

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

LOKACE STANICE V SÍTI INTERNET POMOCÍ ANALÝZY
IP ADRESY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN HENEK



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

LOKACE STANICE V SÍTI INTERNET POMOCÍ ANALÝZY IP ADRESY

GEOLOCATION OF INTERNET NODES USING IP ADDRESS ANALYSIS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAN HENEK

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. DAN KOMOSNÝ, Ph.D.

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Jan Henek

ID: 134487

Ročník: 3

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Lokace stanice v síti Internet pomocí analýzy IP adresy

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s principy vyhodnocování fyzické polohy stanic v síti Internet. Nastudujte systém adresace v Internetu pro IP protokol verze 4 i verze 6. Realizujte aplikaci, která bude zjišťovat polohu stanice pomocí analýzy IP adresy. Pomocí reálných IP adres ověřte správnou činnost navržené aplikace.

DOPORUČENÁ LITERATURA:

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

[2] DOSTÁLEK L. et al.: Velký průvodce protokoly TCP/IP: Bezpečnost. 2. aktualizované vydání. Computer Press, 2003. ISBN 80-7226-849-X.

[3] NEMETH, E., SNYDER, G., HEIN T. Linux - Kompletní příručka administrátora. Computer Press, 2004. 880 s. ISBN: 80-722-6919-4.

Termín zadání: 11.2.2013

Termín odevzdání: 5.6.2013

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

Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato práce se zabývá problematikou určování fyzické polohy pomocí rozboru IP adresy. Zkoumanou metodou geolokace je databáze Whois. Práce je rozdělena na teoretickou část – která obsahuje obecný popis geolokačních metod, práce se souřadnicemi a výpočtem vzdáleností na zemi, a na praktickou část – kde je popsán program využívající Whois k nalezení stanice. Na závěr jsou předloženy a komentovány výsledky měření přesnosti databáze Whois, a také porovnány s jinými metodami geolokace. Aplikace je přiložena na CD.

KLÍČOVÁ SLOVA

geolokace, IP, Internet, Whois, databáze, GPS

ABSTRACT

This paper deals with the issue of the determination of physical position with help of the analysis of IP adress. The examined method of geolocation is the database Whois. The paper is divided into the theoretical part – which contains a general description of geolocation methods, work with coordinates and calculating distances on Earth and into the practical part – where a programe using Whois to find the target is described. At the end there are presented and commented the results of the research of the accuracy of the database Whois, they are also compared with other geolocation methodes. The application is enclosed on the CD.

KEYWORDS

geolocation, IP, Internet, Whois, database, GPS

HENEK, Jan *Lokace stanice v síti Internet pomocí analýzy IP adresy*. bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2012. 57 s. Vedoucí práce byl doc. Ing. Dan Komosný, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Lokace stanice v síti Internet pomocí analýzy IP adresy“ jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské 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)

OBSAH

Úvod	11
1 Geolokace	13
1.1 Základní pojmy	13
1.2 Příklady využití geolokace	14
1.3 Pasivní metody	16
1.3.1 Geolokace na základě IP adresy	16
1.3.2 Geolokace na základě domén	19
1.4 Aktivní metody	20
2 Způsob přiřazování IP adres	21
2.1 Rozdělení organizací	21
2.2 RIPE NCC	23
2.3 Rozdělení adres	23
3 Global Positioning System	25
3.1 Úvod do technologie GPS	25
3.2 Struktura a funkce systému	25
3.2.1 Kosmický segment	26
3.2.2 Řídící segment	26
3.2.3 Uživatelský segment	26
4 Určování pozice a vzdálenosti na zeměkouli	27
4.1 Matematický popis zeměkoule	27
4.1.1 Geoid	27
4.1.2 Rotační elipsoid	28
4.2 Geodetický systém WGS 84	28
4.3 Kartografický zobrazovací systém UTM	29
4.4 Určení vzdálenosti mezi dvěma body	30
5 Popis programu	32
5.1 Způsob přístupu k databázi Whois	33
5.1.1 Struktura programu	33
5.1.2 Komunikace klient-server	33
5.2 Konzolový nástroj whois	34
5.3 Filtrování výstupu Whois	35
5.3.1 Funkce getIn	35
5.3.2 Funkce main	36

5.4	Převod adresy na souřadnice GPS	37
5.4.1	Google Geocoding API	37
5.4.2	GNU Wget	37
5.4.3	Filtrování souřadnic z XML kódu pomocí getLocation	38
5.5	Zobrazení výstupních informací na mapě	40
5.6	Funkce programu	40
6	Analýza naměřených dat	44
6.1	Výpočet odchylky mezi správným a nalezeným bodem.	44
6.2	Zhodnocení výsledků měření	44
6.3	Porovnání přesností metod geolokace	47
7	Závěr	49
	Literatura	51
	Seznam symbolů, veličin a zkratk	54
	Seznam příloh	56
A	Obsah přiloženého CD	57

SEZNAM OBRÁZKŮ

1.1	Odpověď <code>whois</code> na IP adresu 77.251.170.54.	17
1.2	Odpověď <code>whois</code> na IP adresu 80.239.168.0.	18
1.3	Princip DNS při zadání adresy <code>www.seznam.cz</code> . Převzato z [4], str. 15.	19
1.4	Výpis nástroje <code>nslookup</code> pro doménu <code>www.battle.net</code>	20
2.1	Stromová struktura dělení mezinárodních organizací pro rozdělování IP adres.	21
2.2	Celosvětové rozdělení regionů[22].	22
4.1	Rozdíl mezi geoidem, elipsoidem a koulí [3].	27
4.2	Světový geodetický systém WGS 84, převzato z [20].	29
4.3	Souřadnicové zobrazení UTM [3].	30
5.1	Vývojový diagram programu.	32
5.2	Konzolové spouštění aplikace a zadání vstupně-výstupních parametrů.	41
5.3	Příklad zobrazení vstupního a výstupního textového souboru.	42
5.4	Zobrazení průběhu zpracování IP adres.	42
5.5	Zobrazení nalezených IP adres na mapě.	43
5.6	Detail informací nalezených bodů.	43

SEZNAM TABULEK

2.1	Rozdělení bloků IP adres regionům.	23
2.2	Rozsah bloků IPv4 adres pro RIPE NCC.	24
6.1	Tabulka vzdáleností mezi správnou a nalezenou pozicí.	45
6.2	Přesnost Whois.	46
6.3	Porovnání průměrné chyby metod.	47
6.4	Porovnání přesnosti metod u konkrétních IP adres.	48

VÝPISY KÓDŮ

5.1	Definice portu pro komunikaci s Whois.	33
5.2	Definice serveru se službou Whois.	33
5.3	Načtení zadaných parametrů z hlavní třídy.	34
5.4	Připojení k serveru se službou Whois.	34
5.5	Komunikace programu s nástrojem whois.	34
5.6	Vyhledávání v textu pomocí regulárních výrazů.	35
5.7	Filtrování odpovědi z whois a výpis výsledku.	36
5.8	Filtrování odpovědi z google API pomocí getLocation.	39
5.9	Haversine zapsaný v javascript.	40

ÚVOD

Internet se stal samozřejmou součástí lidského života a pronikl snad do všech sfér našeho jednání. Jsme na něm více, či méně závislí, využíváme služeb serverů vzdálených stovky kilometrů, ukládáme na ně data pro sdílení s ostatními uživateli, či pro osobní účely. Obrovská expanze internetu do našich životů, a jeho morálního přijetí ve společnosti, tedy nezbytně vedla k rozšíření trhu dalším směrem. Obecná veřejnost naneštěstí není dostatečně obeznámena s riziky volného sdílení osobních informací, navíc takové jednání podporuje několik masivně rozšířených sociálních sítí, mezi kterými se stává populární sociální geolokační aplikace (Foursquare, Gowalla, Facebook Places). Tyto služby se zaslouhují o rozšíření geolokace do dalších oborů lidské činnosti. Zároveň tak roste riziko zneužití informací, kterým se zabývají složky veřejné správy, orientované na internetovou kriminalistiku. Ovšem samotné zamezení protiprávního jednání není dostatečné, je nutné najít pachatele. Stejně tak, je-li osoba v ohrožení, je nutné co nejdříve zjistit její polohu. Využití geolokace však není omezeno pouze na státní složky, naopak je velké množství komerčních využití – tzv. LBS (*Location-based services*). Cílená reklama, aktuální zobrazování obsahu, či hraní her jako je geocaching je dnes zcela běžné. Geolokace se stává velmi důležitou součástí zábavy, marketingu, bezpečnosti, boje s kriminalitou i poskytování pomoci a ochrany.

Pro geolokaci se využívá nejčastěji metod pracujících s IP adresou. Tato práce se zabývá konkrétním typem pasivní IP geolokace. Cílem je vytvořit algoritmus pro analýzu odpovědi z veřejné databáze Whois a porovnat její přesnost se známou polohou měřených IP adres. Další funkcí programu je převod zjištěné adresy na souřadnice a zobrazení pozice na mapě.

Práce je rozdělena na tři hlavní části – teoretickou, praktickou a analytickou. První kapitoly patří k teoretickému rozboru zkoumané problematiky. Postupně jsou popsány důležité body ohledně geolokace a jejích složkách. Pro všeobecný přehled jsou zde zahrnuty i metody využívající jiných informací, než je IP adresa. Protože při geolokaci využíváme souřadnic, je v kapitole Global Positioning System popsána funkce globálního družicového polohovacího systému, na který navazuje poslední teoretická kapitola Určování pozice a vzdálenosti na zeměkouli, která se podrobně zabývá aplikací GPS a výpočtem vzdálenosti mezi dvěma body na Zemi.

V popisu praktické části této práce, kterou zastřešuje kapitola Popis programu, je představen program pro analýzu výstupu Whois. Kapitola obsahuje vývojový diagram, popis důležitých částí zdrojového kódu a podrobnosti k běhu a spouštění programu. Algoritmus je psaný v jazyce C, využívá unixové konzole – v tomto případě proběhlo testování na Ubuntu – pro dotazování se na Whois. Po obdržení odpovědi z databáze dochází k filtraci textu na jméno serveru/majitele a jeho adresu. Ná-

sledně proběhne převod adresy na GPS souřadnice pomocí Google Geocoding API a zobrazení na mapě.

Poslední kapitola Analýza dat shrnuje zjištěné informace z měření a obsahuje komentář k naměřeným údajům. Výsledky přesnosti databáze Whois jsou porovnány s ostatními metodami geolokace.

1 GEOLOKACE

Geolokace je metoda, který využívá různých prostředků a služeb pro určení geografické – fyzické – polohy objektu (počítač, mobilní telefon, osoba . . .). Na rozdíl od pozičních systémů, které udávají zeměpisné souřadnice, je výsledkem geolokace většinou fyzická adresa – stát, město, ulice, PSČ, či jiný konkrétní údaj. [19]

V rámci geografické lokalizace stanic v internetu existuje více metod, využívajících rozdílných vlastností počítačové sítě, jako například IP geolokace, vyhledávání pomocí domén, měření zpoždění (latence), či pomocí Wi-Fi. Podle metody se liší i přesnost výsledku. Geolokační metody lze v základu rozdělit na:

- Pasivní – IP, DNS, Wi-Fi.
- Aktivní – Ping (měření latence).

Pro zjištění polohy stanice v síti lze využít těchto informací [19]:

- IP adresa – počítače, či klienta využívajícího stanici.
 - Prvotně se vychází ze znalosti centra pro správu IP daného kontinentu,
 - záznam v databázi (Whois, GeoIP, . . .),
 - znalost domény a informací z DNS,
 - informace od poskytovatele internetového připojení,
 - záznam v databázi autonomních systémů.
- Informace poskytnuté webovými nástroji.
 - Natavení jazyka webového prohlížeče, webmailu, Facebooku apod,
 - nastavení času, časové zóny ve webovém prohlížeči,
 - podrobnosti o systému uživatele, získané z JavaScriptu, Adobe Flash apod.
- Metadata uložená v obrázcích a fotkách.
- Informace z platebních karet.

Nejčastěji používanou metodou je IP geolokace, jejíž pasivní odnož je předmětem této práce. Budeme se blíže zabývat konkrétním způsobem lokalizace stanic v síti internetu za pomoci analýzy IP adresy.

1.1 Základní pojmy

Většina literatury zabývající se geolokací, včetně mé práce, obsahuje určité pojmy a výrazy, které jsou v běžném jazyce obecné a jejich užití v odborné terminologii nemusí být zcela jasné. Proto zde upřesním jejich význam v rámci geolokace a jejich metod.

Cíl (target): jedná se o stanici, kterou hledáme pomocí dané metody a jejíž geografickou polohu neznáme.

Referenční bod (landmark): je stanice, jejíž geografickou polohu známe. Pomocí referenčních bodů poté určujeme polohu cíle, proto rozlišujeme referenční body na aktivní a pasivní, tedy schopné vlastního měření a bez schopnosti měřit.

Sonda (probe): je stanice, jejíž hlavním cílem je měřit hledaný údaj, například měření zpoždění k ref. bodu. Hlavní rozdíl mezi sondou a aktivním referenčním bodem je v jejich účelu, tedy u sondy není tolik důležitá geografická pozice, jako výsledek měření, zato u ref. bodu nás především zajímá geografická poloha.

