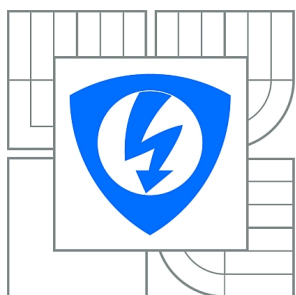


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

ROZBOR PARAMETRŮ KOMUNIKAČNÍHO ŘETĚZCE PRO IP GEOLOKACI

ANALYSIS OF THE COMMUNICATION PATH ATTRIBUTES FOR IP GEOLOCATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN RAJNOHA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JIŘÍ BALEJ

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Martin Rajnoha

ID: 136579

Ročník: 3

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Rozbor parametrů komunikačního řetězce pro IP geolokaci

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte současné možnosti hledání fyzické pozice stanic v síti Internet, zaměřte se především na IP geolokaci založenou na měření zpoždění. Dále nastudujte dílčí faktory způsobující zpoždění při přenosu paketů mezi stanicemi v IP sítích. Vytvořte aplikaci sloužící ke zjištění geografické polohy stanic na základě měření zpoždění mezi stanicemi. Přepočítání zpoždění na geografickou vzdálenost proveďte dle parametrů uvedených v [1]. Z vašich výsledků vyjádřete přesnost uvedeného způsobu geolokace.

DOPORUČENÁ LITERATURA:

- [1] LAKI, S., et al.: A Model Based Approach for Improving Router Geolocation, Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 54, no. 9, pp. 1490-1501, 2010. ISBN: 978-1-4244-2846-5
- [2] BOVY, C. J., et al.: Analysis of End-to-end Delay Measurement in Internet. PAM, 2002. s. 8. 13s.
- [3] GUEYE, B., et al.: Constraint-Based Geolocation of Internet Hosts, IMC, 2004. 6s. ISBN 1-58113-821-0

Termín zadání: 11.2.2013

Termín odevzdání: 5.6.2013

Vedoucí práce: Ing. Jiří Balej

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

Cieľom tejto práce bolo preštudovať súčasné možnosti lokalizácie staníc v sieti Internet, hlavne aktívne metódy, ktoré sú založené na meraní oneskorenia. Popísať vznik oneskorenia a jeho jednotlivých častí. Ďalej vytvoriť aplikáciu, ktorá je schopná vzdialene merať oneskorenie medzi stanicami a prepočítavať oneskorenie na vzdialenosť. Aplikácia pomocou týchto vzdialeností zistí geografickú polohu zadanej stanice. Pri práci bola využitá experimentálna sieť PlanetLab.

KĽÚČOVÉ SLOVÁ

ip, lokalizácia, oneskorenie, planetlab, geolokácia, poloha, súradnice

ABSTRACT

The aim of this thesis was to study current resources to find location of stations in the network Internet, mainly active methods that are based on delay measurements. Describe origin of the delay and its parts. Next create an application that is able remotely measure the delay between stations and convert this delay to distance. Application calculate geographic position of station on based this distances. For measurement was used experimental network PlanetLab.

KEYWORDS

ip, localization, delay, latency, planetlab, geolocation, position, coordinates

PREHLÁSENIE

Prehlasujem, že som svoju bakalársku prácu na tému „Rozbor parametrov komunikačného reťazca pro IP geolokaci“ vypracoval samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúceho autorského zákona č. 121/2000 Sb., o právu autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

Brno

.....

(podpis autora)

POĎAKOVANIE

Rád by som poďakoval vedúcemu Bakalárskej práce pánovi Ing. Jiřímu Balejovi za konzultácie, rady, odbornú pomoc a návrhy k práci, ktoré som využil pri tvorbe tejto práce.

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>

POĎAKOVANIE

Výzkum popsaný v této bakalářské 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



OBSAH

Úvod	11
1 Možnosti IP lokalizácie na internete	12
1.1 Pasívne metódy	12
1.2 Aktívne metódy	13
1.2.1 ShortestPing	13
1.2.2 GeoPing	13
1.2.3 CBG	13
1.2.4 TBG	14
1.2.5 Octant	15
1.2.6 GeoWeight	16
2 Oneskorenie	17
2.1 Oneskorenie v medzilahlých uzloch	17
2.1.1 Oneskorenie dobou spracovania inforámca	17
2.1.2 Oneskorenie vo fronte	18
2.1.3 Iné oneskorenia	18
2.1.4 Serializačné oneskorenie	19
2.2 Oneskorenie spôsobené rýchlosťou šírenia signálu	19
2.3 Závislosť oneskorenia na počte uzlov	19
2.4 Vytvorenie ICMP Echo Reply	20
3 Nástroj ping	22
4 Lokalizácia pomocou oneskorenia	23
4.1 Vzťah medzi oneskorením a vzdialenosťou	23
4.1.1 Hodnota parametru r	23
4.2 Nájdenie súradníc hľadaného bodu	25
4.2.1 Určenie súradníc priesečníkov	25
4.2.2 Výber priesečníkov a nájdenie cieľa	28
5 Riešenie	30
5.1 PlanetLab	30
5.2 Aplikácia	31
5.2.1 Výpočet vzdialeností	32
5.2.2 Hľadanie priesečníkov	34
5.2.3 Získanie súradníc cieľa	35
5.2.4 Cyklus na kombinácie serverov a výpočet súradníc	36

6	Výsledky merania	38
7	Záver	44
	Literatúra	45
	Zoznam symbolov, veličín a skratiek	48
	Zoznam príloh	50
A	Obsah CD	51

ZOZNAM OBRÁZKOV

1.1	Použitie metódy trilaterácie pri CBG.	14
1.2	Metóda Octant.	15
1.3	Pridelenie váh k regiónom pri metóde GeoWeight.	16
2.1	Výpis nástroja ping pre zapojenie s jedným uzlom.	21
2.2	Histogram času potrebného na vygenerovanie Echo Reply.	21
4.1	Hodnoty parametru r prevzaté z [9].	24
4.2	Hodnoty parametru r pri vlastnom meraní vzdialeností.	24
4.3	Priesečníky troch kružníc.	25
4.4	Zmenšovanie vzdialeností a priblíženie ťažiska k cieľu.	27
4.5	Princíp zmenšovania a zväčšovania kružníc pri hľadaní priesečníkov.	28
4.6	Hľadanie priesečníkov dvoch kružníc.	29
5.1	Vzhľad aplikácie vykonávajúcej lokalizáciu.	31
5.2	Chyba, ktorá môže nastať pri určovaní správneho priesečníka.	36
5.3	Podmienka zaisťujúca, aby hľadaný bod ležal vo vnútri trojuholníka.	37
6.1	Porovnanie odchýliek merania pri hodnote $r = 0,0349$	40
6.2	Porovnanie odchýliek merania pri hodnote $r = 0,1888$	40
6.3	Porovnanie odchýliek merania pri hodnote $r = 0,3354$	41
6.4	Odchýlka merania pre rôzne hodnoty r , pri použití metódy priemeru.	41
6.5	Odchýlka merania pre rôzne hodnoty r , pri použití metódy mediánu.	42
6.6	Odchýlka merania pre rôzne nastavené parametre.	43

ZOZNAM TABULIEK

1.1	Tabulka pre určenie váhy k oblastiam u metódy GeoWeight.	16
6.1	Porovnanie chýb pri určovaní polohy.	38
6.2	Počet výsledkov v intervaloch vzdialeností pre priemer.	39
6.3	Počet výsledkov v intervaloch vzdialeností pre medián.	39
6.4	Porovnanie chýb pri meraní s inou hranicou tolerancie a počtom kombinácie serverov.	43

ÚVOD

IP (Internet Protocol) lokalizácia je pomerne nový pojem v oblasti počítačových sietí. Čo si ale pod týmto pojmom predstaviť?

Každé aktívne zariadenie v sieti Internet musí mať pre svoju činnosť jedinečnú adresu – IP adresu, či už je to DNS (Domain Name Server) server, webový server, smerovač a iné. Výnimku tvoria privátne IP adresy, ktoré sa prekladajú na hraničnom smerovači. Nájdenie fyzickej polohy týchto zariadení sa nazýva IP lokalizácia, alebo geolokácia. V súčasnosti existujú dve základné metódy – pasívne a aktívne.

Pasívne fungujú na princípe hľadania záznamov o IP adrese v databázach. Tieto databázy môžu byť plnené napr. DNS záznamami alebo registráciou u dodávateľa internetu.

Aktívne metódy fungujú na princípe merania oneskorenia, prípadne ešte iných sieťových parametrov. Oneskorenie je čas potrebný na doručenie správy zo zdroja do cieľa. Doručenie správy v sieti Internet tvorí množstvo zložitých procesov, ktoré majú vplyv na oneskorenie. Pre IP lokalizáciu je najvýznamnejším parametrom oneskorenie spôsobené rýchlosťou šírenia signálu po prenosovom médiu. Ďalším významným parametrom je oneskorenie spôsobené spracovaním paketu v medziľahlých uzloch. Počet medziľahlých uzlov je možné zistiť z hodnoty TTL (Time To Live) výpisu nástroja ping, ktorý sa používa v tejto práci. Toto oneskorenie sa prepočíta na geografickú vzdialenosť a následne sa určí poloha cieľa.

V tejto práci bola využitá experimentálna sieť PlanetLab, ktorú tvorí množstvo serverov po celom svete. Na tieto servery je možné sa vzdialene prihlásiť pomocou SSH (Secure SHel) a spúšťať základné príkazy.

Na určenie geografickej polohy stanice bola vytvorená aplikácia, ktorá sa vzdialene prihlási na určité servery PlanetLabu a pomocou príkazu ping zistí oneskorenie a počet medziľahlých uzlov k hľadanej stanici. Z týchto parametrov sa vypočíta vzdialenosť a následne pomocou priesečníkov kružníc sa určí poloha hľadanej stanice.

1 MOŽNOSTI IP LOKALIZÁCIE NA INTER- NETE

IP lokalizácia je pomerne nová služba na internete, ktorá určuje geografickú polohu zo zadanej IP adresy. Podľa princípu, akým pracujú lokalizačné metódy, ich je možné rozdeliť na pasívne a aktívne.

1.1 Pasívne metódy

Pasívne metódy fungujú na základe záznamov v databáze, ktoré obsahujú organizačné údaje o IP doméne, alebo skúšajú odvodiť polohu na základe DNS.

Mnohé z týchto metód používajú WHOIS databázu, z ktorej je možné získať meno a adresu organizácie, ktorá je zaregistrovaná na danú IP adresu. V prípade veľkého poskytovateľa internetu ISP (Internet Service Provider) alebo veľkej organizácie, ktorá nesídlí na jednom mieste, sa často líši registrovaná adresa od skutočného umiestnenia.

Ďalšou nevýhodou je veľká chybovosť, ktorá môže nastať pri nesprávne zaradených záznamoch. Keďže sa záznamy do databáz vkladajú manuálne a sú aj udržiavané manuálne, môže sa stať, že niektoré záznamy sú zastaralé, alebo úplne neplatné.

Tieto aplikácie nám vo väčšine prípadov určia polohu poskytovateľa internetu, resp. jeho registrovanú adresu, pričom reálna poloha môže byť úplne niekde inde. Niektoré aplikácie vyhľadajú dokonca len štát, v ktorom sa daná IP adresa nachádza.

Výhodou týchto metód je rýchlosť, akou dokážu lokalizovať IP adresu. Vo všeobecnosti pasívne metódy dokážu veľmi presne lokalizovať hľadanú stanicu, pokiaľ sú presné záznamy v databázach.

V [15] boli vykonané mnohé merania na porovnanie presností niektorých pasívnych metód. IPLigence, Maxmind a IP2Location majú 62 % až 73 % pravdepodobnosť správneho určenia polohy v okruhu 40 km. IPLigence a Maxmind majú dokonca 80 % pravdepodobnosť správneho určenia polohy v okruhu 500 km. Geobytes, HostIP.Info a Netacuity v rámci mesta dokážu presne určiť 33 % až 47 % IP adries. V okruhu 500 km je pravdepodobnosť správneho určenia od 48 % do 60 %.

Na internete nájdeme množstvo online aplikácií, ktoré lokalizujú na základe záznamov v databáze, napr. whois, IpLocator, dnsfrog. . .

1.2 Aktívne metódy

Aktívne metódy fungujú na základe merania oneskorenia, prípadne iných parametrov v sieti, ktoré sa prepočítajú na vzdialenosť. Na zmeranie oneskorenia sa používa nástroj `ping`, alebo `traceroute`, ktorý nám vypíše aj uzly, cez ktoré sa dostane správa k cieľu. Meranie sa realizuje z referenčných bodov, ktorých poloha je známa. Čím viac referenčných bodov je k dispozícii, tým je lokalizácia presnejšia.

1.2.1 ShortestPing

Metóda ShortestPing je veľmi jednoduchá metóda, založená na meraní oneskorenia. K činnosti používa množstvo referenčných bodov, z ktorých sa meria oneskorenie k cieľu. Poloha cieľa sa určí ako poloha referenčného bodu, ktorý má k cieľu najmenšiu hodnotu oneskorenia. Nevýhodou tejto metódy je pomerne veľká nepresnosť, pretože výsledná poloha je určená ako poloha najbližšieho referenčného bodu, pričom reálna poloha môže byť vzdialenejšia. [17].

1.2.2 GeoPing

Metóda GeoPing je podľa [13] metóda, ktorá získava geografickú polohu na základe vzťahu medzi oneskorením a vzdialenosťou. Meranie oneskorenia vykonávajú viaceré zdroje so známou polohou tzv. sondy. Ďalej potrebuje k svojej činnosti veľké množstvo referenčných bodov.

