Metody integracji sieci IPv4 i IPv6

Janusz FURTAK

Instytut Teleinformatyki i Automatyki WAT,
ul. Gen S. Kaliskiego 2, 00-908 Warszawa
jfurtak@wat.edu.pl

STRESZCZENIE: W artykule przedstawiono metody integracji sieci IPv4 i IPv6. Opisano mechanizmy podwójnego stosu IP i translacji protokołów wykorzystywane do komunikacji pomiędzy hostami sieci tylko-IPv4 a hostami tylko-IPv6. Dużo uwagi poświęcono mechanizmom tunelowania stosowanym do wymiany danych pomiędzy „wyspami IPv6" poprzez środowisko IPv4 oraz pomiędzy „wyspami IPv4" poprzez środowisko IPv6. Opisano zasady działania i właściwości tuneli manualnych (*Manual Mode*, GRE i *Generic Packet Tunneling*), tuneli automatycznych (*Automatic IPv4 Compatible Mode*, *Automatic Mode*, *Teredo* i ISATAP) oraz mechanizmów wspomagających konfigurowanie tuneli (*broker tuneli* i *serwer tuneli*).

SŁOWA KLUCZOWE: sieci IPv6, integracja sieci IPv4 i IPv6, tunelowanie protokołów

1. Wprowadzenie

Lata dziewięćdziesiąte XX w. cechowały się burzliwym rozwojem sieci Internet. Wówczas powszechnie stosowany był protokół IP w wersji czwartej (IPv4). Stosowanie adresacji klasowej i nieodpowiednia dystrybucja grup adresów na poszczególne kontynenty (w konsekwencji na kraje) spowodowały szybkie wyczerpywanie puli adresów IPv4. Problem wyczerpywania się adresów IP mógł być rozwiązany albo poprzez wszelkie zabiegi mające na celu oszczędne gospodarowanie adresami, albo przez zwiększenie długości pola adresowego.

Pierwsze z tych rozwiązań możliwe było do wprowadzenia w stosunkowo krótkim czasie. Wykorzystywano tutaj początkowo maski sieciowe stałej długości[1] opisane w RFC 950 (1985 r.) [20], a później maski zmiennej długości

[1] Maski sieciowe stałej długości dawały możliwość wyznaczania identyfikatora sieci o długości większej niż 8, 16 lub 24 bity (zależnie od przynależności danego adresu do klasy sieci

(VLSM) opisane w RFC 1009 (1987 r.) [2] i bezklasowy routing międzydomenowy (CIDR)[2] opisany w RFC 1518 (1993 r.) [24], które są stosowane do dnia dzisiejszego. W lutym 1996 roku dokument RFC 1918 [23] wprowadził podział pełnego zakresu adresów IP na pulę tzw. adresów publicznych i prywatnych. W tym przypadku oszczędność polegała na możliwości wielokrotnego (w skali całej sieci Internet) wykorzystywania adresów prywatnych. Kosztem takiego rozwiązania jest konieczność stosowania serwerów NAT [2], [26], których zadaniem jest translacja adresów prywatnych na adresy publiczne i odwrotnie.

Zwiększenie długości pola adresowego, chociaż pozornie wydaje się rozwiązaniem najprostszym, w rzeczywistości jest rozwiązaniem bardzo skomplikowanym. Wymaga ono zmiany oprogramowania na wszystkich węzłach, które miałyby wykorzystywać nowy adres, i nowej implementacji powszechnie stosowanych aplikacji w sieci Internet (np. serwerów usług poczty elektronicznej, WWW, DNS czy FTP).

W rezultacie rozwiązania mające na celu oszczędne gospodarowanie adresami wprowadzono w połowie lat dziewięćdziesiątych ubiegłego stulecia i równolegle prowadzono prace nad opracowaniem nowego protokołu IP w wersji szóstej (IPv6). Nowy protokół oprócz zwiększenia długości pola adresowego zlikwidował szereg mankamentów protokołu IPv4, do których należy zaliczyć: często używany ruch rozgłoszeniowy (*ang. broadcast traffic*), konieczność przeliczania pól nagłówka każdej ramki na każdym routerze pośredniczącym w transmisji, brak wielopoziomowej hierarchii adresów. Ponadto protokół IPv6 [2] wprowadził nowe rozwiązania, na przykład: autokonfigurację adresu warstwy sieciowej niezauważalną dla użytkowników, mechanizm przenumerowywania adresów, obsługę mobilności i zintegrowane z protokołem IPv6 środowisko IPSec przeznaczone do zabezpieczania transmitowanych danych.

W ramach prac nad protokołem IPv6 zdawano sobie sprawę z tego, że przejście z powszechnie używanego protokołu IPv4 do nowo projektowanego protokołu nie będzie zadaniem łatwym i krótko trwającym. Równolegle z opracowywaniem rozwiązań dla IPv6 przygotowano szereg rozwiązań umożliwiających współistnienie sieci wykorzystujących protokół IPv4 i IPv6. Zasady ogólne stopniowego przejścia od sieci IPv4 do sieci IPv6 zostały określone w dokumentach RFC 4213 [22] i RFC 3904 [16]. Opisy stosowanych rozwiązań w tym zakresie są tematem artykułu.

---

odpowiednio A, B lub C), to znaczy wyznaczania podsieci o rozmiarze dostosowanym do liczności węzłów w sieci, co w konsekwencji prowadziło do zmniejszenia marnotrawienia adresów IP. Ograniczeniem tego rozwiązania była konieczność stosowania takiej samej maski sieciowej we wszystkich podsieciach danej sieci.

[2] Mechanizmy VLSM (Variable Length Subnet Mask) [2] i CIDR (Classless Interdomain Routing) [13], [24] umożliwiają efektywne gospodarowanie adresami IP i istotne zmniejszenie rozmiarów tablic routingu poprzez agregowanie tras.

2. Integracja sieci IPv4 i IPv6

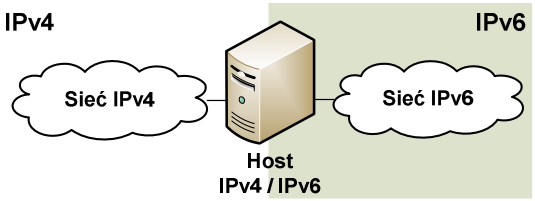
Integrację sieci IPv4 i IPv6 umożliwiają następujące rozwiązania[3]:

- mechanizm podwójnego stosu IP;
- mechanizm tunelowania;
- mechanizm translacji protokołów.

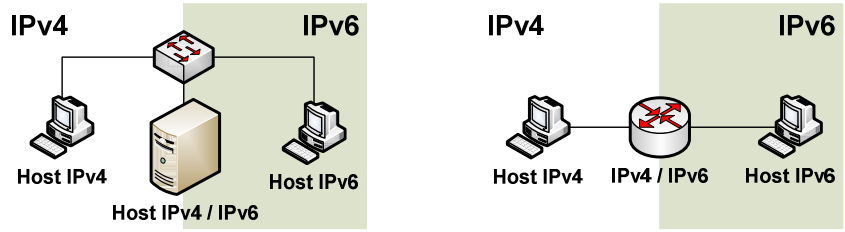
2.1. Mechanizm podwójnego stosu

Termin podwójny stos (*ang. dual stack*) odnosi się do węzłów sieci, których oprogramowanie stosu TCP/IP jest w stanie poprawnie obsługiwać protokół IP w wersji czwartej i w wersji szóstej. Nie oznacza to, że aplikacje uruchomione na komputerze z obsługą podwójnego stosu są w stanie wykorzystać pakiety pochodzące z sieci IPv4 i IPv6 – aplikacje również muszą być dostosowane do obsługi pakietów obu wersji protokołu IP.

