# *Protokół IP wersja 6*

***Piotr Jankowski***

*Opisano budowę protokołu IPv6, a zwłaszcza nagłówek IPv6 i nagłówki rozszerzające. Ponadto wskazano sposób podziału adresów IP na podsieci.*

***Internet, protokół IPv6, nagłówki, adresy IP***

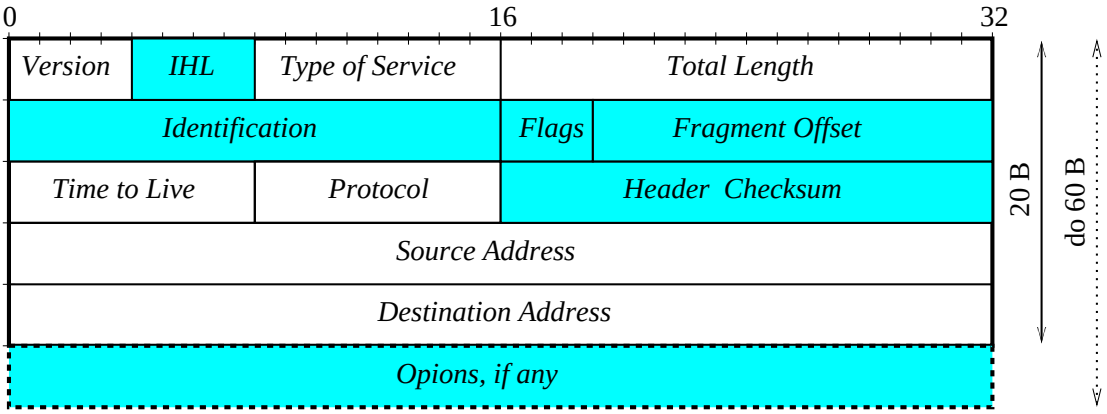
## Wprowadzenie

We wrześniu 2001 roku minęło 10 lat istnienia Internetu w Polsce. Przez te lata jego rola ciągle zmieniała się. W 1991 r. Internet był postrzegany głównie jako pomoc naukowa dla środowisk akademickich. Natomiast obecnie jest on bardzo popularną platformą komunikacyjną, służącą do wymiany różnych typów danych, w tym także danych multimedialnych. Korzystają z niego codziennie zarówno biznesmeni czy naukowcy, jak i dzieci. Dla wielu ludzi jest on taką samą codziennością, jak telefon czy faks. Podstawą Internetu jest protokół IPv4 (*Internet Protocol*), a zatem jego rozwój jest ściśle związany z możliwościami tego protokołu. Niestety protokół IPv4 nie był opracowywany z myślą o Internecie i ma wiele różnorodnych ograniczeń, które mogą zahamować rozwój Internetu. Najistotniejszą wadą jest liczba dostępnych adresów IP. Na szczęście dość szybko zdano sobie sprawę z tych ograniczeń i opracowano następną generację protokołu dla Internetu, czyli protokół IPv6.

## Protokół IP wersja 4

Pierwotne założenia protokołu IP opracowano na zamówienie Departamentu Obrony USA. Na ich podstawie zbudowano sieć ARPANET. Pierwsza specyfikacja RFC 791 pojawiła się w 1981 roku. W protokole zdefiniowano nagłówek, który ma od 40 do 60 B (rys. 1). Zawiera on wiele różnych pól, wśród których najdłuższe są pola określające adresy IP. Protokół znalazł zastosowanie początkowo w sieciach wojskowych, a następnie w sieciach akademickich, stanowiących zaczątek Internetu. Obecnie protokół IP wersja 4, z unikalnym w skali świata 32-bitowym adresem, jest podstawą Internetu. Aby zapewnić unikalność adresów IP, ich przydziałem na świecie zajmuje się IANA (*Internet Assigned Numbers Authority*). Wysyła ona przydzielone adresy bezpośrednio do regionalnych centrów, a dalej przydziałem adresów dla Europy zajmuje się RIPE (*Réseaux IP Européens*). Z roku na rok liczba przydzielonych adresów IP rośnie i pula dostępnych adresów IP maleje.

Adresy IPv4 zapisuje się w postaci *A.B.C.D*, gdzie A, B, C, D są liczbami z zakresu $0 \div 255$. Liczba ta odpowiada wartości jednego bajtu w adresie, a więc teoretycznie umożliwia zaadresowanie $2^{32} = 4\,294\,967\,296$ urządzeń. Jest to dużo, lecz sposób adresowania przyczynił się do znacznego ograniczenia liczby dostępnych adresów IP. W adresacji IP przewidziano istnienie klas adresowych. Miało to ułatwić trasowanie pakietów, ale stało się zmorą Internetu, gdyż powodowało powstanie wielu grup adresów, które były przydzielone i nie używane, a jednocześnie przekazanie ich komuś kto ich potrzebował, było niemożliwe. Zdefiniowano trzy klasy adresowe A, B, C. W klasie A mogło




***Rys. 1.*** *Nagłówek IPv4 (pola z nagłówka zaznaczone ciemniejszym kolorem nie mają swoich odpowiedników w nagłówku IPv6)*

być teoretycznie 127 sieci po 17 777 216 adresów, w klasie B 16 384 sieci po 65 536 adresów, a w klasie C 2 097 152 sieci po 255 adresów. W praktyce najlepiej sprawdzają się sieci klasy C, gdyż większość sieci LAN dołączonych do Internetu nie przekracza rozmiaru takiej sieci. Gdyby adresowanie klasowe obowiązywało w dalszym ciągu w Internecie, to możliwości adresowania zakończyłyby się po przyłączeniu 2 113 663 sieci. Na szczęście ten problem rozwiązano wprowadzając ruting bezklasowy, co rozszerzyło możliwości adresowania sieci, ale jednocześnie skomplikowało tablice rutingu w Internecie. Obecnie ruter pracujący w szkielecie Internetu musi mieć co najmniej 256 MB pamięci operacyjnej. Wzrost liczby użytkowników Internetu powoduje wyczerpywanie się dostępnej puli adresów. Szacuje się, że obecne adresy powinny się skończyć około 2003 roku. Termin jest jednak stale przesuwany, gdyż coraz powszechniej stosuje się techniki NAT, umożliwiające przyłączenie dużej organizacji do sieci Internet za pomocą kilku adresów IP. Jednocześnie organizacje rozdzielające adresy IP dokładniej przeglądają wnioski. Bardziej groźną dla protokołu IPv4 jest integracja usług. Protokół nie jest na to przygotowany. Nagłówek IPv4 ma zmienną długość[①], a ponadto jest w nim wyliczana suma kontrolna. Obniża to efektywność trasowania pakietów. Biorąc pod uwagę doświadczenia płynące ze stosowania protokołu IPv4 oraz konieczność jego wymiany (ze względu na kurczenie się puli dostępnych adresów IP), postanowiono zatem opracować nowy protokół dla Internetu.

