5213] S. Gundavelli (red.), K. Leung, V. Devarapalli, K. Chowdhury, B.Patil, *Proxy Mobile IPv6*, Internet RFC 5213, sierpień 2008.

[RFC5220] A. Matsumoto, T. Fujisaki, R. Hiromi, K. Kanayama, *Problem Statement for Default Address Selection in Multi-Prefix Environments: Operational Issues of RFC 3484 Default Rules*, Internet RFC 5220 (informational), lipiec 2008.

[RFC5350] J. Manner, A. McDonald, *IANA Considerations for the IPv4 and IPv6 Router Alert Options*, Internet RFC 5350, wrzesień 2008.

[RFC5380] H. Soliman, C. Castelluccia, K. ElMalki, L. Bellier, *Hierarchical Mobile IPv6 (HMIPv6) Mobility Management*, Internet RFC 5380, październik 2008.

[RFC5568] R. Koodli (red.), *Mobile IPv6 Fast Handovers*, Internet RFC 5568, lipiec 2009.

[RFC5570] M. StJohns, R. Atkinson, G. Thomas, *Common Architecture Label IPv6 Security Option (CALIPSO)*, Internet RFC 5570 (informational), lipiec 2009.

[RFC5865] F. Baker, J. Polk, M. Dolly, *A Differentiated Services Code Point (DSCP) for Capacity-Admitted Traffic*, Internet RFC 5865, maj 2010.

[RFC5944] C. Perkins (red.), *IP Mobility Support for IPv4, Revised*, Internet RFC 5944, listopad 2010.

[RFC6178] D. Smith, J. Mullooly, W. Jaeger, T. Scholl, *Label Edge Router Forwarding of IPv4 Option Packets*, Internet RFC 6178, marzec 2011.

[RFC6275] C. Perkins (red.), D. Johnson, J. Arkko, *Mobility Support in IPv6*, Internet RFC 6275, czerwiec 2011.

[RFC6301] Z. Zhu, R. Rakikawa, L. Zhang, *A Survey of Mobility Support in the Internet*, Internet RFC 6301 (informational), lipiec 2011.

[RTAOPTS] *http://www.iana.org/assignments/ipv6-routeralert-values*

[THL06] N. Thompson, G. He, H. Luo, *Flow Scheduling for End-Host Multihoming*, Proc. IEEE INFOCOM, kwiecień 2006.

[TWEF03] J. Touch, Y. Wang, L. Eggert, G. Flinn, *A Virtual Internet Architecture*, Proc. ACM SIGCOMM Future Directions in Network Architecture Workshop, marzec 2003.

[W03] T. Wu, *Network Neutrality, Broadband Discrimination*, „Journal of Telecommunications and High Technology Law", 2, 2003 (zrewidowane w 2005).

Skorowidz

D

E

F

G

H

I

J

K

L

Ł

M

N

P

Q

R

S

Ś

T

U

V

W

Z

Ż

Wydanie II

TCP/IP od środka Protokoły

Vademecum profesjonalisty

Kompletne źródło informacji na temat możliwości TCP/IP!

TCP/IP to model, bez którego nie byłoby sieci Internet – takiej, jaką dziś znamy. Pomimo słusznego wieku (pierwsze próby odbywały się w latach 70.) jest nadal w pełni wystarczający. Główne założenie modelu TCP/IP to podział całego procesu komunikacji na współpracujące ze sobą warstwy. Na tej podstawie zbudowane są różne protokoły transmisji danych, takie jak FTP, HTTP czy SMTP.

TCP/IP od środka. Protokoły. Wydanie II to szczegółowy, opatrzony wieloma ilustracjami przewodnik po współczesnych protokołach grupy TCP/IP. Uwzględnia najnowsze wersje tych protokołów i pokazuje ich funkcjonowanie „na żywo", w środowisku popularnych systemów operacyjnych, takich jak Windows, Linux i Mac OS X. Nie ma lepszego sposobu na wyjaśnienie, dlaczego właśnie tak wyglądają poszczególne aspekty działania TCP/IP, jak zmienia się ono w różnych okolicznościach oraz jak wykorzystać jego różne możliwości. To wyjątkowe opracowanie stanowi obowiązkową lekturę dla wszystkich osób chcących dowiedzieć się więcej o podwalinach współczesnej sieci. W trakcie lektury poznasz założenia architektoniczne, architekturę adresów internetowych oraz znaczenie i rolę poszczególnych warstw modelu TCP/IP. Dowiesz się, jak korzystać z komunikatów ICMP, rozgłaszać informacje w sieci, kontrolować przeciążenia w protokole TCP oraz korzystać z mechanizmów kryptograficznych. Znajdziesz tu dogłębne i intuicyjne wyjaśnienie wielu meandrów TCP/IP i internetu, co pozwoli Ci bardziej efektywnie zarządzać swoimi sieciami i tworzyć lepsze aplikacje internetowe.

Kevin R. Fall zajmuje się protokołami TCP/IP od ponad ćwierćwiecza. Jest członkiem organizacji Internet Architecture Board oraz współzarządzającym grupy roboczej IETF Delay Tolerant Networking Research (DTNRG), zajmującej się problematyką wydajnego funkcjonowania sieci w warunkach ekstremalnych. Należy również do IEEE.

W. Richard Stevens był jednym z tych pionierskich autorów, na których książkach wychowało się całe pokolenie specjalistów od sieci TCP/IP, sukcesywnie sprowadzających internet z wyżyn akademickich katedr do codziennego życia każdego człowieka. Wśród bestsellerów jego autorstwa można wymienić wszystkie trzy tomy *TCP/IP Illustrated* (Addison-Wesley) oraz *UNIX Network Programming* (Prentice Hall).

W tym znakomitym podręczniku znajdziesz informacje na temat:

- ***modelu TCP/IP***
- ***bezprzewodowych sieci LAN***
- ***architektury adresów internetowych***
- ***protokołu PPP***
- ***możliwości autokonfiguracji z wykorzystaniem DHCP***
- ***datagramów użytkownika — UDP***

helion.pl
księgarnia internetowa

Nr katalogowy: 11709

Księgarnia internetowa:
http://helion.pl

Zamówienia telefoniczne:
0 801 339900
0 601 339900

Helion

Sprawdź najnowsze promocje:
http://helion.pl/promocje
Książki najchętniej czytane:
http://helion.pl/bestsellery
Zamów informacje o nowościach:
http://helion.pl/nowosci

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
http://helion.pl

sięgnij po WIĘCEJ
KOD KORZYŚCI

ISBN 978-83-246-4815-3

9 788324 648153

Cena 129,00 zł

Informatyka w najlepszym wydaniu