Uniwersytet w Białymstoku

Wydział Matematyczno-Fizyczny

Instytut Matematyki

Justyna Brzozowska

Obsługa domen na serwerze DNS u dostawcy usług internetowych

Praca dyplomowa napisana
pod kierunkiem
dr. Mariusza Żynela

Białystok 2006

Spis treści

Wstęp

DNS, czyli System Nazw Domen umożliwia funkcjonowanie Internetu. DNS-u używamy wtedy, gdy używamy internetu. Wysyłając pocztę elektroniczną lub surfując przez WWW, za każdym razem jesteśmy zależni od DNS-u. My jako istota ludzka wolimy zapamiętywać nazwy komputerów, natomiast komputery adresują się nawzajem za pomocą liczb. W internecie te liczby mają długość 32 bitów. Jest to proste do zapamiętania dla komputerów, gdyż mają one pamięć dostosowaną do przechowywania liczb, w przeciwieństwie do nas. Nie jesteśmy w stanie zapamiętać np. dziesięciu przypadkowych telefonów z książki telefonicznej, z tego też powodu potrzebujemy systemu nazw domen, czyli DNS-u.

DNS jest standardowym w Internecie mechanizmem ogłaszania i dostępności nie tylko adresów, ale wszystkich rodzajów informacji o hostach i jest wykorzystywany przez praktycznie całe oprogramowanie międzysieciowe, włączając w to pocztę elektroniczną, programy przesyłania plików jak *ftp* i przeglądarki sieciowe np. Internet Explorer czy Mozillę.

DNS pozwala na rozdzielenie zarządzania informacją hosta między wiele ośrodków i organizacji. To również udostępnianie informacji hosta wszędzie w Internecie. DNS jest swoistą „książką telefoniczną" Internetu zamieniającą słowne nazwy domen (np. math.uwb.edu.pl) na zrozumiały dla komputerów numer IP.

Bez DNS Internet by nie działał. DNS jest specyficzną bazą danych, która u dostawcy usług internetowych może być tak duża i skomplikowana, że bez dodatkowych narzędzi ciężko jest nią zarządzać. Stąd właśnie zrodził się pomysł napisania aplikacji wspomagającej administartorów sieci, co jest tematem mojej pracy. W bazie danych aplikacji przechowywane są dane o obsługiwanych domenach. Przyjazny interfejs aplikacji obsługujący przeglądarką internetową ułatwia wykonywanie codzienych obowiązków administratora DNS. Niektóre operacje ,jak generowanie plików konfiguracyjnych jest zautomatyzowane. Aplikacja z przykładowymi danymi dostępna jest pod adresem:

`http://math.uwb.edu.pl/dns/`

Rozdział 1

Domain Name Service

1.1 Adresy IP

Komputery komunikują się w sieciach TCP/IP, wykorzystując adresy IP do identyfikacji. Protokół TCP/IP to podstawa współczesnych technologii sieciowych, a przede wszystkim Internetu. Do rozróżniania komputerów TCP/IP używa 32 bitowej liczby całkowitej, nazywanej adresem IP. Pomysłowość tego systemu adresowania polega na tym, że umożliwia on efektywne wyznaczanie tras pakietów. Jest to możliwe dzięki temu, że adres IP zawiera informację o tym do jakiej sieci jest włączony dany komputer oraz jednoznaczny adres komputera w tej sieci. Adres IP jest używany przy wszystkich operacjach związanych z wymianą informacji z daną maszyną. Ogólnie przyjętym sposobem zapisu adresu IP w sposób czytelny dla użytkownika jest format bajtowo-dziesiętny - adres zapisywany jest w postaci czterech liczb dziesiętnych,które oddzielone są kropkami, przy czym każda liczba dziesiętna odpowiada 8 bitom adresu IP. Taki zapis nosi nazwę notacji dziesiętnej z kropkami (ang. dotted quad notation). Zapis taki jest z pewnością o wiele bardziej czytelny dla człowieka niż zapis bitowy. Obserwując najstarsze bity adresu możemy stwierdzić do jakiej klasy należy dany adres, w efekcie możemy stwierdzić ile bitów będzie adresowało sieć, ile zaś sam komputer. Aby określić przynależność do jednej z trzech zasadniczych klas (A, B, C) wystarczą dwa pierwsze bity. Łatwo zauważyć, że adresów klasy A wykorzystywanych przez duże sieci jest niewiele (na adres sieci przeznaczone jest 7 bitów, więc sieci takich jest $2^7 - 1 = 127$) ale w każdej z sieci tej klasy może być ponad 16 milionów komputerów (na adres maszyny przeznaczone jest 24 bity więc otrzymujemy 2^24 maszyn). Klasa B przeznaczona jest dla sieci średniej wielkości mających od 2^8 (tj. 256) do 2^16 maszyn - 14 bitów określa sieć,zaś 16 bitów komputer. W efekcie otrzymujemy 16384 sieci, które mogą mieć do 65535 komputerów każda. W klasie C sieć adresowana jest za pomocą 21 bitów - daje to 2^21 sieci (ponad 2 miliony) ale w każdej z nich może być co najwyżej $2^8 = 256$ maszyn. Adres klasy D (ang. multicast address) ma specjalne znaczenie - jest używany w sytuacji gdy ma

miejsce jednoczesna transmisja do większej liczby urządzeń.

Podział na klasy wygląda następująco:

Klasa	Najniższy adres	Najwyższy adres
A	0.1.0.0	126.0.0.0
B	128.0.0.0	191.255.0.0
C	192.0.1.0	223.255.255.0
D	224.0.0.0	239.255.255.255
E	240.0.0.0	247.255.255.255

Postać adresu IP umożliwia szybkie określenie zawartego w nim adresu sieci i adresu maszyny. Wykorzystują to **routery**, które wymagają możliwości sprawnego wyróżnienia tego adresu w celu szybkiej pracy. Adres IP każdego urządzenia, które może być połączone z intersiecią musi być unikalny w skali światowej. W celu zapewnienia jednoznaczności identyfikatorów sieci, wszystkie adresy przydzielane są przez jedną organizację. Zajmuje się tym Internet Network Information Center (INTERNIC). Przydziela ona adresy sieci, zaś adresy maszyn w ramach sieci administrator może przydzielać bez potrzeby kontaktowania się z organizacją. Organizacja ta przydziela adresy tym instytucjom, które są lub będą przyłączone do ogólnoświatowej sieci INTERNET. Każda instytucja może sama wziąć odpowiedzialność za ustalenie adresu IP, jeśli nie jest połączona ze światem zewnętrznym. Nie jest to jednak dobre rozwiązanie, gdyż w przyszłości może uniemożliwić współpracę między sieciami i sprawiać trudności przy wymianie oprogramowania z innymi ośrodkami. Adresy IP są długie i trudne do zapamiętania, dlatego do powszechnego użytku zastosowano system specyficznych nazw np. math.uwb.edu.pl. Tego typu nazwy są łatwiejsze do zapamiętania, ponieważ mają określoną strukturę, pozwalającą domyśleć się ich znaczenia. Wystarczy użyć takiej nazwy, a routery zamienią ją na równoważny adres IP, wykorzystywany do przesyłania pakietów danych. System nazw DNS (ang. Domain Name System), opisujący adresy komputerów i organizacji, do których one należą, ma układ odwrotny od wykorzystywanego przez liczbowe adresy IP.

