

**WYŻSZA SZKOŁA TECHNOLOGII INFORMATYCZNYCH
W KATOWICACH**

WYDZIAŁ INFORMATYKI

KIERUNEK: INFORMATYKA

Tomasz xxx

Nr albumu 2466

Studia niestacjonarne

Wdrożenie adresacji IPv6 w sieci lokalnej

Promotor: dr inż. Mateusz xxx

opiekun dodatkowy: mgr inż. Paweł xxx

w roku akademickim 2010/2011

Katowice 2011

Spis treści

1	Ogólne informacje o protokole IPv6.....	7
1.1	Różnice między protokołami IPv4, a IPv6.....	7
1.2	Formaty prezentacji adresu IPv6.....	9
1.3	Metody wdrożenia adresacji IPv6 w działającej sieci IPv4 do sieci.....	12
2	Aplikacja LMS.....	14
2.1	Opis systemu LMS.....	14
2.2	Konstrukcja.....	14
2.3	Biblioteka SMARTY.....	15
2.4	Język T-Script.....	16
3	Inne aplikacje wykorzystane w projekcie.....	17
3.1	OpenVPN.....	17
3.2	IP6TABLES.....	17
3.3	Radvd.....	17
3.4	Image_Graph.....	17
4	Konfiguracja wstępna routerów.....	19
4.1	Ustalenie adresacji IPv6.....	19
4.2	Połączenie routerów tunelem szyfrowanym.....	22
5	Modyfikacje bazy danych MySQL.....	25
5.1	Opis dodanych pól dla sieci.....	25
5.2	Opis dodanych pól dla komputerów.....	26
5.3	Opis dodanych pól dla statystyk.....	26
6	Modyfikacje LMS dla protokołu IPv6.....	27
6.1	Obsługa sieci IPv6.....	27
6.1.1	Dodawanie.....	27
6.1.2	Informacje/Edycja.....	33
6.1.3	Usuwanie.....	34
6.1.4	Klasy adresowe dla konfiguracji statycznej.....	35
6.1.5	Klasy adresowe dla autokonfiguracji bezstanowej (stateless).....	36
6.2	Obsługa komputerów IPv6.....	36
6.2.1	Dodawanie.....	36

6.2.2 Edycja.....	39
6.2.3 Usuwanie.....	40
6.2.4 Klasy adresowe.....	41
6.3 Statystyki komputerów.....	44
6.4 Przeładowanie konfiguracji.....	47
7 Dostęp do sieci.....	50
7.1 Skrypt zabezpieczający sieć.....	50
7.2 Metoda autoryzacji klientów.....	52
7.3 Klasy adresowe dla klientów – routing.....	55
8 Skrypty automatyzujące proces konfiguracji wstępnej.....	58
9 Opis wybranych fragmentów kodu źródłowego.....	62
9.1 Klasa ipv6.....	62
9.2 Dodane moduły.....	63
10 Wnioski.....	65

Wstęp

Praca poświęcona jest wdrożeniu protokołu IPv6 w sieci dostawcy internetu. Jest to ważne zagadnienie, ponieważ nawet najwięksi dostawcy usług internetowych nie oferują swoim klientom dostępu do najnowszej wersji tego protokołu. Wynika to głównie z małej popularności samego protokołu, jak i niewielkiej wiedzy nt. Korzyści z jego wdrożenia. Niewielka wiedza spowodowana jest małą ilością publikacji książkowych i internetowych, a jeśli są, to zajmują niewielką objętość z pominięciem istotnych kwestii dotyczących konfiguracji na poszczególnych systemach operacyjnych.

Protokół IPv6 nie jest nowy, został przedstawiony w 1992 roku przez IETF¹ i wydaje się potrzebnym i dobrze przygotowanym rozwiązaniem wobec zmniejszającej się puli adresów IPv4. Adres IPv6 składa się ze 128 bitów, co w porównaniu do adresu w obecnie używanym protokole IPv4, który składa się z 32 bitów jest różnicą ogromną. Protokół IPv6 [12] pozwala nadać adresy dla 2^{128} urządzeń. Niestety ze względu na brak wdrożeń IPv6 w sieciach szkieletowych, dostawcy internetu przydzielają swoim klientom adresy IPv4, których pula powoli się kończy.

Rozwiązaniem, które powoduje wydłużenie żywotności IPv4 jest translacja adresów NAT², czyli udostępnienie klientom prywatnych adresów IPv4, które są zamieniane przez router dostawcy na publiczny adres IPv4, jednakowy dla wszystkich komputerów w sieci prywatnej. Minusem tego rozwiązania jest większe obciążenie procesora routera dostawcy co wymaga większych nakładów finansowych w architekturę sprzętową. Kolejną niedogodnością jest brak możliwości bezpośredniego połączenia się z internetem z hostem, będącym za NATem, posiadającym wewnętrzny adres IP. Wszystkie komputery w sieci prywatnej widoczne są pod tym samym adresem

1 IETF – ang. Internet Engineering Task Force - to międzynarodowe stowarzyszenie projektantów sieci, operatorów, producentów i naukowców zajmujących się rozwojem architektury internetu

2 NAT – ang. Network Address Translation – przesyłanie pakietów, zmieniając ich adresy IP źródłowe lub docelowe przez router

IP, więc nie ma możliwości łatwego odróżnienia poszczególnych komputerów w tej sieci. Stosowane w tym celu mechanizmy translacji portów są niewygodne tak dla administratora sieci, jak i dla użytkowników.

Z uwagi na kończącą się już pulę adresów IPv4, coraz więcej uwagi poświęca się protokołowi IPv6. Wydaje się, że jego wdrożenie jest nieuchronne, dlatego praca skupia się na wprowadzeniu dodatkowych możliwości dla popularnego, bezpłatnego systemu zarządzania siecią o nazwie LMS (LAN Management System), przeznaczony dla dostawców internetu (ISP). System sam w sobie nie ma obsługi nowego protokołu internetowego, dlatego nie ma tam zaimplementowanych funkcji do jego obsługi. Całość pracy wymagała napisania od podstaw klasy interpretującej dla IPv6, koniecznej do obsługi nowych modułów. Administrator dzięki nowym funkcjonalnościom jest w stanie zarządzać sieciami i komputerami IPv6 z zachowaniem wszelkich zabezpieczeń uniemożliwiających nieautoryzowany dostęp do internetu, a wyświetlane statystyki w formie graficznej zobrazują wykorzystywany ruch poszczególnych hostów. Wstępna konfiguracja routera do obsługi protokołu IPv6 po wprowadzeniu sieci jest ściągana jako skrypt, a klienci korzystający ze statycznej konfiguracji będą mogli skonfigurować swoją sieć również za pomocą skryptów automatyzujących proces konfiguracji IPv6. Będzie też możliwość przydzielenia adresów IPv6 za pomocą autokonfiguracji bezstanowej (stateless). Bardziej wymagający klienci dostaną do własnej dyspozycji pulę publicznych adresów o długości 120 bitów.

1 Ogólne informacje o protokole IPv6

1.1 Różnice między protokołami IPv4, a IPv6

Podstawową różnicą pomiędzy protokołem IPv6, a obecnie używanym IPv4, widoczną gołym okiem jest wydłużony do 128 bitów adres urządzenia. Adres IPv4 zawiera 32 bity. Ipv6 pozwala więc zaadresować 2^{96} , czyli prawie 10^{29} razy więcej hostów w porównaniu do IPv4. Obrazuje to ogrom możliwości zaadresowania każdego urządzenia, posiadającego obsługę IPv6.

IPv6 posiada prostszy, posiadający 8 pól nagłówek, w porównaniu do nagłówka IPv4, mającego 14 pól. Nagłówek IPv6 posiada stały rozmiar, a rozmiar nagłówka IPv4 może się zmieniać. Wszystkie pola (tab. 1), za wyjątkiem Adresu źródłowego i docelowe, są wyrównane do 64 bitów. Umożliwia to bezpośrednie składowanie i dostęp do pamięci, co ma wpływ na szybkość przetwarzania.

Wersja (4 bity)	Priorytet (8 bitów)	Etykieta przepływu (20 bitów)
Długość danych (16 bitów)	Następny nagłówek (8 bitów)	Limit przeskoków (8 bitów)
Adres źródłowy (128 bitów)		
Adres docelowy (128 bitów)		

Tab. 1: Opis pól nagłówka IPv6 (źródło: dokument RFC 2460)

Pole sumy kontrolnej zostało usunięte z nagłówka IPv6, aby zwiększyć wydajność routingu. Detekcją błędów zajmuje się obecnie warstwa łącza danych (2 warstwa modelu OSI) i warstwa transportowa (4 warstwa modelu OSI). Obliczana suma kontrolna w tych warstwach jest wystarczająca, aby pominąć obliczanie sumy w warstwie sieciowej (3 warstwa).

W protokole istnieją również nagłówki rozszerzające, które służą do zwiększania możliwości protokołu IPv6. Obecnie zdefiniowanych jest 6 rozszerzeń protokołu:

- Hop-by-Hop Options
- nagłówek routingu
- nagłówek fragmentacji

- nagłówek opcji docelowych
- nagłówek uwierzytelniania
- Encrypted Security Payload

Mobilność dzięki protokołowi MobileIP³, jest wbudowana w protokół IPv6. W protokole IPv4 jest to tylko dodatek. Z sieci IPv6 usunięto transmisję broadcast⁴ w zamian dodając transmisję anycast⁵. Transmisja unicast⁶ pozostaje w nie zmienionej formie.

W odróżnieniu od IPv4, gdzie hosty otrzymywały adresy w sposób statyczny, a protokoły dynamicznej konfiguracji, np. DHCP, pojawiły znacznie później niż sam protokół, twórcy IPv6 założyli że autokonfiguracja – stanowa lub bezstanowa – będzie podstawowym sposobem nadawania adresów.

Autokonfiguracja bezstanowa (ang. stateless autoconfiguration) hostów IPv6, używa procesu zwanego "Router Advertisement". Host po włączeniu interfejsu otrzymuje prefiks, wysłany przez router. Prefiks zawiera adres sieci IPv6, który ma długość 64 bitów (np. 2001:470:C8C6:1::/64), tylko adresy o długości 64 bitów są akceptowane przez bezstanową autokonfigurację. Po otrzymaniu prefiksu hosty generują kolejne 64 bity na podstawie adresu MAC karty sieciowej. Odbywa się to w następujący sposób [17]:

Rozprowadzany przez router adres sieci:

2001:470:C8C6:1::/64

1. Generowanie części hosta na podstawie adresu MAC, podzielone na etapy:

00	50	8B	0A	B8	52
----	----	----	----	----	----

2. Dopisanie do środkowej części adresu MAC ciągu: FFFE

3 protokół komunikacyjny, umożliwiający urządzeniom mobilnym zmianę sieci, urządzenie takie musi mieć przypisany stały adres IP

4 Broadcast – transmisja danych polegająca na wysyłaniu pakietów do ostatniego adresu w danej sieci, z tego adresu pakiety trafiają do wszystkich hostów w sieci

5 Anycast – jest to rodzaj unicastu, przypisanego jednocześnie do kilku hostów. Pakiet, który zostanie wysłany na adres anycast, będzie dostarczony do najbliższego hosta, do którego jest przypisany ten adres. Najbliższy host jest wybierany przez protokół routingu. Adres anycast może być wykorzystany tylko jako adres docelowy.

6 Unicast – transmisja między dwoma hostami, pakiet wysyłany jest z jednego hosta do innego

00	50	8B	FF	FE	0A	B8	52
----	----	----	----	----	----	----	----

3. Następuje rozbitcie dwóch początkowych liczb (00) do formatu binarnego w przypadku, gdy prefiks rozprowadzany przez Radvd jest adresem publicznym:

0000 0000

następuje inwersja 7 bitu, licząc od lewej strony:

0000 0010

liczba zamieniana jest na format szesnastkowy:

02

4. Po wszystkich operacjach adres części hosta, zgodny z formatem EUI-64 wygląda następująco:

0250:8BFF:FE0A:B852

Finalnie adres IPv6 po dodatkowym skróceniu zer, który wygenerował sobie host ma postać:

2001:470:C8C6:1:250:8BFF:FE0A:B852 /64

Wygenerowany adres jest przypisany do interfejsu i jest oznaczony jako Globalny. Brama jest automatycznie przypisana na adres routera, z którego host otrzymał prefiks.

Prócz bezstanowej autokonfiguracji, możliwa jest również **autokonfiguracja stanowa** (ang. stateful). Działa na takiej samej zasadzie jak serwer DHCP w protokole IPv4. Trzeba skonfigurować serwer obsługujący DHCPv6, który będzie przydzielał adresy w protokole IPv6.

1.2 Formaty prezentacji adresu IPv6

Adres IPv6 można zaprezentować, za pomocą 3 formatów:

- **Preferowany**

Adres w formacie preferowanym jest pokazywany w całości, 8 pól szesnastkowych (jedno pole jest równe 16 bitom), z czego każde pole ma wartość szesnastkową w przedziale od **0x0000** do **0xFFFF**.

Przykładowy zapis w formacie preferowanym:

2001:0470:1f13:0068:0000:0000:0014:0000

- **Skompresowany**

Skrócenia adresu można dokonać, wtedy gdy na początku 16-bitowego pola występuje przynajmniej jedno zero, lub jedno lub więcej pól składa się z samych zer. Jeśli na początku pola występuje jedno lub więcej zer, możemy je po prostu usunąć. Wyjątek stanowi pole, składające się z samych zer, wówczas zostawiamy jedno zero. Kolejnym etapem skrócenia adresu jest usunięcie pól równych 0x0000. Pola takie, jeśli występują kolejno, możemy usunąć, zastępując je znakiem „:”. Znak „:” w adresie może wystąpić tylko raz. Zastosowanie go więcej niż jeden raz, uniemożliwiłoby poprawną zamianę na format preferowany z uwagi na brak możliwości oceny, w którym miejscu i ile pól z zerami należy wpisać. W poniższej tabeli zaprezentowano etapy skracania adresów.

Format preferowany	Format skrócony (usunięcie zer)	Format skrócony z użyciem „::”
0000:0000:0000:0000:0000 :0000:0000:0001	0:0:0:0:0:0:1	::1
0000:0000:0000:0000:0000 :0000:0000:0100	0:0:0:0:0:0:100	::100
2001:0470:1f13:0068:0000: 0000:0014:0000	2001:470:1f13:68:0:0:14:0	2001:470:1f13:68::14:0
2001:0000:0000:0068:0000 :0000:0014:0000	2001:0:0:68:0:0:14:0	2001::68:0:0:14:0

Tab. 2: Proces skracania adresu IPv6 (źródło: Opracowano na podstawie publikacji: Regis Desmeules: IPv6: Sieci oparte na protokole IP w wersji 6. WN PWN, Warszawa 2006, s. 74.)

- **z osadzonym adresem IPv4**

W takiej sytuacji adres IPv6 jest dzielony na 2 części. Pierwsza jest w postaci heksadecymalnej i składa się z **96 bitów**, a druga jest w postaci dziesiętnej, zawierającej adres IPv4, składa się z **32 bitów**. Adresy takie używane są do ustanowienia automatycznego tunelu służącego do przenoszenia pakietów IPv6 przez sieci IPv4. Przykładowy adres:

::1fa:ff:213.214.215.216

Adres IPv6 wpisany w przeglądarkę internetową, musi być zamknięty w nawiasach kwadratowych:

http://[2001:470:1f13:68::14:0]/podstrona

Jeśli konieczne jest wpisanie numeru portu, trzeba to zrobić za nawiasem kwadratowym:

http://[2001:470:1f13:68::14:0]:4321/podstrona⁷

⁷ Reprezentacja adresu IPv6 w formacie URL

1.3 Metody wdrożenia adresacji IPv6 w działającej sieci IPv4 do sieci

- **Podwójny stos (Dual-Stack)**

Metoda podwójnego stosu polega na tym, że na sieci lub urządzeniu dostępne są zarówno adresy IPv4 jak i IPv6. Jest to najczęściej wykorzystywana metoda przy wdrożeniu, ponieważ nie ma wpływu na działanie starszej wersji protokołu IP. Podczas wysyłania zapytań domenowych, urządzenie o podwójnym stosie w odpowiedzi zwrotnej dostanie pakiet IPv6 jeśli w nazwie o którą zapytało urządzenie jest rekord typu AAAA⁸, lub pakiet IPv4, jeśli w nazwie o którą zapytało urządzenie jest rekord typu A.

Cechą ujemną jest to, że urządzenia korzystające z podwójnego stosu, muszą posiadać adres IPv6 jak i IPv4, w przeciwnym razie, gdy będziemy korzystać tylko z IPv6, nie będzie możliwe otrzymywanie pakietów IPv4. [18]

W niniejszej pracy skorzystano właśnie z tej metody wdrożenia adresacji IPv6.

- **NAT Klasy Operatorskiej (Carrier Grade NAT)**

CGN (ang. Carrier Grade NAT) stosuje translację (NAT) adresów IPv6 na IPv4, pozwala to stosować na urządzeniach klienckich tylko i wyłącznie adresy IPv6, równocześnie, ciągle będzie dostęp do sieci IPv4 za pośrednictwem CGN. Metoda wdrażana tylko na urządzeniach operatora sieci. [15]

- **NAT464**

Polega na komunikacji między CGN dostawcy, a urządzeniem klienta tylko i wyłącznie za pomocą protokołu IPv6. Klienci otrzymują prywatne adresy IPv4.

Pakiet wychodzący od klienta jest tłumaczony przez bramkę na adres IPv6 (NAT46), dalej kieruje się do urządzenia operatora CGN, które sprawdza czy pakiet ma trafić do sieci IPv6 czy IPv4; jeśli ma trafić do IPv6, trafia bezpośrednio do tej sieci; jeśli ma trafić do IPv4, jest ponownie zamieniany na IPv4 (NAT64) i przesyłany dalej.