Korzystanie z podwójnego stosu w rzeczywistości oznacza równoczesne korzystanie z sieci IPv4 i IPv6 (logiczna topologia sieci wykorzystująca węzły z obsługą podwójnego stosu jest pokazana na rys. 1, a przykłady implementacji takiej sieci na rys. 2).



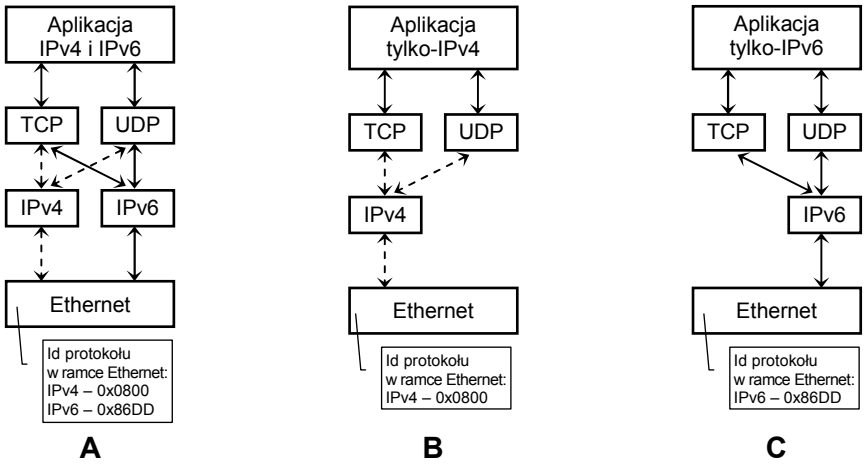
Rys. 1. Logiczna topologia sieci wykorzystująca węzły z obsługą podwójnego stosu



Rys. 2. Przykłady implementacji sieci z węzłami z obsługą podwójnego stosu

[3] Internet Engineering Task Force w dokumencie RFC 4213 [22] wprowadził mechanizm podwójnego stosu i mechanizm tunelowania, a o translacji protokołów jest mowa w RFC 2766 [30].

Funkcjonowanie podwójnego stosu można schematycznie przedstawić tak, jak to pokazano na rys. 3.



Rys. 3. Schemat działania podwójnego stosu (na podstawie [6])

W przypadku aplikacji tylko-IPv4 (rys. 3B) wykorzystywana jest część stosu dla IPv4 i wtedy w nagłówku ramki Ethernet w miejscu identyfikatora protokołu warstwy sieciowej jest wpisywana wartość 0x0800, a w przypadku aplikacji tylko-IPv6 (rys. 3C) wykorzystywana jest część stosu dla IPv6 i wtedy w nagłówku ramki Ethernet wpisywana wartość 0x86DD. Pojawia się problem dla aplikacji obsługujących zarówno IPv4 jak i IPv6 (rys. 3A): który protokół powinien być używany? Protokół jest określany na podstawie użytego adresu. Natomiast
w przypadku próby pozyskania adresu z serwera DNS i uzyskania w odpowiedzi adresów IPv4 i IPv6 powinien być preferowany adres IPv6. Różne rozwiązania dotyczące podwójnego stosu są opisane w dokumentach RFC:

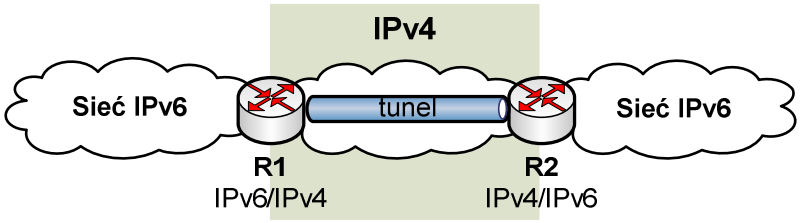
- technika *Bump-In-the-Stack (BIS)* – RFC 2767 [31];
- mechanizm *SOCKS-based IPv6/IPv4 gateway* – RFC 3089 [17];
- technika *Bump-in-the-API (BIA)* – RFC 3338 [19];
- programowanie na poziomie gniazd (*ang. socket Application Program Interface (API)*) – RFC 3493 [10] i 3542 [27].

Podwójny stos jest elastycznym narzędziem do ustanawiania sesji z innymi węzłami w sieci Internet wykorzystujących jedną z wersji protokołu IP. Możliwość automatycznego skorzystania z IPv6 lub pozostanie przy IPv4 (gdy nie jest możliwe zastosowanie protokołu w nowej wersji) powoduje, że proces migracji jest niezauważalny dla użytkownika końcowego. Możliwość obsługi podwójnego stosu przez węzły sieciowe warunkuje funkcjonowanie innych rozwiązań integracji sieci IPv4 i IPv6.

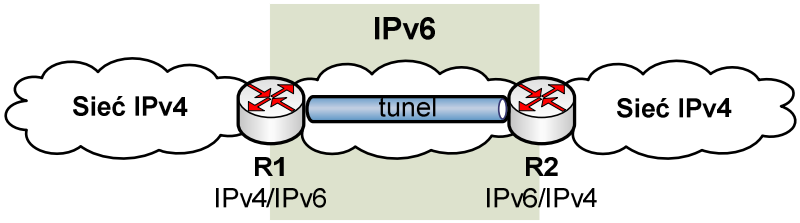
2.2. Tunelowanie

Tunelowanie polega na tym, że cały pakiet jednej wersji protokołu IP jest traktowany jako ładunek (to znaczy, że cały pakiet jest umieszczany w polu danych) pakietu innej wersji protokołu IP. Czynność ta nosi nazwę kapsułkowania lub enkapsulacji (*ang. encapsulation*). Czynność odwrotna polegająca na odpakowaniu pakietu nosi nazwę dekapsulacji (*ang. decapsulation*). Wykorzystując mechanizm tunelowania można transferować pakiety jednej wersji protokołu przez infrastrukturę drugiej wersji protokołu. Istnieje możliwość kapsułkowania pakietów protokołu IPv6 w pakietach protokołu IPv4 i odwrotnie. Tunelowanie pakietów może być zastosowane w dwóch różnych sytuacjach:

- izolowane sieci IPv6 (tzw. wyspy IPv6) wymieniają dane poprzez infrastrukturę IPv4 (rys. 4);
- izolowane sieci IPv4 (tzw. wyspy IPv4) wymieniają dane poprzez infrastrukturę IPv6 (rys. 5).



