Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Vizualizace DNS dotazu

2019

Lukáš Sotorník

Vedoucí práce: doc. Mgr. Jan Outrata, Ph.D.

Studijní obor: Aplikovaná informatika, prezenční forma

Bibliografické údaje

Autor:	Lukáš Sotorník
Název práce:	Vizualizace DNS dotazu
Typ práce:	bakalářská práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2019
Studijní obor:	Aplikovaná informatika, prezenční forma
Vedoucí práce:	doc. Mgr. Jan Outrata, Ph.D.
Počet stran:	36
Přílohy:	1 CD/DVD
Jazyk práce:	český

Bibliographic info

Author:	Lukáš Sotorník
Title:	DNS query visualization
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2019
Study field:	Applied Computer Science, full-time form
Supervisor:	doc. Mgr. Jan Outrata, Ph.D.
Page count:	36
Supplements:	1 CD/DVD
Thesis language:	Czech

Anotace

DNS je uspořádaný systém doménových jmen, který slouží především k překladu doménových jmen na IP adresy. Pokud by systém DNS neexistoval, do prohlížeče by uživatelé museli zadávat přímo IP adresy serverů, které však nejsou pro člověka tak lehce zapamatovatelné. Tato aplikace má uživateli zobrazit trasu průběhu zpracování DNS dotazu, jak se IP adresa ze systému DNS získá, přes jaké servery se musí dotaz dostat a jaké odpovědi servery poskytly.

Synopsis

DNS is an organized system of domain names used to translating the names into IP addresses. If there was no DNS, users would have to enter server IP addresses into web browser manually but the adresses are not so easy to remember. This application aims to display to a user a route of DNS query processing, how an IP address is obtained from the system, through which servers and with what answers.

Klíčová slova: Vizualizace DNS dotazu; DNS dotaz; Systém doménových jmen

Keywords: DNS query visualization; DNS query; Domain name system

Rád bych poděkoval svému vedoucímu bakalářské práce za vstřícné jednání a konzultace.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce podpis autora

Obsah

Seznam obrázků

Seznam tabulek

Seznam zdrojových kódů

1 Úvod

DNS (Domain Name System) je systém, který slouží pro překlad doménových jmen na IP adresy. Pokud do prohlížeče zadáte například
www.example.com, počítač se nejprve dotáže takzvaného DNS serveru, který je uveden v konfiguraci počítače, na IP adresu této domény a jakmile dostane od tohoto serveru odpověď (IP adresu), počítač se připojí k serveru a zobrazí webovou stránku.

Tato práce má sloužit jako pomůcka k pochopení, jak celý systém překladu funguje. Součástí práce je aplikace, která disponuje vizualizací dotazu za pomocí Google Maps a testování funkčnosti DNS serverů.

V sekci o DNS podrobně popíši celý systém DNS. Vysvětlím, co je to doména, z jakých částí se skládá a jakou hierarchii tvoří. Popíši DNS servery, jaké typy existují, jaké informace a v jakém formátu jsou informace uloženy. I jaké typy dotazů na ně můžeme dotazovat a jaké odpovědi od nich můžeme získat. Uvedu, jak řešitel postupně získává informace od DNS serverů. V této části se také budeme zabývat protokolem DNS a nakonec vysvětlím, jak funguje zabezpečení pomocí DNSSEC.

V druhé části jsou pak stručně popsány použité technologie v rámci backend struktury, tedy logice celého projektu. V této části se čtenář dozví, co je to program dig a PHP. Přiblížím Nette framework, který jsem při psaní této aplikace použil. V této části popíši jen to nejdůležitější o Nette. Bude se jednat o architekturu MVP. Dále zmíním frontend technologie, tedy i vizuání stránku aplikace. Nastíním, že se bude jednat o HTML, CSS, preprocesor Less, jQuery a Google maps API.

V třetí části bude popsána aplikace z pozice programátora. Tato část bude sloužit také jako programátorská dokumentace a bude zahrnovat popis zpracování uživatelského dotazu, získání geografických informací o DNS serverech, jaké informace server při dotazu poskytne a další.

V poslední části uvedu, jaké požadavky jsou potřeba k nainstalování a ovládání celé aplikace, bude se jednat tedy o uživatelskou dokumentaci, která bude popisovat, jak uživatel může vizualizovat a testovat své dotazy.

2 DNS

2.1 Co je to DNS

Jak bylo zmíněno již v úvodu systém DNS slouží pro překlad doménových jmen na IP adresy. Jedná se o decentralizovanou distribuovanou databázi záznamů. Záznamy jsou rozmístěné na jmenných serverech. Řešitel, který žádá o překlad jména (domény), požaduje od jmenného (DNS) serveru konečnou odpověď. Ten ji buď má ve vyrovnávací paměti a nebo se na ni doptá jiných jmenných serverů. Celý tento systém se pak skládá z domén, reverzních domén, zón, jmenných serverů, záznamů, řešitele a protokolů. Ještě dodám, že celý DNS popisuje dokument

RFC 1035.

2.2 Doména

Jedná se o strukturovaný název (jméno) oddělený tečkami. Každá doména, tvoří stromově hierarchické skupiny uzlů, kde každou skupinu spravuje jeden nebo více DNS serverů. Plně kvalifikované doménové jméno (FQDN) je úplný název domény pro konkrétní počítač. FQDN se skládá ze dvou částí: název hostitele a název domény. Například FQDN pro poštovní server může být `mx1.seznam.cz.` Název hostitele je mx1 a ten se nachází v doméně seznam.cz.. Když se podíváme znovu na předchozí doménové jméno `mx1.seznam.cz.`, tak pokud toto jméno budeme číst obráceně, tedy zprava, potom první je takzvaná kořenová doména (.), anglicky root doména, tato doména většinou u doménového jména není součástí, hovoříme pak o relativním jméně. Pokud je na konci i tečka, pak se jedná absolutní jméno. Pod kořenovou doménou leží top-level domény, známé také jako domény 1. řádu, ty jsou spravované organizací IANA a jsou rozděleny do skupin generických, sponzorovaných, národních a internacializovaných. Další úrovně tvoří domény 2. řádu, domény 3. řádu, atd.. Takto složené jméno může mít až 255B. Hierarchie je zachycena na obrázku 1 .

2.2.1 Reverzní doména

Reverzní doména slouží pro překlad IP adresy na doménové jméno. K IP adrese je přiřazeno doménové jméno v doméně `in-addr.arpa`. Jméno je tedy vytvořeno tak, že se otočí IP adresa a přidá se na konec řetězec in-addr.arpa. Například pro IP adresu 37.210.9.44 bude jméno 44.9.210.37.in-addr.arpa. Jména reverzních domén jsou překládána stejným způsobem jako překlad na IP adresu. Hierarchie je opět zachycena na obrázku 2.

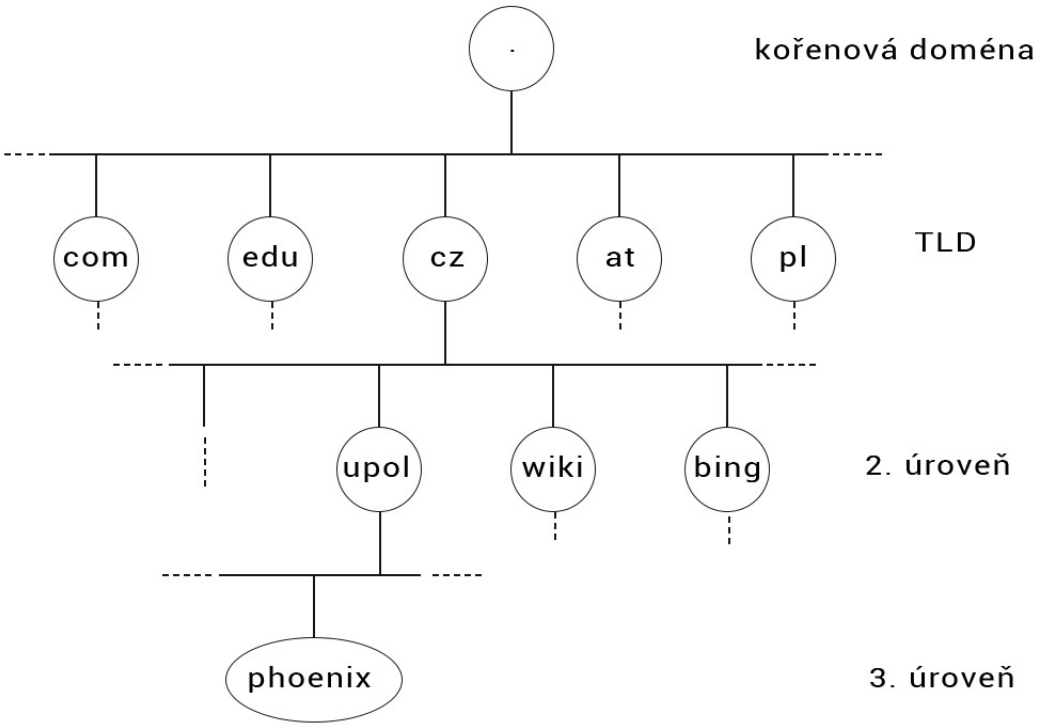
2.3 Zóna

Zóna DNS je nějaká část domény spravována jedním jmenným serverem.