Dzięki tej metodzie operator ma przydzieloną adresację IPv6 dla klientów co pozwoli na późniejszą migrację. Minusem jest kompatybilność urządzeń klienta, muszą

⁸ Rekord AAAA systemu DNS w protokole IPv6 mapuje nazwę hosta na jego adres IPv6

obsługiwać metody translacji NAT464, a niewiele urządzeń to potrafi. [16]

- **Podwójny Stos 'Lite' (ang. Dual-Stack Lite)**

Wysyłany pakiet IPv4 jest opakowany w pakiet IPv6 na bramce domowej; na urządzeniu operatora CGN pakiet jest odpakowany i dzięki NAT44 na urządzeniu operatora przesyłany dalej. Natomiast dla adresów IPv6 jest czysta droga do celu.

Dzięki metodzie DS-Lite, możemy korzystać z urządzeń posiadających pojedyncze adresy IPv6 jak i IPv4. Do poprawności działania metody, konieczna jest aktualizacja bramek CPE, umożliwiająca obsługę DS-Lite. [14]

2 Aplikacja LMS

2.1 Opis systemu LMS

LAN Management System [13] jest zintegrowanym systemem służącym do zarządzania sieciami, przeznaczonym dla różnej wielkości dostawców internetu. Zarządzanie siecią odbywa się poprzez przeglądarkę internetową, wpisując adres serwera i katalog z zainstalowanym LMS-em. Oprogramowanie jest stworzone w językach: PHP, Perl i C, współpracuje z różnymi bazami danych (MySQL, PostgreSQL), składa się z przyjaznego i intuicyjnego interfejsu użytkownika oraz programów instalowanych na serwerze dostępowym udostępniając następujące funkcjonalności:

- zarządzanie dostępem do internetu (w tym kontrola pasma i statystyki)
- ewidencja klientów i sprzętu (mapa sieci)
- moduły finansowo-księgowe z fakturowaniem
- korespondencja seryjna i wiadomości administracyjne do klientów
- zarządzanie kontami i hostingiem
- system obsługi zgłoszeń (helpdesk)
- zarządzanie dowolnymi usługami
- zarządzanie czasem (terminarz)
- panel obsługowy dla abonenta

System ma wiele zalet, ale z punktu widzenia niniejszej pracy niestety ma też dużą wadę: brak obsługi protokołu IPv6. Kod opublikowany jest na licencji GNU GPL v2.

2.2 Konstrukcja

System LMS jest podzielony na kilka katalogów, z których każdy pełni osobną rolę:

- **backups** – katalog przeznaczony na kopie zapasowe systemu, konieczne jest ustanowienie praw zapisu

- **bin** – skrypty pisane w języku PERL, służą do operacji administracyjnych wykonywanych przez opcję przeładowania systemu za pomocą przeglądarki internetowej lub poprzez demona LMS wykonującego przeładowanie ręcznie lub co określony czas, ustawiony dla każdego zadania z osobną (przykładowo odświeżanie statystyk dla hostów co 10 minut)
- **contrib** – formularze służące do wyświetlania faktur, wiadomości administracyjnych dla klientów, itp.
- **daemon** – demon LMS napisany w języku C, służący do wykonywania przeładowań systemu, demona trzeba dodać do katalogów odpowiedzialnych za start aplikacji w systemie linux, jeśli chodzi o mój projekt (system Debian), ścieżka wygląda następująco: /etc/init.d, do skryptu startującego trzeba stworzyć dowiązanie symboliczne w katalogu /etc/rc2.d
- **doc** – dokumentacja techniczna LMS
- **documents** – zeskanowane dokumenty klientów w formie plików graficznych
- **img** – grafika systemu LMS
- **lib** – biblioteki, klasy systemu
- **modules** – moduły systemu, część programowa wszystkich podstron, język programowania: PHP
- **sample** – pliki konfiguracyjne i skryptowe z domyślną zawartością
- **Smarty** – zawartość z obiektową biblioteką smarty, która pozwala na podział aplikacji na część programową (PHP) i część prezentacyjną (HTML) pozwala to na lepsze uporządkowanie systemu
- **templates** – moduły systemu, część prezentacyjna wszystkich podstron (HTML)
- **templates_c** – tymczasowo wygenerowane pliki HTML dzięki bibliotece smarty na podstawie modułów PHP i HTML

2.3 Biblioteka SMARTY

Smarty jest obiektową biblioteką, która służy do tworzenia szablonów dla aplikacji skryptowych PHP. Została wykorzystana przez twórców systemu LMS, a także podczas przygotowywania prezentowanych w pracy rozszerzeń LMSa. Pozwala na separację aplikacji PHP od warstwy prezentacyjnej (HTML). Biblioteka działa

poprzez umieszczenie w pliku HTML znaczników, które są zastępowane przez właściwą zawartość pliku PHP. Pozwala także na umieszczenie struktur decyzyjnych, pętli. Wszystkie polecenia muszą być zawarte między klamrami {} w przeciwnym wypadku nie zostaną zinterpretowane przez bibliotekę. Zmienne wysyłane są do pliku HTML dzięki klauzuli:

```
Smarty->assign('net_eui', $net_eui);
```

a część prezentacyjną wywołujemy:

```
Smarty->display('netadd6.html');
```

gdzie znajduje się lokalizacja pliku HTML.

2.4 Język T-Script

Przeznaczeniem tego języka jest generowanie plików tekstowych na podstawie danych pobieranych z różnych źródeł np. bazy danych SQL. Składnia tego języka jest podobna do języków: C czy Javascript. Wszystkie podane polecenia muszą być zawarte między klamrami {} w przeciwnym wypadku dane zostaną zapisane do pliku tekstowego w niezmienionej postaci.

3 Inne aplikacje wykorzystane w projekcie

3.1 OpenVPN

Jest to pakiet służący do tworzenia szyfrowanych tuneli między komputerami, serwerami. Szyfrowanie informacji jest możliwe dzięki bibliotece OpenSSL⁹ oraz protokołów SSLv3 i TLSv1. Uwierzytelnienie połączenia może być wykonane za pomocą: kluczy, certyfikatu lub nazwy użytkownika i hasła. Aplikacja dostępna jest na większość systemów operacyjnych, tj. Linuks, Unix, Mac, Windows. W pracy pakiet został zastosowany do stworzenia tunelu szyfrowanego między routerami.

3.2 IP6TABLES

Program służący do filtrowania pakietów dla protokołu IPv6. Jest używany głównie jako firewall dla systemu Linuks/Unix/BSD. Posiada możliwości tworzenia, usuwania i sprawdzania właściwości tabel zarządzania pakietami sieciowymi IPv6. Pakiety są filtrowane, rutowane z poziomu jądra systemu, dlatego jądro musi mieć włączone moduły obsługujące **ip6tables**, jak i wszystkie reguły użyte w skrypcie tworzącym reguły **ip6tables**.

3.3 Radvd

Program w systemie operacyjnym Linuks odpowiada za rozprowadzanie prefiksu (Advertisement) podczas bezstanowej autokonfiguracji adresów IPv6. Prefiksem jest adres sieci składający się z 64 bitów, przeważnie jest to adres z puli publicznej.

Aby radvd uruchomił się prawidłowo, konieczne jest włączenie przekazywania pakietów IPv6: w pliku **/proc/sys/net/ipv6/conf/all/forwarding** należy ustawić wartość **1**.

3.4 Image_Graph

Zawiera zestaw klas, które tworzą wykresy liniowe, słupkowe, punktowe,

⁹ OpenSSL Project to zestaw narzędzi Open Source takich jak: protokoły Secure Sockets Layer (SSL v2/v3) oraz Transport Layer Security, dzięki czemu możliwe jest szyfrowanie danych

kołowe, mapy oparte na danych numerycznych pobieranych z różnych źródeł (baza SQL, plik tekstowy). Możliwe jest też wygładzanie linii (antialiasing).

Wykres jest wysoce konfigurowalny, dzięki czemu mamy możliwość uzyskać dokładny wygląd i odwzorowanie danych.

Wyświetlenie wykresu jest kontrolowane przez zestaw klas o nazwie `Image_Canvas`, który umożliwia łatwe wyjście na wiele różnych formatów, między innymi: GD (PNG, JPEG, GIF, WBMP), PDF (za pomocą `pdflib`), Scalable Vector Graphics (SVG). `Image_Graph` jest kompatybilny z PHP5.

4 Konfiguracja wstępna routerów

4.1 Ustalenie adresacji IPv6

Sieć IPv6 zostanie skonfigurowana na 3 routerach, umieszczonych w Chorzowie (główny router, do którego przypisane będą adresy IPv6 udostępnione przez dostawcę tego protokołu) na którym jest 1 podsieć, Katowicach na którym jest 1 podsieć, Mysłowicach na którym są 3 podsieci. Routery w Katowicach i Mysłowicach, będą się łączyły do routera w Chorzowie za pomocą tunelu VPN. Do interfejsów tunelowych (dla routerów w Katowicach i Mysłowicach) przypiszemy adresy IPv6, co pozwoli na pełną komunikację tych routerów z siecią IPv6. Po prawidłowej konfiguracji tunelów, przygotujemy klasy adresowe jakie udostępnimy każdemu z nich (biorąc pod uwagę zapotrzebowanie danej podsieci). Adresy IPv6 z udostępnionej klasy zostaną przypisane do wszystkich podsieci na interfejsach lokalnych na danym routerze, co umożliwi częściowe przygotowanie do uruchomienia sieci IPv6 dla klientów. Wszystkie routery mają dostęp do internetu za pośrednictwem protokołu IPv4.

Do ustalenia adresacji IPv6 potrzebna jest pula adresów publicznych. Jako, że dostawca internetu dla routera testowego nie oferuje klientom adresów IPv6, konieczne było uzyskanie adresów z innych źródeł za zgodą dostawcy. Jednym z tych źródeł jest stworzenie tunelu IPv4 z dostawcą udostępniającym klasy IPv6, wybrano: **www.tunnelbroker.net**. Po rejestracji możemy wygenerować sobie do 5 tuneli na 5 różnych zewnętrznych adresów IPv4. Uzyskano następujące klasy adresowe:

- 2001:470:c8c6::/48 – klasa służy do przyznawania adresów klientom za pomocą autokonfiguracji bezstanowej przez program Radvd, który potrzebuje klas o długości równej 64 bity
- 2001:470:1f13:68::/64 – klasa przeznaczona do tworzenia mniejszych klas adresowych, zaadresowanie interfejsów tunelowych, udostępniania klientom klas adresowych o długości 120 bitów (255 hostów)

Zdecydowano się na przyznawanie klientom klas adresowych, ponieważ coraz więcej klientów posiada w domu routery. Za routerami hosty już nie są widoczne

poprzez zastosowanie mechanizmu NAT. Udostępnienie publicznych klas adresowych pozwala na ominięcie mechanizmu NAT, poprzez zastosowanie zwykłego routingu. Klient konfiguruje własny router przypisując adres IPv6 do portu wyjściowego (przeważnie oznaczony jako WAN), po czym pierwszy adres klasy publicznej (udostępnionej klientowi) na interfejs lokalny routera (przeważnie routery takie posiadają 4 porty lokalne, plus połączenie bezprzewodowe, dlatego adres IPv6 powinien być przypisany do interfejsu mostkowego, dzięki temu wszystkie interfejsy lokalne będą w jednej sieci IPv6), a następnie hosty podłączone do routera, nadając im kolejne adresy IPv6. Alternatywnym oprogramowaniem umożliwiającym obsługę IPv6 jest DD-WRT¹⁰ dla różnych routerów domowych, router kliencki testowy, jest oznaczony modelem Linksys WRT-54G-TM.

Na routerze głównym (Chorzów) konfigurujemy adres tworząc skrypt w ścieżce **/etc/network/if-up.d** spowoduje to uruchomienie skryptu podczas podnoszenia się interfejsów:

```
#!/bin/sh
# podniesienie interfejsu sit0; interfejs sit0 służy do tworzenia tunelu typu
punkt-punkt w celu enkapsulacji pakietów IPv6 w pakietach IPv4
ifconfig sit0 up

# adres IPv6 z osadzonym adresem IPv4 używany w tunelach w celu przenoszenia
pakietów IPv6 przez sieć IPv4
ifconfig sit0 inet6 tunnel ::216.66.84.42
ifconfig sit1 up

# dodanie adresu IPv6 do interfejsu sit1, który zajmuje się w całości ruchem
IPv6
ifconfig sit1 inet6 add 2001:470:1f12:68::2/128
# dodanie trasy domyślnej
route -A inet6 add ::/0 dev sit1

# zabezpieczenie przed zapętleniem w sytuacji, gdyby jakaś część z klasy 64
bitowej nie została nigdzie przerutowana, pakiety trafiające na taką sieć będą
odrzucone
ip -6 route add 2001:470:1f12:68::/64 dev lo metric 2
```

¹⁰ Jest to oprogramowanie alternatywne, zawierające dodatkowe funkcjonalności, jakich nie znajdziemy w standardowym oprogramowaniu routerów domowych, m.in. obsługa protokołu IPv6. Oprogramowanie to można zastosować w wybranych modelach urządzeń. Pełna ich lista znajduje się na stronie domowej oprogramowania.

- **Chorzow** – główny router, który rozdziela klasy adresowe na inne routery, dla klientów dostępny 1 interfejs lokalny **eth0**, dla którego przeznaczona jest klasa 64 bitowa, ponieważ klienci będą dostawać adresy automatycznie. Dodatkowo klienci jeśli będą chcieli mogą dostać klasę adresową, publiczną, 120 bitową przeznaczoną na indywidualne potrzeby klienta.

Przypisane adresy IPv6 do interfejsów:

tun0: 2001:470:1f13:68::101/120 # adres IP przypisany do interfejsu tunelowego

eth0: 2001:470:c8c6:1::1/64

brama dla eth0: 2001:470:1f12:68::2

klasy adresowe od: 2001:470:1f13:68::40:0/120 **do:** 2001:470:1f13:68::47:fff/120

Przerutowane klasy adresowe na adresy IP routerów:

```
# router - Myslowice
ip -6 route add 2001:470:1f13:68::10:0/108 via 2001:470:1f13:68::102
ip -6 route add 2001:470:c8c6:2::/64 via 2001:470:1f13:68::102
# router - Katowice
ip -6 route add 2001:470:1f13:68::20:0/108 via 2001:470:1f13:68::103
```

- **Katowice** – router posiada 1 interfejs lokalny **eth0.22**:

Dla interfejsu **eth0.22** przeznaczona jest maska 117 bitowa z uwagi na to, iż na interfejs przypada ok 200 klientów.

Przypisane adresy IPv6 do interfejsów:

tun0: 2001:470:1f13:68::103/120 # adres IP przypisany do interfejsu tunelowego

eth0.22: 2001:470:1f13:68::20:1/117

brama dla eth0.22: 2001:470:1f13:68::101

```
ip -6 route add default dev tun0 metric 1 # trasa domyślna
ip -6 route add 2001:470:1f13:68::20:0/108 dev lo metric 2 # zabezpieczenie
przed zapętleniem
```

- **Myslowice** – router posiada 3 interfejsy lokalne eth0.11, eth0.22, eth0.33:

Dla interfejsów eth0.11, eth0.33 przeznaczona jest maska 119 bitowa z uwagi na to, iż na poszczególny interfejs przypada ok 100 klientów, udostępniona zostanie im klasa adresowa o długości 120 bitów, czyli każdy klient ma możliwość rozdyskrybowania we własnej sieci lokalnej 255 publicznych adresów IPv6. Interfejs eth0.22 posiada maskę 64 bitową, ponieważ będzie obsługiwany przez program Radvd, służący do autokonfiguracji bezstanowej.

Przypisane adresy IPv6 do interfejsów:

tun0: 2001:470:1f13:68::102/120 # adres IP przypisany do interfejsu tunelowego

eth0.11: 2001:470:1f13:68::10:1/119

brama dla eth0.11: 2001:470:1f13:68::101

klasy adresowe od: 2001:470:1f13:68::10:400/120 **do:** 2001:470:1f13:68::13:ffff/120

eth0.22: 2001:470:c8c6:2::1/64

brama dla eth0.22: 2001:470:1f13:68::101

eth0.33: 2001:470:1f13:68::18:1/119

brama dla eth0.33: 2001:470:1f13:68::101

klasy adresowe od: 2001:470:1f13:68::18:400/120 **do:** 2001:470:1f13:68::1b:ffff/120

```
ip -6 route add default dev tun0 metric 1 # trasa domyślna
ip -6 route add 2001:470:1f13:68::10:0/108 dev lo metric 2 # zabezpieczenie
przed zapętleniem
```

4.2 Połączenie routerów tunelem szyfrowanym

Połączenie routerów tunelem szyfrowanym jest konieczne wówczas, gdy będziemy posiadać jedną, bądź kilka klas adresowych, przypisanych do jednego, zewnętrznego adresu IPv4. Tunele są zestawione za pomocą programu **OpenVPN**. W tym przypadku łączymy ze sobą 3 miasta za pomocą IPv6. Jedno pełni rolę routera głównego (**Chorzów**) z którym zestawione jest połączenie tunelowe z www.tunnelbroker.net, a kolejne routery (**Katowice**, **Mysłowice**) łączą się z głównym tunelem szyfrowanym autoryzowanym za pomocą certyfikatów. Na routerze głównym

do tunelu, trasowane są klasy adresowe IPv6, przeznaczone do przydziału dla klientów.

Dla tunelu OpenVPN została przydzielona maska 120 bitowa, co pozwala na zaadresowanie 255 hostów. Jest to liczba zupełnie wystarczająca, ponieważ na chwilę obecną, dla tunelu, w użyciu są 3 adresy IP i liczba routerów może wzrosnąć co najwyżej o 2.

