Protokół internetowy IPv6

Autor: Tomasz Czeluśniak IVFDS

STRESZCZENIE

Opracowanie to poświęcone jest protokołowi internetowemu IPv6. Przedstawia wymagane oprogramowanie niezbędne przy podłączeniu do sieci 6BONE oraz sposoby konfiguracji sprzętowej. W przeważającej części uwzględnia system operacyjny LINUX. Dokument zawiera także wskazówki, gdzie szukać dodatkowych informacji oraz niezbędnego oprogramowania.

SPIS TREŚCI

1. WSTĘP

Najistotniejszym problemem Internetu jest aktualnie wyczerpujący się zasób wolnych adresów IP. Problem ten jest zaledwie częściowo rozwiązywany poprzez stosowanie translacji adresów (*NAT*).Globalnym rozwiązaniem ma być obecnie rozwijana nowa wersja protokołu internetowego - **IPv6** (znanego również jako **IPng – IP Next Generation**). IPv6, poza rozwiązaniem problemu braku adresów, wprowadza wiele udogodnień i ulepszeń, omówionych w dalszej części projektu. Obecnie dostępnych jest kilkanaście implementacji IPv6 (są to implementacje m.in. dla Linuxa, *BSD/KAME, Solarisa oraz Windows 9x/NT). Poniższy artykuł uwzględnia przede wszystkim Linuxa.

2. PODSTAWY TEORETYCZNE

Adresy IPv6 składają się z **128 bitów** (natomiast adresy IPv4 składają się tylko z 32 bitów). Łatwo jest sprawdzić, że liczba wszystkich adresów IPv6 to liczba 39 cyfrowa, wyznaczona przez czynnik 2^{96} (dla IPv4 tylko 10 cyfrowa)! Przykładowy adres IPv6 wygląda następująco: **3ffe:803:101::/48** (adres sieci). Nie podane bity są domyślnie równe „0” (np. „::” == „:0000:”). Powyższy przykładowy adres podany z wykorzystaniem wszystkich bitów wyglądał będzie tak: **3ffe:0803:0101:0000:0000:0000:0000:0000/48**, gdzie „/48” to długość prefiksu (prefiks jest innym sposobem przedstawienia netmaski) w bitach. Taki zapis zgodny jest ze **specyfikacją CIDR.**

W adresach IPv6 zasięg (*scope*) adresu definiowany jest przez początkowe bity adresu. Adresy rozpoczynające się od *fe80*: to adresy „*link-local*” - zasięg „*local*”, czyli adresy lokalne. Poza zasięgiem *local* istnieją także: *host*, *site*, *global*. [2]. Należy zaznaczyć, że adresy z zasięgiem local są widoczne wyłącznie w obrębie sieci, do których podpięliśmy Linuxa oraz do serwerów, z którymi nasz Linux ma połączenie (bądź bezpośrednie bądź przy pomocy tunelu).[1]

Główną zaletą IPv6 jest możliwość **autokonfiguracji.** Funkcje protokołu autokonfiguracji, niezależnie od przynależności państwowej (Stateless Autoconfiguration Protocol) umożliwią włączenie komputera na zasadzie „plug-and-play”. Protokół autokonfiguracji zapewnia środki, by komputer włączony do dowolnej sieci sam sobie przydzielił adres Ipv6, który w części oparty jest na jego karcie sieciowej. Ponieważ adres karty jest unikatowy, to przydzielony sobie adres Ipv6 też będzie unikatowy, co zapobiegnie dublowaniu się adresów.
Hosty IPv6 wykorzystują między innymi protokół Neighbor Discovery (ND) pozwalający im znaleźć sąsiadujące routery i inne hosty. Dzięki ND serwery mogą śledzić, które routery lub serwery są aktywne i osiągalne, a następnie modyfikować swoje tablice routinguIstnieją dwie metody takiej konfiguracji:

- **stateful** - hosty uzyskują wszelkie potrzebne informacje z serwera, który zawiera odpowiednią bazę danych. Metoda ta wykorzystuje DHCPv6.[1]