2.4 Jmenné servery

Jmenný server obsahuje a spravuje záznamy pro svou zónu. Obsahuje IP adresy kořenových serverů a seznam jmenných serverů pro podřízené domény, jedná se o takzvané autoritativní záznamy. Mimo to si ukládá výsledky předchozího hledání do cache paměti. Proč si výsledky ukládá do cache? Snižuje tak nároky na výpočetně náročnější rekurzivní překlad. Po dobu TTL (time to live) si nechává výsledky dotazů v této paměti a pokud se řešitel znovu dotáže, pak poskytne odpověd, kterou má uloženou právě v této paměti. Hovoříme tak o neautoritativní odpovědi. Jmenné servery dělíme podle typů:

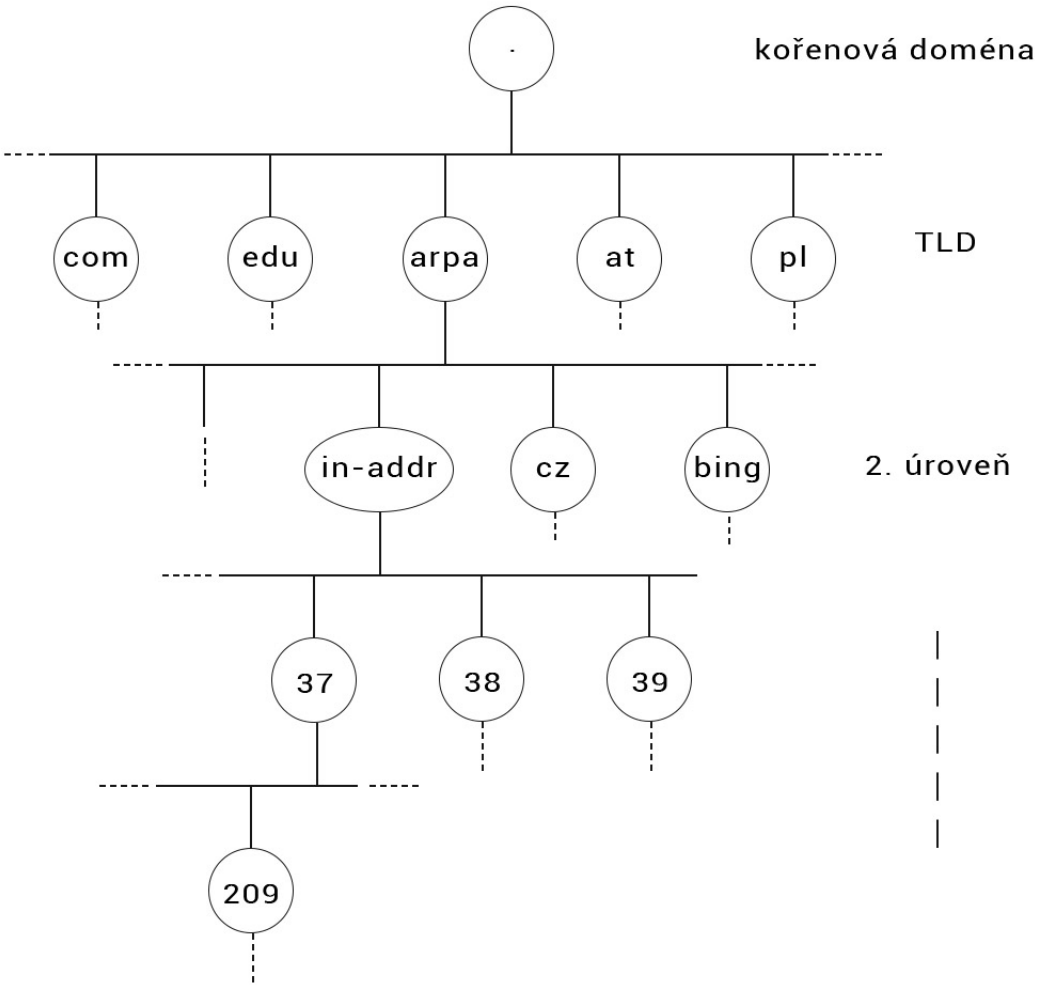
- primární



Obrázek 1: Hierarchie domén

- sekundární
- kořenový
- caching only
- forwarder

Primární server poskytuje autoritativní odpovědi ze své zóny a může obsahovat i neautoritativní odpovědi z cache paměti. Sekundární server pravidelně duplikuje záznamy zóny z primárního serveru. Kořenový server poskytuje záznamy pro kořenovou zónu. Kořenových serverů je na světě 13, přičemž každý z nich je jištěn minimálně jednou kopií. Tyto servery jsou spravovány organizacemi, které uvadí následující tabulka 1. Caching only servery poskytují pouze neautoritativní záznamy z cache paměti. Forwarder překládá dotaz pro jiný server. Pro každou doménu existují vždy minimálně dva jmenné servery: primární a sekundární.



Obrázek 2: Reverzní domény

2.5 Řešitel

Řešitel neboli resolver (může se jednat o komponentu operačního systému či aplikace), který se dotazuje na překlad jména, vyžaduje od jmenného serveru konečnou odpověď. Resolver může podobně jako jmenný server obsahovat cache s výsledky předchozích dotazů. Pokud však tuto cache neobsahuje, jedná se o pahýlový resolver. Odpověď od serveru může být v případě úspěchu kladná, požadovaná IP adresa byla nalezena, nebo při nenalezení záporná, neexistující doména. Řešitel má v konfiguraci IP adresy jmených serverů místní domény, kterých se táže. Konfiguraci je možné staticky nebo dynamicky naplnit. V operačních systémech odvozených od Unixu se jedná o soubor `/etc/resolv.conf` a v MS Windows se jedná o záložku nastavení TCP/IP.

Tabulka 1: Kořenové DNS servery

Název	Správce
A	VeriSign Global Registry Services
B	University of Southern California - Information Sciences Institute
C	Cogent Communications
D	University of Maryland
E	NASA Ames Research Center
F	Internet Systems Consortium, Inc.
G	U.S. DOD Network Information Center
H	U.S. Army Research Lab
I	Autonomica/NORDUnet
J	VeriSign Global Registry Services
K	RIPE NCC
L	ICANN
M	WIDE Project