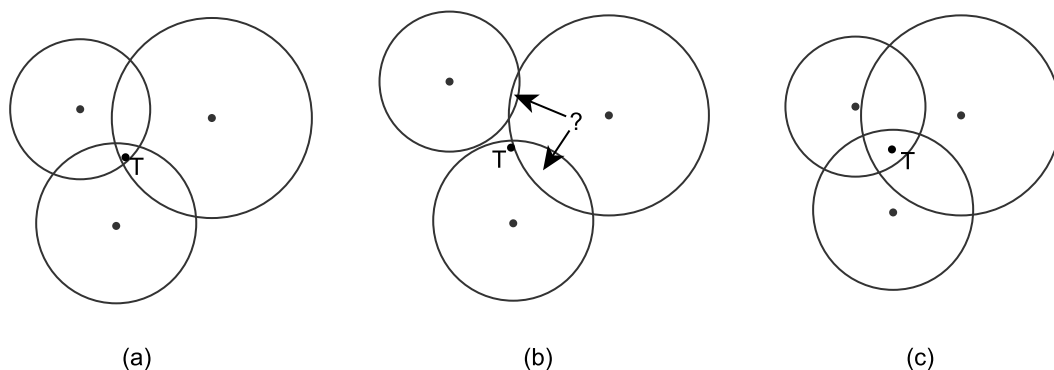
GeoPing využíva metódu Nearest Neighbor in Delay Space (NNDS). Je založená na myšlienke, že ak je oneskorenie zmerané medzi sondou a cieľom rovnaké ako medzi sondou a referenčným bodom, musí sa nachádzať v jeho blízkosti.

Prvým krokom teda musí byť vytvorenie „mapy oneskorenia“ (z ang. `delay map`), ktorej každý záznam obsahuje súradnice referenčného bodu a „vektor oneskorenia“ (z ang. `delay vector`) $DV = (d_1, \dots, d_N)$, obsahujúci zmerané oneskorenie z N sondy. Ďalej pre lokalizovanie cieľa T , sa musí zmerať oneskorenie zo sond a zkonštruovať vektor $DV' = (d'_1, \dots, d'_N)$.

Nakoniec sa vyhledá v mape oneskorenia vektor DV , ktorý sa najviac zhoduje s DV' . K tomu sa použije Euklidovská vzdialenosť $\sqrt{(d_1 - d'_1)^2 + \dots + (d_N - d'_N)^2}$. Nevýhodou tejto metódy je, že nelokalizujeme priamo cieľ, ale najbližší referenčný bod.

1.2.3 CBG

Metóda Constraint Based Geolocation (CBG) podľa [7] používa na získanie polohy cieľa metódu trilaterácie. Ak by CBG meralo oneskorenie k cieľu priamo a prepočí-



Obr. 1.1: Použitie metódy trilaterácie pri CBG, kde (a) ukazuje správny výpočet vzdialeností, (b) je tzv. mismatch – prienik neexistuje a u väčšiny metód lokalizácie sa nedá určiť a (c) ukazuje správny výpočet, lenže oblasť, kde môže ležať hľadaný bod je veľká.

tavalo na vzdialenosť, metóda by stratila presnosť. K tomuto účelu prechádza každý referenčný bod fázou kalibrácie. Meria vzdialenosť ku známym referenčným bodom a porovnáva s reálnou vzdialenosťou. To mu umožní prispôbiť sa premenným sieťovým podmienkam v reálnom čase. Až po tomto kroku prichádza na rad lokalizácia. Vykoná sa priame meranie oneskorenia k cieľu a prepočet na vzdialenosť.

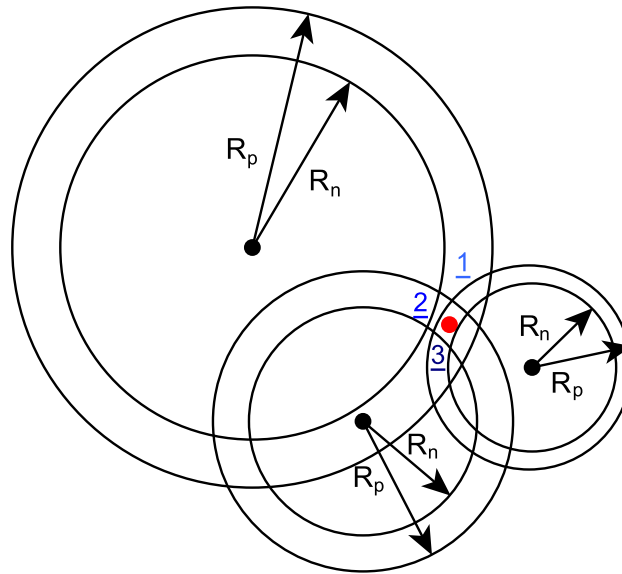
Vo všeobecnosti, každá vzdialenosť definuje kružnicu okolo bodu, z ktorého bolo meranie vykonané. Výsledná poloha hľadaného bodu je určená priesečníkom všetkých kružníc. V ideálnom prípade sa všetky kružnice pretnú v jednom bode – mieste, kde sa nachádza cieľ. V reálnej situácii však nezískame jeden bod, ale oblasť, v ktorej môže hľadaný bod ležať. Na obr. 1.1 môžeme vidieť rôzne scenáre metódy trilaterácie.

1.2.4 TBG

Metóda Topology Based Geolocation (TBG) používa k svojej práci meranie oneskorenia a informácie zo sieťovej topológie. Lokalizuje všetky smerovače medzi referenčnými bodmi a cieľom. Tým si dokáže vytvoriť akúsi mapu cesty k cieľu. Tým sa z veľkej časti priblíži reálnej vzdialenosti k cieľu. Metóda TBG používa trilateráciu k získaniu polohy cieľa, ale zároveň aj všetkých medzilahých smerovačov. Základný nástroj pre fungovanie TBG je `traceroute`. [8].

1.2.5 Octant

Podľa [6] sa u metódy Octant pri hľadaní oblasti využívajú tzv. pozitívne informácie – ohraničujú miesto, kde s najväčšou pravdepodobnosťou leží daný cieľ a negatívne informácie – udávajú oblasť, kde cieľ neleží. Každý referenčný bod meria oneskorenie k ostatným referenčným bodom a ukladá do tabuľky, ktorú pravidelne aktualizuje. K hodnotám oneskorenia priradí vzdialenosť (pozícia referenčných bodov je známa, tým pádom aj vzdialenosť medzi nimi). Ďalej si každý referenčný bod vypočíta pomer medzi oneskorením a vzdialenosťou, pomocou ktorého si určí hranice obmedzenia (napr. pre oneskorenie 50 ms sa cieľ T nachádza medzi $2500 \text{ km} < T < 4000 \text{ km}$). Metóda Octant ďalej pracuje s váhovými koeficientmi, ktoré priraduje jednotlivým referenčným bodom, na základe zameraného oneskorenia. Čím menšia hodnota oneskorenia, tým vyššia hodnota váhového koeficientu. Predpokladá sa, že odhad polohy nameraný z referenčných bodov s veľkou vzdialenosťou je menej pravdepodobný, ako z referenčných bodov s menšou vzdialenosťou. Ako konečná poloha sa určí ťažisko vzniknutej oblasti.



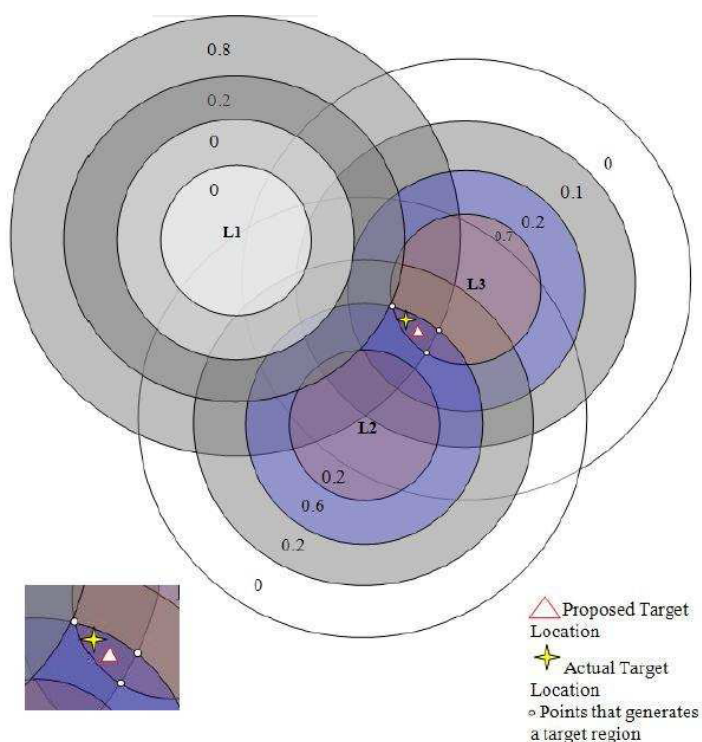
Obr. 1.2: Metóda Octant – vymedzenie oblastí pomocou R_p – pozitívna informácia a R_n – negatívna informácia. Princíp pridelenia váhových koeficientov na základe vzdialenosti – menšia vzdialenosť = vyššia váha.

1.2.6 GeoWeight

Metóda GeoWeight podľa [2] na základe minimálneho a maximálneho oneskorenia si pre každý referenčný bod vytvorí región, kde môže ležať cieľ, ohraničený týmito oneskoreniami. Tento región môže byť veľmi veľký a počítanie s ním môže mať za následok veľkú chybu merania. Preto sa región rozdeľuje na menšie oblasti o rozličných váhach. Jednotlivé váhy určujú pravdepodobnosť výskytu cieľa v danej oblasti (vyššia váha = väčšia pravdepodobnosť výskytu). Váha sa potom priradí pomocou prepočítavacej tabuľky, ktorá môže vypadáť ako tabuľka 1.1.

Ping Time (ms)	0-250 km	250-500 km	500-750 km	750-1000 km
100	0	0	0,2	0,8
35	0,2	0,6	0,2	0
15	0,7	0,2	0,1	0

Tab. 1.1: Tabuľka pre učenie váhy k regiónom u metódy GeoWeight.



Obr. 1.3: Pridelenie váh k regiónom u metódy GeoWeight pre $L1 = 100$ ms, $L2 = 35$ ms a $L3 = 15$ ms. Pridelenie váh je podľa tabuľky 1.1. Prezaté z [2].

Konečná poloha cieľa sa určí ako stred regiónu, ktorý vznikol ako prienik regiónov s najväčšími váhami. Princíp je zobrazený na obr. 1.3.

2 ONESKORENIE

Pod pojmom oneskorenie si môžeme predstaviť čas, ktorý je potrebný na prenesenie určitých dát z jedného miesta na druhé. Je to súčet všetkých faktorov, ktoré „zdržujú“ správu počas trasy do cieľa.

Oneskorenie sa rozdeľuje na jednosmerné a obojsmerné. Jednosmerné oneskorenie (z ang. *one-way-delay*) je čas, ktorý uplynie medzi vyslaním paketu a jeho prijatím v cieľi. Obojsmerné oneskorenie (z ang. *round-trip-time*) je vlastne suma jednosmerného oneskorenia od zdroja k cieľu a od cieľa ku zdroju plus oneskorenie spracovaním cieľovou stanicou.

2.1 Oneskorenie v medzilahľých uzloch

Veľkosť oneskorenia ovplyvňujú do istej miery aj aktívne sieťové prvky. Je to čas, ktorý strávi správa vo vnútri smerovača – proces smerovania. Smerovanie je zložitý proces, ktorý má na oneskorenie značný vplyv. Závisí na množstve softvérových a hardvérových faktoroch, ako napr. sieťový protokol, výpočtová rýchlosť procesoru smerovača, množstvo pamäte a iné. Oneskorenie v medzilahľých uzloch pozostáva z viacerých zložiek, ktoré sú podrobnejšie popísané v nasledujúcich sekciách.

2.1.1 Oneskorenie dobou spracovania inforámca

Označené D_{pc} , je čas, ktorý je potrebný na spracovanie paketu a určenie odchádzajúceho rozhrania. Podľa [3] musí smerovač vykonať niekoľko krokov, ktoré zvyšujú oneskorenie. Ako prvé musí skontrolovať správnu veľkosť rámca a zistiť, či je jeho kontrolná suma FCS (Frame Check Sequence) správna. Ďalej sa kontroluje hlavička IP paketu, či nie je paket určený pre lokálnu sieť. Ak nie, musí overiť TTL. Ak hodnota TTL nie je väčšia ako 1, paket je zahodený a odosielateľ je upozornený ICMP (Internet Control Message Protocol) správou. Ak však TTL je väčšie ako 1, smerovač pokračuje vyhľadávaním v smerovacej tabuľke, v ktorej hľadá vhodný záznam k IP adrese príjemcu. Ďalej smerovač zisťuje, či tento záznam odkazuje na východzie rozhranie, ktorým má paket odoslať. Ak neodkazuje na východzie rozhranie, potom odkazuje na IP adresu ďalšieho skoku. V tomto prípade si adresu smerovač zapamätá a vráti o krok späť. Ak však odkazuje na východzie rozhranie, tak pokračuje tým, že v ARP (Address Resolution Protocol) tabuľke vyhľadá informácie na vytvorenie rámca na linkovej vrstve. Následne smerovač musí znížiť hodnotu TTL a znovu prepočítať kontrolnú sumu FCS. Ako posledné smerovač zapúzdri paket pomocou informácií vyhľadaných v ARP tabuľke do nového rámca a odošle.