1.2 Příklady vyžití geolokace

S velkým rozvojem internetu se logicky rozšířil počet jeho uživatelů a informací na něm obsažených. Tím vzrostl i zájem různých firem, společností a celkově organizací o využití těchto zdrojů k vlastnímu obohacení. Internet má těžko určitelné hranice, proto se stala možnost vyhledat konkrétní stanici a určit její fyzickou polohu další součástí podnikání a rozvoje. Geolokace je zajímavý nástroj, sloužící nejen dané organizaci, ale i samotnému zákazníkovi, či uživateli. Může jít o cílenou reklamu, shromažďování informací o svém produktu, či přímo záchranné akce. Takovéto využití geolokace je poměrně nové, proto jsou možnosti stále omezené. Ze současného vývoje lze ovšem předpokládat, že v budoucnu expanduje využití geolokace do mnoha odvětví lidské činnosti. Zde si uvedeme jen některé z možných využití geolokace v současnosti:

Cílená reklama – Geo Targeting

Vzhledem k současnému rozvoji internetu je již samozřejmostí, že každá firma má svoji webovou stránku s produkty, které je zapotřebí propagovat. Cílená reklama využívá odhadu lokace uživatele k zprostředkování regionálních firem a jejich nabídek. Díky geolokaci tak lze zaměřit své produkty do cílených regionů, či pro cílené uživatele. S přidáním dalších parametrů lze docílit poměrně přesné a účelné reklamě, například při propagování restaurace v Paříži, lze zadat parametry: země – Francie, město Paříž a jako doplňkový parametr jazyk: francouzský, anglický, německý. Takto se reklama zobrazí jen návštěvníkům Paříže, jejichž vyhledávání bude mít nastaveno daný jazyk. Je možné využít i geo targetingu bez geolokace, kde polohu nezískává aplikace, ale zadá ji sám uživatel [2].

Zobrazování lokálních informací

Většina lidí ocení, když jim samotné stránky vyjdou vstříc a nabízí již lokální služby. Uživatel tak dostane přímo informace blízké své poloze, jako je nejbližší obchod s potravinami, kino, letiště, restaurace, ale i regionální zprávy, počasí, kulturní akce apod. Velkou výhodou může být tento nástroj pro lidi, kteří často cestují, a vyhledávání v cizím jazyce by mohl být problém.

Aktuální informace o dopravě

Jde vlastně o podobný způsob jako předchozí příklad, kdy aplikace zjistí polohu uživatele a díky tomu nabídne možnosti dopravního spojení v jeho okolí [13]. Další možností je zobrazování aktuálních zpráv o dopravní situaci v okolí uživatele.

VoIP technologie

Geolokace ve VoIP technologiích ještě nejsou běžné, přesto je zde poměrně dobré využití, především v případech, kdy je volající v ohrožení a není schopen udat svoji polohu (šok při autonehodě, dusí se kouřem, nebo upadá do bezvědomí). Obsluha v tu chvíli potřebuje co nejdříve zjistit polohu volajícího, aby mohla vyslat pomoc.

Ochrana před zneužitím

Této technologii využívají především servery, operující s platebními kartami, případně samotné banky. Uživatel zadá při registraci místo bydliště a při manipulaci s penězi pomocí on-line platebních terminálů se ověřuje lokace stanice, z které požadavek k transakci pochází. Pokud tedy bydlím trvale v Brně a bance přijde žádost o platbu z Číny, mohu být varován, případně bude celý požadavek pozdržen. Konkrétní postup již záleží na organizaci. Tuto technologii využívají i některé herní společnosti, například Battle.net, pro ochranu před krádeží hesla.

Hry

Vzhledem k lidské povaze je logicky dalším z uplatnění geolokace zábava. Uživatel si sám ukládá svoji polohu prostřednictvím chytrých mobilních telefonů s geolokací na sociálních geolokačních službách (Foursquare, Gowalla, Facebook Places), čímž může dát svým přátelům na vědomí, kde zrovna je a co dělá. Těchto aplikací využil i marketing a některá zařízení nabízí za „přihlášení“ (tzv. check-in) v jejich prostorách různé výhody, někdy za pouhý check-in, jindy za získanou hodnotu (tu

získáte určitým počtem přihlášení na jednom místě). Dalším využitím jsou odměny v podobě virtuálních odznaků za plnění úkolů, kde lze přikládat fotografie [2]

1.3 Pasivní metody

Podstatou pasivních geolokačních metod je využívání databází, které obsahují požadované informace. Protože tyto databáze jsou velmi rozsáhlé, je poměrně náročné a nákladné udržet je aktuální. Z toho důvodu je většina kvalitních databází komerční. Vyskytují se v nich jak chyby vzniklé při manuálním zápisu, tak chyby způsobené nezaregistrovanou změnou IP adresy, kdy jedna IP adresa může mít přiděleno i více pozic. Další problém nastane v blízké době, kdy bude zaměněna již nedostačující IP v4 za delší IP v6. Změna povede k nutnému zvětšení databází, což může přinést další problémy s chybovostí záznamů. Přesto jsou tyto databáze často využívány, pro jejich rychlost a relativní spolehlivost.

Pasivní geolokační metody lze rozdělit podle využívaných informací na:

- geolokace na základě IP adresy,
- geolokace na základě domén,
- geolokace na základě Wi-Fi.

1.3.1 Geolokace na základě IP adresy

Patří mezi jednodušší metody geolokace, protože IP adresa síťového zařízení, či brány, přes kterou se daná stanice připojuje k internetu, je většinou snadno přístupná [13]. Rozdělování IP adres má na starost organizace IANA (*Internet Assigned Numbers Authority*), která je delegována pro různé části světa organizací RIR (*Regional Internet registry*). Tato organizace zastřešuje koordinační střediska pro kontinenty a jejich další podsložky. Rozdělení je následovné:

- Americký Registr pro Internet Numbers (ARIN),
- Evropské síťové koordinační centrum (RIPE NCC),
- Asijsko-Pacifické síťové informační centrum (APNIC),
- Latinsko Americké a Karibické Internetové adresy registrů (LACNIC),
- Africké síťové informační centrum (AfriNIC).

Nás logicky zajímá RIPE NCC. Tato organizace zahrnuje další podsložky, kde se dělením na regiony dostaneme přímo ke společnostem pro dané země. Jak již bylo řečeno, existují různé databáze, z nichž ty kvalitnější jsou soukromé. Mezi nejznámější veřejné databáze patří Whois a GeoIP (veřejná je pouze databáze pro státy).

Databáze Whois

Je nejznámější veřejná databáze, která se zaměřuje na uchování informací o majitelích internetových domén a IP adres, jako je jméno vlastníka doména, kontaktní adresa, e-mail, apod. O registraci údajů do databáze IP adres se starají regionální centra RIR, podléhající organizaci IANA. V případě domén to jsou národní registry, kde každý národní registrátor vede vlastní Whois server.

Do databáze se dá přistupovat pomocí webových stránek, kde se do pole pro vyhledávání vypíše IP adresa. Aplikace stránky zprostředkuje dotaz na whois server a vypíše odpověď. Další možností je využít v unixových systémech konzole, zadáním příkazu „whois“ a následně hledané IP adresy. V prostředí Windows je zapotřebí stáhnout utilitu Whois, která se dá spustit obdobným způsobem. Velkou nevýhodou této databáze je nejednotnost výstupu, který dostáváme formou textového výpisu. Odlišné registry mohou podat jiný formát výstupu, což poměrně ztěžuje programovou selekci lokace a její rozumnější interpretaci. Některé záznamy tedy mohou obsahovat jak zemi, město, ulici i s číslem popisným, jiné vypíšou pouze zemi. Někdy je pod adresou uveden název firmy apod.

Z hlediska geolokace tedy nejde o přesnou metodu, protože se nemůžeme spolehnout na jednotnost a úplnost záznamu. Protože jde o databázi shromažďující informace o majiteli IP adresy, případně domény, nemusí být kontaktní adresa shodná s umístěním serveru, čímž se přesnost výsledku dostává na úroveň nalezení země. Následný převod adresy na GPS souřadnice musí obstarat jiná utilita.

Příklad výpisu z odpovědi na dotaz `whois 77.251.170.54` (herní server v Nizozemsku) a `whois 80.239.168.0` (herní server ve Francii):

```
% Information related to '77.251.162.0 - 77.251.173.255'  
  
inetnum:      77.251.162.0 - 77.251.173.255  
netname:      UPC-NL  
descr:        CPE Customers NL  
country:      NL  
admin-c:      LGI-RIPE  
tech-c:       LGI-RIPE  
status:       ASSIGNED PA  
remarks:      Contact abuse@upc.nl concerning criminal  
remarks:      activities like spam, hacks, portscans  
mnt-by:       MNT-LGI  
source:       RIPE # Filtered
```

Obr. 1.1: Odpověď whois na IP adresu 77.251.170.54.

```

% Information related to '80.239.186.0 - 80.239.186.255'

inetnum:      80.239.186.0 - 80.239.186.255
netname:      FR-BLIZZARD
descr:        Blizzard Entertainment
org:          ORG-BE1-RIPE
country:      FR
admin-c:      AL5843-RIPE
tech-c:       AL5843-RIPE
status:       ASSIGNED PA
mnt-by:       TELIANET-LIR
source:       RIPE # Filtered

```

Obr. 1.2: Odpověď whois na IP adresu 80.239.168.0.

Databáze GeoIP

Je vyvinuta americkou společností MaxMind, která se stará o další vývoj a aktuálnost databáze. Jde o placenou službu, která nabízí několik stupňů přesnosti, od určení měst až po určení země. Databáze je logicky členěna dle zeměpisných lokací s odpovídajícími bloky IP adres, což urychluje vyhledávání a snižuje potřebnou velikost databáze [4].

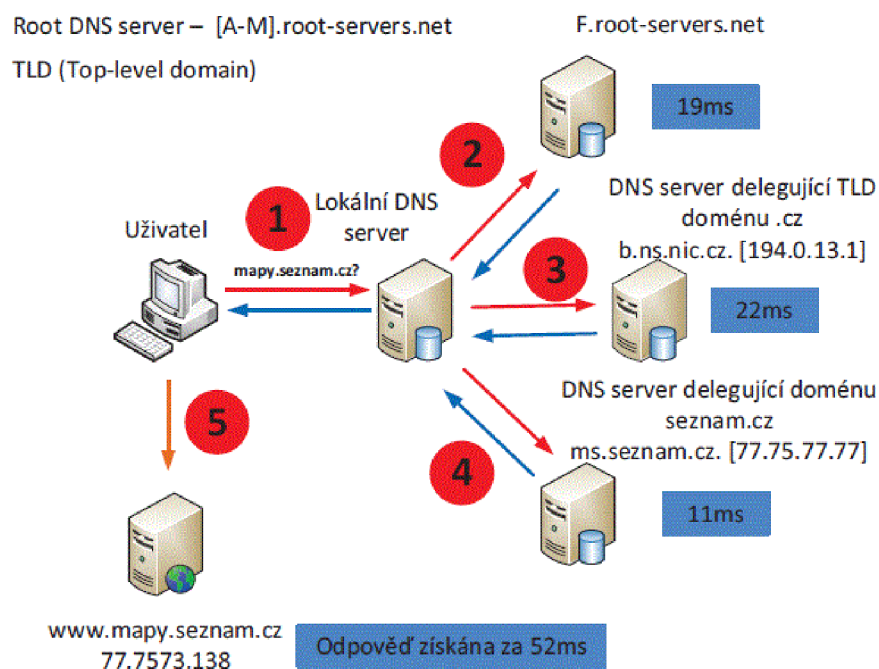
Klient si buď zakoupí licenci a může plně využít dané databáze, nebo si předplatí počet dotazů do databáze. Pak může propojit databázi například s aplikací na webových stránkách, jejíž skript bude vyhledávat pozici uživatele a následně se přizpůsobí k jeho potřebám. Kupříkladu stránky nahrávací společnosti, které určí polohu uživatele z Francie, přepnou jazyk na francouzštinu a změní cenu dopravy, případně i měnu.

MaxMind nabízí i veřejnou variantu GeoLite, která má oproti placené verzi přesnost pouze na zemi původu, což bývá pro některé aplikace dostatečné. Nevýhodou GeoIP databáze je důvěryhodnost získaných informací. Registrace IP adres probíhá skrz různé registrační formuláře, které mohou obsahovat chybné údaje. Proto je nutné takové údaje před zapsáním do databáze napřed ověřovat. Firma MaxMind udává přesnost databáze na 99,8 % v úrovni země, 90 % na úrovni regionu a 80 % na úrovni města [16].

Výsledný rozdíl v přesnosti mezi GeoIP a GeoLite je patrný z následujícího testu. Placená verze našla polohu Kvasice, což je vzdálenost přibližně 8 km od správné fyzické polohy cíle. Neplacená verze určila za lokaci město Zdounky, čímž se vzdálila od správné fyzické polohy cíle o 11 km. V obou případech jde o přesnou lokaci, která je pro většinu civilních potřeb zcela dostačující [4].

1.3.2 Geolokace na základě domén

Dalším způsobem zjišťování polohy cíle je odhad na základě domény. DNS (*Domain Name System*) je hierarchický systém doménových jmen, kde DNS servery slouží pro překlad IP adresy (např.: 80.239.168.0) na lépe zapamatovatelné doménové názvy (např.: www.battle.net), přičemž překlad může fungovat i obráceně. Lépe je vše vidět na obrázku 1.3, kde uživatel zadal adresu www.seznam.cz:



Obr. 1.3: Princip DNS při zadání adresy www.seznam.cz. Převzato z [4], str. 15.

Hierarchický zápis doménového jména odděluje pomocí tečky domény vyšších a nižších řádů. Na konci (nejvíce vpravo) je zapsána doména nejvyššího řádu (nejméně konkrétní) a směrem vlevo se doména konkretizuje. Jedním z možných způsobů využití DNS je hrubý odhad země pomocí domény nejvyššího řádu. Pokud jméno končí příponou některé ze zemí (třeba `.cz`) nebo unie (třeba `.eu`), je odhad snadný. Dalším nástrojem může být i databáze Whois, která vede záznam doménových jmen, nebo využití nástroje `nslookup`. Tento nástroj příkazové řádky využívá lokální DNS knihovnu z operačního systému a funguje na dvou módech – interaktivním a neinteraktivním. Slouží především pro dotazování na doménové jméno, IP adresu a výpisu dalších DNS parametrů [10]. Výpis pro dotaz `nslookup` na www.battle.net můžeme ověřit z dřívějšího výpisu Whois1.2:


```
benedikt@Benedictus-VirtualBox:~$ nslookup www.battle.net
Server:          127.0.1.1
Address:         127.0.1.1#53

Non-authoritative answer:
Name:   www.battle.net
Address: 80.239.186.26
```

Obr. 1.4: Výpis nástroje `nslookup` pro doménu `www.battle.net`.