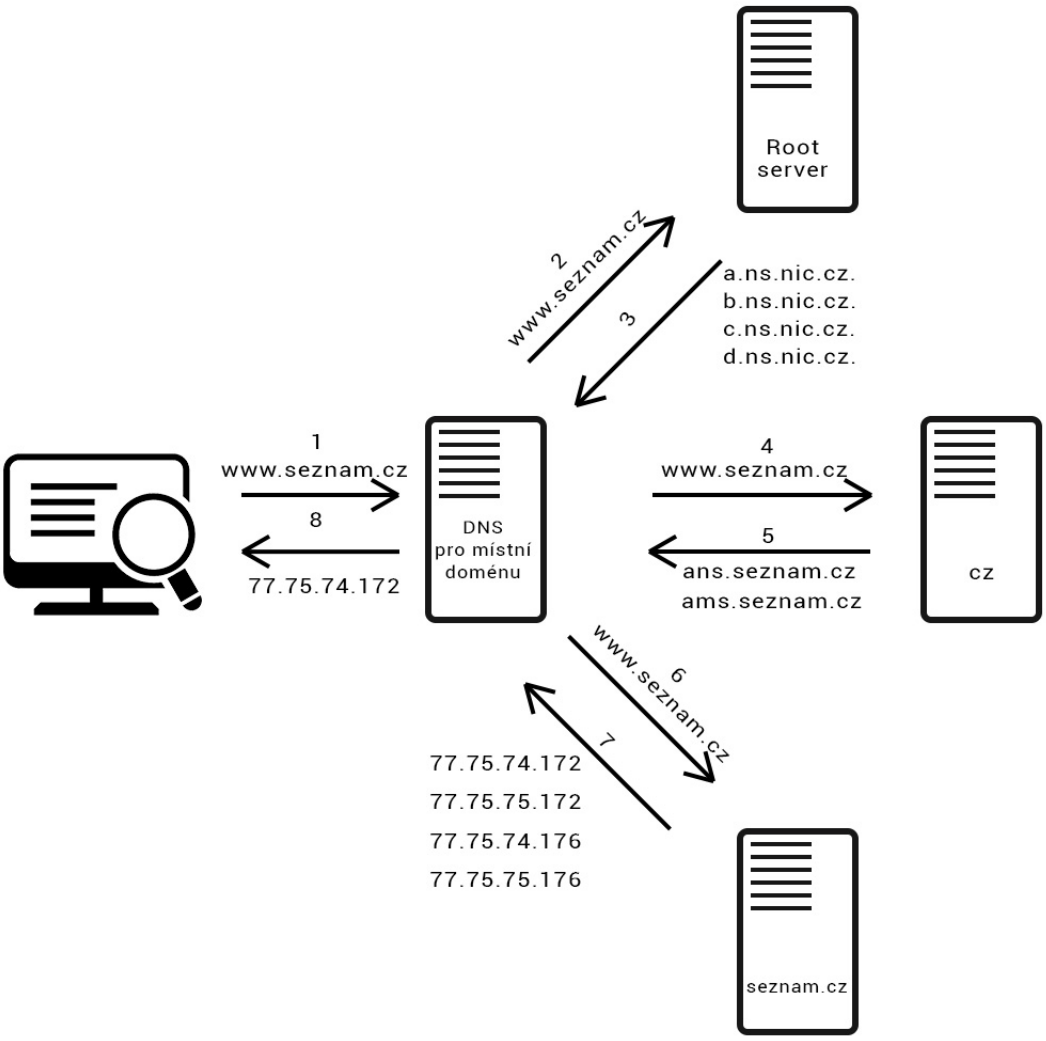
Příklad vyřešení dotazu

Klient požádá řešitele o překlad. Pokud řešitel má cache, tak ji prohledá a v případě cache hit vrátí odpověď, jinak řešitel vznese rekurzivní dotaz na jmenný server pro místní doménu. Tento server prohledá svoji cache a v případě cache hit vrátí výsledek, jinak vznese nerekurzivní dotaz na kořenovou doménu. Kořenový server v případě, že nezná odpověď, poskytne seznam jmenných serverů pro top-level domény. Náš server vznese nerekurzivní dotaz na některý server pro top-level doménu a ten vrátí seznam serverů pro domény vyššího řádu. A tak dále. Tento proces se nazývá rekurzivní překlad. Překlad je zobrazen na obrázku 3. Tato odpověď bude vždy neautoritativní. DNS server, kterého se dotazujeme, si ji nejprve uloží do paměti cache a poté ji vrátí. Jestliže potřebujeme z nějakého důvodu autoritativní odpověď, může řešitel vykonat iterativní dotaz, kde se postupně dotazuje jednotlivých DNS serverů, přičemž začíná od kořene a postupuje směrem k doménám vyššího řádu.

2.6 DNS záznamy

Nazývají se také RR (resource records) věty. Jedná se o způsob uložení dat v DNS serverech a v DNS paketech. Každý záznam se skládá z pěti částí:

- vlastník - jedná se o doménové jméno
- TTL - specifikuje, jak dlouho může být záznam uložen v cache
- třída - slouží k identifikaci protokolu



Obrázek 3: Rekurzivní překlad

- typ - hodnota, která specifikuje záznam
- rdata - data záznamu

Příklad konkrétního záznamu: `seznam.cz. 300 IN A 77.75.74.176`

- vlastník - seznam.cz.
- TTL - 300 (sekund)
- třída - A
- typ - IN (internet)

Tabulka 2: RR věty

Typ	Význam
A	Určen pro nastavení IPv4 adresy
AAAA	Určen pro nastavení IPv6 adresy, pokud server podporuje
NS	Obsahuje jména autoritativních DNS serverů podřízené domény
TXT	Slouží pro zapsání libovolného textového řetězce
SOA	Nese informace o zóně
CNAME	Oznamuje alias domény
PTR	Reverzní záznam
MX	Slouží k nastavení emailového serveru

- rdata - 77.75.74.176

Tříd existuje jen malé množství. Nejduležítější z nich jsou třídy IN a ANY. IN je zkratka pro internet a třída ANY je použita pouze při dotazech. Obsah dat (rdata) je závislý na typu. Pro typ = A, rdata obsahuje adresu IPv4. Všechny záznamy jsou uloženy v textové podobě. V podsekci níže se zaměřme na typy záznamů.

2.6.1 Typy záznamů

Nejdůležitější typy záznamů a jejich význam popisuje následující tabulka 2. Pokud byste se chtěli podívat na všechny typy, pak vás odkáži na stránku wikipedia (https://en.wikipedia.org/wiki/List_of_DNS_record_types). Záznamy typu RRSIG, DNSKEY a DS rozeberu podrobněji v kapitole 2.8 DNSSEC. Nyní jen nastíním, jaké informace tyto záznamy obsahují. Záznam RRSIG nese podpis DNSSEC pro množinu záznamů. Záznam DNSKEY obsahuje veřejný klíč, který mohou resolvery použít k ověření podpisů v záznamech RRSIG. DS odkazuje na záznam DNSKEY v nadřazené zóně. Způsob ověření podpisu přijatého se záznamy je popsán v již zmíněné kapitole o DNSSEC.

2.7 Protokol DNS

Jedná se o protokol aplikační vrstvy referenčního modelu ISO/OSI, který pracuje pod architekturou klient-server. Klient odešle dotaz a server na něj odpoví. Protokol používá pro přenos dva transportní protokoly - TCP a UDP, oba s porty 53. Primárně se používá UDP. V případě, že je odpověď dlouhá, jde přenos přes TCP. Používá se stejný protokol, jak pro dotaz, tak i pro odpověď. Základní operace pro získání informací z jmenných serverů se nazývá DNS query. Paket DNS query je rozdělen na 5 částí:

- záhlaví

- sekce dotazů
- sekce odpovědi
- sekce autoritativních serverů
- sekce doplňujících informací

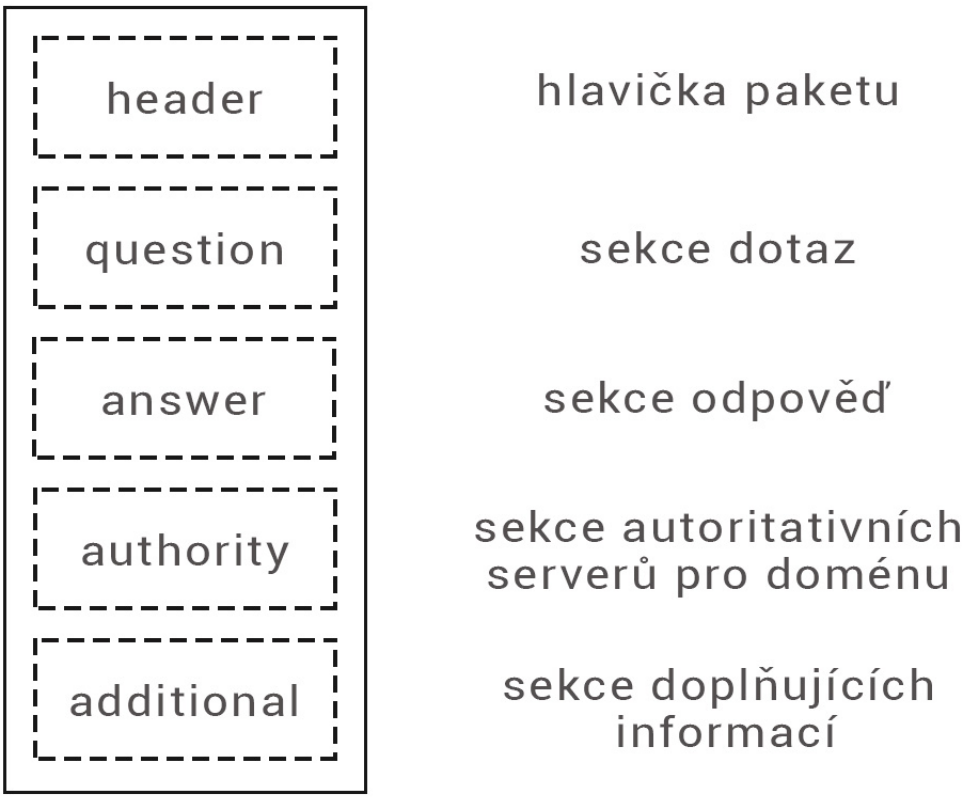
Záhlaví (HEADER) má pevnou délku a obsahuje ID, příznaky a informace o ostatních sekcích. Sekce dotazů (QUESTION SECTION) obsahuje dotaz (NAME), typ dotazu (TYPE) a třídu dotazů (CLASS). Sekce s odpověďmi (ANSWER SECTION) může obsahovat seznam záznamů, z nichž každý obsahuje NAME, TYPE, CLASS, TTL a RDATA. Autoritativní sekce (AUTHORITY SECTION) může obsahovat seznam záznamů, které odkazují na autoritativní servery. Doplňující sekce (ADDITIONAL SECTION) může obsahovat doplňující informace. Jedná se například o IP adresy autoritativních serverů. Schéma DNS paketu si můžete prohlédnout na obrázku 4.

