

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2016

Bc. Pavol Il'ko



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

MONITOROVÁNÍ PŘENOSOVÝCH PARAMETRŮ SÍTĚ INTERNET

MONITORING OF COMMUNICATION PROPERTIES IN INTERNET

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Pavol Il'ko

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Dan Komosný, Ph.D.

BRNO 2016



Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Pavol Il'ko

ID: 134505

Ročník: 2

Akademický rok: 2015/16

NÁZEV TÉMATU:

Monitorování přenosových parametrů sítě Internet

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s experimentální sítí PlanetLab – <http://www.planet-lab.org/>. Vytvořte seznam stanic sítě PlanetLab, které se nacházejí v Evropě. Pro tyto stanice zjistěte jejich přenosové parametry v síti Internet (dostupnost, časová prodleva komunikace). Tyto parametry analyzujte vzhledem ke geografické poloze serverů. Analyzujte také parametry služby vzdáleného připojení SSH (Secure Shell).

DOPORUČENÁ LITERATURA:

[1] PUŽMANOVÁ, R. TCP/IP v kostce. 2. vyd. Kopp, 2009. 620 s. ISBN: 978-80-7232-388-3.

[2] Linux Dokumentační projekt. 4. vyd. Computer Press, 2008. 1336 s. ISBN: 978-80-251-1525-1.

[3] COOPER, M. Advanced Bash-Scripting Guide. Steve Glines, 2010. 518 s. ISBN: 978-14-357-5218-4.

Termín zadání: 1.2.2016

Termín odevzdání: 25.5.2016

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

Konzultant diplomové práce:

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

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové 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

Táto práca sa zaoberá meraním prenosových vlastností siete Internet, konkrétne odozvy na program ping, protokol SSH a na meranie prenosovej rýchlosti. Práca je rozdelená na teoretickú a praktickú časť. V teoretickej časti je opísaná sieť PlanetLab, jej stručná história a aktuálne prebiehajúce projekty. Taktiež sú v nej opísané nástroje, pomocou ktorých je možné dolovať informácie z webových stránok. Tieto teoretické poznatky sú prakticky využité na automatické vytvorenie zoznamu staníc siete PlanetLab a na naprogramovanie aplikácií, ktoré merajú zadané prenosové vlastnosti siete. Vytvorené programy, zoznam staníc a namerané dáta sú priložené na DVD.

KĽÚČOVÉ SLOVÁ

PlanetLab, geolokácia, dolovanie dát, SSH, Ping, prenosová rýchlosť, Python, Bash

ABSTRACT

This thesis deals with measuring transmission parameters of the Internet network, in particular latency of ping, SSH protocol and bandwidth. The thesis is divided into a theoretical and a practical part. Theoretical part describes PlanetLab network, its brief history and contemporary projects. At the same time, tools for data mining from web pages are described. These information obtained from the theoretical part are used for creating PlanetLab nodes list and for programming applications which measure the network transmission parameters. Applications, list of nodes and obtained data are attached on DVD disc.

KEYWORDS

PlanetLab, geolocation, data mining, SSH, Ping, bandwidth, Python, Bash

ILKO, Pavol *Monitorování přenosových parametrů sítě Internet: diplomová práce*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 52 s. Vedúci práce bol doc. Ing. Dan Komosný, Ph.D.

VYHLÁSENIE

Vyhlasujem, že som svoju diplomovú prácu na tému „Monitorování přenosových parametrů sítě Internet“ vypracoval(a) samostatne pod vedením vedúceho diplomovej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

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

Brno

.....

podpis autora(-ky)

POĎAKOVANIE

Rád by som poďakoval vedúcemu diplomovej práce pánovi doc. Ing. Danovi Komosnému, PhD. za odborné vedenie, konzultácie a podnetné návrhy k práci. Taktiež by som rád poďakoval slečne Bc. Lýdii Rezničákovej za trpezlivosť, korekciu textu a hlavne za podporu, ktorú mi celé tie roky prejavovala. Chcem tiež poďakovať kamarátke Terezke, ktorá ma naučila skvelý recept na celozrnné palacinky.

Celozrnné palacinky na sladko

Doba prípravy je cca 30minút.

Na približne 8 palacínok budeme potrebovať:

- 130g celozrnej múky
- 120ml mlieka
- 120ml perlivej minerálnej vody
- 30g rozpusteného masla
- 2 vajcia
- štipka soli
- kvapka mandľového extraktu (pokiaľ nemáte, vanilkovým nič nepokazíte)
- rôzne ovocie, orechy, marmelády, šľahačka atď.