1.4 Aktivní metody

Aktivní metody využívají k určení polohy informace, naměřené v internetové síti, mezi cílovou stanicí a sondou. Mezi tyto data může patřit zpoždění mezi různými částmi sítě, ležící v cestě od vysílající stanice k cíli. Hlavním rozdílem mezi pasivní a aktivní metodou je tedy získávání vlastních dat měřením aktivní metodou. Získaná data se neporovnávají s databází, ale jsou dále zpracována algoritmy, které za pomoci referenčních bodů a sond odhadnou lokaci cílové stanice, případně nejbližšího uzlu. Mezi nejčastější měření patří měření zpoždění a to:

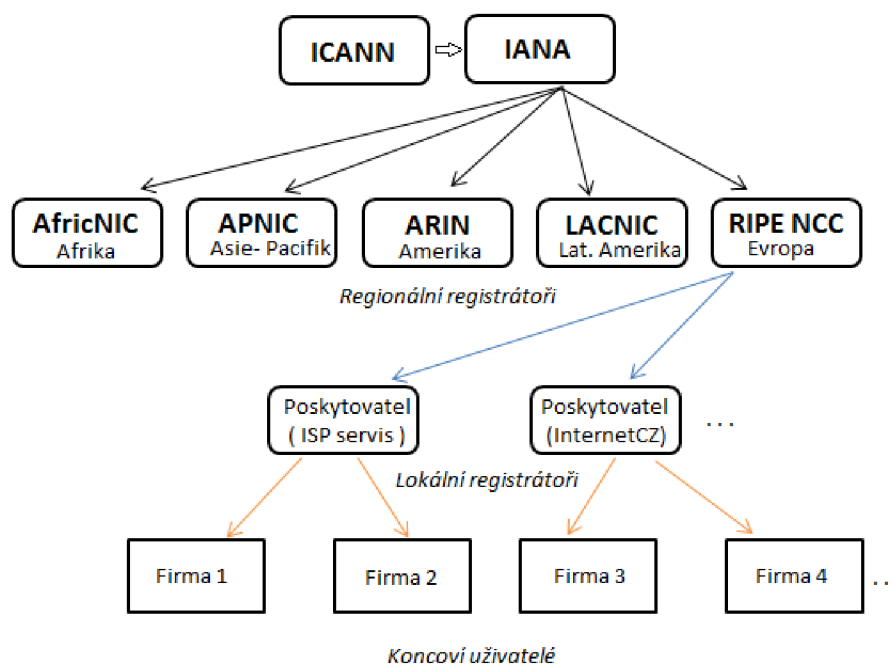
- zpoždění na koncových zařízeních,
- zpoždění na mezilehlých uzlech,
- zpoždění na přenosových linkách.

Nejčastěji používanými nástroji pro měření přenosového zpoždění a sledování cesty k cíli jsou `ping` a `traceroute`. Tyto nástroje měří hodnotu zpoždění při cestě od zdroje signálu k cíli a zpět, protože tyto cesty nemusejí být totožné. Taková hodnota se potom nazývá RTT (*Round-Trip Time*) [13].

2 ZPŮSOB PŘÍŘAZOVÁNÍ IP ADRES

Organizované přiřazování IP adres, doménových jmen a čísel autonomních systémů je nezbytné pro správnou funkci sítě, včetně internetu. Z toho důvodu bylo založeno několik mezinárodních organizací, s hierarchickou strukturou správy, které celosvětově řídí jednoznačné přidělování IP adres, doménových jmen a čísel autonomních systémů.

Mezinárodní organizace jsou uspořádány stromově, podle světového působení a důležitosti, jak je znázorněno na obrázku 2.1:



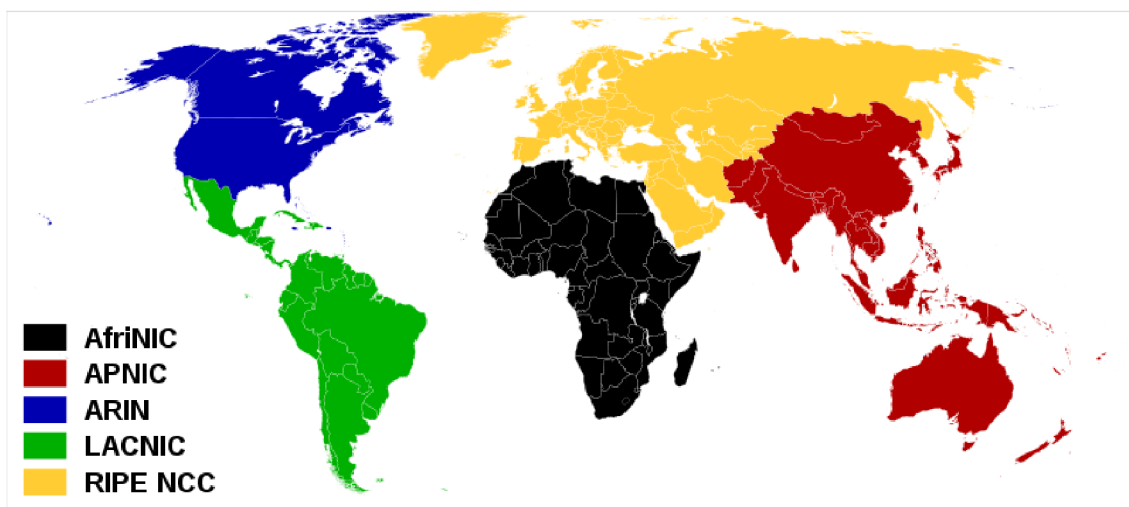
Obr. 2.1: Stromová struktura dělení mezinárodních organizací pro rozdělování IP adres.

2.1 Rozdělení organizací

Nejvyšším orgánem přiřazování zmíněných parametrů internetu je IANA (*The Internet Assigned Number Authority*), kterou v současnosti řídí nezisková organizace ICANN (*Internet Corporation for Assigned Names and Numbers*), sídlící v Kalifornii.

ICANN dohlíží na chod IANA a podřízených organizací, provádí registraci generických domén. Byla také zřízena kvůli ochraně stability internetu, podpory soutěže a rozvoj pravidel ochrany internetu.

IANA tedy jednoznačně udává rozmezí čísel pro IP adresy, doménové jména apod. a ty dále přiděluje svým podřízeným organizacím, které se obecně označují jako regionální registry RIR (*Regional Internet Registries*). Každý regionální registr spravuje velkou oblast, většinou kontinent. Svět je tímto způsobem rozdělen na několik regionů, spravovaných regionálními registry. V každém konkrétním regionu je větší skupina lokálních registrů, podřízených RIR, což jsou především poskytovatelé internetového připojení, telekomunikační společnosti a velké firmy zainteresované v telekomunikaci, finančně se podílející na činnosti regionální RIR [12]. Rozdělení regionů je vidět na obrázku 2.2:



Obr. 2.2: Celosvětové rozdělení regionů[22].

Regionální registry s rozdělením pro příslušné regiony:

- Americký Registr pro Internet Numbers (ARIN),
- Evropské síťové koordinační centrum (RIPE NCC),
- Asijsko-Pacifické síťové informační centrum (APNIC),
- Latinsko Americké a Karibické Internetové adresy registrů (LACNIC),
- Africké síťové informační centrum (AfriNIC).

Lokální registry komunikují s koncovým uživatelem, což bývá síťové oddělení podniku, kterému je přidělen rozsah IP adres, subdomén apod. Komunikace probíhá většinou směrem od koncového uživatele k lokálnímu registrátorovi, který posílá přeformulovanou žádost k regionálnímu registrátorovi. Zde je žádost zpracována, nejčastěji programem (automaticky) a vrácena odpověď s přiděleným rozsahem dotazované domény, IP adresy apod.

2.2 RIPE NCC

RIPE NCC (*Réseaux IP Européens Network Coordination Centre*) je mezinárodní nezisková organizace, fungující jako regionální internetový registr pro Evropu, Blízký východ a část Asie (*viz obrázek 2.2*). Bližší seznam zemí je možné nalézt na stránkách RIPE NCC¹. Kromě rozdělování IP adres, domén apod. také vede veřejně přístupnou databázi RIPE. Do ní ukládá údaje o internetových sítích spravovaného regionu, které dokáže provázat s informacemi o doménách nejvyššího řádu z této oblasti (ty ovšem spravuje správce domén nejvyšší úrovně, u nás jde o cz.nic). V databázi jsou uloženy informace o IP adrese, vlastníkově, a v nesjednocené formě i podrobnější záznamy, jako kontaktní adresa, či telefonní číslo. Tyto informace se dají zjistit pomocí služby Whois, která byla již popsána v předcházejících kapitolách.

2.3 Rozdělení adres

Rozdělení bloků IP adres třídy C organizací IANA byl původně rozdělen s přesvědčením, že nebudou vyčerpány. Rozdělení udává RFC1466 [6] a je následovné:

Oblast využití	Rozsah IP adres
Multi-regional	192.0.0.0 – 193.255.255.255
Europe	194.0.0.0 – 195.255.255.255
Others	196.0.0.0 – 197.255.255.255
Various Registries	198.0.0.0 – 198.255.255.255
North America	199.0.0.0 – 199.255.255.255
Central/South America	200.0.0.0 – 201.255.255.255
Pacific Rim	202.0.0.0 – 203.255.255.255
North America	204.0.0.0 – 209.255.255.255
Pacific Rim	210.0.0.0 – 211.255.255.255
Europe	212.0.0.0 – 213.255.255.255
US-DOD	214.0.0.0 – 215.255.255.255
North America	216.0.0.0 – 216.255.255.255
Reserved	217.0.0.0 – 223.255.255.255

Tab. 2.1: Rozdělení bloků IP adres regionům.

¹Seznam zemí spadajících pod RIPE NCC: <https://www.ripe.net/membership/indices/>

Pro RIPE NCC je rozsah bloků s prefixem /8 v současnosti následující [11]:

Prefix	Designation	Date
077/8	RIPE NCC	2006-08
078/8	RIPE NCC	2006-08
079/8	RIPE NCC	2006-08
080/8	RIPE NCC	2001-04
081/8	RIPE NCC	2001-04
082/8	RIPE NCC	2002-11
083/8	RIPE NCC	2003-11
084/8	RIPE NCC	2003-11
085/8	RIPE NCC	2004-04
086/8	RIPE NCC	2004-04
087/8	RIPE NCC	2004-04
088/8	RIPE NCC	2004-04
089/8	RIPE NCC	2005-06
090/8	RIPE NCC	2005-06
091/8	RIPE NCC	2005-06
092/8	RIPE NCC	2007-03
093/8	RIPE NCC	2007-03
094/8	RIPE NCC	2007-03
095/8	RIPE NCC	2007-07

Tab. 2.2: Rozsah bloků IPv4 adres pro RIPE NCC.

Problémem je nedostatek volných IPv4 adres, kterých zbývá přibližně pouhé 4 miliony (ke dni 5.9.2012), pro RIPE NCC předtím, než se otevře poslední volný blok prefixu 185.0.0.0/8. Poté nastane vyčerpávání posledního prefixu volných adres pro region Evropy a Blízkého východu, proto vydala RIPE NCC sadu zvláštních pravidel pro přidělování IP adres. Cílem těchto pravidel je zpřísnit přidělování IP adres a maximálně využít zbylé adresy. Předpokládaná měsíční spotřeba IPv4 adres jsou totiž ony poslední 4 miliony. RIPE NCC zveřejnila graf využití adres, který je denně aktualizován ². Obdobná situace již nastala u registru APNIC, kde v dubnu 2011 došli volné adresy.

² Sledovat tento graf je možné z [:http://www.ripe.net/internet-coordination/ipv4-exhaustion/ipv4-available-pool-graph](http://www.ripe.net/internet-coordination/ipv4-exhaustion/ipv4-available-pool-graph)

3 GLOBAL POSITIONING SYSTEM

3.1 Úvod do technologie GPS

Systém GPS je součástí globálního družicového polohovacího systému (GNSS) a potomek NAVSAT (někdy také Transit). Původně se jednalo o čistě vojenský projekt a GPS je stále provozováno Ministerstvem obrany Spojených států amerických. Jeho vývoj začal v letech 1973 pro účely americké armády. Veřejnosti byl uvolněn po roce 1983, kdy bylo omylem sestřeleno civilní letadlo v sovětském vzdušném prostoru¹. Systém se postupně rozšiřoval po celém světě, až do roku 1994, kdy bylo GPS dostupné po celém světě. Z důvodu bezpečnosti byl systém zatížen úmyslnou odchylkou, která neumožňovala přesnější určení polohy a zaměření cíle. V roce 2000 byla tato odchylka odstraněna². GPS umožňuje určení polohy kdekoliv na zemi s přesností cca 15m (ovšem je možné dosáhnout až na jednotky cm), rychle se rozšířilo do mnoha oblastí lidské činnosti a jsou vyvíjeny další alternativy, jako je evropský projekt Galileo [5], či ruská GLONASS [1].

3.2 Struktura a funkce systému

Způsob fungování systému GPS je postaven na jednostranné komunikaci mezi družicí a uživatelem, pomocí složitého signálu. Uživatel disponuje přijímačem, který obdrží od družice informace o své poloze a přibližné poloze ostatních družic. Na základě toho se pak počítá aktuální pozice přijímače, tzv. pseudovzdálenost, což je termín zavedený pro hrubější výpočet polohy, který je následně zapotřebí upřesnit doplňujícími výpočty. Pseudovzdálenost se určí jako vzdálenost mezi přijímačem a viditelnými družicemi – nacházející se nad obzorem. Tento výpočet zahrnuje i znalost rychlosti šíření družicového signálu a doby mezi vysláním a příjmem signálu. Pro určení dvojrozměrné polohy je potřeba minimálně tři družice, pro výpočet třírozměrné polohy je zapotřebí minimálně čtyři družice. Ve zkratce tedy lze říci, že poloha se určuje pomocí vzdálenosti mezi přijímačem a alespoň třemi družicemi, vypočtenou jako zpoždění signálu mezi družicí a přijímačem.