Příznaky v záhlaví

- QR - 1 pro odpověď a 0 pro dotaz
- OPCODE - 0000 = dotaz
- AA - autoritativní odpověď
- TC - zkrácená odpověď
- RD - požadavek na rekurzivní překlad
- RA - dostupná rekurze
- AD - ověřená data (DNSSEC příznak)
- CD - zakázaná kontrola (DNSSEC příznak)

2.8 DNSSEC

Již v sekci o jmenných serverech jsem zmínil, že data jsou uložena a přenášena v textové podobě. Můžeme se spoléhat na to, že data na serverech jsou důvěryhodná? Uživatel si nemá jak ověřit původ požadovaných dat. Představme si následující situaci: Útočník udělá kopii například internetového obchodu a zaútočí na DNS server. Změní IP adresu a nasměruje záznam na svůj DNS server. Vy, jakožto klient nic nepoznáte. Url v prohlížeči bude stejná, ostatně i vzhled stránky bude souhlasit. Při objednání zboží zadáte údaje o své platební kartě a odešlete formulář. Útočník tak získá vaše údaje ke kartě! Jednou z možností obrany je DNSSEC. DNSSEC (Domain Name Security Extensions), jedná se o rozšíření protokolu DNS, které zabezpečuje záznamy na jmenných serverech a v DNS paketech pomocí digitálních podpisů.



Obrázek 4: DNS paket

2.8.1 DNSSEC podpis a validace

Prvním krokem k zabezpečení je seskupení všech záznamů se stejným typem do množiny záznamů (RRset). Poté je množina digitálně podepsána. K podepisování se používají dva typy klíčů. ZSK (Zone signing key), který podepisuje obsah zóny a KSK (Key signing key), který se používá pro podepisování DNSKEY i dalších záznamů. KSK obsahuje větší počet bitů a je většinou silnější. Podepsání a validace přes tento klíč je výpočetně náročnější, proto se používá pouze pro vytvoření jednoho podpisu v zóně. ZSK je pak klíč, který má menší počet bitů než KSK a je použitý pro podpis všech záznamů v zóně. Takže když záznamy máme digitálně podepsány, je potřeba dále v zóně mít přístupný veřejný ZSK. Resolver si pak může vyžádat záznam DNSKEY obsahující veřejný ZSK od DNS serveru. Odpověď ověří společně s RRsetem, RRSIG a s veřejným ZSK. Pokud resolver důvěřuje klíči podepisování zón v záznamu DNSKEY, může důvěřovat všem záznamům v zóně. Co když byl podvržen klíč podepisující zónu? Potřebujeme najít způsob, jak ověřit veřejně dostupný ZSK. K ověření ZSK se použije KSK. KSK validuje DNSKEY záznam stejně, jako ZSK. Takže validace může vypadat takto: Dorazí k nám požadovaná skupina záznamů (RRset) a k nim

Tabulka 3: RRSIG záznam

Položka v záznamu	Popis
NS	Typ podepsaných záznamů
8	Použitý algoritmus
2	Počet podepisovaných názvů doménového jména
518400	TTL původního záznamu
20190728140313	Datum, kdy podpis expiruje
20190628123313	Datum začátku platnosti podpisu
1202	Keytag klíče použitého pro vytvoření podpisu
.	Jméno zóny
RvO0ZOJiwpy...	Samotný digitální podpis

Tabulka 4: DNSSEC záznamy

Typ	Význam
RRSIG	Obsahuje podpis pro sadu záznamů
DNSKEY	Obsahuje veřejný podepisovací klíč
DS	Obsahuje hash záznamů DNSKEY
NSEC a NSEC3	Poskytují informaci o neexistejci záznamů

digitální podpis RRSIG. Resolver si vyžádá záznamy DNSKEY obsahující veřejný ZSK a veřejný KSK, které jsou také digitálně podepsány. Ověří se RRSIG RRsetu veřejným ZSK a ověří se RRSIG DNSKEY RRsetu pomocí veřejného klíče KSK. Nyní jsme si vybudovali důvěru v naší zóně. DNS je ale hierarchický systém. Veřejný KSK je podepsán sám svým KSK soukromým klíčem, což zase neposkytuje žádnou další důvěru. Potřebujeme způsob, jak důvěřovat i nadřazené zóně. To nás přivádí k pojmu DS (Delegation signer). Záznam DS slouží pro přenos důvěry z nadřazené zóny do podřízené zóny. Správce zóny vytvoří hash záznamu DNSKEY obsahující veřejné KSK a předá ji do nadřazené zóny jako DS záznam. Pokaždé, když je resolver odkazován na podřízenou zónu, poskytne nadřazená zóna záznam DS. Tento záznam slouží k tomu, aby resolver věděl, že v podřízené zóně je povoleno DNSSEC. Chce-li zkontrolovat platnost veřejného KSK podřízené zóny, porovná hash se záznamem DS od rodičovské zóny. Pokud se shodují, tak resolver předpokládá, že veřejný KSK nebyl změněn, což znamená, že může důvěřovat všem záznamům v podřízené zóně. Takto se v DNSSEC vytváří řetězec důvěry.

Do DNS infrastruktury tedy muselo přibýt několik nových DNS záznamů, které uvádí tabulka 4. K těmto záznamům lze přistupovat běžným způsobem jako ke všem ostatním záznamům. Schéma validování je zobrazeno na obrázku 5.

Tabulka 5: DNSKEY záznam

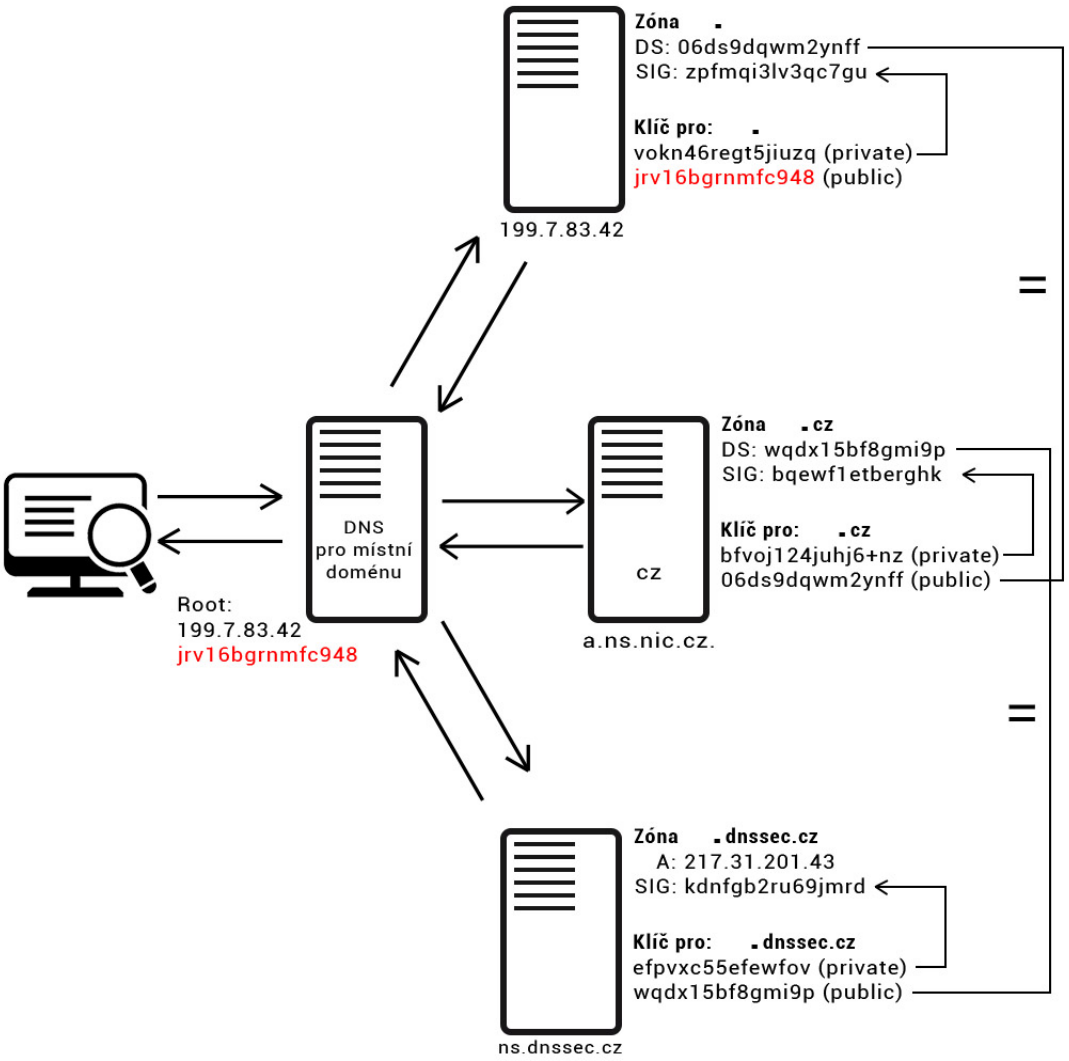
Položka v záznamu	Popis
256	Příznak (ZSK nebo KSK)
3	Protokol (zafixované číslo)
13	Použitý algoritmus
xLycjTuc8KrysXE...	Veřejný klíč