Podstawowa konfiguracja skryptów, koniecznych do ustawienia adresów IPv6 i tras dla interfejsu tunelowego tun0 dla routerów:

- **Chorzów** – router główny

```
#!/bin/bash
# zmienne pobierające nazwę interfejsu tunelowego i MTU11
INTERFACE=$1; shift;
TUN_MTU=$1; shift;

# włączenie interfejsu i ustawienie wartości MTU
ip link set ${INTERFACE} up
ip link set mtu ${TUN_MTU} dev ${INTERFACE}

# dodanie adresu IP do interfejsu tunelowego
ip -6 addr add 2001:470:1f13:68::101/120 dev ${INTERFACE}
# klasy adresowe kierowane na adres IP routera w Mysłowicach
ip -6 route add 2001:470:1f13:68::10:0/108 via 2001:470:1f13:68::102
ip -6 route add 2001:470:c8c6:2::/64 via 2001:470:1f13:68::102

# klasy adresowe kierowane na adres IP routera w Katowicach
ip -6 route add 2001:470:1f13:68::20:0/108 via 2001:470:1f13:68::103

# zakończenie skryptu, 0 oznacza prawidłowe zakończenie
exit 0
```

- **Katowice** – router łączący się do routera w Chorzowie

```
#!/bin/bash
INTERFACE=$1; shift;
TUN_MTU=$1; shift;

ip link set ${INTERFACE} up
ip link set mtu ${TUN_MTU} dev ${INTERFACE}

ip -6 addr add 2001:470:1f13:68::103/120 dev ${INTERFACE}
```

¹¹ ang. Maximum Transmission Unit - największy pakiet jaki można wysłać przez sieć

```
# brama domyślna
ip -6 route add default dev ${INTERFACE} metric 1

# zabezpieczenie przed zapętleniem klasy nie będącej w użyciu
ip -6 route add 2001:470:1f13:68::20:0/108 dev lo metric 2
```

- **Mysłowice** – router łączący się do routera w Chorzowie

```
#!/bin/bash

INTERFACE=$1; shift;
TUN_MTU=$1; shift;
ip link set ${INTERFACE} up
ip link set mtu ${TUN_MTU} dev ${INTERFACE}

ip -6 addr add 2001:470:1f13:68::102/120 dev ${INTERFACE}
ip -6 route add default dev ${INTERFACE} metric 1

ip -6 route add 2001:470:1f13:68::10:0/108 dev lo metric 2
```


5 Modyfikacje bazy danych MySQL

Do napisania modułów dla systemu LMS, konieczne było stworzenie 3 nowych tabel w istniejącej bazie MySQL. Większość pól jest obowiązkowo do uzupełnienia za pomocą formularzy. Niektóre uzupełniają się automatycznie, np.: identyfikatory, daty, wartości domyślne. Część pól jest opcjonalnych, służących jako informacja, a niektóre istnieją w bazie do dalszego rozwoju aplikacji i jak na razie nie są potrzebne w projekcie.

5.1 Opis dodanych pól dla sieci

Nazwa pola	Typ pola	Opis
id	int(11)	Identyfikator, automatycznie inkrementujący się
name	varchar(255)	Nazwa własna sieci, bez spacji
address	varchar(150)	Adres sieci
mask	int(3)	Maska sieci
clientmask	int(3)	Maska klas adresowych przeznaczonych dla klientów
eui64	enum('0', '1')	Wartość 1 oznacza sieć przeznaczoną do autokonfiguracji bezstanowej, maska sieci musi się równać 64
class_eui	int(9)	Wprowadzona sieć jest klasą adresową, która jest udostępniona klientom, jest łączona do już istniejącej sieci (której pole eui64 jest równe 1), pole zawiera id sieci docelowej przeznaczonej do autokonfiguracji bezstanowej
gateway	varchar(150)	Brama jaką muszą wpisać klienci (opcjonalnie)
gateway_int	varchar(150)	Brama dla interfejsu (opcjonalnie)
interface	varchar(8)	Interfejs od strony sieci LAN (opcjonalnie)
dns	varchar(150)	Pierwszy serwer DNS (opcjonalnie)
dns2	varchar(150)	Drugi serwer DNS (opcjonalnie)
wins	varchar(150)	Serwer WINS (opcjonalnie)

Tab. 3: Pola MySQL dla tablicy: networks6 (źródło: własne)

5.2 Opis dodanych pól dla komputerów

Nazwa pola	Typ pola	Opis
id	int(11)	Identyfikator, automatycznie inkrementujący się
name	varchar(30)	Nazwa własna komputera, bez spacji
mac	varchar(20)	Adres MAC komputera
ipaddr	varchar(150)	Adres IP
ipaddr_class	varchar(150)	Przydzielona klasa IP (opcjonalnie)
class_eui	enum('0', '1')	Wartość 1 oznacza klasę adresową, przeznaczoną dla adresu przydzielonego z autokonfiguracji bezstanowej
netid	int(4)	ID sieci
ownerid	int(11)	ID klienta
creationdate	int(11)	Data utworzenia komputera
moddate	int(11)	Data modyfikacji komputera
creatorid	int(11)	ID administratora dodającego
modid	int(11)	ID administratora modyfikującego
access	tinyint(1)	Blokada dostępu (do przyszłych zastosowań)
warning	tinyint(1)	Ostrzeżenie (do przyszłych zastosowań)
lastonline	int(11)	Czas ostatniej aktywności (do przyszłych zastosowań)
info	text	Informacje nt. komputera

Tab. 4: Pola MySQL dla tablicy: nodes6 (źródło: własne)

5.3 Opis dodanych pól dla statystyk

Nazwa pola	Typ pola	Opis
id	int(9)	Identyfikator, automatycznie inkrementujący się
time	int(14)	Czas aktualizacji statystyki
down_rate	int(30)	Prędkość pobierania
up_rate	int(30)	Prędkość wysyłania
down_pkts	int(20)	Liczba pakietów odebranych
up_pkts	int(20)	Liczba pakietów wysłanych
nodeid	int(5)	ID komputera

Tab. 5: Pola MySQL dla tablicy: stats6 (źródło: własne)

6 Modyfikacje LMS dla protokołu IPv6

6.1 Obsługa sieci IPv6

6.1.1 Dodawanie

Nowa funkcjonalność pozwala na dodawanie klas adresowych do bazy danych, wraz z zachowaniem zabezpieczeń we wprowadzaniu danych. Zabezpieczenia polegają na:

- sprawdzeniu czy nazwa sieci nie zawiera spacji i znaków specjalnych
- zamianie adresu na format pełny, po czym sprawdzeniu długość adresu jest prawidłowa
- sprawdzeniu czy wprowadzony adres jest adresem sieci
- sprawdzeniu czy adres sieci nie jest już wprowadzony do bazy
- sprawdzeniu czy maska zawiera się w przedziale od 1 do 128

Na rys. 1 przedstawiono przykład wprowadzenia adresu, nie będącego adresem sieciowym.

The screenshot shows the 'Nowa Sieć IPv6' configuration page in the LMS 1.8.14 Nesa interface. The 'Adres sieci/maska' field is highlighted in red and contains the address '2001:470:1f13:68::40:4 / 105'. A red error message box is overlaid on the address field, stating 'Wpisany adres nie jest adresem sieci'. The interface includes a sidebar with navigation options like 'Administracja', 'Klienci', 'Statystyka klientów', 'Komputery', 'Osprzęt sieciowy', 'Stany Magazynowe', 'Sieci IP', and 'Sieci IPv6'. The main form fields include 'Nazwa sieci', 'Adres sieci/maska', 'Maska dla klientów', 'EUI-64', 'Klasa dla EUI-64', 'Interfejs', 'Domena', 'Brama dla klientów', 'Brama dla interfejsu', and 'Serwery DNS'. Buttons for 'Zapisz' and 'Anuluj' are visible at the bottom right.

Rys. 1: Dodawanie sieci, komunikat o nieprawidłowym adresie (źródło: własne)

Administrator ma możliwość zdefiniowania następujących pól:

- **Nazwa sieci** – powinna zawierać nazwę sugerującą przynależność do konkretnego routera (przeważnie nazwa routera), nie może zawierać znaków specjalnych ani spacji
- **Adres sieci** – adres sieci IPv6 obojętnie w jakim formacie (dozwolony format preferowany, czyli pełny i format skrócony), po wysłaniu formularza adres zostanie zamieniony do formatu skróconego
- **Maska** – maska sieci (w przedziale od 1 do 128), przeznaczona na daną podsieć
- **Maska dla klientów** – jeśli chcemy zaoferować dodatkowe klasy adresowe, wpisujemy maskę jaka będzie dostępna dla każdego klienta, pole nie może zostać puste, jeśli nie chcemy udostępnić klientom klas adresowych, wpisujemy wartość 128
- **EUI-64** – zaznaczenie opcji oznacza iż na daną podsieć przeznaczamy klasę o długości 64 bitów, a klasa będzie dostępna dla autokonfiguracji bezstanowej, adresy będą rozdystrybuowane za pomocą programu **Radvd**; po zaznaczeniu tej opcji musimy wprowadzić maskę równą 64 (pole **Maska**) i wprowadzić **Maskę dla klientów** równą 128
- **Klasa dla EUI-64** – pole wyboru, w przypadku gdy chcemy zaoferować klientom klasę adresową, którym adresy przyznawane są za pomocą autokonfiguracji bezstanowej ze skonfigurowanej wcześniej klasy EUI-64, wybieramy istniejącą sieć EUI-64
- **Interfejs** – nazwa interfejsu
- **Domena** – pole opcjonalne, służące do celów informacyjnych
- **Brama dla klientów** – brama domyślna, którą trzeba skonfigurować na hostach (za wyjątkiem adresów EUI-64), pole opcjonalne, służące do celów informacyjnych
- **Brama dla interfejsu** – brama, którą należy skonfigurować na interfejsach routera, pole opcjonalne, służące do celów informacyjnych
- **Serwery DNS** – pole opcjonalne, służące do celów informacyjnych

Mamy możliwość wprowadzenia następujących sieci:

- Sieć udostępniająca klientom jedynie adres IPv6, który muszą sobie skonfigurować statycznie, ręcznie, lub za pomocą skryptu, którego kod

generowany jest z poziomu panelu administracyjnego systemu LMS. Konieczne zdefiniowanie **maski dla klientów** o wartości równej 128 (rys. 2).

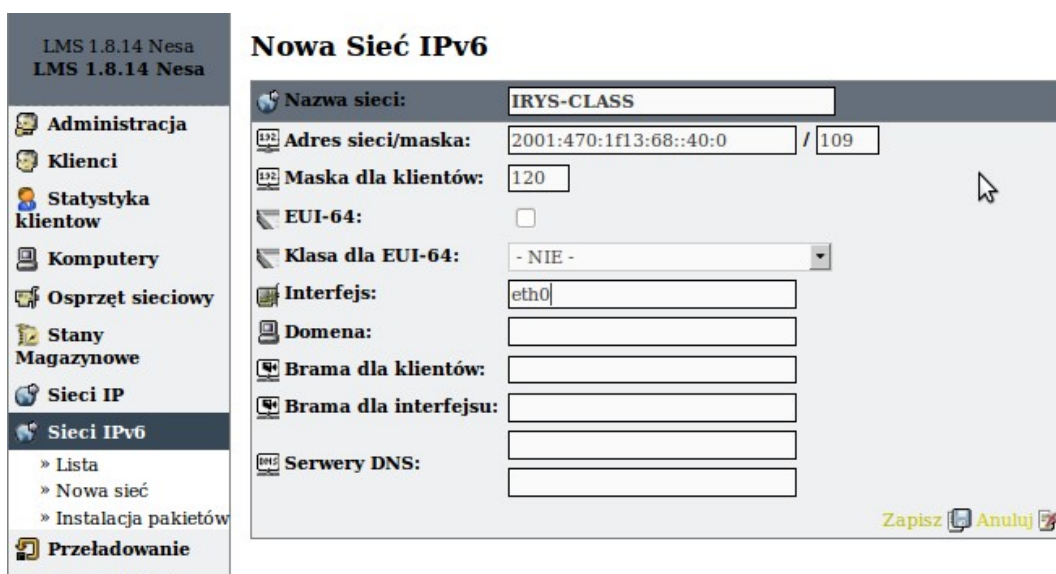
The screenshot shows the 'Nowa Sieć IPv6' configuration page in the LMS 1.8.14 Nesa interface. The sidebar on the left contains the following menu items: Administracja, Klienci, Statystyka klientów, Komputery, Osprzęt sieciowy, Stany Magazynowe, Sieci IP, Sieci IPv6 (with sub-items: » Lista, » Nowa sieć, » Instalacja pakietów), and Przeladowanie. The main content area is titled 'Nowa Sieć IPv6' and contains the following fields:

- Nazwa sieci: IRYS-SIMPLE-ETH0
- Adres sieci/maska: 2001:470:1f13:68::40:0 / 109
- Maska dla klientów: 128
- EUI-64:
- Klasa dla EUI-64: - NIE -
- Interfejs: eth0
- Domena: [empty]
- Brama dla klientów: [empty]
- Brama dla interfejsu: [empty]
- Serwery DNS: [empty]

At the bottom right of the form, there are buttons for 'Zapisz' (Save), 'Anuluj' (Cancel), and a close icon.

Rys. 2: Dodawanie sieci, przykładowo wypełnione pola, dla pojedynczych adresów
(źródło: własne)

- Sieć która będzie posiadała możliwość udostępnienia klientom klas adresowych, co oznacza zdefiniowanie **maski dla klientów**, nadając jej wartość adekwatną do przestrzeni jaką chcemy udostępnić (rys. 3). Klienci adres muszą skonfigurować statycznie, ręcznie, lub za pomocą skryptu, którego kod generowany jest z poziomu panelu administracyjnego systemu LMS



Rys. 3: Dodawanie sieci, przykładowa konfiguracja dla klas adresowych (źródło: własne)

W przypadku wybrania takiej konfiguracji sieci, następuje podział przestrzeni adresowej jaką mamy do dyspozycji (**Adres sieci/maska**) na 2 części. Część pojedynczych adresów IPv6, obowiązkowo przypisanych do hosta klienta, komunikującego się z siecią IPv6. Część klasowa (o długości bitów skonfigurowanej w polu **Maska dla klientów**), która jest trasowana na pojedynczy adres IPv6 przypisany do hosta klienta.

Liczba adresów jaką możemy uzyskać wyliczana jest ze wzoru:

$$2^{(\text{Maska dla klientów} - \text{maska})}$$

Wynik daje optymalny rezultat i daje gwarancję, że zawsze liczba pojedynczych adresów IP będzie większa od liczby klas adresowych, które powodują ogólne zmniejszenie przestrzeni adresowej w ramach całej klasy do dyspozycji.

Maska sieci dla pojedynczych adresów IP wyliczana jest ze wzoru:

$$128 - (\text{Maska dla klientów} - \text{maska})$$

Wzór pozwalający obliczyć liczbę klas adresowych:

$$(2^{(\text{Maska dla klientów} - \text{maska})}) - (2^{((128 - (\text{Maska dla klientów} - \text{maska})) - \text{maska})})$$

Liczba uzyskana z ostatniego wzoru daje ogólną liczbę możliwych do przypisania dla klientów adresów IP i klas adresowych, ponieważ podczas dodawania adresu nadawany jest pojedynczy adres IPv6 razem z klasą. Przykład podziału sieci, wg powyższych wzorów znajduje się na rys. 4.

Klient, posiadający router, może klasę rozdysponować wg własnego uznania we własnej sieci lokalnej. Adresy pojedyncze i klasy adresowe są generowane automatycznie, wybierając pierwszy wolny w bazie adres lub klasę.

Informacje o Sieci: MYS-IP6-SEG1

MYS-IP6-SEG1 (0002)

Adres/maska; maska dla klientów:	2001:470:1f13:68::10:0/110; 120
Adres sieci dla klientów:	2001:470:1f13:68::10:0/118 (1022 hostów)
Klasy sieci dla klientów:	Od: 2001:470:1f13:68::10:400/120 Do: 2001:470:1f13:68::13:ffff/120 Łącznie dostępnych: 768 sieci
EUI-64:	Nie
Klasa dla EUI-64:	Nie
Interfejs:	eth0.11
Brama dla klientów:	
Brama dla interfejsu:	
Serwer WINS:	
Serwery DNS:	
Domena:	

Edytuj Usun

Rys. 4: Szczegóły sieci, pokazano podział przestrzeni pojedynczych adresów i klas adresowych (źródło: własne)

- Sieć która będzie przydzielała adresy IPv6 automatycznie na zasadzie autokonfiguracji bezstanowej za pomocą programu **Radvd** (przykładowa konfiguracja na rys. 5). W przypadku wybrania tej opcji, konieczne jest zaznaczenie opcji **EUI-64** i wpisanie maski równej **64**. Po wybraniu opcji **EUI-64**, ukrywana jest lista z wyborem **klasy** dla **EUI-64**, ponieważ adresy pojedyncze nie mogą być jednocześnie klasą. Na hostach nie potrzebna jest dodatkowa konfiguracja, adres IP jest przydzielany zaraz po włączeniu interfejsu. Proces generowania adresu został omówiony w rozdziale 1.1.