Oneskorenie dobou spracovania inforámca je v každom smerovači rôzne, no napriek týmto premenlivým faktorom môžeme toto oneskorenie určiť ako konštantnú hodnotu pre každý uzol $D_{pc}^i = D_{pc}$ pre $i = 1, \dots, H$. Podľa [9] budeme používať hodnotu

$$D_{pc} \approx 100 \mu\text{s}. \quad (2.1)$$

2.1.2 Oneskorenie vo fronte

Označené D_q , je podľa [9] doba, počas ktorej musí správa čakať, kým sa dostane do fronty na odoslanie. V prípade, že prepínač prijme dáta z viacerých zdrojov a majú byť odoslané na rovnaký výstupný port, musí ich radiť do fronty. Klasická počítačová sieť nerozlišuje druh ani pôvod paketu a snaží sa každý paket doručiť v čo najkratšej dobe tzv. metódou **best effort**. Všetky pakety majú rovnakú šancu na doručenie, ale aj na zahodenie, pokiaľ príde k zahlteniu siete. Pokiaľ chceme uprednostniť napríklad multimedialne služby, musíme implementovať do siete mechanizmus pre zaistenie kvality služieb QoS (Quality of Service). Potom sa vyberajú jednotlivé dátové toky a určujú prioritu podľa ich dôležitosti.

V prípade, že správa nebude čakať vo fronte, môžeme toto oneskorenie zanedbať

$$D_q = 0 \text{ s}. \quad (2.2)$$

2.1.3 Iné oneskorenia

Smerovač môže vykonávať okrem smerovania aj radu ďalších operácií, ktoré majú vplyv na zväčšenie oneskorenia. Sú to niektoré z voliteľných konfigurovateľných operácií ako napr.:

- Filtrovanie dát pomocou prístupových zoznamov¹ (ACL – Access Control List).
- Preklad adries (NAT – Network Address Translation).
- Smerovacia metóda
- Šifrovanie/dešifrovanie.

Tieto oneskorenia môžeme podľa [3] vyjadriť ako

$$D_o \approx 100 \mu\text{s}. \quad (2.3)$$

¹Každý ACL sa vždy prechádza od začiatku, postupne jedno pravidlo za druhým, až pokiaľ nedôjde k zhode.

2.1.4 Serializačné oneskorenie

Označené D_{tr} , je čas potrebný na umiestnenie správy na prenosové médium. Pre správu o veľkosti p a médium o kapacite C môžeme vyjadriť serializačné oneskorenie ako

$$D_{\text{tr}} = p/C. \quad (2.4)$$

Serializačné oneskorenie je vzhľadom k celkovému oneskoreniu veľmi malé, tým pádom ho môžeme zanedbať. Napríklad ak použijeme packet o veľkosti $p = 56$ Bytes na linke s kapacitou $C = 1$ Gbps, serializačné oneskorenie sa rovná $D_{\text{tr}} = p/C = 0,448 \mu\text{s}$. Na linkách s menšou kapacitou je serializačné oneskorenie väčšie. Linky s kapacitou menšou ako 1 Gbps sú v hlavných chrbticových sieťach minulosťou. Serializačné oneskorenie teda môžeme zanedbať. [9].

$$D_{\text{tr}} \approx 0 \mu\text{s}. \quad (2.5)$$

2.2 Oneskorenie spôsobené rýchlosťou šírenia signálu

Označené D_{pg} , je čas, ktorý je potrebný na doručenie správy z jedného konca média na druhý. Toto oneskorenie je najviac smerodajné pre lokalizáciu, závisí totiž od dĺžky média. Podľa [5] je rýchlosť šírenia signálu v optickom kábli 2/3 rýchlosti svetla vo vákuu a rýchlosť šírenia signálu v metalickom vedení je 72% rýchlosti svetla vo vákuu. Pre médium s dĺžkou l a jeho rýchlosťou v môžeme vyjadriť toto oneskorenie ako

$$D_{\text{pg}} = l/v. \quad (2.6)$$

2.3 Závislosť oneskorenia na počte uzlov

Oneskorenie závisí na počte uzlov – hopov, cez ktoré musí správa prejsť, aby sa dostala do cieľa. Celkové oneskorenie zdroj(s) – cieľ(d) – zdroj(s) môžeme podľa [9] vyjadriť ako

$$d(s, d, s) = \sum_{i=1}^H (D_{\text{pc}}^i + D_{\text{q}}^i + D_{\text{tr}}^i + D_{\text{o}}^i + D_{\text{pg}}(n_{i-1}, n_i)), \quad (2.7)$$

kde H je počet uzlov a n jednotlivé uzly na celej trase. Z výrazu (2.7) je pre geografickú vzdialenosť smerodajný iba jeden parameter – D_{pg} . Zvyčajne používame na

výpočet celkového oneskorenia $d(s, d, s)$ všetky parametre oneskorenia, keďže D_{pg} nie je možné priamo zmerať. Pre určenie hodnoty $D_{pg}(s, d, s)$ musíme odčítať tieto parametre od celkového oneskorenia

$$D_{pg}(s, d, s) = d(s, d, s) - \sum_{i=1}^H (D_{pc}^i + D_q^i + D_o^i + D_{tr}^i). \quad (2.8)$$

Nakoľko nie je možné mať detailné informácie o všetkých uzloch, cez ktoré správa prechádza, určíme si konštantnú hodnotu pre každý uzol d_h . Na základe parametrov z (2.1), (2.2), (2.3) a (2.5) budeme ako konštantu oneskorenia v uzle používať

$$d_h = D_{pc} + D_q + D_o + D_{tr} = 200 \mu s. \quad (2.9)$$

V tomto prípade celkové oneskorenie spôsobené rýchlosťou šírenia signálu medzi zdrojom a cieľom môžeme napísať ako

$$\begin{aligned} D_{pg}(s, d, s) &= d(s, d, s) - \sum_{i=1}^H (D_{pc}^i + D_{tr}^i) \\ &= d(s, d, s) - H \cdot (D_{pc} + D_{tr}) \\ &= d(s, d, s) - H \cdot d_h. \end{aligned} \quad (2.10)$$

Vzťah poukazuje na to, že oneskorenie spôsobené rýchlosťou šírenia signálu je možné určiť z celkového oneskorenia $d(s, d, s)$, počtu uzlov H nachádzajúcich sa na meranej ceste a konštantného oneskorenia v uzle d_h .

2.4 Vytvorenie ICMP Echo Reply

Na meranie oneskorenia sa používa nástroj ping. Skladá sa z dvoch častí, zo správ **Echo Request** – požiadavka a **Echo Reply** – odpoveď, viď obr. 2.1. Oneskorenie vzniká po príchode požiadavky do cieľovej stanice, ktorá musí vygenerovať odpoveď. Tento čas, potrebný na vygenerovanie správy, si označíme D_g . Autori [9] vykonali merania, ktorými zisťovali čas potrebný k vygenerovaniu správy **Echo Reply**. Na základe meraní vytvorili histogram, viď obr. 2.2, ktorý zobrazuje zmerané D_g . Histogram má jednu špičku okolo $500 \mu s$ a väčšinu zmeraných hodnôt medzi $300 \mu s$ a $1000 \mu s$. V tejto práci sa používa minimálna hodnota

$$D_g = 300 \mu s, \quad (2.11)$$

aby toto oneskorenie neovplyvňovalo výpočet vzdialenosti.

```
C:\WINDOWS\system32\cmd.exe

C:\Documents and Settings\Student>ping 192.168.1.1

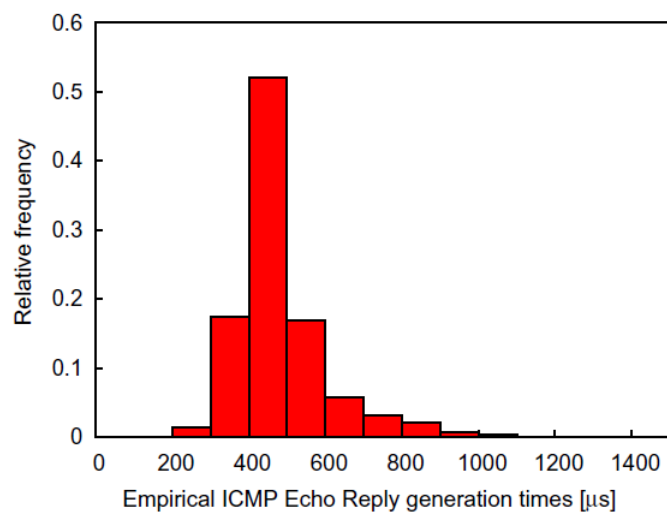
Příkaz PING na 192.168.1.1 s délkou 32 bajtů.

Odpověď od 192.168.1.1: bajty=32 čas=4ms TTL=255
Odpověď od 192.168.1.1: bajty=32 čas=9ms TTL=255
Odpověď od 192.168.1.1: bajty=32 čas=4ms TTL=255
Odpověď od 192.168.1.1: bajty=32 čas=7ms TTL=255

Statistika ping pro 192.168.1.1:
Pakety: Odeslané = 4, Přijaté = 4, Ztracené = 0 (ztráta 0%),
Přibližná doba do přijetí odezvy v milisekundách:
Minimum = 4ms, Maximum = 9ms, Průměr = 6ms

C:\Documents and Settings\Student>
```

Obr. 2.1: Výpis nástroja ping pre zapojenie s jedným uzlom. Dôležité údaje ako TTL a minimálne oneskorenie je vyznačené červenou farbou.



Obr. 2.2: Histogram času potrebného na vygenerovanie Echo Reply. Prevzaté z [9].

3 NÁSTROJ PING

Na jednoduché meranie RTT (Round Trip Time) sa najčastejšie používa nástroj ping, ktorý je súčasťou takmer každého operačného systému. Medzi jeho hlavné výhody patrí jednoduchosť, ale hlavne dostupnosť, pretože sa za jeho používanie nemusí platiť. Pôvodne bol vyvinutý na zisťovanie dostupnosti jednotlivých zariadení alebo staníc v sieti. Využíva správy **Echo Request** – požiadavka a **Echo Reply** – odpoveď. Zdrojová stanica odosiela požiadavky cieľovej stanici. V prípade, že sieť a jej prvky sú nakonfigurované správne, cieľová stanica obdrží požiadavku, ktorú spracuje a odošle odpoveď. Ping ku svojej činnosti využíva protokol ICMP. Výpis nástroja ping je znázornený na obr. 2.1.

Z výpisu sú pre prácu dôležité hodnoty TTL a minimálna doba oneskorenia. Minimálna doba oneskorenia je hodnota RTT, potrebná na prepočet oneskorenia na vzdialenosť.

Hodnota TTL je číslo, ktoré obmedzuje dobu platnosti dát, alebo počet priechodov paketov cez aktívne prvky počítačovej siete. Toto obmedzenie chráni sieť pred zahltením, ktoré by mohli spôsobiť pakety zacyklené v nekonečných slučkách (spôsobené napríklad chybou, alebo nesprávnym nastavením smerovania). Pole TTL je nastavené na predvolenú hodnotu pri vytvorení dát a automaticky sa znižuje o 1 pri priechode akýmkoľvek smerovačom. Pri dosiahnutí nuly je paket zahodený.

Hodnoty TTL sa podľa [18] automaticky nastavujú na typické hodnoty:

- 30,
- 32,
- 60,
- 64,
- 128,
- 200,
- 254,
- 255.

Keďže sa hodnota TTL znižuje o 1 pri priechode každým smerovačom, môže sa efektívne využiť na zistenie počtu uzlov. Dôležité však je, že pri vytvorení správy **Echo Reply** sa hodnota TTL nastaví na predvolenú hodnotu, tým pádom sa zistí počet uzlov len na ceste z cieľa ku zdroju. Túto predvolenú hodnotu TTL je možné len odhadnúť z typických hodnôt.

4 LOKALIZÁCIA POMOCOU ONESKORENIA

Oneskorenie, ktoré sa nameria z referenčného bodu, sa prepočíta na vzdialenosť. Tým sa získa kružnica okolo tohto bodu, ktorá vytvorí oblasť, kde sa môže cieľová stanica nachádzať. Pri vysielaní z viacerých referenčných bodov sa na priesečníku všetkých kružníc nachádza cieľová stanica. V ideálnom prípade sa všetky kružnice pretnú v jednom bode.

4.1 Vzťah medzi oneskorením a vzdialenosťou

Výpočet budeme realizovať pre jeden referenčný bod. Reálna geografická vzdialenosť, označená l medzi zdrojom(s) a cieľom(d), je podľa [9] vyjadrená ako

$$l(s, d) = \frac{1}{2} \cdot c \cdot r \cdot (d(s, d, s) - (H_{fw} + H_{bw}) \cdot d_h - D_g), \quad (4.1)$$

kde c je rýchlosť svetla vo vákuu, H_{fw} a H_{bw} je počet uzlov – hopov v doprednom a spätnom smere a r je nový parameter, popísaný v ďalšej sekcii. Pre správnu funkčnosť vzorca musí platiť $0 < r < 1$.