Tabulka 6: DS záznam

Položka v záznamu	Popis
51575	Identifikátor DNSKEY záznamu
13	Použitý algoritmus
2	Digest Typ - hash použitý k vytvoření hodnoty Digest
B09A202905D6...	Digest - Hodnota hašování záznamu DNSKEY

Tabulka 7: Typy algoritmů

Číslo	Typ
1	RSA/MD5
2	Diffie-Hellman
3	DSA
5	RSA/SHA1
6	DSA/SHA1/NSEC3
7	RSA/SHA1/NSEC3
8	RSA/SHA-256
10	SA/SHA-512
12	GOST R 34.10-2001
13	ECDSA Curve P-256 with SHA-256
14	ECDSA Curve P-384 with SHA-384
15	Ed25519



Obrázek 5: DNSSEC validace

3 Použité technologie

3.1 Dig program

Dig (domain information groper) je nástroj příkazového řádku pro dotazování DNS serverů. Provádí vyhledávání DNS a zobrazuje odpovědi, které jsou získány z dotazovaného serveru, popřípadně serverů. Většina správců jmenných serverů jej používá k řešení problémů z důvodů snadného použití a přehlednosti výstupů. Ostatní vyhledávací nástroje mají obvykle menší funkčnost než dig. Program dig disponuje různými možnostmi výstupů. Tyto výstupy záleží na použití argumentů. Stručný souhrn jeho parametrů a možností naleznete, pokud napíšete příkaz `dig -h` do příkazového řádku. Pokud není zádán dotaz na konkrétní DNS

server, dig se dotáže serverů uvedených v konfiguraci počítače. Typické použití tohoto nástroje vypadá následovně `dig @server name type`, kde `@server` je jméno nebo IP adresa dotazovaného serveru, `name` pak slouží k zadání dotazu a `type` indikuje, jaký typ záznamu požadujeme. Pokud se potřebujete podívat na detailnější popis programu dig, pak vás odkáži na stránku ftp.isc.org (https://ftp.isc.org/isc/bind/9.11.0a1/doc/arm/man.dig.html).

3.2 PHP

PHP je skriptovací dynamicky typovaný jazyk, který pracuje na straně serveru. Jedná se nejrozšířenější skriptovací jazyk pro tvorbu webových stránek.

3.2.1 Nette framework

Nette je relativně obsáhlý PHP framework, který se používá pro řadu webových aplikací. Jedná se o soubor knihoven, které umožní lehčí práci při psaní kódu. Disponuje srozumitelnou syntaxí, zahrnuje nejdůležitější bezpečnostní funkce, obsahuje velice kvalitní ladicí program s názvem TRACY a mnohé další.

3.2.2 Packages

Nette je tvořený sadou balíčků. Doporučený způsob instalace je pomocí composeru a to příkazem `composer require (název balíčku)`. Příklad composeru se nachází ve zdrojovém kódu 1.

```
1 {
2    "require": {
3       "php" : ">=7.0",
4       "nette/nette": "~2.1.0"
5    }
6 }
```

Zdrojový kód 1: Ukázka composeru

3.2.3 Architektura MVP

Nette je postaven na architektuře MVP, je to zkratka pro Model, View, Presenter (častěji je však používáno místo presenter controller), je to tedy architektura, která se používá v aplikacích s grafickým uživatelským rozhraním.

3.2.4 Model

Model zahrnuje aplikační logiku. V tomto projektu je hlavní model BaseManager, který se stará o rozparsování výsledku z dig programu.

3.2.5 View

View je aplikační vrstva, která se stará o zobrazení aplikace uživateli. K vykreslení se využívají šablony a šablonovací systém pod názvem Latte. Latte umí zabezpečit výstup před zranitelnostmi. Obsahuje také mnoho užitečných maker. Používají se dva druhy speciálních značek:

- makra ve složených závorkách - `{foreach ...}`
- n:makra - `n:if="výraz"`

Latte disponuje vlastností, že proměnné, které vkládáte do view, escapuje. Zanedbáním nebo opomenutím by mohla vzniknout bezpečnostní díra Cross Site Scripting (XSS). Názorná ukázka podmínky if a foreach cyklu se nachází ve zdrojovém kódu 2.

```
1 <ul n:if="$items">
2    <li n:foreach="$items as $item">{$item|capitalize}</li>
3 </ul>
```

Zdrojový kód 2: Ukázka podmínky if a cyklu foreach v Latte

3.2.6 Presenter

Presenter předává do šablony potřebné a proměnné, vytváří formuláře a provádí také jejich zpracování. Presenter se stará o volání aplikační logiky (modelu) a po zpracování řekne prohlížeči, že může data zobrazit. Hlavním presenterem této aplikace je BasePresenter, který obsahuje metodu na vytvoření formuláře pro dotaz a metodu na zpracování uživatelského dotazu.

3.3 HTML

HTML je zkratka pro Hypertext Markup Language. Jedná se tedy o značkovací jazyk. HTML se používá k tvorbě webových dokumentů. Ukázka html se nachází ve zdrojovém kódu 3.

3.4 CSS

CSS (cascading style sheets). Kaskádové styly slouží k vylepšení vzhledu webových stránek. Při psaní webové aplikace, tedy při psaní CSS nám vznikne jeden problém, a to, jak pojmenovávat třídy, aby v aplikaci nevznikl chaos. Existuje mnoho metodik, jak CSS organizovat. Já jsem zvolil metodiku BEM, která má určitá pravidla a omezení způsobu pojmenovávání tříd, jak ukazuje následující tabulka 8.

```
1
2  <div class="col-md-12">
3     <div id="map"></div>
4  </div>
5  <div class="col-md-12 full-respond text-center">
6     <button class="btn btn--red js-button-full-respond">
7        Záznam komunikace
8     </button>
9  </div>
```

Zdrojový kód 3: Ukázka zdrojového kódu v HTML

Tabulka 8: BEM jako Blok, Element, Modifikátor

Typ třídy	Způsob pojmenovávání
Blok	.block
Element	.block__element
Modifikátor	.block–modifier
Modifikátor elementu	.block__element–modifier

3.4.1 Preprocesor less

Preprocesor less umožňuje pracovat s proměnnými, matematickými operacemi, funkcemi a zejména zamezuje redundanci kódu. Názorná ukázka použití lessu se nachází ve zdrojovém kódu 4.

```
1  @width: 10px;
2  @height: @width + 10px;
3  #header {
4    width: @width;
5    height: @height;
6  }
```

Zdrojový kód 4: Ukázka preprocesoru Less

3.5 Javascript

Jedná se o dynamicky typovaný skriptovací jazyk, ale tentokrát na straně klienta. Javascript nevyžaduje žádný preprocessing a je interpretován přímo prohlížečem. Umožňuje reagovat na události, provádět změny obsahu stránky bez nutnosti znovunačtení stránky a další.

```
1 <script src="assets/js/script.js"></script>
```

Zdrojový kód 5: Začlenění javascriptu do HTML

3.5.1 jQuery

Účelem jQuery je usnadnit používání javascriptu na webových stránkách. Jedná se tedy o javascriptovou knihovnu, která především zpřehledňuje psaní kódu. jQuery se v tomto projektu výrazně podílí na vizualizaci cesty v mapách, dále pak s pomocí jQuery je zajištěno odchytávání události kliknutím na tlačítko **zobrazení / skrytí záznamu celé komunikace**. Dále pak při kliknutí na kontaktované servery zobrazí vyskakovací okno s informacemi a při zvolení možnosti `Použít jiný DNS server` zobrazí pole pro zadání serveru. Ukázka jQuery se nachází ve zdrojovém kódu 6.

```
1 var inputQueryText = $('.js-input-query-text');
2 var inputQuerySubmit = $('.js-input-query-submit');
3 inputQueryText.on('focus', function (event) {
4    event.preventDefault();
5    inputQuerySubmit.addClass('btn-border--red');
6 });
```

Zdrojový kód 6: Ukázka zdrojového kódu jQuery

3.6 Google Maps javascript API

Google maps API poskytují perfektní nástroje pro simulaci cesty, proto jsem se rozhodl je v této aplikaci použít. Nastavení mapy je velice dobře přizpůsobitelné, je zde nástroj na simulaci cesty mezi vytvořenými značkami na mapě. API umožňuje nastavit tloušťky čar, rychlost, tvar, barvu cesty nebo přidávání vyskakovacích oken při najetí myši na čáru či značku. Příklad vytvoření značky na mapě za pomocí Google maps API se nachází ve zdrojovém kódu 7.

```
1
2 var marker = new google.maps.Marker({
3   position: latLngPoint,
4   pos: pos,
5   map: map,
6   visible: true,
7   latLng: latLngPoint,
8 });
```

Zdrojový kód 7: Vytvoření značky na mapě

To by bylo z použitých technologií vše. Nyní se pojďme podívat na dokumentaci z pozice programátora.

