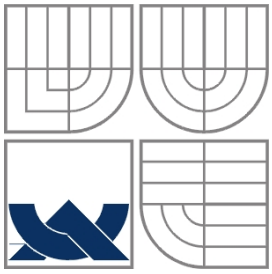
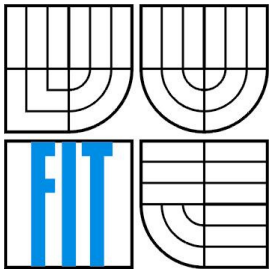


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SLUŽBA DNS S ROZŠÍŘENÍM O INFORMACE O POLOZE SLUŽEB

DNS SERVICE WITH EXTENSION OF THE INFORMATION ABOUT LOCATION

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE — Vladimír VESELÝ
AUTHOR

VEDOUCÍ PRÁCE — Ing. Rudolf ČEJKA
SUPERVISOR

BRNO 2007

Zadání

Služba DNS s rozšířením o informace o poloze služeb

DNS Service with Location Information Extension

Vedoucí:

Čejka Rudolf, Ing., CVT FIT VUT

Oponent:

Matoušek Petr, Ing., Ph.D., UIFS FIT VUT

Přihlášen:

Veselý Vladimír

Zadání:

- Nastudujte, jak funguje služba DNS (RFC 1034 a 1035);
- Seznamte se s problémy, se kterými se současný internet potýká z pohledu polohy a poskytování služeb;
- Nastudujte RFC 1876 a RFC 2782 pro navrhovaná řešení některých problémů;
- Zjistěte možnosti využití těchto rozšíření v praxi;

Kategorie:

Počítačové sítě

Literatura:

- RFC 1034: Domain Names - Concepts and Facilities
- RFC 1035: Domain Names - Implementation and Specification
- RFC 1876: A Means for Expressing Location Information in the Domain Name System
- RFC 2782: A DNS RR for specifying the location of services (DNS SRV)

Licenční smlouva

Kopie licenční smlouvy je uložena v archivu Fakulty informačních technologií při Vysokém učení technickém v Brně.

Abstrakt

S rozšiřujícím počtem uživatelů Internetu je nutné začít řešit problém optimálnějšího zpřístupnění klíčových služeb koncovému klientovi. Služba DNS jakožto všeobecně užívaná standardizovaná hierarchická distribuovaná databáze všech zařízení na Internetu nabízí velké možnosti v tomto odvětví činnosti.

Moje práce se zabývá existujícími rozšířeními systému DNS o informaci o poloze serveru či služby a způsobem distribuce dané informace koncovým klientům. Zároveň s tím seznamuje s problematikou vyvažování zátěže.

Klíčová slova

DNS, záznam LOC, záznam SRV, poloha serveru, poloha služby, Round Robin, GeoDNS, BIND, LSNAT, load sharing NAT, DNSProxy, doména, zóna.

Abstract

Accordingly to spreading of Internet users it is necessary to start solving the problem of optimized client access to key services. DNS service as widely used standard of hierarchy distributed database of all connected devices offers a great options in this branch of IT.

My thesis deals with existing extensions of DNS system about location of server or service. Simultaneously informs reader with issues about load balancing and improving network performance.

Keywords

DNS, LOC record, SRV record, location of server, location of service, Round Robin, GeoDNS, BIND, LSNAT, load sharing NAT, DNSProxy, domain, zone.

Služba DNS s rozšířením o informace o poloze služeb

Prohlášení

Tímto prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Rudolfa Čejky. Další cenné informace mi předali Ing. Petr Matoušek, Ph.D. a Ing. Vladimír Veselý. Touto cestou bych jim rád vyjádřil vděčnost a poděkování za profesionální odborné vedení, které mi poskytli a jež předcházelo a provázelo proces tvorby mé bakalářské práce.

Uvedl jsem všechny elektronické či literární prameny a publikace, ze kterých jsem čerpal.

Svým vlastnoručním podpisem stvrzuji, že práce ani jakákoli její část není plagiátem.

……
Vladimír VESELÝ
15. května 2007

Obsah

1 Úvod

1.1 Problematika

V současnosti se spojování a směrování klientů v Internetu přesunulo od původních problémů s konektivitou (nedostatečné množství linek či špatné parametry existujících linek) k problémům s optimalizacemi v přístupu ke službám. Ať už je či bude propustnost linek libovolné vysoká , nikdy nemůže být dostatečná vzhledem k jejich neustále se zvyšující zátěži. A tak se na delší vzdálenosti mezi klientem a serverem objevují problémy s interaktivitou způsobené prodlevami v komunikaci. Proto je snahou klienty směrovat na geograficky či topologicky nejbližší poskytovatele dané služby, co nejvíce zmírnit dopad vzdálenosti koncových bodů.

Tato práce si klade za cíl shromáždit existující možnosti využití služby DNS (jakožto celosvětově rozšířeného standardu) týkající se právě možných řešení integrace informace o poloze služby či serveru. V rámci tohoto tématu se pokusí též probrat některé přidružené problematiky jako distribuce zátěže či otázky bezpečnosti. Zároveň s tím se bude zabývat i řešeními, která nejsou vystavěna přímo na DNS, ale i na jiných přístupech a metodologiích.

1.2 Stručný obsah kapitol

1.2.1 Služba DNS

Tato kapitola čtenáře seznámí s historií vzniku, principy architektury i vyhledávání a fungování služby DNS. Je základním stavebním kamenem pro pochopení zbytku bakalářské práce, seznamuje s kořeny nutnými k hlubšímu proniknutí do popisované látky.

1.2.2 Záznam SRV

Popisuje funkci RR záznamu typu SRV, který je schopen specifikovat koncové zařízení, jež provozuje danou službu v rámci domény. Zabývá se jak syntaxí a sémantikou, tak i způsobem distribuce zátěže, kterou tento záznam umožňuje.

1.2.3 Záznam LOC

Zabývá se podrobným popisem funkce speciálního druhu RR záznamu, tzv. záznamu typu LOC. Obsahem kapitoly je vysvětlení syntaxe a sémantiky tohoto záznamu i jeho současného použití v praxi.

1.2.4 Load sharing NAT

Podrobně obeznamuje s technikou Load-sharing NAT, která je v současnosti často používaná při vyvažování zátěže provozu při přístupu k jedné službě.

1.2.5 Round Robin

Seznamuje s tímto specializovaným algoritmem rotování záznamu v seznamu navracených IP adres na klasický DNS dotaz. Kromě samotného principu fungování je algoritmus jednoduše demonstrován na reálném příkladu a základní informace jsou doplněny i o obecně uznávaná doporučení při implementaci a užívání této metody.

1.2.6 BIND

Lehké obeznámení s touto snad nejrozšířenější aplikací poskytující DNS server a DNS resolver. Informace zde obsažené jsou dobré k lepšímu pochopení následující kapitoly o iniciativě GeoDNS.

1.2.7 GeoDNS

Informuje o iniciativě vzniklé pro směrování klienta v závislosti na jeho geografické poloze získané přes jeho IP adresu. Obsahuje princip, nad kterým byl vystavěna pro program BIND a některé zpřesňující detaily své činnosti.

1.2.8 PlanetLab

Obeznámení s existencí této celosvětové počítačové laboratoře. Je zmiňována nejen proto, že slouží jako vývojová základna pro iniciativu DNSProxy, která využívá velmi zajímavého způsobu zjišťování klientovi nejblíže položenému uzlu poskytujícím danou službu, ale hlavně pro její neocenitelný přínos při rozvoji síťových technologií a metodologií.

1.2.9 DNSProxy

Kapitola čtenáře seznamuje s iniciativou pro službu DNS, která využívá metriky počtu hopů jako rozhodujícího kritéria pro odpovědi na DNS dotazy.

2 Služba DNS

2.1 Stručný popis

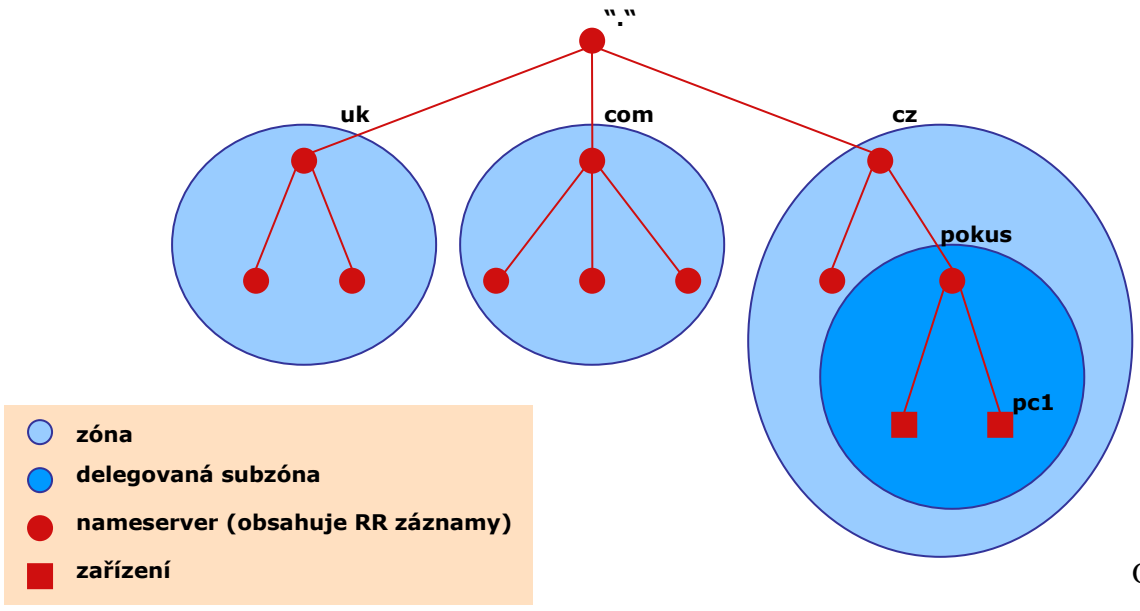
Než byla služba v roce 1987 uvedena do praxe, bylo mapování doménového jména na IP řešeno sdílením statického souboru HOSTS, který byl spravován jednou centrální autoritou. Avšak se vzrůstajícím počtem připojených zařízení na Internetu se stala situace neudržitelná, bylo čím dál tím těžší tento soubor obhospodařovat a distribuovat – nestíhal se dostatečně přizpůsobovat rychle měnícímu se okolí.