1.2 Historia systemu nazw domen

Korzenie Internetu sięgają eksperymentalnej sieci komputerowej łączącej organizacje badawcze, zwanej ARPAnet. ARPAnet był małą, przyjacielską wspólnotą kilku setek hostów. W latach siedemdziesiątych minionego stulecia sieć ARPAnet liczyła zaledwie kilkaset hostów. Wraz z momentem kiedy opracowany na Uniwersytecie Kalifornijskim w Berkeley zestaw protokołów TCP/IP stał się standardowym protokołem w ARPAnecie oraz gdy dołączono go do systemu Unix BSD łączność z ARPAnetem stała się dostępna dla wielu organizacji. Sieć rozrosła się do tysięcy hostów. Człowiek nie jest maszyną dlatego też w przeciwieństwie do nich łatwiej mu zapamiętywać nazwy niż cyfry. Tak oto powstał plik HOSTS.TXT przechowujący odwzorowania nazw komputerów

ARPAnetu na adresy IP. Pojedyńczy plik, HOSTS.TXT, zawierał wszystkie informacje o nich: mapowanie pomiędzy nazwą i adresem dla każfdego hosta dołączonego do ARPAnetu. Plik ten redagowało Centrum Informacji Sieciowej (NIC) w Instytucie Badawczym Stanforda, a rozpowszechniany był z hosta SRI-NIC. Plik HOSTS.TXT był utrzymywany przez Centrum Informacji Sieciowej SRI i rozsyłany z pojedyńczego hosta. Jak łatwo się domyślić takie rozwiązanie spowodowało iż wraz ze wzrostem liczby hostów rosła wielkość pliku, a co ważniejsze zwiększył się ruch w sieci powodowany ciągłym uaktualnianiem tego pliku. Obciążenie hosta SRI-NIC zbliżało się do wartości krytycznej. Administratorzy ARPANetu najczęściej przesyłali zmiany do NIC pocztą elektroniczną, a okresowo łączyli się przez ftp ze SRI-NIC i pobierali aktualny plik HOSTS.TXT. Ich zmiany były kompilowane do nowego pliku HOSTS.TXT raz albo dwa razy w tygodniu. Jednak gdy ARPAnet rozrastał się, stawało się to nie wykonalne. Rozmiar pliku HOSTS.TXT rósł proporcjonalnie do wzrostu liczby hostów w ARPANecie. I kiedy ARPAnet przeszedł na protokoły TCP/IP, populacja sieci eksplodowała.

Było mnóstwo problemów z plikiem HOSTS.TXT:

- **Ruch i obciążenie**
  Wykorzystanie SRI-NIC stało się nierealne ze względu na ruch w sieci i obciążenie procesora wymagane do rozdzielania pliku.

- **Kolizje nazw**
  W pliku HOSTS.TXT nie może być dwóch hostów o tej samej nazwie. Jednak chociaż NIC mógł przypisywać adresy w sposób gwarantujący ich unikatowość, nie miał żadnego wpływu na nazwy hostów. Nic nie chroniło przed dodaniem hosta z kolidującą nazwą, łamiącego cały układ. Na przykład ktoś, dodając hosta z taką samą nazwą jak nazwa większego centrum pocztowego, mógłby zakłócić usługi pocztowe w wiekszości ARPAnetu.

- **Spójność**
  Utrzymywanie spójności pliku w powiększającej się sieci było coraz trudniejsze. W czasie, gdy nowy plik HOSTS.TXT docierał do najdalszych krańców powiększającego się ARPAnetu, hosty w innych miejscach sieci zmieniały adresy lub pojawiał się nowy host, do którego chcieli się dostać użytkownicy.

Zasadniczym problemem było to, że mechanizm pliku HOSTS.TXT nie umożliwiał powiększania skali sieci. Jak na ironie, sukces eksperymentu ARPAnet prowadził do niepowodzenia i zaniku pliku HOSTS.TXT. Zarządcy ARPAnetu zlecili badania nad opracowaniem następcy pliku HOSTS.TXT – systemu, który rozwiązywałby problemy tkwiące w zunifikowanym systemie tabeli hostów. Nowy system powinnien pozwalać na lokalne zarządzanie danymi, a jednocześnie udostępniać te dane w skali globalnej. Lokalne zarządzanie

ułatwiłoby utrzymywanie i aktualizację danych. System powinien stosować hierarchiczną przestrzeń nazw hostów, co zapewniłoby ich unikatowość. W 1984 roku Paul Mockapetris wydaje dokumenty RFC 882 i 883 opisujące domenowy system nazw (DNS).

1.3 Struktura DNS

Struktura bazy danych DNS-u jest bardzo podobna do budowy systemu plików UNIX-a. Cała baza danych(lub system plików) jest przedstawiona jako odwrócone drzewo, z węzłem głównym(korzeniem) na górze. W tekście węzeł główny jest oznaczony jako pojedyńcza kropka (”.”), a systemie plików UNIX-a jako ukośnik (”/”). Podobnie jak system plików, drzewo DNS może mieć dowolną liczbę rozgałęzień (dróg) w każdym punkcie przecięcia, nazywanym węzłem. Głębokość drzewa jest ograniczona do 127 poziomów. Każdy węzeł w drzewie ma etykietę tekstową(bez kropek) o długości do 63 znaków. Zerowa etykieta (o zerowej długości) jest zarezerwowana dla węzła głównego. Pełna nazwa domeny dowolnego węzła drzewa to sekwencja etykiet na ścieżce od tego węzła do węzła głównego. Nazwy domen zawsze czytane są od danego węzła w kierunku węzła głównego (”w górę” drzewa), z kropkami rozdzielającymi poszczególne nazwy w ścieżce.

W DNS-ie każda domena może być zarządzana przez inną organizację. Każda organizacja może podzielić swoją domenę na pewną liczbę subdomen i przekazać odpowiedzialność za nie innym organizacjom. Nazwy domen są stosowane jako indeksy bazy danych DNS. W systemie plików katalogi zawierają pliki i podkatalogi, podobnie domena może zawierać zarówno hosty, jak też subdomeny. Domena zawiera te hosty i subdomeny, których nazwy znajdują się wewnątrz niej. Każdy host w sieci ma nazwe domeny, która wskazuje na informacje o nim. Informacja może zawierać adresy IP, informację o trasowaniu poczty itd.