# Protokół IP wersja 6

## *Założenia projektowe*

Opracowując nowy protokół dla Internetu, wykorzystano doświadczenia z użytkowania wersji 4 protokołu IP. Sformułowano kryteria projektowe[②] dla nowej generacji protokołu, zwanej IPng (*Internet Protocol Next Generation*). Warto wymienić najważniejsze kryteria, czyli:

– skalowalność (możliwość adresacji do $10^{12}$ systemów końcowych oraz $10^9$ sieci);
– niezależność topologiczną (możliwość budowy sieci z zastosowaniem różnych topologii);

① *Zazwyczaj nagłówek ma stałą długość 20 bajtów – opcje są rzadko stosowane.*
② *Opisane w dokumencie RFC1726: „Technical Criteria for Choosing the Next Generation”.*

– elastyczną migrację z wersji IPv4 na nową wersję;
– niezależność od mediów transmisyjnych;
– wsparcie dla autokonfiguracji hostów i ruterów;
– wsparcie dla autentykacji i zabezpieczeń;
– wsparcie dla mobilności zarówno pojedynczych hostów, jak i całych sieci;
– wsparcie dla nowych usług (takich, jak transmisja głosu, wideo itp.).

Spełnienie wszystkich tych kryteriów okazało się niemożliwe. Opracowano jednak rekomendację RFC 1752 (znaną jako IPv6), łączącą różne cechy z kilku wybranych propozycji.

## *Budowa nagłówka protokołu IPv6*

Projektanci nagłówka protokołu IPv6 wzięli pod uwagę doświadczenia z użytkowania dotychczasowego protokołu. Musieli zmieścić w nagłówku 128-bitowe pole adresowe, sądząc, że wówczas nagłówek będzie maksymalnie krótki, co przyspieszy jego przetwarzanie. Z porównania nagłówka IPv6 z nagłówkiem protokołu IPv4 wynika, że poczyniono dwie zasadnicze zmiany:

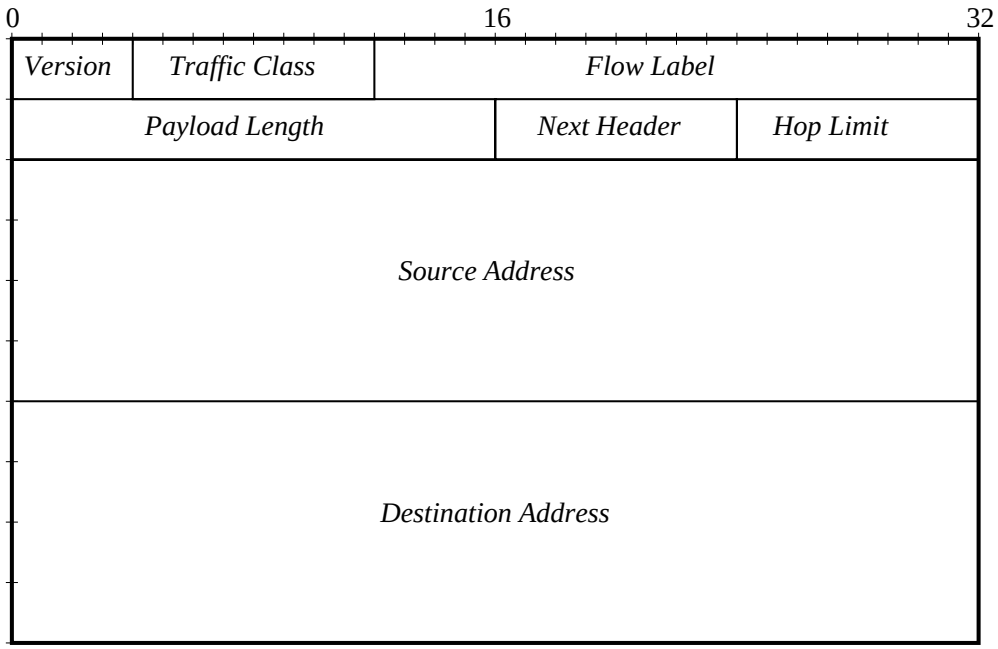
– usunięto część pól,
– ustalono stałą długość nagłówka.

Te dwa posunięcia powodują, że – pomimo wydłużenia adresów z 32 do 128 bitów – nagłówek IPv6 ma tylko 40 bajtów. Nagłówek IPv4 może mieć od 20 do 60 B. Utrudnia to jego przetwarzanie przez rutery pracujące w Internecie, muszą one bowiem, oprócz wyznaczania trasy dla pakietu, analizować cały nagłówek pod kątem zawartości różnych opcjonalnych pól. Usunięto także wszystkie inne pola, które powodowały przedłużenie przetwarzania przez ruter. W nagłówku IPv6 jest tylko 8 pól (rys. 2): *Version, Traffic Class, Flow Label, Payload Length, Next Header, Hop Limit*, *Source Address* oraz *Destination Address*. Zredukowanie liczby pól i ustawienie stałej długości nagłówka umożliwia szybsze przetwarzanie pakietów przez rutery, które nie muszą np. wyliczać sumy kontrolnej nagłówka. Jest to operacja złożona matematycznie i zajmuje niepotrzebnie czas. Kontrolę błędów powierzono innym warstwom stosu protokołów. Zbędne się stały także pola związane z długością nagłówka IPv4: *Header Length, Fragment Offset, Identification, Flags.*

W stosunku do protokołu IPv4 całkowicie przeorganizowano fragmentację pakietów. Zlikwidowano pola dotyczące fragmentacji pakietów w nagłówku IPv6 i przeniesiono ich funkcję do specjalnego nagłówka rozszerzającego. Rutery transmitujące pakiety IPv4 muszą każdorazowo badać rozmiar pakietu i porównywać jego rozmiar z MTU (*Maximum Transfer Unit*), obowiązującym na danym łączu oraz w razie konieczności dokonywać jego defragmentacji. Tego typu podejście obniża wydajność urządzeń, a zatem i sieci. W protokole IPv6 przed wysłaniem pakietu stacja nadawcza musi określić MTU dla całej trasy pakietu za pomocą specjalnego protokołu *MTU Path Discovery*.