Byl tedy vytvořen standard DNS, navrhující distribuovanou databázi založenou na tzv. *nameserverech*, která by byla dostupná globálně všem zařízením a poskytovala by oporu i dalším protokolům a aplikacím. Služba DNS jako taková úzce souvisí s adresováním zařízení v Internetu. Její primární použití je v překladu doménových jmen (pro člověka snadněji zapamatovatelných reprezentací) na IP adresy (užívané počítači) a obráceně. Spolu s těmito údaji je však schopna uchovávat i jiné informace spojené s doménovými jmény.

2.2 Princip

Doménový prostor je ve službě DNS členěn do hierarchické stromové struktury, přičemž každý list v této struktuře uchovává informace o té dané konkrétní doméně. Doménové jméno se obvykle skládá z několika částí oddělených tečkami. Obrázek [Obr.1] ukazuje rozložení doménového jména i jeho reprezentaci v rámci struktury DNS:

<hostitel>.<subdoména>.<d. druhé úrovně>.<d. nejvyšší úrovně> např. pc1.pokus.cz



Obr.1

Domény nejvyšší úrovně (*TLD = top level domain*) jsou buď **národní** (např. CZ, EU, UK, SK) nebo **tématické** (např. GOV, COM, NET, ORG). Jednotlivé části doménového jména (řetězce znaků mezi tečkami) mohou mít maximálně 63 znaků (přičemž množina znaků je [a-z] [A-Z] [0-9] a „–“ pomlčka), celková délka doménového jména pak může být maximálně 255 znaků a členit se může až do 127 úrovní. Doménové jméno je necitlivé na velikost písmen (tzn. www.example.org je to samé co WwW.EXAmPLe.OrG). Doménový strom je rozdělen do zón, pod správou rozdílných administrativních organizací. Pro danou zónu poskytuje autoritativní informace organizací řízený autoritativní DNS server. Jak bylo řečeno na počátku, Internet je obrovská globální síť a její granulace na dílčí části a delegace pravomocí mnoha zodpovědným subjektům přispívá k její lepší správě. A právě protokol DNS pomáhá k rychlému šíření informací o jednotlivých zónách.

DNS servery můžeme rozdělit do několika skupin:

- **Primární server** – každá doména má právě jeden primární server, obsahuje úplné a nejaktuálnější informace, které spravují, a poskytují autoritativní odpovědi pro doménu;
- **Sekundární server** – každá doména musí mít alespoň jeden sekundární server, který je kopií primárního serveru (*zone transfers*) a slouží jednak jako záloha v případě výpadku primárního serveru, za druhé pak pro rozkládání provozní zátěže u velkých domén;
- **Caching-only server** – předává dotazy dalším nameserverům, odpověď od nich si schovává a pokud k němu dorazí od klienta stejný dotaz jako v minulosti, vyloví odpověď z historie (jedná se tedy o jistý druh vyrovnávací paměti, který službě DNS umožňuje snížit úroveň vytížení linek);

Kořenové servery (*root servers*) jsou páteří celého Internetu, poskytují kořenový zónový soubor ostatním DNS serverům po celé Síti, tedy informace o všech TLD doménách a umístění jejich autoritativních serverů. Kořenový zónový soubor je spravován organizací IANA, jež volí jednotlivé správce root serverů, kteří garantují jejich celoroční bezchybný provoz a jejich dostupnost:

Server	Operátor	IP adresa
A	VeriSign Naming and Directory Services	198.41.0.4
B	Information Sciences Institute	192.228.79.201
C	Cogent Communications	192.33.4.12
D	University of Maryland	128.8.10.90
E	NASA Ames Research Center	192.203.230.10
F	Internet Systems Consortium, Inc.	192.5.5.241
G	U.S. DOD Network Information Center	192.112.36.4
H	U.S. Army Research Lab	128.63.2.53
I	Autonomica/NORDUnet	192.36.148.17
J	VeriSign, Inc.	192.58.128.30
K	Reseaux IP Europeens Network Coordination Centre	193.0.14.129
L	Internet Corporation for Assigned Names and Numbers	198.32.64.12
M	WIDE Project	202.12.27.33

Serverů je sice aktuálně třináct, ale lokální uzly těchto serverů jsou rozsety po celém světě, bližší informace jsou k dispozici v Příloze [Priloha13.1]. Velmi zajímavým dokumentem je i RFC2870 [Rfc7], kde jsou shrnuty obecné i technické nároky na root servery stejně jako dosavadní zkušenosti s provozem a všeobecně uznaná doporučení (*best practices*).

2.3 Vyhledávání

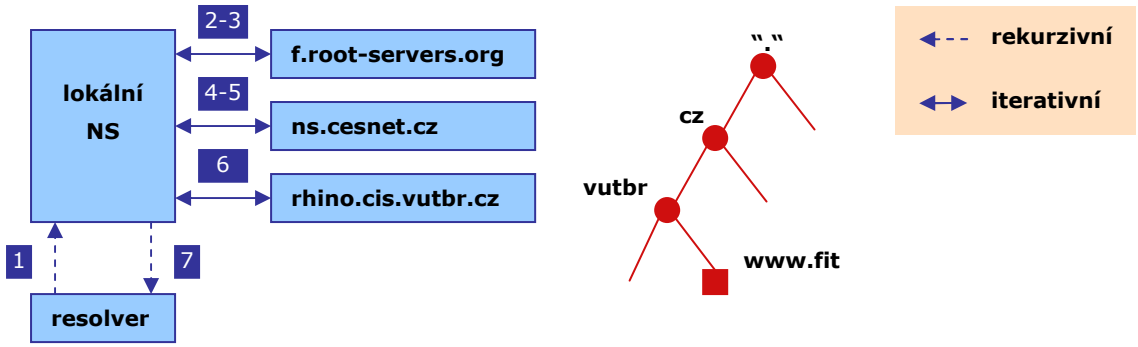
Nejprve se doménové jméno rozloží na dílčí části (oddělené tečkami), tímto jménem se postupuje zprava doleva (od nejobecnějšího k nejkonkrétnějšímu), což odpovídá pohybu DNS stromem od kořene k jednomu z jeho listů, který obsahuje námi hledaný záznam.

Pokud se na tento problém podíváme trochu podrobněji, tak každý počítač připojený do sítě Internet má ve své síťové konfiguraci nastavený lokální DNS server, na který se obrací pomocí tzv. dotazů (*query*). Každý DNS server zná adresy všech (v současnosti třinácti) kořenových DNS serverů a pokud není schopen uspokojivě odpovědět na dotaz klienta (nezná odpověď a nebo nemá výsledek dotazu uložený ve své vyrovnávací paměti *cache*), tak kontaktuje předpokládaný root server a od něj se sérií dotazů a delegovaných odpovědí dobere až k autoritativnímu serveru pro dané doménové jméno a původnímu klientovi vrátí požadovaný záznam.

Mějme následující příklad, kdy si student chce z domova prohlédnout své oblíbené stránky vlastní fakulty, do svého internetového vyhledávače tedy zadal adresu `www.fit.vutbr.cz`. Následující popis a obrázek [Obr.2] názorně ilustrují průběh vyhledávání.

1. *Resolver* (program starající se o překlad doménového jména na IP adresu, který je pravděpodobně součástí internetového prohlížeče) se po zkontrolování síťové konfigurace obrací na svůj lokální DNS server (IP 160.218.10.200) s dotazem;
2. Lokální server si projde svoji cache (tedy paměť výsledků doposud zodpovězených dotazů) a zjišťuje, že nezná odpověď, proto se obrátí na jeden z kořenových serverů, a to ten nejbližší možný lokální uzel umístěný v Praze `f.root-servers.org` (IP 192.5.5.241);
3. Tento kořenový server také nezná odpověď, ovšem zná autoritativní servery pro doménu nejvyšší úrovně CZ, poskytne tedy tazateli (lokálnímu DNS serveru) jejich IP adresy;
4. Lokální server si jednu z nich vybere, nejpravděpodobněji server Cesnetu `ns.ces.net` (IP 195.113.144.233) a pošle mu dotaz na klientem požadované doménové jméno;
5. Oslovený server opět nezná autoritativní odpověď, ale poskytne tazateli seznam jmenných serverů pro doménu `vutbr.cz`, což je server `rhino.cis.vutbr.cz`, tento server je zároveň i jmenným serverem pro doménu `fit.vutbr.cz`;

6. Lokální server pošle na adresu `rhino.cis.vutbr.cz` dotaz na požadované doménové jménu a ten mu odpoví, že server, který hledá (tedy `www.fit.vutbr.cz`) má IP adresu 147.229.9.22;
7. Lokální server odpoví původnímu dotazu *Resolveru*, že IP adresa, kterou potřebuje je stejná jako ta, kterou získal od nameserveru domény `vutbr.cz`.



Obr.2

Výše uvedený příklad z praxe je kompletní a sází na úplnou neznalost všech jmenných serverů po cestě. Ve skutečnosti je mnohem pravděpodobnější, že odpověď (nebo mezivýsledek odpovědi) na dotaz bude ve vyrovnávací paměti některého z postupně dotazovaných serverů.