Hosty mogą też mieć jeden albo więcej *aliasów* (synonimów) *nazwy domeny*, które są wskaźnikami od jednej nazwy domeny (aliasu) do innej (urzędowej lub zwyczajowej). Rozproszona baza danych DNS jest indeksowana przez nazwy domen. Każda nazwa domeny to zasadniczo ścieżka w wielkim odwróconym drzewie, nazwanym *przestrzenią nazw domeny* (ang. *domain name space*). Domena jest po prostu poddrzewem przestrzeni nazw domen. Nazwa domeny jest taka sama jak nazwa węzła na szczycie domeny. Każda nazwa domeny w poddrzewie jest rozpatrywana jako część domeny. Ponieważ nazwa domeny może występować w wielu poddrzewach, może też występować w wielu domenach. Krótko mówiąc, domena jest właśnie poddrzewem przestrzeni nazw domen. Hosty są reprezentowane przez nazwy domen. Hosty to nazwy domen, które wskazują na informacje o indywidualnych hostach, a domena zawiera wszystkie hosty, dla których nazwy domen znajdują się w jej granicach. Hosty są powiazane logicznie, często przynależnością geograficzną lub

organizacyjną, a nie tylko siecią czy adresem lub rodzajem sprzętu. Można mieć dziesięć różnych hostów, każdy z nich w innej sieci i nawet w różnych państwach, wszystkie zaś w tej samej domenie.

Mechanizm DNS jest realizowany w oparciu o hierarchiczny podział sieci na *domeny* (ang.domains). Pierwszym etapem podziału są tzw. domeny najwyższego poziomu. W regularnym użytku znajduje się siedem takich domen:

ARPA -domena na wewnętrzny użytek Internetu;

COM -domena dla zastosowań komercyjnych; (ang.commercial), t.j Hewlett Packard(hp.com), Sun Microsystems(sun.com) i IBM(ibm.com),

EDU -domena instytucji edukacyjnych;organizacje oświatowe (ang. educational), t.j Uniwersytet w Białymstoku (białystok.edu),

GOV -domena rządowa; organizacje rządowe (ang.governement), t.j NASA (nasa.gov) i National Science Foundation (nsf.gov),

MIL -domena organizacji wojskowych (ang.military), t.j U.S.Army (army.mil) i Navy (navy.mil),

ORG -domena zastosowań nie komercyjnych, t.j Electronic Frontier Foundation (eff.org),

NET -organizacje tworzące sieci (ang.networking), główne centra kontroli pracy sieci t.j NSFNET (nsf.net),

INT -organizacje międzynarodowe (ang.international), t.j NATO (nato.int),

BIZ -domena organizacji biznesowych,

INFO -domena informacyjna.

Dalsze stopnie podziału nazywane są *poddomenami* albo *subdomenami* (ang.subdomains). Rozpatrzmy przykładową nazwę:

```
uniwersytet.wiedza_tajemna.białystok.edu.pl
```

Widoczna jest nazwa kategorii(.edu) i oznaczenie kraju(.pl). Poprzedzone są one nazwą domeny białystok, oraz dwiema nazwami poddomen (wiedza_tajemna oraz uniwersytet). Możliwe jest skrótowe odwoływanie się do sieci przez dwa komputery tej samej domeny lub poddomeny dla przykładu komputer:

```
sekretariat.wiedza_tajemna.białystok.edu.pl
```

może odwołać się do komputera:

```
kawiarnia.wiedza_tajemna.białystok.edu.pl
```

podając jedynie skrótową nazwę kawiarnia.

1.4 Delegacje i strefy

W DNS-ie każda domena może być zarządzana przez inną organizację. Każda domena może podzielić swoją domenę na pewną liczbę subdomen i przekazać odpowiedzialność za nie innym organizacjom. Jednym z głównych celów projektu systemu nazw domen jest decentralizacja zarzadzania. Osiąga się to za pomocą *delegacji* (ang.delegation). Delegowana domena działa podobnie jak zadania delegowane(przydzielane) w pracy. Dyrektor może podzielić wielki projekt na mniejsze zadania i delegować odpowiedzialność za każde z nich do różnych pracowników. Organizacja zarządzająca domeną może podzielić ją na subdomeny, a każda z nich może być delegowana do innej organizacji, która staje się odpowiedzialna za utrzymanie wszystkich danych w tej subdomenie. Może swobodnie zmieniać dane i nawet podzielić swoją subdomenę na jeszcze więcej subdomen, które deleguje dalej.

Domena rodzicielska zawiera tylko wskaźniki do źródeł danych subdomeny, więc może odsyłać tam zapytania. Nie wszystkie organizacje delegują całą domenę, podobnie jak nie wszyscy dyrektorzy delegują całą pracę. Domena może mieć kilka subdomen i zawierać również host, który do nich nie należy. Programy, które zapamiętują informacje o przestrzeni nazw domeny, nazywane są *serwerami nazw*. Zasadniczo serwery nazw dysponują całkowitą informacją o pewnej części przestrzeni nazw domeny,nazywanej *strefą*. Różnica między strefą a domeną jest ważna, ale subtelna. Wszystkie domeny najwyższego poziomu i wiele domen drugiego i niższych poziomów, jest dzielonych na mniejsze, łatwiejsze do zarządzania jednostki metodą delegacji. Te jednostki są nazywane strefami. Strefa zawiera te nazwy domen, które zawiera domena z taką samą nazwą, poza nazwami delegowanych subdomen. Jeżeli subdomena domeny nie jest delegowana dalej, strefa zawiera nazwy domen i dane subdomeny.

Specyfikacja DNS definiuje dwa rodzaje serwerów nazw:

1 podstawowy serwer główny (ang.primary masters)

2 drugorzędny serwer główny (ang.secondary masters)

Podstawowy serwer główny nazw strefy czyta dane strefy z pliku. Drugorzędny serwer główny nazw strefy dostaje dane strefy od innego serwera nazw, wiarygodnego dla strefy, dla której jest serwerem głównym. Często serwer główny jest podstawowym serwerem głównym strefy, lecz nie zawsze: drugorzędny serwer główny może ładować dane strefy od innych serwerów drugorzędnych. Po uruchumieniu drugorzędny serwer kontaktuje się ze swoim głównym serwerem nazw i jeśli to konieczne, pobiera dane strefy. Nazywa się to przekazaniem strefy (ang.zone transfer). Obecnie drugorzędny serwer główny częściej jest określany jako pomocniczy (ang.slave), chociaż wiele osób ciągle jeszcze nazywa go drugorzędnym. DNS dostarcza dwa rodzaje serwerów nazw w celu ułatwienia zarządzania, rozładowania obciążenia i eliminowania skutków awarii.