¹Sestřelené letadlo byl Boeing 747 společnosti Korean Air. Dodnes nebylo objasněno, proč se letadlo po vzletu odchýlilo ze své trasy a vletělo do vzdušného prostoru Sovětského svazu.

²Americká armáda vyvinula technologii pro rušení GPS signálu [25].

Obecně se systém GPS dělí na 3 segmenty [9]:

- kosmický,
- řídicí,
- uživatelský.

3.2.1 Kosmický segment

Původně byl kosmický segment složen z 24 družic, přičemž 3 sloužili jako záložní. Později počet družic vzrostl na 32, což je maximální počet. Tyto družice krouží kolem Země po šesti kruhových drahách se sklonem 55° k rovině rovníku a vzájemně posunutých o 60° , ve výšce kolem 20 190 km, rychlostí $11\,300\text{ km} \cdot \text{h}^{-1}$ a jeden oběh trvá 11 h a 58 minut [23]. Každá družice disponuje vysílačem, přijímačem, atomovými hodinami a dalšími přístroji, které slouží k navigaci nebo jiným potřebným úkolům. Družice jsou ovládány z pozemního řídicího centra, které přijímá a vysílá informace potřebné k správnému chodu sítě družic.

3.2.2 Řídicí segment

Slouží k řízení a správě kosmického segmentu, tedy k monitorování stavu družic a jejich atomových hodin, vyhodnocování informací, korekci dráhy letu, synchronizaci atomových hodin, úpravu signálu družic. Dříve také zajišťovali zhoršení přesnosti GPS. Sestává z pěti monitorovacích stanic – rovnoměrně rozmístěných po obvodu země, čtyř pozemních vysílačů a z Hlavního řídicího centra v Colorado Springs.

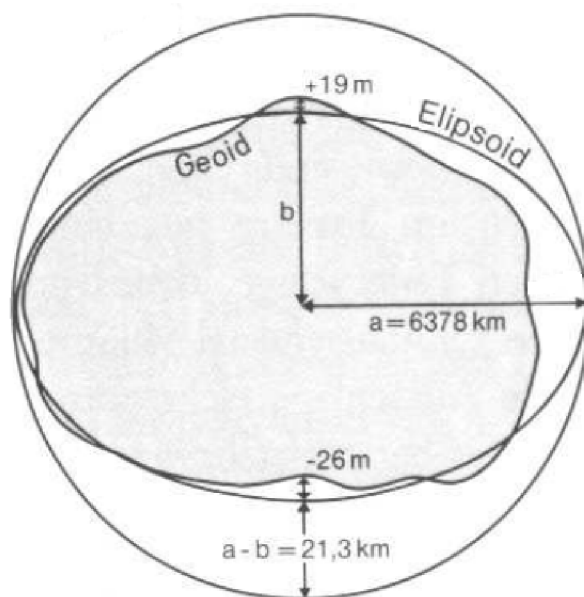
3.2.3 Uživatelský segment

Jde o přijímače GPS signálů, které vyhodnocují signály z několika družic a následně zjišťují svoji polohu výpočtem rovnic o více neznámých. Pro určení správné polohy je zapotřebí minimálně 4 rovnic, kvůli nesynchronizovaným hodinám družice a přijímače. Odchylka 1 milisekundy dá chybu přibližně 300 km [25], což je nepřijatelné. Z tohoto důvodu je v GPS signálu přenášena informace o odchylce velmi přesného času z atomových hodin družice od systémového času GPS a z této zprávy je dále počítán rozdíl mezi časem přijímače a časem satelitu. Problémem lokalizace pomocí GPS je členitý terén, který může zakrýt část oblohy (tento jev se nazývá elevační maska) a tak omezit počet dostupných viditelných satelitů.

4 URČOVÁNÍ POZICE A VZDÁLENOSTI NA ZEMĚKOULI

4.1 Matematický popis zeměkoule

Základním problémem při definici přesného geodetického (souřadnicového) systému je tvar Země, která je z geometrického hlediska fyzikální těleso, jehož tvar je důsledkem působení dvojice sil, a to přitažlivé síly a odstředivé síly [3]. Protože Země není koule a je nutné přihlídnout k členitosti a nepravidelnosti povrchu, je velmi obtížné matematicky popsat její skutečný tvar. Nejvíce se podobá zemskému tělesu geoid, který ovšem není příliš vhodný jako vztažná výpočetní plocha pro definici geodetického systému. Z toho důvodu se pro potřeby mapování využívá náhrady geoidu za referenční plochy, z nichž nejpoužívanější je rotační elipsoid, který je definován dvěma základními parametry: délkou hlavní a vedlejší poloosy. V současnosti využíváme geodetického systému WGS 84, který definuje elipsoid nejvíce podobný zemskému tělesu[20].



Obr. 4.1: Rozdíl mezi geoidem, elipsoidem a koulí [3].

4.1.1 Geoid

Jde o fyzikální model povrchu Země, který lze z hlediska geografie definovat jako těleso, které atmosféra odděluje od střední klidové hladiny oceánů. Tedy jde o mo-

del, který průměruje nepravidelnosti zemského povrchu. Vědní obor, zabývající se zkoumáním geoidu, se nazývá geodézie.

4.1.2 Rotační elipsoid

Je rotační těleso zploštělé na pólech, které je určeno dvěma konstantami, nejčastěji hlavní a vedlejší poloosou, ale může se vyskytnout i parametr zploštění, či první excentricity. Využívá se jako referenční plocha pro geodetické systémy, a v takovém případě je střed elipsoidu ztotožněn s těžištěm geoidu. Podle vedlejší osy určujeme dva typy referenčního elipsoidu; pokud je vedlejší osa ztotožněná s osou rotace Země, pak jde o zemský elipsoid, je-li vedlejší osa rovnoběžná s osou rotace Země, jde o referenční elipsoid.

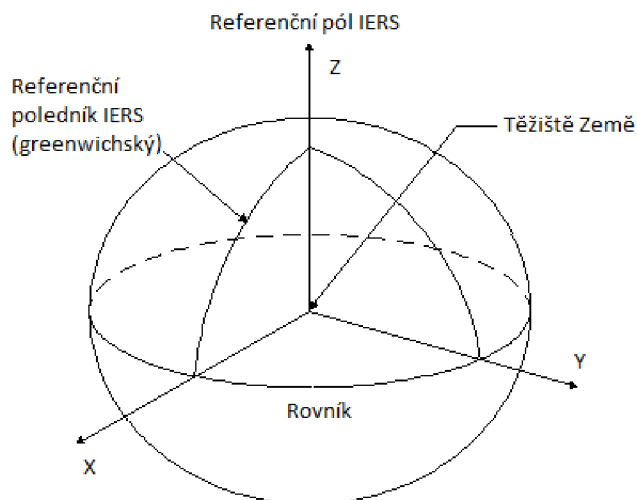
4.2 Geodetický systém WGS 84

Jedná se o vojenský souřadnicový systém rozšířený v Evropě, mezi státy NATO. Byl definován na základě referenční plochy – referenčního elipsoidu WGS 84 (*World Geodetic System*) [3]. V roce 1984 proběhlo satelitní měření a pomocí tehdejších nejnovějších technologií bylo nashromážděno dostatečné množství informací pro vytvoření nového rotačního elipsoidu, který se ukázal být dostatečně přesným.

Definiční parametry elipsoidu WGS 84:

- Hlavní (velká) poloosa: 6 378 137 m.
- Úhlová rychlost rotace Země: $7\,292\,115 \cdot 10^{-11} \text{ rad} \cdot \text{s}^{-1}$.
- Geocentrická gravitační konstanta: $3\,986\,004,418 \cdot 10^8 \text{ m}^3 \cdot \text{s}^{-2}$.
- Reciproká hodnota zploštění: 298, 257 223 563.
- Vedlejší poloosa: 6 356 752 m.

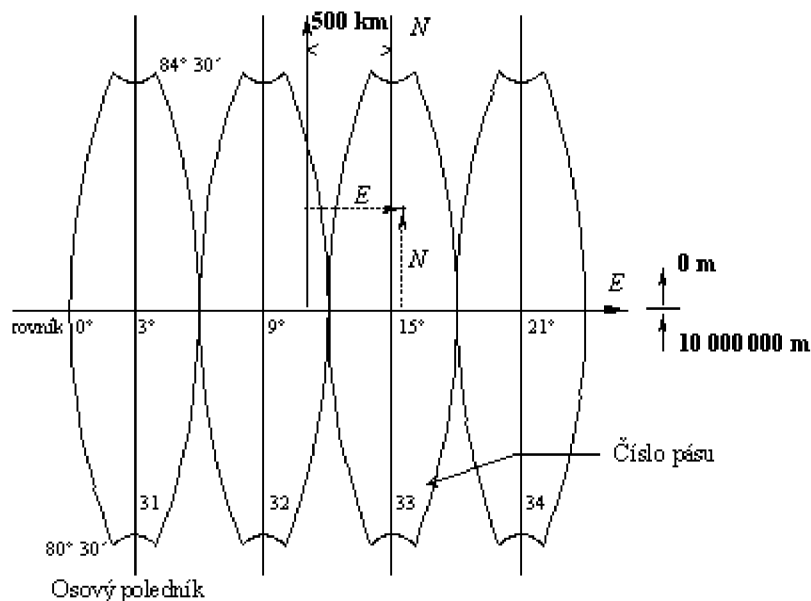
Souřadnicový systém má umístěn počátek souřadnic ve středu Země, jde tedy o geocentrický systém. Osu Z má ztotožněnou s osou rotace Země z roku 1984 [3]. Jako referenční poledník je zvolen poledník Greenwichský, v jehož průsečnici s rovníkem je umístěna právě osa X. Osa Y leží v rovině rovníku 90° východně od osy X. Pro úplné propojení elipsoidu se Zemí se využívá tzv. definičních bodů. Což jsou pevně stabilizované body, jejichž souřadnice tvoří referenční rámec. Systém WGS 84 má tyto definiční body rozmístěné po celém povrchu Země a tak jde o systém globální. Použité kartografické zobrazení je systém UTM.



Obr. 4.2: Světový geodetický systém WGS 84, převzato z [20].

4.3 Kartografický zobrazovací systém UTM

Pro snadnější orientaci v mapách se využívá zobrazování souřadnic v rovině, namísto prostorových souřadnic vynesných na glóbu. Pro převod referenčního elipsoidu a jemu přiřazený systém souřadnic na rovinné zobrazení slouží metody zvané kartografická zobrazení. U takového převodu ale dochází ke zkreslení délek a ploch, zato úhly zůstávají nezměněny, takové zobrazení se označuje za konformní. NATO používá v topografii universální transversální Mercatorovo zobrazení, zkracované jako UTM (*Universal Transverse Mercator*). Jedná se o konformní příčné válcové kartografické zobrazení 6° poledníkových pásů s délkovým zkreslením na osovém poledníku 0,996. Osové poledníky jsou 3° , 9° , 15° , 21° atd. [3] Celková plocha referenčního elipsoidu je tedy rozdělena na 36 pásů. Když rozvineme plášť válce, vznikne rovinné zobrazení. Takto zobrazuje systém UTM povrch Země mezi 80° jižní zeměpisné šířky a 84° severní zeměpisné šířky [20]. Souřadnice narůstají od počátku směrem na východ a na sever. Při zobrazování UTM získáme vodorovnou osu E pro východní směr a svislou osu N pro severní směr. Pro odstranění záporných hodnot je v případě potřeby přičítána konstanta 500 km na ose E (tzv. False Easting) a počátek pásu pro severní polokouli tak bude 500 km E a 0 km N. Stejným principem je pak jižní polokoule upravena, akorát pro souřadnici N se přičte hodnota 10 000 km (tzv. False Northing)[3].



Obr. 4.3: Souřadnicové zobrazení UTM [3].

4.4 Určení vzdálenosti mezi dvěma body

Běžnou praxí u navigačních přístrojů u výpočtu vzdálenosti mezi dvěma body na mapě, jsou algoritmy pracující s modelem Země jako s koulí, což přináší určitou chybu. Na druhou stranu jsou to výpočty rychlé a nenáročné. Mezi tyto výpočetní modely se řadí algoritmus Great Circle, kde se počítá vzdálenost jako nejkratší spojnice dvou bodů na kulové ploše (ortodroma), která leží na kružnici se středem v jejím těžišti. Obdobným způsobem výpočtu je algoritmus Haversine, u kterého se udává chyba přibližně 3 m na 1 km.

Haversine algoritmus využitý pro výpočet vzdálenosti mezi body A a B je zapsán následovně [15]:

$$a = \sin^2\left(\frac{\Delta\text{lat}}{2}\right) + \cos(\text{lat}_1) \cdot \cos(\text{lat}_2) \cdot \sin^2\left(\frac{\Delta\text{long}}{2}\right) \quad (4.1)$$

$$c = 2 \cdot \arctan(\sqrt{a}, \sqrt{1-a}) \quad (4.2)$$

Arcus tangens je úhel mezi osou x a přímkou obsahující počátek (0;0) a bod o souřadnicích (x,y). Tento úhel je udán v radiánech v intervalu $-\pi$ až π , s výjimkou hodnoty $-\pi$ [17]. V případě rovnice 4.2 dosazujeme za souřadnice (x,y) úpravy proměnné a.

$$d = R \cdot c \quad (4.3)$$

Kde je:

R ... poloměr Země,

$\Delta\text{lat} = \text{lat}_2 - \text{lat}_1$... rozdíl mezi zeměpisnou šířkou bodů A a B,

$\Delta\text{long} = \text{long}_2 - \text{long}_1$... rozdíl mezi zeměpisnou délkou bodů A a B.