4.1.1 Hodnota parametru r

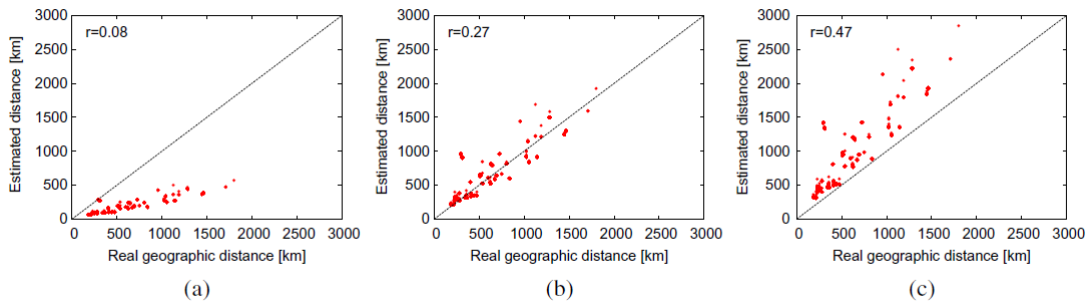
Pri meraní oneskorenia je nutné zohľadniť aj fakt, že sieťové káble nevedú priamo z bodu do bodu, sú zahnuté, prekonávajú prevýšenia, zákruty... To isté platí pre uzly na celej ceste, neležia na jednej priamke. Tieto faktory sú veľmi premenlivé a u každého výpočtu iné, každopádne zvyšujú oneskorenie. Parameter r zohľadňuje všetky tieto faktory a navyše je v ňom zahrnutá konštanta šírenia signálu po médiu. Autori [9] vykonali mnoho meraní vo výzkumnej sieti GEANT2. Výsledky nameraných hodnôt parametru r sú zobrazené v grafoch na obr. 4.1.

Vlastné meranie hodnoty parametru r

Parameter r , kľúčový pre prepočet oneskorenia na vzdialenosť, je možné vyjadriť pomocou vzorca (4.1)

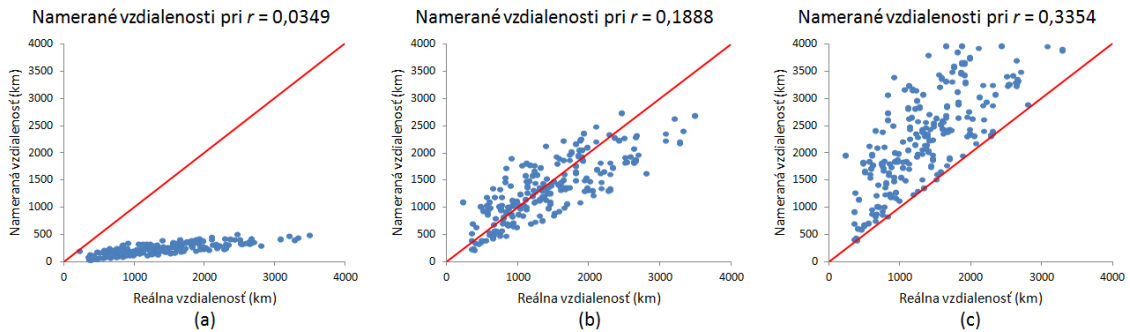
$$r = \frac{2 \cdot l(s, d)}{c \cdot (d(s, d, s) - (H_{fw} + H_{bw}) \cdot d_h - D_g)}. \quad (4.2)$$

Pre výpočet je nutné poznať vzdialenosť medzi meranými bodmi, RTT oneskorenie a počet medziľahlých uzlov. Meranie prebiehalo medzi všetkými servermi použitými v aplikácii.



Obr. 4.1: Hodnoty parametru r pri meraní vzdialeností, kde (a) ukazuje minimálnu hodnotu, (b) priemernú a (c) maximálnu. Prevzaté z [9].

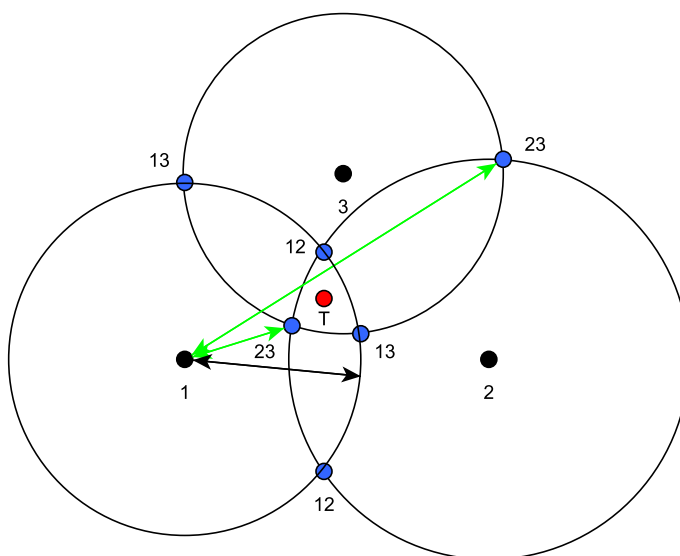
Z každého serveru sa pomocou nástroja `ping` zmeria RTT oneskorenie ku všetkým ostatným serverom. Z výpisu sa pomocou poľa TTL zistí počet medzilahlých uzlov, ktorý je ale potrebné vynásobiť dvakrát (predpokladá sa symetrické smerovanie). Pomocou týchto parametrov sa vypočíta hodnota r pre danú kombináciu serverov. Zistená minimálna, priemerná a maximálna hodnota parametru r sa rovná 0,0349, 0,1888 a 0,3354. Výsledky merania vzdialeností s týmito hodnotami r sú zobrazené v grafe 4.2. Pri spustení aplikácie je prednastavená hodnota r na priemernú hodnotu 0,1888, ktorú je však možné zmeniť na ľubovoľnú hodnotu v rozsahu $0 < r < 1$.



Obr. 4.2: Hodnoty parametru r pri vlastnom meraní vzdialeností. Graf (a) ukazuje minimálnu hodnotu r , graf (b) priemernú a graf (c) maximálnu.

4.2 Nájdenie súradníc hľadaného bodu

Na určenie konkrétneho miesta je potrebné poznať polohu minimálne troch serverov a vzdialenosti od týchto serverov k cieľu. Získame tri kružnice, ktoré sa v ideálnom prípade pretnú v jednom bode. Keď neuvažujeme ideálny prípad, získame 6 priesečníc, ako môžeme vidieť na obrázku 4.3. Z týchto priesečníc je potrebné vybrať tri správne, medzi ktorými leží hľadaný bod. Princíp výberu správnych priesečníc je zobrazený na obr. 4.3.



Obr. 4.3: Priesečníky troch kružníc. Čierna farba označuje polohu serverov, červená farba hľadaný cieľ, modrou farbou sú označené všetky priesečníky. Zelené šípky znázorňujú vzdialenosti od serveru 1 k priesečnikom kružníc serverov 2 a 3. Správny priesečník pre určenie cieľa je ten, ktorý má menšiu alebo rovnakú vzdialenosť ako polomer kružnice zo serveru 1.

4.2.1 Určenie súradníc priesečníc

Hľadanie priesečníku je realizované vždy pre dvojicu serverov. Celkom treba hľadať priesečníky trikrát. Prvým krokom je vypočítanie vzdialenosti medzi jednotlivými servermi. Označme si súradnice prvého serveru ako lat_1 a lon_1 , súradnice druhého serveru lat_2 a lon_2 . Potom pre výpočet vzdialenosti medzi servermi môžeme podľa [16] použiť `haversine` formulu, pomocou ktorej môžeme vypočítať najkratšiu vzdia-

lenosť medzi dvoma bodmi na povrchu Zeme.

$$\begin{aligned}
 a &= \sin^2 \left(\frac{lat_2 - lat_1}{2} \right) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2 \left(\frac{lon_2 - lon_1}{2} \right) \\
 c &= 2 \cdot \text{atan2} \left(\sqrt{a}, \sqrt{(1-a)} \right) \\
 l &= R \cdot c,
 \end{aligned} \tag{4.3}$$

kde R je polomer Zeme $R \approx 6371$ km. Aby sme našli priesečníky, musí byť splnená podmienka pre pretnutie kružníc

$$l < (r_1 + r_2), \tag{4.4}$$

kde r_1 a r_2 sú vzdialenosti od serverov. Namerané vzdialenosti sa delia deliteľom s určitým krokom, pokiaľ platí podmienka (4.4). Týmto postupom sa však dosiahne nežiadúci stav, pri ktorom kružnice nezostanú pretnuté, viď obr. 4.5 a). Preto je potrebné vzdialenosti nakoniec vynásobiť predposledným deliteľom aby, zostali kružnice pretnuté s najmenšími možnými vzdialenosťami, viď obr. 4.5 b). Túto úpravu je potrebné vykonať pre všetky tri vzdialenosti, aby zostal zachovaný pomer medzi vzdialenosťami. Tým získame najmenšie možné vzdialenosti a priblížime priesečníky najbližšie k cieľu. Z priesečníkov sa vytvorí trojuholník, ktorého ťažisko značí hľadaný cieľ viď obr. 4.4.

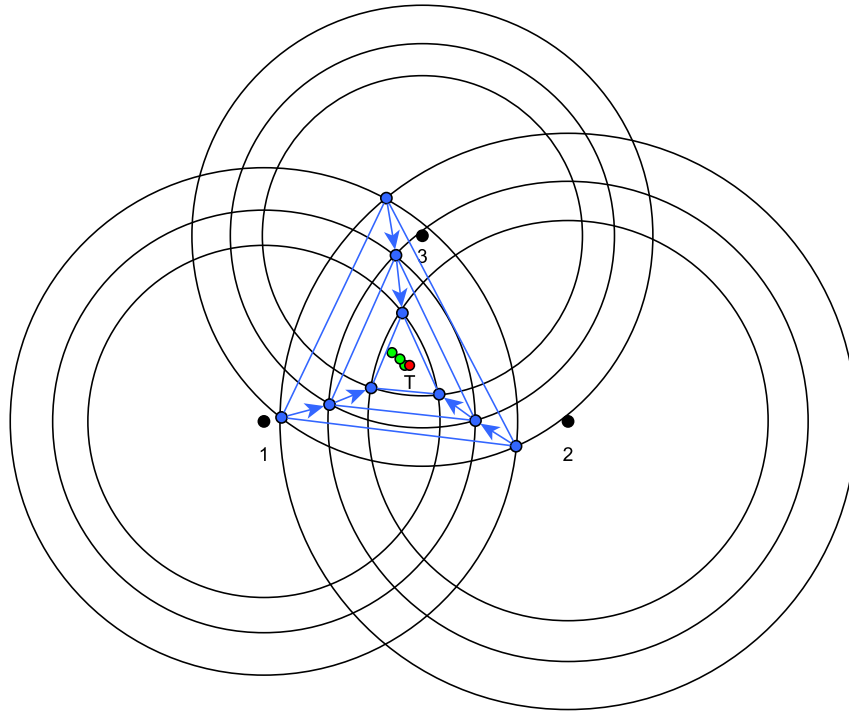
Podľa [16] je možné z počiatočného bodu na Zemi so súradnicami lat_1 , lon_1 a pomocou azimutu a vzdialenosti určiť súradnice cieľového bodu lat_x a lon_x

$$\begin{aligned}
 lat_x &= a \sin \left(\sin(lat_1) \cdot \cos \left(\frac{l}{R} \right) + \cos(lat_1) \cdot \sin \left(\frac{l}{R} \right) \cdot \cos(b) \right) \\
 lon_x &= lon_1 + \text{atan2} \left(\sin(b) \cdot \sin \left(\frac{l}{R} \right) \cdot \cos(lat_1), \cos \left(\frac{l}{R} \right) - \sin(lat_1) \cdot \sin(lat_x) \right),
 \end{aligned} \tag{4.5}$$

kde l je vzdialenosť a b je azimut. Tento bod je potrebné hľadať z obidvoch serverov. Je nutné použiť cyklus, ktorý 360 krát zmení azimut u prvého serveru o 1° a zároveň u druhého serveru taktiež zmení každým cyklom azimut o 1° , ale v rozsahu celých 360° viď obr. 4.6. Tento cyklus zaisťuje postupne vystriedanie všetkých kombinácií. Akonáhle vzdialenosť medzi týmito dvoma bodmi spadne pod určitú hranicu tolerancie¹, cyklus sa zastaví a za priesečník sa prehlási bod, ktorý je priemerom z týchto dvoch bodov. Pre prípad na obr. 4.6 je to priemer zo súradníc 45;14 a 45; 16 = 45; 15.

Tým získame jeden priesečník. Druhý priesečník vytvoríme ako obraz cez bod A , viď obr. 4.6, čo je bod, cez ktorý prechádza priamka spájajúca priesečníky kružníc,

¹Hranicu tolerancie je možné nastaviť pred štartom aplikácie.