1.5 Resolwery

Programy nazywane serwerami nazw (ang.name servers) tworzą serwerową część mechanizmu klient-serwer DNS-u. Serwery nazw zawierające informacje o jakimś segmencie bazy danych i udostępniające je klientom nazywane są *resolwerami* (ang.resolvers). Resolwery to często procedury biblioteczne, które tworzą zapytania i przesyłają jej przez sieć do serwerów nazw. Resolwer to klient, który ma dostęp do serwerów nazw. Działające na hoście programy, potrzebując informacji z przestrzeni nazw domeny, stosują program resolwera. Obsługuje on :

- zapytania serwera nazw
- interpretację odpowiedzi(która może być rejestrem zasobu lub błędem)
- zwrot informacji do programów, które o nią prosiły.

Serwery nazw są biegłe w odzyskiwaniu danych z przestrzeni nazw domen.

Muszą takie być, biorąc pod uwagę ograniczone możliwości niektórych resolwerów. Nie tylko przekazują dane o strefach, dla których są wiarygodne, mogą też w przestrzeni nazw domeny wyszukiwać dane, dla których nie są wiarygodne. Ten proces nazywa się *rozróżnianiem nazw* (ang.name resolution) lub po prostu *rozróżnianiem*. Główny serwer nazw wie, gdzie znajdują sie wiarygodne serwery nazw dla każdej z domen najwyższego poziomu. Pytane o nazwę domeny, główne serwery nazw mogą dostarczyć co najmniej nazwy i adresy serwerów nazw wiarygodnych dla domeny najwyższego poziomu, w której znajduje się nazwa domeny. Serwery nazw głównego poziomu mogą dostarczyć listę serwerów nazw, wiarygodnych dla domeny drugiego poziomu,w której znajduje się nazwa domeny. Każdy serwer nazw, podaje informację, jak dostać się "bliżej" poszukiwanej odpowiedzi, lub sam dostarcza tę odpowiedź.

1.5.1 Rekurencja

Rekurencja lub *rozróżnianie rekurencyjne* jest nazwą procesu stosowanego przez serwery nazw, gdy otrzymują zapytanie rekurencyjne. Przy rekurencji resolwer wysyła do serwera nazw pytanie rekurencyjne o konkretną nazwę domeny. Pytany serwer nazw jest zobowiązany odpowiedzieć żądanymi danymi albo komunikatem o błędzie, określającym, że dane tego rodzaju lub wyszczególniona nazwa domeny nie istnieją. Serwer nazw nie może skierować zapytania do innego,ponieważ pytanie było rekurencyjne. Gdy pytany serwer nazw nie jest wiarygodny dla żądanych danych, będzie musiał pytać inne serwery nazw, by znaleźć odpowiedź. Może wysyłać do nich zapytania rekurencyjne, zobowiązując do znalezienia i zwrotu odpowiedzi.

Serwer nazw otrzymując zapytanie rekurencyjne, na które nie może sam odpowiedzieć, będzie pytał "najbliższe znane" serwey nazw. Są to serwery

wiarygodne dla strefy najbliższej nazwy szukanej domeny. Stosowanie najbliższych znanych serwerów nazw zapewnia, że rozróżnianie trwa tak krótko, jak tylko jest to możliwe.

Serwer nazw, który otrzymał zapytanie rekurencyjne, zawsze przesyła to samo pytanie, które otrzymał od resolwera. Nigdy nie przesyła zapytań jawnie o serwery nazw, chociaż ta informacja też znajduje się w przestrzeni nazw. Przesyłanie jawnego pytania może spowodować problemy.

1.5.2 Iteracja

Iteracja lub *rozróżnienie iteracyjne* odnosi się do procesu stosowanego przez serwery nazw, gdy otrzymują zapytanie iteracyjne. Rozróżnienie iteracyjne przesyła do pytającego najlepszą znaną mu odpowiedź. Nie potrzeba żadnych dodatkowych pytań. Pytany serwer nazw radzi się swoich lokalnych danych szukając właściwych danych. Jeśli ich nie znajduje, odsyła najlepszą możliwą odpowiedź, która pomoże kontynuować rozróżnianie. Zwykle jest to nazwa domeny i adresy najbliższych znanych serwerów nazw.

Resolwer pyta lokalny serwer nazw, który przesyła zapytania pewnej liczbie innych serwerów nazw w poszukiwaniu odpowiedzi. Każdy pytany serwer nazw odnosi to do innych serwerów nazw, które są wiarygodne dla strefy znajdującej się niżej w przestrzeni nazw i bliżej nazwy szukanej domeny. W końcu lokalny serwer nazw pyta wiarygodny serwer nazw, który zwraca odpowiedź.

1.5.3 Zapytania odwrotne

Zapytanie odwrotne polega na poszukiwaniu nazwy domeny dla znanego adresu IP i jest przetwarzane wyłącznie przez serwery nazw otrzymujące zapytanie. Sprawdzają one wszystkie swoje dane lokalne szukając podanej informacji i zwracają, jeśli to możliwe, nazwę domeny, która je indeksuje. Jeśli nie mogą znaleźć danych, kończą proces przetwarzania. Nie podejmują żadnych prób przekazania zapytania do innego serwera nazw.

Ponieważ każdy serwer nazw zna tylko część całkowitej przestrzeni nazw domeny, przy zapytaniu odwrotnym nigdy nie ma gwarancji zwrotu odpowiedzi. Na przykład jeśli serwer nazw otrzymuje zapytanie odwrotne o adres IP, o którym nic nie wie, to nie tylko nie może zwrócić odpowiedzi, ale także nie wie, czy taki adres IP istnieje, ponieważ zawiera tylko część bazy danych DNS. Co więcej, realizacja zapytań odwrotnych jest według specyfikacji DNS opcjonalna.

1.6 Buforowanie

Rozróżnianie może wydawać się procesem bardzo zagmatwanym i niewygodnym dla kogoś przyzwyczajonego do prostych przeszukiwań przez tabelę ho-

stów. W rzeczywistości przebiega on dość szybko. Znacznie przyśpiesza go między innymi *buforowanie* (ang.caching). Serwer nazw przetwarzający zapytanie rekurencyjne może wysyłać na zewnątrz kilka pytań, by znaleźć odpowiedź. Przy okazji uzyskuje dużo informacji o przestrzeni nazw domen. Za każdym razem, gdy odnosi się do innej listy serwerów nazw, dowiaduje się, że są one wiarygodne dla jakiejś strefy i uczy się ich adresów. Na zakończenie rozróżniania, gdy już znajdzie dane potrzebne do odpowiedzi na pierwotne zapytanie, może je również przechować. W ten sposób, jeśli pytanie o domenę powtórzy się, to odpowiedź jest natychmiastowa i nie wymaga rozróżnienia.