Mnohem přesnějšího výsledku dosáhneme tak, že použijeme model nejvíce podobný geoidu, tedy rotační elipsoid. S takovým algoritmem přišel v roce 1975 americký geodet Thaddeus Vincenty a díky novému způsobu výpočtu dosáhl přesnosti až 0,5 mm v rámci celého elipsoidu [21]. Nevýhodou Vincenty je složitost výpočtu. Z důvodu rozsáhlosti řešení výpočtu Vincentyho rovnice uvádím pouze základní tvar¹

$$\sin \sigma = \sqrt{(\cos U_2 \sin \lambda)^2 + (\cos U_1 \sin U_2 - \sin U_1 \cos U_2 \cos \lambda)^2} \quad (4.4)$$

Kde je:

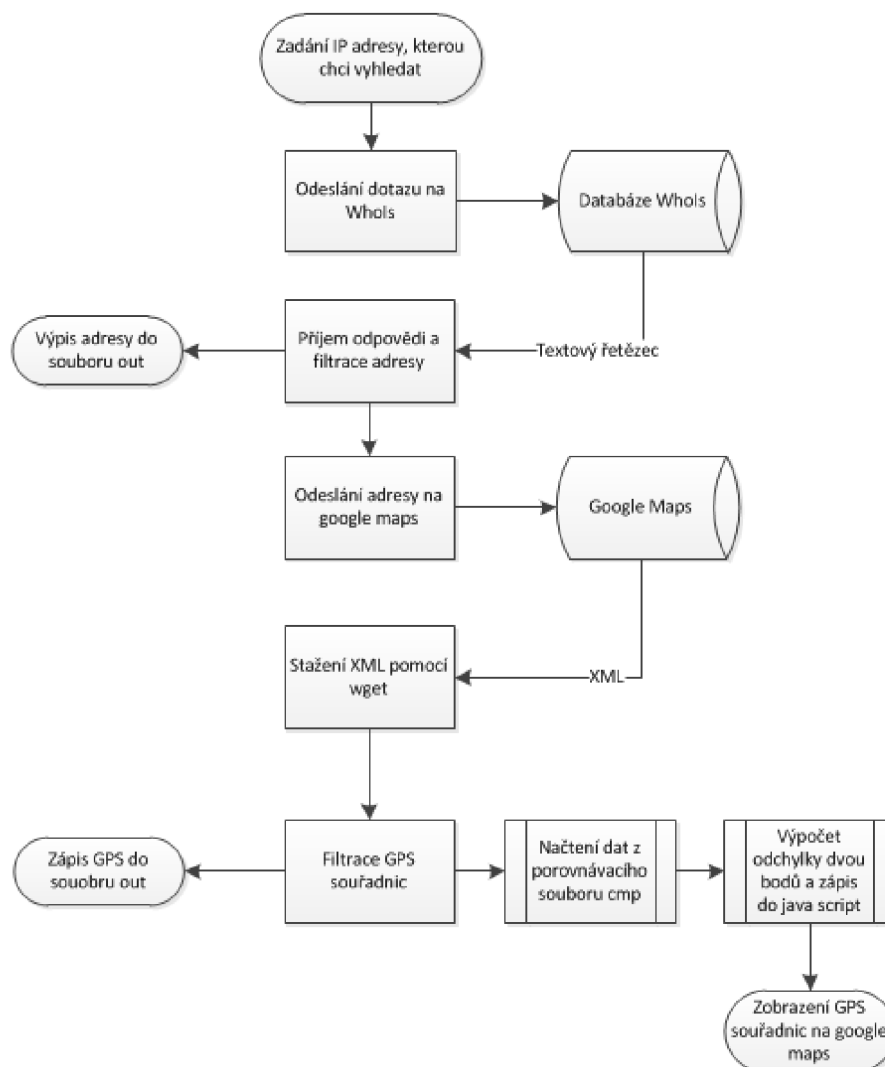
U ... zkrácená zeměpisná šířka

$\lambda = L$... zeměpisná délka v pomocné oblasti, první aproximace

¹Velmi dobře je problém popsán na stránce <http://www.movable-type.co.uk/scripts/latlong-vincenty-direct.html>, je zde i návrh implementace do zdrojového kódu.

5 POPIS PROGRAMU

Hlavním cílem této práce je zjistit metody lokalizace stanic v síti internet pomocí IP adresy a konkrétně se zaměřit na databázi Whois. Jak již bylo zmíněno dříve, patří databáze Whois pod regionální registr RIPE NCC a lze ji využít více cestami. První možností je využít webových aplikací, které nabízí zprostředkování dotazu na Whois a výpisu odpovědi. Druhou možností je využít nástroje `whois`, obsaženého v příkazové řádce operačního systému (pro Windows je zapotřebí stáhnout příslušnou aplikaci nástroje, některé unixové systémy mají `whois` již implementován, u některých je zapotřebí doplnit). Praktickou částí této práce je vytvořit program, který dokáže využít databáze Whois a interpretovat část výsledku, který nás zajímá, tedy geolokaci. Celý program je psán v jazyce C, s využitím prostředí Code::Blocks.



Obr. 5.1: Vývojový diagram programu.

5.1 Způsob přístupu k databázi Whois

Prvním problémem, který je potřeba vyřešit při návrhu algoritmu je způsob přístupu, zda-li bude využit patřičný port Whois, nebo využít nástroje v operačním systému. Protože výsledný program bude umístěn na webové stránky běžící pod Linuxem a přidán do obsáhlejšího kódu, rozhodl jsem se využít konzolového nástroje Whois v unixové struktuře.

5.1.1 Struktura programu

Jádro programu se skládá ze čtyř funkcí. Zde jsou sepsány podle hierarchické důležitosti při volání:

1. **main** – volá funkce `readFile`, `getLocation` a `getIn`, pomocí nich načte soubor, vyšle dotaz na Whois, vyhledá souřadnice a analyzuje výsledek,
2. **readFile** – přečte celý soubor a vrátí ho jako dynamicky alokovaný řetězec,
3. **getIn** – najde a vrátí dynamicky alokovaný řetězec mezi 2 regulárními výrazy `patr1` a `patr2`,
4. **getLocation** – najde GPS souřadnice pro adresu zjištěnou z Whois.

5.1.2 Komunikace klient-server

Zde teoreticky navrhnu možný způsob připojení přes port, pouze s tím rozdílem, že se bude jednat o programovací jazyk Java. Whois komunikuje s klientem přes port 43, proto jej primárně nastavíme do proměnné `port`, kterou budeme dále využívat, a zároveň si definujeme proměnnou `server`.

```
1 String server;  
2 int port = 43;
```

Výpis kódu 5.1: Definice portu pro komunikaci s Whois.

V programovacím jazyce Java je zvykem vytvářet hlavní třídu `main.java` a v ní volat další třídy a objekty, proto se zaměříme při výběru serveru, z kterého chceme dostat odpověď, na hlavní třídu, a zde server definujeme:

```
1 public static void setup(){  
2     Whois = new Whois("whois.arin.net");  
3     String /* [] */ result = Whois.lookup("72.10.47.40");  
4 }
```

Výpis kódu 5.2: Definice serveru se službou Whois.

Následně ve třídě obsahující metody čtení, zápisu apod., pojmenovanou *Whois*, určíme způsob připojení k serveru, zadaného v hlavní třídě *main.java*. Prvně načteme do objektu *server* informaci z hlavní třídy (v našem případě „whois.arin.net“)

```
1 public Whois(String server) {  
2     this.server = server;  
3 }
```

Výpis kódu 5.3: Načtení zadaných parametrů z hlavní třídy.

A poté můžeme zahájit připojení k serveru:

```
1 Socket connection = new Socket(server, port);  
2 PrintStream out = new PrintStream(connection.getOutputStream());  
3 BufferedReader in = new BufferedReader(new InputStreamReader(connection  
    .getInputStream()));
```

Výpis kódu 5.4: Připojení k serveru se službou Whois.

Kde přiřadíme objektu *Socket* parametry serveru a portu, k *PrintStream* přiřadíme parametr předchozí funkce a s pomocí *BufferedReader* budeme číst ze spojení.

5.2 Konzolový nástroj whois

Zbytek popisované činnosti již bude probíhat v jazyce C, protože je v něm psán program, náležící k této semestrální práci. Pokud chceme využít nástroje *whois* z operačního systému, je samotný zápis velmi prostý. Stačí přiřadit funkci *popen* potřebné parametry.

```
FILE * f = popen(command, "r")
```

Funkce *popen()* je obsažena v knihovně *stdio.h*. Při zavolání spustí program, nebo příkaz, zadaný pod *command* a jeho výstup přiřadí k souboru s ukazatelem *f* [24].

r – čtení z příkazu (*command*),

w – zápis do příkazu (*vstupu*).

Tedy konkrétní aplikace je pro nás:

```
1 #define CMD "whois "  
2 strepy(buffer, CMD);  
3 str = readFile ((FILE *) popen(buffer, "r"));
```

Výpis kódu 5.5: Komunikace programu s nástrojem whois.

Kde *readFile* je metoda ke čtení souboru se vstupními daty a *str* je řetězec s načteným výstupem.

5.3 Filtrování výstupu Whois

Při filtraci je potřeba řešit problém nejednotné odpověď z Whois, kde se nedá jistě předpokládat pořadí výpisu adresy, ani jeho forma. Proto jsem využil regulárních výrazů k nalezení určujícího klíče v textu.

5.3.1 Funkce `getIn`

Protože v jazyce C jsem nenašel vyhovující metody pro hledání v řetězci, musel jsem vytvořit vlastní metodu `getIn`, která najde a vrátí dynamicky alokovaný řetězec mezi dvěma regulárními výrazy `patrn1` a `patrn2`. Samotný algoritmus využívající regulárních výrazů k nalezení klíče v textu je poměrně dlouhý, proto je zobrazena pouze funkční část. V již načteném výstupu z Whois hledám pomocí regulárních výrazů `patrn1` a `patrn2` hledaný výraz (klíč) a vracím jeho polohu pomocí `begin` a `end`. Řetězec mezi regulárními výrazy načítám do `str`, což je také výstup metody `getIn`. Nejdůležitější část metody `getIn` je vypsána zde:

```
1  if ((regexec(&rgT1, string + offset, 1, &match, 0)) == 0)
2      { // pokud jsme našli první výraz patrn1
3          offset += (int) match.rm_eo; // pomocný offset ... musíme
4              hledat od pozice kde jsme našli první výraz
5
6          if ((regexec(&rgT2, string + offset, 1, &match, 0)) == 0)
7              { // hledáme druhý výraz od pozice, kde jsme našli první (
8                  string + offset)
9                  *begin = offset; // pokud se povede nalézt i druhý výraz,
10                     uložíme si pozici začátku
11                     *end = (int) match.rm_so + offset; // a konce
12
13                     str = malloc(sizeof(char) * (*end - *begin + 1)); // pak si
14                         vytvoříme prostor pro nalezený řetězec v paměti RAM
15                     strncpy (str, string + *begin, *end - *begin); //
16                         překopírujeme podřetězec, který jsme našli, do str
17                     str[*end - *begin] = '\0'; // nezapomeneme přidat nakonec
18                         ukončovací znak
19                 }
20      }
```

Výpis kódu 5.6: Vyhledávání v textu pomocí regulárních výrazů.

`Offset` slouží jako pomocná proměnná pro posun o písmenko dál, protože prohledávám text po písmenku. Zároveň slouží jako ukazatel pozice pro `begin` a `end`. Jakmile je nalezen klíč, který hledáme, je řetězec (text) uložen do `str` a zakončen ukončovacím znakem `\0`.

5.3.2 Funkce main

Filtrování odpovědi z Whois v hlavní části programu pak probíhá s pomocí funkce `getIn`. Postupně zadáváme jako parametr vyhledávání méně konkrétní udání lokace, protože z Whois můžeme dostat výpis jenom země, nebo pro změnu i ulice s číslem. Nenajde-li program slovo *address*, pokračuje ve vyhledávání *city* apod. Pokud program nalezne hledaný výraz, například *address*, zapíše text za ním následující do `outFile` a před něj přidá text „adresa:“, takto pokračuje ve výpisu na každý nový řádek, dokud nedojde ke konci. Pak je uvolněna paměť `str` a ukazatelů `rgT1`, `rgT2` (nejsou zobrazeny v kódu).

```
1 while ((match = getIn(str, "address:[ ]*", "(\\r\\n|\\n)[^ ]|$", &begin, &
   end, end)) != NULL)
2     while ((match = getIn(str, "address:[ ]*", "(\\r\\n|\\n)[^ ]|$", &
   begin, &end, end)) != NULL)
3         { // smyčka hleda adresu ve vypisu z whois
4             fprintf (outFile, "adresa: %s\\n", match); // zapise adresu
   do outFile
5             free(match); // uvolnime pamet kterou alokovala funkce
   getIn
6         }
7     while ((match = getIn(str, "city:[ ]*", "(\\r\\n|\\n)[^ ]|$", &
   begin, &end, end)) != NULL)
8         { // smyčka hleda mesto ve vypisu z whois
9
10            fprintf (outFile, "mesto: %s\\n", match); // zapise mesto do
   outFile
11            free(match); // uvolnime pamet kterou alokovala funkce
   getIn
12        }
13    while ((match = getIn(str, "country:[ ]*", "(\\r\\n|\\n)[^ ]|$", &
   begin, &end, end)) != NULL)
14        { // smyčka hleda zemi ve vypisu z whois
15
16            fprintf (outFile, "zeme: %s\\n", match); // zapise zemi do
   outFile
17            free(match); // uvolnime pamet kterou alokovala funkce
   getIn
18        }
19    fprintf (outFile, "\\n"); // jen pomocny oddelovac
20    free(str); // uvolnime pamet kterou alokovala funkce readfile
```

Výpis kódu 5.7: Filtrování odpovědi z whois a výpis výsledku.

