VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

REGISTRAČNÍ DATABÁZE IP ADRES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

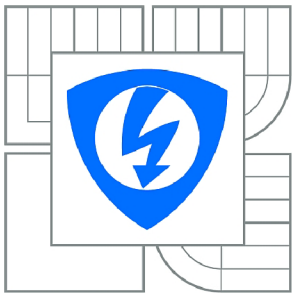
AUTOR PRÁCE
AUTHOR

Bc. LUKÁŠ SMRČKA

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

REGISTRAČNÍ DATABÁZE IP ADRES

REGISTRATION DATABASE FOR IP NODES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

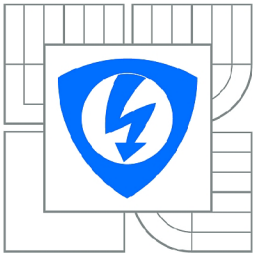
AUTOR PRÁCE — Bc. LUKÁŠ SMRČKA

AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. DAN KOMOSNÝ, Ph.D.

SUPERVISOR

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Bc. Lukáš Smrčka	***ID:***	74577
Ročník:	2	***Akademický rok:***	2014/2015

NÁZEV TÉMATU:

Registrační databáze IP adres

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte problematiku určení geografické polohy stanic pomocí registrační databáze RIPE NCC. Vytvořte seznam stanic se známou geografickou polohou. Pro tyto stanice zjistěte jejich polohu pomocí registrační databáze RIPE NCC. Vytvořte webovou stránku, která bude umožňovat zobrazení polohy pro zadané IP adresy.

DOPORUČENÁ LITERATURA:

[1] PUŽMANOVÁ, R. TCP/IP v kostce. 1. vyd. České Budějovice : Kopp, 2004. 607 s. ISBN 80-7232-236-2.
[2] COOPER, M. Advanced Bash-Scripting Guide. Lulu.com, 2010. ISBN: 978-1435752191.
[3] MUIR, J., OORSCHOT, P.: Internet geolocation: Evasion and counterevasion. ACM Computing Surveys (CSUR). ACM, 2009.

Termín zadání:	9.2.2015	***Termín odevzdání:***	26.5.2015

Vedoucí práce: doc. Ing. Dan Komosný, Ph.D.
Konzultanti diplomové práce:

doc. Ing. Jiří Mišurec, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

Tato práce je zaměřena na hledání fyzické polohy stanic pomocí pasivních geolokačních metod, zejména pomocí registrační databáze IP adres. První dvě části jsou věnovány teoretickému rozboru uvedené problematiky, následující dvě části se věnují vlastnímu řešení a analýze výsledků.

KLÍČOVÁ SLOVA

Geolokace, WHOIS, RIPE, GeoPing, ShortestPing, GeoIP, IP adresa

ABSTRACT

This thesis is focused on finding the physical location of stations by the passive geolocation techniques, particularly using the registration database of IP addresses. The first two part are focused on a theoretical analysis of this problem, the next two parts of this thesis deal with the solution of this problem and discussion of the results.

KEYWORDS

Geolocation, WHOIS, RIPE, GeoPing, ShortestPing, GeoIP, IP address

SMRČKA, Lukáš *Registrační databáze IP adres*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015. 50 s. Vedoucí práce byl doc. Ing. Dan Komosný, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Registrační databáze IP adres“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu doc. Ing. Danu Komosnému, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

(podpis autora)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic

http://www.six.feec.vutbr.cz

PODĚKOVÁNÍ

Výzkum popsaný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno . . . . . .

. . . . . .

(podpis autora)

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OP Výzkum a vývoj
pro inovace

OBSAH

SEZNAM OBRÁZKŮ

ÚVOD

V posledních letech je zaznamenáván velký rozmach Internetu, ať už z hlediska obsahu nebo z hlediska počtu a typu připojených zařízení. Je využíván k práci, zábavě, reklamě i vzdělávání. Pro efektivnější využívání jeho možností vyvstala potřeba některá data cílit na konkrétní skupiny uživatelů (stanic, zařízení...). Jednou z možností jak tohoto docílit, je účinná lokalizace geografické polohy stanice, tzv. geolokace.

Tato diplomová práce se v její první části zabývá problematikou určení geografické polohy stanice pomocí aktivních a zejména pak pasivních geolokačních metod. U jednotlivých typů metod jsou pak uvedeny jejich zástupci, u kterých je vysvětlen jejich princip geolokace. V další části práce je detailněji vysvětlena problematika registrační databáze IP adres, která je využita při realizaci metody pasivní geolokace stanic na základě IP adresy. Jedná se o databázi evropského RIR (Regional Internet Registry), která nese jméno RIPE Whois Database. U uvedené databáze je probrán její vznik, historie, struktura a způsoby dotazování se na tuto databázi. Třetí část této práce je věnována vlastní realizaci webového rozhraní, které má za úkol ověřit přesnost lokace stanice na základě informací, které jsou o stanici uvedeny v registrační databázi IP adres. Poslední část práce je věnována analýze výsledků lokace polohy stanic, kde jsou uvedeny výsledky lokace stanic, pro které byla za účelem porovnání přesnosti geolokačních metod zjištěna jejich skutečná poloha.

1 URČENÍ GEOGRAFICKÉ POLOHY STANIC

Internet přináší v dnešní době velké množství informací využitelných mnoha jeho uživateli. Pro zvýšení efektivity využití síťových prostředků z hlediska relevantnosti přenášených informací a rychlosti jejich přenosu je vhodné znát polohu stanice, u níž se uživatel dotazující se na informace, resp. data, nachází. Při znalosti polohy stanice je tedy také možné a žádané výstupy dotazů uživatele cíleně filtrovat a vracet uživateli informace s ohledem na jeho polohu. Polohu stanice je možno chápat jako polohu virtuální nebo polohu fyzickou.

Virtuální polohou stanice v síti se rozumí její umístění z pohledu topologie sítě. Pro určení virtuální polohy je potřebné provést měření parametrů sítě (např. měření hodnoty zpoždění mezi stanicemi (uzly) v síti - tzv. RTT - obousměrné zpoždění – Round Trip Time). Toto měření parametrů sítě však síť zatěžuje a proto se pro určení virtuální polohy v síti používají metody, které určí polohu bez tohoto měření nebo pomocí jen minimálního počtu měření. Jsou to metody využívající umělých souřadnicových systémů. Příkladem těchto metod je zejména metoda GNP (Global Network Positioning), metoda Lighthouses nebo také metoda Vivaldi. Díky znalosti virtuální polohy lze tedy snížit redundanci vysílaných dat a tím snížit zatížení sítě. Dále je díky znalosti virtuální polohy také možné vybrat trasu, která je pro přenos dat nejvhodnější a nejefektivnější (např. volba nejbližšího zdroje dat). Pro další zvýšení efektivity přenášených informací z hlediska obsahu však potřebujeme znát také polohu fyzickou.

Fyzickou polohou stanice v síti se rozumí její poloha vztažená k místu na Zemi. Hledání fyzické polohy objektu, v našem případě stanice, kde na základě různých informací o uživateli Internetu je určována jeho geografická poloha, je nazýváno geolokace. Geolokace k nalezení polohy objektu využívá různé techniky a standardním výstupem je adresa, kterou se myslí stát, město, ulice nebo PSČ. Díky určení geografické polohy uživatele Internetu je uživateli umožněno získávat relevantnější vyhledávané informace a snížit tak čas potřebný k jejich vyhledání.

Vezmeme-li v úvahu například obchodního zástupce nějaké firmy, který se v rámci jednání s klientem, resp. s budoucím zákazníkem neplánovaně zdrží déle než předpokládal, může si na Internetu vyhledat informaci o možnostech ubytování, stravování a případně bankomatech pro doplnění finanční hotovosti. Tomuto obchodnímu zástupci budou díky geolokaci zobrazeny informace s ohledem na jeho polohu, resp. s ohledem na polohu zařízení, ze kterého informace vyhledával. V tomto případě budou informace obsahovat např. přehled nejbližších hotelů, restaurací a bankomatů, ale mohou mu poskytovat i další údaje o institucích a úřadech v jeho okolí a informovat ho o aktuálním počasí v jeho lokalitě. Využití služeb geolokace je také ideálním prostředkem pro reklamní společnosti, které tak mohou uživatelům Internetu nabí-

zet lokálně zaměřenou reklamu na nejrůznější výrobky a služby. Služby geolokace mohou být také využity k povolení nebo zamítnutí přístupu k různým službám, případně k přesměrování na jinou cílovou službu, na základě informace o příslušnosti IP adresy stanice k nějakému státu. V základním pohledu na dělení geolokace rozlišujeme dvě metody, jsou to aktivní metody geolokace a pasivní metody geolokace [3] [4].

1.1 Metody aktivní geolokace

Aktivní metody geolokace jsou takové metody, u kterých je vyžadována jistá součinnost zařízení, pro něhož hledáme polohu. Nejčastěji je to měřením hodnoty zpoždění RTT, které je dáno součtem zpoždění na cestě od zdroje dat přes síťové uzly k cíli a zpět. Hodnota tohoto zpoždění je dána fyzickou polohou zdroje a cíle, počtem a kvalitou síťových uzlů, aktuálním zatížením sítě a mnoha dalšími faktory. Kvůli proměnlivosti jednotlivých faktorů v relativně krátkém čase je hodnota RTT také velice proměnlivá a je třeba pro získání relativně přesných údajů jednotlivá měření v rozumném počtu opakovat. Příkladem metod aktivní geolokace je např. metoda GeoPing a ShortestPing. Extrémním příkladem je ale také využití GPS (Global Positioning System) modulů v zařízeních, jež jsou tímto modulem vybaveny [3] [5].

1.1.1 Metoda GeoPing

Geolokační metoda GeoPing využívá k určení polohy stanice měření zpoždění vůči landmarkům. Landmarky jsou uzly v síti u nichž je známa jejich poloha a fungují tedy jako referenční uzly v síti. Pro stanici, u níž hledáme polohu, je změřeno zpoždění (RTT) ke všem landmarkům a vytvořen vektor zpoždění, který definuje jak blízko landmarky jsou. Stanice je mapována na nejpravděpodobnější landmark a souřadnice tohoto landmarku jsou použity jako odhadovaná poloha stanice. Pravděpodobnost je vypočítána na základě Eukleidovské vzdálenosti mezi vektory zpoždění.

Metoda GeoPing umožňuje rozšířit landmarky o sadu pasivních, u kterých se neprovádí měření zpoždění ke stanici. V takovémto případě je stanice mapována na aktivní nebo pasivní landmark. Z hlediska zatížení sítě měřením zpoždění je lepší využít mapovaní na pasivní landmark, protože nevede k dalšímu zatížení sítě [6].

1.1.2 Metoda ShortestPing

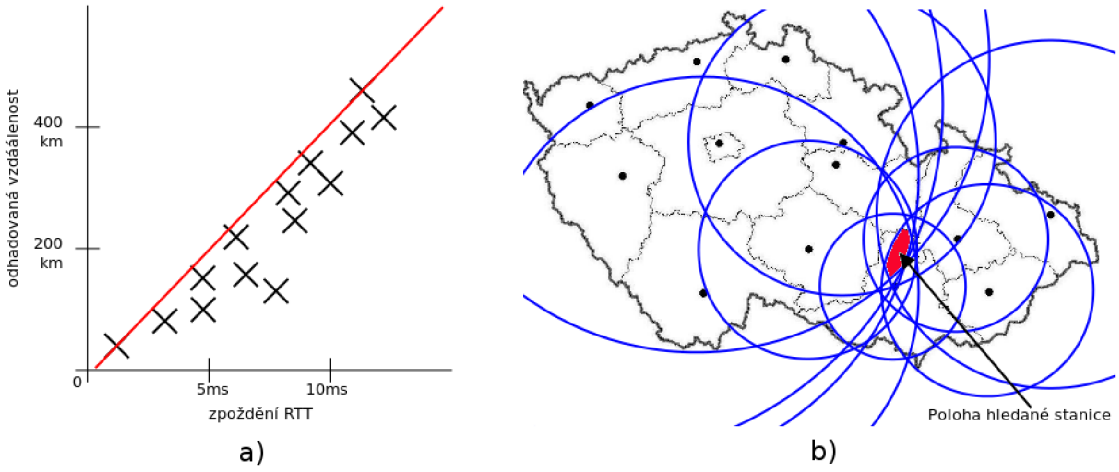
Metoda ShortestPing je jednou z nejjednodušších metod, založených na měření hodnoty zpoždění RTT. Každá stanice, pro kterou hledáme polohu, je mapována na lan-

dmark, ketrý je k ní z hlediska hodnoty RTT nejblíže. K tomu je třeba provést měření od stanice ke každému landmarku což představuje nechtěné zatížení sítě. Pro snížení zatížení sítě je možné využít síťové souřadnice (např. pomocí metody Vivaldi) a nejprve určit malé množství nejbližších landmarků a následně provést měření pouze vůči těmto nejbližším landmarkům [6].

1.1.3 Metoda Constraint-Based Geolocation

Metoda Constraint Based Geolocation (CBG) neprovádí mapování stanice na polohu landmarku, ale využívá pro geolokaci kombinaci hodnot zpoždění RTT z více landmarků s technikou podobnou triangulaci, čímž získá polohu landmarků ležících mezi měřenými landmarky[6].

Každý landmark proměří zpoždění mezi sebou a všemi ostatními landmarky. Podle naměřených hodnot zpoždění je pro tato data a závislost vzdálenosti na zpoždění stanovena přímka, tzv. bestline, která slouží k převodu hodnot zpoždění RTT na odhadované vzdálenosti (obr. 1.1 a). Poloha hledané stanice je pak předpokládána uvnitř kružnice, jejíž poloměr je dán odhadovanou vzdáleností se středem v daném landmarku. Metoda CBG pak kombinuje odhady vzdáleností všech landmarků, přičemž odhadovaná poloha stanice se nachází v těžišti oblasti vymezené průnikem všech kružnic (obr. 1.1 b). Testy ukázaly, že metoda CBG poskytuje přesnější odhady polohy hledané stanice než metoda GeoPing a geolokace založené na službě DNS[6].