Obr. 4.4: Zmenšovanie vzdialeností a priblíženie ťažiska k cieľu. Čierna farba označuje polohu serverov, červená farba hľadaný cieľ, modrá farba značí priesečníky a trojuholník z nich vytvorený, zelená farba určuje ťažisko trojuholníka.

kolmá na úsečku d . Vzdialenosť z prvého serveru k bodu A vypočítame podľa [10] ako

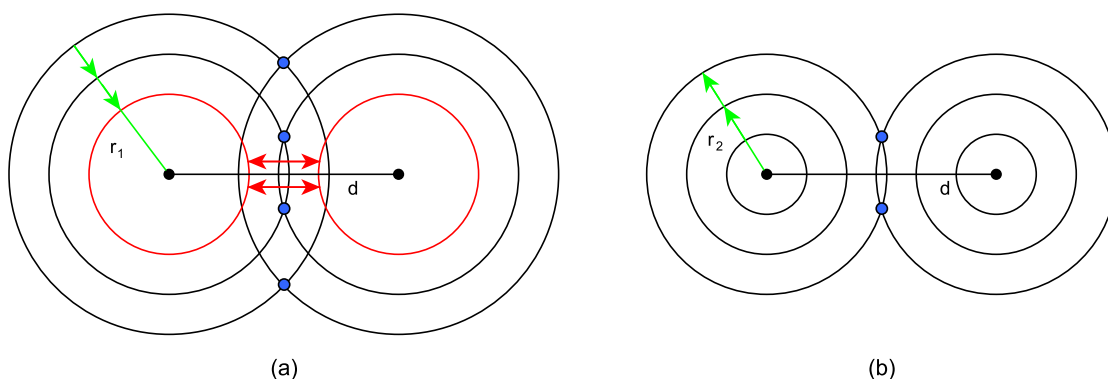
$$a = \frac{r_1^2 - r_2^2 + l^2}{2 \cdot l}. \quad (4.6)$$

Teraz je potrebné určiť súradnice tohto bodu, lat_A a lon_A . Najskôr je nutné konvertovať sférické súradnice do Kartézskej 3D sústavy. Podľa [22] je prevod do súradníc x, y, z nasledujúci

$$\begin{aligned} x &= R \cdot \sin(lat) \cdot \cos(lon) \\ y &= R \cdot \sin(lat) \cdot \sin(lon) \\ z &= R \cdot \cos(lat). \end{aligned} \quad (4.7)$$

Pomocou týchto súradníc sa dajú určiť súradnice bodu A

$$\begin{aligned} x_A &= x_1 + \frac{(x_2 - x_1) \cdot a}{d} \\ y_A &= y_1 + \frac{(y_2 - y_1) \cdot a}{d} \\ z_A &= z_1 + \frac{(z_2 - z_1) \cdot a}{d}. \end{aligned} \quad (4.8)$$



Obr. 4.5: Princíp znižovania a zvyšovania kružníc pri hľadaní priesečníkov. Prípad a) podmienka: ak platí $l < r_1 + r_2$ zmeň vzdialenosti – ukazuje možnú chybu pri znižovaní vzdialeností – priesečníky nie je možné nájsť. Prípad b) podmienka: ak platí $l > r_1 + r_2$ zveď vzdialenosti – vytvorí práve dva priesečníky pri najmenších vzdialenostiach.

Posledným krokom je spätný prevod do sférických súradníc lat_A a lon_A

$$\begin{aligned} lat_A &= \text{asin} \left(\frac{z_A}{R} \right) \\ lon_A &= \text{atan2} \left(\frac{y_A}{x_A} \right). \end{aligned} \quad (4.9)$$

Pomocou tohto bodu určíme druhý priesečník vid', obr. 4.6.

4.2.2 Výber priesečníkov a nájdenie cieľa

Tento postup sa opakuje aj pre ďalšie dvojice serverov (1 a 3, 2 a 3). Tým získame 6 priesečníkov vid', obr. 4.3, z ktorých je potrebné vybrať z každej dvojice práve jeden správny priesečník. Správny priesečník medzi servermi 1 a 2 je možné získať porovnaním vzdialeností medzi priesečníkmi a stredom kružnice zo serveru 3 vid', obr. 4.3. Týmto spôsobom určíme správne priesečníky u každej dvojice serverov. Zostanú tri priesečníky – body tvoriace trojuholník. Odhadujeme, že v ťažisku tohto trojuholníka sa nachádza hľadaný bod.

5 RIEŠENIE

Aby bolo možné merať vzdialenosť určitej stanice v sieti internet, je potrebné toto meranie vykonať zo stanice, ktorej poloha je známa. Na meranie jednej vzdialenosti by stačil aj domáci počítač. Pri lokalizácii je ale nutné merať vzdialenosť z viacerých referenčných bodov so známou polohou. Pre tento účel bola zvolená experimentálna sieť PlanetLab.

5.1 PlanetLab

PlanetLab [12], [21] je experimentálna sieť, skladajúca sa z množstva uzlov umiestnených vo viac ako 40 krajinách. To umožňuje vedcom prevádzkovať svoje experimenty v reálnych internetových podmienkach. K dispozícii sú uzly, ktoré sú vzdialené tisíce kilometrov od seba. Software bežiaci na uzloch je linux. Prihlásenie na uzly je realizované pomocou protokolu SSH.

Pred použitím týchto zariadení, je nutné sa zaregistrovať na stránkach PlanetLabu. Pri registrácii treba zvoliť používanú site¹. V našom prípade je to CESNET - Czech Education and Research Network.

Aby bolo možné komunikovať s uzlami PlanetLabu, je potrebné vygenerovať dvojicu SSH-2 RSA kľúčov o dĺžke 1024 – verejný kľúč (z ang. `public key`) a súkromný kľúč (z ang. `private key`). Do PlanetLabu je potrebné nahrať verejný kľúč a súkromný kľúč si uchovať na bezpečnom mieste (pomocou tohto kľúča je možné sa prihlásiť na ktorýkoľvek uzol, z akéhokoľvek miesta). Trvá nejaký čas, kým sa verejný kľúč rozšíri medzi všetky uzly PlanetLabu.

Vygenerovať SSH kľúče môžeme pomocou operačného systému UNIX. V príkazovom riadku je potrebné zadať príkaz

```
ssh-keygen -t rsa -f ~/.ssh/id_rsa
```

Kľúče, ktoré sú vytvorené v systéme UNIX, je možné použiť v programovacom jazyku Java. Pomocou tohto kľúča ale nie je možné pracovať s programom putty.

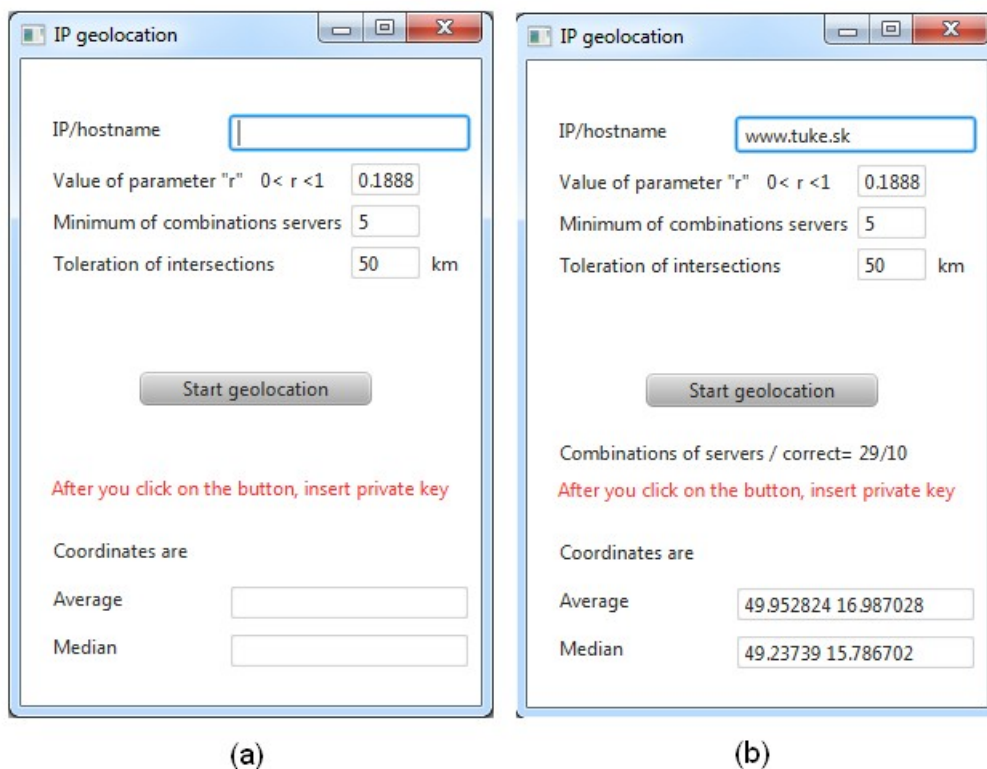
¹Site je autonómny systém organizácie, do ktorej patrí užívateľ.

5.2 Aplikácia

Aplikácia v tejto práci bola vytvorená v programovacom jazyku Java. Tento jazyk bol zvolený z niekoľkých dôvodov:

- je to moderný a v súčasnej dobe rozšírený programovací jazyk,
- programovací jazyk a vývojové prostredie je možné získať zdarma,
- vytvorené programy fungujú v každom operačnom systéme,
- umožňuje jednoduché vytvorenie spustiteľného súboru a nasadenie aplikácie na webové rozhranie.

Aplikácia po zadaní hľadanej IP adresy (príp. hostname), vypíše súradnice tejto stanice. V aplikácii je možné nastaviť hodnotu parametru r , minimálny počet kombinácií serverov, ktoré vykonávajú výpočet a hranicu tolerancie priesečníkov. Pri štarte aplikácie sú hodnoty prednastavené. Aplikácia je zobrazená na obr. 5.1.



Obr. 5.1: Vzhľad aplikácie vykonávajúcej lokalizáciu a) po spustení, b) po výpočte.

Program tvorí niekoľko metód, cyklov a podmienok. Prihlasuje sa na 22 serverov rozmiestnených po Európe. Všetky významné časti programu sú po poradí rozobrané v nasledujúcich sekciách.

5.2.1 Výpočet vzdialeností

Po stlačení tlačítka „Start geolocation“ je nutné do dialógového okna vložiť private key. Pri chybnom kľúči program vráti hlášku „Incorrect private key“. Ak je kľúč správny, overí sa pole pre zadanie hostname. Pokiaľ je prázdne, program vráti hlášku „Incorrect hostname“. Pokiaľ nie je pole hostname prázdne, ale host neodpovedá, program končí výpočet so záverom „Not responding“. Ak host odpovedá, program pokračuje cyklom, ktorý zaistí prihlásenie na každý server a načíta oneskorenie a TTL vrátené metódou `ssh` do premenej `vypis`. Ďalej vypočíta vzdialenosť od každého serveru k cieľu.

```
for (int i=0; i<pocetServerov; i++){
    vypis[i] = ssh(server[i],i);
    vzdialenost[i] = Float.valueOf(distance
        (delay(vypis[i]), ttl(vypis[i]),hodnotaR));
}
```

Prihlásenie na server

Prihlásenie na server prebieha pomocou knižnice `jcrafft`, ktorá je voľne stiahnuteľná na <http://www.jcraft.com/jsch/>. Metóda je nazvaná `ssh` a vstupujú do nej dva parametre:

- `server` – IP adresa, alebo meno serveru, kam sa chceme prihlásiť (odkiaľ sa má merať vzdialenosť),
- `poradie` – poradové číslo serveru, v prípade merania hodnoty parametru `r` (pri meraní `r` musí byť ošetrený stav, aby sa nemerala vzdialenosť medzi tým istým serverom, kvôli tomu do metódy vstupuje `poradie`).

Pri vytváraní spojenia je nutné zadať ďalšie parametre – `user`, `port`, `privateKey`. `User` je prihlasovacie meno pre danú site. `Port` definuje službu, SSH používa 22. `PrivateKey` je súkromný kľúč, bez ktorého sa nie je možné prihlásiť na servery (je potrebné ho vložiť do aplikácie). Metóda vracia hodnotu v podobe textového reťazca – `String`.

```
public String ssh(String server, int poradie){
    JSch jsch = new JSch();
    String host = server; //vstupny parameter
    String user = "cesnet_feec";
    int port = 22;

    jsch.addIdentity(privateKey);
    Session session = jsch.getSession(user, host, port);
```

```

java.util.Properties config = new java.util.Properties();
config.put("StrictHostKeyChecking", "no");
session.setConfig(config);
session.connect();

//ďalej nasleduje spustenie pingu a načítanie rtt a ttl
...

```

V prípade, že SSH server neodpovedá, je na konci metódy vytvorená podmienka, ktorá prideli veľkú hodnotu oneskorenia, tým pádom pri výpočte vzdialenosti získame veľmi veľkú hodnotu. V cykle pri výpočte súradníc nebude potom tento server zahrnutý do výpočtu, viď podmienka v cykle v sekcii 5.2.4.

Výber hodnoty oneskorenia a TTL

Metóda `ssh`, vráti textový reťazec v tvare napr.:

```

64 bytes from 147.229.10.1: icmp_ seq=5 ttl=252 time=1.03 ms.
rtt min/avg/max/mdev = 0.884/0.986/1.087/0.087 ms

```

Aby bolo možné vypočítať vzdialenosť, je nutné z tohto reťazca vybrať hodnoty oneskorenia a TTL. Pre vybranie hodnoty minimálneho oneskorenia bola vytvorená metóda `delay`, ktorá rozdelí vstupný `String` hneď ako narazí na znak „=“. Tým dostaneme 5 častí, z ktorých je nutné ešte rozdeliť druhú časť:

```

0.884/0.986/1.087/0.087 ms64 bytes from 147.229.10.1: icmp_ seq=

```

po každom znaku „/“. Tým sme získali minimálnu hodnotu RTT, v tomto prípade 0,884.

Pre vybranie hodnoty TTL bola vytvorená metóda `ttl`, ktorá pracuje na rovnakom princípe ako metóda `delay`.

Výsledné číslo musí však prejsť podmienkami, ktoré zisťujú, aká hodnota TTL bola nastavená pri generovaní `Echo Reply`. Potom od tejto hodnoty odčíta hodnotu získanú zo `Stringu`.

Môže nastať ešte jedinečný prípad, ošetrovaný podmienkou, ak by bol cieľ vzdialený iba jeden uzol. V tomto prípade by sa hodnota TTL neznížila a odrátala by sa od prednastavenej. Tým pádom by bol výsledok 0 a vo výpočte by nebolo možné ďalej pokračovať.

Prepočet na vzdialenosť

Prepočet oneskorenia na vzdialenosť je naprogramovaný podľa rovnice (4.1). Do metódy vstupujú tri parametre – oneskorenie a celkový počet uzlov, získané v predchádzajúcich metódach, a hodnota parametru r zadaná pred štartom výpočtu.

```
public float distance(float delay, float hop, float r){
    float prepočet = (float)
        (0.5*300000000*r*(((delay*Math.pow(10, -3))
        -(hop*2*(200*Math.pow(10, -6)))-300*Math.pow(10, -6))) /1000);
    if(prepočet<0){
        prepočet = Float.POSITIVE_INFINITY;}
    }
    return(prepočet);
}
```

V kóde je vytvorená podmienka, ktorá ošetruje nežiadúci stav – vypočítaná vzdialenosť je záporná a nebolo by možné ďalej pokračovať v lokalizácii. V tomto prípade sa vypočítanej vzdialenosti priradí hodnota nekonečno.