5.4 Převod adresy na souřadnice GPS

Jakmile program nalezl ve výpisu z databáze Whois fyzickou adresu náležící zadané IP adrese, je zapotřebí převést adresu na souřadnice, s kterými se nadále pracuje. K tomu program využívá nástroj The Google Geocoding API. Pro tuto aplikaci jsou dány dva možné výstupy – JSON a XML. Nás bude zajímat formát XML, který má ale podobné formátování jako JSON.

5.4.1 Google Geocoding API

Google Geocoding API reaguje na dotaz ve správném formátu výpisem XML kódu s předem danou strukturou, proto je poměrně nenáročné filtrovat potřebné informace. Požadavek zadáváme ve formátu `maps.googleapis.com/maps/api/geocode/` a za poslední „/“ vkládáme již adresu, na kterou je veden dotaz. Příkladem může být vyhledávání pro adresu: 600 Amphitheatre Parkway, Mountain View, CA. Dotaz bude vypadat následovně [7]:

```
http://maps.googleapis.com/maps/api/geocode/xml?address=1600
+Amphitheatre+Parkway,+Mountain+View,+CA&sensor=true_or_false
```

Z odpovědi ve formě XML chceme získat pouze souřadnice, které se vždy nacházejí za slovem „<location>“, výsledek by vypadal takto:

```
<location>
  <lat>37.4217550</lat>
  <lng>-122.0846330</lng>
</location>
```

5.4.2 GNU Wget

Pro stažení XML kódu využívá program konzolový nástroj Wget, který je součástí naprosté většiny hlavních distribucí Linuxu. Jde o nástroj pro neinteraktivní stahování souborů – tedy nevyžaduje přihlášení uživatele, běží na pozadí – podporující protokoly HTTP, HTTPS a FTP. Byl navržen pro stahování přes pomalé a nestabilní síťové spojení, pokud by nebyl soubor kompletně stažen, navazuje se stahování v bodě, kde došlo k přerušení. Wget může následovat odkazy stránek z HTML a vytvářet lokální verze vzdálených serverů – přesnou kopii jejich adresářové struktury. Tomuto způsobu se říká rekurzivní stahování [18].

Wget je konzolová aplikace a jako taková má mnoho parametrů (přibližně 100), které vzhledem k původu z konzole GNU mají krátkou i dlouhou formu. Pro účely programu byly použity dva krátké parametry:

```
-O soubor
--output-document=soubor
```

Dokumenty nebudou zapisovány do příslušných souborů, ale všechny budou připojeny do jediného souboru se jménem, určeným v parametru. Když je jméno souboru určeno jako „-“, budou dokumenty zapisovány na standardní výstup. (Pro zápis do souboru pojmenovaného „-“ použijte ./-.) [18]

```
-o logfile
--output-file=logfile
```

Zapisuje všechny zprávy do logfile. Tyto zprávy jsou normálně, zapisovány na standardní chybový výstup [18]. Výsledný zápis pak vypadá následovně:

```
wget -O - -o /dev/null \"maps.googleapis.com/maps/api/geocode/
xml?sensor=false&address=}
```

Soubory budou tedy sloučeny do jediného standardního výstupu umístěného v /dev/null a bude stahovat z adresy

```
maps.googleapis.com/maps/api/geocode/xml?sensor=false&address=
```

kde se ještě doplní hledaná fyzická adresa podle pravidel Google Geocoding API.

Fyzická adresa hledaného IP se zapisuje už přímo v hlavní funkci main při filtrování z výstupu Whois do paměti „address“ postupně od nejpřesnější získané hodnoty (ulice) až po nejméně přesnou lokaci (stát). Později teprve proběhne přepis z „address“ do kódu Wget.

5.4.3 Filtrování souřadnic z XML kódu pomocí getLocation

Jakmile je stažen XML soubor pomocí Wget, program si jej načte do bufferu jako řetězec a pomocí funkce getLocation hledá klíčové slovo <location>, jak již bylo popsáno výše u struktury XML souboru. Když je klíčové slovo nalezeno, hledají se přesnější informace, prvně číslo zapsané mezi <lat> a </lat>, které se uloží jako zeměpisná šířka, poté číslo mezi <lng> a </lng>, což je zeměpisná délka. K filtraci dat se využívá stejného algoritmu – funkce getIn – jako při filtraci z dat Whois.

Jakmile je zjištěna zeměpisná šířka i délka, jsou tyto informace vypsány do souboru nastaveného jako výstupní parametr, běžně jde o soubor out a zároveň jsou použita pro javascript, na zobrazení nalezených pozic na mapě.

```
1  if(address != NULL)
2      { //pokud byla adresa nalezena, hledáme podle adresy
3          strep (address, '\n', ','); // ještě nahradíme konce řádku
4              za čárky, jen pro jistotu
5          strep (address, ' ', '+');
6
7          getLocation(address, &lat, &lng, type, city, NULL); //
8              hledáme nejprve celou adresu (někdy tam je i název
9              firmy)
10         char *pch = address;
11
12         //pokud se nic nenajde (možná právě kvůli názvu firmy atd
13         ..)
14         //zkusíme postupně odmazávat části adresy, oddělovač pro
15         mazání je čárka
16         if(lat == INFINITY || lng == INFINITY)
17         { //infinity znamená, že se nic nenašlo
18             loop = FALSE;
19             while((pch = strchr(pch, ',')) != NULL && !loop)
20             {
21                 pch++;
22                 loop = getLocation(pch, &lat, &lng, type, city,
23                 NULL);
24             }
25         }
26     }
27     // pokud se nic nenašlo podle adresy tak alespoň podle města
28     if(lat == INFINITY || lng == INFINITY)
29     {
30         strep (city, ' ', '+');
31         if(strlen(city) != 0)
32             getLocation(city, &lat, &lng, type, city, country);
33     }
34     // a pokud ani podle města, tak alespoň země
35     if(lat == INFINITY || lng == INFINITY)
36     {
37         strep (country, ' ', '+');
38         if(strlen(country) != 0)
39             getLocation(country, &lat, &lng, type, city, country);
40     }
```

Výpis kódu 5.8: Filtrování odpovědi z google API pomocí getLocation.

5.5 Zobrazení výstupních informací na mapě

Google maps nabízí i možnosti vlastní úpravy a návrhu zobrazených informací, a to prostřednictvím javascriptu. Program má již definovanou většinu javascript kódu, který pak doplňuje o zjištěné údaje, jako je název města, ulice, GPS souřadnice, apod. V hlavičce kódu je pak adresa google maps, která se otevře v prohlížeči firefox¹ a následně je modifikována doplněným kódem javascriptu. Vkládání doplňkových informací do kódu probíhá v hlavní funkci main, po nalezení adresy a souřadnic z Whois a XML. Kód javascriptu je rozdělen na 2 části a následně načítán při běhu main. PART1 je propojená s daty z cmpFile a výstupu google maps API, slouží k výpočtu vzdálenosti mezi dvěma body a deklarací potřebných výrazů. PART2 definuje převážně vzhledové vlastnosti zobrazené mapy, značek (marker), zvýrazněné dráhy, popisků. Veškerá dokumentace k syntaxi je popsána na vývojářských stránkách google maps API [8]. V souboru cmp jsou správné souřadnice hledaných IP adres, tyto data jsou postupně přiřazena zjištěným souřadnicím a zobrazena na mapě. K otevření souboru cmp dochází již na začátku hlavní funkce main a data z něj jsou uložena a propojena s PART1, kde jsou následně použita (po zjištění dat z Whois a XML). V kódu je také vložen Haversin vzorec pro výpočet odchylky mezi kontrolním a zjištěným bodem. Výpočet tedy probíhá přímo v javascriptu a výsledek je zobrazen v popisku tratě, vyznačené červenou čarou mezi příslušnými body. Zápis Haversine vzorce vypadá následovně:

```
1  distHaversine = function(p1, p2) { var R = 6371; var dLat = rad(p2.lat  
    () - p1.lat()); \  
2      var dLong = rad(p2.lng() - p1.lng()); var a = Math.sin(  
        dLat/2) * Math.sin(dLat/2) + Math.cos(rad(p1.lat()  
    )) \  
3      * Math.cos(rad(p2.lat())) * Math.sin(dLong/2) * Math.  
        sin(dLong/2); var c = 2 * Math.atan2(Math.sqrt(a),  
        Math.sqrt(1-a)); \  
4      var d = R * c; return d.toFixed(3); }
```

Výpis kódu 5.9: Haversine zapsaný v javascript.

5.6 Funkce programu

Aplikaci je možné spouštět buď přímo z konzole 5.2, nebo pomocí spouštěcího souboru Whois.

¹je nastaven jako výchozí pro program, počítá se s prací v Linux

Pokud aplikaci spouštíme konzolí, je nutné se prvně přesunout do adresáře se spouštěcím souborem (všeobecně je to: `/bin/Debug`) a pak příkazem `./Whois [vstupní parametr] [výstupní parametr]` určit názvy vstupního souboru se zadanými IP adresami a název výstupního souboru. Program následně vytvoří výstupní soubor (*out*) ve stejném adresáři (`/bin/Debug`). Využijeme-li spouštěcího souboru, není nutné parametry zadávat, protože aplikace má tyto parametry implicitně nastaveny na `in` a `out`.

```
~$ cd /home/benedikt/Whois/bin/Debug
~/Whois/bin/Debug$ ./Whois in out
~/Whois/bin/Debug$
```

Obr. 5.2: Konzolové spouštění aplikace a zadání vstupně-výstupních parametrů.

Vstupním parametrem je primárně soubor `in`, v kterém jsou zapsány hledané IP adresy, vždy jedna na samostatný řádek. Druhým parametrem je porovnávací soubor `cmp`, který obsahuje správné souřadnice IP adres, kde „.“ Slouží jako desetinná čárka a „.“ uděluje zeměpisnou šířku od délky. Záměna významu oddělovacích znaků (v češtině) vznikla kvůli zdrojovému kódu XML, který využívá anglické značení. Vzhledem k jednoduchosti souboru je zřejmé, že musí být sepisován synchronizovaně se vstupním souborem `in`, aby nedošlo k záměně souvisejících souřadnic. Čtení z `cmp` zařizuje v hlavní funkci obslužný program `readFile`. Poté jsou souřadnice provázány s částí kódu javascript `PART1` a společně se zjištěnými souřadnicemi slouží k zobrazení na mapě a výpočtu vzdálenostní odchylky. Soubor `cmp` by měl být umístěn stejně jako soubory `in` a `out` do adresáře se spouštěcím souborem (všeobecně je to: `/bin/Debug`).

Výstupním parametrem je soubor `out`, v kterém jsou postupně vypsány IP adresy a k nim nalezená lokace. Dále jsou v souboru zapsány nalezené souřadnice ve formátu: `souradnice: s.:49.120343: d.: 10.779824` . Pro oddělení zeměpisné šířky a délky slouží „.“ z důvodu snadnějšího importu do aplikace Excel. Po nastavení rozpoznávání sloupců podle znaku „.“ lze jednoduše oddělit číselné informace od přebytečného textu, a ten následně odstranit.

Pro ukázkou vstupního a výstupního souboru zde uvádím záznam obrazovky 5.3:

```
in x | out x
50.22.100.250
83.208.112.79
77.75.72.43

50.22.100.250
adresa: 4849 Alpha Rd.
mesto: Dallas
zeme: US

83.208.112.79
adresa: Telefonica O2 Czech Republic, a.s.
adresa: Za Brumlovkou 2
adresa: Prague 4 - 140 22
adresa: The Czech Republic

77.75.72.43
adresa: Seznam.cz, a.s. ing. Michal Feix Radlicka 608/2 15000 Prague 5
CZECH REPUBLIC
adresa: Radlicka 608/2
150 00 Prague 5
Czech Republic
```

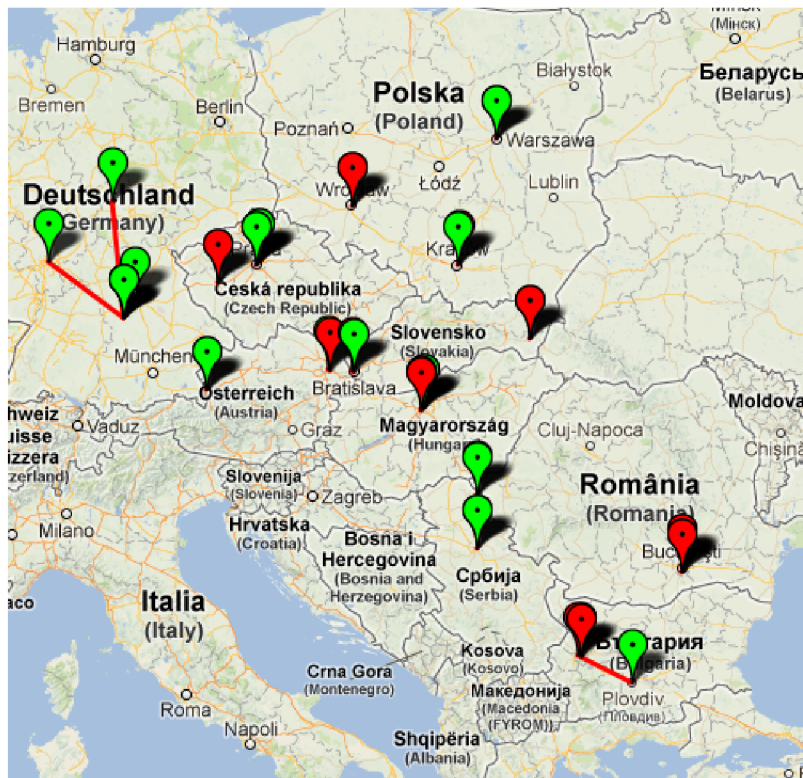
Obr. 5.3: Příklad zobrazení vstupního a výstupního textového souboru.

Po spuštění programu jsou postupně analyzovány IP adresy. Jakmile program dokončí funkci pro IP adresu, vypíše její pořadové číslo do konzole a načítá další. Úspěšné nalezení IP adres je zakončeno zprávou „done“, jak je zobrazeno na obrázku 5.4.