Obr. 1.1: Metoda CBG: a) stanovení přímky tzv. bestline b) kombinace odhadů všech vzdáleností a určení polohy hledané stanice

1.1.4 Metoda Speed of Internet

Metoda Speed of Internet (SOI) je metoda geolokace, která na rozdíl od metody CBG používá jeden koeficient k přepočtu dat pro všechny landmarky.

Při přenosu dat optickými kabely jsou tato data přenášena rychlostí téměř přesně dvou třetin rychlosti světla ve vakuu[6]. Metoda SOI je založena na předpokladu, že geografická vzdálenost mezi stanicemi v Internetu je obvykle mnohem menší než je omezení způsobené rychlostí šíření světla, a také že vícecestné šíření dat, paketizační zpoždění a další zpoždění nezpůsobené šířením světla může pouze prodloužit čas a snížit efektivitu přenosu. Tyto fakta dovolují použít přísnější omezení než dvě třetiny rychlosti šíření světla ve vakuu. Geolokace pomocí metody SOI tedy generuje omezení, které pro převod času na vzdálenost používá koeficient o velikosti 4/9 rychlosti šíření světla ve vakuu[6]. Ve srovnání s metodou CBG je tato metoda mnohem jednodušší, protože nevyžaduje měření zpoždění RTT mezi landmarky a kalibraci. Mimo uvedený koeficient je tato metoda totožná s metodou geolokace CBG.

1.2 Metody pasivní geolokace

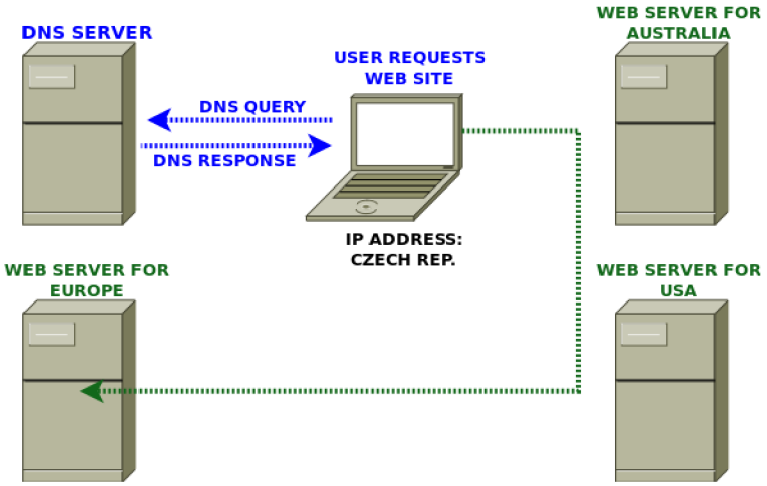
Pasivní metody geolokace jsou takové metody, u kterých není třeba součinnost se zařízením, pro něhož hledáme polohu. To znamená, že není třeba z tohoto zařízení proměřovat parametry sítě a nějakým způsobem dopočítávat polohu zařízení, ale vychází se zde z dostupných informací o zařízení, které jsou většinou zapsány v nějaké databázi (ať už veřejné či komerční - placené). Podle typu informace, na základě které se dotazujeme na dostupné informace o zařízení, případně ke které chceme využít dodatečný zdroj informací, můžeme tyto metody rozdělit na metody využívající jako zdrojovou informaci IP adresu zařízení, metody využívající jako zdrojovou informaci DNS záznamy a metody, které jako zdrojové informace využívají dodatečné informace o daném zařízení (WiFi,GSM) [5] [3] .

1.2.1 DNS Geolokace

Jednou z metod pasivní geolokace je určení polohy pomocí záznamů ve službě DNS (Domain Name Service). Údaje o poloze mohou být vložené přímo v systému DNS, kde jsou dostupné například nástroji `nslookup`. To ovšem představuje ruční vložení těchto informací do systému, kde jsou pak kromě A záznamů také záznamy typu LOC, které u stanice, resp. uzlu, uvádí souřadnice a nadmořskou výšku. Uvedený způsob je hodně pracný a proto se téměř nevyužívá[5].

Další možností, jak využít službu DNS ke geolokaci, je využít k tomu samu podstatu služby DNS. Ta pro snížení nároků na paměť uživatele zajišťuje překlad

doménových jmen na IP adresy, to znamená, že pokud službě DNS zašleme dotaz typu *www.seznam.cz* dostaneme odpověď *77.75.72.3*. Z odpovědi je zřejmé, že server *www.seznam.cz* má IP adresu *77.75.72.3*. Pokud bychom tedy použili reverzního překladu adresy, což znamená, že známe IP adresu a k ní bychom potřebovali dostat doménové jméno, z doménového jména bychom poté byli schopni určit polohu stanice (serveru) s přesností minimálně na úrovni státu. Tohoto je možno využít např. při filtrování obsahu webových stránek podle toho, ze kterého státu se uživatel k webovému serveru připojuje. Na obr. 1.2 je znázorněn příklad, ve kterém se uživatel chce připojit na server *www.example.com*. DNS server rozeznal na základě reverzního překladu jeho IP adresy, že se jedná o uživatele z České Republiky a v odpovědi mu odeslal IP adresu serveru *www.example.com* pro Evropu[8].



Obr. 1.2: Možné využití geolokace na základě DNS záznamů

1.2.2 Geolokace s využitím dodatečných zdrojů informací (WiFi, GSM)

Jedná se o metody, které předpokládají využití dodatečný zdrojů informací o daném zařízení. Zde se však nedá spolehnout na trvalou dostupnost těchto informací a také nelze tyto informace vždy použít, např. díky některým omezením, vyplývajících z legislativních norem. Jako dodatečných zdrojů informací lze například použít dostatečný počet přístupových bodů WiFi nebo základnových stanic GSM v dosahu zařízení a velikostí jejich signálů. Díky znalosti polohy přístupových bodů nebo základnových stanic a velikostí jejich signálů je možné dopočítat polohu zařízení. Zde je zcela zřejmý fakt, že pro dokončení výpočtu a určení polohy zařízení je třeba znát polohu přístupových bodů. K tomu je tedy nezbytné mít přístup k databázi, která

by tyto údaje obsahovala. Jednou z možností je databáze přístupových bodů WiFi, která je obsažena v GoogleMaps [5].

Tato metoda je ideální ukázkou toho, jak se mohou typy geolokací prolínat a toho, jak křehká může být hranice mezi těmito typy. Myslí se zde prolínání metod pasivní a aktivní geolokace. Vyjdeme-li z předpokladu, že aktivní geolokace klade na stanici nároky spojené s jistým proměřením okolní sítě, pak by metody geolokace s využitím dodatečných zdrojů informací mohly být zařazeny pod oběma typy.

1.2.3 Geolokace na základě IP adresy

Metody, které využívají jako zdrojovou informaci IP adresu zařízení, pro něhož hledáme polohu, patří mezi nejjednodušší a nejrozšířenější. Každé zařízení, které hodlá nějakým způsobem prostřednictvím Internetu komunikovat, totiž má nějakou adresu IP přidělenu.

IP adresa

O přidělování a registrování adres v globálním adresním prostoru se stará centrální správce - IANA (Internet Assigned Numbers Authority). Organizace IANA je součástí společnosti ICANN (Internet Corporation for Assigned Names and Numbers), internetové společnosti pro přiřazení jmen a čísel. IANA je odpovědná za údržbu některých klíčových prvků, díky kterým je umožněn hladký běh a funkčnost Internetu. Konkrétně přiděluje a udržuje jedinečné kódy a systémy číslování, užívané v technických standardech (protokolech), prostřednictvím kterých je řízen Internet. Aktivity společnosti IANA mohou být obecně rozděleny do tří kategorií[12]:

- doménová jména (Domain Names) - spravuje domény *.int* a *.arpa* a spravuje kořen služby DNS,
- adresy (Number Resources) - koordinuje globální zásoby IP adres a čísel autonomních systémů, které přiděluje regionálním registrům,
- registr protokolů (Protocol Assignments) - systémy číslování internetových protokolů, které spravuje ve spolupráci s normalizačními orgány.

IANA je členěna, podle území která spravuje, do pěti organizačních složek[5] [3]:

- RIPE NCC (Evropa a země bývalého Sovětského svazu),
- ARIN (Severní Amerika),
- LACNIC (Latinská Amerika),
- APNIC (Asie, Austrálie a země v Pacifiku),

- AfriNIC (Afrika).

Ve výše jmenovaných regionech dále působí organizace, prostřednictvím kterých je možné získat IP adresní prostor. Tyto organizace komunikují přímo s koncovým zákazníkem a jsou označovány jako *Local Internet Registry* (LIR). Jejich počet v jednotlivých zemích se pohybuje v řádu stovek. Příkladem jednoho z nich v ČR je VUT v Brně.

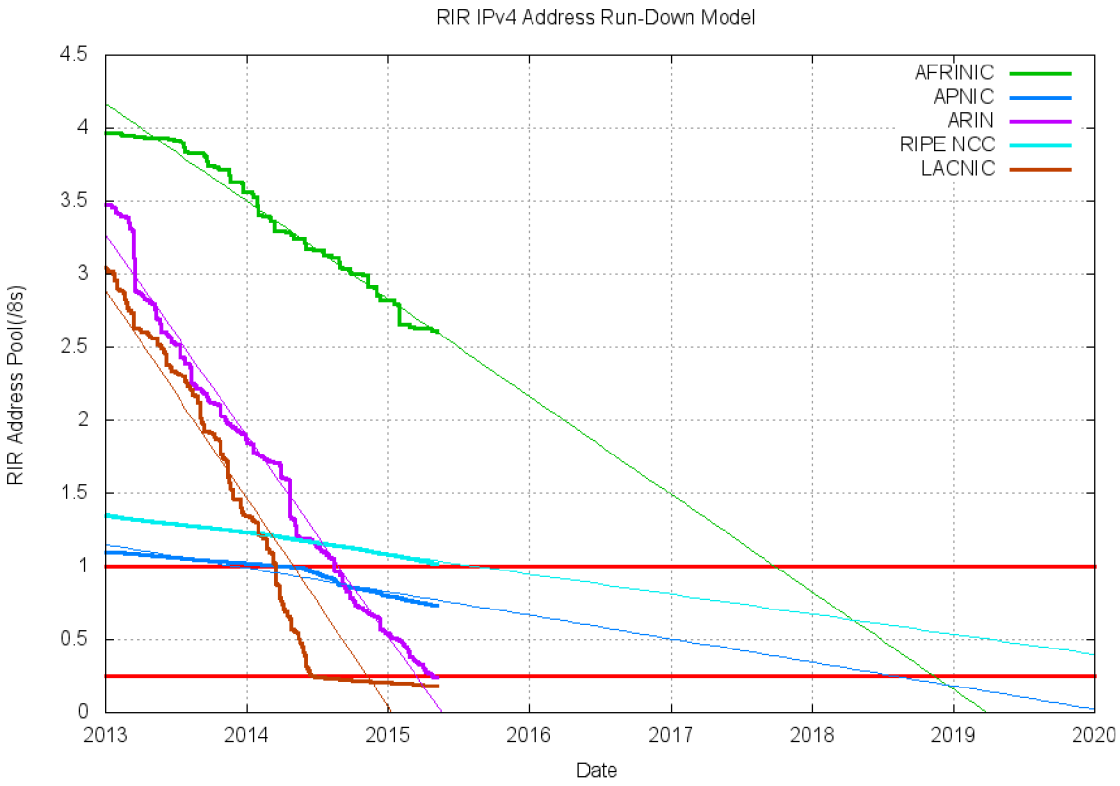
Přidělené IP adresy mohou být ve dvou formátech, a to ve verzi IPv4 a ve verzi IPv6. IP protokol verze 6 (IPv6) je nástupcem protokolu IPv4 a vznikl zejména kvůli předpokládanému vyčerpání adres koncových zařízení v Internetu, využívajících již zmiňovaný protokol IPv4. Vznik nového protokolu byl výsledkem snahy řešit některé problémy a nedokonalosti, které pomocí původního protokolu, zejména díky jeho vlastnostem, nemohly být efektivně řešeny. V souvislosti s novou verzí protokolu došlo také k nárůstu velikosti adresy z 32 na 128 bitů.

Základní problémy, které měl nový protokol vyřešit, byl neustále narůstající rozsah internetových směrovacích tabulek a nedostatečná velikost adresního prostoru[16]. Implementací techniky NAT (Network Adress Translation) a jejím následným velkým rozšířením došlo k zpomalení nárůstu adres koncových zařízení a k oddálení vyčerpání adres v globálním adresním prostoru. V současné době je však centrální zásoba adres u IANA vyčerpána a v blízké době dojde i k vyčerpání zásob jednotlivých regionálních registrů (RIR - Regional Internet Registries). Časový vývoj počtu zbývajících adres IPv4 a předpoklad vyčerpání zásob regionálních registrů RIR je možné vidět na obr. 1.3 .

Geolokace

Jak již bylo uvedeno v úvodu této podkapitoly, při geolokaci podle IP adresy se dotazujeme nějaké databáze, zda má informace o IP adrese zařízení pro které hledáme polohu. Těchto databází je velké množství a jsou buď veřejné, které se většinou vyznačují zcela neomezeným přístupem, nebo jsou komerční, kde je přístup k informacím obsažených v těchto databázích nějakým způsobem omezen. Oba typy databází se dále mohou lišit strukturou informací nebo také obsahem. Dá se předpokládat, že komerční databáze nabídnou větší podporu např. pravidelnou aktualizaci informací v nich obsažených. Mezi ty databáze, kterým je při hledání informací pro geolokaci zařízení jako parametr zadávána IP adresa, patří například databáze GeoIP nebo databáze RIPE NCC.