- **stateless** - nie wymaga żadnego konfigurowania hosta i wymaga minimalnej konfiguracji routerów. Metoda ta pozwala hostom na wygenerowanie własnego adresu na podstawie lokalnie dostępnych informacji i informacji rozgłaszanych przez routery. Routery w tym przypadku rozgłaszają tylko prefiks sieci. Otrzymany od routera prefiks jest następnie uwzględniany podczas generowania adresów lokalnych interfejsów. Jeśli router z jakiegoś powodu nie rozgłasza odpowiednich informacji, host może wygenerować automatycznie tylko adresy *link-local, co* pozwala na ograniczoną komunikację wyznaczoną zasięgiem (scope) local.[1]

Należy zaznaczyć, że hosty mogą wykorzystywać równocześnie obie metody do autokonfiguracji. Mechanizm obsługi IPv6 pozwala także na tworzenie dynamicznych tuneli dla

pakietów IPv6 w istniejącej infrastrukturze IPv4, pod warunkiem, że adres źródłowy i docelowy pakietu to adres kompatybilny z IPv4.

Wyróżnia się trzy rodzaje adresów protokołu Ipv6:

- Unicast – zapewnia komunikację typu punkt-punkt (point-to-point).
- Anycast – pozwala komunikować się z najbliższym urządzeniem z grupy urządzeń.
- Multicast – pozwala komunikować się z wieloma urządzeniami z grupy urządzeń.

Adresowanie unicast zapewnia przyłączalność od jednego urządzenia końcowego do drugiego. Protokół Ipv6 obsługuje kilka odmian adresów unicast.

Adres dostawcy usług internetowych (ISP)

Podczas gdy protokół Ipv4 z góry przyjął grupy użytkowników wymagających przyłączalności, Ipv6 dostarcza format adresu unicast, specjalnie przeznaczony dla dostawców usług internetowych, w celu przyłączania indywidualnych użytkowników do Internetu. Te oparte na dostawcach adresu unicast oferują unikatowe adresy dla indywidualnych użytkowników lub małych grup, uzyskujących dostęp do Internetu za pośrednictwem dostawcy usług internetowych. Architektura adresu zapewnia wydajną agregację tras w środowisku użytkowników indywidualnych.

Format adresu unicast ISP jest następujący:

- 3-bitowa flaga adresu unicast ISP, zawsze ustawiona na”010”
- Pole ID rejestru, o długości „n” bitów
- Pole ID dostawcy, o długości „m” bitów
- Pole ID abonenta, o długości „o” bitów
- Pole ID podsieci, o długości „p” bitów
- Pole ID interfejsu, o długości 128-3-(n + m + o + p) bitów

Litery n, m, p, o oznaczają zmienne długości pól. Długość pola ID interfejsu stanowi różnicę długości adresu (128) i łącznej długości pól poprzedzających, wraz z trójbitową flagą.[11]

Przykładem adresu tego typu może być 010:0:0:0:0:x, gdzie „x” może być dowolną liczbą. Ponieważ większość nowej przestrzeni adresowej dopiero musi zostać przypisana, adresy te będą zawierać mnóstwo zer. Dlatego grupy zer mogą być zapisywane skrótem w postaci podwójnego dwukropka (::)- skróconą formą adresu 010:0:0:0:0:x jest więc 010::x.

Inne rodzaje adresów unicast są przeznaczone do użytku lokalnego. Adresy użytku lokalnego mogą być przypisane do urządzeń sieciowych w samodzielnym Intranecie lub do urządzeń w Intranecie, którym potrzebny jest dostęp do Internetu.

Adres użytku lokalnego dla łącza

Adres użytku lokalnego dla łącza jest przeznaczony dla pojedynczego łącza, do celów takich jak konfiguracja auto-adresu, wykrywanie sąsiadów, a także w przypadku braku routerów. Adresy lokalne dla łącza mają następujący format:

- 10-bitowa flaga adresu lokalnego, zawsze ustawiona na ;1111111011”
- Zarezerwowane, nienazwane pole, mające długość „n” bitów, ale ustawione domyślnie na wartość „0”
- Pole ID interfejsu o długości 118 – n bitów

ID interfejsu może być adresem MAC karty sieciowej Ethernetu. Adresy MAC, będące teoretycznie adresami unikalnymi, mogą być skojarzone z przedrostkami standardowego adresu IP w celu utworzenia unikalnych adresów dla mobilnych lub zastępczych użytkowników. Przykładem adresu użytku lokalnego dla łącza z adresem MAC mógłby być 1111111011:0:adres_mac.[11]