Część z pól nagłówka IPv6 ma swoje odpowiedniki w nagłówku IPv4. Są to oczywiście 128-bitowe pola *Source Address* oraz *Destination Address*, a także pole *Version*. W polu *Version* na stałe jest wpisana binarnie wartość 6 (0110), określająca protokół IPv6. Umożliwia to urządzeniom odróżnienie pakietów IPv4 od pakietów IPv6. Swój odpowiednik ma również pole *Hop Limit (Time to Live* w IP4). Jest ono używane do przerywania pętli w rutingu, przez zmniejszanie maksymalnej wartości przeskoku (*Hop Value*) o jeden dla każdego węzła na trasie pakietu. Wartość pola ustawia węzeł źródłowy. Gdy w którymś węźle wartość pola osiągnie zero, pakiet jest odrzucany. Pole to ma 8 bitów długości, co ogranicza rozmiar sieci do 255 przejść, ale wbrew pozorom jest to bardzo dużo i powinno

wystarczyć na potrzeby Internetu opartego na nowym protokole. Pole z nagłówka IPv4 *Type of Service* zostało zastąpione polem *Traffic Class* o tej samej długości. Umożliwia ono definiowanie usług typu *Quality of Service* (QoS). Rozszerzenie możliwości świadczenia usług z określoną jakością daje 20-bitowe pole *Flow Label*, lecz na razie nie ma szczegółowych ustaleń co do wykorzystania tego pola. Pole *Payload Length* wyraża długość pakietu w oktetach. Dopuszczalne są wartości z zakresu od 576 do 65 535 B. W razie konieczności jest możliwe ustawienie wartości 0 w tym polu i przesłanie dłuższego pakietu, zwanego *Jumbodatagramem*.




***Rys. 2.*** *Nagłówek IPv6*

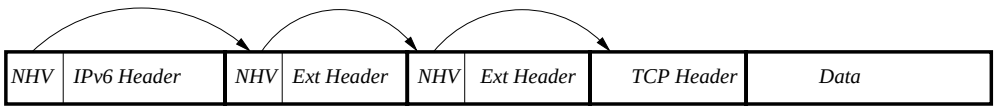
## *Rozszerzenia nagłówka*

W nagłówku IPv6 znajduje się pole *Next Header*, z pomocą którego można określić, jaki nagłówek rozszerzający znajduje się za głównym nagłówkiem IP (a przed nagłówkiem transportowym). Nagłówki te tworzą ściśle określone moduły, co umożliwia ich szybkie przetwarzanie. Nie ma ograniczenia ich liczby, istotna jest jednak kolejność ich występowania. Nagłówki te zapewniają między innymi możliwość fragmentacji i autentykacji pakietów. Podobne możliwości miały opcjonalne pola w nagłówku IPv4, lecz nie były zbyt szeroko stosowane, gdyż obniżały wydajność przetwarzania pakietów. Koncepcja rozszerzonych nagłówków umożliwia łatwe rozbudowanie protokołu IPv6, bez konieczności zmian w głównym nagłówku. Dotychczas zdefiniowano następujące nagłówki:

– *Hop-by-Hop Options Header*,
– *Destination Options Header-1*,
– *Source Routing Header*,
– *Fragment Header*,

– *Authentication Header*,
– *IPv6 Encryption Header*,
– *Destination Option Header-2*.

Obecność rozszerzonego nagłówka jest sygnalizowana przez odpowiednią wartość pola *Next Header Value* nagłówka IPv6. Każdy z nagłówków rozszerzonych ma własne ośmiobitowe pole *Next Header Value*. W tych polach powinny się znaleźć również wartości określające typ następnego nagłówka. W ostatnim nagłówku rozszerzonym znajduje się wartość pola, określająca typ nagłówka warstwy transportowej, np. nagłówek TCP (rys. 3).




***Rys. 3.*** *Rozszerzone nagłówki*

Zdefiniowane obecnie wartości pola *Next Header Value* zawiera tablica 1. Jak widać, oprócz wartości kodów odpowiadających nagłówkom rozszerzonym są także kody odpowiadające nagłówkom TCP, IP, ICMP. Ośmiobitowe pole *Next Header Value* umożliwia definiowanie w razie potrzeby kolejnych typów nagłówków (bez modyfikacji protokołu). Obecna implementacja protokołu IPv6 zakłada istnienie

***Tabl. 1. Kody nagłówków***

| Wartość | Typ nagłówka |
|---|---|
| 0 | *Hop-by-Hop* |
| 4 | *Internet Protocol* |
| 6 | *Transmission Control Protokol* |
| 17 | *User Datagram Protocol* |
| 43 | *Routing Header* |
| 44 | *Fragment Header* |
| 45 | *Inter-Domain Routing Protocol* |
| 46 | *Resource Reservation Protocol* |
| 50 | *Encapsulating Security Payload* |
| 51 | *Authentication Header* |
| 58 | *Internet Control Message Protocol* |
| 59 | *No Next Header* |
| 60 | *Destination Options Header* |

sześciu nagłówków rozszerzonych. Każdy z nagłówków powinien pojawić się co najwyżej raz w obrębie pakietu. Wyjątkiem jest nagłówek *Destination Header*, który może wystąpić dwukrotnie. Zawiera on informacje, które są przeznaczone dla docelowego hosta. Jeżeli w pakiecie jest obecny *Routing Header*, dodatkowy nagłówek *Destination Header* może przenosić informacje, które będą przetwarzane przez wszystkie węzły sieci, wymienione w nagłówku *Routing Header*. Rozszerzone nagłówki (z wyjątkiem

*Hop-by-Hop*) są przetwarzane wyłącznie przez docelowe hosty. Nagłówek *Hop-by-Hop* może być użyty do transmisji dodatkowych informacji, które będą przetwarzane przez pośredniczące hosty. Aby nagłówki były poprawnie przetwarzane, muszą się pojawiać w ściśle określonej kolejności. Opcje w nagłówku oraz semantyka związana z jego przetwarzaniem może spowodować odrzucenie pakietu oraz wysłanie komunikatu *ICMP Parameter Problem*. Odrzucenie pakietu może być spowodowane, np. nieoczekiwaną zawartością pola *Next Header*. Nagłówki *Hop-by-Hop* oraz *Destination Options* mogą przenosić zmienną liczbę opcji, zakodowaną w formacie *Type-Hop-Value* w skrócie *TVL* (rys. 4).

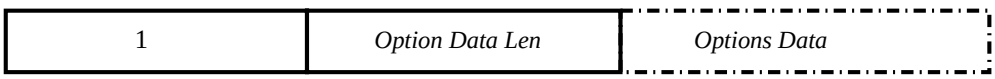

***Rys. 4.*** *Format TVL*

Ośmiobitowe pole *Options Type* ma opcje związane z przetwarzaniem nagłówka. Dwa najstarsze bity określają akcje, jakie ma podjąć host docelowy w przypadku nierozpoznania przez niego opcji (tabl. 2).