Databáze GeoIP poskytuje různé způsoby přístupu k informacím o konkrétní IP adrese, zejména přes webové rozhraní a prostřednictvím souborové databáze s komerčním nebo veřejným přístupem [7]. Je provozována a spravována společností



Obr. 1.3: Vývoj počtu zbývajících adres IPv4 dle jednotlivých RIR - převzato z[11].

MaxMind. Společnost MaxMind byla založena v roce 2002 a od té doby se stala vedoucím podnikem se schopnostmi získat a aplikovat znalosti a dovednosti v oblasti IP služeb a on-line nástrojů pro odhalování podvodů. Společnost sídlí ve Spojených státech amerických ve městě Waltham a je v soukromém vlastnictví.

V současné době existuje databáze ve verzi GeoIP2 a GeoIP Legacy, kde verze GeoIP2 poskytuje nejnovější verzi zmiňovaných služeb [7]. Asi největší změnou ve verzi GeoIP2 je zahrnutí lokalizovaných názvů míst.

Jak již bylo uvedeno výše, databáze GeoIP poskytuje různé způsoby přístupu k informacím uvedených v databázi. Webový přístup zahrnuje především využití různých rozhraní (API - Application Programming Interface). Oficiálně podporované společností MaxMind jsou API pro následující programovací jazyky - .NET (C#), Java, Perl, PHP a Phyton. Jako parametr by těmto API měla být předávána IP adresa ve verzi IPv4 nebo IPv6. Ve verzi IPv4 by měla být předána ve standardní formě, např. 147.229.2.90. Ve verzi IPv6 by měla být adresa předána ve formě řetězce, který odpovídá platnému formátu adresy IPv6. Třetí možností je místo IP adresy předat řetězec „me“, který je vhodný použít když stanice, která provádí dotaz, je za zařízením se službou NAT.

Databáze RIPE NCC je veřejnou databází a poskytuje informace, které re-

gistrují a aktualizují národní registrátoři IP adres. Databázi RIPE NCC obsahuje informace sloužící zejména k těmto účelům:

- k zajištění jedinečnosti registru internetových adres prostřednictvím evidence informací od registrátorů a správců těchto adres,
- k poskytování přesných registračních informací o internetových adresách pro různé provozní účely,
- k publikování směrovací politiky síťových operátorů,
- pro usnadnění koordinace mezi síťovými operátory,
- k vědeckému výzkumu síťových operací a topologií,
- k poskytování informací stranám, zúčastněných ve sporech o registraci internetových adres

Databáze RIPE NCC bude více přiblížena v následující kapitole.

2 DATABÁZE RIPE NCC

V dobách ARPANETu, kdy se registrací všech domén zabývala pouze jedna organizace jménem DARPA, existoval server jménem NICNAME/WHOIS a prostřednictvím sítě poskytoval adresářovou službu o internetových uživatelích. Byla to jedna z řady jmenných služeb, poskytovaných organizací NIC - Network Informatin Center. Server této služby byl server, u něhož komunikace byla typu dotaz - odpověď a byl dostupný napříč Internetem pomocí protokolu TCP programům bežícím na lokálních stanicích a poskytoval údaje o uživatelích, kteří byli registrováni v databázi NIC. Tyto údaje zahrnovaly například celé jméno uživatele, poštovní adresu, telefonní číslo a e-mailovou adresu. Server, společně s odpovídající databází, dokázal poskytnout také on-line vyhledání informací o jejich organizacích, síťových uzlech a přidělených počítačích[9].

Požadavkem organizace DCA (Defense Communications Agency) bylo, aby každý kdo měl adresář přístupný prostřednictvím Internetu na síti ARPANET nebo na síti MILNET, byl registrovaný v databázi NIC WHOIS. K registraci bylo zapotřebí zaslat elektronicky na adresu *REGISTRAR@SRI-NIC.ARPA* jméno, příjmení, poštovní adresu, PSČ, telefon a e-mailovou adresu.

K přístupu k informacím v databázi se používal program jménem NICNAME, který však byl mnohem známější pod názvem WHOIS. Tento program existoval, kromě jiných verzí, také ve verzi pro systém UNIX, která byla napsána v programovacím jazyce C. Po vyvolání programu mu byl v parametru předán hledaný dotaz, který byl takto hned poslán serveru. Server po zpracování dotazu a zaslání odpovědi spojení ukončil. Pro jasnější představu je zde uveden příklad odpovědi serveru NICNAME/WHOIS při vyhledání uživatele *fisher*[9]:

Command line: fischer
Response:
Fischer, Charles (CF17) fischer@UWISC (608) 262-1204
Fischer, Herman (HF) HFischer@USC-ECLB (818) 902-5139
Fischer, Jeffery H. (JHF1) FISCHER@LL-XN (617) 863-5500
Fischer, Kenneth (KF8) SAC.SIUBO@USC-ISIE (402) 294-5161
Fischer, Marty (MF28) MFISCHER@DCA-EMS (703) 437-2344
Fischer, Michael J. (MJF) FISCHER@YALE (203) 436-0744
Fischer, Nancy C. (NANCY) FISCHER@SRI-NIC (415) 859-2539
Fischer, Richard A. (RAF4) Fisher.Richa@LLL-MFE (415) 422-5032

Z odpovědi serveru je zřejmé, že server na daný dotaz vrátil výpis všech záznamů obsahujících klíčové slovo. Pro upřesnění dotazu bylo nutné zadat jako parametr ob-

sah kulatých závorek a vložit před něj vykřičník. Upřesněný dotaz pak vypadal následovně[9]:

Command line: !nancy
Response:
Fischer, Nancy C. (NANCY) FISCHER@SRI-NIC SRI International
Telecommunication Sciences Center
333 Ravenswood Avenue, EJ289
Menlo Park, California 94025
Phone: (415) 859-2539
MILNET TAC user

Na základě smlouvy s NSF (National Science Foundation) byla v roce 1993 založena organizace InterNIC, která měla za úkol komerční řízení registrace internetových domén. Později, v roce 1999, bylo vedení domén nejvyššího řádu přiděleno organizaci ICANN, kde byl použit model WHOIS.

Databáze WHOIS je veřejnou databází, obsahující veřejně přístupný soubor dat. Tyto data spravuje a zpřístupňuje RIR (Regional Internet Registry) daného regionu a jsou to zejména údaje z registru Internetových adres INR (Internet Number Registry), z registru směrování IRR (Internet Routing Registry) a reverzní údaje. Databáze také obsahuje některé neveřejné údaje, které jsou potřebné pro provoz databáze a registrů. Každý RIR provozuje a spravuje vlastní databázi WHOIS. Databáze evropského RIR se jmenuje RIPE Network Management Database, která je někdy také označována jako RIPE Whois Database. Oba názvy znamenají totéž a jsou zaměnitelné[10].

2.1 Objekty a atributy databáze

Jak již bylo nastíněno výše, databáze WHOIS obsahuje záznamy o alokaci a přidělení IP adresního prostoru, reverzní doménové záznamy, údaje o směrovací politice a kontaktní informace o osobách, které jsou zaregistrovány jako kontaktní osoby pro přidělený rozsah adres, který je využíván jejich organizací pro provoz a směrovaní sítí v jejich organizaci. RIR v databázi udržuje z globálního hlediska unikátní registr, prostřednictvím něj, přidělených IP adres a autonomních systémů. Tyto objekty v databázi jsou označeny jako „RIPE-REGISTRY-RESOURCE“. Ostatní objekty jsou označeny jako „RIPE-USER-RESOURCE“ a jsou udržovány osobami, uvedenými u jednotlivých záznamů (organizací) jako kontaktní osoby.

Databázový objekt je definován jako dvojice atributů ve formě prostého textu. Tyto atributy se dělí na povinné, volitelné a generované. Povinné atributy jsou uvedeny u každého databázového objektu. Volitelné atributy jsou uvedeny v případě, že jejich uvedení považuje tvůrce objektu za nezbytné nebo alespoň užitečné, případně je-li jejich uvedení vyžadováno obchodními pravidly. Generované atributy mohou být vytvořeny tvůrcem objektu, ale jejich hodnoty jsou kontrolovány a ověřovány databází. Pro zajištění prohledávání databáze pomocí dotazů jsou atributy mnoha způsoby indexovány a mohou být použity jako primární klíč, vyhledávací klíč, inverzní klíč nebo jejich kombinace. Vlastnosti atributu jsou určeny typem objektu, u kterého je atribut použit a pro každý typ objektu jsou uvedeny v jeho šabloně[10].

2.1.1 Databázové objekty

Databázový objekt je v souvislosti s touto databází chápán jako záznam, tedy jako dvojice atributů ve formě prostého textu. V databázi RIPE NCC existuje mnoho typů objektů, jejichž základní syntaxi definuje RPSL (Routing Policy Specification Language). Tyto objekty mohou obsahovat informace týkající se internetových zdrojů (internetový zdroj je zde chápán jako IP adresa, rozsah IP adres, případně doména), případně plní podpůrnou nebo administrativní funkci. Objekty se mohou odkazovat i na jiné objekty a pro získání všech informací o daném zdroji se musí tyto odkazy následovat. Díky tomu, že není využívána dědičnost, jsou tak některé informace reprodukovány z velkého množství objetků[10].

Objekty v databázi RIPE NCC se dají rozdělit do dvou kategorií, primární a sekundární. Primární objekty obsahují provozní údaje. Příkladem primárních objektů jsou objekty *inetnum* a *aut-num*, které obsahují informace o internetovém zdroji. Dále pak objekty *route* a *domain*, které obsahují informace o směrování a reverzním překladu zdroje. V současné době jsou databází RIPE NCC podporovány primární objekty uvedené v tab. 2.1.

Sekundární objekty databáze poskytují podpůrné a administrativní informace o primárních objektech. Příkladem sekundárních objektů databáze mohou být objekty *person* a *organisation*, kde objekt *person* poskytuje údaje o nějaké osobě, mající nějaký vztah k danému internetovému zdroji, zatímco objekt *organisation* poskytuje údaje o vlastníkovi daného internetového zdroje. Sekundární objekty databáze jsou uvedeny v tab. 2.2.

2.1.2 Vlastnosti databázových objektů

Všechny objekty v databázi RIPE NCC mají stejnou strukturu. Obsahují atributy, jejichž hodnoty jsou ve formě prostého textu, které však mohou mít různou formu.

Tab. 2.1: Primární objekty databáze RIPE NCC

Typ objektu	Popis objektu
aut-num	Nese informaci o čísle autonomního systému (AS). Pokud má „status: ASSIGNED“ pak je tento zdroj autoritativní, přidělený prostřednictvím RIPE NCC a je součástí registru internetových adres (INR). Pokud také nese informaci o směrovací politice autonomního systému, pak je součástí registru směrování(IRR)
domain	Reverzní doménové informace. Změny provedené v tomto objektu databáze jsou promítány do zónových souborů služby DNS.
inet6num	Alokace a přiřazení IPv6 adresního prostoru.
inetnum	Alokace a přiřazení IPv4 adresního prostoru.
route	IPv4 trasa inzerovaná na Internetu.
route6	IPv6 trasa inzerovaná na Internetu.
as-set	Sada **aut-num** objektů.
filter-set	Sada tras vyhovujících pro nastavení filtru.
inet-rtr	Směrovač internetu.
peering-set	Set of peerings.
route-set	Sada tras.
rtr-set	Sada směrovačů.

Někdy bývají atributy označovány jako „klíče“. Hodnoty atributů, stejně jako veškerý obsah databáze, jsou v současné době ve znakové sadě Latin-1, která je však pomalu nahrazována znakovou sadou UTF-8 [10]. Jakákoli jiná znaková sada než Latin-1 může v současné době způsobit problémy s funkčností databáze. Pro lepší čitelnost mohou být atributy a jejich hodnoty od sebe odděleny mezerou. V okamžiku, kdy je objekt ukládán v databázi, je zároveň jako jedna položka uložen úplný text poskytnutý uživatelem. Pro zvýšení rychlosti a funkčnosti databáze jsou z tohoto textu extrahovány a ukládány části objektů, které jsou ukládány v samostatných tabulkách a jsou indexovány generovanými metadaty.

Data poskytnutá uživatelem jsou upravena podle následujících pravidel:

- přidání (pokud chybí) datumu u atributu „změněno“;
- provedení změny nebo generování hodnoty u atributů, které se generují;
- u objektu databáze **aut-num** přidán chybějící atribut „status“ s generovanou hodnotou;
- nastavení hodnoty atributu „status“ na „LEGACY“ u objektů, které jsou více specifické;

Tab. 2.2: Sekundární objekty databáze RIPE NCC

Typ objektu	Popis objektu
as-block	Delegace rozsahu čísel autonomních systémů (AS) daného RIR nebo označeno jako rezervované.
irt	Kontaktní a ověřovací informace o Computer Security Incident Response Team (CSIRT)
key-cert	Veřejný certifikát klíče, který je uložen na serveru a je využíván objektem *mntner* pro ověřování při provádění aktualizace.
mntner	Informace o ověřování potřebné k povolení vytvoření, odstranění nebo úpravy objektů chráněných prostřednictvím *mntner*.
organisation	Informace o organizaci, která je vlastníkem nějakých internetových zdrojů (adres).
person	Technické, administrativní a DNS zónové kontakty. Někdy je využíváno k zobrazení koncového uživatele internetového zdroje, který se nepodílí na správě internetových zdrojů (adres). Obsahuje osobní informace.
poem	Vtipný text k udržení dobré nálady síťových inženýrů.
poetic-form	Typ humoru pro objekt *poem*.
role	Popisuje roli, kterou daný člověk vykonává (technická, administrativní, DNS zóna nebo tzv. abuse kontakt). Měly by být uvedeny pouze údaje, které se vztahují k dané organizaci.