Všimneme si druhů dotazů mezi resolverem a lokálním serverem a dále mezi lokálním serverem a dalšími jmennými servery. **Rekurzivní dotaz** znamená, že se dotazovaný (v našem případě lokální server) aktivně postará o vyhledání odpovědi a posílá získanou odpověď tazateli (u nás resolveru). Rekurzivní způsob sice zatěžuje server (musí si ukládat mezivýsledky, spotřebovává se procesorový čas na udržení relace), avšak snadno se s jeho pomocí buduje vyrovnávací paměť dotazů. Při **iterativním dotazu** server nic neřeší a tazatele deleguje na adresy jiných serverů, kterých by bylo podle něj vhodnější se zeptat k dosažení hledané odpovědi. Iterativní přístup neumožňuje budování vyrovnávací paměti (protože neznáme výsledek dotazu a jen přesměrujeme dotaz dál), na druhou stranu to ani v kořeni ani v bližších patrech DNS stromové struktury není vhodné, protože by docházelo k přílišnému zatěžování systémových prostředků, které vyřizují dotazy mnoha a mnoha tazatelů.

2.4 Záznamy

Výsledkem vyhledávání v hierarchické struktuře služby DNS je obvykle autoritativní odpověď v podobě RR záznamu (*resource record*), který obsahuje informace, které požadujeme (tj. adresu IP, doménové jméno či server pro zpracování elektronické pošty).

Záznamy si můžeme představit jako položky databáze s konkrétní syntaxí a předem určenou sémantikou, která je následovná:

```
[<owner>] [<ttl>] <class> <type> <rdata>
```

Přičemž tato struktura je podrobně rozepsána v RFC1034 [Rfc1] a RFC1035 [Rfc2]. Tak jako tak se tady zmiňme alespoň zevrubně.

Proměnná `<owner>` označuje jméno objektu (tedy jméno koncového zařízení nebo domény). Toto jméno je chápáno v kontextu celé domény, tzn. pokud je to např. `ws10` v rámci domény `example.org`, je majitelem záznamu chápáno zařízení `ws10.example.org`. Pokud se chceme zbavit tohoto kontextu můžeme proměnnou `<owner>` zakončit tečkou „ . “ a pak je definitivní a už se k němu přípona domény nepřidává.

`<ttl>` proměnná udává dobu, po kterou je záznam platný v případě, že s ním pracujeme jako s neautoritativní odpovědí, tzn. pokud ho používáme v rámci vyrovnávací paměti – jak dlouho je tento záznam platný poté, co je zapsán do cache. `<ttl>` se obvykle vynechává, je totiž většinou definována direktivou `$ttl` na začátku zónového souboru, případně se použije příslušná hodnota uvedená v záznamu typu SOA.

`<class>` udává třídu, v rámci které je tento záznam používán. Dnes je tato hodnota obvykle IN. Jsou však definovány i jiné systémy (v současnosti již spíše jako historický přežitek):

- IN (třída Internet);
- CH (třída Chaos);
- CS (třída CSNET);
- HS (třída Hesiod).

Hodnota `<type>` určuje jak už název napovídá typ RR záznamu. Má vliv na sémantiku záznamu a podmiňuje syntaxi nosného obsahu tohoto záznamu tedy hodnoty uložené v proměnné `<rdata>`. Některé z nich i s popisem jsou uvedeny v následující tabulce:

<type>	Např.:	Popis
A	www.sest3.com IN A 193.86.238.18	překlad doménového jména na IPv4
AAAA	ns.isc.org. IN AAAA 2001:4f8:0:2::13	překlad doménového jména na IPv6
PTR	147.229.9.22 IN PTR www.fit.vutbr.cz	překlad IP na doménové jméno
CNAME	www IN CNAME tereza.fit.vutbr.cz	alias pro doménové jméno
NS	google.com. IN NS ns1.google.com	nameserver pro danou doménu
MX	stud IN MX 10 kazi.fit.vutbr.cz	poštovní server pro danou doménu

Kromě základních klíčových typů jsou i mnoha standardy a implementacemi dodefinovány další typy a některé z nich jsou velmi podrobně rozebrány níže. Podrobný rozbor jednoho výtažku z fiktivního zónového souboru je pro větší názornost uveden v Příloze [Priloha13.2].

2.5 Užití v praxi

V současnosti je služba DNS celosvětově rozšířena a je základním stavebním kamenem operativnosti sítě Internet. Obvykle je na jednotlivých jmenných serverech implementována pomocí specializovaného programu jako např. BIND (který je podrobněji probrán v jedné z dalších kapitol), MSDNS, MyDNS, aj. Srovnání nejpoužívanějších programů je pak uvedeno na [7].

3 Záznam SRV

3.1 Stručný popis

Speciální druh záznam SRV definovaný přesně v dokumentu RFC2782 [Rfc5] zavádí do služby DNS specifické informace o dostupných službách poskytovaných v rámci dané domény (respektive serverech, které tyto služby umožňují).

3.2 Princip

Bez použití záznamu SRV v současnosti musí klientská strana buď znát adresu zařízení rovnou, nebo se pomocí broadcastu[I] zeptat – ani jeden z těchto způsobů však není ideální. První z nich se může ukázat jako špatný v případě náhlého výpadku služby, nedostatečně flexibilní a neschopný rychlé konvergence. Druhý pak zatěžuje zbytečným provozem síť.

Základní idea vychází z představy, že klient hledající v rámci dané domény protokol nebo server s požadovanou službou dostane díky těmto záznamům soupisku příslušných koncových zařízení. Došlo by tak k výraznému zrychlení a zkvalitnění služeb.

3.3 Syntaxe a sémantika

Jedná se o další RR záznam, platí pro něj tedy všechna pravidla a omezení jako pro jiné typy záznamů. Číselný kód tohoto typu záznamu je 33. Záznam typu SRV má v zónovém souboru pro danou doménu následující formát:

```
_Service._Proto.Name <TTL> <class> SRV Priority Weight Port Target
```

[I] **Broadcasting** = možnost komunikace (oslovení) se všemi zařízeními na síti zaráz. Jedná se o vyslání paketu se speciální broadcastovou adresou, který bude doručen všem zařízením ve stejné síti (v praxi všem zařízením ve stejné broadcastové doméně, tedy segmentu dané sítě obvykle ohraničené routerem). Broadcastová adresa se pro danou síť vytváří tak, že hostitelská část IP adresy [20] určená maskou podsítě je v binární reprezentaci tvořena samými 1 (např. pro síť 192.168.100.0 / 24 je adresa broadcastu 192.168.100.255). Použitím této adresy v hlavičce paketu dojde k tomu, že je rozeslán a zejména zpracován všemi zařízeními v dané síti, tedy i těmi, kterým nemusí být primárně určen. Zahlcení sítě tímto druhem provozu může vést až k tzv. *broadcast storms*, kdy je síť přetížena a drasticky zpomalena, nelze navázat nová spojení a ta stávající trpí buď prodlevami nebo úplnými výpadky. Broadcast jako takový se používá např. když počítač získává svou adresu pomocí DHCP (počítač požádá o přidělení adresy všechna zařízení na síti, ale jen DHCP server mu odpoví) nebo při ARP dotazech (ptá se všech zařízení, jestli neví, jaká MAC adresa patří ke konkrétní IP adrese a síťový prvek s danou IP adresou pak odpoví).

Použití podtržítek „_“ není pouze kosmetickou záležitostí, je to proto, aby se jednoznačně předešlo případným kolizím v reprezentaci s jinými RR záznamy, a to proměnnou `<owner>`. Proměnné `<TTL>` a `<class>` jsou definovány v RFC1034 [Rfc1] a RFC1035 [Rfc2] v rámci základů služby DNS a jejich použití je pro záznam typu SRV úplně stejné.

Proměnná `Service` určuje poskytovanou službu [8] (např. LDAP, POP, SMTP, aj.), může být vyjádřena buď slovně (pozor je citlivé na velikost písmen) či číslem. Ať už se zvolí mnemonický nebo numerický zápis, měl by být v souladu s definovaným standardizovaným výčtem služeb uváděných v RFC1700 [Rfc3]. Proměnná (necitlivá na velikost písmen) `Proto` definuje protokol, nad kterým služba funguje, standardně se užívá transportních protokolů TCP či UDP. Proměnná `Name` označuje doménu, ke které se SRV záznam vztahuje.

`Priority` určuje prioritu, se kterou bude kontaktováno zařízení s tímto SRV záznamem. Klient se vždy pokouší přistupovat k zařízení s nejnižší možnou prioritou, má-li na výběr z více možností. Prioritní číslo je kódováno na 16 bitech, tím pádem rozsah možných hodnot je 0 až 65536. Pokud mají dva a více záznamů stejnou prioritu, rozhoduje číslo uvedené v proměnné `Weight`. Toto 2 B číslo může být opět v rozsahu 0 až 65536. Více o způsobu výběru koncového zařízení a rozkládání vytíženosti poskytne další podkapitola.

Proměnná `Port` určuje číslo portu na koncovém zařízení, na kterém je specifikovaná služba k dispozici. Rozsah této proměnné je 0 až 65536, více o rozdělení portového prostoru na privátní, registrované a dobře známé porty je k dispozici např. na [14]. Proměnná `Target` na závěr určuje doménové jméno koncového zařízení ve smyslu jež definuje RFC1034 [Rfc1], avšak nelze použít alias, tedy jméno definované pomocí záznamu CNAME! Pokud je obsahem této proměnné tečka „ . “, znamená to, že má být daná služba z rozhodnutí vědomě pro tuto doménu nedostupná.

3.4 Rozložení zátěže a záložní zařízení

Pomocí proměnných `Priority` a `Weight` se dá celkem snadno docílit rozložení zátěže v provozování stejné služby mezi více koncových zařízení, která ji poskytují. Alternativně lze vytvářet i záložní zařízení v případě, že by došlo k neočekávaným výpadkům v poskytování služeb.