***Tabl. 2. Opcje pola Options Type***

| Dwa najstarsze bity | Rodzaj akcji |
|---|---|
| 00 | Pomiń opcje, kontynuuj przetwarzanie pakietu |
| 01 | Usuń pakiet |
| 10 | Usuń pakiet i niezależnie od tego, czy adres docelowy jest *Multicastem*, czy nie, do nadawcy wyślij *ICMP Parameter Problem Code 2* |
| 11 | Usuń pakiet i jeśli adres docelowy nie jest *Multicastem*, do nadawcy wyślij *ICMP Parameter Problem Code 2* |

Niektóre z opcji zawartych w pakiecie mogą być zmieniane podczas przetwarzania pakietu przez węzły pośredniczące w jego transmisji. Trzeci bit w polu *Options Type* jest używany do określania, kiedy opcje nie mogą być modyfikowane. Bit ustawiony na zero zabrania modyfikacji opcji. Możliwość ta jest wykorzystywana wtedy, gdy używa się nagłówka *Authentication*. Zawartość tego nagłówka jest obliczana na postawie zawartości pakietu i gwarantuje, że zawartość pakietu nie uległa zmianie. Nagłówek powinien mieć długość wyrażoną liczbą podzielną przez 8. Aby to osiągnąć, stosuje się

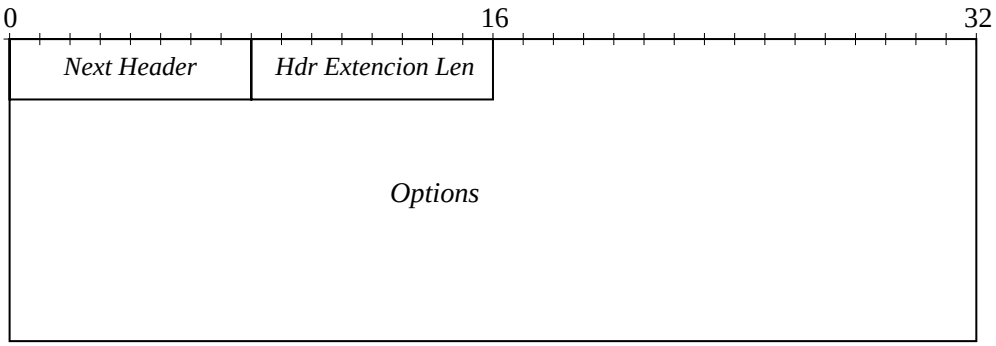



***Rys. 5.*** *Format PadN*

specjalne opcje (*Pad1* i *PadN*) dodające dodatkowe oktety do nagłówka. Format *PadN* przedstawiono na rys. 5. Pierwszy bajt ma wartość 1, a w drugim jest zawarta liczba określająca liczbę bajtów „wyrównujących", przy czym dla $N$ bajtów wyrównujących należy podać liczbę $N-2$. Wynika to z konieczności uwzględnienia dwóch bajtów przeznaczonych na nagłówek wyrównujący. Bajty uzupełniające zawierają zera. Opcja *Pad1* ma ustawione pole nagłówka, określające liczbę dodanych bajtów na zero i składa się z dwóch bajtów nagłówka wyrównującego.

### *Nagłówek Hop-by-Hop*

Nagłówek *Hop-by-Hop* przenosi wiele informacji, które muszą być przetwarzane przez wszystkie węzły sieci pośredniczące w jego transmisji. Format nagłówka pokazano na rys. 6.



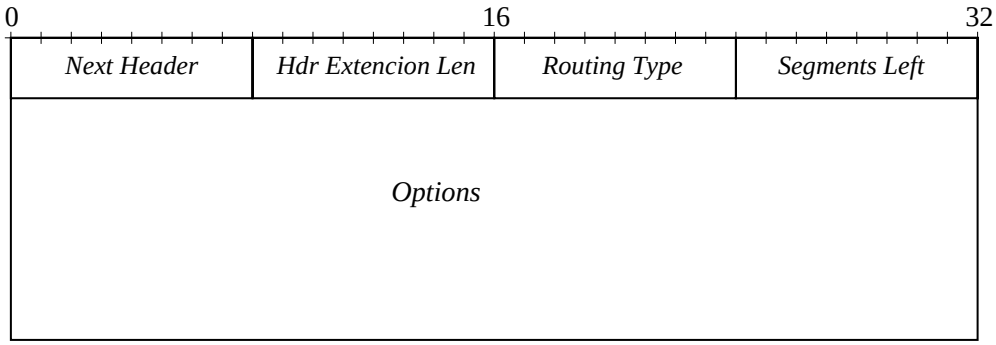

***Rys. 6.*** *Nagłówek Hop-by-Hop*

W nagłówku znajdują się dwa ośmiobitowe pola *Next Header* i *Header Extention Len* oraz pole *Options* niezdefiniowanej długości. Pole *Next Header* zawiera kod nagłówka, który wystąpi po nagłówku *Hop-by-Hop*. Pole *Header Extention Len* określa całkowitą długość nagłówka, wyrażoną w ośmiooktetowych jednostkach bez wliczania pierwszych ośmiu oktetów nagłówka. W polu *Options* są opcje zakodowane w formacie TVL, które będą przetwarzane przez wszystkie węzły pośredniczące w transmisji pakietu IPv6. Przetwarzanie nagłówka przez wszystkie węzły pośredniczące może być wykorzystywane, np. przez protokół RSVP, do „przeegzaminowania" ruterów pośredniczących na temat możliwości rezerwacji pasma. Można też wykorzystać nagłówek do sprawdzenia, czy urządzenia pośredniczące w transmisji mogą przenosić pakiety *Jumbodatagram*.

### *Nagłówek Routing Header*

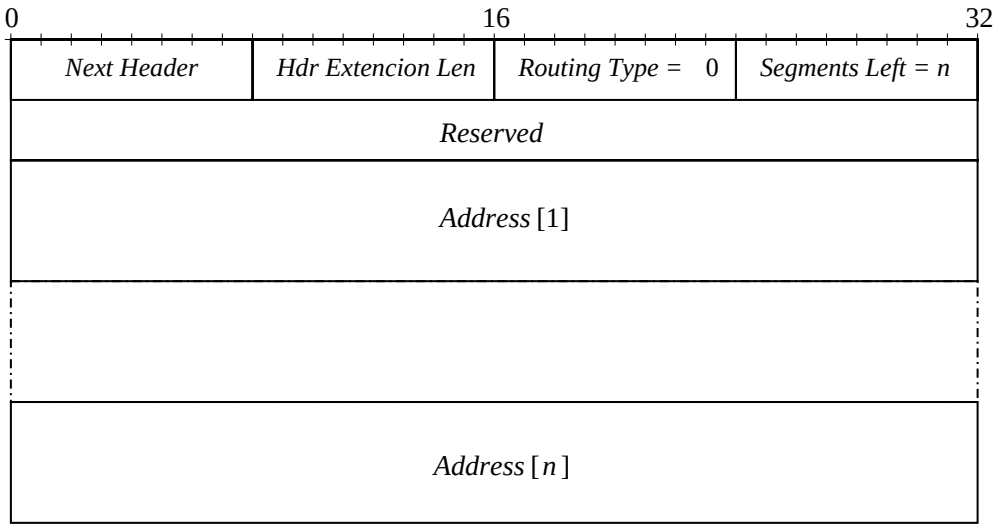
Za pomocą nagłówka *Routing Header* host źródłowy może określić węzły, przez które pakiet musi przejść, aby dojść do celu. Umożliwi to, po ustanowieniu połączenia między dwoma węzłami, ujednolicenie drogi wszystkich pakietów transmitowanych w ramach tego połączenia, a także wsparcie protokołu RSVP. Innym wykorzystaniem możliwości nagłówka *Routing Header* może być komunikacja między siecią firmową a mobilnym użytkownikiem, znajdującym się poza siecią firmy. Format nagłówka *Routing Header* przedstawiono na rys. 7. Pola *Next Header* i *Header Extention Len* mają takie same znaczenie, jak w nagłówku *Hop-by-Hop*. Ośmiobitowe pole *Routing Type* określa jeden z typów rutingu, który będzie stosowany. W polu *Segments Left* jest zawarta liczba hostów, które musi pokonać pakiet, aby osiągnąć cel.