Rys. 4. Wyspy IPv6 wymieniają dane poprzez infrastrukturę IPv4



Rys. 5. Wyspy IPv4 wymieniają dane poprzez infrastrukturę IPv6

Pierwszy przypadek występuje częściej i jest odzwierciedleniem aktualnej sytuacji na świecie, w której pojawiające się izolowane sieci IPv6 wymieniają dane poprzez sieć Internet, w której powszechnie używanym protokołem jest IPv4. Drugi przypadek występuje w sytuacjach sieci zamkniętych (np. sieci wojskowych), w których infrastruktura jest zbudowana z urządzeń obsługujących nowy protokół IPv6, a użytkowane systemy mogą funkcjonować tylko w środowisku IPv4. W obu przypadkach routery **R1** i **R2** pełnią rolę końców tuneli i muszą obsługiwać podwójny stos.

Przy konstruowaniu tuneli trzeba mieć świadomość skutków ich stosowania. Należy tutaj zwrócić uwagę na następujące zagadnienia:

- zmniejszenie maksymalnej jednostki transmisji (MTU) tunelu (dołożenie nagłówka IPv4 do pakietu IPv6 zmniejsza maksymalny rozmiar pakietu IPv6 o 20 oktetów, co może spowodować fragmentację pakietu);
- niemożliwe jest skonstruowanie tunelu w sytuacji używania usługi NAT w protokole IPv4 z dynamiczną translacją adresu/portu (w takich sytuacjach można wykorzystać tylko tunelowanie Teredo);
- starsze routery przy sygnalizacji błędów w komunikacie ICMPv4 wysyłają do nadawcy nagłówek IPv4 pakietu, który jest źródłem sygnalizowanego błędu, i tylko osiem oktetów znajdujących się za tym nagłówkiem, co powoduje, że węzeł źródłowy przy wystąpieniu błędu nie jest w stanie rozpoznać adresów z pól zapakowanego pakietu IPv6;
- przy tunelowaniu pakietów IPv6 w pakietach IPv4 pole numer protokołu pakietu IPv4 ustawione jest na wartość 41, co bez modyfikacji konfiguracji funkcjonującej zapory sieciowej może spowodować niezamierzone odrzucanie pakietów.

Tunele mogą być konstruowane pomiędzy dwoma hostami, pomiędzy hostem a routerem i pomiędzy routerami. Mechanizmy tunelowania można pogrupować ze względu na sposób konfiguracji tuneli i zastosowania tuneli. Klasyfikacja mechanizmów tunelowania jest podana w tabeli 1.

Tabela 1. Klasyfikacja mechanizmów tunelowania

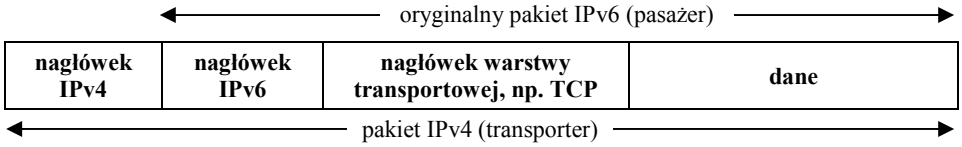
	wyspy IPv6 wykorzystujące infrastrukturę IPv4	**wyspy IPv4 wykorzystujące infrastrukturę IPv6**
tunele manualne	- tunelowanie w trybie *Manual Mode* [12], [22]; - tunelowanie GRE [7],[10]	- tunelowanie GRE; - tunelowanie w trybie *Generic Packet Tunneling* [4]
tunele automatyczne	- tunelowanie w trybie *Automatic IPv4 Compatible Mode* [12], [22]; - tunelowanie w trybie *Automatic Mode* [2]; - tunelowanie Teredo [14], [29]; - tunelowanie ISATAP [28]	---
mechanizmy wspomagające konfigurowanie tuneli	- broker tunelu (*ang. Tunnel Broker*) [8]; - serwer tuneli	---

2.2.1. Tunele manualne

Pojęcie tuneli manualnych odnosi się do tuneli, których konfiguracja jest wykonywana przez administratora na urządzeniach będących zakończeniami tunelu. Nakład pracy administratora jest proporcjonalny do liczby konfigurowanych tuneli. Rodzaj konfigurowanego tunelu nie ma istotnego wpływu na procedurę konfiguracji. Tunele konfigurowane manualnie różnią się między sobą przeznaczeniem tuneli (dane o zastosowaniach podane są w tabeli 1) i sposobem pakowania pakietu kapsułkowanego (pakietu pasażera) w pakiecie transportowym.

Tunelowanie w trybie *Manual Mode*

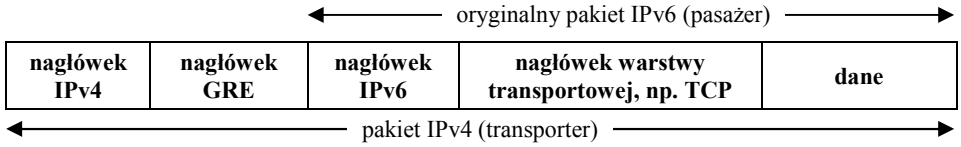
Ten sposób tunelowania może być wykorzystany do transferowania pakietów IPv6 poprzez infrastrukturę IPv4. Sposób kapsułkowania pakietów IPv6 w pakietach IPv4 w trybie *Manual Mode* jest pokazany na rys. 6. W polu „protokół" nagłówka IPv4 wpisywana jest liczba 41 wskazująca, że pakiet IPv6 jest ładunkiem tego pakietu, pole „adres nadawcy" określa adres interfejsu wyjściowego kapsułkującego węzła, a pole „adres odbiorcy" określa adres zdalnego końca tunelu. Pozostałe pola nagłówka IPv4 wyznaczane są zgodnie ze standardową procedurą.



Rys. 6. Kapsułkowanie pakietów IPv6 w pakietach IPv4 w trybie *Manual Mode*

Tunelowanie GRE

Tunelowanie GRE może być wykorzystane zarówno do transferowania pakietów IPv6 poprzez infrastrukturę IPv4 (przypadek «1») jak i do transferowania pakietów IPv4 poprzez infrastrukturę IPv6 (przypadek «2»). Sposób kapsułkowania pakietów IPv6 w pakietach IPv4 przy tunelowaniu GRE jest pokazany na rys. 7.



Rys. 7. Kapsułkowanie pakietów IPv6 w pakietach IPv4 przy tunelowaniu GRE

W przypadku kapsułkowania pakietów IPv4 w pakietach IPv6 przy

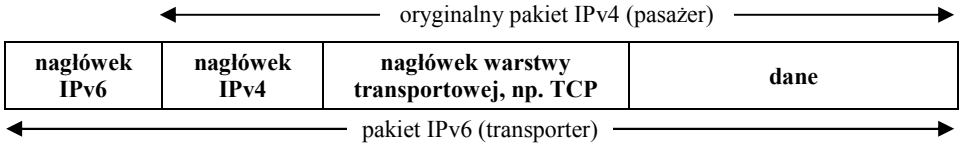
tunelowaniu GRE kapsułkowanie jest podobne (rolę pasażera pełni pakiet IPv4, a transportera pakiet IPv6).