Jak již bylo napsáno, pokud DNS dotaz na konkrétní službu vrátí více SRV záznamů, musí klient přistupovat ke koncovému zařízení, která má nejnižší možnou prioritu. V případě, že je toto zařízení nedostupné, využije informace v následujícím záznamu s druhou nejnižší prioritou, a tak dále dokud nevyčerpá všechny možnosti.

Pokud je výsledkem DNS dotazu více SRV záznamů se stejnou hodnotou priority, pak rozhoduje číslo uvedené v proměnné `Weight`. Ke kterému ze zařízení se stejnou prioritou se přistoupí se rozhoduje podle následujícího doporučeného algoritmu, pro lepší abstrakci odkazuji čtenáře na níže uvedený praktický příklad a zejména obrázek [Obr.3]:

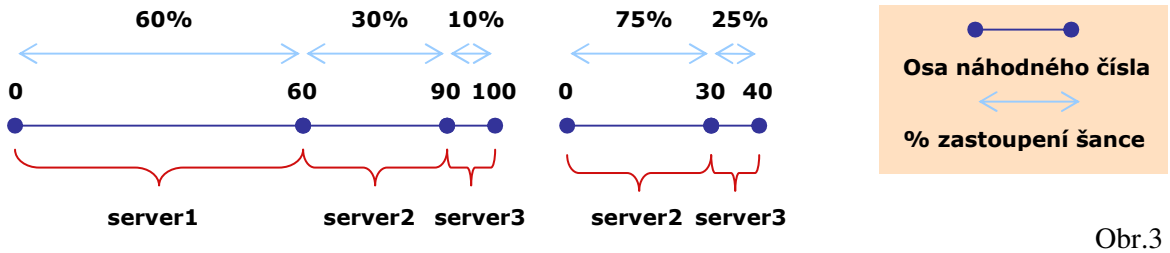
1. W je množina všech proměnných `Weight` těch použitelných SRV záznamů, jež mají stejnou prioritou
2. $k = |W|$ a pak spočítáme $s = \sum_{i=1}^{k} W_i$
3. Z množiny $\langle 0; s \rangle$ přidělíme každému prvku množiny W popořadě postupně spojitě číselný prostor tak, aby platilo $s = \frac{W_1}{s} + \frac{W_2}{s} + \ldots + \frac{W_k}{s}$
4. Generujeme náhodné číslo X, tak aby platilo $X \in \langle 0; s \rangle$
5. Zvolíme ten SRV záznam, kterému přináleží daný prvek množiny W, do jehož číselného prostoru patří náhodné číslo X.

3.5 Typický příklad

Mějme tedy následující platný výňatek SRV záznamů ze smyšleného zónového souboru:

```
_sip._tcp.pokus.com. 86400 IN SRV 10 60 5060 server1.pokus.com.
_sip._tcp.pokus.com. 86400 IN SRV 10 30 5060 server2.pokus.com.
_sip._tcp.pokus.com. 86400 IN SRV 10 10 5060 server3.pokus.com.
_sip._tcp.pokus.com. 86400 IN SRV 20 0  5060 zalozni.pokus.com.
```

Z extraktu je vidět, že doména `pokus.com` poskytuje službu SIP nad transportním protokolem TCP, a to hned na čtyřech koncových zařízeních. První tři záznamy sdílejí stejnou prioritu. Ze součtu `Weight` vidíme, že při rozhodování, který server nakonec použijeme, budeme náhodně generovat číslo z rozsahu 0 až 100. V poměru to znamená, že ze 60% ($X \in \langle 0; 60 \rangle$) bude zvolen klientem pro službu SIP server s názvem `server1.pokus.com`, ze 30% ($X \in \langle 60; 90 \rangle$) `server2.pokus.com` a z 10% ($X \in \langle 90; 100 \rangle$) `server3.pokus.com`. V případě výpadku `server1.pokus.com`, bude mezi zbývající dva servery distribuována zátěž v poměru 75% ku 25%. Pokud by došlo k výpadku nebo jiné nedostupnosti všech hlavních serverů, bude zvoleno další následující zařízení s nejnižší možnou prioritou, a to server `zalozni.pokus.com`. První dva případy názorně ilustruje obrázek [Obr.3]:



Obr.3

3.6 Užití v praxi

Nesporně užitečnou věcí při používání záznamů SRV je možnost přidělit známé službě jakékoli číslo portu, na kterém je dostupná. Tedy i jiné, než je obvyklé v návaznosti na seznam [14] dobře známých (*well-known*) portů. Radikálně se tím dá zvýšit bezpečnost serveru v rámci útoků a scanování otevřených portů. Pro klienta je situace snazší, protože pokud v rámci nějaké aplikace zapomene nastavení pro komunikaci se vzdáleným serverem, díky DNS dotazu ji snadno získá.

SRV záznamy se v současnosti používají jako zavedená praxe pro aplikační protokoly jako XMPP, SIP nebo LDAP. V budoucnu se vidí jako výhoda užití tohoto záznamu pro služby VoIP.

Od serverové verze Windows 2000 a výše Microsoft často používá zónové soubory s DNS záznamy typu SRV jako adresovatele pro řadič domény [13].

Jako zavedená metodologie se v praxi užívá následující číslování proměnné `Weight`. Součet všech těchto proměnných pro záznamy se stejnou prioritou by měl být buď 100 nebo 1000. Je to z toho důvodu, aby se pro lidi transparentněji prováděly výpočty ohledně využití vyvažování jednotlivých serverů v případech údržby zónového souboru a nebo při hledání problému. V případě součtu 100 udává každý jednotlivý `Weigth` údaj při využití každého záznamu se stejnou `Priority` vytížení daného zařízení v %, pro součet 1000 pak v ‰.

4 Záznam LOC

4.1 Stručný popis

Speciální druh záznamu LOC definovaný v RFC1876 [Rfc4] do služby DNS implementuje způsob zaznamenání geografické polohy daného serveru/služby.

4.2 Princip

Způsob zaznamenání informace se odkazuje na zavedený a revidovaný standard určení polohy WGS84 [15] vypracovaný a podporovaný Ministerstvem obrany USA. Tento způsob umožňuje přesně určit polohu daného místa při znalosti geografické šířky, délky a výšky s možností definice sférické velikosti samotné hledané entity s horizontální a vertikální přesností.

4.3 Syntaxe a sémantika

Jelikož se jedná o další RR záznam platí pro něj všechna pravidla a omezení jako pro jiné typy záznamů (např. A, AAAA, PTR, CNAME, MX). Číselná reprezentace tohoto druhu záznamu je 29. Záznam typu LOC má v zónovém souboru pro danou doménu následující implicitní tvar:

```
<owner> <TTL> <class> LOC (d1 [m1 [s1]] {"N"|"S"}
                           d2 [m2 [s2]] {"E"|"W"}
                           alt["m"] [siz["m"] [hp["m"]  [vp["m"]]]])
```

Proměnné `<owner>`, `<TTL>` a `<class>` jsou podrobněji vysvětleny v RFC1034 [Rfc1] a RFC1035 [Rfc2], jedná se o funkční záležitosti služby DNS, pro popis záznamu LOC jsou nepodstatné. Ze schéma je patrné, že výskyt proměnných `d*` a `alt` je v záznamu LOC povinný, zatímco proměnné `m*`, `s*`, `siz`, `hp` a `vp` jsou uváděny volitelně.

Pokud nejsou proměnné minuty a vteřiny uvedeny je jejich implicitní hodnota chápána jako nula. Implicitní hodnota proměnné `siz` tedy velikosti entity je 1 metr. Jinak je to u přesností, implicitní hodnotou horizontální přesnosti je 10 000 metrů, vertikální pak 10 metrů, protože standardně reprezentují v USA velmi zhruba velikost oblasti, které přináleží vlastní ZIP/PSČ číslo.

Údaje o geografické šířce a délce je možné uvádět postupně na stupně, minuty a vteřiny (tyto pak s přesností až na tisíciny). Nadmořská výška `alt` (při užití záporného znaménka mínus „ – “ se jedná o hloubku) je uváděna v metrech s přesností na setiny. Velikostí `siz` je chápána myšlená koule o daném průměru v metrech obklopující lokalizovaný objekt. Horizontální a vertikální přesnosti se

udávají v metrech a dokreslují myšlenou představu o rozměrech entity. Podrobněji shrnuje následující tabulka:

Identifikátor	Obor hodnot	Význam
d1	<0; 90>	stupně šířky
d2	<0; 180>	stupně délky
m1	<0; 59>	minuty šířky
m2	<0; 59>	minuty délky
s1	<0; 59.999>	vteřiny šířky
s2	<0; 59.999>	vteřiny délky
alt	<-100 000.00; 42 849 672.95>	metrů výšky/hloubky
siz	<0; 90 000 000>	metrů velikosti
hp	<0; 90 000 000>	metrů horiz. přesnosti
vp	<0; 90 000 000>	metrů vert. přesnosti

4.4 Typický příklad

Následující příklad ukazuje výpis záznamu LOC ze zónového souboru domény `yahoo.com`, který určuje svoji polohu jako 37° 23’ 30.9‘‘ severní šířky, 121° 59‘ 19‘‘ západní délky, 7 metrů nad úrovní moře jakožto středu myšlené koule o poloměru 100 metrů s průměrem horizontální kruhové chyby 10 metrů a s průměrem vertikální kruhové chyby 2 metry:

```
yahoo.com. 7 IN LOC 37 23 30.900 N 121 59 19.000 W 7.00m 100m 10m 2m
```

4.5 Užití v praxi

Ke zdrojům na Internetu se v současnosti ve většině případů přistupuje nezávisle na tom, je-li vzdálenost od klienta k cílovému serveru/službě pár metrů (typicky zařízení na stejné síti či jejím segmentu) či tisíce kilometrů. Naneštěstí geografická poloha mezi klientem a koncovým uzlem hraje roli např. při prodlevě či celkové (mnohdy zbytečné) vytíženosti šířky pásma. A právě toto bylo jedním z důvodů k zavedení RR záznamu typu LOC a jeho integrace do služby DNS.