Dotychczas zdefiniowano tylko jeden typ rutingu –i typ 0. Format *Routing Header Type 0* nagłówka zaprezentowano na rys. 8. W 32-bitowym polu *Reserved* znajdują się zera, które są ignorowane przez




***Rys. 7.*** *Nagłówek Routing Header*

host docelowy. Natomiast w polach *Address* występuje *N* 128-bitowych adresów hostów IPv6, przez które będzie musiał przejść pakiet.

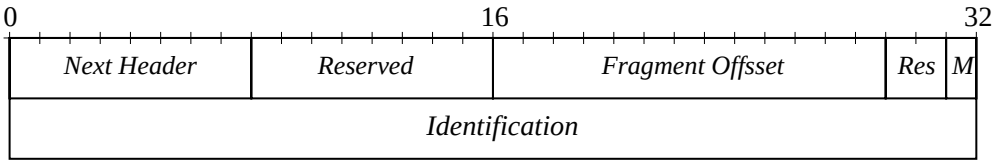



***Rys. 8.*** *Nagłówek Routing Header Type 0*

### *Nagłówek Fragment Header*

W protokole IPv6 – przed wysłaniem pakietu – nadawca bada, jaki jest minimalny MTU na ścieżce, po której będzie wędrował pakiet. Minimalne MTU staje się obowiązujące dla nadawcy. Dzięki temu w czasie transmisji nie jest dokonywana (przez węzły pośredniczące) defragmentacja pakietów. Zadanie defragmentacji obowiązuje wyłącznie nadawcę. *Fragment Header* jest identyfikowany przez kod 44 i składa się z 6 pól (rys. 9). Pole *Next Header* określa typ nagłówka, który znajduje się za nagłówkiem *Routing Header*, jeśli jest to pierwszy z pakietów. W następnych pakietach jest umieszczana wartość 59, co oznacza brak kolejnych nagłówków. Nagłówki rozszerzające, które mogą być w pakiecie, powinny być przetransmitowane w pierwszym z przefragmentowanych pakietów. Pole *Reserved* zawiera zera i jest ignorowane przez host docelowy. Trzynastobitowe pole *Fragment Offset* określa przesunięcie

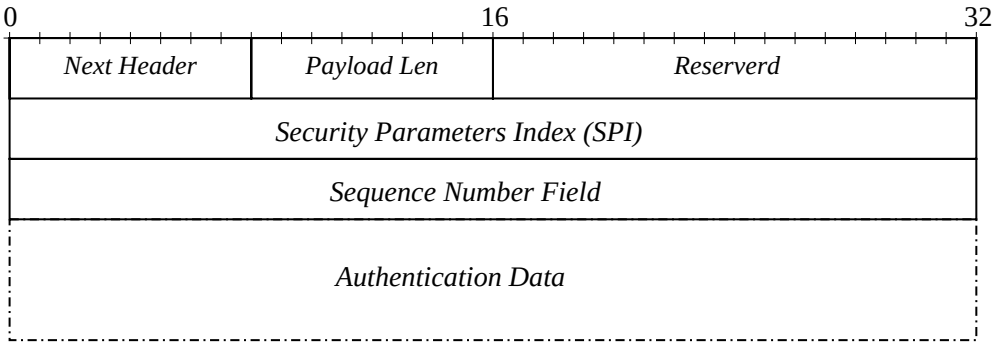
fragmentu danych w stosunku do początku defragmentowanego pakietu. Przesunięcie wyrażone jest w ośmiooktetowych jednostkach. Pole *Res* jest zarezerwowane i wyzerowane. Wartość 1 w polu *M* oznacza, że są następne fragmenty, a 0 – że to ostatni pakiet. W polu *Identification* jest umieszczany unikalny identyfikator, taki sam dla wszystkich fragmentowanych pakietów.




***Rys. 9.*** *Nagłówek Fragment Header*

### *Nagłówek Authentication Header*

Nagłówek *Authentication Header* zapewnia bezpieczne połączenia między nadawcą a odbiorcą. Gdy w pakiecie znajduje się nagłówek *Authentication Header*, pakiet nie może być modyfikowany. Format nagłówka pokazano na rys. 10. Nagłówek składa się z sześciu pól. Pole *Next Header* określa typ następnego nagłówka, a ośmiobitowe pole *Payload Len* – długość nagłówka w czterooktetowych jednostkach bez pierwszych ośmiu oktetów. Szesnastobitowe pole *Reserved* jest wypełnione zerami i nie jest przetwarzane. Natomiast 32-bitowe pole *Security Parameters Index* jest kombinacją zawartości tego pola, adresu docelowego i protokołu bezpieczeństwa. Pole to określa, tak zwany *Security Assotiation*. W polu *Sequence Number* jest kolejny numer sekwencyjny, zwiększany w czasie transmisji. Pole *Authentication Data* zawiera tzw. *Integrity Check Value* – ICV. Długość ICV powinna być liczbą podzielną na ośmiooktetowe jednostki. Zależy ona od rodzaju użytego algorytmu.




***Rys. 10.*** *Nagłówek Authentication Header*

Nagłówek *Authentication Header* może być użyty w dwóch „trybach" pracy. W jednym z trybów nagłówek jest dodany do pakietu za nagłówkiem IPv6 i *Hop-by-Hop*, zapewniając integralność całego pakietu. W trybie tunelowania taki pakiet jest umieszczany w nowym pakiecie IPv6.