Príprava:

1. Preosejeme múku so štipkou soli do misy a pridáme všetky tekuté ingrediencie (mlieko, vodu, maslo, extrakt, vajce) a miešame, dokiaľ nevznikne tuhšie cesto.
2. Na rozpálenú panvicu s kvapkou oleja vylejeme kúsok cesta a rovnomerne ho rozotrieme po povrchu. Keď už sa palacinka zdá pevná, opečieme ju z druhej strany. Opečiť jednu palacinku trvá približne 3 minúty.
3. Zatiaľ čo sa palacinky opekajú môžeme nakrájať rôzne druhy ovocia a nachystať ingrediencie, ktoré máme na palacinkách radi.

Mojou obľúbenou kombináciou je nutella, banán a lieskové orechy posekané na drobno s trochou tvarohu a čerstvými lesnými plodmi. Dobrú chuť.

Brno

.....

podpis autora(-ky)



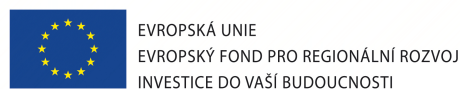
Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

POĎAKOVANIE

Výzkum popsaný v tejto diplomovej práci bol realizovaný v laboratóriách podporených projektom SIX; registračné číslo CZ.1.05/2.1.00/03.0072, operačný program Výzkum a vývoj pro inovace.

Brno

.....
podpis autora(-ky)



OBSAH

Úvod	10
1 Experimentálna sieť PlanetLab	11
2 Dolovanie dát z webových stránok	13
2.1 Nástroje pracujúce s webovým prehliadačom	13
2.2 Dostupné nástroje pracujúce z príkazového riadku	14
3 Program vytvorený na dolovanie dát z webových stránok	19
3.1 Varianta cez webový prehliadač	19
3.2 Varianta bez prehliadača	24
4 Meranie parametrov siete PlanetLab	29
4.1 Odozva na SSH	30
4.2 Odozva na ping	34
4.3 Meranie prenosovej rýchlosti	35
5 Analýza dosiahnutých výsledkov	40
5.1 Geografická analýza staníc siete PlanetLab	40
5.2 Výkonová analýza siete	42
6 Záver	48
Literatúra	50
Zoznam symbolov, veličín a skratiek	52

ZOZNAM OBRÁZKOV

3.1	Informácie o organizácii na webe siete PlanetLab	20
3.2	Prihlasovacie okno do webu https://www.planet-lab.org/	21
3.3	Vývojový diagram dolovania informácií z webu cez webový prehliadač	23
3.4	Vývojový diagram dolovania informácií z webu pomocou knižnice Re- quests	28
4.1	Vývojový diagram prekladu mena na IP adresu	29
4.2	Príprava na meranie prenosových parametrov	31
4.3	Vývojový diagram merania doby odozvy na SSH	33
4.4	Vývojový diagram merania dostupnosti serveru pomocou programu ping	36
4.5	Vývojový diagram merania prenosovej rýchlosti	39
5.1	Percentuálne rozdelenie staníc na kontinentoch	40
5.2	Graf zastúpenia staníc pre každý kontinent	41
5.3	Percentuálne rozdelenie hodnôt IP Flag	41
5.4	Percentuálny graf zobrazenia hodnôt IP Flag	42
5.5	Graf dostupnosti uzlov siete PlanetLab na program ping	43
5.6	Graf dostupnosti uzlov siete PlanetLab na protokol SSH	43
5.7	Najkratší nameraný čas, za ktorý sa bolo možné pripojiť ku stanici .	44
5.8	Priemerný čas potrebný na pripojenie ku stanici	45
5.9	Najdlhší nameraný čas, za ktorý sa bolo možné pripojiť ku stanici . .	45
5.10	Priemerná rýchlosť odosielania na server v sieti PlanetLab	47
5.11	Priemerná rýchlosť sťahovania zo serveru v sieti PlanetLab	47

ZOZNAM TABULIEK

3.1	Vstupné parametre programu <code>planetlab_list_creator.py</code>	25
3.2	Výstup programu <code>planetlab_list_creator.py</code>	27
4.1	Čiastočný zoznam staníc, z ktorých bolo vykonávané meranie	30
4.2	Použité argumenty pri pripájaní na server pomocou SSH	34
4.3	Použité argumenty programu <code>Rsync</code>	37
5.1	Priemerné hodnoty jednotlivých meraní	46

ÚVOD

Kapitola 1 tejto diplomovej práce oboznamuje so všeobecnými vlastnosťami experimentálnej siete PlanetLab. Predstavuje jej stručnú históriu, ako aj niektoré zaujímavé projekty realizované v tejto sieti.