Hierarchická distribuovaná struktura DNS umožňuje každé přidružené organizaci samostatně spravovat geografické informace a zároveň je tak jednoduše šířit do celého světa bez nutnosti vytvářet nějakou centrální databázi zabývající se schraňováním těchto dat. V rámci autonomní oblasti umožňuje jejím správcům určovat, které její části stojí za to být geograficky lokalizovány. V případě místopisných přesunů se změna těchto informací omezuje na jednoduché přepsání dat v zónovém souboru.

Naneštěstí jsou tyto iniciativy teprve v plenkách [3]. A to i díky bezpečnostním rizikům, neb přesná znalost geografické polohy díky datům v záznamu LOC nahrává k plánování fyzických útoků (podvratná ekonomicko-hospodářská činnost, teroristický útok, válečný konflikt) vedoucích k možným selháním poskytovaných mnohdy klíčových služeb. Můžeme však předpokládat, že se tato

situace bude měnit a nad otázkou bezpečnosti převládne snaha o optimálnější přístup služeb koncovým uživatelům. Některé z větších domén, které zveřejňují v rámci svých zónových souborů záznamy LOC jsou uvedeny v Příloze [Priloha13.3] k této bakalářské práci.

5 Load sharing NAT

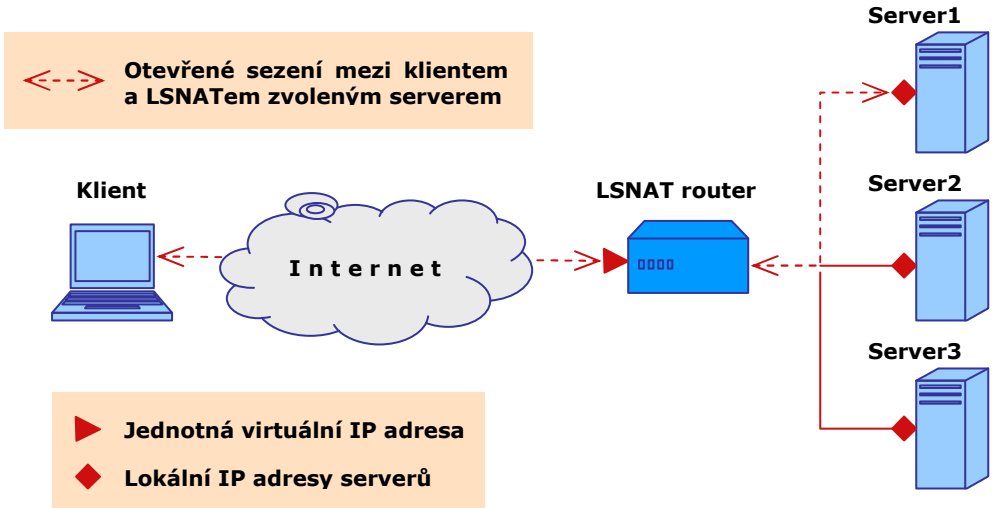
5.1 Stručný popis

S masivním rozvojem Internetu na konci 90. let 20. století se začaly objevovat první problémy s jeho velikostí spojené – zácpy (*bottlenecks*) na různých segmentech sítě, přetížené nebo úplně spadlé servery díky zahlcení (*congestions*). Začalo se tedy přemýšlet nad různými optimalizačními technikami ohledně rychlého, bezpečného a spolehlivého spojení ke službám. A právě jednou z těchto iniciativ je i v srpnu 1998 vzniklý LSNAT, později standardizovaná a definovaná v RFC2391 [Rfc6].

5.2 Princip

Základní idea je následující – servery poskytující stejné služby jsou sdruženy do serverových farem pod stejnou administrativní správou. S Internetem jsou propojeni pomocí specializovaného směrovače (*routeru*). Ten jim přiděluje **lokální IP** adresy a porty, přičemž vnější klienti přistupují na **virtuální IP** adresu LSNAT routeru. Směrovač pak na základě výsledků použitého sofistikovaného zátěž distribuujícího algoritmu (Round Robin, aj.) spojuje klienty s jednotlivými servery.

Tento způsob spojení s Internetem má výhodu v tom, že příchozí klienti si jsou vědomi jen virtuální IP adresy užívané LSNAT routerem a nevidí IP adresy jednotlivých serverů, což poskytuje aspoň základní způsob zabezpečení. Díky užitému vyvažujícími algoritmu je docíleno i částečné spolehlivosti k přistupované službě (když jeden server trpí výpadkem, LSNAT router jednoduše přeposílá jeho klienty jinému serveru ze serverové farmy poskytujícímu stejnou službu) a celkovému zlepšení síťového provozu. Následující obrázek [Obr.4] schématicky naznačuje princip fungování LSNATu, kdy klient vzdáleně přistupuje k virtuální IP adrese služby a LSNAT směrovačem je spojen s nejoptimálnějším serverem, který ho obslouží:



Obr.4

5.3 Vysvětlení činnosti

LSNAT umožňuje jedné službě dostupné na klienty předpokládané IP adrese a portu namapovat více zařízení. Z původní IP adresy a portu se stává tzv. virtuální IP adresa, přičemž v případě, že se na LSNAT routeru vyskytne tato adresa jako cílová, tak ji router přeloží na lokální existující IP adresu serveru podle zvoleného zátěž distribuujícího algoritmu. V případě odpovědi klientovi pak modifikuje původní reálnou IP adresu serveru na virtuální IP adresu. LSNAT router vlastně provádí úpravu příchozích a ochozích paketů.

Problémem, kterým je nutné se zabývat v rámci distribuce zátěže LSNAT routeru je způsob, jakým se tento směrovač dozví, že server, na který odkazuje klienty přistupující na virtuální IP adresu, je nedostupný a tím pádem ho má vyloučit z procesu rozdělování. Jednou z možností je např. vysílání dotazu pomocí specializovaného programu PING [16], ovšem to zbytečně zatěžuje provoz na síti, což je vždy nežádoucí. Místo toho se ve většině případů používá analyzování klient-server sezení spojeného pomocí LSNAT. Pokud ze strany serveru delší dobu nepřišla klientovi odpověď, je tato linka označena za „mrtvou“ a LSNAT přestává spojovat příchozí klienty na adresu pravděpodobně vypadlého, zahlceného nebo s technickými problémy se potýkajícího serveru.

LSNAT překlad adres se rozděluje do tří fází:

1. **Session binding** – v této fázi je server vybrán z množiny dané serverové farmy a je asociován s příchozím sezením (tento výběr může být buď statický nebo dynamický podle metrik dodaných během inicializace sezení zátěž vyvažujícím algoritmem);
2. **Address lookup and translation** – pokud došlo k provázání příchozího sezení s konkrétním serverem modifikuje zařízení (typicky router), na kterém běží LSNAT, políčka zdrojových/cílových IP adres v hlavičkách průchozích paketů, zároveň s tím přepočítává kontrolní součty pro tato políčka i pro samotné datagramy. Stejně zachází i s obdobnými políčky kontrolních ICMP paketů;
3. **Session unbinding** – fáze, ve které zaniká sezení mezi serverem a klientem, dochází k rozpojení a především k vyvázání mapování virtuální IP adresy na LSNATem přidělený server.

5.4 Vyvažovací algoritmy

Pro zvolení ideálního serveru pro obsloužení klienta je nanejvýš vhodné znát co nejpřesněji následující základní údaje:

a) **Cenu** přístupu do segmentu sítě, ve které se server nachází;
b) **Vytížení** rozhraní, které se používá k přístupu k serveru;

c) **Dostupnost zdrojů** (myšlena zaneprázdněnost CPU, využití paměti, atd.) na samotném serveru;

V zásadě se dají algoritmy rozdělit do dvou hlavních skupin, podle způsobu funkčnosti. A to algoritmy **pasivní** a **aktivní**.

První jmenované zjišťují vytíženost cílových serverů na základě analýzy provozu na síti, ty druhé se aktivně serverů dotazují na jejich aktuální stav. Výhodou pasivního přístupu je, že nijak nezatěžuje servery, je nenáročný na provoz z pohledu HW nároků, avšak může poskytovat špatné metriky – je závislý na informacích „z druhé ruky“. Nevýhodou aktivního přístupu je fakt, že pravidelné se vyptávání serverů je přídavnou režií k již tak mnohdy vysokému zatížení provozem, o který se servery starají; na druhou stranu ale poskytuje nejpřesnější možné informace o statutu koncových zařízení.

5.5 Stručný popis užívaných algoritmů

5.5.1 Ping individual server

V pravidelných intervalech LSNAT router vysílá dotazy pomocí programu PING [16] na servery a příchozí klienty směruje vždy na server s nejvíce úspěšnými odpověďmi nebo nejkratší odpovědní dobou na ping.

5.5.2 Least weighted load first

Administrátorovi umožňuje přidělit váhový koeficient, a to:

a) Jednotlivým sezením na základě množství prostředků, které spotřebují;
b) Jednotlivým serverům na základě dostupností jejich systémových zdrojů;

Přičemž pro spojení je vybrána vždy ta možnost, která má nejnižší možný váhový koeficient.

5.5.3 Weighted least load first

Oproti předchozímu algoritmu se k metrice přičítá i možná nestejná cena přístupu k jednotlivým serverům poskytujícím totožnou službu.