### *Nagłówek Encapsulating Security Payload Header*

Za pomocą nagłówka *Authentication Header* można jedynie zapewnić integralność przesyłanych danych, ale nie można zabezpieczyć ich przed przeglądaniem. Natomiast nagłówek *Encapsulating*

*Security Payload Header* umożliwia szyfrowanie całego pakietu. Dane mogą być zaszyfrowane razem z pakietem (który może być tunelowany) lub z częścią nagłówka.

***Nagłówek Destination Options Header***

Czasami nadawca musi dostarczyć pakiet dodatkowych informacji, które mogą być przetworzone przez odbiorcę. Można do tego użyć nagłówka *Destination Options*, wyłącznie przetwarzanego przez host docelowy. Format nagłówka, sposób kodowania opcji oraz wyrównywania długości nagłówka jest taki, jak nagłówka *Hop-by-Hop*. Są dwie możliwości umieszczania nagłówka *Destination Options*. W wersji pierwszej znajduje się on tuż za nagłówkiem *Hop-by-Hop* i jest przetwarzany przez host docelowy. W wersji drugiej jest ulokowany jako ostatni, czyli może zostać zaszyfrowany razem z danymi. Umożliwia to dostarczenie dodatkowych opcji zabezpieczonych przed podsłuchem i modyfikacją.

## *Adresowanie w protokole IPv6*

Pole adresowe w protokole IPv6 ma 128 bitów. Teoretycznie można więc zaadresować $2^{128}$ urządzeń. Praktycznie możliwości są nieco mniejsze. Wynika to głównie z dążenia do optymalizacji tablic rutingu. W przypadku protokołu IPv4 nie jest możliwa agregacja tablic rutingu i nie można tego już usunąć. System rozdawania w sposób nieco nieskoordynowany adresów spowodował rozrost tablic rutingu. Aby tego uniknąć, w nowym protokole zdecydowano się na wprowadzenie hierarchicznego sposobu rozdziału i rutowania. W tym celu na pierwszych bitach pola adresowego określa się typ adresu IPv6.

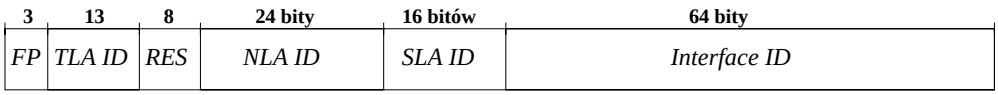
Wyróżniono następujące prefiksy i odpowiadające im typy adresów:

– 0000 0000: zarezerwowany,
– 0000 001: zarezerwowany na potrzeby NSAP,
– 0000 010: zarezerwowany na potrzeby protokołu IPX[①],
– 001: *Global Unicast Address*,
– 1111 1110 10: *Link-Local Unicast Address*,
– 1111 1110 11: *Site-Local Unicast Address*,
– 1111 1110 11: *Multicast Address*.

Pozostałe prefiksy mają status *Unasigned* i ich przynależność zostanie rozstrzygnięta w przyszłości.

***Adresy Unicast***

Z punktu widzenia Internetu najbardziej liczą się adresy *Unicast*, gdyż będą służyły do podstawowej wymiany danych. Aby usprawnić ruting, w RFC2374 zaproponowano podział globalnie rutowalnych adresów według struktury przedstawionej na rys. 11.



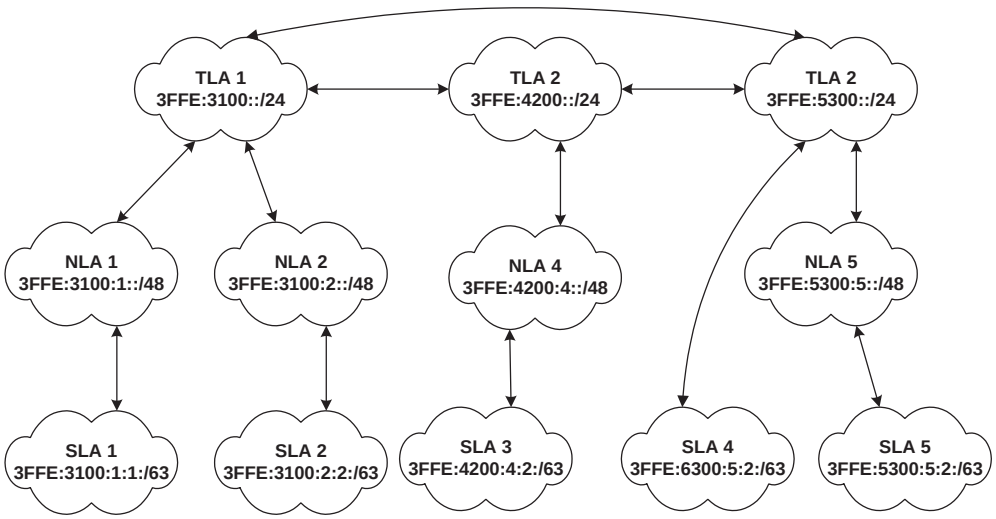

***Rys. 11.*** *Podział adresów IPv6*

① *Firma Novell, która opracowała protokół IPX, przestała wspierać jego rozwój.*

Jak pokazano na rys. 11, adres został podzielony na sześć sekcji. Ich znaczenie jest następujące:

- **FP** (*Format Prefix*): 3-bitowe pole, które ma stałą wartość binarną 001; określa to adres typu *Global Unicast Address*, umożliwiający komunikację typu 1:1 między dwoma hostami, mającymi ten typ adresu w obrębie całej sieci Internet;
- **TLA ID** (*Top Level Aggregator Identifier*): 13-bitowe pole, określające dostawcę usług internetowych; w zamierzeniu twórców tego podziału identyfikator TLA mają otrzymywać wyłącznie operatorzy ponadnarodowi, działający w samym „rdzeniu" Internetu, tacy jak *Sprint* czy *Word Com*; TLA mają delegować bloki adresowe do mniejszych operatorów lub klientów;
- **RES**: osiem bitów na razie zarezerwowanych do wykorzystania w przyszłości; mogą one zostać użyte do rozszerzenia bądź pola TLA, bądź SLA;
- **NLA ID** (*Next Level Aggregator Identifier*): 24-bitowe pole, określające lokalnego dostawcę usług internetowych bądź organizacje; NLA jest otrzymywane od TLA; TLA może mieć jedno lub więcej połączeń do NLA;
- **SLA ID** (*Site Level Aggregator*): 16-bitowe pole, określające bądź klienta, bądź małego dostawcę internetowego; SLA powinno być delegowane przez NLA, choć nie jest wykluczone delegowanie go przez TLA;
- **Interface ID** (*Interface Identifier*): 64-bitowe pole, przeznaczone do określenia identyfikatora interfejsu hosta; identyfikator może być tworzony automatycznie, np. z wykorzystaniem MAC Address karty sieciowej oraz uzupełnieniem dwóch pierwszych bajtów wartością :FF:FE.