- změna názvů všech atributů na malá písmena;
- převedení hodnoty zdroje na velká písmena,
- odebrání koncové tečky u klíčů reverzních domén;
- odstranění tabelátorů a přebytečných mezer u hodnot objektů **inetnum** a **inet6num**;
- odstranění úvodní nuly u objektu **inetnum**;
- převedení hodnoty prefixu objektu **inetnum** na rozsah;
- převedení hodnot u objektu **inet6num** na kanonický formát;
- odstranění připomínek u primárních objektů;
- spojení rozdělených hodnot u primárních objektů.

V některých případech je po takovéto úpravě přidána do potvrzení, které je zasíláno zpět k uživateli provádějícímu aktualizaci, informační zpráva.

Příklad informační zprávy:

****Info: Value 193.in-addr.arpa. converted to 193.in-addr.arpa*

2.1.3 Atributy v databázových objektech

První atribut objektu databáze musí mít stejné jméno (název) jako typ objektu, protože je tímto určen typ objektu. Název atributu musí být složen z alfanumerických znaků a případně pomlček, jiné znaky nejsou povoleny a začátek názvu atributu musí být na pozici nultého sloupce (písmene). Každý název atributu musí být ukončen dvojtečkou (:) před níž nesmí být žádná mezera, to znamená že název musí být bezprostředně následován dvojtečkou. Velká a malá písmena nejsou rozlišována, protože databáze stejně převede všechny znaky názvu atributu na malá písmena.

Ostatní atributy mohou být v libovolném pořadí, ale uživatelé většinou dodržují pořadí dané šablonami objektů, kde jsou pro každý typ objektu definovány možné atributy.

Každá instance objektu je jednoznačně definována primárním klíčem. Primární klíč je představován u většiny typů objektů hodnotou prvního atributu. V ojedinělých případech však může být představován hodnotou jiného atributu nebo může být složený z hodnot více atributů. V rámci jednoho typu objektu musí být primární klíč jedinečný. U rozdílných typů objektu se může stát, že je primární klíč stejný [10]. Pro příklad - „FRED“ je platný název objektu *mntner* a zároveň je to platná hodnota atributu *nic-hdl:*, která může být primárním klíčem objektů *person* nebo *role*.

Každý název atributu musí začínat na samostatném řádku, prázdný řádek značí konec objektu. Úplně prázdný řádek ve středu objektu však do tohoto pravidla zahrnout nelze, protože z technického hlediska jsou za konec objektu považovány dva po sobě jdoucí znaky nového řádku, tedy \n\n.

Pokud je atribut součástí objektu, musí mít nějakou hodnotu (s výjimkou volného textu)[10]. To se týká i generovaných atributů, které jsou součástí objektu, ty musí mít platnou hodnotu. Databáze může tyto hodnoty měnit, avšak v případě chybějící hodnoty je vyvolána chyba syntaxe. Hodnota atributu může začínat bezprostředně za dvojtečkou, která ukončuje název atributu, nebo může být pro lepší přehled za dvojtečku vložena mezera případně tabelátor. Hodnota atributu může obsahovat některá předem definovaná klíčová slova, odkazy na jiné objekty nebo volný text. Odkazování na jiné objekty je řešeno prostřednictvím hodnot primárních klíčů a takovýto odkaz i klíč mají přesně definovanou syntaxi. V případě odkazování se na objekt v databázi který neexistuje, skončí aktualizace chybou a je zobrazena chybová zpráva. Pro volný text není stanovena žádná syntaxe, ale může obsahovat pouze znaky ze znakové sady Latin-1.

2.2 Způsoby zadávání dotazů

Na informace z databáze RIPE NCC se lze dotazovat čtyřmi způsoby[10]:

- použitím telnetu na port 43 serveru např. whois.ripe.net,
- využitím REST API, bežícím na serveru rest.db.ripe.net,
- použitím webového rozhraní na webové stránce RIR ... např.`http://www.ripe.net/whois`,
- použitím *whois* klienta, který využívá *whois* protokol.

Použití programu telnet není z bezpečnostního hlediska příliš vhodné. V úvahu tedy připadají dotazy s využitím webového rozhraní, API a klienta *whois*. Poslední způsob je možné realizovat, jak již bylo uvedeno v úvodu této kapitoly, využitím programu *WHOIS*. Ten se na UNIXových systémech spouští příkazem `whois` a jako parametr se mu zadává např. IP adresa serveru, pro něhož hledáme informace v databázi. Výstup příkazu `whois` s parametrem *147.229.2.90* je zobrazen na následujících řádcích (zapoznámkované a prázdné řádky jsou vynechány):

NetRange: 147.228.0.0 - 147.237.255.255
CIDR: 147.232.0.0/14, 147.228.0.0/14, 147.236.0.0/15
NetName: RIPE-ERX-147-228-0-0
NetHandle: NET-147-228-0-0-1
Parent: NET147 (NET-147-0-0-0-0)
NetType: Early Registrations, Transferred to RIPE NCC
OriginAS:
Organization: RIPE Network Coordination Centre (RIPE)
RegDate: 2003-10-08
Updated: 2003-10-08
Comment: These addresses have been further assigned to users in
Comment: the RIPE NCC region. Contact information can be found in
Comment: the RIPE database at http://www.ripe.net/whois
Ref: http://whois.arin.net/rest/net/NET-147-228-0-0-1
OrgName: RIPE Network Coordination Centre
OrgId: RIPE
Address: P.O. Box 10096
City: Amsterdam
StateProv:
PostalCode: 1001EB
Country: NL

RegDate:
Updated: 2013-07-29
Ref: http://whois.arin.net/rest/org/RIPE
ReferralServer: whois://whois.ripe.net:43
OrgAbuseHandle: ABUSE3850-ARIN
OrgAbuseName: Abuse Contact
OrgAbusePhone: +31205354444
OrgAbuseEmail: abuse@ripe.net
OrgAbuseRef: http://whois.arin.net/rest/poc/ABUSE3850-ARIN
OrgTechHandle: RNO29-ARIN
OrgTechName: RIPE NCC Operations
OrgTechPhone: +31 20 535 4444
OrgTechEmail: hostmaster@ripe.net
OrgTechRef: http://whois.arin.net/rest/poc/RNO29-ARIN
Nalezen odkaz na whois.ripe.net:43.
inetnum: 147.229.0.0 - 147.229.254.255
netname: VUTBRNET
descr: Brno University of Technology
country: CZ
admin-c: TP4960-RIPE
tech-c: TP4960-RIPE
status: ASSIGNED PA
mnt-by: VUTBR-MNT
person: Backbone Admins
address: Brno University of Technology
address: Antoninska 1
address: 601 90 Brno
address: The Czech Republic
phone: +420 541145453
phone: +420 723047787
nic-hdl: TP4960-RIPE
mnt-by: VUT-BATCH-MNT
mnt-by: VUTBR-MNT
route: 147.229.0.0/17
descr: VUTBR-NET1
origin: AS197451
mnt-by: VUTBR-MNT

Z výpisu je vidět, že může být značně obsáhlý a proto je vhodné dotaz nějakým způsobem blíže specifikovat. Při zadávání dotazu lze pro bližší specifikaci dotazu využít určité parametry, tzv. „flags“. Tyto parametry lze využít přímo u dotazů zadávaných prostřednictvím příkazového řádku (*whois klienta*) a při dotazech zadávaných prostřednictvím API. U dotazů zadávaných s využitím webového rozhraní lze některé parametry zadat pomocí zatržení příslušného políčka, jiné mohou být zadány jako součást řetězce daného dotazu. U dotazů pomocí webového rozhraní však také některé parametry nedávají smysl a proto jsou ignorovány, jako například parametr „-k“, který slouží k nastavení trvalého připojení.

Jak již bylo uvedeno v předchozí podkapitole, všechny objekty v databázi mají primární klíč. Zadání dotazu s využitím primárního klíče k vyhledávání v databázi RIPE NCC je nejzákladnějším typem dotazu. Téměř ve všech případech je u takového dotazu, kdy má objekt databáze primární klíč, který přesně odpovídá argumentu dotazu, vrácena odpověď s jedním objektem (za předpokladu, že v dotazu nebylo vybráno zahrnutí přímo odkazovaných objektů). V případě, kdy bude argumentu dotazu odpovídat pouze část primárního klíče, nebude v odpovědi vrácen žádný objekt [10]. Jako příklad může být uveden dotaz s argumentem „aardvark-mnt“ na který je v odpovědi vrácen *mntner* objekt, který má uvedený argument jako primární klíč. Kdyby však u tohoto dotazu bylo v argumentu uvedeno pouze „aardv“, uvedený objekt by jako odpověď vrácen nebyl. Další možností je zadání argumentu „aardvark“. V takovém dotazu by jako odpověď nebyl vrácen *mntner* objekt, ale jakýkoliv objekt obsahující slovo „aardvark“ ve vyhledávacím klíči (lookup keys), které jsou používány u mnoha objektů databáze při dotazech vyhledávacího typu.

U primárních klíčů je tedy potřeba pro vyhledání konkrétního objektu přesná shoda argumentu s hodnotou klíče. U vyhledávacího klíče pak k vyhledání jakéhokoli objektu obsahujícího argument dotazu stačí, když je argument dotazu shodný s částí hodnoty vyhledávacího klíče.

V některých případech, kdy jsou hodnoty primárního klíče hierarchicky uspořádány, nemusí vrácený objekt přesně odpovídat argumentu dotazu [10]. Příkladem takového dotazu může být dotaz s argumentem „147.229.2.90“, kde v odpovědi na tento dotaz je vrácen objekt *inetnum* s hodnotou primárního klíče „147.229.0.0 - 147.229.254.255“.

Databáze RIPE NCC poskytuje informace o IP sítích, alokovaných uvnitř regionu spravovaného prostřednictvím RIPE NCC. Informace o těchto sítích jsou nejčastěji uloženy v objektech *inetnum*, *inet6num*, *route* a *route6*, přičemž tyto objekty obsahují informace o jednotlivých IP adresách nebo o rozsahu adres.

2.3 Filtrování výstupu dotazu

Filtrování výstupu dotazu z databáze RIPE NCC je zavedeno z důvodu ochrany kontaktních údajů před nechtěným zveřejněním, zejména pak e-mailové adresy, kde je filtrováním výstupu dotazu ušetřen čas uživatele a zobrazen pouze tzv. abuse kontakt. Vyhledává-li totiž uživatel kontaktní informace, někdy se mu podaří získat všechny e-mailové adresy, nalezené ve všech objektech vypsaných daným dotazem. Tento výpis však může často obsahovat e-mailové adresy osob, které nejsou odpovědné za vyřizování žádostí spojených s danou doménou. Pro řešení tohoto problému je tedy zavedeno filtrování, které standardně některé atributy obsahující e-mailové adresy filtruje. Jedinou výjimkou, která není nikdy filtrována, je atribut „abuse-mailbox:“[10]. Pokud je nějaký atribut filtrován, je ve výstupu dotazu přidána poznámka o jeho filtrování - *„Filtered“*.

3 REALIZACE WEBOVÉHO ROZHRANÍ

Jak bylo řečeno v předchozí kapitole, výstup dotazu na databázi RIPE NCC může být poměrně rozsáhlý a může obsahovat mnoho údajů, ať už jsou to údaje o provozovateli databázového serveru nebo o provozovateli sítě, ve které se nachází hledaná adresa. Údaje o provozovateli sítě pak obsahují zejména rozsah přidělených síťových adres, název provozovatele, údaje o směrování sítě, kontaktní údaje atd. Jako kontaktní údaje pak jsou zde uvedeny telefonní čísla, e-mailové adresy, příp. jména kontaktních osob a adresy. Adresy bývají nejčastěji zapsány v atributu databázového objektu, který nese název *address*. V případě, kdy by uživatelé provádějící aktualizaci jednotlivých objektů databáze dodrželi zápis adresy do atributu databázového objektu *address*, bylo by poměrně snadné provést geolokaci stanice načtením hodnot atributů *address* s následným získáním souřadnic takovéto adresy.

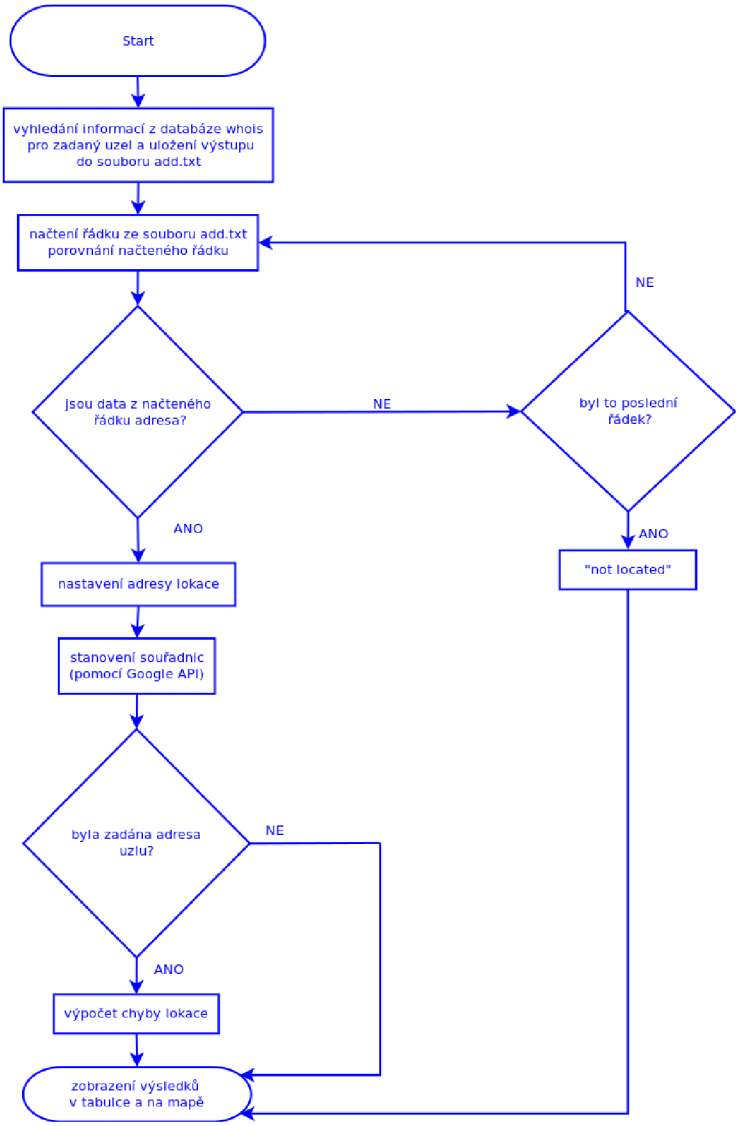
Problémem této databáze je však jistá benevolence zápisu údajů o provozovateli sítě do jednotlivých polí. V příkladu výpisu dotazu na databázi RIPE NCC pomocí klienta `whois` v předchozí kapitole, bylo možné vidět, že adresa provozovatele sítě VUTBRNET je *Antonínská 1, 601 90 Brno* a že je zcela správně zapsána v atributu *address*. Může se však stát, že adresa v atributu *address* zapsána nebude a bude zapsána např. v atributu *descr*, případně některém jiném. Uvedený problém byl motivací při realizaci webové části této diplomové práce.