LMS 1.8.14 Nesa
LMS 1.8.14 Nesa

- » Administracja
- » Klienci
- » Statystyka klientów
- » Komputery
- » Osprzęt sieciowy
- » Stany Magazynowe
- » Sieci IP
- » Sieci IPv6
- » Lista
- » Nowa sieć

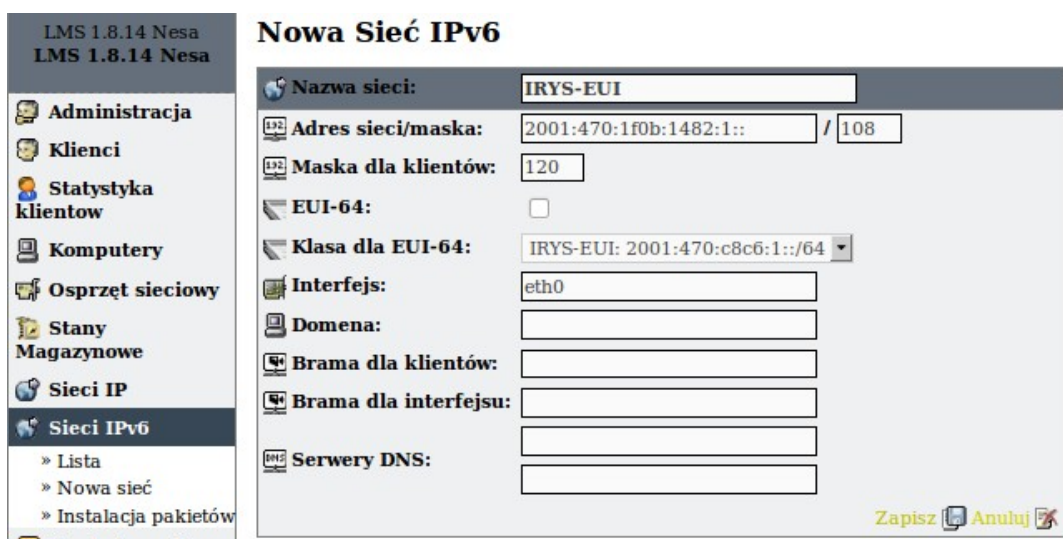
Nowa Sieć IPv6

Nazwa sieci:	<input type="text" value="IRYS-EUI"/>	
Adres sieci/maska:	<input type="text" value="2001:470:c8c6:1::"/>	<input type="text" value="64"/>
Maska dla klientów:	<input type="text"/>	
EUI-64:	<input checked="" type="checkbox"/>	
Interfejs:	<input type="text" value="eth0"/>	
Domena:	<input type="text"/>	
Brama dla klientów:	<input type="text"/>	
Brama dla interfejsu:	<input type="text"/>	
Serwery DNS:	<input type="text"/>	

Zapisz

Rys. 5: Dodawanie sieci, przykładowana konfiguracja dla autokonfiguracji bezstanowej (źródło: własne)

Do istniejącej sieci w formacie **EUI-64** możemy utworzyć klasę adresową (rys. 6), udostępniając klientom pulę adresowe. W tym celu wpisujemy adres sieci do dyspozycji wraz z **maską** i **Maską dla klientów**. Z listy **Klasa dla EUI-64** należy wybrać zdefiniowaną wcześniej sieć w formacie **EUI-64**, dla której ta klasa będzie dostępna. Klasy adresowe można dołączać klientom ręcznie, udostępniając je tylko tym, którzy będą nimi zainteresowani.



Rys. 6: Dodawanie klasy dla sieci w formacie EUI-64 (źródło: własne)

Wszystkie nowe sieci dopisywane są do bazy danych MySQL, identyfikator nadawany jest automatycznie.

6.1.2 Informacje/Edycja

Po prawidłowym wprowadzeniu sieci do bazy, możemy wyświetlić informacje na jej temat. W tym celu wchodzimy do menu Sieci IPv6, wybierając podstronę Lista (rys. 7).

Nazwa:	ID	Serwery DNS:	Brama:
Sieć/maska/maska dla klientów (interfejs)	Pule adresowe (wszystkie/wolne):		
MYS-IP6-SEG1	(0002)		
2001:470:1f13:68::10:0 / 110 / 120 (eth0.11)	768/767		
MYS-IP6-SEG2	(0003)		
2001:470:1f13:68::14:0 / 110 / 120 (eth0.22)	768/768		
MYS-IP6-SEG3	(0004)		
2001:470:1f13:68::18:0 / 110 / 120 (eth0.33)	768/768		
KAT-IP6	(0005)		
2001:470:1f13:68::20:0 / 109 / 120 (eth0.22)	1792/1792		
IRYS-IP6	(0006)		
2001:470:1f13:68::8:0 / 109 / 120 (eth0)	1792/1791	2001:470:1f13:68::8:1	
IRYS-IP6-SIMPLE	(0008)		
2001:470:1f13:68::28:0 / 116 / - (eth0)	4095/4094		
IRYS-EUI-CLASS	(0015)		
2001:470:1f13:68::40:0 / 109 / 120 (eth0)	1792/1792		
IRYS-EUI	(0014)		
2001:470:c8c6:1:: / 64 / - (eth0)	EUI-64		
Razem 11773/11772			

Rys. 7: Informacje podstawowe, nt. wprowadzonych sieci (źródło: własne)

Wyświetlone zostają wszystkie wprowadzone sieci i ich najważniejsze informacje. Ikony w ostatniej kolumnie informacji o sieci służą do różnych operacji jakie można wykonać na danym rekordzie:

- lewa ikona – usunięcie sieci
- środkowa ikona – edycja sieci (rys. 8)
- prawa ikona – szczegółowe informacje nt sieci; wyświetlenie informacji może odbyć się również po kliknięciu na obszar w obrębie podświetlonej sieci

Edycja sieci: MYS-IP6-SEG1

The screenshot shows a web-based configuration form for a network segment. The title is 'Edycja sieci: MYS-IP6-SEG1'. The form contains the following fields and options:

- Nazwa:** MYS-IP6-SEG1 (0002)
- Adres/maska; maska dla klientów:** 2001:470:1f13:68::10:0 / 110 ; 120
- EUI-64:**
- Klasa dla EUI-64:** - NIE - (dropdown menu)
- Interfejs:** eth0.11
- Brama dla klientów:** 2001:470:1f13:68::10:1
- Brama dla interfejsu:** 2001:470:1f13:68::101
- Serwer WINS:** (empty field)
- Serwery DNS:** (empty field)
- Domena:** (empty field)

At the bottom right, there are three buttons: 'Zapisz' (Save), 'Usuń' (Delete), and 'Anuluj' (Cancel).

Rys. 8: Edycja sieci (źródło: własne)

Edycję przeprowadzamy z zachowaniem tych samych reguł co dodawanie nowej sieci. W przypadku wprowadzenia błędnych danych (zły adres sieci, błędny format adresu IPv6, maska sieci poza zakresem), formularz edycji zostanie wyświetlony ponownie, razem z informacjami wyszczególnionymi w kolorze czerwonym.

Żeby zaakceptować wprowadzone zmiany w sieci, konieczne jest użycie opcji **Zapisz**. Naciśnięcie tej opcji spowoduje aktualizację rekordu edytowanej sieci i przejście na stronę informacji szczegółowych sieci.

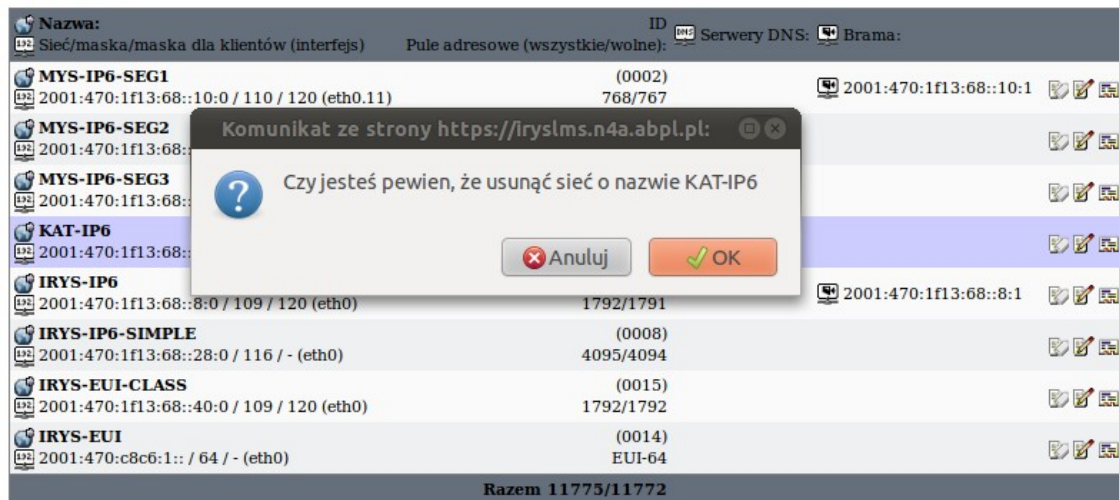
Opcja **Anuluj** służy do powrotu, do strony informacyjnej sieci, bez zapisania wprowadzonych zmian w formularzu.

6.1.3 Usuwanie

Usunąć sieć można na 2 sposoby. Pierwszym jest wylistowanie wprowadzonych sieci, po czym kliknięcie na pierwszą ikonę z lewej strony w ostatniej kolumnie. Pojawi się komunikat z pytaniem czy na pewno chcemy usunąć sieć (rys. 9),

na wypadek pomyłki.

Sieci IPv6



Rys. 9: Komunikat pojawiający się podczas usuwania sieci (źródło: własne)

Należy mieć na uwadze fakt, iż usunięcie sieci spowoduje bezpowrotne usunięcie hostów z tej sieci. Wszelkie nierozważne decyzje mogą spowodować niezadowolenie wśród klientów, korzystających z sieci IPv6, poprzez błąd administratora.

Drugim, a zarazem ostatnim sposobem usuwania sieci jest wejście do szczegółów sieci. Znajduje się tam opcja w formie tekstowej z oznaczeniem graficznym. Działanie jest identyczne jak w poprzednim sposobie, po naciśnięciu wyświetli się komunikat z prośbą o potwierdzenie swojej decyzji. Po zatwierdzeniu następuje usunięcie komputerów z bazy, należących do usuwanej sieci.

6.1.4 Klasy adresowe dla konfiguracji statycznej

W przypadku wybrania konfiguracji statycznej sieci, mamy możliwość wyboru, czy chcemy udostępnić klientom do dyspozycji klasę adresowe, czy nie. W obydwu przypadkach klient musi samodzielnie skonfigurować na hoście adres IP. Jeśli będziemy udostępniać klasę adresowe, wówczas następuje podział przestrzeni adresowej (**Adres sieci/maska**) na 2 części. Część pojedynczych adresów IPv6, obowiązkowo przypisanych do hosta klienta, komunikującego się z siecią IPv6. Część klasowa (o długości bitów skonfigurowanej w polu **Maska dla klientów**), która jest

trasowana na pojedynczy adres IPv6 przypisany do hosta klienta.

W przypadku systemu Windows XP konfiguracja adresu IPv6 jest utrudniona. Protokół IPv6 domyślnie nie jest zainstalowany w systemie, dlatego trzeba go doinstalować poleceniem, wywołanym z wiersza poleceń:

```
netsh interface ipv6 install
```

Proces dodawania adresu IPv6 i bramy domyślnej, również jest utrudniony z uwagi na brak opcji edycji samego protokołu w opcjach połączenia. Dodanie adresu i bramy, robimy poprzez wywołanie poleceń z wiersza poleceń:

```
netsh interface ipv6 add address "Połączenie lokalne" 2001:470:1f0a:1484::10:2
```

Nazwa w cudzysłowie jest nazwą interfejsu, można ją odczytać otwierając okno zarządzania połączeniami. Widać tam wszystkie urządzenia związane z siecią. Należy zwrócić uwagę, iż przy dodawaniu adresu IPv6, nie potrzebna jest maska sieci.

```
netsh interface ipv6 add route 2001::/3 "Połączenie lokalne"  
2001:470:1f13:68::10:1
```

Bramę dodajemy wpisując adres klasy globalnej, nazwę połączenia (przeważnie będzie to „Połączenie lokalne” w przypadku połączenia kablowego do sieci Ethernet) i adres routera.

6.1.5 Klasy adresowe dla autokonfiguracji bezstanowej (stateless)

Jest to najdogodniejsza forma podłączenia klientów do sieci IPv6, ponieważ na hostach nie potrzeba żadnej specjalnej konfiguracji. Adres jest otrzymywany zaraz po włączeniu interfejsu, pod warunkiem, że sieć działa prawidłowo.

6.2 Obsługa komputerów IPv6

6.2.1 Dodawanie

Administrator sieci, obsługujący system LMS, musi mieć możliwość dopisywania komputerów do kont klientów. Nowa funkcjonalność to umożliwia. Aby dodać komputer, należy wejść na konto klienta, któremu chcemy dopisać adres IPv6

(rys. 10).

Informacje o kliencie: ██████████ Tomasz

The screenshot shows a client information page for 'Tomasz (0906)'. On the left, there is a sidebar with icons for connection status (podłączony), PESEL, Saldo (0,00 zł), Klient z CC (Nie), Pokaz w statystyce (Nie), and creation/modification dates. The main area is divided into two sections: 'Komputery klienta (1):' and a list of computers. The list includes:

- 80-238-70-66**: 80.238.70.66 / 00:0F:EA:EC:FC:6D (2648)
- IRYS-IP6-9**: 2001:470:1f0a:1484::8:2 / 117 | 00:1F:F3:D6:CE:07 (0001)
- IRYS-IP6-SIMPLE-82021**: 2001:470:1f0a:1484::28:2 / 116 | 00:30:4F:1B:8B:4B (0008)
- MYS-IP6-SEG1-871**: 2001:470:1f0a:1484::10:2 / 118 | 00:24:1D:DB:E0:58 (0009)
- IRYS-EUI-17999**: 2001:470:c8c6:1:21c:10ff:fec4:8c51 / 64 | 00:1C:10:C4:8C:51 (0014)

At the bottom of the list, there are buttons: 'Podłącz wszystkie', 'Odłącz wszystkie', 'Włącz ostrzeżenia wszystkim', 'Wyłącz ostrzeżenia wszystkim', 'Nowy IPv4', and 'Nowy IPv6'.

Rys. 10: Adresy IP przypisane do klienta (źródło: własne)

W prawej części informacji o kliencie, jest wyświetlona lista komputerów z przypisanymi adresami IPv4 i IPv6. Dodanie komputera umożliwia opcja **Nowy IPv6**.

The screenshot shows a form for adding a computer. The fields are:

- Nazwa**: [text input]
- Wybierz sieć**: [dropdown menu] IRYS-IP6-SIMPLE: 2001:470:1f13:68::28:0/116
- Adres MAC**: [text input]
- Status**: [dropdown menu] podłączony
- Klient**: [dropdown menu] ██████████ Tomasz (0906)
- Urządzenie sieciowe**: [dropdown menu] - wybierz urządzenie -
- Opis**: [text area]

At the bottom, there are buttons: 'Skanuj', 'Zapisz', 'Anuluj', and a checkbox: 'Po dodaniu komputera wyświetl ten formularz ponownie'.

Rys. 11: Formularz dodawania adresu IPv6 (źródło: własne)

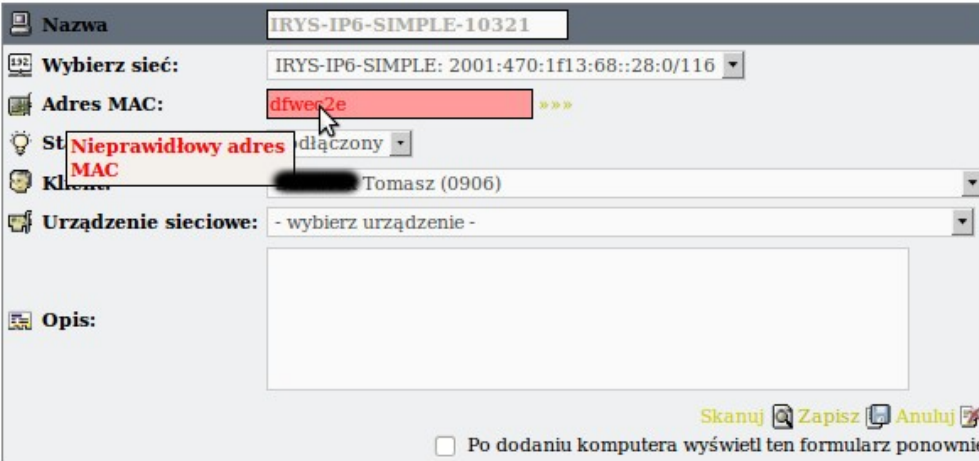
Podczas dodawania komputera (rys. 11) konieczny jest wybór sieci, do której dołączony będzie komputer. Sieć musi być wcześniej skonfigurowana w systemie LMS. Podajemy również adres MAC karty sieciowej hosta. Nazwa jest generowana automatycznie na podstawie nazwy sieci, dodając na końcu numer generowany losowo w przedziale 1 – liczba możliwych adresów IP do przypisania, wcześniej sprawdzając, czy nazwa nie jest już w użyciu.

Pole z wyborem o nazwie **Status**, służy do blokowania internetu dla komputera. Zawiera dwie opcje do wyboru: **podłączony**, **odłączony**. Na chwilę obecną powoduje tylko zapis tej opcji do bazy danych, bez żadnych konsekwencji w zakresie blokady. Pole stworzone dla przyszłych zastosowań.

Pole z wyborem o nazwie **Klient** umożliwia wybrania klienta, dla którego dodajemy komputer. Domyślnie jest to klient, którego dane znajdują się z lewej części strony.

Dzięki wyborowi **Urządzenia sieciowe**, mamy możliwość podłączenia komputera do konkretnego urządzenia sieciowego, wcześniej zdefiniowanego.

W polu **Opis**, możemy zawrzeć informacje nt komputera, zwykle bardzo pomocne dla administratora, jak na przykład dane dostępu do routera klienta, jeśli to administrator go konfigurował, a klient zapomniałby hasła.



The screenshot shows a web form with the following fields and values:

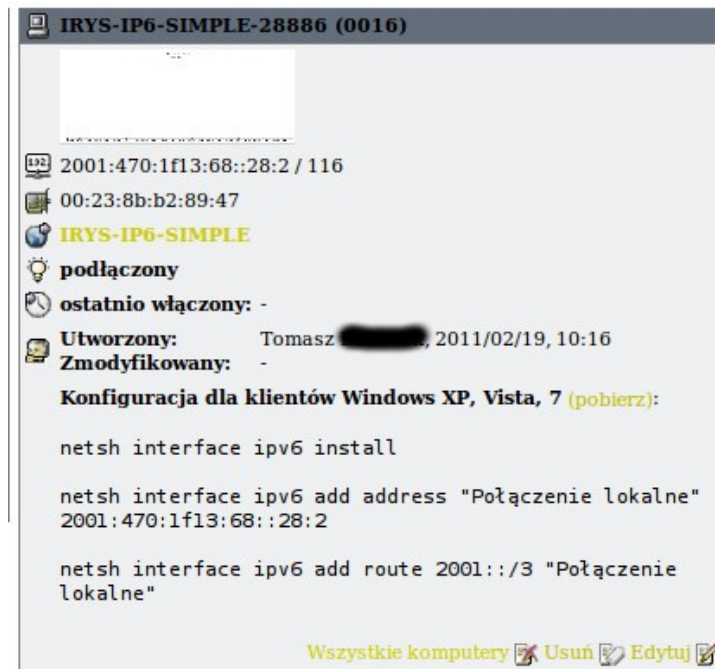
- Nazwa:** IRY5-IP6-SIMPLE-10321
- Wybierz sieć:** IRY5-IP6-SIMPLE: 2001:470:1f13:68::28:0/116
- Adres MAC:** dfwer2e (highlighted in red with a tooltip: "Nieprawidłowy adres MAC")
- Status:** podłączony
- Klient:** Tomasz (0906)
- Urządzenie sieciowe:** - wybierz urządzenie -
- Opis:** (empty text area)

At the bottom of the form, there are buttons: "Skanuj", "Zapisz", and "Anuluj". Below these buttons is a checkbox labeled "Po dodaniu komputera wyświetl ten formularz ponownie".