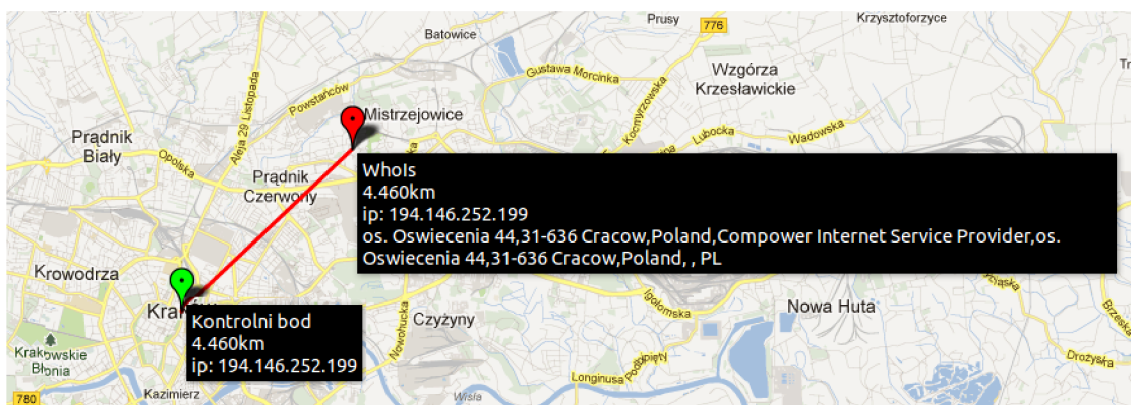
```
Whols
progress: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
done!
```

Obr. 5.4: Zobrazení průběhu zpracování IP adres.

Následně jsou všechny nalezené IP adresy zobrazeny pomocí internetového prohlížeče na mapě, a společně se správnou polohou zjištěnou ze souboru cmp jsou propojeny čarou, zobrazující odchylku mezi body (viz obr. 5.5). Každý zobrazený bod (marker) je barevně rozlišen, červenou barvou je zobrazen nalezený bod a zelenou barvou je označen kontrolní bod. Po „kliknutí“ na marker se zobrazí další informace, jako je vzdálenost v kilometrech, IP adresa, případně lokace. Vše je zobrazeno na obr. 5.6.



Obr. 5.5: Zobrazení nalezených IP adres na mapě.



Obr. 5.6: Detail informací nalezených bodů.

6 ANALÝZA NAMĚŘENÝCH DAT

Výstupní data programu jsem importoval ze souboru out do sešitu Excel a dále zpracoval. Pro importování souřadnic je zapotřebí použít jako identifikátor sloupců znak „:“ a dále vymazat přebytečný text. K tomu slouží přidané makro, které se spouští kombinací kláves `ctl + e`.

6.1 Výpočet odchylky mezi správným a nalezeným bodem.

Pro ověření přesnosti metod a správnosti výpočtu jsem použil jako kontrolní výpočet Haversinův algoritmus a pro hlavní výpočet jsem použil přesnější Vincenty formuli. Vzhledem k náročnosti Vincenty formule jsem použil již připravené makro, stažené z fóra www.path.cz/forum [21], autorem je Pavel Černák. Ze všech zjištěných hodnot jsem následně odstranil příliš velké chyby vzniklé během vyhledávání z databáze Whois, čímž vzrostla relevantnost výsledku.

6.2 Zhodnocení výsledků měření

Měření přesnosti metody Whois proběhlo pro 48 IP adres, z nichž jedna nebyla nalezena v databázi Whois (152.66.115.224). Vzhledem k některým vysokým chybám jsem stanovil hranici maximální odchylky na 800 km a všechny přesahující vzdálenosti považuji za nelokalizovatelné. Vysoké odchylky vznikly pravděpodobně chybou informací v databázi Whois, kdy sídlo zaregistrované firmy má zcela odlišnou adresu od provozovaného serveru, případně jde o zastaralý záznam. Po odstranění příliš velkých chyb a dalších dezinformací zbylo k porovnání 39 IP adres, kde byl proveden výpočet vzdálenosti mezi správnou a zjištěnou adresou (převedenou na GPS souřadnice). Pro ověření správného výsledku jsem použil dvě metody výpočtu – Haversine a Vincenty, přičemž Vincenty je použit jako směrodatný a Haversine k ověření.

V tabulce 6.1 jsou uvedeny jednotlivé IP adresy a k nim náležící vzdálenostní rozdíly správné a zjištěné polohy. Kompletní data měření jsou přiložena v elektronické formě na CD. Příloha obsahuje vzorce výpočtu, konkrétní souřadnice i vyřazené IP adresy.

Tab. 6.1: Tabulka vzdáleností mezi správnou a nalezenou pozicí.

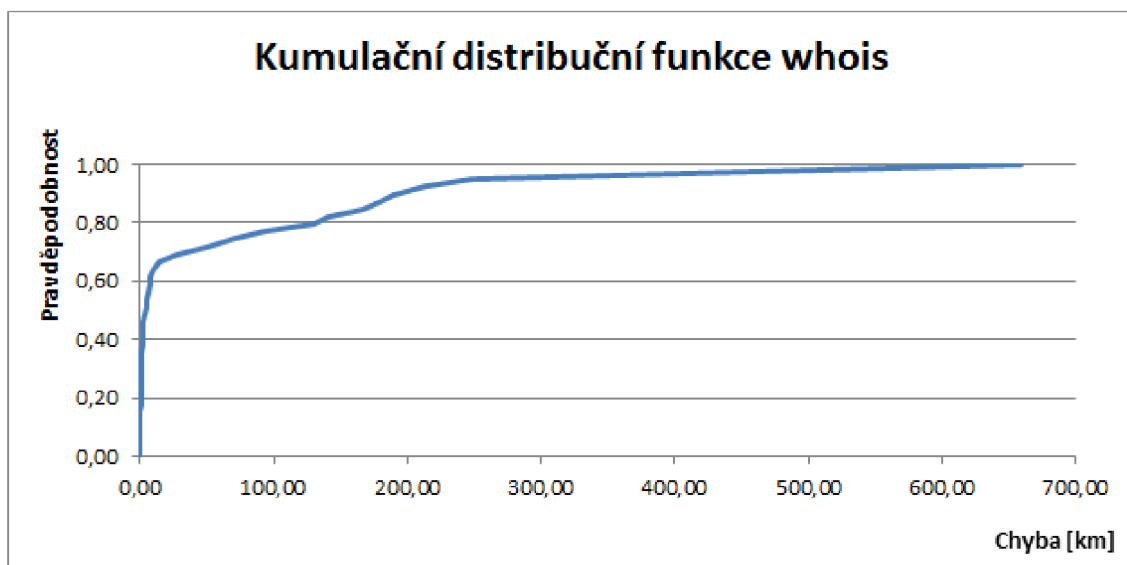
IP	Haversine [km]	Vincenty [km]
176.9.55.42	228,89	228,71
91.82.84.185	9,13	9,14
77.251.170.54	66,20	66,20
83.222.115.86	0,37	0,37
80.72.40.110	0,00	0,00
89.215.114.195	128,67	128,77
109.70.149.49	2,64	2,64
85.11.157.54	6,13	6,14
78.46.90.47	1,97	1,97
88.198.19.202	42,76	42,75
212.67.73.150	1,25	1,25
5.9.59.165	187,21	187,39
213.153.32.170	0,48	0,48
217.79.128.22	10,91	10,92
194.146.252.199	4,46	4,47
213.94.75.9	7,28	7,28
94.136.136.2	0,72	0,72
89.216.2.122	0,39	0,39
91.90.160.3	0,63	0,63
95.77.94.89	0,78	0,78
213.249.64.165	0,00	0,00
62.65.173.6	4,51	4,52
77.75.76.3	0,00	0,00
212.200.163.178	2,00	2,00
85.236.100.91	7,40	7,41
188.165.34.68	213,18	213,04
88.190.22.159	21,65	21,64
93.115.207.238	0,00	0,00
81.201.56.141	0,00	0,00
193.166.3.2	0,00	0,00
132.252.181.87	176,52	176,74
212.87.14.41	1,29	1,29
130.235.209.220	657,63	658,59
130.59.10.36	614,91	614,70

Z naměřených a vypočtených dat lze dále odvodit průměrnou vzdálenost mezi nalezenými body, která vychází na 71,5 km, přičemž medián je 4,5 km. Takto velký rozdíl mezi mediánem a průměrnou hodnotou je způsoben značným rozsahem přesností nalezených adres, kde nejnižší hodnota je 0 km pro 5 IP adres, zato největší rozdíl je 658,59 km. Dalším faktorem je pomalý postupný růst vzdáleností až po hodnotu 228 km a následně velký skok k 614,91 km. Proto je vhodnější používat průměr jako směrodatnou hodnotu.

Tab. 6.2: Přesnost Whois.

Průměr [km]	Medián [km]
71,476	4,465

Pro lepší přehled jsem graficky sestrojil kumulační distribuční funkci pro vypočtené vzdálenosti, která zobrazuje pravděpodobnost, že chyba metody bude menší, než daná vzdálenost.



Metoda Whois dokáže spolehlivě určit polohu na úrovni krajů, v mnoha případech také na úrovni měst a to až v přesnosti na řády kilometrů. Není plně spolehlivou geolokační metodou.

6.3 Porovnání přesností metod geolokace

Naměřená data z Whois jsem porovnal s výsledky dalších metod geolokace, které testovali mí kolegové. Prvním způsobem je využití databáze MaxMind, konkrétně neplacené verze GeoLite, kterou analyzoval Josef Pokorný. Další metodou geolokace je využitím domén od Ondřeje Jelínka. Poslední zkoumanou metodou je kombinace technik Constraint Based Geolocation a Speed of Internet od Michaela Horáka. Protože je ve všech metodách použit stejný seznam IP adres, je srovnání těchto geolokačních technik vhodným ukazatelem přesnosti. Porovnávaná data metod MaxMind a Databáze byla již zbavena chyby překračující vzdálenost 300 km, proto jsem i ostatní metody zbavil odchylek nad 300 km, čímž vzniká nižší průměrná chyba metody, než jaká je uváděna u samostatného rozboru výsledku Whois (800 km).

Tab. 6.3: Porovnání průměrné chyby metod.

Whois (Henek)	CBG a SOI (Horák)	Domény (Jelínek)	MaxMind (Pokorný)
40,96 km	99,01 km	26,84 km	49,04 km

Z pasivních metod geolokace je nejvíce přesné vyhledávání podle domén, které sice v principu neumožňuje větší přesnost¹, zde ale došlo k přesným výsledkům díky konkrétně stavěné databázi na názvy (domény) firem, jde tedy o přesnou databázi, ale pouze pro malý zeměpisný úsek, či úzký okruh firem. Metody využívající globálních neplacených databází² dosahují blízkých výsledků (rozdíl pouze 9 km), jde tedy o metody určující lokaci na úrovni krajů. Hybridní metoda CBG a SOI je jediným zástupcem aktivní geolokace a patří mezi nejméně přesné. Měření aktivních metod ovlivňuje rozmístění aktivních měřících prvků sítě (sond), což je prvek těžko manipulovatelný. Jak již bylo výše zmíněno, využívají metody stejných IP adres, proto zde uvádím srovnání přesnosti u 10 IP adres.

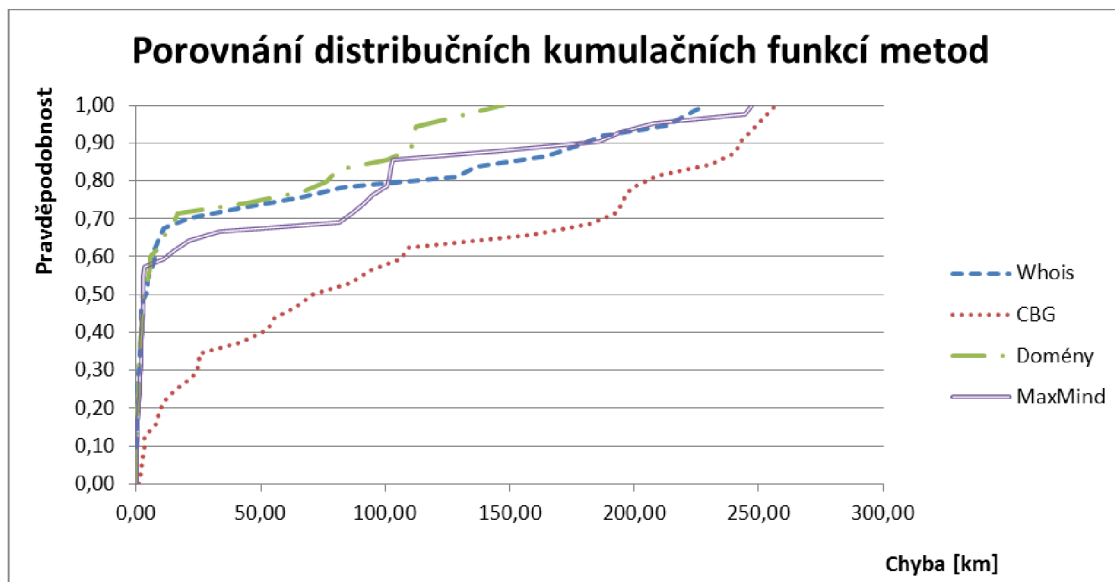
¹V případě nalezení pouze domény státu jsou vypsány souřadnice geografického středu tohoto státu, což je nejčastější případ. Domény měst a krajů nejsou tolik rozšířené.

²U MaxMind byla měřena neplacená verze – GeoLite. Placená databáze dosahuje podle společnosti MaxMind přesnějších výsledků.

Tab. 6.4: Porovnání přesnosti metod u konkrétních IP adres.