Adres użytku lokalnego dla miejsca
Adresy lokalne dla miejsca są przeznaczone do stosowania w pojedynczym miejscu. Mogą być używane w miejscach lub organizacjach, które nie są przyłączone do globalnego Internetu. Nie muszą żądać czy też „kraść" przedrostka adresu z przestrzeni adresowej globalnego Internetu. Zamiast tego mogą używać adresów protokołu Ipv6 lokalnych dla miejsca. Gdy organizacja łączy się z globalnym Internetem, może utworzyć unikatowe adresy globalne, zastępując przedrostek lokalny dla miejsca przedrostkiem abonenta, zawierającym identyfikatory rejestru, dostawcy i abonenta.
Adresy lokalne dla miejsca mają następujący format:

- 10-bitowa flaga użytku lokalnego, zawsze ustawiona na „1111111011"
- Zarezerwowane, nienazwane pole, mające długość „n" bitów, ale ustawione domyślnie na wartość „0"
- Pole ID podsieci o długości „m" bitów
- Pole ID interfejsu o długości 118 – (n + m) bitów

Przykładem adresu lokalnego dla miejsca jest: 1111111011:podsiec:interfejs.[11]

Struktury zastępczych adresów unicast Ipv6
Dwa specjalne adresy unicast protokołu Ipv6 zostały określone jako mechanizmy przejściowe, umożliwiające hostom i routerom dynamiczne trasowanie pakietów Ipv6 przez infrastrukturę sieci protokołu Ipv4 i na odwrót.

- Standardowe (zgodny z Ipv4)

80 bitów	16 bitów	32 bity
0000.....0000	0000	Adres IPv4

- Wzorowany na IPv4

80 bitów	16 bitów	32 bity
0000.....0000	FFFF	Adres Ipv4

Struktury adresów anycast Ipv6
Adres anycast, wprowadzony w protokole Ipv6, jest pojedynczą wartością przypisaną do więcej niż jednego interfejsu. Zwykle interfejsy te należą do różnych urządzeń. Pakiet wysłany pod adres anycast jest trasowany tylko do jednego urządzenia. Jest on wysyłany do najbliższego- według zdefiniowanej przez protokoły trasujące miary odległości- interfejsu o tym adresie. Na przykład, strona WWW może być powielona na kilku serwerach. Dzięki przypisaniu tym serwerom adresu anycast żądania połączenia z tą stroną WWW są automatycznie trasowane do tylko jednego serwera- najbliższego względem użytkownika.
Adresy anycast są tworzone (pobierane) z przestrzeni adresów unicast i mogą przybrać formę dowolnego typu adresu unicast. Tworzy się je, przypisując po prostu ten sam adres unicast więcej niż jednemu interfejsowi.[11]

Struktury adresów multicast Ipv6
Protokół IPv4 obsługiwał multicasting, ale wymagało to niejasnego adresowania klasy D. Protokół Ipv6 rezygnuje z adresów klasy D na korzyść nowego formatu adresu, udostępniającego tryliony możliwych kodów grup multicast. Każdy kod grupy identyfikuje dwóch lub więcej odbiorców pakietu. Zakres pojedynczego adresu multicast jest elastyczny. Każdy adres może

być ograniczony do pojedynczego systemu, do określonego miejsca, powiązany z danym łączem sieciowym lub rozpowszechniany globalnie. Nadawanie adresów IP również zostało wyeliminowane i zastąpione nowym multicastowym formatem adresu.

Do pozostałych zalet IPv6 należy zaliczyć także zmianę formatu nagłówka pakietów na nowy, pozwalający bez większych problemów dodawać w przyszłości nowe opcje bez poważnych zmian w samym nagłówku. IPv6 umożliwia także na wysyłanie datagramów zwanych **jumbogramami** o wielkości większej niż 65535 bajtów.
By móc wykorzystać IPv6 w obrębie dzisiejszego Internetu wykorzystującego nadal protokół IPv4 stosuje się **SIT** (*Simple Internet Transition*) do tunelowania pakietów IPv6 wewnątrz pakietów IPv4.
Istnieje ogólnoświatowa, wirtualna sieć bazująca na protokole IPv6. Jest to sieć **6BONE.** Wirtualna, dlatego, że bazuje nie na własnych, oddzielnych łączach, lecz wykorzystuje istniejące łącza Internetu.
Struktura sieci składa się z głównych węzłów - **pTLA** (*pseudo Top Level Aggregator*), węzłów podrzędnych - **pNLA** (*pseudo Next Level Aggregator*) oraz podpiętych do nich pozostałych hostów (**leaf sites**). W Polsce jedynym na dzień dzisiejszy pTLA jest **ICM** (*Interdyscyplinarne Centrum Modelowania Matematycznego i Komputerowego w Warszawie*), a osobą zajmującą się siecią 6bone w ICM jest **Rafał Maszkowski** <*rzm@icm.edu.pl*>.[1]