W przypadku «1» w polu „protokół" nagłówka IPv4 wpisywana jest liczba 47 wskazująca, że ładunkiem tego pakietu jest protokół GRE, pole „adres nadawcy" określa adres interfejsu wyjściowego kapsułkującego węzła, a pole „adres odbiorcy" określa adres zdalnego końca tunelu. W nagłówku GRE w polu „typ protokołu" wpisana jest liczba 0x86DD wskazująca, że ładunkiem pakietu GRE jest pakiet IPv6.

W przypadku «2» w polu „następny nagłówek" nagłówka IPv6 wpisywana jest liczba 47 wskazująca, że ładunkiem tego pakietu jest protokół GRE, pole „adres nadawcy" określa adres interfejsu wyjściowego kapsułkującego węzła, a pole „adres odbiorcy" określa adres zdalnego końca tunelu. W nagłówku GRE w polu „typ protokołu" wpisana jest liczba 0x0800 wskazująca, że ładunkiem pakietu GRE jest pakiet IPv4.

Tunelowanie w trybie *Generic Packet Tunneling*

Ten sposób tunelowania jest przeznaczony do transferowania pakietów IPv4 poprzez infrastrukturę IPv6. Sposób kapsułkowania pakietów IPv4 w pakietach IPv6 w trybie *Generic Packet Tunneling* jest pokazany na rys. 8. W polu „następny nagłówek"[4] nagłówka IPv6 wpisywana jest liczba 4 wskazująca, że ładunkiem tego pakietu jest pakiet IPv4, pole „adres nadawcy" określa adres interfejsu wyjściowego kapsułkującego węzła, a pole „adres odbiorcy" określa adres zdalnego końca tunelu.



Rys. 8. Kapsułkowanie pakietów IPv4 w pakietach IPv6 w trybie *Generic Packet Tunneling*

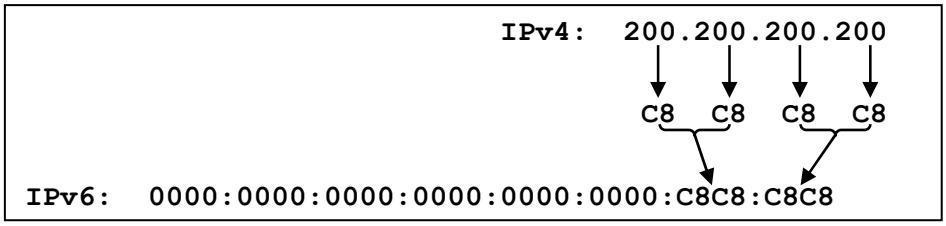
2.2.2. Tunele automatyczne

Pojęcie tuneli automatycznych odnosi się do sytuacji, w której urządzenie samodzielnie tworzy tunel do routera obsługującego podwójny stos. Mechanizmy tuneli automatycznych są przeznaczone tylko do transferowania danych pomiędzy wyspami IPv6 poprzez infrastrukturę IPv4.

[4] Kapsułkujący protokół IPv6 może wykorzystywać niezależnie od tunelowania nagłówki rozszerzenia. W takiej sytuacji w polu „następny nagłówek" głównego nagłówka IPv6 będzie wpisana liczba 60, a dopiero tak wskazany nagłówek rozszerzenia będzie zawierał liczbę 4 wskazującą pakiet IPv4 jako ładunek.

Tunelowanie trybie *Automatic IPv4 Compatible Mode*

W trybie *Automatic IPv4 Compatible Mode* sposób kapsułkowania pakietów IPv6 w pakietach IPv4 jest identyczny jak w trybie *Manual Mode* (rys. 6). W polu „protokół" nagłówka IPv4 wpisywana jest liczba 41 wskazująca, że pakiet IPv6 jest ładunkiem tego pakietu. W tym trybie tunelowania wykorzystywany jest specjalny rodzaj adresu IPv6 kompatybilny z IPv4. Adres ten charakteryzuje się tym, że bardziej znaczących 96 bitów adresu jest wyzerowanych, a pozostałe 32 bity odpowiadają adresowi IPv4 (sposób tworzenia takiego adresu jest pokazany na rys. 9).



Rys. 9. Procedura tworzenia adresu IPv6 kompatybilnego z adresem IPv4

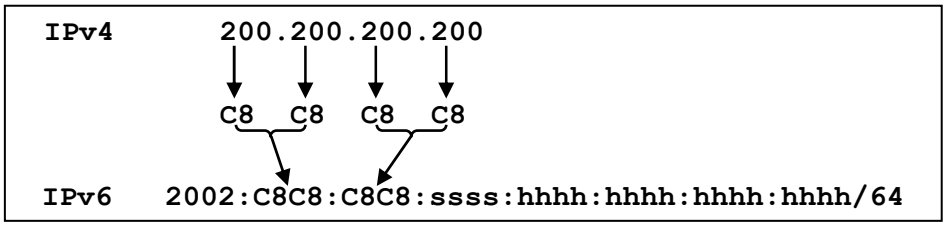
Konfiguracja tego trybu tunelowania wymaga skonfigurowania początku tunelu i uruchomienia rutowania dynamicznego po to, aby znana była trasa do docelowej sieci IPv6. Na trasie tunelu nie można używać usługi NAT. Moduł obsługi tunelowania odrzuca wszystkie pakiety IPv6 z wbudowanym adresem IPv4, który jest adresem rozgłoszeniowym, grupowym, adresem 0.0.0.0 lub 127.0.0.1.

Tunelowanie w trybie *Automatic Mode*

Sposób kapsułkowania pakietów IPv6 w pakietach IPv4 w trybie *Automatic Mode* jest identyczny jak w trybie *Manual Mode* (rys. 6). W polu „protokół" nagłówka IPv4 wpisywana jest liczba 41 wskazująca, że pakiet IPv6 jest ładunkiem tego pakietu. Różnice dotyczą wykorzystania adresów. Interfejs, który jest początkiem tunelu, powinien używać adresu IPv4 rutowalnego w sieci IPv4[5], przez którą będą przekazywane tunelowane pakiety. Adres początku tunelu składa się z prefiksu 2002/16[6] i zapisanego w postaci szesnastkowej adresu IPv4 początku tunelu. Sposób tworzenia adresu początku tunelu jest pokazany na rys. 10 (pole **`ssss`** określa numer podsieci, pole **`hhhh`** określa identyfikator hosta – o wartościach tych pól decyduje administrator).

[5] Zgodnie z RFC 3056 [2] adres interfejsu nie powinien być adresem prywatnym.

[6] Organizacja IANA (Internet Assigned Numbers Authority) przeznaczyła prefiks 2002/16 do wykorzystania przy tworzeniu tuneli automatycznych.



Rys. 10. Procedura tworzenia adresu początku tunelu

Konfiguracja tego trybu tunelowania wymaga skonfigurowania początku tunelu i określenia w tablicy rutingu trasy do drugiego końca tunelu. Ten typ tunelu ma charakter wiele-do-jednego, to znaczy, że wiele tuneli ma różne początki, ale jeden koniec, co może powodować przeciążane tego węzła sieci. Na trasie tunelu nie można używać usługi NAT.