Ten podział adresów wymusza agregację adresów na najwyższym poziomie. Usprawnia to ruting, gdyż decyzje o wyborze drogi są podejmowane na podstawie kilku pierwszych bitów adresu. Przyszła struktura Internetu powinna być taka, jak na rys. 12.



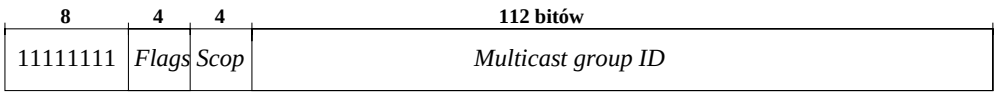

***Rys. 12.*** *Struktura Internetu na adresach IPv6*

Jest to nieco odmienne od tego, co występuje obecnie. Dziś protokół IPv4 ma systemy autonomiczne. Dostawca usług lub firma może wystąpić o otrzymanie własnego numeru autonomicznego. Wraz z nim

może uzyskać też pewną pulę numerów IP, określanych jako adresy PI (*Provider Independent*). Posiadanie własnego numeru ASi (*Autonomous System*) oraz adresów IP umożliwia dołączenie firmy do jednego lub wielu operatorów, a także dowolną zmianę operatora. Możliwość ta jest określana jako *Multihoming*. W przypadku podziału zaproponowanego dla IPv6 *Multihoming* jest bardzo utrudniony, a nawet niemożliwy. Pakiet transmitowany od SLA 1 do SLA 4 (rys. 12) musi przejść przez urządzenia należące do obu TLA, od których otrzymał numery IP. Jest to uwstecznienie w stosunku do tego, co jest obecnie, ale należy pamiętać, że podział ten został zaproponowany w 1998 r. i może być na nowo przedyskutowany przed ostatecznym wprowadzeniem protokołu IPv6 w życie.

### *Adresy Multicast*

Internet oparty na protokole IPv6 ma wspierać różne rodzaje transmisji danych. W klasycznym przypadku jest to wymiana danych typu 1 : 1. Są jednak takie rodzaje transmisji (np. radio i telewizja), które charakteryzują się transmisją $1 : N$ lub $N : M$, gdzie $N$, $M > 1$. Aby zapewnić lepsze wsparcie dla tego typu transmisji, w protokole IPv6 zdefiniowano specjalny adres multicastowy. Format tego adresu zaprezentowano na rys. 13.




***Rys. 13.*** *Adres multicastowy w IPv6*

Osiem pierwszych bitów ustawionych na 1 tworzy adres multicastowy. Kolejne cztery, nazywane *Flags*, nie są do końca zdefiniowane. Pierwsze trzy powinny być ustawione na zero, natomiast czwarty (zwany *t*-bit) wskazuje, czy ustawienie adresu multicastu jest na stałe[①], czy tymczasowe[②]. *Scop* oznacza zakres widoczności adresu multicastowego. Zdefiniowano następujące wartości tego pola:

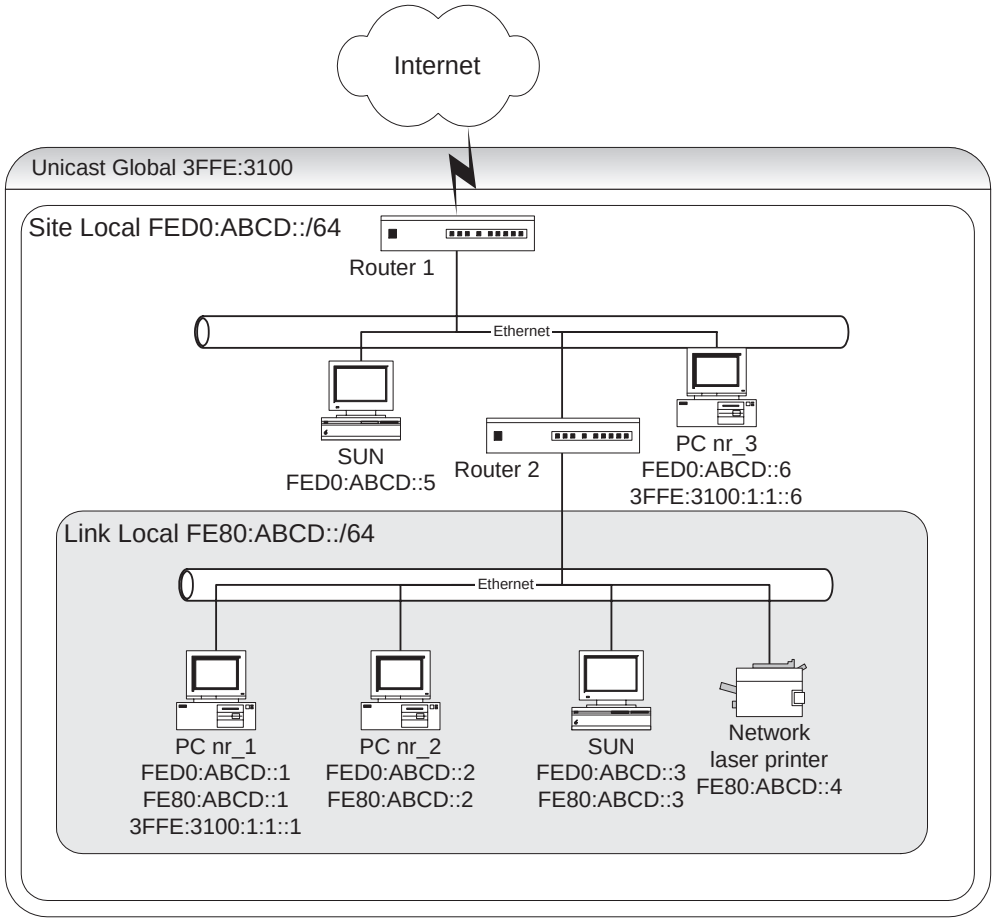
- 0000: zarezerwowane,
- 0001: *Node-Local Scope*,
- 0010: *Link-Local Scope*,
- 0101: *Site-Local Scope*,
- 1000: *Organization-Local Scope*,
- 1110: *Global Scope*,
- 1111: zarezerwowane.

Za pomocą ustawienia odpowiednich wartości pola *Scop* można rozszerzać lub ograniczać zakres widoczności danego adresu – od sieci lokalnej, przez sieć firmy, aż do całego Internetu.

### *Adresy Link-Local i Site-Local*

W protokole IPv4 wyodrębniono trzy grupy adresów prywatnych. Mają one swój odpowiednik w protokole IPv6. Są to adresy typu *Link-Local* i *Site-Local*, których funkcjonowanie pokazano na rys. 14. Adres *Link-Local* ma na pierwszych 10 bitach pola adresowego ustawioną wartość 1111 1110 10.