Zadaním tejto práce je vytvoriť zoznam staníc siete, ktoré sa nachádzajú v Európe. Teoretickú časť, a teda technológie použité na dolovanie dát z webových stránok, opisujem v kapitole 2. Na webových stránkach siete PlanetLab má každá pripojená stanica zverejnené informácie, ktoré má zoznam obsahovať. Tieto informácie som mohol zo stránky extrahovať ručne, ale keďže sa jedná o viac ako 1000 staníc, takáto manuálna práca by zabrala značné a zbytočné časové zdroje. Z tohto dôvodu som sa zaoberal automatizovaným dolovaním dát. V podkapitole 2.1 sa venujem technológii, ktorá na túto činnosť dolovania dát využíva webový prehliadač. V ďalšej podkapitole 2.2 sa venujem technológiám, ktoré na získanie dát využívajú iné možnosti, napríklad vytvorenie HTTP spojenia a posielanie GET, či POST požiadaviek.

Pomocou týchto technológií som navrhol jednotlivé programy, ktoré som napísal v jazyku Python a spoločne so štruktúrou výsledného zoznamu je ich možné nájsť v kapitole 3. V podkapitolách 3.1 a 3.2 zobrazujem jednotlivé varianty a fungovanie týchto programov.

Ďalšou časťou zadania je zmerať prenosové parametre siete, konkrétne dostupnosť na ping, dĺžku odozvy na SSH a prenosovú rýchlosť, ktoré bližšie popisujem v kapitole 4. V podkapitole 4.1 znázorňujem vytvorený skript na meranie odozvy serveru na SSH, teda čas, za ktorý je možné pripojiť a odpojiť sa zo serveru. Následne v podkapitole 4.2 rozoberám meranie odozvy na ping. Ďalšie meranie, ktoré vykonávam v sieti PlanetLab, je meranie prenosovej rýchlosti, ktoré popisuje podkapitola 4.3. Všetky programy v tejto kapitole sú naprogramované v jazyku Bash.

Z vytvoreného zoznamu staníc a získaných dát analyzujem dosiahnuté výsledky v kapitole 5. Podkapitola 5.1 sa zaoberá geografickou analýzou staníc v sieti, zatiaľ čo podkapitola 5.2 sa venuje výkonovej analýze siete. Na záver popisujem výsledky meraní vytvorených programov.

1 EXPERIMENTÁLNA SIET' PLANETLAB

Podľa [8] je PlanetLab globálna výskumná sieť, ktorá podporuje vývoj nových sieťových služieb. Od jej oficiálneho založenia v roku 2003 už mnoho tisíc výskumníkov z akademických inštitúcií a výskumných laboratórií využilo PlanetLab na vývoj nových technológií pre distribuované úložiská, mapovanie sietí, peer-to-peer systémy, distribuované hashovacie tabuľky, či spracovanie rôznych požiadaviek. Sieť PlanetLab sa ukázala ako neoceniteľná platforma na vzdelávanie sa v rámci široko oblastných sieťových javov, vytváraní nových sieťových protokolov, overovaní nových, či existujúcich sieťových služieb, na získavaní skúseností so sieťovými systémami bežiacimi v celosvetových merítkach, či pri zavádzaní nových služieb, ktoré vylepšujú schopnosti siete internet. PlanetLab sa stal podstatným nástrojom na výskum sieťových a distribuovaných systémov. Vývojári majú takto možnosť demonštrovať, ako dizajn ich protokolov, či služieb funguje v rámci siete internet v reálnych podmienkach [7].

Základná terminológia

Podľa [3] a [5] je nasledujúci zoznam základnou terminológiou použitou pri opise architektúry siete PlanetLab:

- Organizácia – *site* je fyzické miesto, kde sú umiestnené servery siete PlanetLab označované ako *nodes*.
- Uzol – *node* je server, na ktorom prebiehajú súčasti služieb siete PlanetLab. *Node* musí mať aspoň jednu nezdieľanú IP adresu.
- Projekt – *slice* je sada virtuálnych serverov distribuovaných cez sieť PlanetLab. Pre väčšinu užívateľov *slice* znamená meno účtu unixového terminálu s prístupom na množstvo PlanetLab uzlov (*node*).
- *Sliver* – je sada alokovaných zdrojov na jednom PlanetLab uzle.
- Virtuálny server (VServer) – *VServer* zabezpečuje oddelenie výkonu medzi sliverami na jednom fyzikom serveri.