Tunelowanie Teredo

Tunelowanie Teredo jest jedynym mechanizmem umożliwiającym integrację sieci IPv4 i IPv6 w przypadku używania w sieciach IPv4 usługi NAT. Mechanizm ten umożliwia wykorzystanie infrastruktury IPv4 do komunikacji w następujących sytuacjach:

- hosty sieci tylko-IPv4 umieszczone za mechanizmem NAT komunikują się z hostami sieci tylko-IPv6,
- hosty sieci tylko-IPv6 transferują dane do hostów sieci tylko-IPv4 ulokowanych za mechanizmem NAT,
- wymiana danych pomiędzy hostami sieci tylko-IPv4 ulokowanymi za mechanizmem NAT w tej samej i różnych lokalizacjach[7].

Podstawowymi elementami mechanizmu są: serwer Teredo (*ang. Teredo Server*), przekaźnik Teredo (*ang. Teredo Relay*) i klient Teredo (*ang. Teredo Client*). Na rys. 11 pokazana jest przykładowa topologia obejmująca elementy mechanizmu Teredo.

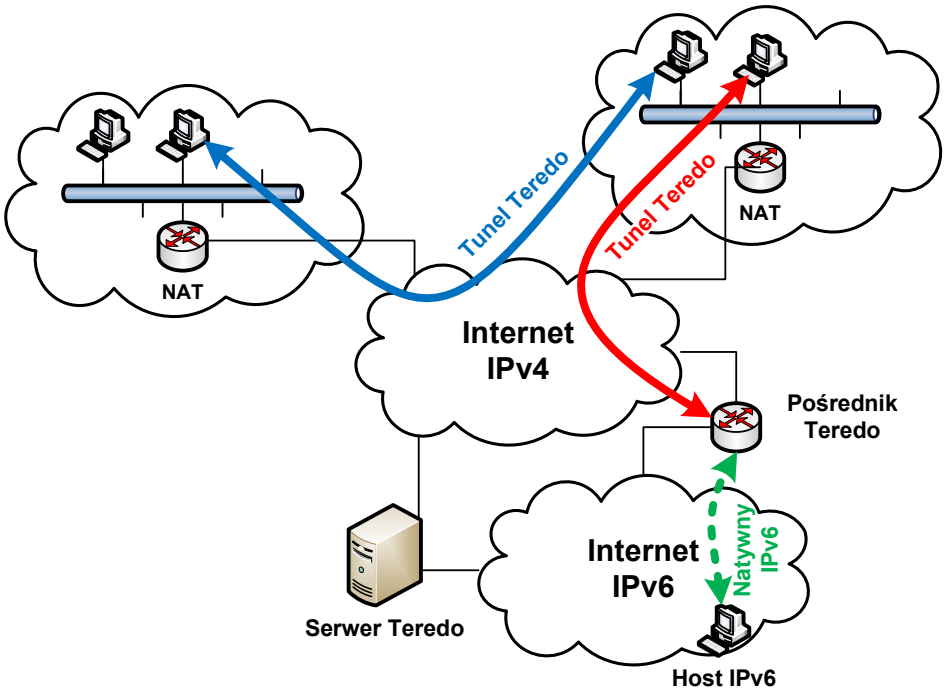
Klientem tego tunelu może być dowolny host posiadający dostęp do Internetu IPv4, dla którego osiągalny jest serwer i przekaźnik Teredo.

Przekaźnik Teredo pełni rolę routera. Jedynym jego zadaniem jest przekazywanie pakietów między sieciami IPv4 i IPv6 (tzn. dokonuje kapsulacji i dekapsulacji). Przekaźnik Teredo musi mieć skonfigurowany dostęp do sieci IPv4 i IPv6 oraz uruchomiony proces rutowania pakietów.

Serwer tunelu Teredo ma dostęp do sieci IPv4 i IPv6. Interfejs sieci IPv4 musi mieć skonfigurowane dwa adresy publiczne, a interfejs sieci IPv6 musi

[7] Z punktu widzenia integracji sieci IPv4 i IPv6 ten sposób wymiany danych nie jest interesujący, ale został tutaj podany jako jedna z cech charakterystycznych tunelowania Teredo.

mieć skonfigurowany zagregowany globalny adres jednostkowy. Głównym zadaniem serwera jest utrzymywanie listy aktywnych połączeń z klientami Teredo i wspomaganie klienta tunelu w wyznaczaniu adresu IPv6 wykorzystywanego w komunikacji przez tunel.



Rys. 11. Schemat sieci z komponentami mechanizmu Teredo

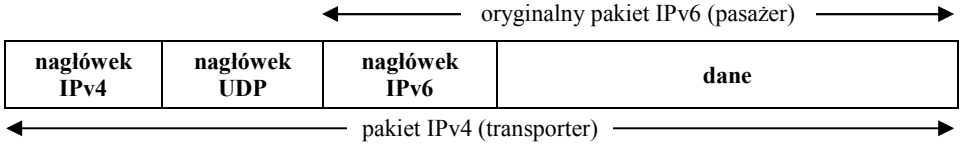
Struktura adresu IPv6 używana w tunelu Teredo jest następująca [15]:

0 – 31	32 – 63	64 – 79	80 – 95	96 – 127
prefiks Teredo	**adres serwera**	**flagi**	**port klienta**	**adres klienta**

- prefiks Teredo – 2001:0000::/32;
- adres IPv4 serwera Teredo w postaci szesnastkowej;
- flagi określające rodzaj serwera NAT, za którym zlokalizowany jest klient Teredo[8];
- określenie portu klienta Teredo (nr portu XOR 0xFFFF);
- określenie adresu IPv4 klienta Teredo (adres XOR 0xFFFFFFFF).

[8] Występują cztery rodzaje implementacji usługi NAT[17][25]: *Full Cone-NAT*, *Restricted Cone NAT*, *Port Restricted Cone NAT* i *Symmetric NAT*.

W komunikacji hostów IPv4 i IPv6 przy inicjowaniu połączenia zawsze uczestniczy serwer Teredo. Sposób wymiany pakietów[9] przy inicjowaniu połączenia jest uzależniony od rodzaju wykorzystywanej usługi NAT, za którą umiejscowiony jest host IPv4. Po zestawieniu połączenia w wymianie pakietów uczestniczy pośrednik Teredo. Serwer tunelu Teredo może pełnić rolę przekaźnika Teredo. Najczęściej używany sposób kapsułkowania pakietów IPv6 jest pokazany na rys. 12. Występują również przypadki, w których pomiędzy nagłówkiem UDP a nagłówkiem IPv6 serwer Teredo wstawia pole wskazujące pochodzenie pakietu, pole zawierające dane uwierzytelniające końce tunelu albo oba te pola.