5.5.4 Weighted least traffic first

Po určitou testovací dobu je pozorován síťový provoz mezi konkrétním serverem a jeho klienty. Obvykle se počítá počet protečených byte a nebo paketů oběma směry. Ve výsledku se

do této metriky ještě začlení cena přístupu k serveru a algoritmus pak směruje příchozí klienty na „nejlevnější nejméně síťovým provozem vytížené servery“.

5.5.5 Round Robin

Tomuto algoritmu je věnována celá následující kapitola, takže čtenáře odkazuji na ni.

5.6 Užití v praxi

Technologie LSNAT se využívá čím dál častěji, a to díky specifickým pozitivům, která poskytuje. Při procesu instalace není nutné měnit žádnou konfiguraci pro klienty i servery.

Jednoduchá údržba v podobě centrálního zařízení LSNAT routeru umožňuje snadno přidávat/odebírat nové servery. Na druhou stranu právě LSNAT router se může stát tzv. *single point of failure*, tedy místem, kde při selhání může dojít k zhroucení celého systému. Tím pádem je vhodné se spolu s implementací LSNAT zamýšlet i nad záložními variantami v případě výpadku. Jednoznačnou výhodou je možnost filtrování provozu na LSNAT směrovačích pomocí přístupových pravidel (*access lists*). Již zmíněnou výhodou je i relativní bezpečnost, neboť přistupujícím klientům není známa koncová IP adresa serveru, který je obsluhuje. Všichni přistupují jednotně k virtuální IP adrese služby, což zabraňuje možným přímým útokům na konkrétní servery.

Ovšem je nutné mít na zřeteli i některá omezení! LSNAT porušuje TCP/IP pragma přímé spojenosti koncových účastníků (*end-to-end connection*), což může vést k problémům s distribucí šifrovacích klíčů (DNSSec). Také některé služby, které spoléhají na přenos IP adres i mimo hlavičky paketů (např. SNMP), nemusí korektně fungovat.

6 Round Robin

6.1 Stručný popis

Je technika [4] implementována většinou dnes hojně užívaných DNS serverů (BIND, MS DNS, IP Control, aj.), která samotnému jmennému serveru umožňuje řídit rozložení zátěže v rámci dané (obvykle topologicky rozsáhlé) domény, aby nedocházelo k zbytečnému přetěžování jednoho serveru.

6.2 Princip

Na klasický dotaz klienta k přeložení doménového jména na IP adresu obvykle odpovídá seznamem IP adres koncových zařízení, která všechna zprostředkovávají danou službu. Tento seznam však není v čase stále stejný. Algoritmus se stará o to, aby docházelo k pravidelnému cyklování IP adres v navraceném seznamu. Jednoduše první vracená IP adresa se po určité době posune na úplný konec seznamu s tím, že druhá IP adresa v původním pořadí zabírá její místo – popřípadě opačný směr, první IP adresa propadává seznamem směrem dolů a její místo nahrazuje IP adresa na původně posledním místě.

Klient obvykle vybere první IP adresu ze seznamu a zbytek zahodí, avšak v případě nedostupnosti první IP adresy může pokračovat dalšími položkami seznamu. Některé sofistikované úpravy tohoto druhu algoritmu dokonce navracejí seznam upravený a setřízený podle určitého kritéria, například geografické vzdálenosti koncových serverů od klienta – viz. GeoDNS.

6.3 Typický příklad

Tento demonstruje dva časově po sobě následující výňatky zónového souboru domény `google.com`, v současnosti nejvytěžovanějšího internetového vyhledávače:

```
google.com.          221294     IN    NS    ns1.google.com.
google.com.          221294     IN    NS    ns2.google.com.
google.com.          221294     IN    NS    ns3.google.com.
google.com.          221294     IN    NS    ns4.google.com.
```

A po sekundě získaný záznam ukazuje rotování Round Robin směrem dolů:

```
google.com.          221293     IN    NS    ns4.google.com.
google.com.          221293     IN    NS    ns1.google.com.
google.com.          221293     IN    NS    ns2.google.com.
google.com.          221293     IN    NS    ns3.google.com.
```

6.4 Užití v praxi

Algoritmus Round Robin se užívá k vyvažování provozu u rozsáhlých domén, aby nedocházelo k zbytečnému zatěžování jediného serveru a zpomalování jím poskytovaných služeb. Je však důležité si uvědomit, že se jedná o rozkládání zátěže založené na statistickém přístupu – jednoduše rovnoměrně rozděluje seznam IP adres v závislosti na příchozích DNS dotazech. Avšak toto rozdělení nemusí vůbec brát do úvahy stav koncových zařízení – jejich vytížení, možnou dobu zpracovávání jednotlivých klientů, zahlcení daného segmentu sítě a jiné. Je proto doporučené Round Robin používat jen pro servery se stejnou HW konfigurací, které všechny zprostředkovávají jednu a tutéž službu. Ovšem samotný algoritmus může být vhodně upraven a provázán s různými skripty tak, aby se dalo předejít výše zmíněnému nekorektnímu chování. Jedno z možných řešení nabízí iniciativa GeoDNS blíže popsaná níže.

7 BIND

7.1 Stručný popis

Je v současnosti snad nevíc rozšířenou verzí DNS serveru. Za dobu svého vývoje se stal na tomto poli uznávaným standardem a je podporován Internet Systems Consortium [17]. Aktuální verze tohoto produktu je verze BIND 9.4.1 [18] a vznikla za spolupráce výrobců unixových systémů a Ministerstva obrany USA.

Nynější verze podporuje celou řadu nových technologií a zabezpečení – směrování IPv6, podpisy transakcí TSIG, šifrované přenosy DNSSEC, podporu pro multiprocesorový běh, aj. Jako balíček v sobě obsahuje tři základní aplikace pro plnohodnotnou funkci, a to DNS server, DNS resolver library a nástroje pro ladění a ověřování funkčnosti.

Jedná se pouze o jednu z možných aplikací (výčet ostatních hráčů na tomto poli je uveden v Referencích na [7]), které jsou na Internetu k dispozici, ale jak již bylo řečeno na začátku, je všeobecně uznáván jako standard, a to i přes velkou kritiku, které sklízely předchozí verze. Důvod, proč je naznačena jeho existence v této práci, je iniciativa GeoDNS na BINDu vystavěná, které je věnována následující kapitola.

7.2 Relevantní historie verzí

V případě užití záznamu SRV v zónovém souboru je tento typ podporován od verze BIND 8.2.2 a vyšší. Co se záznamu LOC týče, je podporován od verze BIND 4.9.7. Pokud se zaměříme na distribuci zátěže, pak první implementace algoritmů jako Round Robin se datuje k verzi BIND 4.8.3.

8 GeoDNS

8.1 Stručný popis

Iniciativa vzniklá [5] jako doplňující patch k aplikaci BIND 9.2 a jejím vyšším verzím. Zabývá se možnostmi směrování v návaznosti na geografickou polohu klienta.

8.2 Princip

Základní idea vychází z následují úvahy: Mějme doménu se dvěma web servery s úplně totožným obsahem, přičemž jeden server leží v USA a druhý ve Velké Británii. Je tedy logické směrovat návštěvníky z celé Ameriky na server v USA a návštěvníky z Evropy a přilehlého okolí na britský server.

BIND 9.2 a vyšší umožňuje několik různých pohledů na jednu a tu samou doménu, což znamená, že odpovědi na identické DNS dotazy se můžou lišit v závislosti na klientech, kteří se ptají. Avšak BIND jako takový dokáže klienty rozlišovat jen podle IP adres či síťových částí těchto adres.

GeoDNS propojuje rozsáhlou komerční databázi všech známých IP adres sítí GeoIP [19] s filtračními schopnostmi BINDu. GeoDNS je na základě licence GNU volně k dispozici v podobě čtyřiceti řádkového patche, přičemž odkaz na něj je uveden v Referencích [6].

8.3 Užití v praxi

Iniciativa GeoDNS je první vlaštovkou v možnostech užívání služby DNS i jako poskytovatele informace o lokalitě serveru a služby. Její zjevnou nevýhodou je, že ke své funkci využívá komerční databázi GeoIP, což znamená přídavné náklady v případě nasazení do provozu.

9 PlanetLab

9.1 Stručný popis

Jedná se o samostatnou globální počítačovou [9] síť založenou v roce 2002, a to za účelem poskytnutí testovací laboratoře pro studium síťování počítačů a výzkum distribuovaných systémů a služeb. Vývojářům umožňuje vytvářet především vlastní nezávislé aplikace, které běží v rámci celé sítě na každém uzlu na oddělených virtuálních vrstvách. Lze vytvářet úplně vlastní struktury uzlů, přičemž lze použít své způsoby směrování toku a adresace objektů, služeb či uzlů.

Do PlanetLabu je v současnosti zapojena špička výzkumných akademických center i nejprestižnějších firem (Intel, HP, Google, Wikipedia) pracujících v odvětví informačních technologií. V průběhu vytváření této práce je tato iniciativa tvořena 782 uzly (připojenými zařízeními) na 382 místech po celém světě. Od roku 2003 PlanetLab posloužil jako základna pro mnoho nezávislých i komerčních projektů, které se plánují v budoucnu nasadit do Internetu, nebo už byly publikovány.

10 DNSProxy

10.1 Stručný popis

Jde o iniciativu [10] v rámci PlanetLabu [9], která se zabývá nadstandardními postupy ve službě DNS, co se týče hledání nejvhodnějších kandidátů v rámci odpovědí na DNS dotazy ohledně služeb či překladu adres.