Webová část této diplomové práce má tedy za úkol získat informace z databáze RIPE NCC, tyto informace zpracovat s cílem získání polohy hledané stanice (uzlu) a tuto polohu zobrazit na mapě. Dále pokud byly zadány údaje o skutečné poloze, tyto údaje porovnat a provést výpočet odchylky (vzdálenosti) nalezené a zadané polohy stanice. Celá tato část je umístěna na fakultním serveru[1] a je tvořena především soubory `index.php`, `search.php` a `compare.php` a podpůrnými soubory, které jsou určeny pro porovnání textových řetězců při určení adresy (`CZ.txt`, `AT.txt`, `DE.txt`, `PL.txt` a `SK.txt`). Podpůrné soubory pochází z geografické databáze GeoNames, která obsahuje celosvětové informace o jednotlivých místech (např. názvy měst, obcí, ...) v různých jazycích s jejich souřadnicemi a nadmořskou výškou [13]. Tyto soubory jsou pravidelně aktualizovány a jsou zdarma ke stažení přímo na stránkách uvedené databáze. Kromě podpůrných souborů jsou pro porovnání textových řetězců vyvářeny také pomocné textové soubory, které jsou aplikací využity při operacích s textovými řetězci.

[1] http://geolocation.utko.feec.vutbr.cz/xsmrck00/

3.1 Popis činnosti aplikace

Aplikaci je možno rozdělit do několika částí. První částí je zadání požadovaných dat do jednotlivých polí na úvodní stránce aplikace (viz obr. 3.2), odeslání dotazu databázi RIPE NCC a následný příjem a zpracování odpovědi na tento dotaz. V druhé části jsou data získaná v prvním kroku porovnávána vůči podpůrným souborům s cílem určit, zda je či není daný řetězec adresou. V další části je pak provedeno pro takto získanou adresu vyhledání souřadnic a případně provedení výpočtu chyby lokace. V poslední části aplikace je provedeno vykreslení nalezených dat do tabulky a mapy nebo jsou tato data zformátována do JSON formátu pro případné využití jinou aplikací. Celý proces realizovaný aplikací je znázorněn ve vývojovém diagramu aplikace na obr. 3.1.



Obr. 3.1: Vývojový diagram zhotovené aplikace

Cílem první části je tedy získání odpovědi na dotaz zaslaný databázi RIPE NCC. Jednotlivé dotazy obsahují data, zadaná do formulářového pole následovaného po textu „Enter the IPv4 address:“ na úvodní stránce aplikace (viz obr. 3.2). Z těchto dat je sestaven dotaz, který je databázi odesílán jako URL (Uniform Resource Locator) řetězec po odeslání formuláře souboru `index.php`. V tomto řetězci je specifikováno v jakém formátu bude odpověď na daný dotaz a dále je prostřednictvím tohoto řetězce zasílán primární klíč, který, jak už bylo uvedeno v předchozí kapitole, slouží pro vyhledání konkrétního objektu databáze. V tomto případě je primárním klíčem IP adresa stanice pro kterou provádíme geolokaci, tedy pro kterou hledáme její geografickou polohu. V případě, kdy je místo IP adresy do formulářového pole zapsán název domény nebo doménové jméno stanice, je nejprve proveden převod tohoto doménového názvu na adresu IP pomocí programu `nslookup`. Adresa IP je pak zaslána jako součást URL řetězce v dotazu na databázi a je tedy primárním klíčem.

Odpověď na dotaz, zaslaný databázi prostřednictvím URL řetězce, je uložena do souboru `json.txt`. Po uložení odpovědi je tento soubor následně zpracován filtrem `awk`, čímž jsou získány hodnoty jednotlivých atributů objektů databáze, které jsou následně uloženy do souboru `add.txt`. URL řetězec, kterým je prováděn výše uvedený dotaz a příkaz, který s využitím filtru `awk` provádí filtraci hodnot atributů objektů databáze, je vypsán ve výpisu kódu 3.1.

Obr. 3.2: Úvodní stránka zhotovené aplikace

```
1 $wget = "wget –O json.txt http://rest.db.ripe.net/search.json?query-
    string=$findip"; //Sestaveni prikazu pro odeslani dotazu a~ziskani
    odpovedi
2 $grep_json = "grep value json.txt | awk –F: '{ print $2 }' > add.txt";
    //Sestaveni prikazu pro filtraci hodnot atributu objektu databaze
3 echo shell_exec ("$wget"); // Provedeni prikazu $wget
4 echo shell_exec ("$grep_json"); //Provedeni prikazu k~filtraci hodnot
```

Výpis kódu 3.1: `search.php` - sestavení dotazu; jeho odeslání a filtrace odpovědi

V druhé části je, jak již bylo nastíněno v předchozím textu, prováděno porovnávání textových řetězců s podpůrnými soubory za účelem vyhledání adresy. Z uvedeného vyplývá skutečnost, že aplikace je při geolokaci limitována těmito soubory. Uvedené podpůrné soubory totiž představují seznamy měst a obcí jednotlivých států. Pro potřeby této práce je tedy aplikace schopna provádět geolokaci stanice zemí České Republiky, Slovenska, Polska, Německa a Rakouska. Rozšíření o další země je možné, je však třeba aplikovat do vyhledávacího algoritmu jistá hlediska spojená s různorodostí názvů obcí a měst v jednotlivých zemích.

Porovnávání řetězců tedy předchází určení země (státu), ve kterém se stanice nachází respektive jaká země je pro hodnotu primárního klíče, použitého v dotazu, registrována. Společně s určením země (státu) je také určen podpůrný soubor, který je použit při porovnávání řetězců. V případě geolokace stanice z jiné země, než země která byla uvedena v předchozím odstavci, jsou hodnoty hledaných souřadnic nastaveny na „DB error“. Příkaz, kterým je v programovacím jazyku PHP realizováno určení země, je zobrazen v následujícím výpisu kódu 3.2. Z výpisu je možné vidět, že k určení státu jsou použita data získaná z odpovědi databáze na dotaz s primárním klíčem, ze kterých je opět s využitím filtru `awk` získána hodnota atributu databáze

- country, tzv. *country code*.

```
1 $cc =<<<qqq
2 cat json.txt | awk 'BEGIN { RS = "{"; FS = "," } /country/ {print $2}'
    | awk 'BEGIN { FS = ":" } {print $2}' > country.txt
3 qqq; //Filtrovani hodnoty atributu "country" a~jeji ulozeni do souboru
    country.txt
4 echo shell_exec ("$cc"); //Provedeni prikazu $cc
```

Výpis kódu 3.2: `search.php` - zjištění hodnoty atributu *country* tzv. *country code*

Po určení země a tedy i souboru pro porovnání řetězců následuje porovnávání textových řetězců, které jsou cyklicky načítány řádek po řádku ze souboru `add.txt`, resp. ze souboru `add_out.txt`, který se od souboru `add.txt` liší inverzním pořadí řádků. Inverzní pořadí se ukázalo jako vhodné pro zrychlení procházení textových řetězců, protože ve většině případů se adresa nachází ve druhé polovině seznamu hodnot, vyfiltrovaných z odpovědi dotazu na databázi RIPE NCC. Porovnávání probíhá tak dlouho, dokud není nalezen řetězec, který při porovnání vyhověl a tedy je adresou. Vzhledem k tomu, že data vůči kterým jsou řetězce porovnávány jsou v podstatě seznamy měst a obcí, při porovnání vyhoví řetězec, který odpovídá nějakému městu nebo obci. Druhou situací, kdy dojde k ukončení porovnávání textových řetězců, je dosažení konce souboru.

Každý ověřovaný řetězec je ještě před porovnáváním „rozřezán“ na jednotlivá

```
1  $grep = <<<qqq
2  cat -v -e -t $country.txt | grep -c -w 'I$parse1' > grep.txt
3  qqq; //Sestaveni prikazu pro porovnani textovych retezcu
4  echo shell_exec ("$grep"); //Provedeni prikazu $grep (porovnani)
5  $readoption = fopen("./grep.txt", "r"); //Otevreni souboru s~vysledky
       porovani
6  $option=fgets($readoption); //Nacteni prvniho radku porovnani
7  $trimmedoption = trim($option); //Orezani prazdneho mista kolem
       vysledku
8  fclose("./grep.txt"); //Zavreni souboru s~vysledky porovnani
9  if(strcasecmp("0", $trimmedoption) == 0) //Jestlize je vysledek shodny
       s~cislem 0
10   {$city = "not found"; //...nastav mesto nenalezeno
```

Výpis kódu 3.3: `compare.php` - porovnání řetězců

slova, která jsou postupně porovnávána se souborem obsahujícím seznam měst daného státu, čímž je ověřováno, zda je dané slovo město (obec) daného státu či nikoliv. Pro případ, že by se jednalo o město, které obsahuje v názvu více slov, je v případě pozitivního výsledku porovnání provedeno ještě porovnání řetězce složeného ze slova s pozitivním výsledkem a následujícího slova. Pokud je i toto porovnání pozitivní, je provedeno další porovnání slov s pozitivním výsledkem a následujícím slovem. V případě, že už první výsledek není pozitivní, je toto slovo nastavením proměnné *city* na hodnotu *not found* z porovnávání vynecháno a v porovnávání je pokračováno s následujícím slovem, u kterého je v případě pozitivního výsledku porovnání opět prováděno porovnání řetězce složeného z pozitivního slova a slova následujícího dle výše popsaného postupu. Samotné porovnání končí v okamžiku, kdy proměnná *city* obsahuje jinou hodnotu než hodnotu *not found*.

Na výpisu kódu 3.3 je zobrazena část zdrojového kódu, kterou je realizováno první porovnání řetězců. První tři řádky obsahují sestavení linuxového příkazu v programovacím jazyku PHP a tento příkaz je ve čtvrtém řádku proveden. Z uvedeného příkazu je zřejmý způsob porovnávání řetězců. Nejprve je pomocí příkazu `cat` provedeno vypsání souboru, který je vybrán pro porovnání. Přepínače u tohoto příkazu jsou použity pro zobrazení některých netisknutelných znaků při výpisu souboru, ze kterých je zejména využit znak „I“ značící znak tabelátoru. Výstup příkazu `cat` je pak přiveden na vstup příkazu `grep`, který slouží k filtraci vstupních dat dle zadaných parametrů. Zde je jako parametr uveden porovnávaný řetězec, před kterým je uveden znak „I“, to znamená, že před porovnávaným řetězcem je navíc tabelátor. V podpůrném souboru jsou totiž jednotlivé položky od sebe odděleny tabelátory a proto lze předpokládat, že jméno města bude následovat bezprostředně po tomto znaku. Přepínače u příkazu `grep` slouží k tomu, aby byly v podpůrném souboru

hledány řetězce s úplnou shodou (pouze celá slova) (přepínač *-w*) bez ohledu na velikost písmen (přepínač *-i*) a aby bylo místo výpisu jednotlivých řádků, ve kterých byla nalezena shoda, vypsáno číslo, které představuje počet těchto řádků. Toto číslo je pak uloženo do souboru `grep.txt` odkud je pak načteno do proměnné *trimmedoption*, pro kterou je následně porovnáváno zda je toto číslo rovno číslu nula nebo je rozdílné od nuly. První případ, kdy je číslo rovno číslu nula, značí takovou situaci ve které porovnávaný řetězec porovnávání nevyhověl a tedy se nejedná o název města (obce). Druhý případ znamená, že porovnávaný řetězec v podpůrném souboru existuje a tedy je daný řetězec názvem nebo alespoň částí názvu města (obce) a dále pak pokračuje porovnávání dle postupu, popsaného v předchozím odstavci.