1.7 Jak działa DNS?

Oto przykład działania systemu DNS. Użytkownik komputera wpisuje w swojej przeglądarce stron WWW adres internetowy 'pl.wikipedia.org'. Przeglądarka musi znać adres IP komputera będącego serwerem WWW dla tej strony. Cały proces przebiega następująco:

1. Przeglądarka wysyła do serwera DNS zdefiniowanego w konfiguracji systemu operacyjnego zapytanie:
   'Jaki jest adres IP komputera pl.wikipedia.org?'

2. Serwer DNS wysyła do jednego z 13 serwerów głównych np. tego o adresie IP 198.41.0.4 zapytanie 'Czy znasz adres IP komputera pl.wikipedia.org?'

3. Serwer 198.41.0.4 odpowiada:
   'Nie znam, ale dla domeny 'org' serwerami są 204.74.112.1 oraz 204.74.113.1'

4. Serwer DNS wysyła do jednego z tych 2 serwerów np. tego o adresie IP 204.74.112.1 zapytanie 'Czy znasz adres IP komputera pl.wikipedia.org?'

5. Serwer 204.74.112.1 odpowiada:
   'Nie znam, ale dla domeny wikipedia.org serwerami są 216.21.226.87 i 216.21.234.87'

6. Serwer DNS wysyła do jednego z tych 2 serwerów np. tego o adresie IP 216.21.226.87 zapytanie 'Czy znasz adres IP komputera pl.wikipedia.org?'

7. Serwer 216.21.226.87 odpowiada:
   'pl.wikipedia.org ma adres IP 130.94.122.197'

8. Serwer DNS odpowiada przeglądarce
   'pl.wikipedia.org ma adres IP 130.94.122.197'

9. Przeglądarka wysyła pod adres IP 130.94.122.197 żądanie wyświetlenia strony WWW o adresie pl.wikipedia.org.

10. Serwer WWW o adresie IP 130.94.122.197 odsyła treść tej strony, którą użytkownik może obejrzeć.

1.8 Podsumowanie

System DNS posiada następujace cechy:

1. Nie ma jednej centralnej bazy danych adresów IP i nazw.
   Najważniejsze jest te 13 serwerów, które są rozrzucone na różnych kontynentach.

2. Serwery DNS przechowują dane tylko wybranych domen.

3. Każda domena ma co najmniej 2 serwery DNS obsługujące ją, jeśli więc nawet któryś z nich będzie nieczynny to drugi może przejąć jego zadanie.

4. Serwery DNS przechowują przez pewien czas odpowiedzi z innych serwerów, a więc proces zamiany nazw na adresy IP jest często krótszy niż w podanym przykładzie.

5. Każdy komputer może mieć wiele różnych nazw. Na przykład komputer o adresie IP 130.94.122.197 ma nazwę 'pl.wikipedia.org' oraz 'de.wikipedia.org'

6. Czasami pod jedną nazwą może kryć się więcej niż 1 komputer po to aby jeśli jeden z nich zawiedzie inny mógł spełnić jego rolę.

7. Jeśli chcemy przenieść serwer WWW na inny szybszy komputer, z lepszym łączem ale z innym adresem IP to nie musimy zmieniać adresu WWW strony a jedynie w serwerze DNS obsługującym domenę poprawiamy odpowiedni wpis o adresie IP.

8. Protokół DNS posługuje się do komunikacji głównie protokołem UDP

Rozdział 2

Realizacja DNS

2.1 Implementacja DNS

Istnieją dwie implementacje usługi DNS: BIND (ang. Berkeley Internet Name Domain) oraz djbdns (nazwa od nazwiska twórcy D.J.Bernsteina).

BIND (zob. [2]) jest jednym z najpopularniejszych serwerów DNS wykorzystywanym w systemach Unix i Linux. Wielu użytkowników globalnej sieci bezwiednie korzysta z serwera BIND, kiedy ich przeglądarka WWW odpytuje go o adres IP komputera udostępniającego interesującą ich stronę. BIND był rozwijany od wczesnych lat 80. XX wieku w ramach projektów DARPA. W połowie dekady pracę nad serwerem przejęła korporacja DEC. Jednym z jej pracowników biorących udział w projektowaniu BIND-a był Paul Vixie, który kontynuował swoją pracę po opuszczeniu tej firmy. Potem stał się jednym z założycieli ISC (ang.Internet Software Consortium), organizacji zarządzającej standardami Internetu, która przejęła prace nad dalszym rozwojem BIND-a.

BIND powstał w czasach, kiedy z Internetu korzystali wyłącznie naukowcy i nikomu nie śniło się, że w sieci mogą znaleźć się miliony ludzi. Jest to stare oprogramowanie obarczone wieloma naleciałościami. Mimo to sprawnie funkcjonuje na wielu serwerach, także na tych największych.

BIND dostępny jest na wiele platform, w tym również na Linuksa. Aktualną wersję programu można znaleźć w Internecie pod adresem:

`http://www.isc.org/products/BIND/`

Istnieją dwie główne edycje programu: BIND 4 oraz BIND 8. Ze względów bezpieczeństwa najlepiej używać nowszej wersji, która jest wolna od wcześniejszych błędów.

Drugą implementacją DNS jest *djbdns* (zob [3]. *djbdns* to zestaw programów typu DNS, których autorem jest Daniel J. Bernstein, autor bardzo popularnego serwera poczty QMail. Programy powstały w odpowiedzi na szereg błędów, dziur i niedoskonałości szeroko stosowanego BINDa. *djbdns* to implementacja zabezpieczeń DNS. Jej sposób działania bardzo różni się od tego z Bind.

2.2 Konfiguracja DNS

Moja przykładowa domena jest przeznaczona dla Instytutu Matematyki. Nosi ona nazwę *im.uwb.edu.pl*. Dane DNS-u zawierają wiele plików. Jeden plik mapuje wszystkie nazwy hosta na adresy. Inne pliki mapują z powrotem adresy na nazwy hosta. Każda sieć ma swoje własne pliki do mapowania odwrotnego. Plik mapujący nazwy hosta na adresy jest nazywany `db.DOMAIN`. Pliki mapujące adresy na nazwy hosta są nazywane `db.ADDR`, gdzie ADDR jest numerem sieci (bez końcowych zer). W moim przykładzie te pliki będą się nazywały db.192.168.0 i db.192.168.1; po jednym dla każdej sieci. Początek nazwy pliku (db) to skrót od *database* (baza danych). Zbiór plików `db.DOMAIN` i `db.ADDR` będę nazywać `plikami db` lub `plikami bazy danych DNS-u`. Istnieją jeszcze inne pliki danych: `db.cache` i `db.127.0.0`. Są to pliki stałe. Każdy serwer nazw musi je mieć i dla każdego są one mniej więcej takie same. Aby powiązać wszystkie pliki `db` razem,serwer nazw potrzebuje pliku inicjacyjnego-dla BIND-a wersji 4 tym plikiem zwykle jest `/etc/named.boot`, a dla BIND-a wersji 8 – `/etc/named.conf`. Pliki db są specyficzne dla DNS-u, zaś plik inicjacyjny dla realizacji serwera nazw, w tym wypadku BIND-a. Ponieważ używam najnowszą wersję BIND 9, więc plik inicjacyjny to `/etc/named.conf`. Także nie będę więcej odwoływać się do starszych wersji.