10.2 Princip

Pokud klient vstupuje do konkrétní vrstvy (termín „vrstva“ je v rámci PlanetLabu srovnatelný s pojmem „doména“ v Internetu), musí nejdříve najít IP adresu příslušného vstupního uzlu. Přičemž tato iniciativa si klade za cíl, aby pro navracený vstupní uzel platilo, že je aktivní (ve smyslu, že poskytuje danou službu a nepotýká se s žádnými technickými výpadky či nedostatky) a je topologicky nejbližší klientově uzlu.

10.3 Vysvětlení činnosti

DNS dynamicky udržuje provázanou tabulku služeb a adres vrstvových uzlů, které tyto služby poskytují. Při DNS dotazu pro vstup do dané vrstvy vytvoří server setříděný seznam pěti nejbližších vstupních uzlů a klientovi navrátí první záznam seznamu. To že se jedná o topologicky nejbližší uzly server zajistí tak, že všechny nalezené záznamy nejprve seřadí podle počtu *hopů* (mírou 1 hopu je cesta mezi dvěma bezprostředními síťovými zařízeními, která jsou vzájemně přímo propojena) od klienta k cílovému vstupnímu uzlu. Pokud má dva nebo více záznamů z potencionální pětky stejný počet hopů, je navracená adresa vstupního uzlu vybrána náhodně, a to z důvodu vyvažování zátěže pro danou službu.

10.4 Užití v praxi

Tato iniciativa je teprve ve stádiu vývoje, avšak již nyní ukazuje na zajímavý způsob zjišťování polohy klienta pomocí hopů. Zázemí PlanetLabu ji však dává velmi slušnou šanci k tomu, že dříve nebo později bude uvedena v ostrý provoz v rámci Internetu.

11 Závěr

Tato práce shrnula možnosti rozšíření služby DNS o informaci o poloze služby či samotného serveru a uvedla jejich příklady využití v již existujících projektech či iniciativách. Zároveň s tím seznámila čtenáře i se způsoby distribuce zátěže provozu pomocí různých algoritmů i jejich implementaci buď ve službě DNS a nebo za použití LSNAT. Co se dosažení předsevzatých cílů týká, je tato práce uceleným materiálem poskytující informace, jak k nastavení parametrů zónových souborů, tak k užití specializovaných typů záznamů a k pochopení principů funkčnosti.

Optimalizační snahy, zmiňované v úvodní kapitole, při přístupu ke službě už se tedy začínají řešit a mnohé z přijatých standardů jsou již užívány v praxi. V budoucnu můžeme předpokládat, že tyto snahy převáží a metrika nejoptimálnějšího přístupu ke službě se stane jednou z nejdůležitějších při rozhodování o směrování klientů. Další z možností pro rozvoj je např. nasazení rezidentních programů na klíčových DNS serverech, které by kontrolovaly parametry připojené sítě (dostupnost, prodleva, vytížení) a poskytovaly komplexní vědomosti jako metriky ostatním aplikacím či protokolům.

Iniciativa GeoDNS jednoznačně dominuje, co se týče směrování na základě geografické informace. Ovšem slabinou tohoto projektu je jeho provázanost s GeoIP, tedy s komerčně prodávanou databází, což brzdí jeho masivnější rozšíření.

Námět na další rozvíjení této problematiky vidím např. v komplexním zpracování tabulky volně dostupných adres IP a k nim přiřazených obecných zeměpisných informací (země původu či ještě lépe město), které se takto stanou lehce dostupné pro aplikace. Též by bylo možné sestavit program mapující IP adresní a doménový prostor (vytahující nejen záznamy SRV či LOC) a navazující takto získané poznatky na geografické rozložení.

12 Reference

12.1 Obecné informace

Všechny níže uvedené URL odkazy jsou platné ke dni 15. května 2007. Autor bakalářské práce nevylučuje, že v průběhu času mohou být tyto odkazy nedostupné či informace, které poskytují, zastarají či ztratí na významu nebo odborné úrovni.

12.2 Literatura

[Rfc1] Mockapetris, P.: **Domain Names - Concepts and facilities**, IETF, listopad 1987. Dokument dostupný na URL: http://tools.ietf.org/html/rfc1034

[Rfc2] Mockapetris, P.: **Domain Names - Implementation and specification**, IETF, listopad 1987. Dokument dostupný na URL: http://tools.ietf.org/html/rfc1035

[Rfc3] Reynolds, J., Postel, J.: **Assigned Numbers**, IETF, říjen 1994. Dokument dostupný na URL: http://tools.ietf.org/html/rfc1700

[Rfc4] Davis, C., aj.: **A Means for Expressing Location Information in the DNS**, IETF, leden 1996. Dokument dostupný na URL: http://tools.ietf.org/html/rfc1876

[Rfc5] Gulbrandsen, A., Vixie, P., Esibov, L.: **A DNS RR for specifying the location of services DNS SRV**, IETF, únor 2000. Dokument dostupný na URL: http://tools.ietf.org/html/rfc2782

[Rfc6] Strisuresh, P.: **Load Sharing using IP Network Address Translation – LSNAT**, IETF, srpen 1998. Dokument dostupný na URL: http://tools.ietf.org/html/rfc2391

[Rfc7] Bush, R., aj.: **Root Name Server Operational Requirements**, IETF, červen 2000. Dokument dostupný na URL: http://tools.ietf.org/html/rfc2870

12.3 Elektronické prameny

[1] **SRV record**, Wikipedia, revize 30. března 2007. URL: http://en.wikipedia.org/wiki/SRV_record

[2] **LOC record**, Wikipedia, revize 30. března 2007. URL: http://en.wikipedia.org/wiki/LOC_record

[3] **CKDHR.com**, revize 28. listopadu 2005. URL: http://www.ckdhr.com/

[4] **Round robin DNS**, Wikipedia, revize 30. března 2007.
URL: http://en.wikipedia.org/wiki/Round_robin_DNS

[5] **GeoDNS Bind patch**, Caraytech, 9. března 2004. URL: http://www.caraytech.com/geodns/

[6] **Patch diff**, Caraytech, 9. března 2004. URL: http://www.caraytech.com/geodns/patch.diff

[7] **Comparison of DNS server software**, Wikipedia, revize 27. března 2007.
URL: http://en.wikipedia.org/wiki/Comparison_of_DNS_server_software

[8] **DNS SRV (RFC 2782) Service Types**, DNS Service Discovery.
URL: http://www.dns-sd.org/ServiceTypes.html

[9] **PlanetLab**, revize 31. ledna 2007. URL: http://www.planet-lab.org/

[10] **DNSProxy**, PlanetLab, revize 26. července 2005.
URL: https://wiki.planet-lab.org/twiki/bin/view/Planetlab/DNSProxy?topic=DNSProxy

[11] **Load Sharing using IP Address Translation**, Lucy O'Sullivan, NTRG.
URL: http://ntrg.cs.tcd.ie/undergrad/4ba2.01/group8/LSNAT.html

[12] **Domain Name System**, Wikipedia, revize 1. dubna 2007.
URL: http://cs.wikipedia.org/wiki/Domain_Name_System

[13] **How DNS works**, Microsoft, 28. března 2003.
URL: http://technet2.microsoft.com/WindowsServer/en/library/19a63021-cc53-4ded-a7a3-abaf82e7fb7c1033.mspx?mfr=true

[14] **PORT NUMBERS**, IANA, revize 20. dubna 2007.
URL: http://www.iana.org/assignments/port-numbers

[15] **World Geodetic System**, Wikipedia, revize 6. dubna 2007.
URL: http://en.wikipedia.org/wiki/World_Geodetic_System

[16] **ping**, Wikipedia, 24. dubna 2007. URL: http://en.wikipedia.org/wiki/Ping

[17] **Internet System Consortium**. URL: http://www.isc.org/index.pl

[18] **ISC BIND**, ISC Inc., revize 2004. URL: http://www.isc.org/index.pl?/sw/bind/

[19] **GeoIP – IP address location information**, MaxMind LLC, revize 2007.
URL: http://www.maxmind.com/app/ip-location

[20] **IPv4 addressing**, Wikipedia, revize 11. května 2007.
URL: http://en.wikipedia.org/wiki/IPv4#Addressing

13 Seznam příloh

13.1 Další informace o root serverech

Internetová stránka schraňující podstatná data o všech kořenových serverech po celém světě je www.root-servers.org. Některé ze serverů mají lokální uzly, které jsou blíže koncovým klientům a jejich redundance pod stejnou IP adresou slouží zároveň jako bezpečností opatření při výpadku. V případě nedostupnosti lokálního uzlu přistoupí klient ke globálnímu uzlu pro danou serverovou administrativní rodinu (servery sdílející stejné identifikační písmeno):

Server	Lokální uzel	Země	Podpora	Druh uzlu
F	Auckland	Nový Zéland	IPv4, IPv6	lokální
	Amsterdam	Nizozemí	IPv4, IPv6	lokální
	Barcelona	Španělsko	IPv4, IPv6	lokální
	Brisbane	Austrálie	IPv4	lokální
	Karakas	Venezuela	IPv4	lokální
	Paříž	Francie	IPv4, IPv6	lokální
	Jakarta	Indonésie	IPv4	lokální
	Dhaka	Bangladéš	IPv4	lokální
	Dubaj	UAE	IPv4	lokální
	Buenos Aires	Argentina	IPv4	lokální
	São Paulo	Brazílie	IPv4	lokální
	Hong Kong	Čína	IPv4	lokální
	Johannesburg	Jihoafrická Rep.	IPv4	lokální
	Karáčí	Pákistán	IPv4	lokální
	Osaka	Japonsko	IPv4, IPv6	lokální
	Los Angeles	USA	IPv4, IPv6	lokální
	London	UK	IPv4, IPv6	lokální
	Lisabon	Portugalsko	IPv4, IPv6	lokální
	New York	USA	IPv4, IPv6	lokální
	Chennai	Indie	IPv4	lokální
	Madrid	Španělsko	IPv4	lokální
	Mnichov	Německo	IPv4, IPv6	lokální
	Nairobi	Keňa	IPv4	lokální
	Chicago	USA	IPv4, IPv6	lokální
	Palo Alto	USA	IPv4, IPv6	globální
	Peking	Čína	IPv4	lokální
	Praha	Česká Rep.	IPv4, IPv6	lokální
	Rome	Itálie	IPv4	lokální
	Santiago de Chile	Chile	IPv4	lokální
	Soul	Jižní Korea	IPv4, IPv6	lokální
	San Francisco	USA	IPv4, IPv6	globální
	Singapure	Singapure	IPv4	lokální
	San Jose	USA	IPv4	lokální
	Moskva	Rusko	IPv4	lokální
	Tel Aviv	Izrael	IPv4	lokální
	Taipei	Taiwan	IPv4	lokální
	Torino	Itálie	IPv4	lokální
	Ottawa	Kanada	IPv4, IPv6	lokální
	Toronto	Kanada	IPv4	lokální