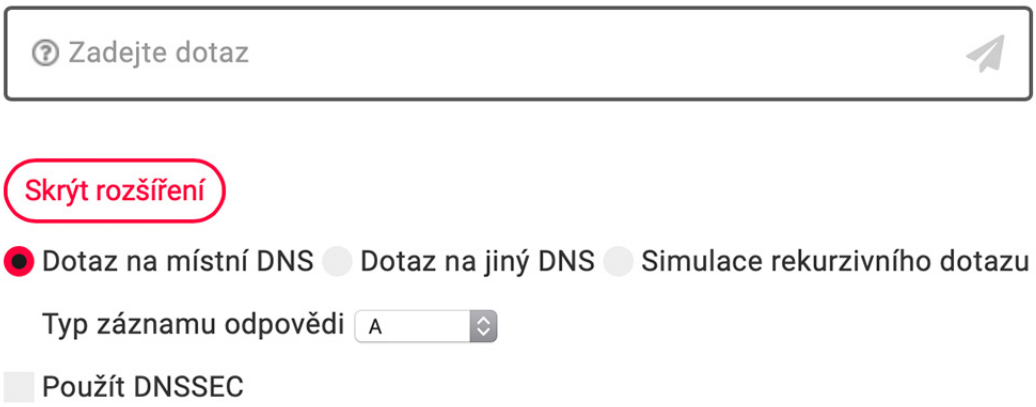
4 Programátorská dokumentace

4.1 Zvolení nástroje pro dotazy na DNS

Před samotným začátkem návrhu programu, jsem narazil na problém jaký vlastně zvolit nástroj pro řešení překladu uživatelského dotazu. Samotné PHP disponuje funkcí `dns_get_record`, které však pouze vrací záznamy. Sekci jako je HEADER nevrátí nebo nejde zadat dotaz na jiný DNS server. Ve skutečnosti není příliš mnoho nástrojů na testování funkčnosti DNS serverů. Například Linux i Windows disponuje programem příkazové řádky `nslookup`, který je ovšem zastaralý a nejsou v něm již opravovány chyby. Na OS odvozených od Unixu lze k dotazům na DNS servery použít programy jako `dig` nebo `host`. Programy dig a host se chovají částečne odlišně. Výstupy jsou zachyceny na obrázcích dig 8 a host 7. Pro řešení dotazů jsem zvolil program dig. Ano, je sice pravda, že dig se vyskytuje pouze u operačních systémů odvozených od Unixu, avšak zle jej stáhnout a naistalovat i na Windows. Viz sekce 5.2 instalace. A proč jsem zvolil program dig? Jeho výstupní formát je více "surový". Dig ve svém výstupu přímo zobrazuje všechny 4 sekce: sekce s dotazem, sekce s odpovědí, autoritativní a doplňující sekci. Také zobrazuje příznaky v záhlaví a ještě k tomu má mnoho nepovinných parametrů a tudíž více možností výstupů. Pokud zavoláme program dig s jediným parametrem, s doménou, vrátí se nám v případě úspěchu seznam IP adres dané domény. Jestliže se potřebujeme dotázat například na MX záznam, přidáme parametr `MX`, který napíšeme před doménu a dig v případě nalezení vrátí MX záznamy. Výstup si můžete prohlédnout na obrázku 9.

4.2 Rozdělení aplikace

Aplikace je rozdělena do dvou šablon: Vizualizace a Testování. V šabloně s vizualizací je vykreslen formulář, který obsahuje vstupní pole a selectbox s výběrem odpovědi. Přepínač na simulaci rekurzivního dotazu je automaticky zaškrtnut. Bylo by nelogické provádět vizualizaci bez rekurze, to by se pak zobrazila pouze trasa od řešitele k jednomu kontaktovanému serveru. Jestli by tento server provedl rekurzi, to už by záleželo na něm. Nenašel jsem způsob, jak tohle detekovat. Snad jen podle ttl přijatého záznamu. To jak se řeší rekurzivní dotaz se dozvíte v sekci 4.3.1 rekurzivní a nerekurzivní dotaz. V šabloně s Testování je formulář (zachycen na obrázku 6) poněkud obsáhlejší. Obsahuje přepínací tlačítka na jaký DNS se má dotaz provést. Pokud uživatel zvolí možnost použít místní nebo jiný DNS server, pak je dotaz vykonán na tento server a ten ji buď má v cache a nebo se na ni rekurzivně doptá a vrátí výsledek. Je zde také možnost použít DNSSEC. Na konci šablon je tlačítko pro zobrazení záznamu celé komunikace.



Obrázek 6: Formulář v šabloně Testování

4.3 Zpracování uživatelského dotazu

4.3.1 Rekurzivní a nerekurzivní dotaz

Již jsem zmínil, že nelze určit DNS servery, kterých se resolvující DNS server musel postupně doptat při rekurzivním dotazu, proto jsem vytvořil uměle rekurzivní dotaz, který je zpracovánán ve `while` cyklu způsobem zakázání rekurze v dig programu. Z pole, které jsem naplnil informacemi o kořenových DNS serverech, se náhodně vybere jeden kořenový DNS a dig poté vykoná dotaz na tento server. Ten vrátí seznam serverů podřízených domén, kde se vybere první z tohoto seznamu, na který je vykonán další dotaz a tak dále, až do konečné odpovědi. Musíme si uvědomit, že na takto uměle vytvořeném rekurzivním dotazu na autoritativní servery (to samé platí na jakýkoliv DNS server, který nepodporuje rekurzi) nelze jen tak použít parametr `+dnssec` k ověření pravosti přijatých dat. Je potřeba se znovu doptat na digitální podpisy a ty dále ověřit. Tuto funkcionalitu bych rád implementoval na magisterském studiu. Pokud tedy zvolíme přepínač na rekurzivní dotaz, zaškrtávací políčko na použití dnssec zmizí. Nyní se zaměřme na rozparsování výstupu z dig.

4.3.2 Parsování výstupu z programu dig

Výstup je ve formátu řetězce, proto je nutné tento řetězec nějak rozparsovat. Jako první přijde na řadu rozdělení podle konců jednotlivých řádků. Při procházení jednotlivých řádků v cyklu jsem si pohlídal, kde se vyskytuje sekce HEADER

a extrahoval z ní informace, které jsem uložil do proměnných. Pokud dig vrátil sekci s odpověďmi (ANSWER SECTION), tak jednotlivé záznamy jsou zpracovány funkcí `parseRecord`, která sestavila jméno funkce pro konkrétní typ záznamu. Například záznam typu A nese pouze hodnotu IP adresy, kdežto záznam typu DS nese hodnot více. Samozřejmě, že záznamů na DNS serverech je spousta, já jsem program vytvořil pro těchto několik základních typů. A, AAAA, NS, MX, TXT, RRSIG, CNAME, SOA, DS, DNSKEY, PTR. Kompletní typy záznámů můžete naleznout na adrese
wikipiedia (https://en.wikipedia.org/wiki/List_of_DNS_record_types). Pokud není přítomna sekce ANSWER, a je zvolen rekurzivní dotaz, přijde na řadu parsování záznamů s jmennými servery, to se provede stejným způsobem jako parsování záznamů v sekci ANSWER, ale nastaví se příznak, že se nejedná o odpověď, ale o odkazované servery. Pokud výsledek obsahuje i sekci ADDITIONAL, tak se k jenotlivým záznamům nastaví položky z této sekce. Poslední z výstupu dig je název kontaktovaného serveru a režijních informací jako je velikost přijatých dat a čas vykonání dotazu.