2.2.1 Plik db

Większość pozycji w plikach db jest nazywana *rekordami zasobów* (ang. *resource records*) DNS-u. Sprawdzenia DNS-u są nie czułe na wielkość liter, więc można wprowadzać nazwy w swoich plikach db literami wielkimi, małymi lub mieszając oba rodzaje. Rekordy zasobów muszą zaczynać się w pierwszej kolumnie. Kolejność rekordów zasobów w plikach db jest następująca:

SOA – wskazuje pełnomocnictwo dla tych danych strefy,

NS – wymienia serwery nazw dla tej strefy,

MX – trasowanie poczty elektronicznej,

A – mapowanie nazw na adresy,

PTR – mapowanie adresów na nazwy,

CNAME – nazwa kanoniczna dla aliasów.

Pliki db są łatwiejsze do czytania, jeśli zawierają komentarze i puste wiersze. Komentarze zaczynają się średnikiem i kończą się z końcem wiersza. Serwer nazw ignoruje komentarze i puste wiersze.

Rekordy SOA

Pierwszą pozycją w każdym z plików db jest rekord zasobu SOA (początek pełnomocnictwa). Rekord SOA wskazuje, że serwer nazw jest najlepszym źródłem danych w tej strefie. Rekord SOA jest wymagany w każdym pliku `db.DOMAIN` i `db.ADDR`. W pliku db może istnieć jeden i tylko jeden rekord SOA. Dodałam następujące rekordy SOA do pliku *db.im.uwb.edu.pl*:

```
im.uwb.edu.pl. IN SOA omega.im.uwb.edu.pl. sysadm.math.uwb.edu.pl. (
                      1          ; Numer seryjny
                      10800      ; Odświeżanie po 3 godzinach
                      3600       ; Ponawianie po jednej godzinie
                      604800     ; Ważność 1 tydzień
                      86400 )    ; Minimalny TTL 1 dzień
```

Nazwa *im.uwb.edu.pl* musi zaczynać się w pierwszej kolumnie pliku. IN oznacza Internet. To jest jedna z klas danych. Pole klasy jest opcjonalne. Jeśli klasa została pominięta, przyjmowana jest klasa IN. Pierwsza nazwa po SOA *omega.im.uwb.edu.pl* jest nazwą podstawowego serwera nazw dla tych danych. Druga nazwa *sysadm.math.uwb.edu.pl* jest adresem poczty elektronicznej osoby opiekującej się danymi (jeśli zastąpię pierwszą kropkę znakiem @). Nawiasy pozwalają rekordowi SOA zajmować więcej niż jedną linię. Większość pól między nawiasami używana jest przez pomocnicze serwery nazw. Dodałam podobny rekord SOA do początku plików `db.192.168.0` i `db.192.168.1`. W plikach tych zmieniłam pierwszą nazwę w rekordzie SOA z *im.uwb.edu.pl* na nazwę właściwą dla domeny *in-addr.arpa*: odpowiednio `0.168.192.in-addr.arpa` i `1.168.192.in-addr.arpa`.

Rekordy NS

Następnymi pozycjami, które dodałam do każdego z tych plików, są rekordy zasobów NS (ang. *name server*). Dodałam jeden rekord NS dla każdego serwera nazw dla mojej strefy. Poniżej znajdują się rekordy NS z pliku *db.im.uwb.edu.pl* :

```
im.uwb.edu.pl. IN NS omega.im.uwb.edu.pl.
im.uwb.edu.pl. IN NS theta.im.uwb.edu.pl.
```

Rekordy te wskazują, że istnieją dwa serwery nazw dla strefy *im.uwb.edu.pl*. Znajdują się one na hostach *omega* i *theta*. Jako serwery nazw hosty wielosieciowe, jak *omega*, to doskonały wybór, ponieważ są „dobrze połączone" oraz bezpośrednio dostępne przez hosty z więcej niż jednej sieci. Podobnie jak w wypadku rekordu SOA, dodałam rekordy NS do plików `db.192.168.0` i `db.192.168.1`.

Adresy i rekordy aliasów

Następnie utworzyłam mapowanie nazw na adresy. Dodałam następujące rekordy zasobów do pliku `db.im.uwb.edu.pl` :

```
;
; Adresy hostów
;
localhost.im.uwb.edu.pl.  IN A  127.0.0.1
omega.im.uwb.edu.pl.      IN A  192.168.0.1
math.im.uwb.edu.pl.       IN A  192.168.0.2
theta.im.uwb.edu.pl.      IN A  192.168.0.3
;
;Hosty wielosieciowe
;
omega.im.uwb.edu.pl.    IN A     192.168.0.1
omega.im.uwb.edu.pl.    IN A     192.168.1.1
;
;Aliasy
;
www.im.uwb.edu.pl.     IN CNAME math.im.uwb.edu.pl.
ftp.im.uwb.edu.pl.     IN CNAME math.im.uwb.edu.pl.
mail.im.uwb.edu.pl.    IN CNAME math.im.uwb.edu.pl.
matfiz.im.uwb.edu.pl.  IN CNAME math.im.uwb.edu.pl.
```

„A" oznacza adres i każdy rekord zasobu mapuje nazwę na adres. Host *omega* działa jako ruter. Ma dwa adresy skojarzone ze swoją nazwą i dlatego dwa rekordy adresowe. Trzeci blok zawiera tablicę aliasów hostów. Rekord CNAME mapuje aliasy na ich nazwy kanoniczne. Serwer nazw posługuje się rekordami CNAME w inny sposób, niż są obsługiwane w tabeli hosta. Kiedy poszukując nazwy znajdzie rekord CNAME, zastępuje ją nazwą kanoniczną i szuka nowej. Na przykład, gdy serwer nazw będzie szukał *www*, znajdzie rekord CNAME wskazując na *math.im.uwb.edu.pl.*. Zawsze należy stosować nazwę kanoniczną w części danych (na końcu) rekordu zasobów. Rekordy NS, które utworzyłam, wykorzystują nazwy kanoniczne.