V další části aplikace je pro adresu, nalezenou v předchozí části aplikace, provedeno určení souřadnic GPS. Určení souřadnic je pro tuto adresu realizováno prostřednictvím Google Geocoding API[14], což je proces, ve kterém jsou pro zadanou adresu získány souřadnice GPS (viz výpis zdrojového kódu 3.4). Zde je v prvním řádku výpisu uveden URL řetězec, odesílaný jako dotaz serverům Google. Na tento dotaz jsou v odpovědi vráceny informace o umístění. Adresa, pro kterou jsou tyto informace získávány, je uložena v proměnné *placeurl*. Tato adresa byla do této proměnné uložena po porovnávání řetězců a tedy obsahuje řetězec, který při porovnávání s podpůrnými soubory porovnání vyhověl.

```
1 $url = "https://maps.googleapis.com/maps/api/geocode/xml?address=
    $placeurl&key=$API_key"; // Ulozeni retezce pro google geocode api
    do promene url
2 $FGC = file_get_contents($url);// Nacteni informaci o~umisteni
3 file_put_contents("./FGC.txt", $FGC); // Ulozeni informaci o~umisteni
```

Výpis kódu 3.4: `search.php` - získání souřadnic pomocí Google Geocode API

Informace o umístění, získané prostřednictvím Google Geocode API, jsou uloženy do souboru `FGC.txt` odkud jsou následně získány souřadnice GPS, tedy geografické souřadnice *latitude* a *longtitude*.

Po procesu získání souřadnic následuje ověření, zda byly zadány také skutečné souřadnice stanice. Pokud zadány nebyly, je poloha stanice zobrazena na mapě a její souřadnice vypsány v tabulce. Pokud byly zadány skutečné souřadnice, je proveden výpočet chyby lokace stanice a následně jsou v tabulce vypsány nalezené i zadané souřadnice a tyto zobrazeny na mapě. Dále je při zadání skutečných souřadnic také provedeno uložení dat geolokace do souboru `data.txt`. V tomto souboru pak každý řádek reprezentuje geolokaci stanice se zadanou skutečnou polohou.

Popis výpočtu chyby lokace stanice a zobrazení výsledků je předmětem následujících dvou podkapitol.

3.2 Výpočet chyby lokace

Jak již bylo uvedeno v předchozí podkapitole, pokud je při zadávání IP adresy (doménového jména stanice) do formulářového pole (viz obr. 3.2) zadána také skutečná adresa stanice, případně její souřadnice, je proveden výpočet chyby lokace polohy stanice. Při výpočtu chyby lokace polohy je pro dosažení nejlepšího výsledku vhodné použít, z důvodu nejlepší podoby se zemským tělesem, elipsoid definovaný systémem WGS84. Pro zjednodušení výpočtu je však zde pro výpočet chyby lokace polohy vycházeno z předpokladu, že Země je kulové těleso. Za tohoto předpokladu je možno při tomto výpočtu použít tzv. Haversinův algoritmus[17], který má pro výpočet vzdálenosti mezi dvěma body následující tvar:

$$a = sin^2(\frac{\triangle lat}{2}) + cos(lat_1) \cdot cos(lat_2) \cdot sin^2(\frac{\triangle long}{2}) \tag{3.1}$$

$$c = 2arctan(\sqrt{a}, \sqrt{1-a}) \tag{3.2}$$

$$d = R \cdot c \tag{3.3}$$

Kde R je poloměr Země = 6371 km;

Jak je možné vidět ve vzorci 3.1, k výpočtu je tedy potřeba dvě dvojice souřadnic *latitude* a *longtitude*. První dvojice reprezentuje skutečné souřadnice stanice. Druhá dvojice představuje souřadnice stanice nalezené aplikací na základě informací z databáze RIPE NCC. Postup získání souřadnic na základě informací z databáze RIPE NCC byl uveden v předcházející podkapitole. Skutečné souřadnice stanice mohou být získány dvěma způsoby. Tím prvním je zadání adresy do formulářového pole, před kterým je uvedeno „Enter the right address“, na úvodní stránce aplikace. Tato adresa je po načtení formulářového pole načtena do proměnné *location* a dále zpracována. Toto zpracování je realizováno následující částí zdrojového kódu (výpis kódu 3.5), kde je adresa uložená v proměnné *location* nejprve rozdělena na město, ulici a stát. Následně jsou pro toto umístění z Google map získány informace o zadané adrese, ze kterých jsou opět s využitím filtru `awk` získány souřadnice stanice pro zadanou adresu. Tyto souřadnice jsou pak uloženy do proměnných *LatE* a *LongE*.

Druhým způsobem získání souřadnic zadané adresy stanice pro výpočet chyby lokace je jejich přímé zadání do formulářového pole na úvodní stránce aplikace, před kterým je uvedeno „GPS:“. Tyto souřadnice mohou být vyčteny z GPS zařízení nebo mohou být stanoveny z mapy (např. z Google map[2] nebo na mapách společnosti Seznam.cz, a.s.[3]). V případě, že jsou na úvodní stránce aplikace vyplněna

[2] na adrese https://www.google.cz/maps/

[3] na adrese www.mapy.cz

```
list($cityE, $streetE, $countryE) = explode(",", $location);// Nactena
    poloha stanice z~formulare je rozdelena na casti mesto, ulice a~
    stat
$cityE_url = urlencode($cityE); // Uprava formatu promenne pro pouziti
    v~url retezci
$streetE_url = urlencode($streetE); // Uprava formatu promenne pro
    pouziti v~url retezci
$geo_enter_in = "wget –O in_enter.txt https://www.google.cz/maps/place/
    $cityE_url+$streetE_url"; // Prikaz k~ziskani informaci o~zadane
    poloze a~jejich ulozeni do souboru
$geo_enter_out = <<<qqq
cat in_enter.txt | awk 'BEGIN { RS = "{"; FS = ";" } {print $0 }' |
    grep viewport_center | awk 'BEGIN { RS = ";"; FS = "=" } {print $2
    }' > out_enter.txt
qqq;// Definovani prikazu pro provedeni filtrace dat a~ulozeni vystupu
    do souboru
echo shell_exec ("$geo_enter_in");// Provedeni prikazu ziskani dat
echo shell_exec ("$geo_enter_out");// Provedeni prikazu filtrace dat
$soubor = fopen("./out_enter.txt", "r"); // Otevreni souboru s~vysledky
     filtrace
$LatE=fgets($soubor); // Nacteni hodnoty latitude zadane polohy stanice
$LongE=fgets($soubor); // Nacteni hodnoty longtitude zadane polohy
    stanice
fclose($soubor);//zavreni souboru
```

Výpis kódu 3.5: `search.php` - určení souřadnic zadané adresy stanice

obě formulářová pole, jsou upřednostněny souřadnice, které byly zadány. Hodnoty proměnných *LatE* a *LongE* jsou nastaveny podle zadaných souřadnic a již tedy není prováděno hledání souřadnic pro zadanou adresu.

V tomto okamžiku již jsou známy obě dvojice souřadnic a je tedy možné provést výpočet chyby. Ten je proveden podle vzorců 3.1, 3.2 a 3.3 a jeho realizace pomocí programovacího jazyka PHP je zobrazena ve výpisu kódu 3.6, kde proměnné *LatE* a *LongE* tedy představují skutečné souřadnice stanice a proměnné *LatS* a *LongS* představují souřadnice získané na základě informací z databáze RIPE NCC.

3.3 Zobrazení výsledku geolokace

Poslední částí zhotovené webové aplikace, po provedené geolokaci stanice s využitím dat z databáze RIPE NCC, je zobrazení výsledků geolokace v tabulce a na mapě. V tabulce jsou jednotlivá data zadané a hledané polohy stanice zobrazovány v řádcích a následně je poloha stanice vykreslena v mapě. Pokud byla zadána skutečná

```
$deltaLat = $LatE - $LatS; // Vypocet delta lat
$deltaLong = $LongE - $LongS; // Vypocet delta long
$distance = ((1 - cos(deg2rad($deltaLat))) / 2) + cos(deg2rad($LatS)) *
    cos(deg2rad($LatE)) * ((1 - cos(deg2rad($deltaLong))) / 2); //
    Realizace Haversinova algoritmu
$distance = atan2((sqrt($distance)) , (sqrt(1 - $distance))); //
    Realizace Haversinova algoritmu
$distance = 2 * 6371 * $distance; // Realizace Haversinova algoritmu
```

Výpis kódu 3.6: `search.php` - výpočet chyby lokace

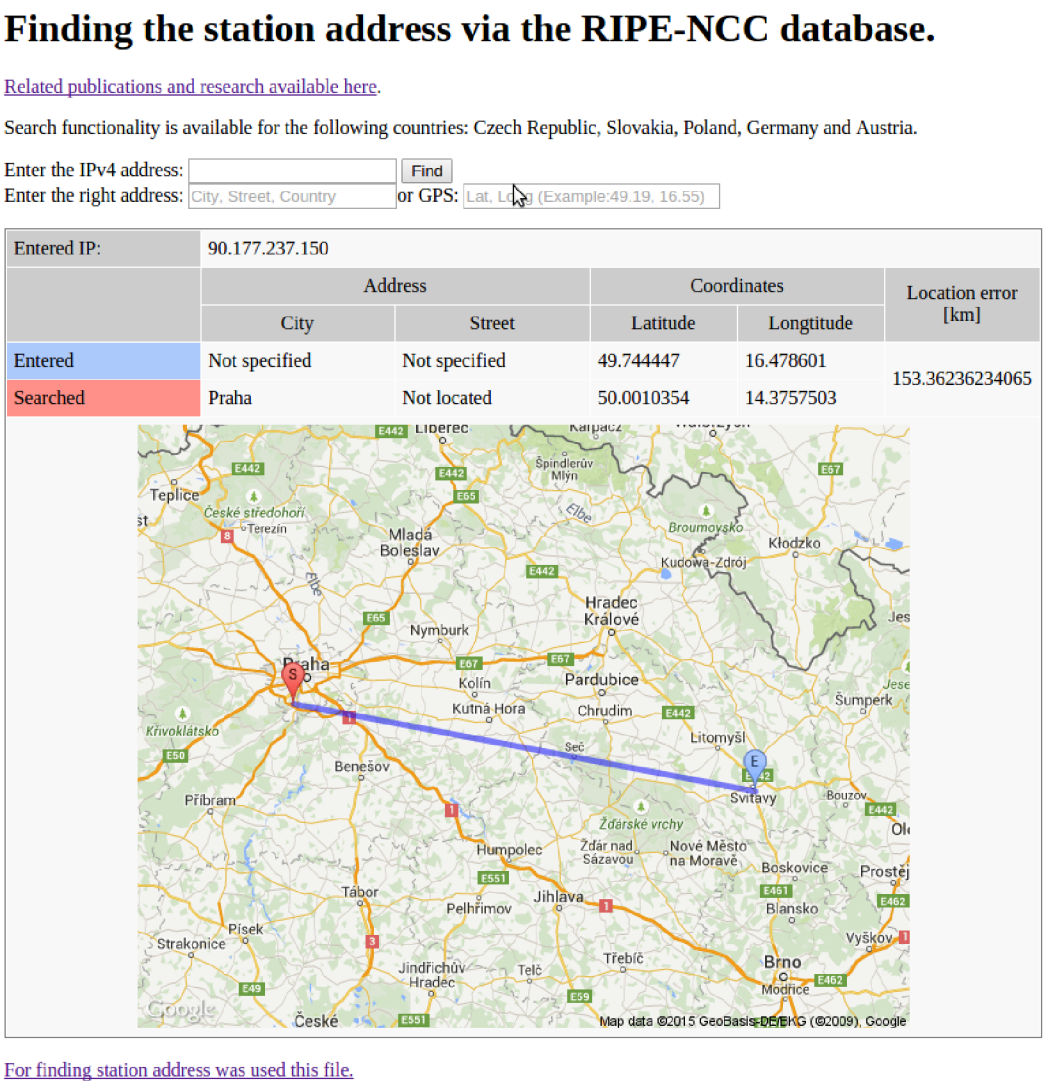
adresa stanice, jsou na mapě vykresleny dvě polohy a je vypočítána chyba lokace, která je uvedena v kilometrech. Červený bod na mapě, s označením „S“ (Searched), znázorňuje nalezenou polohu stanice, modrý bod na mapě, s označením „E“ (Entered), znázorňuje zadanou polohu stanice. Zobrazení polohy stanice je realizováno využitím Google Static Maps API v2[15]. Toto rozhraní bylo pro tento úkol vybráno z důvodu snížení nároků na programové vybavení při využití této aplikace (není třeba JavaScript). Vykreslení mapy je provedeno v poslední buňce tabulky, kdy je v této buňce umístěn obrázek, který má zdrojovou adresu v Google mapách. Tento zdroj obrázku je pro názornost uveden ve výpisu kódu 3.7. V tomto kódu je patrná výše zmiňovaná dvojice souřadnic *latitude* a *longtitude* a některé další nastavitelné parametry jako např. velikost mapy v pixelech (*size*), přiblížení mapy (*zoom*), typ mapy (*maptype*) a ukazatelé míst uváděných souřadnicemi (*markers*), které ukazují zadanou a nalezenou polohu stanice.

```
$tested = <<<qqq
<img src="https://maps.googleapis.com/maps/api/staticmap?zoom=10&size
    =700x500&scale=1&maptype=roadmap&markers=color:red%7Clabel:S%7
    C$LatS, $LongS&markers=color:blue%7Clabel:E%7C$LatE, $LongE&path=
    color:0x0000ff|weight:5|$LatS,$LongS|$LatE,$LongE">
qqq; //nastaveni promenne, kterou je realizovano vykresleni mapy v~
    tabulce
```

Výpis kódu 3.7: `search.php` - realizace mapového vykreslení

Na obr.3.3 je zobrazen příklad, ve kterém byla hledána poloha stanice s IP adresou 90.177.237.150, jejíž skutečná poloha je v městě Svitavy. Poloha stanice, nalezená na základě informaci v databázi RIPE NCC, je v městě Praha. Chyba lokace polohy stanice byla v tomto případě asi 153 km.