4.3.3 Zjištění geografických informací

Ke kontaktovanému serveru je potřeba zjistit geografické informace pro zobrazení na mapě. Po dlouhém hledání a zkoušení různých možností jsem našel stránku ip-api (http://ip-api.com), která poskytuje potřebná data o lokalizaci. Pokud v PHP vykonáte příkaz

```
@file_get_contents(http://ip-api.com/json/188.175.84.97);
```

dostanete informace o poloze v podobě json formátu. Informace jsou zachyceny na obrázku 10. Je tu ale jisté omezení. Mezi dvěmi žádostmi na server musí být rozdíl minimálně dvou sekund, což se zajistí ve while cyklu příkazem `sleep`. Raději jsem zadal jako parametr 3 sekundy. Je tu ještě jedna maličkost. Někdy informace ze serveru `ip-api.com` nejsou 100%. Stalo se mi, že server nevrátil zeměpisnou šířku a délku, a tak následovalo další hledání, jak informace získat. Nakonec se mi podařilo najít stránku
`http:/ipinfodb.com`, která poskytuje stejné informace, jako předchozí server. Takže pokud se nepodaří získat informace o poloze z prvního serveru, je dotázán právě tento server. Pokud se při rekurzivním dotazu použije jako první DNS server uvedený v konfiguraci počítače, pak se většinou jedná o IP adresu výchozí brány routeru. Tato adresa je ve většině případů ve tvaru `192.168`. Jelikož se jedná IP adresu z rozsahu privátní sítě, nelze u této IP adresy zjistit geografické informace, a proto je značka na mapě zobrazena se souřadnicemi zeměpisné šířky 0 a délky 0. Pokud se jedná o řešitele, tak je poloha zjištěna z veřejné IP adresy klienta. Jestliže není dostupná veřejná IP adresa nebo ji nelze zjistit, je použita IP adresa ve tvaru 195.113.148.241, která má geografickou polohu v Olomouci. Pro účel vizualizace je tato poloha dostatečná. Poslední věc, kterou bych zmínil o poloze je, že k některým IP adresám je přiřazena stejná zeměpisná šířka a délka, to znamená, že by se na mapě tyto body překrývaly. Tento problém jsem vyřešil tak, že pokud jsou dva body velmi blízko sebe, je

prvnímu z nich upravena zeměpisná šířka a délka o náhodné necelé číslo, a tak jsou tyto dva body na mapě od sebe vzdálené o několikset metrů. Jakmile jsou připravena všechna data, může začít samotná vizualizace za pomocí javascriptu, která je popsána v sekci vizualizace.

```
example.com has address 93.184.216.34
example.com has IPv6 address 2606:2800:220:1:248:1893:25c8:1946
```

Obrázek 7: Výstup z programu HOST

4.4 Vizualizace dotazu

K samotné vizualizaci trasy na mapě Zemi jsem použil programovací jazyk javascript společně s Google Maps javascript API. Kód se vyskytuje v latte šabloně `visualization`. Jakmile máme připravena data z presenteru nastává práce na straně javascriptu. Jako první je zavolána funkce `initMap()`, která zjistí z DOM struktury element s mapou podle identifikátoru a předá ji potřebná nastavení. Do proměnné `bounds` je vložen každý kontaktovaný server, to aby se okno mapy mohlo přiblížit přesně na tyto servery. Dále je vytvořena značka řešitele a jmenných serverů pomocí `google.maps.Marker`, které jsou zobrazeny na mapě. Poté je naplněná proměnná `polylines`, která nese hodnoty typu `google.maps.Polyline`, tedy trasu mezi dvěma servery. Příklad vytvoření je ukázán ve zdrojovém kódu 8. Trasa se postupně vykreslí pomocí `setInterval`. Dále se ke každému serveru (bodu na mapě) vytvoří vyskakovací okno a při najetí kurzorem na bod se toto okno zobrazí. V okně jsou informace o serveru tj. **IP adresa serveru**, pokud má server jméno, je zobrazeno **jméno** + ve které **zemi** a **městě** se nachází. Následně je spuštěna animace cest z počátečního bodu `questionerMarker` do prvního DNS serveru, dále nastává iterativní proces, kdy se DNS server postupně doptává jiných jmenných serverů. Samotná animace se provádní pomocí `setInterval`, kde jako parametr je funkce v jejímž těle se provádí výpočet animace. Při každém průhodu intervalu je zvýšena proměnná `step` o 1 a tedy polyline se prodlouží také o jednu jednotku. Již při vykreslování je možno na polyline namířit kurzorem, kde se zobrazí okno s informacemi. Okno je rozdělené na čtyři části: část s dotazem, zde nalezneme položky **dotaz**, **typ dotazu**, **zdroj dotazu**, **cíl** (ke komu dotaz putuje), část s odpovědí, zde jsou zahrnuty položky: **typ záznamu**, **TTL**, **obsah přijatého záznamu**, **zdroj** a **cíl**. Třetí část je společná a obsahuje režijní informace **čas** a **velikost** obdrženého

```
; <<>> DiG 9.10.3-P4-Debian <<>>   www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61046
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.example.com.               IN      A

;; ANSWER SECTION:
www.example.com.        86400   IN      A       93.184.216.34

;; AUTHORITY SECTION:
example.com.            86400   IN      NS      a.iana-servers.net.
example.com.            86400   IN      NS      b.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.     16522   IN      A       199.43.135.53
a.iana-servers.net.     16522   IN      AAAA    2001:500:8f::53
b.iana-servers.net.     16522   IN      A       199.43.133.53
b.iana-servers.net.     16522   IN      AAAA    2001:500:8d::53

;; Query time: 353 msec
;; SERVER: 158.194.80.254#53(158.194.80.254)
;; WHEN: Mon May 06 13:00:27 CEST 2019
;; MSG SIZE  rcvd: 196
```

Obrázek 8: Výstup z programu DIG

výsledku. V poslední části mohou být vypsány příznaky, které server poskytl. Pro zpřehlednění je každá polyline opatřena číslovkou.

Zmiňovaný javascriptový kód na vizualizaci cesty DNS dotazu není jediný javascriptový kód v této aplikaci. Aplikace obsahuje soubor `script.js`, který zajišťuje další potřebné funkce celé aplikace:

1. Přizpůsobení labelu ve formuláři při focusu
2. Zobrazení celé komunikace dotazu po kliknutí na tlačítko
3. Zobrazení přidání DNS serveru
4. Zobrazení nastavení u Testování dotazu

```
; <<>> DiG 9.10.6 <<>> MX seznam.cz
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18124
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;seznam.cz.                     IN      MX

;; ANSWER SECTION:
seznam.cz.              82      IN      MX      10 mx1.seznam.cz.
seznam.cz.              82      IN      MX      20 mx2.seznam.cz.

;; ADDITIONAL SECTION:
mx1.seznam.cz.          82      IN      A       77.75.78.42
mx1.seznam.cz.          82      IN      A       77.75.76.42
mx1.seznam.cz.          82      IN      AAAA    2a02:598:a::78:42
mx1.seznam.cz.          82      IN      AAAA    2a02:598:2::42
mx2.seznam.cz.          82      IN      A       77.75.76.32
mx2.seznam.cz.          82      IN      A       77.75.78.32
mx2.seznam.cz.          82      IN      AAAA    2a02:598:a::78:32
mx2.seznam.cz.          82      IN      AAAA    2a02:598:2::32

;; Query time: 46 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Thu Jun 27 14:37:36 CEST 2019
;; MSG SIZE  rcvd: 254
```

Obrázek 9: Výstup z programu DIG s parametrem MX

5. Zobrazení okna u zobrazených DNS serverů

4.5 Testování dotazu

V poslední řadě je tu šablona Testování, kde si uživatel může zvolit dotaz na server uvedený v konfiguraci počítače, jiný DNS, který zadá IP adresou nebo názvem serveru, či rekurzivní dotaz. Pokud uživatelský dotaz proběhne úšpěšně, tak pomocí foreach cyklu jsou vypsány u každého kontaktovaného serveru jeho IP adresa, název, příznaky, zda server poskytl odpověď, případně na jaké DNS odkázal. Při kliknutí na tyto servery se zobrazí okno s popisem a dalšími informacemi. Pokud je přítomna odpověď, jsou vypsány jednotlivé položky záznamů.

```
{
    "query": "188.175.84.97",
    "status": "success",
    "continent": "Europe",
    "continentCode": "EU",
    "country": "Czechia",
    "countryCode": "CZ",
    "region": "71",
    "regionName": "Olomoucky kraj",
    "city": "Prostějov",
    "district": "",
    "zip": "796 01",
    "lat": 49.4333,
    "lon": 17.1167,
    "timezone": "Europe/Prague",
    "currency": "CZK",
    "isp": "RIO Media",
    "org": "",
    "as": "AS16246 RIO Media a.s.",
    "asname": "AS16246 Internet Provider",
    "mobile": false,
    "proxy": false
}
```


Obrázek 10: Informace o poloze

Každý výsledek obsahuje vrácenou hodnotu, typ, TTL, od koho byl záznam přijat a v jakém čase byl výsledek obdržen.