Rekordy PTR

Rekord PTR spełnia w plikach tzw. baz danych odwrotnych analogiczną rolę do rekordu A - łączy numery IP z adresami symbolicznymi. Służy on do konstrukcji hierarchii domen in-addr.arpa. Plik `db.192.168.0` mapuje adresy na nazwy hostów dla sieci 192.168.0. Rekordy zasobów DNS stosowane do tego mapowania to rekordy wskaźników PTR (ang. *pointer*). Dla każdego interfejsu hosta w sieci istnieje po jednym rekordzie. Oto rekordy PTR, które dodaliśmy dla sieci 192.168.0:

```
1.0.168.192.in-addr.arpa.     IN PTR  omega.im.uwb.edu.pl.
2.0.168.192.in-addr.arpa.     IN PTR  math.im.uwb.edu.pl.
3.0.168.192.in-addr.arpa.     IN PTR  theta.im.uwb.edu.pl.
```

Adresy powinny wskazywać tylko na jedną nazwę, kanoniczną. Można utworzyć dwa rekordy PTR, ale większość systemów zobaczy tylko jedną nazwę dla adresu.

CNAME
Rekord ten pozwala deklarować kilka nazw alternatywnych dla każdego komputera. Pozwala to na wygodne manipulowanie powiązaniem funkcji komputerów (związanych z łatwym do zapamiętania adresem) z fizyczną ich lokalizacją. Na przykład próbujemy swoich sił w stawianiu serwera WWW. Wybieramy do tego celu jakiś najmniej obciążony komputer, nadajemy mu alias „www.domena", uruchamiamy serwer, wypełniamy go wstępnymi informacjami i ogłaszamy o jego istnieniu.

Serwer nazw potrzebuje jednego dodatkowego pliku `db.ADDR`, który zawierałby sprzężenie zwrotne sieci: specjalny adres, którego używa host do bezpośredniego powrotu. Ta sieć prawie zawsze ma numer 127.0.0, a host prawie zawsze numer 127.0.0.1, dlatego nazwą pliku jest `db.127.0.0`. Wygląda on jak inne pliki `db.ADDR`. A oto zawartość pliku `db.127.0.0`:

```
0.0.127.in-addr.arpa. IN SOA omega.im.uwb.edu.pl. sysadm.math.uwb.edu.pl. (
                      1           ; Numer seryjny
                      10800       ; Odświeżanie po 3 godzinach
                      3600        ; Ponawianie po jednej godzinie
                      604800      ; Ważność 1 tydzień
                      86400 )     ; Minimalny TTL 1 dzień
0.0.127.in-addr.arpa.  IN NS   omega.im.uwb.edu.pl.
0.0.127.in-addr.arpa.  IN NS   theta.im.uwb.edu.pl.

1.0.0.127.in-addr.arpa. IN PTR localhost.
```

2.2.2 Plik konfiguracyjny

Po utworzeniu plików db trzeba poinformować serwer nazw, że ma je przeczytać. Dla BIND-a mechanizmem wskazywania serwerowi jego plików db jest plik konfiguracyjny. W BIND-zie 4 komentarze w pliku konfiguracyjnym są takie same jak w plikach db zaczynają się średnikiem i kończą się przy końcu wiersza:

```
;To jest komentarz
```

w BIND-zie 8 można używać któregokolwiek z 3 stylów komentarzy:

```
/* To jest komentarz w stylu C */
// To jest komentarz w stylu C
#  To jest komentarz w stylu shella
```

Plik konfiguracyjny zawiera wiersz wskazujący katalog, w którym są umieszczone pliki. Serwer nazw zmienia na niego swój katalog przed czytaniem plików, co pozwala na umieszczanie względnych nazw plików zamiast nazw z pełną ścieżką dostępu. Oto wiersz katalogu:

```
options  {
         directory  "/usr/local/named";
         // Umieść tu dodatkowe opcje.
};
```

W pliku konfiguracyjnym jest dozwolona tylko jedna instrukcja *options*. Oto kompletna wersja pliku `/etc/named.conf`:

```
// Plik konfiguracyjny BIND

options  {
         directory  "/usr/local/named";
         // Umieść tu dodatkowe opcje.
};

zone "im.uwb.edu.pl" in {
        type master;
        file "db.im.uwb.edu.pl";
};

zone "0.168.192.in-addr.arpa" in {
        type master;
        file "db.192.168.0";
};

zone "1.168.192.in-addr.arpa" in {
        type master;
        file "db.192.168.1";
};

zone "0.0.127.in-addr.arpa" in {
        type master;
        file "db.127.0.0";
};

zone "." in {
         type hint;
         file "db.cache";
};
```

Rozdział 3

Aplikacja wspomagająca obsługę DNS

W ramach niniejszej pracy została opracowana baza danych oraz aplikacja WWW wspomagająca obsługę serwisu DNS. Jako silnik bazy danych wybrałam MySQL, natomiast aplikację napisałam w języku Meta-HTML. Jest ona dostępna w internecie pod adresem:

```
http://math.uwb.edu.pl/dns/
```

W aplikacji możemy wyróżnić dwie części. W części ogólnodostępnej dla wszystkich obejrzeć możemy listę wszystkich obsługiwanych domen i ich podstawowe parametry. W drugiej części, po zalogowaniu się, możemy dodawać, usuwać i modyfikować domeny oraz ich parametry.

3.1 Interfejs użytkownika

Po uruchomieniu aplikacji, pod podanym adresem, mamy dwie opcje do wyboru: pokaż domeny, logowanie. Wybranie pierwszej z nich powoduje wyświetlenie tabeli zawierającej obsługiwane domeny wraz z parametrami t.j: nazwa, typ, administrator, miejsce rejestracji, data utworzenia, data wygaśnięcia (rys. 3.1).

Nazwa	Typ	Registrar	Data utworzenia	Data wygaśnięcia
domena.pl	slave	NASK	2006-10-20	2007-01-20
im.uwb.edu.pl	master	BIAMAN	2003-10-15	2008-10-15

Rysunek 3.1: Lista obsługiwanych domen.

Po kliknięciu nazwy domeny otwiera się strona zawierająca dokładne dane o domenie, czyli: hosty, aliasy, serwery nazw, rekordy MX oraz serwer podstawowy (rys. 3.2).