Rys. 12: Komunikat błędnego wprowadzenia adresu MAC (źródło: własne)

Po wysłaniu wypełnionego formularza, na poziomie języka PHP, sprawdzany jest adres mac. Jeśli jest on nieprawidłowy, pole z miejscem na adres MAC zostanie podświetlone na czerwono, a po najechnaniu na nie, wyświetli się komunikat z przyczyną problemu (rys. 12). Adres MAC musi być wprowadzony wg formatu z dwukropkami:

00:11:22:33:44:55



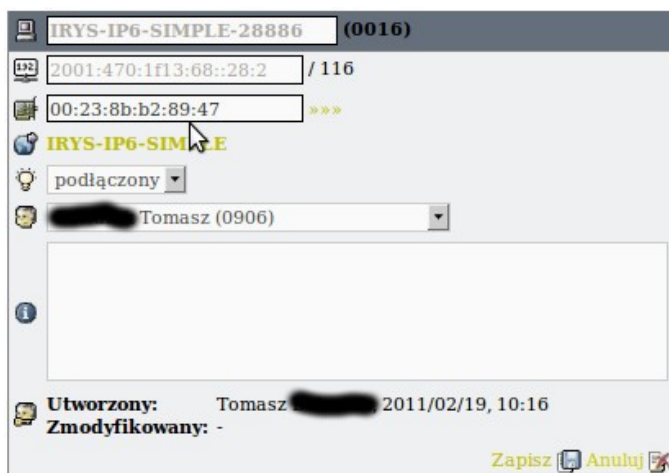
Rys. 13: Szczegółowe informacje nt. wprowadzonego komputera (źródło: własne)

Po prawidłowym wprowadzeniu komputera, otworzy się podstrona ze szczegółowymi informacjami (rys. 13). Na samej górze, znajduje się miniatura z odnośnikiem do statystyk użycia łącza w ciągu ostatnich 24 godzin. Znajdziemy też informacje kto wprowadził komputer i go zaktualizował, podając przy tym dokładną godzinę.

Poniżej są 3 polecenia, które musi wprowadzić użytkownik systemu Windows XP. Użytkownicy pozostałych systemów z rodziny Windows i większości dystrybucji linuxowych, mają możliwość konfiguracji graficznej. Polecenia działają na systemach Windows XP, Vista, 7. Dla opornych we wpisywaniu poleceń, przygotowano plik bat, generowany dla komputerów za pomocą skryptu PHP. Plik ten można pobrać poza strefą autoryzacji systemu LMS.

6.2.2 Edycja

Edycja komputerów jest obowiązkowym elementem umożliwiającym zmianę jego ustawień. Edycji dokonujemy poprzez kliknięcie na ikonie lub odnośniku tekstowym.



Rys. 14: Edycja komputera (źródło: własne)

Przykładowa zawartość formularza edycji na rys. 14. Pola: **Nazwa komputera** i **adres IP** są wyłączane z edycji, ponieważ nadawane są automatycznie podczas dodawania komputera. Wyjątek stanowi adres w formacie **EUI-64**, który generuje się od nowa, jeśli zmieni się adres MAC. Wypełnienie pola MAC podlega tym samym zasadom, co dodawanie komputerów.

Poniżej pola **MAC**, jest odnośnik do szczegółowych informacji sieci, do której przypisany jest komputer. Pole dotyczące blokady, powoduje jedynie zapis wartości do bazy MySQL, nie blokując faktycznie dostępu do internetu, opcja stworzona do przyszłych zastosowań. Podczas edycji, możliwa jest zmiana klienta, do którego przypisany jest komputer, jak również zmiana opisu.

Po aktualizacji wszystkich pól, należy zapisać konfigurację, poprzez odnośnik **Zapisz**. Wysłany formularz zostanie sprawdzony pod kątem poprawnego wprowadzenia danych, a wszystkie błędy zostaną zakomunikowane i oznaczone czerwonym kolorem. Odnośnik **Anuluj** służy do wyjścia z okna Edycji komputera, bez zapisu zaktualizowanych pól.

6.2.3 Usuwanie

Usunięcie komputera powoduje trwałe skasowanie rekordu z bazy danych jak i statystyk użycia łącza.

Usunięcie komputera możemy wykonać poprzez ikonę usuwania, widoczną w ostatniej kolumnie z lewej strony lub wejście do szczegółów sieci, gdzie znajduje się odnośnik w formie tekstowej. Po wybraniu któregośkolwiek odnośnika pojawi się komunikat z prośbą o potwierdzenie swojej decyzji. Po zatwierdzeniu następuje usunięcie komputera z bazy i statystyk do niego należących.

6.2.4 Klasy adresowe

Klient w zależności od konfiguracji sieci, może wykorzystać klasę adresową. Jeśli jest to sieć, skonfigurowana dla konfiguracji statycznej z podziałem na adresy pojedyncze i klasy adresowe o określonej liczbie bitów, do komputera klienta automatycznie przypisana jest klasa adresowa. Adres sieci klasy adresowej i maska, podane są w szczegółowych informacjach komputera.

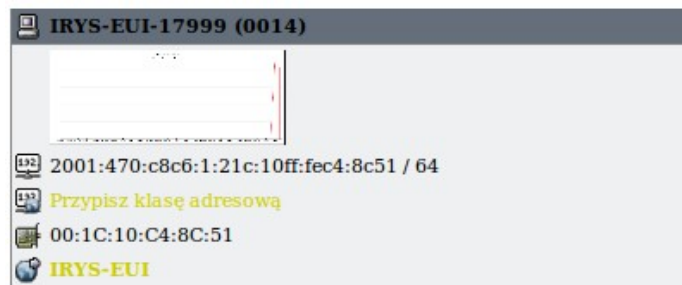


*Rys. 15: Klasa adresowa przypisana do adresu IP
(źródło: własne)*

Na przykładzie rys. 15, klasa adresowa to: **2001:470:1f0a:1484::8:800/120**, która została przypisana automatycznie, odwołując się do bazy **MySQL**, biorąc pierwszy, wolny zakres.

Jeśli jest to sieć, skonfigurowana dla autokonfiguracji bezstanowej (stateless), klasę adresową możemy przypisać ręcznie lub ją odłączyć w podglądzie szczegółowym komputera. Po kliknięciu w odnośnik **Przypisz klasę adresową** (rys. 16), zostanie

pobrana pierwsza, wolna sieć dla komputera. Odnośnik pojawi się tylko i wyłącznie wtedy, gdy wprowadzimy klasę adresową dla adresów w formacie **EUI-64**.



Rys. 16: Adres IP w formacie EUI-64, bez przypisanej klasy adresowej (źródło: własne)

Przypisany adres sieciowy z maską zostanie wyświetlony w miejscu Przypisz klasę adresową. Obok adresu, pojawi się odnośnik **Odlącz klasę adresową**, służący do usunięcia klasy adresowej przypisanej do komputera z bazy **MySQL**.

Klasy adresowe są przypisane do pojedynczego adresu IPv6 (w przykładzie na rys. 16 jest to: 2001:470:c8c6:1:21c:10ff:fec4:8c51), co oznacza, iż możliwość użycia tej klasy nastąpi wówczas, gdy klient na swoim hoście przypisze ten adres do głównego interfejsu. Aby klasa działała prawidłowo, klient na swoim routerze musi przypisać do interfejsu głównego pojedynczy adres IPv6 (w przypadku autokonfiguracji bezstanowej nie musi tego robić, ponieważ adres przypisze się automatycznie), włączyć przekazywanie pakietów IPv6 i przypisać do interfejsu wyjściowego swojego routera (interfejs własnej sieci lokalnej), adres IPv6, mieszczący się w zakresie udostępnionej mu klasy adresowej. Jeśli klient ma router posiadający więcej portów lokalnych, adres IPv6 z klasy adresowej powinien przypisać do mostu (ang. bridge), dla systemów linuxowych (np. z oprogramowaniem **DD-WRT**) interfejs będzie oznaczony jako **br0**.

```

root@seru:~# ip -6 a show wlan1
6: wlan1: <BROADCAST,MULTICAST,UP>
   inet6 fe80::21c:10ff:fec4:8c51/64 scope link
   inet6 2001:470:c8c6:1:21c:10ff:fec4:8c51/64 scope global deprecated dynamic
      valid_lft 1122157sec preferred_lft -865043sec
root@seru:~# ip -6 a show br0
8: br0: <BROADCAST,MULTICAST,PROMISC,UP>
   inet6 2001:470:1f13:68::40:1/120 scope global
   inet6 fe80::21c:10ff:fec4:8c50/64 scope link
root@seru:~# ip -6 route
2001:470:1f13:68::40:0/120 dev br0 metric 256 mtu 1500 advmss 1440
2001:470:c8c6:1::/64 dev wlan1 proto kernel metric 256 expires 1122150sec mtu 1500 advmss 1440
2000::/3 via 2001:470:c8c6:1::1 dev wlan1 metric 1024 mtu 1500 advmss 1440
fe80::/64 dev eth0 metric 256 mtu 1500 advmss 1440

```

Rys. 17: Przykładowa konfiguracja routera klienta (źródło: własne)

Na rys. 17 pokazano konfigurację routera klienta (router testowy oznaczony jest modelem: Linksys WRT54G-TM z wgranym oprogramowaniem DD-WRT), **wlan1** jest wirtualnym interfejsem, do którego przypisany jest port oznaczony jako **WAN**, poniżej gniazda RJ45. Adres IPv6 został przyznany automatycznie. Porty sieci lokalnej zmostkowane są z interfejsem o nazwie **br0**. Do tego interfejsu należy przypisać wolny adres IPv6, z maską, w obrębie klasy adresowej jak pokazano na rys. 17.

```

seru@seru:~$ ip -6 a show dev wlan0
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
   inet6 2001:470:1f13:68::40:3/120 scope global
      valid_lft forever preferred_lft forever
   inet6 fe80::221:63ff:fe38:dccl/64 scope link
      valid_lft forever preferred_lft forever
seru@seru:~$ ip -6 route
2001:470:1f13:68::40:0/120 dev wlan0 proto kernel metric 256 mtu 1500 advmss 1440 hoplimit 0
fe80::/64 dev wlan0 proto kernel metric 256 mtu 1500 advmss 1440 hoplimit 0
default via 2001:470:1f13:68::40:1 dev wlan0 proto static metric 1024 mtu 1500 advmss 1440 hoplimit 0
seru@seru:~$ traceroute6 ripe.net
traceroute to ripe.net (2001:610:240:22::c100:68b) from 2001:470:1f13:68::40:3, 30 hops max, 16 byte packets
 1 2001:470:1f13:68::40:1 (2001:470:1f13:68::40:1)  4.455 ms  0.826 ms  4.759 ms
 2 2001:470:c8c6:1::1 (2001:470:c8c6:1::1)  3.159 ms  7.214 ms  9.706 ms
 3 e4r-1.tunnel.tserv10.par1.ipv6.he.net (2001:470:1f12:68::1)  51.716 ms  53.369 ms  54.735 ms
 4 gige-g2-3.core1.par1.he.net (2001:470:0:7b::1)  53.28 ms  52.163 ms  49.668 ms
 5 10gigabitethernet1-3.core1.lon1.he.net (2001:470:0:42::1)  64.431 ms  63.731 ms  64.405 ms
 6 10gigabitethernet1-1.core1.ams1.he.net (2001:470:0:3f::2)  66.943 ms  65.296 ms  70.032 ms
 7 gw.ipv6.amsix.nikrtr.ripe.net (2001:7f8:1::a500:3333:1)  65.687 ms !S  94.983 ms !S  65.199 ms !S
seru@seru:~$

```

Rys. 18: Testowa konfiguracja komputera klienta i test działania (źródło: własne)

Komputery będące w obrębie sieci lokalnej, podłączonej do routera klienta, powinny mieć nadany adres IPv6 z klasy adresowej, inny niż router klienta. Na przykładzie komputera testowego (rys. 18), jest to adres **2001:470:1f13:68::40:3** z maską **120**. Jako adres bramy musi być ustawiony adres IPv6 routera, na przykładzie komputera testowego jest to: **2001:470:1f13:68::40:1**. Po poprawnej konfiguracji,

komputery będące za routerem, będą miały dostęp do internetu i co najważniejsze, mają publiczne adresy IPv6. Maszyna testowa na rys. 18, posiada dostęp do internetu, za pośrednictwem protokołu IPv6, co potwierdza test wykonany poleceniem **tracert6**. Świadczy to o prawidłowej konfiguracji obu urządzeń klienta. Należy zwrócić uwagę, że pierwszym adresem przeskoku jest router testowy, a kolejny adres przeskoku to router główny, oznacza to, iż ruch IPv6 przechodzi przez router testowy.

6.3 Statystyki komputerów

Statystyki wykorzystania łącza są formą prezentacji graficznej informacji, zapisanej w bazie danych MySQL. Klient, gdy pobiera lub wysyła dane, informacje nt. ilości pobranych/wysłanych bajtów zapisywane są w **ip6tables**, po uprzedniej konfiguracji. Na rys. 19 w 2 kolumnie widać zapisane informacje nt. ilości danych, pasujących do danej reguły. Na tej podstawie są tworzone statystyki liczące ruch.

```
[ 21:18:43 ] root@irys:/usr/share/lms/lib$ ip6tables -v -nL forward_class
Chain forward_class (1 references)
pkts bytes target prot opt in out source destination
0 0 ACCEPT all * * 2001:470:c8c6:1:21c:10ff:fec4:8c51/128 :/0 MAC 00:1C:10:C4:8C:51
0 0 ACCEPT all * * :/0 2001:470:c8c6:1:21c:10ff:fec4:8c51/128
11 1144 ACCEPT all * * 2001:470:1f13:68::40:0/120 :/0 MAC 00:1C:10:C4:8C:51
0 0 ACCEPT all * * :/0 2001:470:1f13:68::40:0/120
0 0 ACCEPT all * * 2001:470:c8c6:1:64b:80ff:fe80:8003/128 :/0 MAC 04:4B:80:80:80:03
0 0 ACCEPT all * * :/0 2001:470:c8c6:1:64b:80ff:fe80:8003/128
```

Rys. 19: Zapisywane na bieżąco informacje statystyczne pobranych/wysłanych danych
(źródło: własne)

W takim celu został stworzony skrypt, który wywołuje komendę wraz z parametrami widocznymi na rys. 19, dodatkowo zerując informacje statystyczne. Skrypt jest wywoływany co 30 minut, z poziomu harmonogramu zadań (**cron**) systemu LMS. Skrypt jest napisany z języku PHP. Aby to wywołanie było możliwe konieczna była wcześniejsza konfiguracja. Po wejściu do opcji konfiguracyjnych i wybraniu odnośnika **Demon**, możemy dodawać, edytować, usuwać instancje. Skupimy się na dodaniu instancji.

Nazwa: statsChorzow

Host: irys

Modul: /usr/share/lms/lib/parser.so

Crontab: */30 ****

Priorytet: 8

Opis:

Status: Wyłączony

Zapisz Anuluj

Rys. 20: Instancja statystyk z wypełnionymi polami
(źródło: własne)

Rys. 20 zawiera dane zdefiniowane dla routera głównego w **Chorzowie**. Pole **Host** oznacza router, którego dotyczy konfiguracja. Pole **Moduł** musi zawierać prawidłową ścieżkę do modułu **parser.so**, służącego do zmiany zawartości plików na routerze i wykonywanie poleceń z parametrami, np. restart **serwera DHCP**. Pole **Crontab** to Harmonogram zadań, czas wywoływania podaje się w takim samym standardzie jak robi się to w systemie Linuks. **Priorytet** powinien zawierać najwyższą możliwą liczbę, ponieważ statystyki nie są najważniejszą instancją, podczas przeładowania konfiguracji w LMS-ie.