História siete PlanetLab

V marci 2002 Larry Paterson z Princeton University a David Culler pracujúci pre Intel Research a University of California Berkeley zorganizovali stretnutie výskumných pracovníkov, ktorí sa zaujímali o globálne počítačové siete. Stretnutie sa konalo za podpory Intel Research v Berkley a zúčastnilo sa ho 30 výskumníkov z MIT, Washington University, Rice, Berkeley, Princeton, Columbia, Duke, CMU a z Utahu.

Na tomto stretnutí navrhli PlanetLab ako komunitné testovacie prostredie. David Tennenhouse z Intel Research súhlasil so začiatočným financovaním projektu so 100 servermi [8]. Sieť sa ale značne rozširovala a už v septembri 2003 dosiahla hranicu 200 uzlov. V decembri roku 2003 dokonca presiahol počet uzlov 300. V januári 2004 bolo formálne vytvorené PlanetLab konzorcium, teda skupina akademických, priemyselných a vládnych organizácií spolupracujúcich na podpore a raste siete PlanetLab. Odvtedy sa sieť PlanetLab masívne rozrástla, nielen medzi univerzitami, ale aj medzi výskumnými organizáciami ako Hewlett Packard, Google, AT&T, Lucent a ďalšími. Aktuálne (máj 2016) pozostáva sieť z 1353 uzlov (*nodes*) v 717 organizáciách (*sites*) [8].

Projekty v rámci siete PlanetLab

Jedným z cieľov siete PlanetLab je slúžiť ako testovacie prostredie, kde vývojárske skupiny môžu experimentovať s pestrú skupinou služieb, ako je zdieľanie dát, vrstvenie QoS, detekcia anomálií v sieti, a mnoho ďalších. Výhodou siete PlanetLab je experimentovanie v reálnych podmienkach. V rámci siete PlanetLab beží aktuálne približne 311 projektov [8].

Príklad aktuálnych projektov [8]:

- Adaptive Rate Allocation in Distributed Stream Processing Systems (DSPA) – test algoritmu na adaptívnu rýchlosť alokácie pri DSPA pod veľkou záťažou.
- Angelo Audio Experiments – experiment, ktorý odosiela audio nahrávky a analyzuje ako ich ovplyvnila stratovosť paketov a doba odozvy siete.
- The Citadel – testovanie overlay multicast protokolu.
- TopHat – merania zamerané na protokol UPCM.
- Web Crawler – testovanie výkonnosti webových stránok a použitia HTTPS či HTTP 2.0 protokolu na testovaných stránkach.

2 DOLOVANIE DÁT Z WEBOVÝCH STRÁNOK

Na webových stránkach siete PlanetLab [8] má každý pripojený uzol do tejto siete zverejnené informácie, ktoré nie je možné jednoduchým spôsobom z webu exportovať. Analyzoval som preto možnosti, ako tieto informácie z webu spracovať. Keďže je mi najbližší programovací jazyk Python, vyhľadal som knižnice pre tento jazyk, ktoré mi umožnia dolovať informácie z webových stránok. Pri analýze som našiel dve možnosti ako prístup k danému problému. Prvou možnosťou bol nástroj pracujúci priamo s webovým prehliadačom, druhou možnosťou boli nástroje pracujúce priamo s HTTP požiadavkami. Tieto možnosti predstavím v nasledujúcich podkapitolách. Ďalej opíšem aj pomocnú knižnicu Geocode na spracovanie údajov.

2.1 Nástroje pracujúce s webovým prehliadačom

Selenium

SeleniumHQ je sada open source nástrojov slúžiaca pre automatizáciu testovania cez webový prehliadač. Ponúka viacero prístupov zameraných na testovanie webových aplikácií [4]. Projekt Selenium WebDriver vznikol spojením projektu Selenium a projektu WebDriver, ktorý bol pod vedením spoločnosti Google. Projekt ponúka knižnice a podporu pre rôzne programovacie jazyky (Java, C#, Python, Ruby, PHP, Perl a JavaScript). Veľkou výhodou je podpora najrozšírenejších internetových prehliadačov (Mozilla Firefox, Google Chrome, Opera a Internet Explorer).