Rys. 12. Kapsułkowanie pakietów IPv6 w pakietach IPv4 w tunelu Teredo

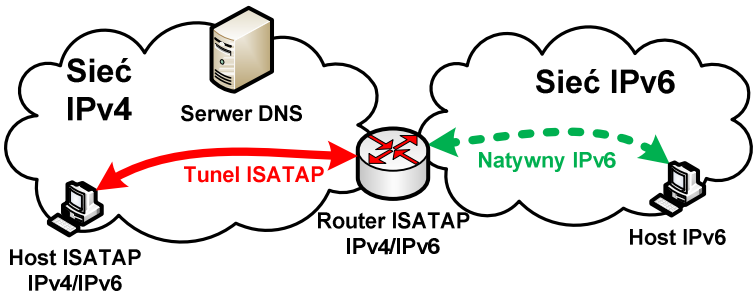
Jeżeli w sieci dostępny jest serwer Teredo[10], to konfigurowanie klienta Teredo wymaga wydania na takim komputerze jednej komendy definiującej adres serwera Teredo.

W związku z tym, że nawiązywanie i utrzymywanie połączeń w tunelu Teredo wymaga wymiany wielu pakietów z serwerem Teredo, tego typu tunelowanie powinno być używane tylko wtedy, gdy host IPv4 nie może korzystać z publicznego adresu IPv4 (to znaczy jest zmuszony do korzystania z usługi NAT). W innych sytuacjach tunelowanie Teredo nie jest zalecanym sposobem komunikacji hostów IPv4 z hostami IPv6.

Tunelowanie ISATAP

Tunelowanie ISATAP może być używane wewnątrz domeny administracyjnej, w której wykorzystywany jest protokół IPv4. Za domenę administracyjną przyjmuje się wycinek sieci IPv4, w której nie jest używana usługa NAT. W takiej domenie musi być skonfigurowany przynajmniej jeden router ISATAP, który będzie jednym z końców wykorzystywanych tuneli i który będzie dokonywał kapsulacji/dekapsulacji pakietów na granicy sieci IPv4 i IPv6. Tunele są tworzone pomiędzy hostami ISATAP a routerem ISATAP. Router ISATAP i hosty ISATAP muszą obsługiwać podwójny stos IPv4/IPv6. Schemat sieci z pokazanymi komponentami mechanizmu ISATAP jest pokazany na rys. 13.

[9] Szczegóły sposobu wymiany pakietów można znaleźć w [20].
[10] Do tego celu można użyć serwer dostępny pod adresem teredo.ipv6.microsoft.com.



Rys. 13. Schemat sieci z komponentami mechanizmu ISATAP

Dla każdego skonfigurowanego interfejsu IPv4 na węźle ISATAP generowany jest 64-bitowy identyfikator ISATAP interfejsu według podanej niżej zasady:

0 … 15	16 … 31	32 … 63
000000ug00000000	0x5EFE	adres IPv4

gdzie: u = 1 – gdy adres IPv4 jest adresem publicznym;
= 0 – w przeciwnym razie;

g = 1 – gdy adres IPv4 jest adresem grupowym;
= 0 – gdy adres IPv4 jest adresem pojedynczym.

Interfejsowi hosta i routera ISATAP w trybie autokonfiguracji nadawany jest adres lokalny łącza wykorzystując opisany wyżej identyfikator ISATAP interfejsu. Na przykład jeżeli interfejsowi zostanie nadany adres IPv4 (ręcznie lub za pośrednictwem usługi DHCPv4), to ten interfejs również otrzyma adres lokalny łącza FE80::5EFE:200.200.200.200 (lub w innej postaci FE80::5EFE:C8C8:C8C8). Klient ISATAP musi znać przynajmniej jeden adres IPv4 routera ISATAP. Adres ten może być skonfigurowany ręcznie albo pozyskany z odpowiednio skonfigurowanego serwera DHCPv4 lub DNS (na przykład w plikach strefowych konfiguracji DNS należy podać odwzorowanie *isatap.example.com* na adres routera ISATAP).

Host ISATAP, gdy zna adres IPv4 routera ISATAP i ma skonfigurowany swój adres lokalny łącza, jest w stanie wymieniać zakapsułkowane pakiety IPv6 (zawierające komunikaty ICMPv6) z routerem ISATAP w celu ustalenia adresów sąsiadów IPv6 i pozyskania prefiksu wykorzystywanej sieci IPv6. Na podstawie tego prefiksu i swojego identyfikatora ISATAP host wyznaczy dla swojego interfejsu zagregowany adres globalny, który będzie wykorzystywany w tunelowanych pakietach przeznaczonych do hostów w sieci IPv6.

Sposób kapsułkowania pakietów IPv6 w pakietach IPv4 w trybie ISATAP jest identyczny jak w trybie *Manual Mode* (rys. 6). W polu „protokół” nagłówka IPv4 wpisywana jest liczba 41 wskazująca, że pakiet IPv6 jest ładunkiem tego pakietu.

2.2.3. Mechanizmy wspomagające konfigurowanie tuneli

Opisane wcześniej mechanizmy tunelowania należą do tuneli konfigurowanych manualnie albo automatycznie. Przygotowanie tuneli konfigurowanych manualnie wymaga zabiegów konfiguracyjnych na wszystkich urządzeniach, które są końcami tuneli. Czynność ta jest jednorazowa, a tworzenie tunelu nie generuje żadnego dodatkowego ruchu w sieci. Pomimo tego tunele manualne są rozwiązaniem bardzo słabo skalowalnym – problemy pojawiają się wtedy, gdy takich tuneli jest wiele. Tunele automatyczne są łatwiejsze w konfiguracji dla użytkownika danego tunelu – wymagają konfiguracji tylko po jednej stronie tunelu. Jednakże samej konfiguracji początku tunelu musi towarzyszyć albo konieczność uruchomienia rutowania dynamicznego (*Automatic IPv4 Compatible Mode*), albo tunele mają charakter wiele-do-jednego (*Automatic Mode*), albo wymagają utrzymywania dodatkowych serwerów i przy zestawianiu tunelu generują dodatkowy ruch sieciowy (Teredo i ISATAP). W celu zmniejszenia opisanych niedostatków zaproponowano rozwiązania ułatwiające zestawianie tuneli konfigurowanych manualnie. Do tych rozwiązań można zaliczyć:

- broker tuneli (*ang. Tunnel Broker*) [8];
- serwer tuneli.

Broker tuneli

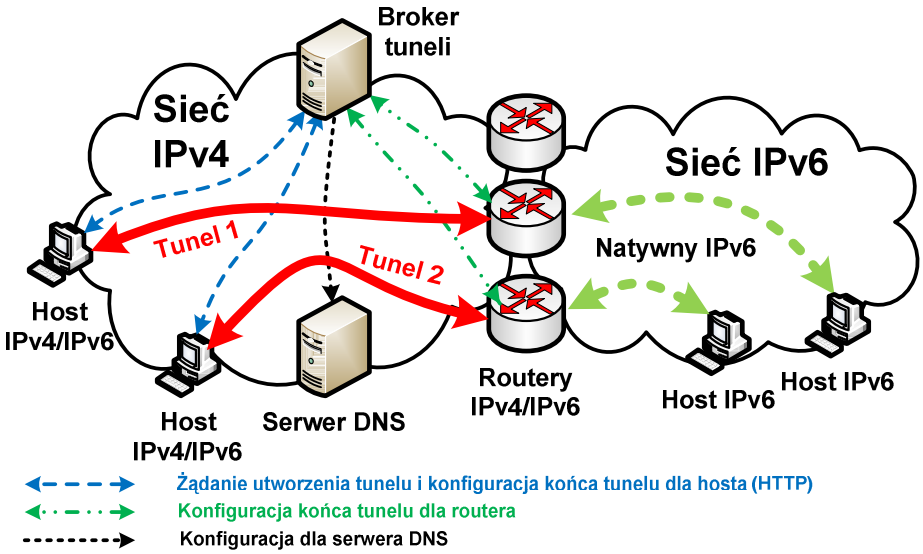
Broker tuneli jest rozwiązaniem przeznaczonym dla małych izolowanych sieci IPv6 i pojedynczych hostów z sieci IPv4, które wymagają komunikacji z hostami istniejących sieci IPv6. Broker tuneli jest postrzegany przez hosty pracujące w sieci IPv4 (klienci IPv4) jako wirtualny dostawca dostępu do sieci IPv6 (*ang. IPv6 ISP*). Schemat działania brokera tuneli jest pokazany na rys. 14.