Pro potřeby porovnání výsledků geolokace, například s jinou metodou geolokace, umožňuje aplikace zobrazení výsledků geolokace ve formátu JSON. Tohoto zobrazení



Obr. 3.3: Zobrazení výsledku geolokace realizovanou aplikací

lze dosáhnout zadáním URL řetězce aplikace, ve kterém bude uveden parametr `json=` *true*. Pro výše uvedený příklad by takový řetězec vypadal následovně:

adresa webového rozhraní. . . `/search.php?json=true&ip=90.177.237.150`

kde odpovědí aplikace na tento URL řetězec (dotaz) by byl prostý text formátovaný pro JSON, obsahující hodnoty jednotlivých proměnných (*ip*, *country*, *city*, *street*, *latitude*, *longtitude* a *NOTE*).

4 ANALÝZA VÝSLEDKŮ GEOLOKACE

Jak již bylo nastíněno v předchozí kapitole, realizovaná webová aplikace provádí v případě geolokace stanice se zadanou polohou uložení získaných (naměřených) dat. Tyto data jsou ukládány na serveru, kde je aplikace uložena, do souboru `data.txt`. Data v tomto souboru jsou pro jednotlivá měření ukládány do jednotlivých řádků, ve kterých jsou jednotlivé hodnoty od sebe odděleny tabelátorem. Každé nové měření je přidáno jako nový řádek na konec souboru. Data ukládaná do souboru `data.txt` pro příklad, který je zobrazen na obr. 3.3 vypadají následovně:

```
90.177.237.150 49.744447 16.478601 50.0010354 14.3757503
153.3697410922
```

Jednotlivé hodnoty v řádcích souboru `data.txt` zleva doprava představují:

- IP adresu stanice, pro kterou je prováděna geolokace;
- hodnotu *latitude* skutečné polohy stanice;
- hodnotu *longtitude* skutečné polohy stanice;
- hodnotu *latitude* polohy stanice nalezené na základě informací z databáze RIPE NCC;
- hodnotu *longtitude* polohy stanice nalezené na základě informací z databáze RIPE NCC;
- vypočtenou chybu lokace stanice uvedenou v km.

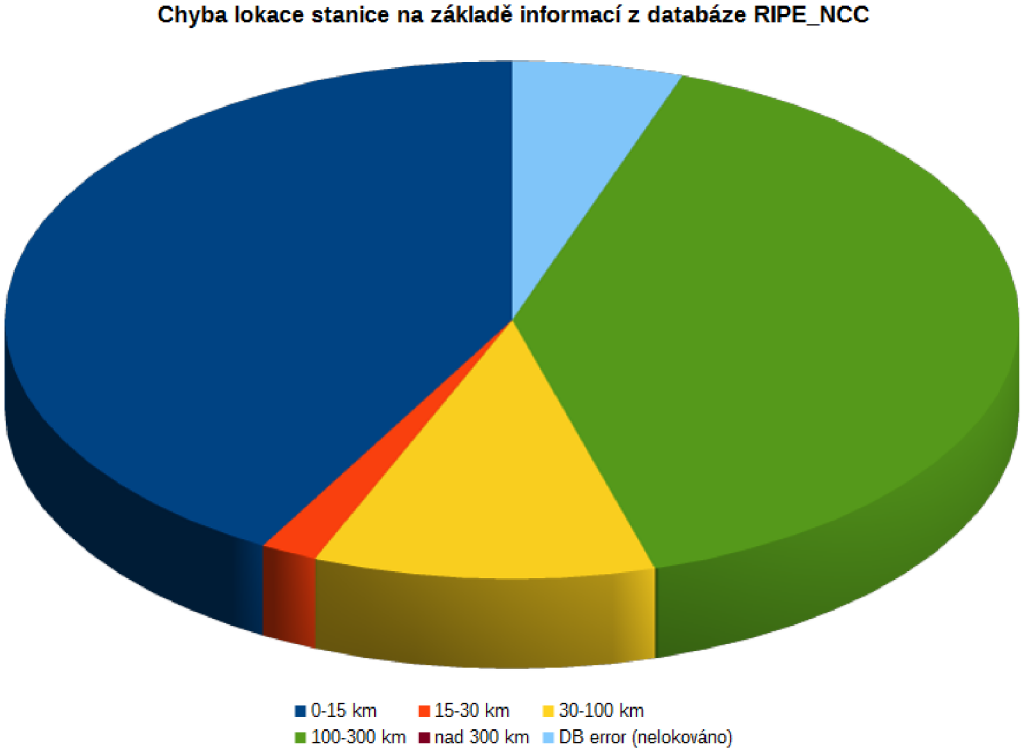
Pro relevantní zhodnocení přesnosti geolokace stanice byl, ve spolupráci se studenty, kteří řeší problematiku geolokace stanice pomocí dalších databází, vytvořen seznam stanic se známou polohou. Tento seznam je, ve formě excelového souboru, přístupný jako sdílený dokument na serveru společnosti Google a také na disku CD, který je přiložen k této práci.

Do tohoto souboru byly jednotlivými studenty, kteří se zabývají problematikou geolokace, postupně doplněny údaje o jednotlivých stanicích pro testování databází, převážně z České republiky. V jednotlivých sloupcích tohoto souboru jsou pak uvedeny tyto údaje:

- IP adresa stanice, pro kterou je prováděna geolokace;
- skutečné souřadnice stanice (zeměpisná šířka a zeměpisná délka);
- zdroj souřadnic (odkud byly získány - mapa, GPS);
- skutečná adresa stanice (ulice, město, kraj a stát);
- datum přidání záznamu;

- jméno studenta, který daný záznam přidal.

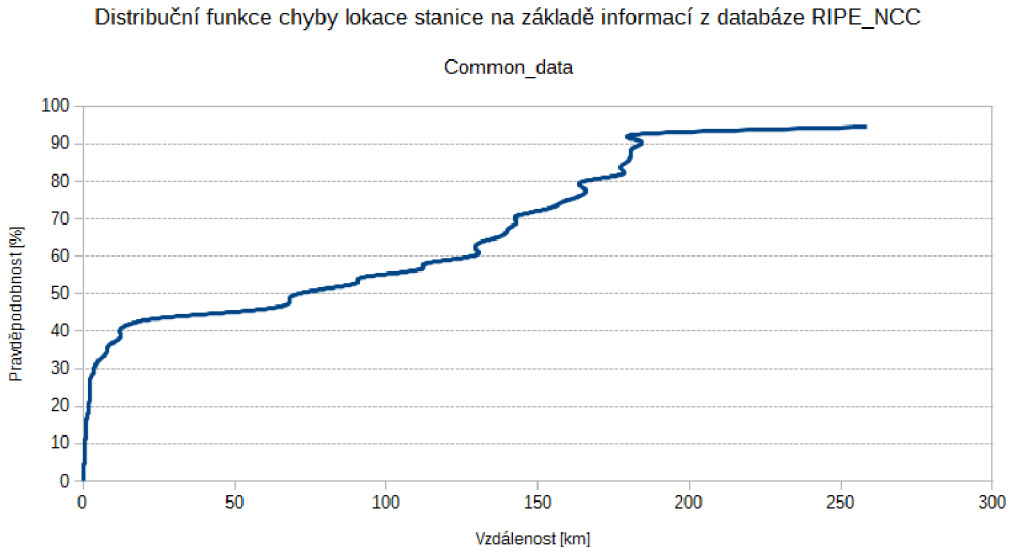
V uvedeném souboru bylo nashromážděno 56 IP adres stanic se známou polohou, na kterých byla ověřena přesnost geolokace jednotlivých metod. V této práci byla ověřena přesnost geolokace stanic, u kterých bylo provedeno určení polohy na základě informací z databáze RIPE NCC.



Obr. 4.1: Chyba geolokace hledané polohy stanice

Na obr.4.1 je zobrazena chyba lokace u jednotlivých stanic z výše uvedeného souboru. Z uvedeného obrázku je patrné, že nejvíce je zastoupena skupina, kde je přesnost lokace polohy stanice do 20 km. U této skupiny stanic lze říci, že přesnost lokace je na úrovni města, to znamená, že u těchto stanic se podařila určit poloha stanice tak, že město nalezené geolokací stanice odpovídá městu, ve kterém se skutečně daná stanice nachází. Druhou velkou skupinou stanic tvoří stanice, u kterých byla přesnost lokace polohy stanice od 100 km do 300 km. Tato skupina je tvořena stanicemi, u kterých se přesnost lokace pohybuje spíše na úrovni státu. Stanice, u kterých se přesnost lokace pohybuje mezi těmito skupinami jsou zastoupeny v mnohem menším množství a dá se říci, že jde o lokaci s přesností na úrovni kraje.

V ověřovaném vzorku stanic se však nalezly také stanice, u kterých nebyla lokace provedena, resp. byla provedena s výsledkem „DB error“. Jak již bylo uvedeno v předchozí kapitole, realizovaná aplikace provádí geolokaci stanic v České republice, Slovensku, Polsku, Německu a Rakousku. Pokud je u nějaké stanice v databázi uveden jiný stát než výše jmenovaný, je u takovéto geolokace automaticky vyplněna hodnota „DB error“. Stanice v ověřovaném vzorku stanic, u kterých byla tato hodnota uvedena, měli v databázi záznam o příslušnosti k Irské republice a proto u nich nebyla lokace provedena.

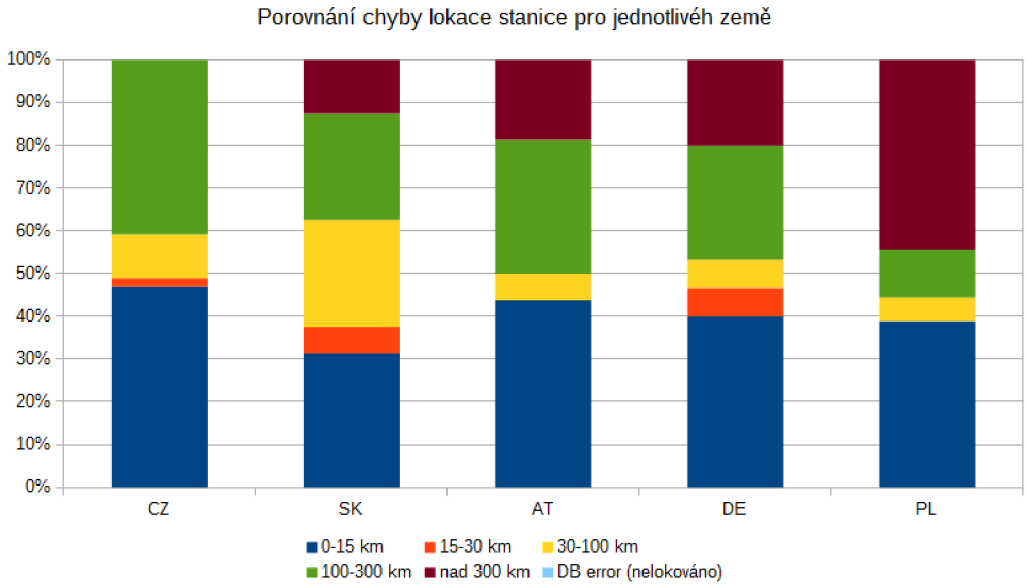


Obr. 4.2: Distribuční funkce chyby lokace stanice - společná data

Pro ověřovaný vzorek stanic byl také zhotoven graf distribuční funkce chyby lokace stanice (obr. 4.2). Ten vyjadřuje pravděpodobnost s jakou se dá očekávat určitá velikost chyby metody (v tomto případě je chyba vyjádřena vzdáleností v km).

Vzhledem ke skutečnosti, že realizovaná aplikace provádí geolokaci pro výše uvedených pět států, bylo vhodné rozšířit ověřovaný vzorek o data stanic, nacházejících se na území jmenovaných států. Proto bylo pro potřeby ověření přesnosti lokace stanic v jednotlivých státech přidáno dalších 63 stanic se známou polohou, které jsou rozmístěny v rámci jednotlivých států. Seznam těchto stanic je také uveden v souboru `geolokace_ip_adresy.xlsx` na přiloženém CD. Výsledky geolokace těchto stanic jsou zobrazeny v grafu na obr. 4.3. Z tohoto grafu je zřejmé, že relativní přesnost geolokace stanice v jednotlivých zemích je, zejména u stanic s chybou lokace zhruba do 100 km, přibližně stejná. Co se týče stanic, u kterých je chyba lokace vyšší, tam už přichází jisté rozdíly mezi jednotlivými zeměmi. V grafu na obr. 4.3 je názorně

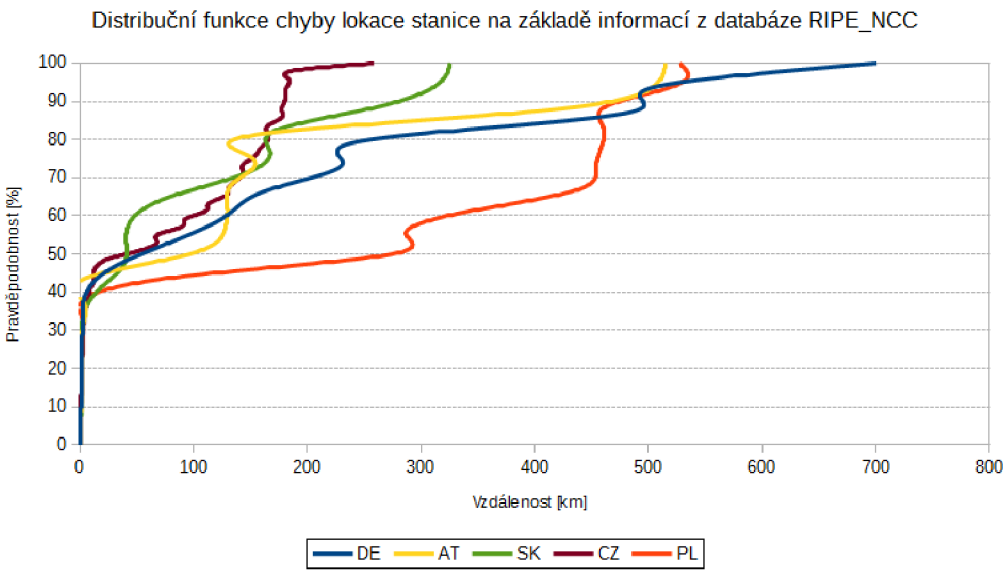
vidět, že nejvíce stanic s chybou lokace polohy stanice nad 300 km je u stanic umístěných v Polsku a Německu. V těchto případech lze říci, že velikost chyby lokace stanice u této metody je do jisté míry ovlivněna také rozlohou dané země.