5.2.2 Hľadanie priesečníkov

Po získaní všetkých vzdialeností od serverov nasleduje výpočet priesečníkov jednotlivých dvojíc serverov. Aby bolo možné nájsť priesečníky, musíme vypočítať vzdialenosť medzi týmito servermi na povrchu Zeme. Metóda na výpočet vzdialenosti medzi dvoma servermi – `haversine`, je naprogramovaná podľa spomínanej `haversine` formule (4.3). Do metódy vstupujú 4 parametre – súradnice serverov (lat_1 , lon_1 , lat_2 a lon_2).

Ďalšou potrebnou metódou je získanie súradníc bodu, ktorý je určený počiatočnými súradnicami, azimutom a vzdialenosťou. Metóda sa volá `point` a je naprogramovaná podľa [16]. vstupujú do nej 4 parametre – súradnice serveru, azimut a vzdialenosť (lat_1 , lon_1 , $bearing$, l).

Pre výpočet priesečníkov slúži metóda `priesečníky`, do ktorej vstupujú súradnice serverov a ich vypočítané vzdialenosti. Metóda vracia vypočítané súradnice priesečníkov a chybu, ktorá sa nastaví na hodnotu 1 v prípade, že minimálna hodnota pri hľadaní priesečníkov nespadá pod hranicu tolerancie. Metóda je založená na princípe zobrazenom na obr. 4.6. V nasledujúcom kóde sú zobrazené najvýznamnejšie časti:

```
for (float i=0; i<360; i++){
    for(float j=0; j<360; j++ ){
        bearing1 = i;
```

```

bearing2 = j;

priesečník1 = point(p1x1, p1y1, bearing1, vzd1);
priesečník2 = point(p2x1, p2y1, bearing2, vzd2);
//následuje vytiahnutie hodnôt lat a lon
vzdialenost1 = haversin(lat1, lon1, lat3, lon3);
//ďalej sa porovnáva aktuálna vzdialenosť s minimálnou,
súradnice sa určia pomocou najmenšej vzdialenosti
...

```

5.2.3 Získanie súradníc cieľa

Na získanie súradníc cieľa je potrebné mať priesečníky z minimálne troch serverov – to je 6 priesečníkov. Pre výber správnych priesečníkov a výpočet súradníc cieľa slúži metóda `trilateration`.

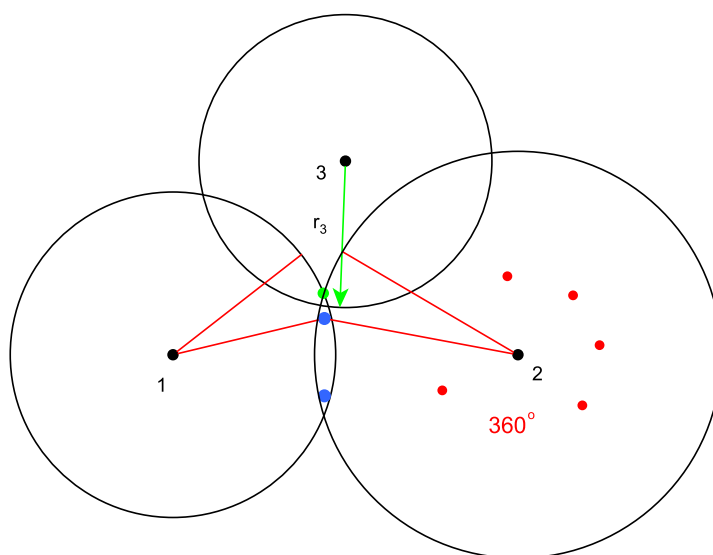
Názov metódy nesúvisí s princípom, na ktorom pracuje. Do metódy `trilateration` vstupujú tri servery – ich súradnice x , y , a tri vzdialenosti namerané týmito servermi. Metóda vracia vypočítané súradnice cieľa a chybu, ktorá sa nastaví na hodnotu 1 v prípade, že do nej vstúpil priesečník z metódy `priesečníky`, ktorý nespadá pod hranicu tolerancie. Výber správneho priesečníka je naprogramovaný podľa obr. 4.3.

```

if(haversin(lat1,lon1,p3x,p3y)<haversin(lat2,lon2,p3x,p3y)){
    p1x1 = lat1;
    p1y1 = lon1;}
else{
    p1x1 = lat2;
    p1y1 = lon2;}
//na rovnakom princípe sa vyberú ostatné správne priesečníky,
ďalej sa z troch priesečníkov vytvorí ťažisko vzniknutého
trojuholníka a vráti súradnice + chybu s hodnotou 0
...

```

Výber správnych priesečníkov v programe musí byť ošetrený voči nežiadúcemu stavu zobrazenému na obr. 5.2. V tomto prípade by podmienke podľa obr. 4.3 nevyhovoval ani jeden priesečník. Preto sa v programe porovnávajú vzdialenosti medzi priesečníkmi a stredom kružnice navzájom a nie voči polomeru.



Obr. 5.2: Chyba, ktorá môže nastať pri určovaní správneho priesečníka. Zelený bod ukazuje správny priesečník a zelená šípka polomer kružnice, v ktorej leží správny priesečník. Program môže určiť priesečník mimo kružnicu 3 (znázornený modrou farbou), pričom sú reálne pretnuté v zelenom bode. V tomto prípade by ani jeden priesečník nevyhovoval podmienke zobrazenej na obr. 4.3.

5.2.4 Cyklus na kombinácie serverov a výpočet súradníc

Vyššie uvedené metódy vypočítajú súradnice cieľa pre tri servery. Pre výpočet viacerých súradníc je nutné vytvoriť viaceré kombinácie po tri servery. Cyklus `while` sa opakuje dovtedy, pokiaľ nie je splnený minimálny počet kombinácií, zadaný pred štartom aplikácie.

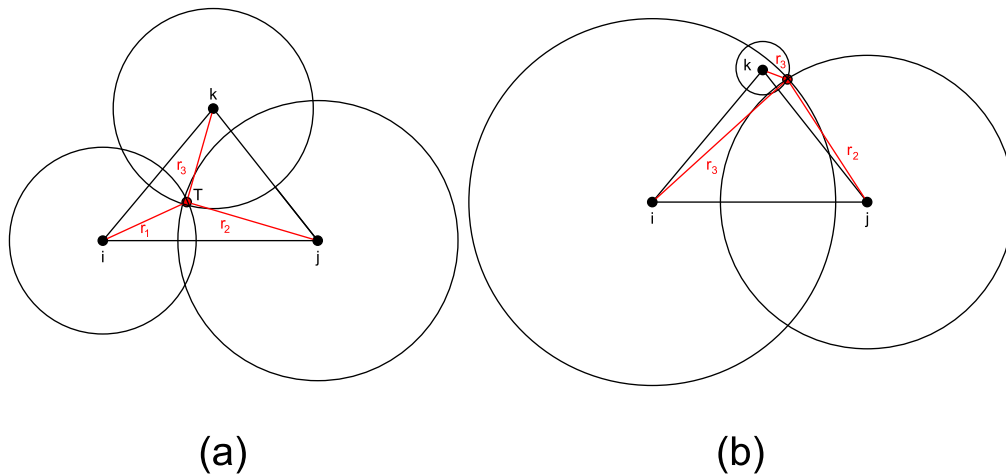
Cyklus `for` zaisťuje vystriedanie všetkých kombinácií troch serverov. Cyklus je ošetrený podmienkou, aby nenastala kombinácia serverov napr. 111 alebo 112, v týchto prípadoch nie je možné vypočítať súradnice.

Podmienka taktiež zaisťuje, aby sa kombinácie serverov neopakovali. Mohlo by sa totiž stať, že pri ďalších podmienkach budú vyhovovať len 2 kombinácie serverov, napríklad 123 a 456. Výpočet súradníc by sa ale opakoval aj pre kombináciu 132, 213 a 231 (tak isto aj pre servery 456). Vo výsledku by sme získali 8 súradníc, pričom by len 2 boli rozdielne. To je nežiadúce.

Následujúca podmienka zaisťuje, aby boli vybrané servery s najmenšou vzdialenosťou – tie, ktoré ležia najbližšie k cieľu. Vzdialenosť sa postupne inkrementuje o 100 km.

Poslednou podmienkou je, aby hľadaný bod ležal vo vnútri trojuholníka, vytvoreného servermi. Pre každý server musí platiť, aby zmeraná vzdialenosť bola menšia

ako vzdialenosť k ostatným serverom. Touto podmienkou sa ošetrí aj nežiadúci prípad kružnice v kružnici. Podmienka je znázornená na obr. 5.3.



Obr. 5.3: Podmienka zaistujúca aby, hľadaný bod ležal vo vnútri trojuholníka. Akákoľvek vzdialenosť zo servera musí byť menšia ako vzdialenosť od servera ku ostatným serverom. Prípad a) ukazuje splnenú podmienku, prípad b) ukazuje nespĺnenú podmienku od serveru i .

Pokiaľ nie sú splnené všetky podmienky a hrozilo by zacyklenie, je na konci kódu vložený tzv. „break“, ktorý preruší cyklus, akonáhle vzdialenosti prídu na hodnotu 4000 km.

Ak nastane tento prípad, program pokračuje rovnakým cyklom, ale bez podmienky, aby hľadaný bod ležal v trojuholníku.

Výsledné súradnice

Po výpočte všetkých kombinácií, ktoré vyhovejú podmienkam, je potrebné z jednotlivých získaných súradníc vytriediť bezchybné. Na to slúži cyklus, ktorý z poľa x , kde sú uložené výsledné súradnice v tvare $(lat, lon, chyba)$, prepustí k ďalšiemu počítaniu len s hodnotou $chyba = 0$. Posledným krokom celého procesu je vypočítať výsledné súradnice. Prvý variant je výpočet priemeru zo všetkých súradníc. Druhý variant vypočíta súradnice pomocou mediánu. Obidva výsledky sa zobrazia po skončení výpočtu aplikácie.

6 VÝSLEDKY MERANIA

Merané stanice pre lokalizáciu boli vyberané z datasetu, ktorý je voľne prístupný na stránkach výskumnej skupiny zaoberajúcej sa geolokáciou, pracujúcou na Ústavu telekomunikácií. Dataset je platný pre Európu a obsahuje IP adresu, alebo hostname uzlu a jeho súradnice. Pri meraní bolo použitých 22 serverov – landmarkov.

Meranie bolo vykonávané pre tri hodnoty parametru r – minimálnu, priemernú a maximálnu, zistenú meraním popísaným v sekcii 4.1.1. Aplikácia vypočíta výsledné súradnice dvomi spôsobmi – pomocou priemeru a mediánu. V tabuľke 6.1 sú zobrazené odchýlky merania pri použití priemeru aj mediánu. Ďalej zobrazuje minimálnu a maximálnu odchýlku. Najlepšie výsledky boli dosiahnuté pri použití priemernej hodnoty parametru $r = 0,1888$ viď tabuľka 6.1.

r	Priemer				Medián			
	max.	min.	priemer	medián	max.	min.	priemer	medián
(–)	(km)	(km)	(km)	(km)	(km)	(km)	(km)	(km)
0,0349	1433	84	586	509	1351	112	560	517
0,1888	1081	43	374	346	1049	75	386	328
0,3354	924	17	416	403	895	42	401	403

Tab. 6.1: Porovnanie chýb pri určovaní polohy. Tabuľka ukazuje maximálne a minimálne hodnoty, priemernú odchýlku a odchýlku vypočítanú pomocou mediánu. Porovnávajú sa chyby pri určovaní súradníc vypočítaných pomocou priemeru a mediánu, pri minimálnej, priemernej a maximálnej hodnote parametru r .

V nasledujúcich tabuľkách sú zobrazené počty výsledkov pre určité intervaly vzdialeností. Pre súradnice vypočítané pomocou priemeru je najviac výsledkov v intervale od 250 do 500 km pre všetky tri hodnoty parametru r viď tabuľka 6.2. Pre súradnice vypočítané pomocou mediánu je najviac výsledkov v intervale od 250 do 500 km pre minimálnu a priemernú hodnotu parametru r . Pre maximálnu hodnotu je najviac výsledkov v intervale od 0 do 250 km viď tabuľka 6.3.

	Priemer			
r	(0–250) km	(250–500) km	(500–750) km	(750 <) km
0,0349	10	19	14	19
0,1888	22	25	10	5
0,3354	19	23	14	6

Tab. 6.2: Počet výsledkov v intervaloch vzdialeností pri použití priemeru pre rôzne hodnoty parametru r .