3. NIEZBĘDNE OPROGRAMOWANIE

- **Stabilna wersja jądra LINUXA**

Należy zaopatrzyć się w stabilną wersję Linuxa. Jądro należy skompilować z aktywnymi następującymi opcjami:

[*] Prompt for development and/or incomplete code/drivers
[*] Kernel/User netlink socket
<M> IP: tunneling
<M> The IPv6 protocol (EXPERIMENTAL)
[*] IPv6: enable EUI-64 token format
[*] IPv6: disable provider based addresses

Powyższe opcje można bądź wkompilować w jądro jak i pozostawić w postaci ładowalnych modułów.

- **Odpowiednia biblioteka glibc**

Do kompilacji programów wykorzystujących IPv6 niezbędna jest biblioteka z nowymi funkcjami [9]. Użytkownicy glibc 2.1.1 (i nowszych) nie będą mieli żadnych problemów, gdyż ich biblioteka zawiera niemal wszystkie potrzebne funkcje. Posiadacze biblioteki libc5 mogą skorzystać z „protezy", jaką jest biblioteka libinet6 zawarta w pakiecie inet6-apps autorstwa Craiga Metza. Dobrym rozwiązaniem jest jednak aktualizacja do najnowszej, stabilnej wersji glibc ze względu na znaczne ułatwienie przy późniejszych kompilacjach programów wykorzystujących IPv6.[1]

- **Narzędzia konfiguracji : net-tools bądź iproute2**

Do konfiguracji IPv6 możemy wykorzystać jeden z dwóch pakietów oprogramowania: net-tools bądź iproute2.[1] Odnośniki do miejsc gdzie można znaleźć wspomniane oprogramowanie znajduje się na końcu artykułu w dziale dodatki.
Kompilacja iproute2 w środowisku wykorzystującym bibliotekę glibc przebiega stosunkowo prosto. W wyniku kompilacji otrzymujemy dwa programy - „ip" oraz „tc". Pierwszy służy do

konfiguracji sieci IPv4/IPv6, natomiast drugi wykorzystuje się do kontrolowania algorytmów kolejkowania pakietów (w tym także IPv6). Mając nowe (obsługujące IPv6) jądro oraz odpowiednie narzędzia można przystąpić do konfiguracji.

4. KONFIGURACJA IPV6 W SYSTEMIE

Sprawdzamy czy Ipv6 jest obecne w systemie:
„ip addr show lo".

```
# ip addr show lo
1: lo: <LOOPBACK,UP> mtu 3924 qdisc noqueue
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 brd 127.255.255.255 scope global lo
inet6 ::1/128 scope host
#
```

IPv6 jest obecne - świadczy o tym linijka „inet6 ::1/128 scope host".
Adres „::1" jest adresem IPv6 interfejsu loopback. Jeśli powyższe polecenie nie pokazuje takiej linijki to najpewniej skompilowano IPv6 jako moduł. Należy wówczas wykonać „modprobe ipv6" i ponownie sprawdzić obecność adresu IPv6.[1]
Działanie IPv6 można sprawdzić przy użyciu np. ping6 z pakietu iputils

```
# ping6 -nc3 ::1
PING ::1(::1) from ::1 : 56 data bytes
64 bytes from ::1: icmp_seq=0 hops=64 time=0.2 ms
64 bytes from ::1: icmp_seq=1 hops=64 time=0.1 ms
64 bytes from ::1: icmp_seq=2 hops=64 time=0.1 ms
--- ::1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.2 ms
#
```

Podobnie jak w IPv4 interfejsy sieciowe (np. eth0) mogą mieć przypisanych wiele adresów IPv6. Przeglądając adresy na interfejsie ethernetowym (ip addr show eth0) można zauważyć adres link-local. Adresy te dla interfejsów ethernet generowane są automatycznie na podstawie identyfikatora **interfejsu** np. adresu MAC karty sieciowej. Narzędzie „iproute2" w przeciwieństwie do „ifconfig" pozwala na oglądanie wszystkich adresów na danym interfejsie. Możliwym jest również dodawanie kilku adresów IPv4/IPv6 do jednego interfejsu bez stosowania aliasów.