5 Uživatelská dokumentace

Tato kapitola slouží jako dokumentace pro uživatele. Popisuje jak aplikaci naistalovat, spustit a jak ji ovládat.

```
var polyline = new google.maps.Polyline({
  from: latLonFrom,
  to: latLonTo,
  geodesic: true,
  strokeColor: colorPath,
  strokeOpacity: 1.0,
  strokeWeight: strokeWeight,
  icons: [{
    icon: lineSymbol,
    offset: '100%'
    }],
  map: map,
  lastp: lastp,
  contentDivFull: contentDivFull,
  pathInfoWindow: pathInfoWindow,
  orderWindow: orderInfoWindow,
});
```

Zdrojový kód 8: Vytvoření polyline

5.1 Požadavky

Ke spuštění aplikace je potřeba mít k dispozici webový server a na něm naistalovaný skriptovací jazyk PHP. Dále je potřeba mít nainstalovaný program `dig`, který je ovšem u operačních systému odvozených od Unixu součástí. Návod jak nainstalovat program `dig` na platformě Windows přikládám jako externí soubor. Nejpoužívanějším webovým serverem je Apache HTTP Server, který již v sobě zahrnuje PHP, proto v sekci instalace bude vysvětleno naistalování aplikace právě na něm. Uživatel si může stáhnout volně dostupný multiplatformní softwarový balíček pod názvem XAMPP. Tento balíček obsahuje webový server Apache, databáze MariaDB, interpret programovacích jazyků PHP a mnoho dalšího. Jen dodám, že pokud nemáte nainstalován dig a XAMPP, tak je prvně potřeba naistalovat program dig, a až poté XAMPP.

5.2 Instalace

Instalace XAMPPu je jednoduchá. Návod jak stáhnout a naistalovat XAMPP naleznete na stránkách https://www.apachefriends.org/index.html. Po naistalování tohoto balíčku je potřeba překopírovat obsah adresáře aplikace `bin` do adresáře `XAMPP/htdocs`. Návod na instalaci programu dig na platformě Windows se vyskytuje v adresáři `instructions`, kde je soubor readme.md, ve kterém je popsán kompletní postup instalace.

5.3 Spuštění

Jakmile spustíte webový server Apache, tak jen stačí zadat do prohlížeče adresu `127.0.0.1` a aplikace je spuštěna. Po spuštění se klient dostane na stránku s vizualizací.

5.4 Vizualizace dotazu

K zadání uživatelského dotazu použije uživatel formulář, který je vidět ihned po načtení stránky. Uživatel si může také vybrat typ očekávané odpovědi. Po odeslání formuláře se zobrazí kontaktované servery na mapě Zemi a nastane proces simulace rekurzivního dotazu. Při najetí kurzorem na bod zobrazující DNS server se objeví okno s informacemi o tomto serveru. Pokud uživatel najede kurzorem na křivku znázorňující cestu putování dotazu mezi dvěma servery, zobrazí se okno s informacemi aktuální cesty.

5.5 Testování dotazu

V této šabloně je formulář s více parametry, než se vyskytuje v šabloně vizualizace. Pokud uživatel nerozklikne tlačítko **Rozšířené nastavení**, vykoná se dotaz na první DNS server uvedený v konfiguraci počítače. Jestliže uživatel vyžaduje vykonat dotaz na jiný DNS server, může využít možnost v rozšířeném nastavení, kde zadá DNS server, buď IP adresou, nebo názvem serveru. U těchto dvou možností je možno zaškrtnout tlačítko `Použít DNSSEC`, kde resolver v případě, že podporuje validaci DNSSEC ověří pravost záznamů. Poslední z přepínače je použití simulace rekurzivního dotazu.

Závěr

Neznalost problematiky s dotazováním DNS serverů a principem fungování DNSSEC mě stálo hodně zbytečného času. Kdybych si prvotně toto téma důkladněji nastudoval, ušetřil bych si práci s několika zbytečnými úpravami kódu. Ono vlastně těch úprav byla spousta. Z prvu jsem začal parsovat výstup z `dig +trace`, který také vystihuje rekurzivní dotaz. Ten jsem ale poté odstranil a nahradil uměle vytvořeným rekurzivním dotazem pomocí cyklu while. Předělávání kódu mi ostatně zabralo nejvíce času z celé práce. Další z problémů, na který jsem narazil, bylo získáváním informací o kontaktovaných serverech. Po nesplnění podmínky, které jsem zmínil výše, se server na několik hodin zablokoval. Musel jsem tedy získat informace z jiných zdrojů. Prvně jsem si myslel, že je chyba někde jinde, ale pak jsem si přečetl dokumentaci.

Další třešničkou na dortu byla samotná vizualizace v javascriptu. Jednou jsem viděl podobnou simulaci trasy v google mapách, a tak při pohledu na toto téma jsem si řekl "Jo, tak toto bych mohl zvládnout". Jenže moje ne příliš obsáhlé znalosti o javascriptu, mi způsobily, že jsem se do vizualizace trochu zamotal. Je pravda, že programovaní a kódování webových stránek mě vcelku baví, jQuery zde hraje významnou roli v dynamičnosti, jenže ve většině případů jsem se setkal pouze s reagováním na události a o nějakém vytváření algoritmů na simulaci cesty jsem neměl sebemenší tušení. Když se všechny tyhle aspekty sešly, způsobilo to, že jsem na práci pracoval enormní dobu. Na druhou stranu jsem obohatil mé znalosti o další nové. Jelikož jsem si k práci vytvořil určitý vztah a zaujalo mě celkově fungováni DNSSEC, chtěl bych na práci, pokud mi bude dovoleno, pokračovat a implementovat další funkcionality. Jako fanoušek Unixových systému bych si rád nakonfiguroval vlastní BIND server a vyzkoušel na něm ověřování podpisů pomocí DNSSEC. Na samotný závěr bych dodal, že jsem na sebe hrdý, že jsem celý projekt dotáhl do konce.

Conclusions

The lack of knowledge of DNS server queries and how the DNSSEC principle works cost me a lot of time. If I studied this topic deeper I would save a lot of time on working with some unnecessary code modifications. It cost actually a lot of time. Firstly I started parsing the output from `dig + trace` which also describes the recursive query. But then I replaced it with an artificially created recursive query using a while loop. Redoing the code took the most time out of my work. Another of the issues I encountered was getting information about contacted servers. After not meeting the condition I mentioned above, the server blocked for a few hours. So I had to look for information from other sources. First I thought there was a mistake somewhere else, but then I read the documentation.

Another incident was the javascript visualization itself. Once I saw a similar route simulation in google maps, when I looked at this topic I said "Yeah, so I could do this". But I am not very knowledgeable about javascript and that caused me to get a bit tangled up in the visualization. It is true that I am happy with programming and coding web pages. jQuery plays an important role in dynamics, but in most cases I only met with reaction to events. I had no idea about creating algorithms for simulation and path simulation. When all these aspects came together, I was spending enormous amounts of time on my work. On the other hand, I have enriched my knowledge with new ones. Since I have established a relationship with work and I am interested in the overall functioning of DNSSEC. I would like to continue working and implementing other functionalities if I am allowed. As an Unix system fan, I would like to configure my own BIND server and test own DNSSEC signatures on it. I am proud of myself for completing the project.

A Obsah přiloženého CD

Popis důležitých souborů potřebných pro bezproblémový běh aplikace. Důraznou pozornost věnujte prosím souboru `readme.txt` v adresáři `instructions`, kde je popis instrukcí potřebných pro bezproblémový provoz aplikace.

`instructions/`
: Tento adresář obsahuje soubor `readme.txt`, kde se nacházejí instrukce pro nasazení webové aplikace na webový server, včetně všech požadavků pro její bezproblémový provoz. Je zde také IP adresa, na které je aplikace nasazena pro účel testování. Dále se v adresáři `dig_install_windows` vyskytuje návod k instalaci dig programu na platformě Windows.

`src/`
: V tomto adresáři se nacházejí soubory k nahrátí na webový server.

`doc/`
: Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh. Vyskytují se zde i všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu.

`literature/`
: Seznam použité literatury

Bibliografie

[1] Jak funguje dns. Dostupné z http://www.jakfungujedns.cz/

[2] Co je to DNSSEC. Dostupné z https://www.dnssec.cz/

[3] Typy DNS záznamů. Dostupné z https://en.wikipedia.org/wiki/List_of_DNS_record_types

[4] Milan KERSLAGER, Analýza DNS programem dig. Dostupné z: https://www.pslib.cz/milan.kerslager/Analýza_DNS_ programem_dig

[5] James F. KUROSE, Keith W. ROSS. (2014). Počítačové sítě. Computer Press.

[6] A. KABELOVÁ, L. DOSTÁLEK : Velký průvodce protokoly TCP/IP a systémem DNS (5. vydání). Computer Press, 2008.

[7] William STALLINGS. (2016). Network Security Essentials: Applications and Standards (6th Edition). Pearson.

[8] Program dig. Dostupné z: https://ftp.isc.org/isc/bind/9.11.0a1/doc/arm/man.dig.html

[9] Google Maps javascript API. Dostupné z: https://developers.google.com/maps/documentation/javascript/tutorial

[10] IP-API; Získávání geografických informací. Dostupné z: http://ip-api.com

[11] ApiInfoDb; Získávání geografických informací. Dostupné z: https://ipinfodb.com

[12] David GRUDL; Nette - Rychlý a pohodlný vývoj webových aplikací v PHP. Dostupné z: https://doc.nette.org/cs/2.4/