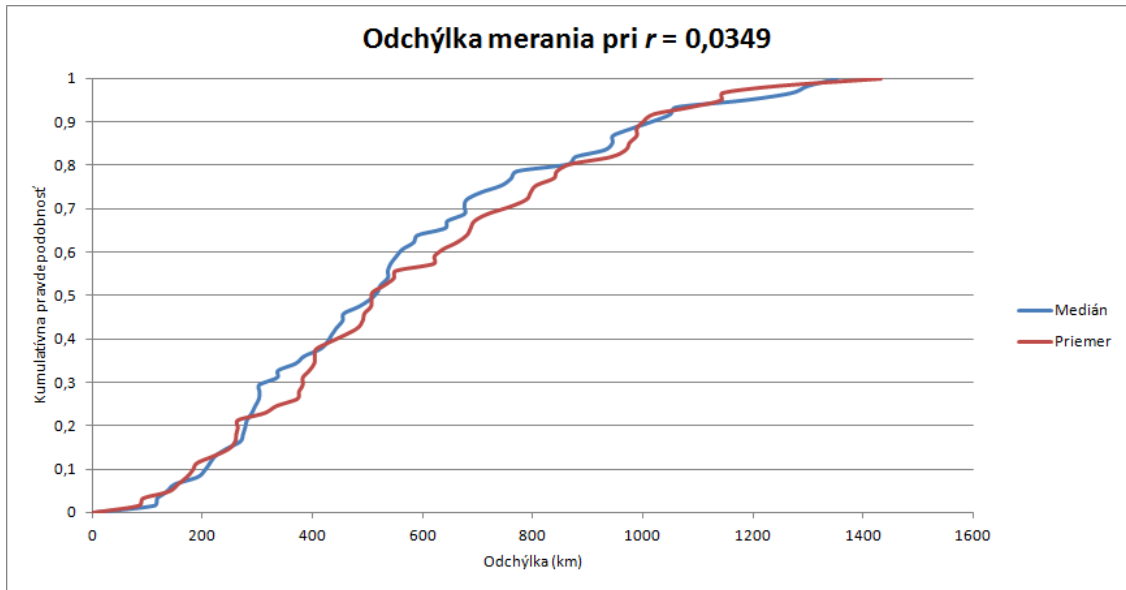
	Medián			
r	(0–250) km	(250–500) km	(500–750) km	(750 <) km
0,0349	10	20	17	15
0,1888	18	28	13	3
0,3354	21	19	17	5

Tab. 6.3: Počet výsledkov v intervaloch vzdialeností pri použití mediánu pre rôzne hodnoty parametru r .

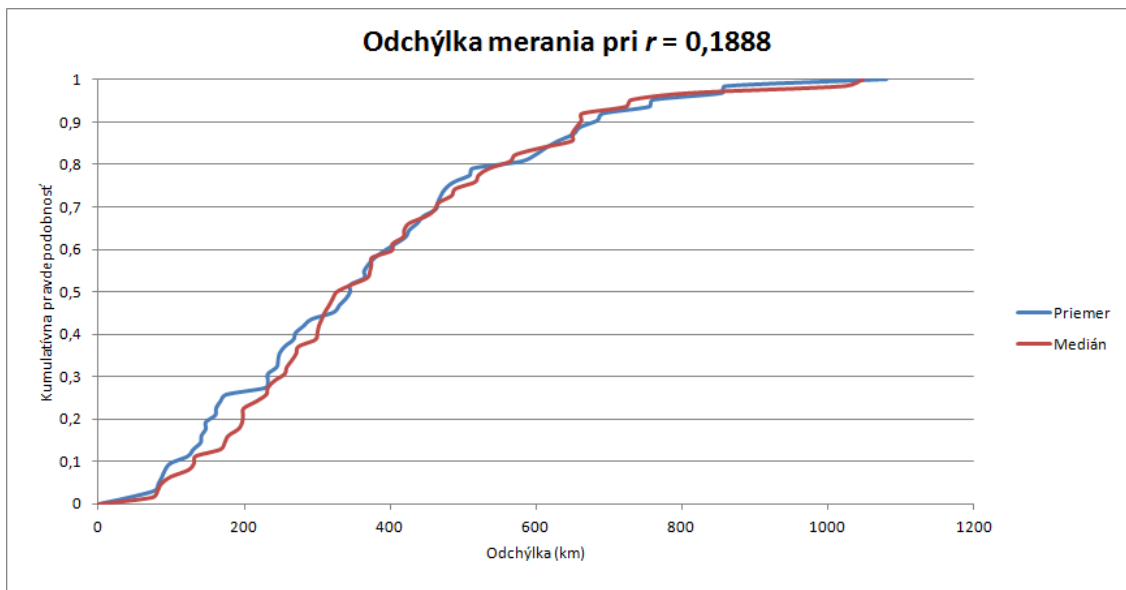
V nasledujúcich grafoch sú porovnané odchýlky merania pre výsledky priemeru a mediánu pre jednotlivé hodnoty parametru r . Grafy sú vytvorené ako kumulatívna distribučná funkcia pre jednotlivé výsledky.

Kumulatívna distribučná funkcia, ďalej len CDF (z ang. Cumulative Distribution Function) zobrazuje pravdepodobnosť výsledku pre danú hodnotu, alebo nižšiu. [11]. Zjednodušene povedané pre tento prípad, určuje pravdepodobnosť správneho určenia polohy do určitej odchýlky. Napríklad pri použití minimálnej hodnoty r zobrazenej v grafe na obr. 6.1, je 34,4 % pravdepodobnosť určenia správnej polohy s maximálnou odchýlkou 400 km pri použití priemeru. Pri použití mediánu, je pravdepodobnosť správneho určenia polohy s maximálnou odchýlkou 400 km 36,9 %.

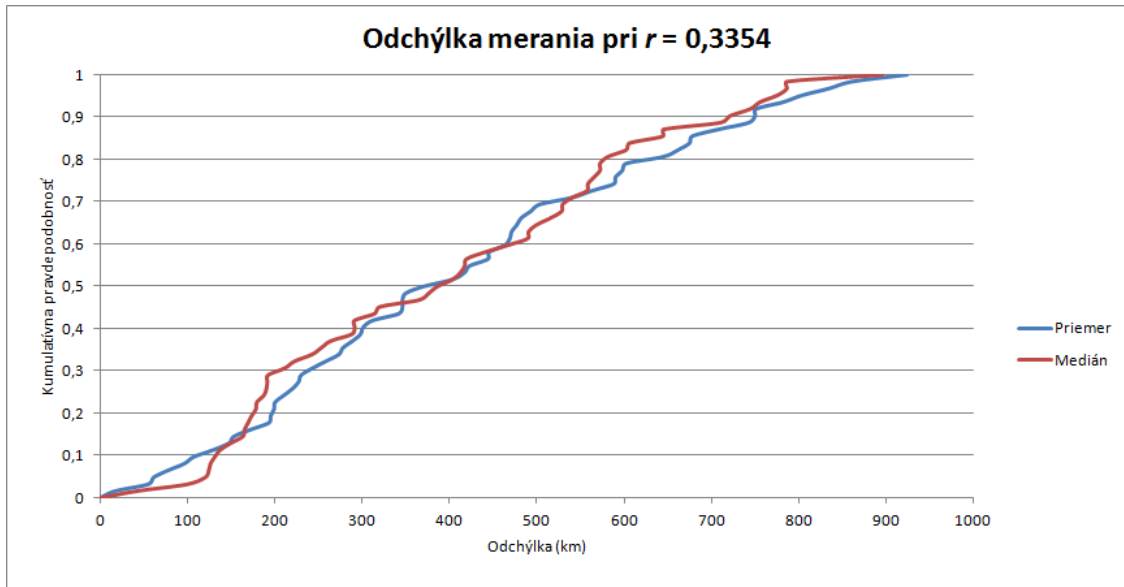
V grafe na obr. 6.2 je zobrazená CDF pre priemernú hodnotu r a v grafe na obr. 6.3 pre maximálnu hodnotu r .



Obr. 6.1: Porovnanie odchýliek merania pri hodnote $r = 0,0349$.

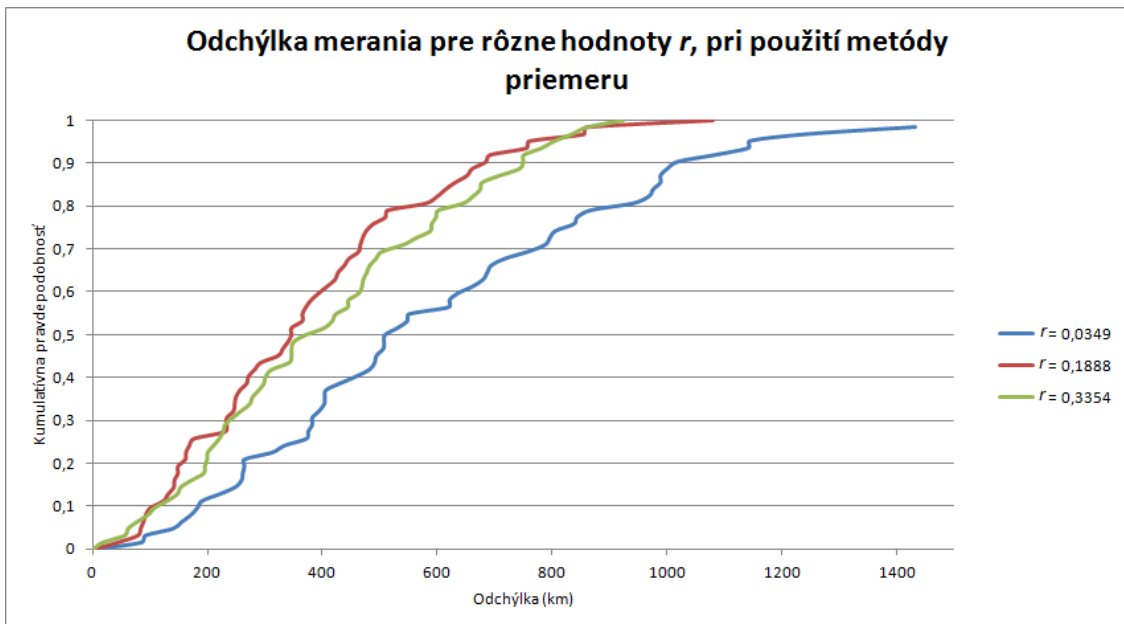


Obr. 6.2: Porovnanie odchýliek merania pri hodnote $r = 0,1888$.

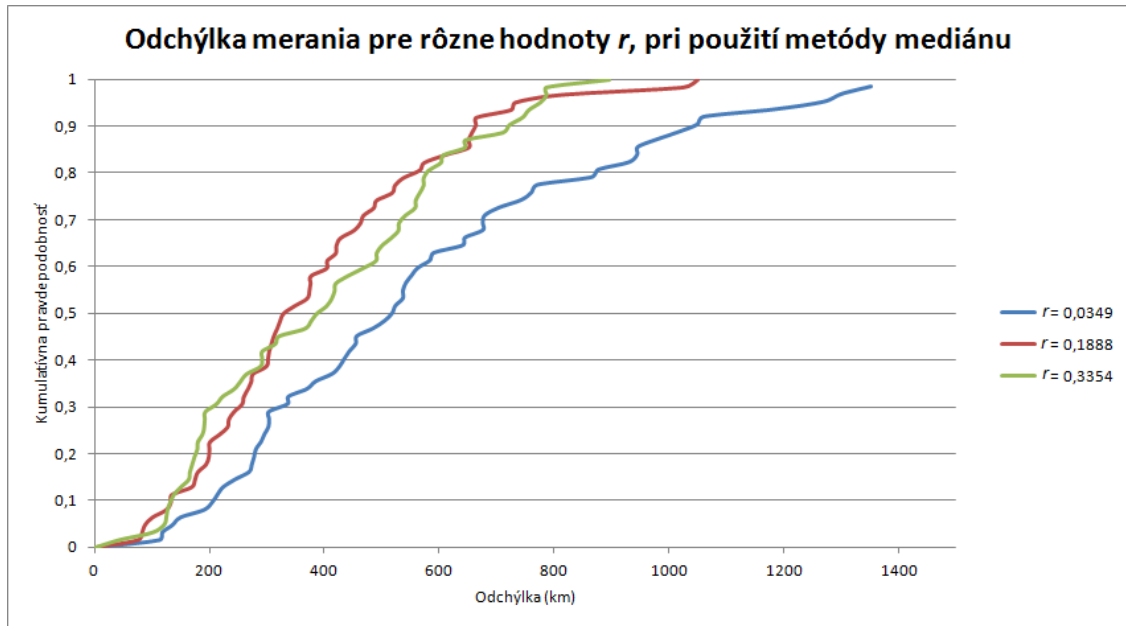


Obr. 6.3: Porovnanie odchýliek merania pri hodnote $r = 0,3354$.

V nasledujúcich grafoch sú porovnané odchýlky merania pre minimálnu, priemernú a maximálnu hodnotu parametru r , v grafe na obr. 6.4 pri použití priemeru a v grafe na obr. 6.5 pri použití mediánu. Grafy sú vytvorené ako CDF.



Obr. 6.4: Odchýlka merania pre rôzne hodnoty r , pri použití metódy priemeru.



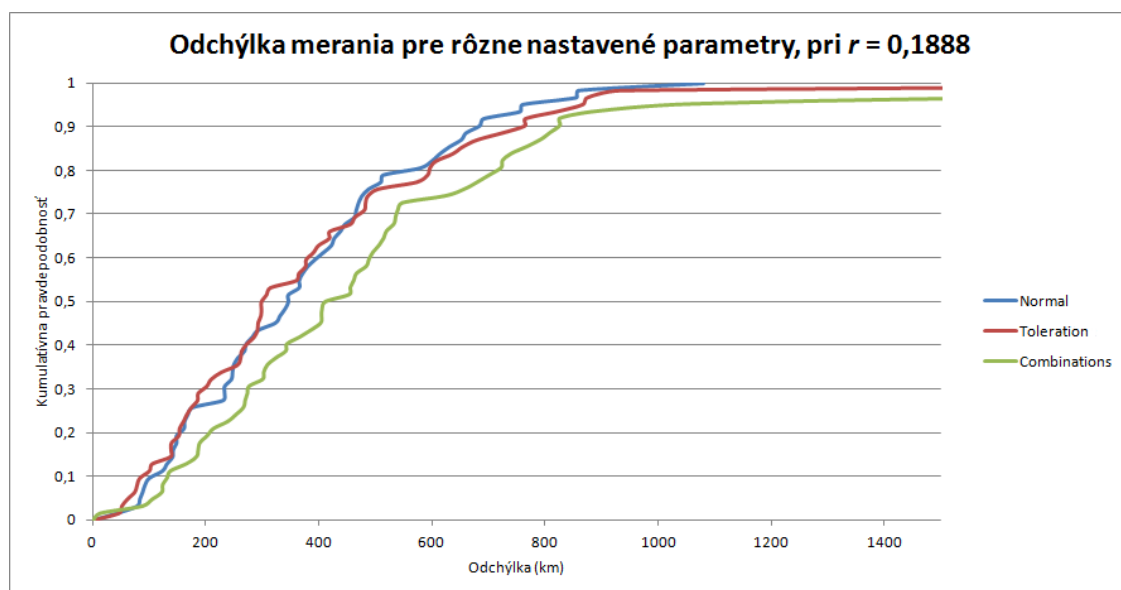
Obr. 6.5: Odchýlka merania pre rôzne hodnoty r , pri použití metódy mediánu.