Konfiguracja instancji: statsChorzow/irys

Nazwa:	Wartość:	Opis:	Dodaj opcję
command	/usr/share/lms/lib/stats		
file	/dev/null		
script	tmp		

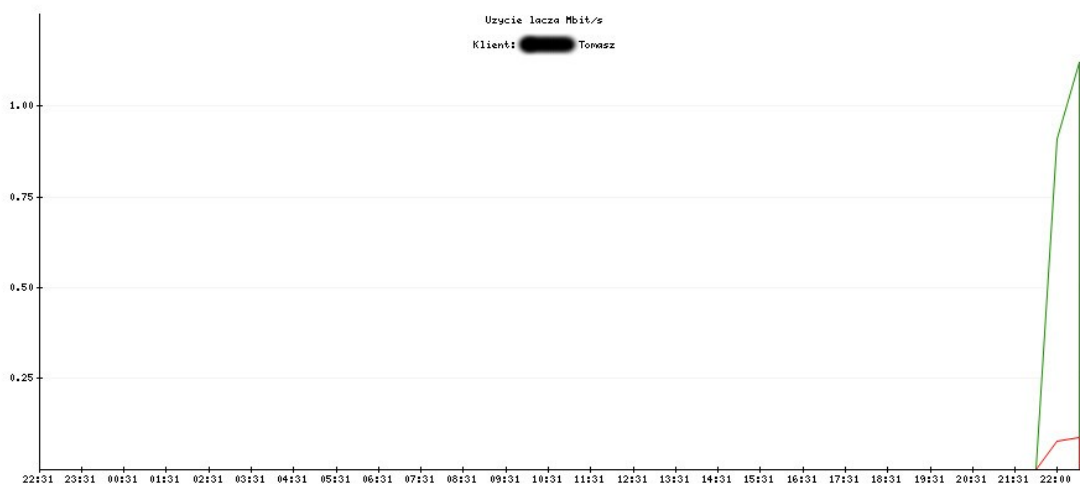
Dodaj opcję

Rys. 21: Opcje dla instancji statystyk (źródło: własne)

Po dodaniu instancji, musimy wprowadzić dla niej opcje. Dla modułu **parser.so**, konieczne jest dodanie 3 opcji: **command**, **file**, **script** (rys. 21). Jeśli którejkolwiek z nich zabraknie, instancja nie będzie mogła być wywołana, a komunikat

o błędzie pojawi się w pliku **syslog**. Za zbieranie i zapisywanie danych odpowiedzialny jest skrypt, dlatego podajemy pełną ścieżkę do niego. Opcje **file** (plik w którym instancja będzie coś zmieniać) i **script** (skrypt w języku **T-Script**) nie są potrzebne, ponieważ nie odwołujemy się do żadnego pliku, ani nic w nim nie zapisujemy. Pole **file** wypełniamy ścieżką **/dev/null**, oznaczającą pustkę, a do pola **script** wpisujemy cokolwiek (nie ma znaczenia co, ponieważ i tak wpadnie w pustkę).

Po prawidłowej konfiguracji instancji statystyk, skrypt będzie odczytywał dane z **iptables** i zapisywał te dane do bazy danych MySQL co 30 minut. Skrypt pomija wartości zerowe, czyli jeśli komputer nie był włączony przez kilka dni, baza danych nie będzie wypełniana niepotrzebnymi danymi. Jeśli klient posiada przypisaną klasę adresową, dane z tej klasy zostają **zsumowane** z danymi adresu pojedynczego przypisanego do klienta.



Rys. 22: Statystyki komputera klienta (źródło: własne)

Aby zobaczyć statystyki komputera, wchodzimy na konto klienta, po czym na komputer z przypisanym adresem IPv6. Miniaturka statystyk widoczna jest w lewym, górnym rogu informacji o komputerze. Po kliknięciu na niej, w nowej zakładce pokażą się statystyki w formie graficznej, jak na rys. 22. **Oś X** zawiera godziny, **oś Y** podaje prędkości liczone w megabitach. Prędkości podane na **osi Y**, będą dostosowane do maksymalnej wartości ściąganych/wysłanych danych. Wykres ściąganych danych oznaczony jest zielonym kolorem, a wykres wysłanych danych oznaczony jest kolorem

czerwonym.

6.4 Przeładowanie konfiguracji

Przeładowanie konfiguracji jest integralną częścią systemu LMS. Służy do zmiany konfiguracji na wybranym routerze i restart usług, których zmiana konfiguracji dotyczy. Zawartość plików konfiguracyjnych, ustala się za pomocą instancji, dzięki modułowi **parser.so**.

Nazwa:	Host:	Priorytet:	Modul: Crontab:	Opis:	Dodaj instancję
accessChorzow	irys	1	/usr/share/lms/lib/hostfile.so		
pingerChorzow	irys	1	/usr/share/lms/daemon/modules/pinger/pinger.so */10 ****		
irys-smokeping	irys	2	/usr/share/lms/lib/parser.so	Aktualizacja konfiguracji smokepinga	
routeChorzowIPv6	irys	2	/usr/share/lms/lib/parser.so		
shapingChorzow	irys	2	/usr/share/lms/lib/tc.so	Shaping Chorzow	
accessChorzowIPv6	irys	3	/usr/share/lms/lib/parser.so		
dhcpChorzowPrv	irys	3	/usr/share/lms/lib/dhcp.so		
dhcpChorzowRipe	irys	4	/usr/share/lms/lib/dhcp.so		
DNSChorzow.zone	irys	4	/usr/share/lms/lib/parser.so		
RevDNSChorzow.zone	irys	4	/usr/share/lms/lib/parser.so		
DNSChorzow-prv.zone	irys	5	/usr/share/lms/lib/parser.so		
RevDNSChorzow-prv.zone	irys	5	/usr/share/lms/lib/parser.so		
statsChorzow	irys	8	/usr/share/lms/lib/parser.so */30 ****		
redirectChorzow	irys	13	/usr/share/lms/lib/parser.so */15 ****	Ustawianie przekierowania hostow na squid aby uzyskac warning */30 ****	

Rys. 23: Lista instancji (źródło: własne)

Podczas przeładowania, pod uwagę brane są tylko i wyłącznie elementy, wyróżnione czarnym kolorem (rys. 23). Instancję dodajemy poprzez wejście w odnośnik **Dodaj instancję**, podając wszystkie wymagane dane, przykład na rys. 20.

Konfiguracja instancji: accessChorzowIPv6/irys

Nazwa:	Wartość:
command	/etc/firewall/firewallv6 restart
file	/etc/firewall/ipv6_forward
script	{result = SELECT nodes6.mac, nodes6.ipaddr, nodes6.ipaddr_class, nodes6.class_e {/for}

Rys. 24: Opcje dla modułu parser.so (źródło: własne)

Aby instancja mogła być uwzględniona podczas przeładowania, konieczne jest dodanie wszystkich opcji, wymaganych przez moduł **parser.so**:

- **command** – wpisanie polecenia, wywołanego po zmianach w pliku
- **file** – pełna ścieżka do pliku, który będzie zmieniony
- **script** – zawartość pliku, możliwe jest zastosowanie języka T-Script, umożliwia to pobieranie danych z bazy MySQL i tworzenie warunków

Przeładowanie konfiguracji

Wybierz hosty do przeładowania konfiguracji:

Nazwa:	Opis:	Ostatnie przeładowanie:
<input type="checkbox"/> bielsko	Serwer monitorujący w Bielsku Białej	2010/01/26 14:15 <input type="checkbox"/>
<input checked="" type="checkbox"/> irys	Serwer monitorujący w Chorzowie	2011/02/26 16:47 <input checked="" type="checkbox"/>
<input type="checkbox"/> katowice	Serwer monitorujący w Katowicach	2011/02/26 16:48 <input type="checkbox"/>
<input type="checkbox"/> lms	Serwer główny LMS, fakturowanie	2009/02/07 08:48 <input type="checkbox"/>
<input type="checkbox"/> myslowice	Serwer monitorujący w Mysłowicach	2011/02/26 16:48 <input type="checkbox"/>

Zaznacz wszystko Zapisz

Rys. 25: Przeładowanie konfiguracji (źródło: własne)

Po wprowadzeniu wszystkich instancji, konieczne jest przeładowanie konfiguracji. Spowoduje to zmiany w plikach konfiguracyjnych na routerze, lub zmiany w bazie danych (w przypadku statystyk). W tym celu wchodzimy na podstronę **Przeładowanie** i wybieramy router, zaznaczając go, naciskając na **Zapisz** (rys. 25). Można zaznaczyć więcej routerów, powodując przeładowanie na wszystkich, w tym

samym czasie. Zmiany na zaznaczonych routerach, dokonują się, gdy na routerze z zainstalowanym systemem LMS, sekundy będą miały wartość 0, czyli równo ze zmianą minuty na zegarze.

7 Dostęp do sieci

7.1 Skrypt zabezpieczający sieć

Skrypt zabezpieczający został stworzony na potrzeby uruchomienia reguł dostępowych dla firewala programowego o nazwie **ip6tables**, w celu zwiększenia poziomu bezpieczeństwa. W tym podrozdziale zostanie omówiona konstrukcja skryptu, oraz znaczenie zmiennych.

Zmienne:

- **FW6**=`whereis ip6tables | awk '{print \$2}'` – do zmiennej zapisana jest pełna ścieżka programu ip6tables, za pomocą komendy whereis, wyszukiwany jest program, podając nazwę programu, pełną ścieżkę do niego i pliki pomocy (manuale), polecenie **awk** za pomocą '{print \$2}' pobiera 2 kolumnę, oznaczającą ścieżkę do programu
- **net_int6** – nazwa interfejsu wyjściowego dla IPv6
- **MY6** – adres IPv6 przypisany do interfejsu głównego
- **FULL_ACCESS_INT6** – nazwy interfejsów, oddzielonych spacjami, nie poddany żadnym regułom firewala
- **FULL_ACCESS_ADDR6** – adresy IP, oddzielone spacjami, nie poddany żadnym regułom firewala
- **ICMP_IP6** – adresy IP, oddzielone spacjami, które mogą pingować router
- **ACCEPT_ALL6** – dostęp z każdego adresu IP do określonych usług, schemat wpisywania: **port:protokół port:protokół ...**
- **ACCEPT_IP6** – dostęp z określonych adresów IP, oddzielonych spacjami, do określonych usług, wyszczególnionych w zmiennej **ACCEPT_IP_PORTS6**
- **ACCEPT_IP_PORTS6** – dostęp z określonych adresów IP, wyszczególnionych w zmiennej **ACCEPT_IP6**, do określonych usług, schemat wpisywania: **port:protokół port:protokół ...**

Podział na funkcje:

- **aktyw_polit_domyslna** – zawiera politykę domyślną, podczas włączania firewala: włączenie przekazywania pakietów IPv6, odrzucanie połączeń przychodzących i przechodzących przez router, akceptacja ruchu wychodzącego z routera
- **zerowanie** – ustalenie polityki domyślnej podczas wyłączenia firewala, domyślna polityka akceptuje wszystkie połączenia i usuwa wszystkie reguły
- **test_firewall** – służy do sprawdzenia, czy przypadkiem, jedną z reguł nie zablokowaliśmy sobie dostępu do routera, test trwa 30 sekund, przez ten czas sprawdzamy czy dostęp do routera jest prawidłowy, po 30 sekundach, wywołana zostaje funkcja **zerowanie**
- **reguly** – zawiera wszystkie reguły, które zostaną wprowadzone podczas startu firewala
- **info** – funkcja wywołuje **ip6tables** z parametrami **-L -v -n**, oznaczającymi wyświetlenie wszystkich reguł w trybie szczegółowym, bez tłumaczenia adresów IP na adresy DNSowe

Skrypt firewala trzeba wykonać z określonymi parametrami:

- **test** – zostanie wykonany test poprawności reguł, poprzez wywołanie kolejno funkcji: **zerowanie**, **aktyw_polit_domyslna**, **reguly**, **test_firewall**
- **start** – uruchomienie firewala, poprzez wywołanie kolejno funkcji: **aktyw_polit_domyslna**, **reguly**
- **restart** – restart firewala, poprzez wywołanie kolejno funkcji: **zerowanie**, **aktyw_polit_domyslna**, **reguly**
- **stop** – zatrzymanie firewala, poprzez wywołanie kolejno funkcji: **zerowanie**
- **info** – wyświetlenie wszystkich reguł i polityki domyślnej dla łańcuchów INPUT, OUTPUT, FORWARD, poprzez wywołanie funkcji **info**

Przykładowe reguły, wprowadzane do **ip6tables**:

```
# Umożliwienie powrotu pakietów
```

```
$FW6 -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
$FW6 -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Zablokowanie możliwości skanowania portów

```
echo "Blokowanie nieprawidłowych pakietów dla IPv6"
```

```
$FW6 -A INPUT -p tcp --tcp-flags ALL ALL -j DROP
```

```
$FW6 -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
```

Pobranie adresów IPv6 ze zmiennej ICMP_IP6, definicja nowego łańcucha icmp, ustawienie reguł, przypisanie łańcucha icmp do łańcucha INPUT

```
echo "Dostęp ICMP"
```

```
$FW6 -N icmp
```

```
for ip in ${ICMP_IP6}
```

```
do
```

```
    $FW6 -A icmp -s $ip -p icmpv6 -j ACCEPT
```

```
done
```

```
$FW6 -A INPUT -j icmp
```

Dostęp do określonych usług na routerze, z określonych adresów IP

```
echo "Dostęp do usług z określonych adresów"
```

```
$FW6 -N serwisy
```

```
for ip in ${ACCEPT_IP6}
```

```
do
```

```
    for b in ${ACCEPT_IP_PORTS6}
```

```
    do
```

```
        PORT=`echo $b | cut -d':' -f1`
```

```
        PROTO=`echo $b | cut -d':' -f2`
```

```
        $FW6 -A serwisy -s $ip -p $PROTO --dport $PORT -j ACCEPT
```

```
    done
```

```
done
```

```
$FW6 -A INPUT -j serwisy
```

7.2 Metoda autoryzacji klientów

Klienci wpisani do systemu LMS, mający mieć dostęp do internetu, muszą być autoryzowani przez router w celu przekazywania pakietów. Jest to potrzebne z uwagi na możliwość wpisania przez innego użytkownika naszej sieci, adresu IPv6 i bramy do swojego hosta, mogącego w ten sposób korzystać z internetu, a nie mającego do tego praw formalnych, czyli nieuprawnione korzystanie z łącza.

Autoryzacja odbywa się na podstawie sprawdzania adresów MAC nadawcy pakietu. Jeśli jest on autoryzowany (wpisany do systemu LMS), pakiet zostanie

przekazany dalej, w przeciwnym wypadku, odrzucony. Pakiety przychodzące do routera od klienta są sprawdzane czy adres MAC, odpowiada adresowi IP w jednej regule. Pakiety przychodzące do routera z internetu, są sprawdzane, czy adres IP zawiera się w regule `ip6tables`. Jeśli powyższe, dwie reguły przepuszczą pakiety, klient może bez przeszkód korzystać z internetu.

Reguły dostępu muszą być wcześniej wygenerowane przez LMS. Dzieje się to po wpisaniu instancji, dodaniu do niej opcji, odpowiedniej konfiguracji opcji i przeładowanie konfiguracji.

Dodajemy instancję o nazwie **accessChorzowIPv6** do routera, gdzie konfiguracja będzie zmieniana. Dodajemy opcje opisane w rozdziale 5.3, z zawartością:

- **command:**

```
/etc/firewall/firewallv6 restart
```

Po zaktualizowaniu pliku konfiguracyjnego, konieczne jest przeładowanie reguł **ip6tables**, za pomocą skryptu

- **file:**

```
/etc/firewall/ipv6_forward
```

Wskazujemy plik, który będzie zaktualizowany

- **script:**

```
{result = SELECT nodes6.mac, nodes6.ipaddr, nodes6.ipaddr_class,
nodes6.class_eui, networks6.id, networks6.name,
networks6.clientmask FROM nodes6, networks6 WHERE networks6.id =
nodes6.netid AND (networks6.name = "IRYS-EUI" OR networks6.name
= "IRYS-EUI-CLASS")}{for (i=0; i<number(result); i++) }
{result[i].mac}{"\t"}{result[i].ipaddr}{result_eui_class =
SELECT clientmask FROM networks6 WHERE class_eui =
{result[i].id}}{if (result[i].ipaddr_class)}{"\t"}
{result[i].ipaddr_class}{"/"}{if (result[i].class_eui == 1)}
{result_eui_class[0].clientmask}{else}{result[i].clientmask}
{/if}{/if}
{/for}
```

Zawartość skryptu, generującego plik konfiguracyjny. Należy wprowadzić zmiany w powyższej konfiguracji, a mianowicie zmienić nazwy sieci w zapytaniu SQL, do której komputery są przypisane. W powyższym przykładzie konfiguracji routera w Chorzowie, nazwy sieci to: **IRYS-EUI**, **IRYS-EUI-CLASS**.

Tworzymy plik konfiguracyjny na routerach:

```
touch /etc/firewall/ipv6_forward
```

Nadajemy prawa odczytu i zapisu tylko dla roota. Podczas przeładowania konfiguracji, do pliku zapisywane są linie z danymi poszczególnych komputerów. Każdy wiersz zawiera **adres MAC** (1 kolumna), **adres IPv6** (2 kolumna) i **klasy adresowej** wraz z **maską** tej klasy (3 kolumna). Kolumny oddzielone są tabulatorem.

Do skryptu firewala, trzeba dopisać polecenia pozwalające na prawidłową obróbkę pliku konfiguracyjnego i zastosowanie reguł w **ip6tables**:

```
#
# Forward IPv6
#
echo "Forward IPv6"
$FW6 -N forward_class
$FW6 -A FORWARD -j forward_class

while read plik; do
    mac=`echo $plik | awk '{print $1}'`
    ip=`echo $plik | awk '{print $2}'`
    class=`echo $plik | awk '{print $3}'`
    if [ "$mac" ]; then
        $FW6 -A forward_class -s $ip -m mac --mac-source $mac -j
ACCEPT
        $FW6 -A forward_class -d $ip -j ACCEPT
        if [ "$class" ]; then
            $FW6 -A forward_class -s $class -m mac --mac-source
$mac -j ACCEPT
            $FW6 -A forward_class -d $class -j ACCEPT
        fi
    fi
done < /etc/firewall/ipv6_forward
```

Skrypt odczyta wiersze, w których kolumna **MAC** nie jest pusta. Docelowo oznacza to, iż nie odczyta pustych wierszy. Jeśli do adresu IP jest przypisana klasa adresowa (ostatnia kolumna), zostaną utworzone dodatkowe, 2 reguły z danymi tej klasy. Po wprowadzeniu wszystkich zmian na routerach, trzeba przeładować konfigurację w systemie LMS. Internet dla adresów pojedynczych będzie działał prawidłowo, dla klas adresowych, konieczne jest wprowadzenie zmian na routerach opisanych w kolejnym podrozdziale.

7.3 Klasy adresowe dla klientów – routing

Klientom, którym została przypisana klasa adresowa, trzeba zapewnić odpowiedni routing do/z tej klasy. Jeśli tego nie zrobimy, wykorzystanie klasy adresowej staje się niemożliwe. Klasy adresowe trasowane są na adres IPv6 przypisany do klienta, dlatego adres IP musi być wpisany do interfejsu głównego routera klienta. Trasy dopisywane są do tablicy routingu z metryką 1023. Umożliwia to szybką lokalizację tras, wprowadzonych przez system LMS, podczas przeładowania konfiguracji. Szybka lokalizacja tych tras, jest konieczna z punktu widzenia przeładowania konfiguracji. Skrypt przeładowujący musi zlokalizować trasy, usunąć je, a następnie wprowadzić nowe do tablicy routingu.

Zanim to nastąpi, potrzebne jest odpowiednie skonfigurowanie systemu LMS, poprzez dodanie instancji obsługującej proces pobierania klas adresowych, które obsługuje dany router, usunięcie aktualnej konfiguracji z pliku i zapis nowych informacji o trasach do pliku konfiguracyjnego. Dodajemy instancję o nazwie **routeChorzowIPv6**. Musimy ustawić priorytet niższy od instancji **accessChorzowIPv6**, ponieważ ta instancja przeładowuje skrypt firewala, dlatego musi być wykonana po wprowadzeniu zmian przez instancję konfigurującą trasy, a wprowadzone zmiany w trasach są aktualizowane poprzez skrypt firewala. Dodajemy opcje opisane w rozdziale 5.3, z zawartością:

- **command**: pozostawiamy puste

- **file**:

```
/etc/firewall/ipv6_routes
```

ścieżka do pliku konfiguracyjnego z trasami

- **script**:

```
{result = SELECT nodes6.ipaddr, nodes6.ipaddr_class,  
nodes6.class_eui, networks6.id, networks6.name,  
networks6.clientmask, networks6.interface FROM nodes6, networks6  
WHERE networks6.id = nodes6.netid AND nodes6.ipaddr_class != ""  
AND (networks6.name = "IRYS-EUI" OR networks6.name = "IRYS-EUI-  
CLASS")}{for (i=0; i<number(result); i++){result[i].interface}  
{"\t"}{result[i].ipaddr}{result_eui_class = SELECT clientmask  
FROM networks6 WHERE class_eui = {result[i].id}}{if  
(result[i].ipaddr_class)}{"\t"}{result[i].ipaddr_class}{"/"}{if  
(result[i].class_eui == 1){result_eui_class[0].clientmask  
{else}{result[i].clientmask}{/if}{/if}
```

```
{/for}
```

Zawartość skryptu, generującego plik konfiguracyjny. Należy wprowadzić zmiany w powyższej konfiguracji, a mianowicie zmienić nazwy sieci w zapytaniu SQL, do której komputery są przypisane. W powyższym przykładzie konfiguracji routera w Chorzowie, nazwy sieci to: **IRYS-EUI**, **IRYS-EUI-CLASS**. Nazwy te powinny być identyczne, jak w konfiguracji instancji **accessChorzowIPv6**.

Tworzymy plik konfiguracyjny na routerach:

```
touch /etc/firewall/ipv6_forward
```

Nadajemy prawa odczytu i zapisu tylko dla roota. Podczas przeładowania konfiguracji, do pliku zapisywane są linie z danymi poszczególnych tras. Każda linia zawiera 3 kolumny. **Nazwa interfejsu** (1 kolumna), **adres IPv6** przypisany do klienta (2 kolumna), **klasa adresowa**, wraz z maską sieci (3 kolumna). Kolumny oddzielone są tabulatorem.

Do skryptu firewala, trzeba dopisać polecenia pozwalające na prawidłową obróbkę pliku konfiguracyjnego i aktualizację tras w tablicy routingu:

```
#
# Czyszczenie tablicy routingu z wpisow klientow
#
ip -6 route |grep "metric 1023" | while read f; do
    klasa=`echo $f | awk '{print $1}'`
    adres=`echo $f | awk '{print $3}'`
    dev=`echo $f | awk '{print $5}'`
    ip -6 route del $klasa via $adres dev $dev
done

#
# Generowanie tablicy routingu dla klientow
#
while read plik; do
    dev=`echo $plik | awk '{print $1}'`
    ip=`echo $plik | awk '{print $2}'`
    class=`echo $plik | awk '{print $3}'`
    ip -6 route add $class via $ip dev $dev metric 1023
done < /etc/firewall/ipv6_routes
```

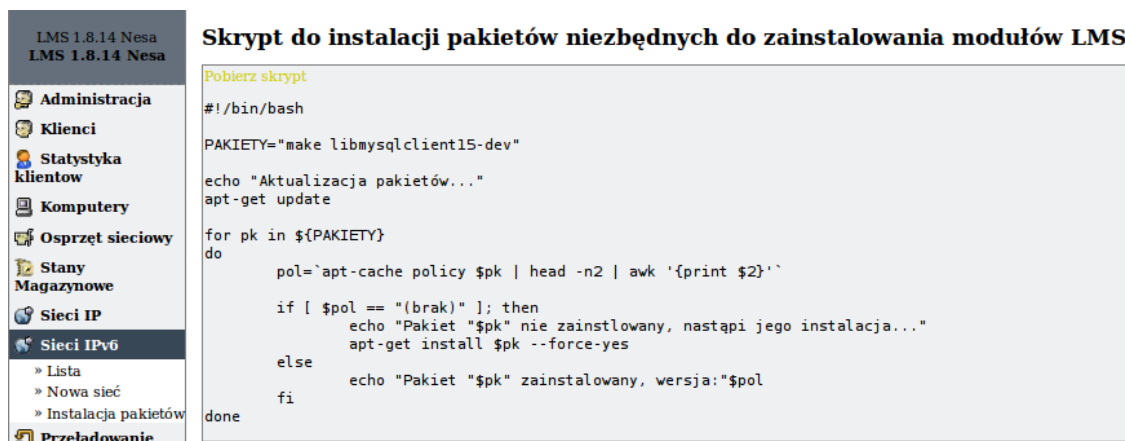
W pierwszym etapie, czyszczona jest tablica routingu, ze wszystkich wpisów klas adresowych, na podstawie metryki o wartości **1023**. W kolejnym etapie, do zmiennych dopisywane są dane z kolumn. Dane pobrane zostają z pliku

konfiguracyjnego: `/etc/firewall/ipv6_routes`. Na podstawie danych zapisanych w zmiennych, dodawana jest trasa do tablicy routingu z metryką o wartości **1023**. Metryka o tej wartości, służy jako wyróżnik spośród pozostałych tras, konieczny do aktualizacji tras.

8 Skrypty automatyzujące proces konfiguracji wstępnej

Skrypty powstały w celu ułatwienia konfiguracji początkowej routerów i komputerów klienckich. Dla routerów, instalują one niezbędne pakiety, wymagane podczas kompilacji modułów systemu LMS, konfigurują adresację IPv6 w pliku `/etc/network/interfaces` i zmieniają zawartość pliku konfiguracyjnego `radvd.conf`.

Aby zainstalować niezbędne pakiety, potrzebne do kompilacji modułów systemu LMS na nowo konfigurowanym routerze, należy wybrać w menu **Sieci IPv6**, podmenu **Instalacja pakietów** (rys. 26).



The screenshot shows the LMS 1.8.14 Nesa web interface. On the left is a navigation menu with options like 'Administracja', 'Klienci', 'Statystyka klientow', 'Komputery', 'Osprzet sieciowy', 'Stany Magazynowe', 'Sieci IP', and 'Sieci IPv6'. Under 'Sieci IPv6', there are sub-options: 'Lista', 'Nowa siec', and 'Instalacja pakietow'. The 'Instalacja pakietow' option is highlighted. To the right, a terminal window displays a shell script titled 'Skrypt do instalacji pakietow niezbednych do zainstalowania moduLOW LMS:'. The script starts with a shebang, sets a variable 'PAKIETY' to 'make libmysqlclient15-dev', updates the package list, and then iterates through the packages to check if they are installed. If not, it installs them with the --force-yes flag. The script ends with 'done'.

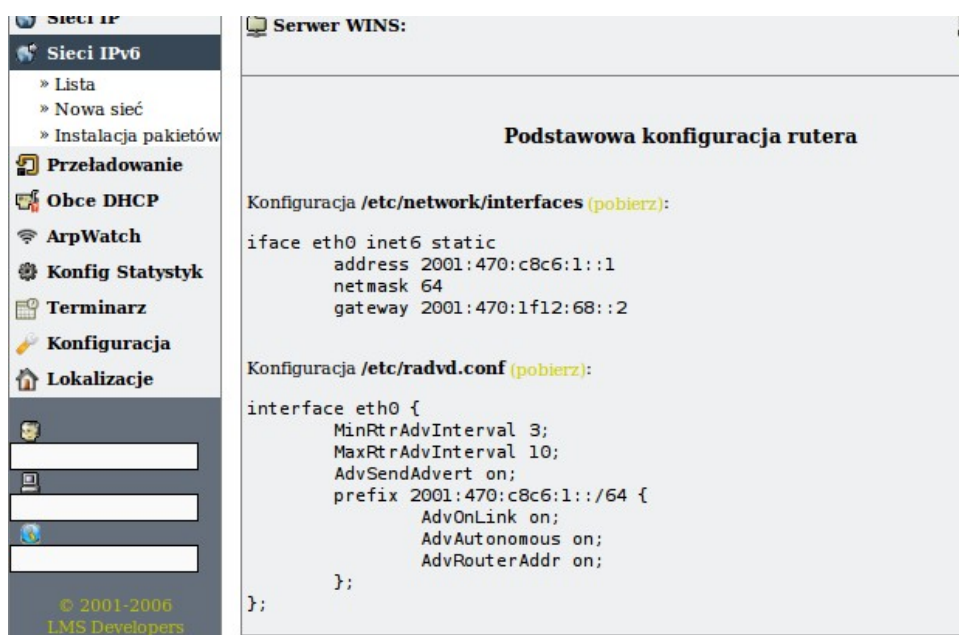
```
#!/bin/bash
PAKIETY="make libmysqlclient15-dev"
echo "Aktualizacja pakietow..."
apt-get update
for pk in ${PAKIETY}
do
    pol=`apt-cache policy $pk | head -n2 | awk '{print $2}`
    if [ $pol == "(brak)" ]; then
        echo "Pakiet "$pk" nie zainstlowany, nastapi jego instalacja..."
        apt-get install $pk --force-yes
    else
        echo "Pakiet "$pk" zainstalowany, wersja:"$pol
    fi
done
```

Rys. 26: Skrypt instalujący pakiety (źródło: własne)

Instalowane są 2 pakiety (jeśli już nie są zainstalowane w systemie), administrator ma możliwość zdefiniowania w zmiennej **PAKIETY**, innych programów do zainstalowania, po uprzednim ściągnięciu skryptu do routera. Po kliknięciu w odnośnik **Pobierz skrypt**, skrypt zostanie ściągnięty i zapisany na routerze. Plik źródłowy jest umieszczony poza strefą autoryzacji systemu LMS, dzięki czemu pobranie jest możliwe za pomocą programu `wget`.

Po instalacji pakietów, konieczna jest konfiguracja interfejsów i pliku `radvd.conf` (jeśli router będzie obsługiwał bezstanową autokonfigurację). Aby było to możliwe, konieczna jest wcześniejsza konfiguracji sieci IPv6. Zawartość konfiguracji

interfejsu i plik konfiguracyjny `radvd.conf`, widoczny jest poniżej szczegółowych informacji o sieci (rys. 27).



Rys. 27: Konfiguracja interfejsów i pliku `radvd.conf` (źródło: własne)

Jeśli sieć jest skonfigurowana do statycznej konfiguracji adresów, konfiguracja dla programu **Radvd** nie jest widoczna. Poprzez odnośniki pobierz, pobierany jest skrypt dopisujący konfigurację interfejsu na końcu pliku:

```
#!/bin/bash
echo "auto eth0" >> /etc/network/interfaces
echo "iface eth0 inet6 static" >> /etc/network/interfaces
echo " address 2001:470:c8c6:1::1" >> /etc/network/interfaces
echo " netmask 64" >> /etc/network/interfaces
echo " gateway 2001:470:1f12:68::2" >> /etc/network/interfaces
```

Skrypt konfiguruje plik `radvd.conf`, działa na podobnej zasadzie, z taką różnicą, że zostaje całkowicie wyczyszczony.

W celu ułatwienia klientom konfigurację adresów IPv6 na komputerach z systemami Windows XP, Vista, 7, możliwe jest wysłanie im **pliku bat**. Po jego uruchomieniu adres z maską zostanie przypisany do interfejsu, wraz z trasą domyślną. Skrypt został napisany z myślą głównie o systemie Windows XP, który nie ma

możliwości konfiguracji adresów, po wejściu do konfiguracji protokołu IPv6.

Pliki generowane są automatycznie, pobierając ustawienia danego komputera z bazy danych. Klient ze swojej strony musi sprawdzić, czy nazwa interfejsu, z którego korzysta z sieci lokalnej, to „Połączenie lokalne”, jeśli nie, musi zmienić nazwę w **pliku bat**.

Wszystkie odnośniki do pobrania plików konfiguracyjnych, prowadzą do skryptu PHP, umieszczonego poza strefą autoryzacji systemu LMS. Jego zawartość:

```
<?
if ($_GET['zaw'] == "pakiety") {
    $name = 'pakiety';
    $zaw = "#!/bin/bash

PAKIETY=\"make libmysqlclient15-dev\"
echo \"Aktualizacja pakietów...\"
apt-get update
for pk in ${PAKIETY}
do
    pol=`apt-cache policy $pk | head -n2 | awk '{print $2}'`

    if [ $pol == \"(brak)\" ]; then
        echo \"Pakiet \"$pk\" nie zainstlowany, nastapi jego
instalacja...\"
        apt-get install $pk --force-yes
    else
        echo \"Pakiet \"$pk\" zainstalowany, wersja:\"$pol
    fi
done";
} elseif ($_GET['zaw'] == 'int') {
    $name = "int";
    $zaw = "#!/bin/bash

echo \"auto $_GET[int]\" >> /etc/network/interfaces
echo \"iface $_GET[int] inet6 static\" >> /etc/network/interfaces
echo \"    address $_GET[addr]\" >> /etc/network/interfaces
echo \"    netmask $_GET[mask]\" >> /etc/network/interfaces
echo \"    gateway $_GET[gw]\" >> /etc/network/interfaces
";
} elseif ($_GET['zaw'] == 'ra') {
    $name = "radvd";
    $zaw = "#!/bin/bash
```

```

echo \"interface $_GET[int] {\\" > /etc/radvd.conf
echo \"MinRtrAdvInterval 3;\\" >> /etc/radvd.conf
echo \"MaxRtrAdvInterval 10;\\" >> /etc/radvd.conf
echo \"AdvSendAdvert on;\\" >> /etc/radvd.conf
echo \"prefix $_GET[addr]/64 {\\" >> /etc/radvd.conf
echo \\"
        AdvOnLink on;\\" >> /etc/radvd.conf
echo \\"
        AdvAutonomous on;\\" >> /etc/radvd.conf
echo \\"
        AdvRouterAddr on;\\" >> /etc/radvd.conf
echo \\"
    };\\" >> /etc/radvd.conf
echo \"};\\" >> /etc/radvd.conf
";

} else {
    $name = "ipv6.bat";
    $dbhost = "localhost";
    $dbuname = "lms";
    $dbpass = "f4h6*!";
    $dbname = "lms";

    @$db = mysql_pconnect($dbhost,$dbuname,$dbpass);
    if(!$db) {
        exit;
    }
    mysql_select_db($dbname);
    $queryNames = "SET NAMES 'latin2'";
    mysql_query($queryNames);
    $result_nodes = mysql_query("SELECT ipaddr FROM nodes6 WHERE
id='$_GET[id]'");
    $row_nodes = mysql_fetch_array($result_nodes);
    $nodeid = $row_nodes['nodeid'];

    $zaw = "netsh interface ipv6 install
netsh interface ipv6 add address \"Połączenie lokalne\"
$row_nodes[ipaddr]
netsh interface ipv6 add route 2001::/3 \"Połączenie lokalne\"
";
}
header("Content-type: application/octet-stream");
header("Content-Disposition: attachment; filename=$name");
echo $zaw;