Obr. 4.3: Chyba lokace stanice v jednotlivých zemích

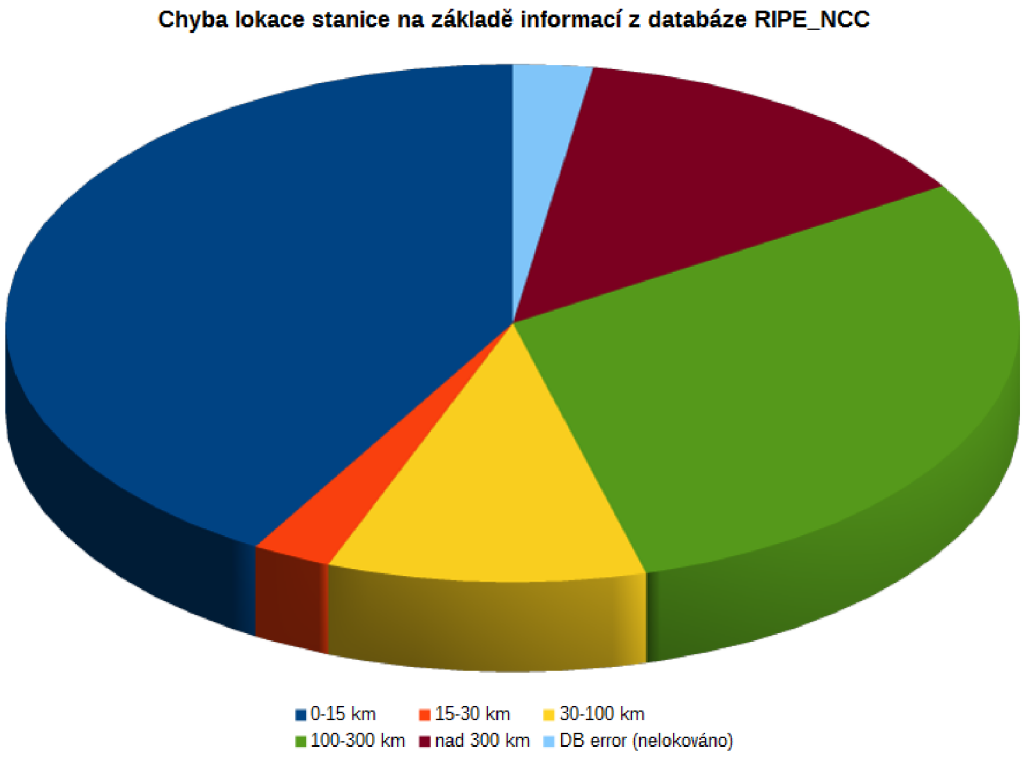
Na obr. 4.4 je pak zobrazen graf distribuční funkce chyby lokace stanice v jednotlivých zemích, který byl sestrojen na základě dat, naměřených při ověřování polohy stanic, nacházejících se v těchto zemích. Z tohoto grafu je patrné, že pravděpodobnost pro malé chyby lokace stanice (do 20 km) je téměř shodná. Od této hodnoty chyby lokace (20 km) se jednotlivé křivky začínají rozcházet, což je zřejmě z části ovlivněno právě rozlohou země.

Na obr. 4.5 je zobrazen graf, znázorňující chybu lokace stanice pro všechna měřená data touto metodou. Z tohoto grafu je zřejmé, že největší část lokalizovaných stanic mělo chybu lokace do 15 km. Dobré výsledky jsou ale vyváženy těmi špatnými, což negativně ovlivňuje celkovou přesnost použité metody. Pro všechna měřená data byl také vytvořen graf distribuční funkce chyby lokace stanice (obr. 4.6, který, tak jako grafy na obr. 4.2 a obr. 4.4, vyjadřuje pravděpodobnost s jakou se dá očekávat určitá velikost chyby metody. Velikost chyby lokace se u této metody pohybuje v poměrně velkém rozsahu hodnot - od zhruba jednotek kilometrů až po stovky kilometrů. Tato nepřesnost geolokace polohy je zapříčiněna zejména způsobem geolokace této metody, kdy geolokace pomocí databáze RIPE NCC je založena na vyčtení adresy provozovatele sítě z databáze. Pokud tedy má provozovatel sítě přidělen velký adresní prostor a pokud provozuje svoji síť na geograficky rozlehlém území, pak se dá

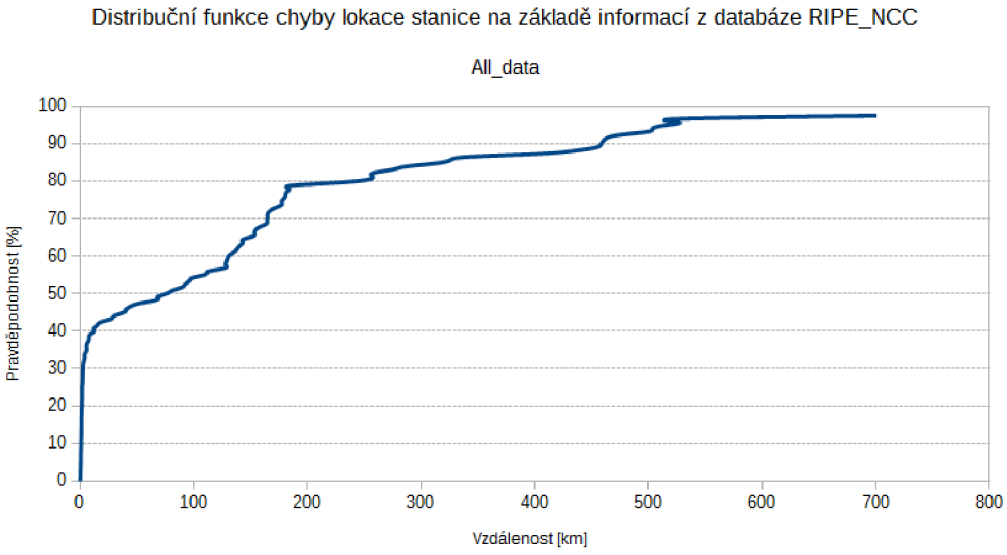


Obr. 4.4: Distribuční funkce chyby lokace stanice pro jednotlivé země

předpokládat, že u stanic z takovéto sítě dojde při této geolokaci k chybě. Původem této chyby je ve většině případů to, že tento provozovatel má v databázi uvedené pouze údaje o adrese centrály (sídla) provozovatele. Na základě provedené analýzy výsledků se dá konstatovat, že celkově tato metoda geolokace nepatří mezi příliš přesné metody.



Obr. 4.5: Chyba geolokace hledané polohy stanice - všechna data



Obr. 4.6: Distribuční funkce chyby lokace stanice - všechna data

5 ZÁVĚR

V této práci byla nastudována a přiblížena problematika určení geografické polohy stanice v síti Internet, přičemž byl položen důraz zejména na určení polohy pomocí registrační databáze IP adres. Uvedená registrační databáze je databází evropského registrátora adres a nese název RIPE Network Management Database. Dále byl v rámci této práce ve spolupráci s ostatními řešiteli obdobného tématu vytvořen seznam stanic se známou geografickou polohou a webové rozhraní pro zjištění polohy stanice prostřednictvím geolokace s využitím databáze RIPE NCC a následným zobrazením hledané polohy stanice na mapě.

Motivací při realizaci tohoto webového rozhraní byla možnost zjištění polohy stanice s využitím informací v databázi, bez ohledu na jejich zařazení v jednotlivých položkách databáze. Odpověď na dotaz této databázi může totiž pokaždé obsahovat informace o poloze v jiné položce podle toho, jak byly zadány při registraci. Webové rozhraní tedy vyhledává pro jednotlivé stanice v této databázi informace o umístění a následně ho zobrazuje na mapě. Pokud je prostřednictvím webového rozhraní zadána i skutečná adresa stanice, je také vypočítána a zobrazena chyba lokace a data jsou ukládána realizovanou aplikací na serveru[1] pro možnost analýzy výsledků geolokace.

Při analýze výsledků bylo zjištěno, že přesnost lokace stanice je u ověřovaného vzorku stanic různá. U téměř poloviny stanic je přesnost na úrovni města, případně konkrétní ulice, avšak tato přesnost lokace je negativně ovlivněna přesností (lépe řečeno nepřesností) dalších stanic v ověřovaném vzorku, kdy největší chyby se objevují zejména u velkých provozovatelů sítí. Tyto chyby jsou způsobeny zejména obecnými záznamy u adres provozovatelů, tj. většinou jsou uvedeny pouze adresy central nebo sídel těchto provozovatelů.

[1] http://geolocation.utko.feec.vutbr.cz/xsmrck00/data.txt

LITERATURA

[1] PUŽMANOVÁ, R. *TCP/IP v kostce. 1. vyd. Kopp. České Budějovice, 2004. 607 s. ISBN 80-7232-236-2.*

[2] FORST, L. *Shell v příkladech aneb ABY VÁŠ UNIX SKVĚLE SHELL. MATFYZPRESS. Praha, 2010. 387 s. ISBN 978-80-7378-152-1.*

[3] VERNER, L.; KOMOSNÝ, D. *Geolokace síťových zařízení v internetových sítích* [online]. 2011, poslední aktualizace 17. 6. 2011 [cit. 24. 5. 2015]. Dostupné z URL: <http://www.elektrorevue.cz/cz/download/geolokace-sitovych-zarizeni-v-internetovych-sitich/>.

[4] TURNER, A. *Geolocation by IP Address* [online]. 2004, poslední aktualizace 25. 10. 2004 [cit. 24. 5. 2015]. Dostupné z URL: <http://www.linuxjournal.com/article/7856>.

[5] ČELADA, P.; KADERKA, J. *Geolokace a bezpečnost počítačových sítí* [online]. 2012, poslední aktualizace 2. 10. 2012 [cit. 24. 5. 2015]. Dostupné z URL: <http://is.muni.cz/repo/1067555/geolokace-a-bezpecnost-pocitacovych-siti.pdf>.

[6] KATZ-BASSET, E.; et al. *Towards IP Geolocation Using Delay and Topology Measurements* [online]. 2006, poslední aktualizace 25.-27. 10. 2006 [cit. 24. 5. 2015]. Dostupné z URL: <http://homes.cs.washington.edu/~arvind/papers/geoloc.pdf>.

[7] *GeoIP Products* [online]. 2014, [cit. 24. 5. 2015]. Dostupné z URL: <http://dev.maxmind.com/geoip/>.

[8] *Geo-DNS Service* [online]. 2012, [cit. 24. 5. 2015]. Dostupné z URL: <http://www.mtgsy.net/dns/geolocation.php>.

[9] HARRENSTIEN, K.; et al. *NICNAME/WHOIS* [online]. 1985, [cit. 24. 5. 2015]. Dostupné z URL: <ftp://ftp.ripe.net/rfc/rfc954.txt>.

[10] *RIPE Database Query Reference Manual* [online]. 2011, poslední aktualizace 20. 8. 2014 [cit. 24. 5. 2015]. Dostupné z URL: <https://www.ripe.net/data-tools/support/documentation/ripe-database-query-reference-manual>.

[11] *IPv4 Address Report* [online]. 2015, [cit. 24. 5. 2015]. Dostupné z URL: <http://www.potaroo.net/tools/ipv4/index.html>.

[12] *Internet Assigned Numbers Authority* [online]. 2015, [cit. 24. 5. 2015]. Dostupné z URL: <`https://www.iana.org`>.

[13] *GeoNames - geographical database* [online]. 2015, [cit. 24. 5. 2015]. Dostupné z URL: <`http://www.geonames.org`>.

[14] *Google Maps Geocoding API* [online]. 2015, [cit. 24. 5. 2015]. Dostupné z URL: <`https://developers.google.com/maps/documentation/geocoding/`>.

[15] *Google Static Maps API* [online]. 2015, [cit. 24. 5. 2015]. Dostupné z URL: <`https://developers.google.com/maps/documentation/staticmaps/`>.

[16] Satrapa, P. *Internetový protokol verze 6. CZ.NIC, z. s. p. o., 2011. ISBN 978-80-904248-4-5.*

[17] MRÁZOVÁ, A. *Porovnání přesnosti metod pasivní IP geolokace. bakalářská práce.Brno, 2014. 50 s.*

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

API	Application Programming Interface
CBG	Constraint Based Geolocation
CSIRT	Computer Security Incident Response Team
DCA	Defense Communications Agency
DNS	Domain Name Service
GNP	Global Network Positioning
GPS	Global Positioning System
IANA	Internet Assigned Numbers Authority
ICANN	Internet Corporation for Assigned Names and Numbers
INR	Internet Number Registry
IRR	Internet Routing Registry
LIR	Local Internet Registry
NAT	Network address translation
NIC	Network Informatin Center
NSF	National Science Foundation
RIR	Regional Internet Registry
RPSL	Routing Policy Specification Language
RTT	obousměrné zpoždění – Round Trip Time
SOI	Speed of Internet
URL	Uniform Resource Locator

SEZNAM PŘÍLOH

A OBSAH PŘILOŽENÉHO CD

Elektronická verze této diplomové práce.
`whois.zip` ... soubory se zdrojovými kódy v programovacím jazyku PHP,
`geolokace_ip_adresy.xlsx` ... seznam stanic se známou polohou,
`data.ods` ... naměřená a analyzovaná data.