5. SIEĆ 6BONE

Gdy mamy działające IPv6 możemy przystąpić do podłączania naszej maszyny do sieci 6bone. Pierwszym krokiem jest uzyskanie puli adresów IPv6. Najprościej jest zwrócić się do administratora Rafała Maszkowskiego <rzm@icm.edu.pl>. Należy podać m.in. adres IPv4 końca tunelu SIT (czyli po prostu adres IPv4 serwera).[1]

Załóżmy, że przyznano sieć 3ffe:803:101::/48. Stosuje się dwie metody konfiguracji tuneli :
Przykłady zaczerpnięto z odpowiedniej literatury[1].

- **bazującą na adresach link-local.**

 # echo 1 >/proc/sys/net/ipv4/ip_forward
 # echo 1 >/proc/sys/net/ipv6/conf/all/forwarding
Pozwalamy na forwardowanie pakietów IPv4 oraz IPv6

 # ip addr add 3ffe:803:101::1/128 dev eth0
 # ip route add 3ffe:803:101::1/128 dev eth0
Przypisujemy adres 3ffe:803:101::1/128 do interfejsu eth0 oraz ustawiamy odpowiedni routing.

*# ip tunnel add tunel mode sit local 192.168.11.30 *
 # remote 212.168..40.34 ttl 64
 # ip link set tunel up
Tworzymy nowy tunel o nazwie „tunel". Typ tunelu to „SIT", lokalny adres IPv4 „192.168.11.30", natomiast adres przeciwnego końca tunelu to „212.168.40.34". Ostatnia komenda „podnosi" interfejs naszego tunelu.

*# ip route add 3fffe::/16 via fe80::167.34.22.76 *
 # dev tunel
Końcową operacją jest ustawienie statycznego routingu (do sieci 3ffe::0/16) poprzez router po przeciwnej stronie tunelu.

- **bazującą na dynamicznych tunelach.**

Dwie pierwsze operacje, czyli pozwolenie na forwardowanie oraz przypisanie adresu do interfejsu są takie same jak w przypadku tunelu bazującego na adresach link-local.

 # ip link set sit0 up
Następnie podnosimy interfejs sit0 będący interfejsem tunelu SIT (IPv6-in-IPv4).

 # ip route add 3ffe::/16 via ::167.34.22.76 dev sit0
Ustawiamy statyczny routing. W powyższych przykładach adres IPv6 serwera to 3ffe:803:101::1, adres IPv4 serwera to 192.168.11.30, natomiast drugiej strony 212.168.40.34. Warto także nadmienić, iż oba sposoby tworzenia tuneli mogą być stosowanie równocześnie. Po tych operacjach serwer po drugiej stronie tunelu powinien odpowiadać na pingi skierowane na jego adres IPv6 (oczywiście po drugiej stronie tunelu także należy wszystko poprawnie skonfigurować, lecz to robi już osoba, od której otrzymaliśmy pulę adresów IPv6).
W przypadku konfigurowania tunelu z pTLA ICM adresem tym jest 3ffe:803::1.
Programem przydatnym w przypadkach, gdy tunel nie działa mimo teoretycznie poprawnej konfiguracji jest tcpdump pozwalający na śledzenie pakietów.