Broker tuneli zwykle jest serwerem usługi WWW i musi być dostępny dla klientów IPv4 brokera. Klienci brokera muszą obsługiwać podwójny stos. Klient po nawiązaniu połączenia z brokerem tuneli i uwierzytelnieniu przekazuje brokerowi tuneli następujące dane:

- swój adres IPv4;
- nazwę, pod którą ma być dostępny w usłudze DNS adres IPv6 przydzielony końcowi tunelu po stronie klienta IPv4;
- funkcję klienta (host czy router).

Następnie broker tuneli wybiera router, który będzie końcem tunelu (przy wyborze routera może kierować się wyrównywaniem obciążenia routerów) i wymusza na tym routerze konfigurację jednego końca tunelu. Na serwerze DNS rejestruje nazwy i adresy przydzielonych końców tunelu. Do klienta IPv4 przekazuje czas życia tunelu i przydzielony klientowi prefiks IPv6 (zwykle

o długości 48 bitów dla lokalizacji albo 64 bitów dla podsieci, albo 128 bitów dla hosta). Zależnie od implementacji brokera tuneli do klienta IPv4 mogą być wysyłane tylko dane do konfiguracji jednego końca tunelu albo na przykład gotowy skrypt do takiej konfiguracji. Od tego momentu klient może kontaktować się z docelowymi węzłami IPv6 bez udziału brokera tuneli.

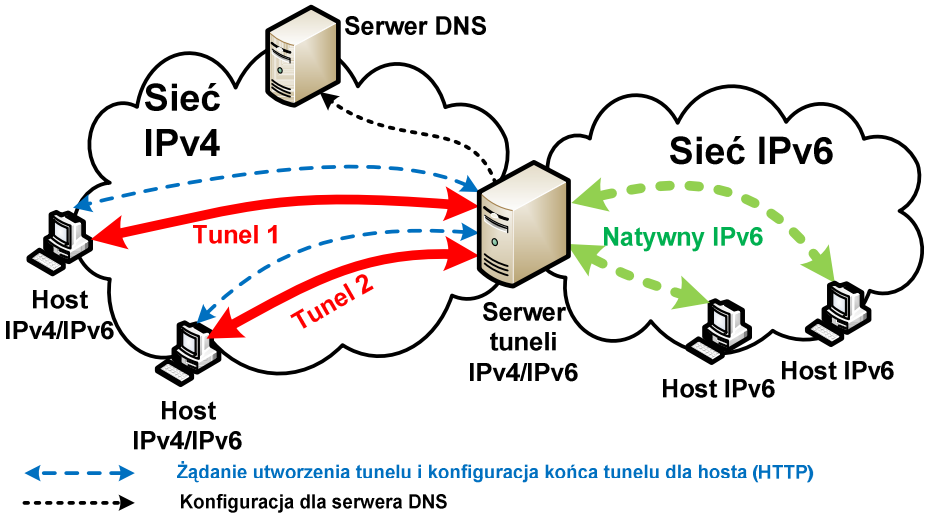


Rys. 14. Schemat działania sieci z brokerem tuneli

Broker tuneli może obsługiwać różne rodzaje tuneli. Ze względu na ograniczenia tuneli broker tuneli nie obsługuje węzłów umiejscowionych za mechanizmem NAT.

Serwer tuneli

Serwer tuneli jest rozwiązaniem, w którym funkcje brokera tuneli i routera będącego drugim końcem tworzonych tuneli są skupione na jednym komputerze. Serwer tuneli musi obsługiwać podwójny stos, musi mieć dostęp do sieci IPv4 i IPv6. Ponadto powinien oferować usługę do współpracy z klientami serwera i zarządzania tunelami. Procedury tworzenia nowego tunelu i zarządzania istniejącymi tunelami są podobne do procedur działania brokera tuneli z wyłączeniem części odpowiedzialnych za współpracę brokera z routerami. Serwer tunelu, z tego względu, że jest jednym urządzeniem, nie ma możliwości równoważenia obciążenia, które było charakterystycznym zachowaniem brokera tuneli. Schemat funkcjonowania serwera tuneli jest pokazany na rys. 15.

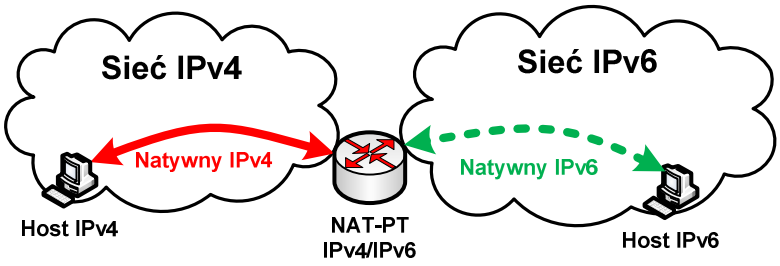


Rys. 15. Schemat działania sieci z serwerem tuneli

2.3. Mechanizm translacji protokołów

Mechanizm translacji protokołów umożliwia integrację sieci opartych wyłącznie na protokole IPv6 z sieciami wykorzystującymi jedynie IPv4. Urządzenie, które dokonuje translacji protokołów, musi obsługiwać podwójny stos IPv4/IPv6.

Jednym z rozwiązań oferujących translację protokołów jest NAT-PT (*ang. Network Address Translation – Protocol Translation*) – przykład sieci wykorzystującej NAT-PT jest pokazany na rys. 16.



Rys. 16. Schemat działania sieci z serwerem NAT-PT

Implementacje mechanizmów translacji protokołów można sklasyfikować następująco:

1. Statyczny NAT-PT (mapowanie adresów jeden–do–jednego pomiędzy IPv4 i IPv6 – każdy host w jednej sieci, np. IPv4 jest zawsze widziany pod tym samym adresem np. IPv6). Każde odwzorowanie musi zostać uprzednio zdefiniowane na routerze NAT-PT.
2. Dynamiczny NAT-PT (mapowanie adresów źródłowych na adresy z puli adresów zewnętrznych). Liczba możliwych do nawiązania połączeń pomiędzy sieciami zależna jest od liczności puli adresów, na które dokonywana jest translacja. Funkcjonowanie mechanizmu dynamicznego NAT-PT musi być wspomagane serwerem DNS, który musi być umiejscowiony po stronie, do której inicjowana jest translacja.
3. NAPT-PT (*ang. Network Address Port Translator - Protocol Translator*) – mapowanie wiele-do-jednego pomiędzy adresami IPv6 i IPv4, które w translacji wykorzystuje numery portów protokołów warstwy transportowej.