Server	Lokální uzel	Země	Podpora	Druh uzlu
J	Dulles	USA	IPv4	globální
	Miami	USA	IPv4	globální
	Atlanta	USA	IPv4	lokální
	Seattle	USA	IPv4	lokální
	Chicago	USA	IPv4	lokální
	New York	USA	IPv4	lokální
	Los Angeles	USA	IPv4	lokální
	Mountain View	USA	IPv4	lokální
	San Francisco	USA	IPv4	lokální
	Amsterdam	Nizozemí	IPv4	lokální
	London	UK	IPv4	lokální
	Stockholm	Švédsko	IPv4	lokální
	Tokyo	Japonsko	IPv4	lokální
	Soul	Jižní Korea	IPv4	lokální
	Peking	China	IPv4	lokální
	Singapure	Singapure	IPv4	lokální
	Dublin	Irsko	IPv4	lokální
	Kaunsas	Litva	IPv4	lokální
	Nairobi	Keňa	IPv4	lokální
	Montreal	Kanada	IPv4	lokální
	Quebec	Kanada	IPv4	lokální
	Sydney	Austrálie	IPv4	lokální
	Cairo	Egypt	IPv4	lokální
	Varšava	Polsko	IPv4	lokální
	Brasilia	Brazílie	IPv4	lokální
	Sao Paulo	Brazílie	IPv4	lokální
	Sofia	Bulharsko	IPv4	lokální
I	Stockholm	Švédsko	IPv4	globální
	Helsinki	Finsko	IPv4	globální
	Milan	Itálie	IPv4	lokální
	London	UK	IPv4	lokální
	Geneva	Švýcarsko	IPv4	lokální
	Amsterdam	Nizozemí	IPv4	lokální
	Oslo	Norsko	IPv4	lokální
	Bangkok	Thajsko	IPv4	lokální
	Hong Kong	Čína	IPv4	lokální
	Brussels	Dánsko	IPv4	lokální
	Frankfurt	Dánsko	IPv4	lokální
	Ankara	Turecko	IPv4	lokální
	Bucharest	Rumunsko	IPv4	lokální
	Chicago	USA	IPv4	lokální
	Washington DC	USA	IPv4	lokální
	Tokyo	Japan	IPv4	lokální
	Kuala Lumpur	Malajsie	IPv4	lokální
	Palo Alto	USA	IPv4	globální
	Jakarta	Indonésie	IPv4	lokální
	Wellington	Nový Zéland	IPv4	lokální
	Johannesburg	SAR	IPv4	lokální
	Perth	Kanada	IPv4	lokální
	San Francisco	USA	IPv4	lokální
	New York	USA	IPv4	lokální
	Singapure	Singapure	IPv4	lokální

Server	Lokální uzel	Země	Podpora	Druh uzlu
K	Londýn	UK	IPv4, IPv6	globální
	Amsterdam	Nizozemí	IPv4, IPv6	globální
	Tokyo	Japonsko	IPv4, IPv6	globální
	Dillí	Indie	IPv4, IPv6	globální
	Miami	USA	IPv4, IPv6	globální
	Budapešť	Maďarsko	IPv4	lokální
	Milan	Itálie	IPv4	lokální
	Helsinky	Finsko	IPv4, IPv6	lokální
	Reykjavik	Island	IPv4, IPv6	lokální
	Poznaň	Polsko	IPv4	lokální
	Frankfurt	Dánsko	IPv4, IPv6	lokální
	Ženeva	Švýcarsko	IPv4, IPv6	lokální
	Athény	Řecko	IPv4, IPv6	lokální
	Doha	Katar	IPv4	lokální
	Novosibirsk	Rusko	IPv4	lokální
	Abu Dhabi	UAE	IPv4	lokální
	Brisbane	Austrálie	IPv4	lokální
A	Dulles	USA	IPv4	globální
B	Marina Del Rey	USA	IPv4, IPv6	globální
C	Herndon	USA	IPv4	lokální
	Los Angeles	USA	IPv4	globální
	New York	USA	IPv4	lokální
	Chicago	USA	IPv4	lokální
D	College Park	USA	IPv4	globální
E	Mountain View	USA	IPv4	globální
G	Columbus	USA	IPv4	globální
H	Aberdeen	USA	IPv4, IPv6	globální
L	Los Angeles	USA	IPv4	globální
M	Tokyo	Japonsko	IPv4, IPv6	lokální
	Soul	Korea	IPv4	lokální
	Paříž	Francie	IPv4, IPv6	lokální
	San Francisco	USA	IPv4, IPv6	globální

13.2 Ukázkový zónový soubor

```
;---------------------------------------------------------------------
$ttl 3600
@          IN    SOA   server.example.org.   admin.example.org.
                       (
                            20070501   ;serial
                            3600       ;refresh
                            300        ;retry
                            604800     ;expire
                            86400      ;TTL
                       )
           IN    NS          server
           IN    NS          ns1.pokus.cz.

           IN    MX    10    mail
           IN    MX    20    mail.pokus.cz.

www        IN    CNAME       server

server     IN    A           190.168.1.1
server     IN    AAAA        2001::02e0:7dff:fe96:daa1
mail       IN    A           190.168.1.2
mail       IN    AAAA        2001::02e0:7dff:fe96:daa2
ldap       IN    A           190.168.1.3
ldap       IN    AAAA        2001::02e0:7dff:fe96:daa3

server     IN    LOC 37 56 31.900 S 12 29 47.000 E 95.00m 100m 10m 2m

_ldap._tcp.example.org.    IN    SRV   0 5 389    ldap.example.org.

pc1        IN    A           190.168.1.11
pc2        IN    A           190.168.1.12
pc3        IN    A           190.168.1.13
;---------------------------------------------------------------------
```

Rozebereme-li si od začátku ukázkový zónový soubor pro doménu `example.org`. První řádek je direktiva `$ttl`, která se implicitně přidává ke každému záznamu a v sekundách určuje životnost daného záznamu při jeho uložení do vyrovnávací paměti. Další je struktura záznamu SOA (*start of authority*), který je zahajujícím záznamem zónového souboru, přičemž `server.example.org` je primárním jmenným serverem pro doménu a `admin.example.org` je emailem na správce domény (klasický zavináč @ je nahrazen tečkou). Údaje v závorkách po řadě znamenají:

- `serial` – sériové číslo záznamu, v podstatě ID, které je třeba zvětšit při změně zónového souboru, aby se dalo na vědomí sekundárním a dalším DNS serverům (obvykle je toto číslo nějakým způsobem odvozeno od data, kdy byl zónový soubor vytvořen);

- `refresh` – určuje interval, jak často se sekundární server primárního dotazuje na sériové číslo a tedy na novou verzi zónového souboru;
- `retry` – určuje periodu dalšího pokusu sekundárního serveru o navázání kontaktu s primárním v případě, že se s ním prvně nepodařilo spojit;
- `expire` – je doba, po které sekundární servery označí své kopie záznamů za prošlé, pokud se jim nedaří spojit se s primárním serverem;
- `TTL` – implicitní doba platnosti záznamu.

Dva záznamy typu NS následující za SOA určují autoritativní nameservery pro danou doménu, přičemž první z nich `server` je **interní** a ns1.pokus.cz je **externí** (nespadá pod administrativní správu naší domény `example.org`).

Další dva záznamy MX určují poštovní servery pro danou doménu, přičemž se bude preferovat lokální server `mail.example.org`. Následuje záznam typu CNAME, tedy kanonického jména (chcete-li aliasu), který určuje, že doménové jméno `www.example.org` je jen jiným názvem pro počítač `server.example.org`.

Následují tři dvojice záznamů, které slouží k překladu doménových jmen na jim odpovídající IP adresy, a to IPv4 (záznam typu A) a IPv6 (záznam typu AAAA). Takže např. `ldap.example.org` se při dotazu na jeho IP adresu v aplikaci přeloží na 190.168.1.3.

Server má i záznam LOC o své geografické poloze, která je 37° 56’ 31.9‘‘ jižní šířky, 12° 29‘ 27‘‘ východní délky a 95m nad úrovní moře s perimetrem 100m.

Následující záznam SRV nám říká, že v doméně je vědomě podporována služba LDAP, a to na serveru `ldap.example.org` na portu 389.

Zbývající záznamy jsou klasickými A záznamy, které třem doménovým jménům jednotlivých počítačů `pc[1-3].example.org` přiřazují IP adresy.

13.3 Některé domény zveřejňující LOC záznamy

Výčet rozhodně není kompletní, vybrány jsou jen ty domény, které ukazují snadno svůj LOC záznam, například přes program DIG:

- `alink.net`
- `caida.org`
- `chagas.eti.br`
- `ckdhr.com`
- `distributed.net`
- `rc5stats.distributed.net`
- `goldenglow.com.au`
- `nikhef.nl`
- `vrx.net`
- `yahoo.com`