6. ROUTING W SIECI 6BONE

Najczęściej w sieci 6bone stosuje się routing statyczny. Jest to rozwiązanie wystarczające w przypadku gdy posiadamy jeden tunel. Gdy liczba tuneli jest większa niż jeden warto zastosować routing dynamiczny bazujący na protokole BGP4+ [3]. Dynamiczny routing pozwala w przypadku awarii jednego z tuneli na skierowanie całego ruchu poprzez inny, istniejący tunel.
Na większości serwerów IPv6 w Polsce pracuje daemon dynamicznego routingu - mrt (Multi-threaded Routing Toolkit).
Poniżej przedstawiono przykładowy plik konfiguracyjny („!" oznacza komentarz):[1]

line vty
login
password twoje_haslo
port 5674

enable password twoje_haslo

! 64123 - Autonomus System Number (ASN). Ze wzgledu na testowy charakter sieci 6bone !można wybrac dowolny ale aktualnie nie uzywany ASN.
router bgp 64123

! podsiec
network 3ffe:803:101::/48

! bedziemy wysylac wylacznie zagregowane trasy (w tym wypadku bedziemy wysyac !informacje o routingu do calej naszej sieci, bez dzielenia tras na mniejsze - zmniejsza
! to obciazenie routerów BGP4+)
aggregate-address 3ffe:803:101::/48 summary-only as-set

! bedziemy informowac o naszych statycznych trasach
redistribute static

! 3ffe:803::1 - adres IPv6 na którym działa daemon dynamicznego routingu naszego sasiada
! 8664 - to numer ASN naszego sasiada
neighbor 3ffe:803::1 remote-as 8664

! ICM - symbol naszego sasiada (bedzie uzywany m.in. w logach mrt)
neighbor 3ffe:803::1 description ICM

! bgp4+ 1 - bedziemy uzywali BGP4+
neighbor 3ffe:803::1 bgp4+ 1

! Trasy statyczne, których mrt sam nie bedzie zmienial
route 3ffe:803:100::/48 :: eth0
route 3ffe:803::1/128 fe80::212.168.40.34 tunel

Informacje o aktualnych trasach, które mrt otrzymuje i rozgłasza możemy zobaczyć łącząc się przez telent z portem 5674 naszego serwera i wydając odpowiednią komendę:

„show bgp”.

7. DNS

Do obsługi adresów IPv6 wprowadzono nowy rekord DNS jakim jest „AAAA”.[1]
Do pliku obsługującego wymienianą domenę należy dopisać:

host-ipv6 IN AAAA 3ffe:803:101::1

Dla odwrotnego DNSu sprawa nieco się komplikuje. Należy utworzyć delegację primary dla domeny **1.0.1.0.3.0.8.0.e.f.f.3.ip6.int**, i podać adres hosta:

1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN

PTR host-ipv6.nasza.domena.pl.

Ponieważ prefiks naszej podsieci składa się z 48 bitów (48/4=12 cyfr) to adres hosta musi się składać z (128-48)/4=20 cyfr. Po delegację odwrotnego DNS-u również musimy się zgłosić do osoby, która przydzieliła nam podsieć.

Sprawdzanie DNS-u :
„host –t AAAA host-ipv6.nasza.domena.pl":

```
# host -t AAAA host-ipv6.nasza.domena.pl
host-ipv6.nasza.domena.pl IPv6 address 3ffe:803:101::1
#
```

Sprawdzanie rDNS-u

```
# nslookup -query=any 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0\
.0.0.0.0.0.0.1.0.1.0.3.0.8.0.e.f.f.3.ip6.int
[...]
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.1.0.
3.0.8.0.e.f.f.3.ip6.int name = host-ipv6.nasza.domena.pl
[...]
```

8. BEZPIECZEŃSTWO

Wraz z IPv6 mają zostać wprowadzone jako standard procedury szyfrowania oraz autentyfikacji pakietów [4],[5]. Niestety aktualnie narzędzia typu firewall IPv6 pod Linuxem dopiero się rozwijają (w Linuxie jest częściowo zaimplementowany firewall IPv6, co możemy przeczytać w opisie jądra (linux/net/ipv6/ip6_fw.c)) w związku z czym nasz serwer może stać się niepożądaną furtką dostępu do sieci lokalnej. Ze względów bezpieczeństwa nie należy uruchamiać serwisów typu „telnetd", „finger" na publicznie dostępnych adresach IPv6.
Zamiast tego możemy je uruchamiać na adresach np. link-local. Opcję taką umożliwia zamiennik inetd - „rlinetd". Warto także zastosować „tcp_wrappers" ze wsparciem dla IPv6. Do autentyfikacji można zastosować Kerberosa 5. Aktualnie dynamicznie rozwija się dystrybucja kerberosa o nazwie kodowej „heimdal". Autorzy „heimdala" chcą włączyć wsparcie dla IPv6 do każdego programu wchodzącego w skład dystrybucji. Wszelkie wymienione narzędzia można znaleźć w PLD.