Mechanizm NAT-PT używa do translacji adresów IPv6 z prefiksem o długości 96 bitów. Pozostała część adresu jest uzupełniana mapowanym adresem IPv4.

Mechanizm NAT-PT był długo postrzegany jako najodpowiedniejszy wybór umożliwiający koegzystencję sieci IPv4 i IPv6, jednak ten mechanizm ma szereg wad, szczególnie translacja każdego pakietu bardzo obciąża zasoby routera NAT-PT, co powoduje spadek przepustowości łącza do kilku procent w stosunku do łącza bez translacji protokołów. Szczegółowy opis niedostatków tego mechanizmu można znaleźć w dokumencie RFC 4966 [1].

3. Zakończenie

W artykule przedstawiono aktualnie najczęściej stosowane rozwiązania umożliwiające integrację sieci IPv4 i IPv6 (nie wspomniano tutaj o kilku rzadziej stosowanych mechanizmach, na przykład o bramach poziomu aplikacji czy mechanizmie SIIT (*ang. Stateless IP/ICMP Translation*).

Z mechanizmami przeznaczonymi do integracji sieci IPv4 i IPv6 ściśle związane są zagadnienia dotyczące: wykorzystania adresacji grupowej (*ang. multicast*), zabezpieczania transmisji (IPSec), mechanizmów zapewnienia jakości transmisji (*ang. Quality of Service*) i rutowania dynamicznego. Każde z tych zagadnień jest bardzo obszerne i wymaga osobnego omówienia w kontekście mechanizmów integracji sieci IPv4 i IPv6.

Literatura

[1] AOUN C., DAVIES E., *Reasons to Move the Network Address Translator – Protocol Translator (NAT-PT) to Historic Status*, RFC 4966, July 2007.

[2] BRADEN R., POSTEL J., *Requirements for Internet Gateways*, RFC 1009, June 1987.

[3] CARPENTER B., MOORE K., *Connection of IPv6 Domains via IPv4 Clouds*, RFC 3056, February 2001.

[4] CONTA A., DEERING S., *Generic Packet Tunneling in IPv6 Specification*, RFC 2473, December 1998.

[5] DEERING S., HINDEN R, *Internet Protocol, Version 6 (IPv6) Specification*, RFC 2460, December 1998.

[6] DESMEULES R, *IPv6: Sieci oparte na protokole IP w wersji 6*, PWN, 2006.

[7] DOMMETY G., *Key and Sequence Number Extensions to GRE*, RFC 2890, September 2000.

[8] DURAND A., FASANO P., GUARDINI I., LENTO D., *IPv6 Tunnel Broker*, RFC 3053, January 2001.

[9] EGEVANG K., FRANCIS P., *The IP Network Address Translator (NAT)*, RFC 1631, May 1994.

[10] FARINACCI D., LI T., HANKS S., MEYER D., TRAINA P., *Generic Routing Encapsulation (GRE)*, RFC 2784, March 2000.

[11] GILLIGAN R. i inni, *Basic Socket Interface Extensions for IPv6*, RFC 3493, February 2003.

[12] GILLIGAN R., NORDMARK E., *Transition Mechanisms for IPv6 Hosts and Routers*, RFC 2893, August 2000.

[13] HINDEN R., *Applicability Statement for the Implementation of Classless Inter-Domain Routing (CIDR)*, RFC 1517, September 1993.

[14] HINDEN R., DEERING S., *IP Version 6 Addressing Architecture*, RFC 4291, February 2006.

[15] HUITEMA C., *Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)*, RFC 4380, February 2006.

[16] HUITEMA C., AUSTEIN R., SATAPATI S., VAN DER POL R., *Evaluation of IPv6 Transition Mechanisms for Unmanaged Networks*, RFC 3904, September 2004.

[17] HUSTON G.*, Anatomy: A Look Inside Network Address Translation*, The Internet Protocol Journal, Volume 7, Number 3, http://www.cisco.com/web/about/ac123 /ac147/archived_issues/ipj_7-3/anatomy.html.

[18] KITAMURA H., *A SOCKS-based IPv6/IPv4 Gateway Mechanism*, RFC 3089, April 2001.

[19] LEE S. i inni, *Dual Stack Hosts Using "Bump-in-the-API" (BIA)*, RFC 3338, October 2002.

[20] Microsoft, *Teredo Overview*, Microsoft TechNet, January 15, 2007, http://technet.microsoft.com/en-us/library/bb457011.aspx

[21] MOGUL J., POSTEL J., *Internet Standard Subnetting Procedure*, RFC 950, August 1985.

[22] NORDMARK E., GILLIGAN R., *Basic Transition Mechanisms for IPv6 Hosts and Routers*, RFC 4213, October 2005.

[23] REKHTER Y. i inni, *Address Allocation for Private Internets*, RFC 1918, February 1996.

[24] REKHTER Y., LI T., *An Architecture for IP Address Allocation with CIDR*, RFC 1518, September 1993.

[25] ROSENBERG J., WEINBERGER J., HUITEMA C., MAHY R., *STUN – Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*, RFC 3489, March 2003.

[26] SRISURESH P., HOLDREGE M., *IP Network Address Translator (NAT) Terminology and Considerations*, RFC 2663, August 1999.

[27] STEVENS W. i inni, *Advanced Sockets Application Program Interface (API) for IPv6*, RFC 3542, May 2003.

[28] TEMPLIN F., GLEESON T., THALER D., *Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)*, RFC 5214, March 2008.

[29] THALER D., KRISHNAN S., HOAGLAND J., *Teredo Security Updates*, RFC 5991, September 2010.

[30] TSIRTSIS G. SRISURESH P., *Network Address Translation – Protocol Translation (NAT-PT)*, RFC 2766, February 2000.

[31] TSUCHIYA K., HIGUCHI H., ATARASHI Y., *Dual Stack Hosts using the "Bump-In-the-Stack" Technique (BIS)*, RFC 2767, February 2000.

Methods of IPv4 and IPv6 networks integration

ABSTRACT: This paper considers methods of IPv4 and IPv6 networks integration. The dual stack and protocol translation technology used to a IPv4-only hosts and IPv6-only hosts communication is described. Tunneling mechanisms for data exchange between "IPv6 islands" through an IPv4 environment and between "IPv4 islands" through an IPv6 environment are taken into consideration. Operating rules and properties for manual tunnels (*Manual Mode*, GRE and *Generic Packet Tunneling*) and automatic tunnels (*Automatic IPv4 Compatible Mode*, *Automatic Mode*, *Teredo* and ISATAP), including tunnels configuration aiding tools (*tunnel broker* and *tunnel server*) are described.

KEYWORDS: IPv6 networks, IPv4 and IPv6 networks integration, protocol tunneling

Praca wpłynęła do redakcji: 28.12.2010