IP Adresa	CBG Chyba [km]	Whois Chyba [km]	Databáze Chyba [km]	Max Mind Chyba [km]
91.82.84.185	3,118	9,144	11,440	91,281
80.72.40.110	0,917	0,000	112,400	2,412
109.70.149.49	247,812	2,641	14,530	146,852
78.46.90.47	42,091	1,975	44,210	244,447
212.67.73.150	7,945	1,251	78,530	81,800
5.9.59.165	182,670	187,387	147,610	101,493
213.153.32.170	240,210	0,481	106,940	33,672
217.79.128.22	3,118	10,917	11,140	3,205
194.146.252.199	230,861	4,468	1,170	2,898
213.94.75.9	11,275	7,283	162,500	247,433

Jak je vidět v tabulce 6.4, jsou přesnosti metod pro konkrétní IP adresu velmi rozdílné, od několika kilometrů až do řádu stovek kilometrů, z tohoto důvodu nelze porovnávat metody přímo v souvislosti s vybranou IP adresou. Konečné porovnání metod pomocí distribuční kumulační funkce je vyneseno na grafu.



7 ZÁVĚR

Ve své práci jsem shrnul poznatky týkající se určení fyzické polohy zjišťovaného cíle. Téma práce je zaměřeno na pasivní metody, využívající IP adresu jako nosnou informaci pro svoji činnost. Ze dvou nejčastějších databází – Whois a GeoIP, jsem v souladu se zadáním využil databázi Whois. Z textu vyplývá, že využívání pasivních metod při geolokaci přináší určitá rizika, s kterými je nutno počítat. Veřejné databáze, jako je Whois a GeoLite, jsou sice označovány za méně kvalitní služby, než komerční alternativy – GeoIP, zato jsou bezplatné a tak dostupné všem. Chceme-li využívat geolokace dlouhodobě, je nutné zvážit potřebu přesnosti nalezených záznamů a případné náklady s tím spojené. Máme-li na svých webových stránkách informační počítač, zobrazující statistiky o návštěvnících a chceme znát odkud se připojují, není nutné vyhledávat přímo města. Pokud se snažíme prosadit svoji společnost pomocí cílené reklamy, je vhodné využít databází s větší přesností, ideálně s určením města, či ulice (pro nabídku nejbližšího obchodu v Brně, apod.). Aktivní metody geolokace jsou vhodnější pro cíle, které mohou měnit polohu, nebo ji skrývat.

V další části práce je popsán algoritmus pro analýzu odpovědi z databáze Whois, pracující s nástrojem whois v Linuxu. Algoritmus byl vytvořen v programovacím jazyce C, pomocí prostředí Code::Blocks. Základním parametrem při spouštění programu je zadání vstupu a výstupu. Ze vstupního textového souboru (in) algoritmus načítá postupně IP adresy a odesílá dotaz na databázi Whois, přičemž každou odpověď analyzuje a filtruje pouze udanou fyzickou adresu, kterou dále převede na souřadnice. Získané informace jsou vypsány do výstupního textového souboru (out). Program také porovnává nalezené souřadnice se správnou polohou, která je zapsána v porovnávacím souboru (cmp), počítá vzdálenost mezi těmito body a následně všechny zjištěné polohy zobrazí na mapě. K testování byl využit seznam 48 IP adres, u nichž jsme zjistili jejich fyzickou polohu a GPS souřadnice.

Zjištěné a vypočtené informace jsou komentovány v kapitole Analýza naměřených dat. Prvně byla odstraněna chyba nad 800 km, která vznikla chybným záznamem v databázi Whois, nebo nesprávně zjištěnou skutečnou polohou porovnávané IP adresy. Filtrací nadměrné chyby jsme získali data pro 39 IP adres. Průměrná vzdálenostní chyba metody Whois je 71,48 km, ovšem při poměrně vysoké toleranci nepřesnosti (oněch 800 km). Statisticky je tedy metoda Whois schopná nalézt IP adresu na úrovni kraje. V porovnání s jinými metodami geolokace, především tedy s pasivními metodami (domény, GeoLite), je Whois méně přesnou metodou, avšak na rozdíl od geolokace pomocí domén má mnohem obsáhlejší databázi. Metoda GeoLite (neplacená verze GeoIP) je svojí průměrnou chybou 49,04 km, při toleranci maxi-

mální chyby 300 km, blízká přesností metodě Whois s průměrnou chybou 40,96 km. Chceme-li jistotu nalezení IP adresy na úrovni měst, je vhodné zvážit komerční databáze. Pro přibližné určení pozice na úrovni krajů jsou zcela postačující neplacené databáze.

LITERATURA

- [1] Andrews Space & Technology. *GLONASS - Summary* [online]. ©2001. [cit. 12. 2. 2013]. Dostupné z URL: <http://www.spaceandtech.com/spacedata/constellations/glonass_consum.shtml>.
- [2] CAJTHAML, Martin Geolokace aneb ideální internet zase o krok blíž *www.symbio.cz* [online]. 2009, poslední aktualizace 18. 2. 2009 [cit. 5. 12. 2012]. Dostupné z URL: <<http://www.symbio.cz/clanky/geolokace-aneb-idealni-internet-zase-o-krok-bliz.html>>.
- [3] ČADA, Václav. *Geodézie* [online]. 27. 4. 2005. [cit. 25. 2. 2013]. Dostupné z URL: <<http://gis.zcu.cz/studium/gen1/html-old/ch02.html>>.
- [4] DVOŘÁK, F. *Využití znalosti topologie páteřních sítí pro určování fyzické polohy stanic v síti Internet*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 77 s. Vedoucí diplomové práce doc. Ing. Dan Komosný, Ph.D. [cit. 5. 12. 2012]. Dostupné z URL: <<http://www.elektrorevue.cz/cz/clanky/komunikacni-technologie/25/geolokace-sitovych-zarizeni-v-internetovych-sitich/>>. ISSN 1213-1539
- [5] European Space Agency. *European Satellite Navigation* [online]. ©2010. [cit. 12. 2. 2013]. Dostupné z URL: <<http://www.satellite-navigation.eu/>>.
- [6] GERICH, E. *Guidelines for Management of IP Address Space* [online]. 1993, [cit. 5. 12. 2012], Dostupné z URL: <<http://tools.ietf.org/html/rfc1466>>.
- [7] Google. *The Google Geocoding API* [online]. poslední revize 13. 2. 2013. [cit. 18. 3. 2013]. Dostupné z URL: <<https://developers.google.com/maps/documentation/geocoding/>>.
- [8] Google. *Google Maps JavaScript API v3* [online]. Poslední revize 14. 3. 2013. [cit. 29. 3. 2013]. Dostupné z URL: <<https://developers.google.com/maps/documentation/javascript/tutorial>>.
- [9] GPS Slansko. *GPS Slansko* [online]. 26. 7. 2008. [cit. 2. 2. 2013]. Dostupné z URL: <<http://gps.slansko.cz/>>.

- [10] HAAS, J. Linux / Unix Command: nslookup. *www.about.com* [online]. 10.4.2012, [cit. 5. 12. 2012]. Dostupné z URL: <http://linux.about.com/od/commands/1/blcmd18_nslooku.htm>.
- [11] IANA IPv4 Address Space Registry. *www.iana.org* [online]. 2012, poslední aktualizace 16. 11. 2012, [cit. 5. 12. 2012], Dostupné z URL: <<http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml>>.
- [12] Internet registry. *www.zam.opf.slu.cz* [online]. 1999, poslední aktualizace 8.6.1999 [cit. 5. 12. 2012], Dostupné z URL: <<http://zam.opf.slu.cz/botlik/CD-1x/17.html>>.
- [13] KOMOSNÝ, D. a VERNER, L. Geolokace síťových zařízení v internetových sítích. *Elektrorevue* [online]. 2011, [cit. 5. 12. 2012]. Dostupné z URL: <<http://www.elektrorevue.cz/cz/clanky/komunikacni-technologie/25/geolokace-sitovych-zarizeni-v-internetovych-sitich/>>. ISSN 1213-1539
- [14] KVAPIL, Jiří. *Kosmický segment GPS a jeho budoucnost* [online], 10. 1. 2005, roč.3, č.2. [cit. 5. 2. 2013]. Dostupné z URL: <http://www.aldebaran.cz/bulletin/2005_02_gps.php>.
- [15] Math Forum. *Deriving the Haversine Formula* [online]. ©2013. [cit. 5. 4. 2013]. Dostupné z URL: <<http://mathforum.org/library/drmath/view/51879.html>>.
- [16] MaxMind, Inc. *GeoIP databases and web services* [online]. [cit. 5. 12. 2012]. Dostupné z URL: <http://www.maxmind.com/en/geolocation_landing>.
- [17] Microsoft Corporation. *Microsoft Excel help – ARCTG2* [online]. ©2013. [cit. 5. 4. 2013]. Dostupné z URL: <<http://office.microsoft.com/cs-cz/excel-help/arctg2-HP005208991.aspx>>.
- [18] NIKSIC, Hrvoje. *ubuntu manuals – wget* [online]. 13. 1. 2008. [cit. 29. 3. 2013]. Dostupné z URL: <<http://manpages.ubuntu.com/manpages/hardy/cs/man1/wget.1.html>>.
- [19] PAREKH, SANJAY M., Friedman, ROBERT B., TIBREWALA, NEAL K., LUTCH, BENJAMIN *Systems and methods for determining collecting and using geographic locations of internet users* [online]. 29.6.2004. [cit. 5. 12. 2012]. Dostupné z URL: <<http://www.freepatentsonline.com/6757740.html>>.
- [20] Path. *Geodetický systém WGS 84*. [online]. 9. 7. 2007. [cit. 2. 3. 2013]. Dostupné z URL: <<http://www.path.cz/forum/viewtopic.php?f=4&t=19#p26>>.

- [21] Path. *Jak Garmin počítá přímou vzdálenost mezi body* [online]. 10.3.2009. [cit. 2.3.2013]. Dostupné z URL: <<http://www.path.cz/forum/viewtopic.php?f=20&t=602>>.
- [22] RIPE NCC *RIPE NCC Annual Report* [online]. 2006, str.6 [cit. 5.12.2012], Dostupné z URL: <http://www.ripe.net/lir-services/ncc/gm/may-2007/supporting-documents/annual_report_2006.pdf>.
- [23] ŠÍMA, Petr. *Navigační systém GPS* [online]. 27.3.2003. [cit. 5.2.2013]. Dostupné z URL:<http://krovak.webpark.cz/triangulace/triangulace_gps.htm>.
- [24] The Open Group Base Specifications Issue 6 *The IEEE and The Open Group* [online]. [cit. 5.12.2012], Dostupné z URL: <<http://pubs.opengroup.org/onlinepubs/009695399/functions/popen.html>>.
- [25] UKO, Tomáš. *GPS navigace na FPGA*. Praha, 2008. Diplomová práce na Elektrotechnické fakultě Českého vysokého učení technického v Praze. Vedoucí práce: Ing. Martin Daněk, Ph.D.
- [26] VENESS, Chris. *Vincenty formula for distance between two Latitude/Longitude points* [online]. ©2012. [cit. 18.3.2013]. Dostupné z URL: <<http://www.movable-type.co.uk/scripts/latlong-vincenty.html>>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

- AfriNIC Africké síťové informační centrum – African Network Information Center
- API rozhraní pro programování aplikací – Application Programming Interface
- APNIC Asijsko-Pacifické síťové informační centrum – Asia-Pacific Network Information Centre
- CBG aktivní metoda geolokace využívající trilaterace – Constraint Based Geolocation
- DNS systém doménových jmen – Domain name system
- FTP síťový protokol pro přenos souborů mezi počítači – File Transfer Protocol
- GPS globální poziční systém – Global Position System
- GNSS Globální družicový polohový systém – Global Navigation Satellite System
- GNU nekompletní počítačový svobodný operační systém (GNU/Linux) – GNU's Not Unix
- HTML značkovací jazyk hypertextu pro vytváření stránek v systému World Wide Web – HyperText Markup Language
- HTTP internetový protokol určený pro výměnu hypertextových dokumentů ve formátu HTML – Hypertext Transfer Protocol
- HTTPS zabezpečený internetový protokol určený pro výměnu hypertextových dokumentů ve formátu HTML – Hypertext Transfer Protocol Secured
- IANA mezinárodní agentura pro přidělování čísel na Internetu – Internet Assigned Numbers Authority
- ICANN mezinárodní nezisková organizace s širší působností na internetu – Internet Corporation for Assigned Names and Numbers
- LACNIC Latinsko Americké a Karibické Internetové adresy registrů – Latin America and Caribbean Network Information Centre
- IP internetový protokol — Internet Protocol
- JSON javascriptový objektový zápis, nezávislý na počítačové platformě, určený pro přenos dat – JavaScript Object Notation

LBS služby stojící na určení polohy uživatele – Location-based services

NATO euroatlantický mezinárodní vojenský pakt – North Atlantic Treaty Organization

NAVSAT historicky první družicový polohový systém – Navy Navigation Satellite System

RG regulární výraz

RIPE NCC regionální registr pro Evropu a Blízký východ – Réseaux IP Européens Network Coordination Centre

RIR regionální Internetový registrátor – Regional Internet registry

RTT obousměrné zpoždění –Round-Trip Time

Wi-Fi aktivní geolokační metoda měřící pomocí zpoždění – Speed of Internet

UTM způsob určování polohy na povrchu Země založený na síti šedesáti zón – Universal Transverse Mercator

VoIP technologie pro přenos digitalizovaného hlasu prostřednictvím počítačové sítě nebo Internetu – Voice over Internet Protocol

WGS 84 geodetický standard vydaný ministerstvem obrany USA roku 1984 – World Geodetic System 1984

Wi-Fi komunikační standard pro bezdrátový přenos dat – Wireless Fidelity

XML obecný značkovací jazyk určený pro výměnu dat mezi aplikacemi a pro publikování dokumentů, nezabývajících se vzhledem – Extensible Markup Language

SEZNAM PŘÍLOH

A Obsah přiloženého CD

57

A OBSAH PŘILOŽENÉHO CD

- elektronická verze práce
- spouštěcí soubor aplikace a příslušné adresáře