① *Np. adres rozgłośni radiowej.*
② *Przykładem może być transmisja meczu odbywająca się w sobotę.*




***Rys. 14.*** *Adresy Link-Local i Site-Local w IPv6*

Ogranicza on „widoczność" hosta do lokalnej sieci, np. drukarka na rys. 14 z tego typu adresem jest dostępna tylko dla komputerów z tego samego segmentu sieci LAN. Ruter 2 nie powinien wypuszczać pakietów z takim adresem źródłowym poza sieć LAN. Adresy typu *Site-Local* mają na pierwszych 10 bitach ustawioną wartość 1111 1110 11. Zapewniają one łączność w obrębie firmy i tylko firmy. Za pomocą takiego adresu nie można połączyć się z hostem w Internecie. Według założeń sieci z rys. 14, ruter 2 powinien rutować takie pakiety, natomiast ruter 1 nie powinien ich wypuszczać poza obręb sieci. Możliwe są więc połączenia między komputerami SUN, ale tylko jeden z adresem typu *Site-Local* może współpracować z drukarką.

### *Adresy Anycast*

Adresy *Anycast* stanowią specjalną odmianę adresów w protokole IPv6. Różnią się one od adresów typu *Unicast* tym, że wielokrotnie mogą występować takie same. Dzięki temu jest możliwe rozkładanie obciążeń między serwerami. Zaleca się ich stosowanie do usług realizowanych z wykorzystaniem bezpołączeniowego protokołu UDP, gdyż w razie zerwania łączności z jednym serwerem nie ma

groźby zerwania sesji. W przypadku protokołu TCP zerwanie łączności oznacza zerwanie sesji, którą z serwerem zapasowym trzeba nawiązać od początku.

## *Wdrożenie protokołu IPv6*

Protokół IPv6, mimo swoich zalet, z trudem toruje sobie drogę do wdrożenia na większą skalę. Termin przejścia z IPv4 na IPv6 jest ciągle przesuwany. Składa się na to wiele przyczyn. Być może najważniejszą są koszty. Pracujące obecnie w Internecie rutery są zoptymalizowane do pracy z pakietami IPv4. Można w nich wprawdzie wymienić oprogramowanie na takie, które będzie wspierało IPv6, lecz nie rozwiąże to problemu sprzętowego wsparcia dla IPv6, czyli wymianę urządzeń. Jest to operacja dość kosztowna i czasochłonna. Trzeba również pamiętać, że w protokole IPv4 zdefiniowano wiele nowych usług multimedialnych, takich jak, np. telefonia IP. W ich wdrożenie operatorzy zainwestowali spore pieniędze, które najczęściej jeszcze się nie zwróciły, więc w obliczu recesji nie są zainteresowani ponoszeniem kosztów nowych inwestycji. Inną, istotną przyczyną opóźniania wdrożenia IPv6 jest brak wsparcia dla tego protokołu w systemach operacyjnych różnych producentów. Firma Microsoft wspiera IPv6 dopiero od wersji Windows 2000, a SUN Microsystem – dopiero od wersji Solaris 8.0. Firma CISCO, największy dostawca ruterów na świecie, promuje protokół IPv6 w swoim systemie operacyjnym dla ruterów dopiero od wersji 12.2T. Czynnikiem, który może wymusić wymianę protokołu IPv4 na IPv6 jest niedostatek adresów IP w protokole IPv4. Nie jest to jednak obecnie tak krytyczne, jak jeszcze kilka lat temu. Główną przyczyną jest zmiana sposobu budowania sieci komputerowych. Kiedyś komputery pracujące w sieci dołączanej do Internetu musiały mieć publiczne numery IP, a dziś stosuje się techniki translacji adresów IP. Numery publiczne mają tylko serwery świadczące usługi, do których jest potrzebny dostęp z Internetu. Zmalało więc zapotrzebowanie na numery IP, co przesunie wyczerpanie się numerów IP na termin późniejszy. Szacuje się, że masowe wdrożenie UMTS może spowodować wzrost zapotrzebowania na adresy IP. Ze względu na koszty oraz recesję nie nastąpi to jednak szybko. Obecnie protokół IPv6 jest testowany w sieci 6bone. W Polsce siecią 6bone zajmuje się Interdyscyplinarne Centrum Modelowania Matematycznego[①]. W ramach tego projektu protokół IPv6 jest tunelowany w protokole IPv4. Testowane są usługi, które mają pracować na podstawie protokołu IPv6. Można będzie zatem uniknąć problemów w aplikacjach, pracujących w nowym środowisku. Te testy oraz pojawienie się ruterów optymalnie pracujących z IPv6 powinno umożliwić jego praktyczne wdrożenie. Kiedy to nastąpi, jeszcze nie wiadomo.

## *Bibliografia*

[1] Deering S., Conta A.: *Internet control message protocol (ICMPv6) for the Internet protocol version 6 (IPv6)*. Network Working Group, Dec. 1998

[2] Deering S., Hinden R.: *IP version 6 addressing architecture*. Network Working Group, July 1998

[3] Deering S., Hinden R., O'Dell M.: *An IPv6 aggregatable global unicast address format*. Network Working Group, Dec. 1998

[4] Hinden R., Borman D., Deering S.: *IPv6 Jumbograms*. Network Working Group, Aug. 1999

[5] Hinden R., Deering S.: *Internet protocol, version 6 (IPv6) specification*. Network Working Group, Dec. 1998

[6] *Ipng standardization status*, http://playground.sun.com/pub/ipng/html/specs/standards.html

[7] *Ipv6: The next generation Internet*, http://www.ipv6.org

① *Więcej informacji na ten temat można znaleźć na stronach internetowych http://www.6bone.pl*

[8] Mogul J., McCann J., Deering S.: *Path MTU discovery for IP version 6*. Network Working Group, Aug. 1996

[9] Narten T., Thompson S.: *IPv6 stateless address autoconfiguration*. Network Working Group, Dec. 1998

[10] Rockell R., Wegner J. D.: *IP addressing and subnetting including IPv6*. Rockland: Syngress Media, 2000

[11] Simpson W., Narten T., Nordmark E.: *Neighbor discovery for IP version 6 (IPv6)*. Network Working Group, Dec. 1998

## *Piotr Jankowski*

Mgr inż. Piotr Jankowski (1969) – absolwent Wydziału Elektroniki i Technik Informacyjnych Politechniki Warszawskiej (1996); pracownik Politechniki Warszawskiej (od 1996) oraz Instytutu Łączności w Warszawie (od 1997); zainteresowania naukowe: sieci komputerowe – bezpieczeństwo i administracja.
e-mail: P.Jankowski@itl.waw.pl