?>

```

9 Opis wybranych fragmentów kodu źródłowego

9.1 Klasa ipv6

Konieczne było napisanie klasy, która potrafi poprawnie zinterpretować adres IPv6, wyłuskać z niego adres sieci, pokazać go w formacie skróconym lub pełnym i inne potrzebne metody. Bez tej klasy, wszystkie operacje na adresie IPv6 byłyby niemożliwe. Po kolei zostanie omówiona każda z metod klasy:

- **check_ip(\$str)** – metoda pobiera zmienną z adresem IPv6, zmienia na format pełny i sprawdza czy liczba bajtów się zgadza (39), i liczba dwukropków wynosi 7
- **full_addr(\$str)** – zamienia przekazany adres IP na format preferowany
- **net_addr(\$str, \$maska)** – na podstawie przekazanego adresu IP i maski, wylicza adres sieci
- **fin_addr(\$str, \$maska)** – na podstawie przekazanego adresu IP i maski, wylicza ostatni adres w sieci
- **short_addr(\$str)** – zamienia przekazany adres IP na format skompresowany
- **cli_rtr_mask(\$mask, \$cli_mask)** – na podstawie maski sieci i maski dla klientów wylicza ilość możliwych do zaadresowania klas dla klientów
- **next_addr(\$address, \$mask)** – na podstawie adresu IP i maski, wylicza kolejny adres sieci, metoda potrzebna w celu wyliczenia kolejnej wolnej klasy adresowej dla klientów
- **next_ip_database(\$netid)** – z przekazanej sieci wylicza kolejny adres IP tak długo, aż znajdzie nie przypisany do klienta
- **next_class_database(\$netid)** – z przekazanej sieci wylicza kolejną klasę adresową tak długo, aż znajdzie nie przypisaną do klienta
- **GetCustomerNodes(\$id)** – wysyła listę komputerów, wraz z wszystkimi polami w bazie danych, na podstawie numeru ID klienta
- **GetNodeOwner(\$nodeid)** – na podstawie numeru ID komputera, wysyła numer ID jego użytkownika
- **GetNode(\$id)** – na podstawie numeru ID komputera, wysyła wszystkie jego

dane

- **GetEui(\$net, \$mac)** – na podstawie adresu sieci i adresu MAC, wylicza publiczny adres IP w formacie EUI-64
- **NodeExists(\$id)** – sprawdza czy komputer występuje w bazie danych
- **GetNodeIDByMAC(\$mac)** – na podstawie adresu MAC, wysyła numer ID komputera
- **bin2hex(\$input)** – zamienia liczbę binarną na heksadecymalną
- **hex2bin(\$input)** – zamienia liczbę heksadecymalną na binarną
- **NetworkExists(\$id)** – sprawdza czy sieć występuje w bazie danych

9.2 Dodane moduły

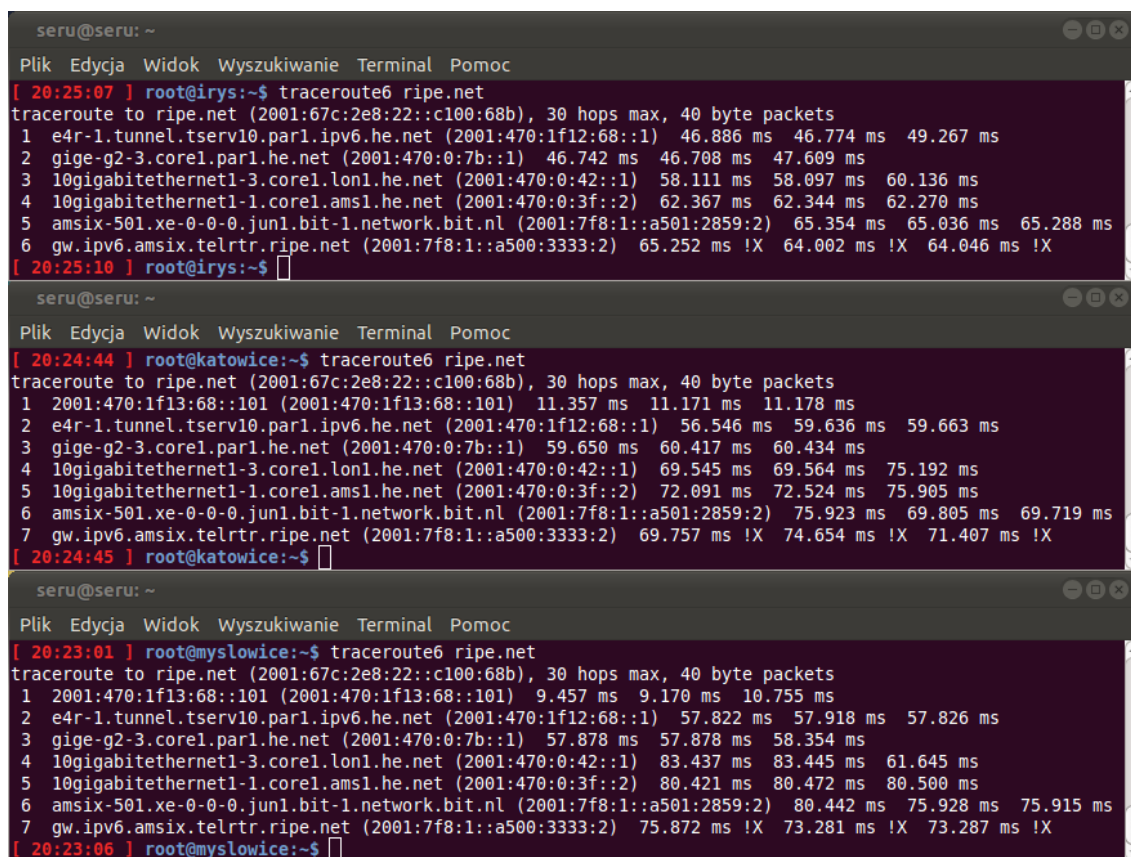
Dodanie obsługi protokołu IPv6 do systemu LMS, wymusiło zaprogramowanie nowych modułów. Bez nich nie byłoby możliwości dopisania sieci, komputerów do bazy danych. Moduły posiadają algorytmy sprawdzające poprawność wpisanych danych, a w przypadku wykrycia błędu, zostanie on napisany i wyróżniony czerwonym kolorem. Każdy moduł podzielony jest na 2 części: kodowa, prezentacji. Część kodowa jest napisana w języku PHP, która zawiera algorytmy, pętle, pobieranie danych z bazy danych, zapis danych do bazy danych i inne funkcje, potrzebne dla danego modułu. Część prezentacji zawiera kod HTML, do którego doklejane są znaczniki SMARTY, głównie zmienne, tablice, przekazane z części kodowej. Lista zaprogramowanych modułów do systemu LMS:

- **netadd6** – umożliwia wyświetlenie formularza z polami niezbędnymi do dodania sieci, sprawdzenia, po czym zapisania ich do bazy danych
- **netinfo6** – odpowiedzialny za wyświetlenie informacji o sieci, oraz wszystkie obliczenia pozwalające na podział pomiędzy adresami indywidualnymi i klasami adresowymi
- **netedit6** – umożliwia wyświetlenie formularza z polami niezbędnymi do edycji sieci, sprawdzenia, po czym zaktualizowania ich w bazie danych
- **netdel6** – odpowiedzialny za usunięcie sieci, wraz ze wszystkimi komputerami do niej należącymi
- **netlist6** – wyświetla wszystkie sieci z podstawowymi informacjami na ich temat

- **nodeadd6** – umożliwia wyświetlenie formularza z polami niezbędnymi do dodania komputera (głównie **wybór sieci i adres MAC**), sprawdzenia, po czym zapisania do bazy danych
- **nodeinfo6** – odpowiedzialny za wyświetlenie informacji o komputerze pobranych z bazy danych
- **nodeedit6** – umożliwia wyświetlenie formularza z polami niezbędnymi do edycji komputera, sprawdzenia, po czym zaktualizowania go w bazie danych
- **node6del** – odpowiedzialny za usunięcie komputera

10 Wnioski

Adresacja IPv6, wdrożona do sieci działa prawidłowo. Routery komunikują się wzajemnie dzięki szyfrowanym tunelom między nimi. Rys. 28 przedstawia prawidłowe działanie internetu na routerach. Router w **Chorzowie** (nazwa routera: **irys**) ma bezpośredni dostęp do sieci IPv6, a pozostałe miasta łączą się do niego za pomocą tunelu co widać na rys. 28: **Katowice**, **Mysłowice** jako pierwszy przeskok mają adres IPv6 routera w **Chorzowie**.



```
seru@seru: ~
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
[ 20:25:07 ] root@irys:~$ traceroute6 ripe.net
traceroute to ripe.net (2001:67c:2e8:22::c100:68b), 30 hops max, 40 byte packets
 1 e4r-1.tunnel.tserv10.par1.ipv6.he.net (2001:470:1f12:68::1) 46.886 ms 46.774 ms 49.267 ms
 2 gige-g2-3.core1.par1.he.net (2001:470:0:7b::1) 46.742 ms 46.708 ms 47.609 ms
 3 10gigabitethernet1-3.core1.lon1.he.net (2001:470:0:42::1) 58.111 ms 58.097 ms 60.136 ms
 4 10gigabitethernet1-1.core1.ams1.he.net (2001:470:0:3f::2) 62.367 ms 62.344 ms 62.270 ms
 5 amsix-501.xe-0-0-0.jun1.bit-1.network.bit.nl (2001:7f8:1::a501:2859:2) 65.354 ms 65.036 ms 65.288 ms
 6 gw.ipv6.amsix.telrtr.ripe.net (2001:7f8:1::a500:3333:2) 65.252 ms !X 64.002 ms !X 64.046 ms !X
[ 20:25:10 ] root@irys:~$

seru@seru: ~
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
[ 20:24:44 ] root@katowice:~$ traceroute6 ripe.net
traceroute to ripe.net (2001:67c:2e8:22::c100:68b), 30 hops max, 40 byte packets
 1 2001:470:1f13:68::101 (2001:470:1f13:68::101) 11.357 ms 11.171 ms 11.178 ms
 2 e4r-1.tunnel.tserv10.par1.ipv6.he.net (2001:470:1f12:68::1) 56.546 ms 59.636 ms 59.663 ms
 3 gige-g2-3.core1.par1.he.net (2001:470:0:7b::1) 59.650 ms 60.417 ms 60.434 ms
 4 10gigabitethernet1-3.core1.lon1.he.net (2001:470:0:42::1) 69.545 ms 69.564 ms 75.192 ms
 5 10gigabitethernet1-1.core1.ams1.he.net (2001:470:0:3f::2) 72.091 ms 72.524 ms 75.905 ms
 6 amsix-501.xe-0-0-0.jun1.bit-1.network.bit.nl (2001:7f8:1::a501:2859:2) 75.923 ms 69.805 ms 69.719 ms
 7 gw.ipv6.amsix.telrtr.ripe.net (2001:7f8:1::a500:3333:2) 69.757 ms !X 74.654 ms !X 71.407 ms !X
[ 20:24:45 ] root@katowice:~$

seru@seru: ~
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
[ 20:23:01 ] root@myslowice:~$ traceroute6 ripe.net
traceroute to ripe.net (2001:67c:2e8:22::c100:68b), 30 hops max, 40 byte packets
 1 2001:470:1f13:68::101 (2001:470:1f13:68::101) 9.457 ms 9.170 ms 10.755 ms
 2 e4r-1.tunnel.tserv10.par1.ipv6.he.net (2001:470:1f12:68::1) 57.822 ms 57.918 ms 57.826 ms
 3 gige-g2-3.core1.par1.he.net (2001:470:0:7b::1) 57.878 ms 57.878 ms 58.354 ms
 4 10gigabitethernet1-3.core1.lon1.he.net (2001:470:0:42::1) 83.437 ms 83.445 ms 61.645 ms
 5 10gigabitethernet1-1.core1.ams1.he.net (2001:470:0:3f::2) 80.421 ms 80.472 ms 80.500 ms
 6 amsix-501.xe-0-0-0.jun1.bit-1.network.bit.nl (2001:7f8:1::a501:2859:2) 80.442 ms 75.928 ms 75.915 ms
 7 gw.ipv6.amsix.telrtr.ripe.net (2001:7f8:1::a500:3333:2) 75.872 ms !X 73.281 ms !X 73.287 ms !X
[ 20:23:06 ] root@myslowice:~$
```

Rys. 28: Prawidłowe działanie internetu na wszystkich routerach (źródło: własne)

Wszystkie elementy programistyczne funkcjonują prawidłowo. Administrator ma możliwość dodawania sieci IPv6, a błędy wykryte po wypełnieniu formularza zostaną opisane i podświetlone na czerwono. Jeśli to konieczne, można także zaktualizować sieci bądź je usunąć. Nowa funkcjonalność wyświetla szczegółowe informacje nt. sieci, podając adres początkowy i końcowy, podając podział na część

adresów indywidualnych, i klas adresowych wraz z maskami sieciowymi, i inne informacje. Dodając sieć, administrator określa w jaki sposób klienci będą konfigurować adresy IPv6, czy ma to być konfiguracja statyczna, czy automatyczna.

Nieodzownym elementem funkcjonowania systemu LMS, było napisanie modułów obsługujących adresy IPv6 przypisywane do klientów. Dzięki nim administrator może dodawać komputery z adresami IPv6, generowanymi automatycznie. Posiada też możliwość dodania klasy adresowej dla klienta w przypadku, gdy sieć jest przystosowana do bezstanowej autokonfiguracji, lub klasy są dodawane automatycznie wraz z nadaniem adresu IPv6 podczas dodawania komputera. Jeśli klasy adresowe nie są potrzebne w danej podsieci, należy podczas dodawania sieci wpisać odpowiednią maskę.

Administrator ma podgląd, ile danych ściągnął, wysłał dany komputer klienta. Statystyki takie wyświetlane są w formie graficznej, podliczając dane z **ip6tables**. Komputery klientów autoryzowane są na podstawie **adresu MAC** ich karty sieciowej. Uniemożliwia to innym hostom, podłączonym do sieci, nie mających mieć dostępu do sieci IPv6, wpisania ręcznie wolnego adresu IPv6, korzystając dzięki temu z internetu.

Na rys. 18 przedstawiono prawidłowe działanie internetu na komputerze klienta, który łączy się przez router. Do klienta przypisana jest klasa adresowa. Router klienta generuje adres IPv6 w formacie EUI-64, dlatego router posiada dostęp do internetu bez żadnej dodatkowej konfiguracji. Aby udostępnić publiczne adresy IPv6 wewnątrz sieci lokalnej klienta (za routerem), potrzebne jest dopisanie adresu do interfejsu obsługującego sieć lokalną (w tym przypadku **br0**) i dopisanie kolejnego adresu do komputera klienta, ustawiając adres IPv6 bramy, skonfigurowany na routerze.

Wszystkie cele pracy zostały osiągnięte. Klienci mają możliwość korzystania z zasobów sieci IPv6, niezależnie od sieci IPv4. Konfiguracje sieci umożliwiają wybór, czy udostępniamy klientom klasy adresowe i w jaki sposób klienci muszą skonfigurować sobie adresy IPv6 (statycznie, bądź automatycznie).

Spis rysunków

Rys. 1: Dodawanie sieci, komunikat o nieprawidłowym adresie (źródło: własne).....	27
Rys. 2: Dodawanie sieci, przykładowo wypełnione pola, dla pojedynczych adresów (źródło: własne).....	29
Rys. 3: Dodawanie sieci, przykładowa konfiguracja dla klas adresowych (źródło: własne).....	30
Rys. 4: Szczegóły sieci, pokazano podział przestrzeni pojedynczych adresów i klas adresowych (źródło: własne).....	31
Rys. 5: Dodawanie sieci, przykładowana konfiguracja dla autokonfiguracji bezstanowej (źródło: własne).....	32
Rys. 6: Dodawanie klasy dla sieci w formacie EUI-64 (źródło: własne).....	33
Rys. 7: Informacje podstawowe, nt. wprowadzonych sieci (źródło: własne).....	33
Rys. 8: Edycja sieci (źródło: własne).....	34
Rys. 9: Komunikat pojawiający się podczas usuwania sieci (źródło: własne).....	35
Rys. 10: Adresy IP przypisane do klienta (źródło: własne).....	37
Rys. 11: Formularz dodawania adresu IPv6 (źródło: własne).....	37
Rys. 12: Komunikat błędnego wprowadzenia adresu MAC (źródło: własne).....	38
Rys. 13: Szczegółowe informacje nt. wprowadzonego komputera (źródło: własne).....	39
Rys. 14: Edycja komputera (źródło: własne).....	40
Rys. 15: Klasa adresowa przypisana do adresu IP (źródło: własne).....	41
Rys. 16: Adres IP w formacie EUI-64, bez przypisanej klasy adresowej (źródło: własne).....	42
Rys. 17: Przykładowa konfiguracja rutera klienta (źródło: własne).....	43
Rys. 18: Testowa konfiguracja komputera klienta i test działania (źródło: własne).....	43
Rys. 19: Zapisywane na bieżąco informacje statystyczne pobranych/wysłanych danych (źródło: własne).....	44
Rys. 20: Instancja statystyk z wypełnionymi polami (źródło: własne).....	45
Rys. 21: Opcje dla instancji statystyk (źródło: własne).....	45
Rys. 22: Statystyki komputera klienta (źródło: własne).....	46
Rys. 23: Lista instancji (źródło: własne).....	47
Rys. 24: Opcje dla modułu parser.so (źródło: własne).....	48

Rys. 25: Przeładowanie konfiguracji (źródło: własne).....	48
Rys. 26: Skrypt instalujący pakiety (źródło: własne).....	58
Rys. 27: Konfiguracja interfejsów i pliku radvd.conf (źródło: własne).....	59
Rys. 28: Prawidłowe działanie internetu na wszystkich ruterach (źródło: własne).....	65

Bibliografia

- [1] – oficjalna strona internetowa www.ietf.org/about – informacja z dnia 18 marca 2011r.
- [2] – R. Graziani, B. Vachon: *Akademia sieci Cisco CCNA Exploration Semestr 4 Sieci WAN – zasady dostępu*. WN PWN, Warszawa 2009, s. 496.
- [3] – Regis Desmeules: *IPv6: Sieci oparte na protokole IP w wersji 6*. WN PWN, Warszawa 2006, s. 175.
- [4] – M. A. Dye, R. McDonald, A. W. Ruffi: *Akademia sieci Cisco CCNA Exploration Semestr 1 Podstawy sieci*. WN PWN, Warszawa 2008, s. 227
- [5] – Carla Schroder: *Sieci Linux. Receptury*. Helion, Gliwice 2009, s. 465
- [6] – M. A. Dye, R. McDonald, A. W. Ruffi: *Akademia sieci Cisco CCNA Exploration Semestr 1 Podstawy sieci*. WN PWN, Warszawa 2008, s. 226
- [7] – Regis Desmeules: *IPv6: Sieci oparte na protokole IP w wersji 6*. WN PWN, Warszawa 2006, s. 80.
- [8] – Regis Desmeules: *IPv6: Sieci oparte na protokole IP w wersji 6*. WN PWN, Warszawa 2006, s. 154.
- [9] – oficjalna strona internetowa www.openssl.org – informacja z dnia 18 marca 2011r.
- [10] – oficjalna strona internetowa www.dd-wrt.com
- [11] – M. A. Dye, R. McDonald, A. W. Ruffi: *Akademia sieci Cisco CCNA Exploration Semestr 1 Podstawy sieci*. WN PWN, Warszawa 2008, s. 182

[12] – RFC2460 oraz RFC4291

[13] – oficjalna strona internetowa www.lms.org.pl

[14] – K. Folga: Trudna droga do IPv6. „NetWorld”. 2010, nr 5, s. 40

[15] – K. Folga: Trudna droga do IPv6. „NetWorld”. 2010, nr 5, s. 38

[16] – K. Folga: Trudna droga do IPv6. „NetWorld”. 2010, nr 5, s. 39

[17] – RFC 4291

[18] – K. Folga: Trudna droga do IPv6. „NetWorld”. 2010, nr 5, s. 38