Následujúce meranie bolo vykonané pri zmenených parametroch minimálneho počtu kombinácií serverov a minimálnej hranice tolerancie pri určení priesečníkov. Hodnota parametru r je pri týchto meraniach zhodná. Vybraná bola na základe tabuľky 6.1, kde najmenšiu odchýlku dosahuje priemerná hodnota 0,1888. Výsledky merania sú zobrazené v tabuľke 6.4.

Na základe tabuľky 6.4 bol vytvorený graf, ktorý porovnáva odchýlky merania pri parametroch nastavených podľa tejto tabuľky. V grafe sú porovnané výsledky pri použití metódy priemeru vid' obr.6.6. Pri väčšom počte kombinácií je odchýlka väčšia, je to spôsobené tým, že do výpočtu sú zahrnuté aj servery, ktoré majú veľké oneskorenie od cieľa. Tým pádom je menej pravdepodobná presnosť prepočtu na vzdialenosť.

	Priemer		Medián	
	priemer	medián	priemer	medián
$r = 0,1888$				
	(km)	(km)	(km)	(km)
combinations = 5; toleration = 50	374	346	386	328
combinations = 5; toleration = 10	407	307	425	346
combinations = 25; toleration = 50	509	432	518	439

Tab. 6.4: Porovnanie chýb pri meraní s inou hranicou tolerancie a počtom kombinácie serverov. Tabuľka ukazuje priemernú odchýlku a odchýlku vypočítanú pomocou mediánu. Porovnávajú sa chyby pri určovaní súradníc vypočítaných pomocou priemeru a mediánu. Parameter r je zhodný pre všetky merania (0,1888). Položka combinations značí minimálny počet kombinácií serverov, položka toleration označuje minimálnu hranicu tolerancie pri určení priesečníkov v km.



Obr. 6.6: Odchýlka merania pre rôzne nastavené parametre, pri použití hodnoty $r = 0,1888$. Parametre sú nastavené nasledovne: Normal – Počet kombinácií serverov = 5, hranica tolerancie = 50 km. Toleration – Počet kombinácií serverov = 5, hranica tolerancie = 10 km. Combinations – Počet kombinácií serverov = 25, hranica tolerancie = 50 km.

7 ZÁVER

Úlohou bakalárskej práce bolo vytvoriť aplikáciu, ktorá je schopná zistiť polohu stanice na základe merania oneskorenia.

V teoretickej časti sú rozobrané súčasné možnosti lokalizácie, ktoré sa delia na pasívne a aktívne metódy. Pasívne sú veľmi presné a rýchle, ale len v prípade, že obsahujú správne záznamy v databázach. Aktívne metódy používajú na lokalizáciu meranie oneskorenia, prípadne iných sieťových prvkov. Oneskorenie sa potom prepočíta na vzdialenosť a následne sa určí poloha.

Oneskorenie a jeho jednotlivé časti, ktoré ho ovplyvňujú, sú popísané v teoretickej časti. Z nameraného oneskorenia sa vypočíta geografická vzdialenosť. Čím je oneskorenie väčšie, tým je aj vzdialenosť väčšia.

Pre výpočet geografickej polohy je najdôležitejšia premenná parameter r . Meraním sa zistila minimálna, priemerná a maximálna hodnota, viď grafy na obr. 4.2.

Aplikácia v tejto práci používa nástroj `ping` k meraniu RTT oneskorenia. Z tejto hodnoty sa potom vypočíta vzdialenosť podľa [9]. Výpočet výsledných súradníc bol vykonaný pre dve varianty – výpočet pomocou priemeru a mediánu.

Pre minimálnu hodnotu r sú výsledky najmenej presné, priemerná odchýlka pri použití priemeru je 586 km a pre medián 560 km. Najlepšie výsledky boli dosiahnuté u priemernej hodnoty r , kde priemerná odchýlka pre priemer je 374 km a pre medián 386 km. Pri použití maximálnej hodnoty r je priemerná odchýlka pri použití priemeru 416 km a mediánu 401 km. Výsledky sú prehľadne zobrazené v tabuľke 6.1.

LITERATÚRA

- [1] ADÁMEK, D. *Implementace QoS v přístupové síti*. Brno: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií, 2008. 86 s. Vedúci diplomovej práce bol prof. Ing. Miloslav Filka, CSc.
- [2] ARIF, M., J., et al. *GeoWeight: Internet Host Geolocation Based on a Probability Model for Latency Measurements*. In *Proc. 33rd Australasian Computer Science Conference*. Brisbane, Australia, 2010. 10 s.
- [3] BEDNÁR, J. *Velikost zpoždění aktivních prvků sítě*. Brno: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2010. 59 s. Vedúci bakalárskej práce bol Ing. Jiří Balej.
- [4] BOURKE P. *Intersection of a Line and a Sphere (or circle)*. [online] April 1992 [cit. 2013-05-14]. Dostupné z WWW: <<http://paulbourke.net/geometry/circlesphere/>>.
- [5] CURTIS, J. *Signal Propagation: Is copper faster than fiber?* [online] 1.11.1999 [cit. 2012-11-27] Dostupné z WWW: <<http://www.networkworld.com/newsletters/lans/0111lan1.html>>.
- [6] 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, Ústav telekomunikací, 2012. 77 s. Vedúci diplomovej práce bol doc. Ing. Dan Komosný, Ph.D.
- [7] GUEYE, B., et al. *Constraint-based Geolocation of Internet Hosts*. IMC 2004, Taormina, Sicily, Italy, October 25-27, 2004. 6 s.
- [8] KATZ-BASSETT, E., et al. *Towards IP Geolocation Using Delay and Topology Measurements*. IMC 2006, Rio de Janeiro, Brazil, October 25-27, 2006. 13 s. ACM 1-59593-561-4/06/0010
- [9] LAKI, S., et al. *A Model Based Approach for Improving Router Geolocation, Computer Networks*. In *The International Journal of Computer and Telecommunications Networking*. Budapešť, Hungary, 2010. 12 s. pp. 1490-1501 . ISBN: 978-1-4244-2846-5.
- [10] LECOCQ, D. *How to Calculate the Intersection of Two Circles*. [online] [cit. 2013-04-15]. Dostupné z WWW: <http://www.ehow.com/how_8205342_calculate-intersection-two-circles.html>.

- [11] MARTZ, E. *What's the Difference Between Probability and Cumulative Probability?* [online] 13.1.2012 [cit. 2013-05-17]. Dostupné z WWW: <<http://blog.minitab.com/blog/understanding-statistics/difference-between-probability-and-cumulative-probability>>.
- [12] MOFASSIR, U., H., PAWLIKOWSKI K. *Step By Step Guide for Using PlanetLab*. University of Canterbury. The Department of Computer Science and Software Engineering, 15.3.2011. 8 s. Dostupné z WWW: <<http://planetlabnz.canterbury.ac.nz/docs/Step%20By%20Step%20Guide%20for%20Using%20PlanetLab.pdf>>.
- [13] PADMANABHAN, V. N., LAKSHMINARAYANAN, S. *An Investigation of Geographic Mapping Techniques for Internet Hosts*. SIGCOMM 2001, San Diego, California, USA, August 27-31, 2001. 13 s. ACM 1-58113-411-8/01/0008
- [14] PECINOVSKÝ, R. *Myslíme objektově v jazyku Java 1.5*. 29.6.2004. 433 s.
- [15] SHAVITT, Y., ZILBERMAN, N. *A Study of Geolocation Databases*. 1.7.2006. 14 s. arXiv:1005.5674v3.
- [16] VENESS Ch. *Calculate distance, bearing and more between Latitude/Longitude points*. [online] 2012, [cit. 2013-05-05]. Dostupné z WWW: <<http://www.movable-type.co.uk/scripts/latlong.html>>.
- [17] VERNER, L., KOMOSNÝ, D. *Geolokace síťových zařízení v internetových sítích*. Elektorevue [online] 17.6.2011, 33, [cit. 2011-12-20]. Dostupný z WWW: <<http://www.elektorevue.cz/cz/clanky/komunikacni-technologie/0/geolokace-sitovych-zarizeni-v-internetovych-sitich/>>. ISSN 1213-1539.
- [18] *binbert.com* [online]. 8.12.2009 [cit. 2013-04-21]. Default Time To Live (TTL) values. Dostupné z WWW: <<http://www.binbert.com/blog/2009/12/default-time-to-live-ttl-values/>>.
- [19] *canseverayberk.blogspot.cz* [online] [cit. 2012-11-27]. SSH client connection example with JCraft's Java Secure Channel (JSch). Dostupné z WWW: <<http://canseverayberk.blogspot.cz/2011/10/ssh-client-connection-example-with.html>>.
- [20] *jcrafft.com* [online] [cit. 2012-11-27]. JSch - Examples. Dostupné z WWW: <<http://www.jcraft.com/jsch/examples/>>.
- [21] *planet-lab.org* [online] 2.1.2006, [cit. 2012-11-27]. User's Guide. Dostupné z WWW: <<http://www.planet-lab.org/doc/guides/user>>.

- [22] *rbrundritt.wordpress.com* [online]. 2008-10-14 [cit. 2013-5-14]. Conversion between Spherical and Cartesian Coordinates Systems. Dostupné z WWW: <<http://rbrundritt.wordpress.com/2008/10/14/conversion-between-spherical-and-cartesian-coordinates-systems/>>.
- [23] *technet.microsoft.com* [online] Október 2007, [cit. 2012-11-27]. Hodnoty time-to-live (TTL). Dostupné z WWW: <[http://technet.microsoft.com/cs-cz/library/cc731236\(v=ws.10\).aspx](http://technet.microsoft.com/cs-cz/library/cc731236(v=ws.10).aspx)>.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

a	Vzdialenosť z prvého bodu k bodu A pri priesečníku kružníc
ACL	Access Control List – Prístupový zoznam
ARP	Address Resolution Protocol – Sieťový protokol datalinkovej vrstvy
b	Bearing – Azimut
c	Rýchlosť svetla vo vákuu
C	Kapacita média
CBG	Constraint Based Geolocation
CDF	Cumulative Distribution Function – Kumulatívna distribučná funkcia
CESNET	Czech Education and Research Network
DNS	Domain Name System – Systém doménových mien
d	Delay – Oneskorenie
DV	Delay Vector- Vektor oneskorenia
D_{pc}	Processing delay – Oneskorenie dobou spracovania informácie
D_q	Queuing delay – Oneskorenie vo fronte
D_o	Other delays – Iné oneskorenia
D_{tr}	Transmission delay – Serializačné oneskorenie
D_{pg}	Propagation delay – Oneskorenie spôsobené rýchlosťou šírenia signálu
D_g	Delay of generation time – Oneskorenie vytvorením Echo Reply
FCS	Frame Check Sequence – Kontrolná suma pre rámce
H	Hop – Počet uzlov
H_{bw}	Hop backward – Počet uzlov v spätnom smere
H_{fw}	Hop forward – Počet uzlov v doprednom smere
ICMP	Internet Control Message Protocol – Signalizačný protokol sieťovej vrstvy

IP	Internet Protokol – Sieťový protokol
ISP	Internet Service Provider – Poskytovateľ internetových služieb
L	Landmark – Referenčný bod
l	Length – Vzdialenosť
lat	Latitude – Zemepisná šírka
lon	Longitude – Zemepisná dĺžka
NAT	Network Address Translation – Preklad adres
NNDS	Nearest Neighbor in Delay Space
p	Packet – Veľkosť paketu
QoS	Quality of Service – Kvalita služby
r	Parameter r
r_1, r_2	Polomery kružníc
R	Earth radius – Polomer Zeme
R_n	Polomer kružnice získaný na základe negatívnej informácie
R_p	Polomer kružnice získaný na základe pozitívnej informácie
RSA	Kryptografický algoritmus, ktorý vytvorili Rivest, Shamir a Adleman
RTT	Round Trip Time – Celkové oneskorenie
SSH	Secure Shel – Protokol na prihlásenie k vzdialenému terminálu
T	Target – Cieľ
TTL	Time To Live – Obmedzenie platnosti dát
TBG	Topology Based Geolocation

ZOZNAM PRÍLOH

A Obsah CD

51

A OBSAH CD

Priložený disk obsahuje nasledujúce položky:

- Elektronická verzia bakalárskej práce
- Dataset uzlov PlanetLabu
- Namerané hodnoty r
- Namerané hodnoty polôh pre rôzne r
- Namerané hodnoty polôh pri $r = 0, 1888$ a rôzne parametre
- Grafy CDF
- Vyexportovaný projekt so zdrojovým kódom
- Spustiteľný jar súbor (nutné zadať private key)