9. ZAKOŃCZENIE

Mimo, iż już działają sieci bazujące na protokole IPv6 to jednak przewiduje się, że protokół IPv4 będzie z powodzeniem panował jeszcze przez ok 5-15lat. Niemniej jednak niedawno dokonano oficjalnego przydziału adresów IPv6 dla amerykańskiego ISP z puli adresów nie testowych. Jak więc widać IPv6 zdobywa coraz większą popularność nie tylko w środowisku administratorów - eksperymentatorów.
Planuje się, że przejście na protokół IPv6 będzie odbywać się stopniowo, a sieć IPv4 i IPv6 będą przez jakiś czas współistnieć. Komunikację pomiędzy obiema sieciami mają zapewnić translatory nagłówków oraz proxy (np. SOCKS64 będący modyfikacją SOCKS5, umożliwiający

komunikację hostom IPv4 z innymi hostami obsługującymi tylko IPv6 i odwrotnie).

DODATKI

Strony WWW oraz serwisy FTP

- http://www.6bone.net/, http://www.6bone.pl/. (siec 6bone na świecie i w Polsce)
- http://cvsweb.pld.org.pl/, ftp://ftp.pld.org.pl/. (Zasoby polskiego Linuxa min spora ilość oprogramowania pod IPv6.)
- ftp://ftp.inr.ac.ru/ip-routing/, ftp://ftp.icm.edu.pl/pub/Linux/iproute/, http://snafu.freedom.org/linux2.2/iproute-notes.html. (Narzędzia do konfiguracji sieci w tym „iproute2", „iputils" wraz z cennymi uwagami.)
- http://www.bieringer.de/linux/IPv6/IPv6-HOWTO/IPv6-HOWTO.html,
- ftp://ftp.pl.kernel.org/pub/kernel/v2.2/. (Stabilne jądro Linuxa.)
- http://www(.ipv6).pld.org.pl/. (Strony Polish Linux Distribution)
- ftp://ftp.inner.net/pub/ipv6/. (Aplikacje przystosowane do IPv6 przez Craiga Metza.)
- http://www.v6.wide.ad.jp/Papers/socks64/. (SOCKS64 - proxy IPv4<->IPv6.)

Listy dyskusyjne

- 6bone - Polska. Adres listy: 6bone-pl@sunsite.icm.edu.pl, zapisy poprzez majordomo@sunsite.icm.edu.pl
- 6bone. Adres listy: 6bone@isi.edu, zapisy poprzez majordomo@isi.edu
- Użytkownicy IPv6. Adres listy: users@ipv6.org, zapisy poprzez majordomo@ipv6.org
- IPv6 w Linuxie. Adres listy: linux-ipv6@inner.net, zapisy poprzez linux-ipv6-request@inner.net

LITERATURA

[1] A. Miśkiewicz, “Ipv6 Protokół Internetowy Następnej Generacji”, Polish Linux Distribution Team, 10 sierpnia 1999

[2] R. Hinden, S. Deering, „IP Version 6 Addressing Architecture”

[3] T. Bates, R. Chandra, D. Katz, Y. Rekhter, „Multiprotocol Extensions for BGP-4”

[4] R. Atkinson, „Security Architecture for the Internet Protocol”

[5] S. Kent, R. Atkinson, „IP Authentication Header”

[6] W. Stevens, M. Thomas, „Advanced Sockets API for IPv6”

[7] M. Allman, S. Ostermann, C. Metz, „FTP Extensions for IPv6 and NATs”

[8] S. Thomson, T. Narten, „IPv6 Stateless Address Autoconfiguration”

[9] R. Gilligan, S. Thomson, J. Bound, W. Stevens, „Basic Socket Interface Extensions for IPv6”

[10] Alexey N. Kuznetsov, „IP Command Reference”

[11] M. Sportack, “Sieci komputerowe – ksiega eksperta”, Helion Gliwice 1999r.

[12] Praca zbiorowa , „Vademecum Teleinformatyka” IDG Poland S.A. W-wa 1999r.

[13] K. Zieliński (red.), „Ćwiczenia do laboratorium sieci komputerowych”, AGH Kraków 1999r.