Nazwa domeny: im.uwb.edu.pl
Typ: master
Registrar: BIAMAN
Data utworzenia: 2003-10-15
Data wygaśnięcia: 2008-10-15

Serwery nazw

Nazwa
omega.uwb.edu.pl
theta.uwb.edu.pl

Rekordy MX

Priorytet	Nazwa
010	math.im.uwb.edu.pl

Hosty

Nazwa	IP
omega	192.168.0.1
math	192.168.0.2
theta	192.168.0.3

Aliasy

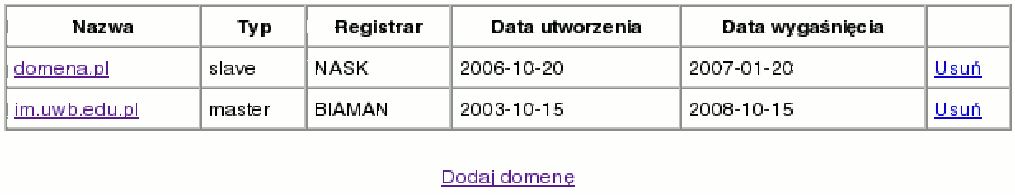
Alias	Host
www	math
mail	math
ftp	math
matfiz	math

Rysunek 3.2: Parametry wybranej domeny.

Po zalogowaniu się do aplikacji jako administrator będziemy mogli wprowadzać zmiany, dodawać i usuwać domeny (rys. 3.3). W aplikacji uproszczono kwestię praw dostępu. Oznacza to, że jest tylko jeden uprzywilejowany użytkownik - administrator, który ma pełne prawa. Nie jest to jakimś ograniczeniem, dlatego że obsługa DNS w systemie wymaga uprawnień administratora, i tylko on może konfigurować tę usługę.

Rysunek 3.3: Ekran logowania do aplikacji.

W panelu administratora aplikacji mamy listę wszystkich obsługiwanych domen. Na tym etapie możemy wybrać domenę aby ją modyfikować, możemy usunąć domenę lub dodać nową domenę (rys. 3.4).



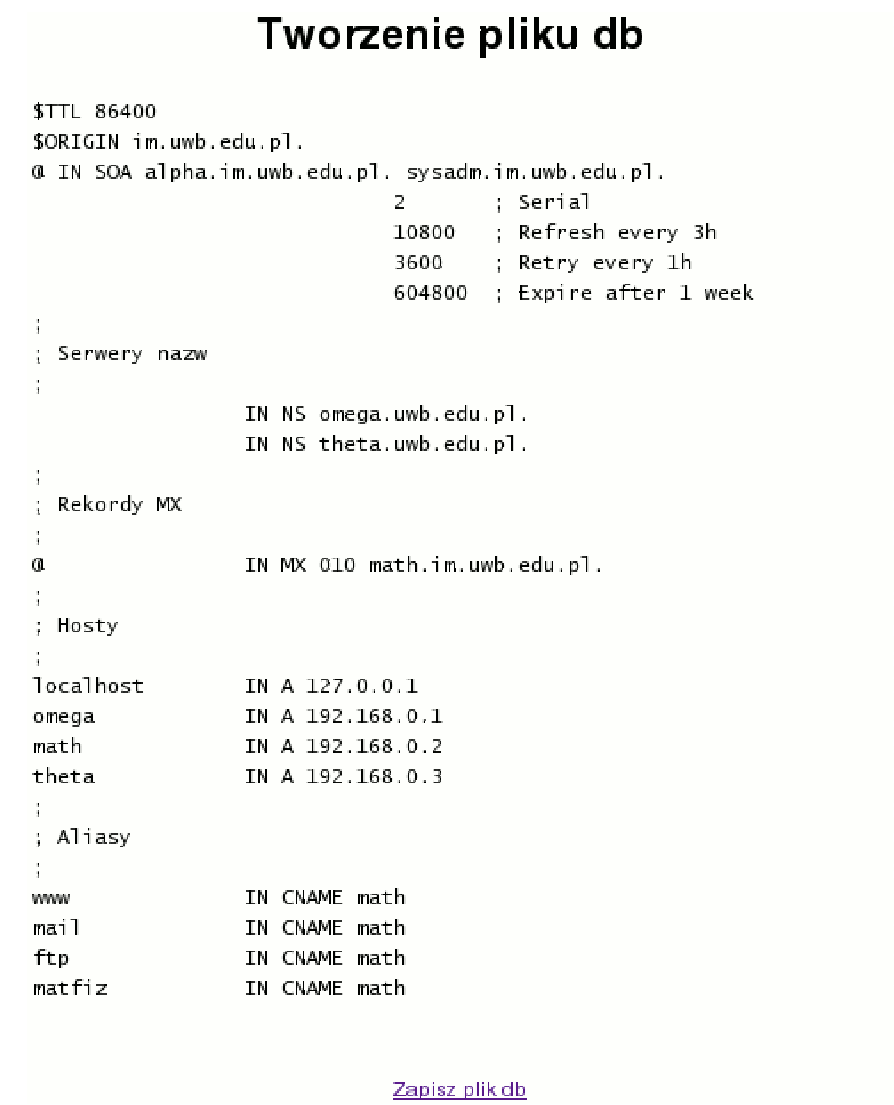
Rysunek 3.4: Panel administracyjny aplikacji.

Po wybraniu domeny otrzymamy pełną informację o tej domenie jaka do tej pory została wprowadzona do bazy danych. Wszystkie te dane możemy modyfikować. Możemy także dodawać i usuwać informacje na temat hostów, aliasów, serwerów nazw, rekordów MX oraz serwera podstawowego (rys. 3.5).

Rysunek 3.5: Fragment ekranu edycji parametrów domeny.

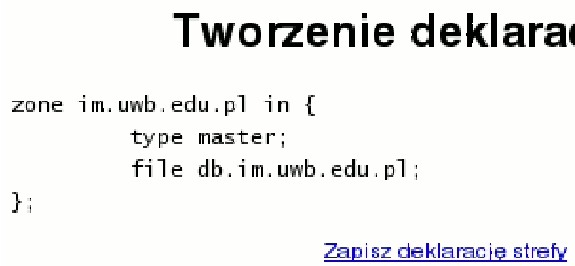
3.2 Współpraca z BIND

Po wprowadzeniu zmian dla domeny jest możliwość wygenerowania pliku konfiguracyjnego strefy, żargonowo zwanego plikiem db (rys. 3.6). Plik ten jest gotowy do wstawienia go do konfiguracji DNS.



Rysunek 3.6: Tworzenie pliku db.

Możemy tutaj również wygenerować opis strefy do umieszczenia w pliku konfiguracyjnym `named.conf` (rys. 3.7).



Rysunek 3.7: Tworzenie deklaracji strefy.

Bibliografia

[1] Albitz P.,*Dns i Bind*, O'Really, 1998.

[2] Strona domowa projektu BIND:
`http://www.isc.org/bind/`

[3] Strona domowa projektu BIND:
`http://cr.yp.to/djbdns.html`

[4] Strona domowa projektu BIND:
`http://www.mysql.com/`

[5] `http://metahtml.sourceforge.net`