Po spustení programu zavedie WebDriver ovládač, pomocou ktorého kompletne ovláda prehliadač. Po zavedení sa vytvorí užívateľský profil a spustí sa grafické okno. Pomocou WebDriver-u môžeme načítavať URL adresy, vyhľadávať elementy na stránke, zisťovať hodnoty elementov, vyplňovať text do polí na stránke, či spúšťať JavaScript príkazy. Silnou stránkou Selenia je tiež možnosť čakať, kým sa na stránke objaví hľadaný element. Počas vykonávania testu nemusí byť okno prehliadača, kde celý testovací cyklus beží, v popredí, ale môže bežať na pozadí. Výpis kódu 2.1 zobrazuje spustenie prehliadača Mozilla Firefox, načítanie stránky `https://www.vutbr.cz/` a následnú kontrolu, či sa v hlavičke webovej stránky nachádza titulok *Vysoké učení technické v Brně*. V prípade, že sa titulok vyhodnotí správne prehliadač sa bez chýb ukončí.

Výpis 2.1: Ukážka využitia Selenia na kontrolu titulku

```
from selenium import WebDriver
browser = WebDriver.Firefox()
browser.get('https://www.vutbr.cz/')
assert 'Vysoké učení technické v~Brně' in browser.title
browser.quit()
```

2.2 Dostupné nástroje pracujúce z príkazového riadku

Nevýhodou použitia Selenia je nutnosť použiť webový prehliadač. Najväčšia doba odozvy nastáva pri načítaní grafického užívateľského rozhrania (GUI) a pri spúšťaní skriptov. Existujú ale možnosti ako sa GUI môžeme vyhnúť.

Requests

Requests je jednoduchá HTTP knižnica pre Python [13], pomocou ktorej môžeme priamo otvoriť HTTP aj HTTPS spojenie a posilať GET, či POST správy [15]. Knižnica je výborne dokumentovaná. Jednoduchým spôsobom sa môžeme prihlásiť na web, stiahnuť dáta, odosielať viacero súborov naraz a mnoho ďalšieho. Práca s knižnicou je zobrazená vo výpise kódu 2.2. Importom použijeme knižnicu, následne odošleme GET požiadavku s užívateľskými údajmi a dostaneme obsah webovej stránky.

Výpis 2.2: Ukážka využitia knižnice Requests

```
>>> import requests
>>> r = requests.get('https://api.github.com/user', auth=('
    user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.text
u'{"type":"User"... '
>>> r.json()
{u'private_gists': 419, u'total_private_repos': 77, ...}
```

Urllib2

Knižnica Urllib2 definuje funkcie a triedy [12], ktoré pomáhajú s otváraním URL adries, najmä HTTP, či HTTPS spojení. Pracuje veľmi dobre s presmerovaním, cookies, či autentifikáciou. Knižnica je veľmi obsiahla, ale aj napriek tomu si zachováva konzistenciu a prehľadnosť. Ukážka funkčnosti je vyobrazená na výpise kódu 2.3.

Výpis 2.3: Ukážka využitia knižnice Urllib2

```
>>> import urllib2
>>> response = urllib2.urlopen('https://vutbr.cz:443/')
>>> print 'RESPONSE:', response
RESPONSE: <addinfourl at 50938864 whose fp = <socket.
    _fileobject object at 0x03091830>>
>>> print 'URL      :', response.geturl()
URL      : https://www.vutbr.cz/
>>> headers = response.info()
>>> print 'DATE      :', headers['date']
DATE     : Tue, 10 May 2016 20:21:07 GMT
>>> print headers
Date: Tue, 10 May 2016 20:21:07 GMT
Server: Apache/2.2.15 (Scientific Linux)
Set-Cookie: PHPSESSID=p2gts3g145d6ui3ja3ahksh38em3tnlk; path
    =/
Last-Modified: Tue, 10 May 2016 20:21:08 GMT
Content-Type: text/html; charset=utf-8
...
>>> data = response.read()
>>> print 'LENGTH  :', len(data)
LENGTH   : 32682
>>> print data
<!DOCTYPE HTML><html><head><meta charset="UTF-8" /><title>
    Vysoké učení technické v~Brně</title>...
>>> print response.read(150)
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta http-equiv="Content-Type" content
```


LXML

Autori knižnice LXML o tejto knižnici tvrdia, že obsahuje najväčšie množstvo funkcií na spracovanie XML a HTML pre Python [16]. Je odvodená z knižníc Libxml2 a Libxslt, ktoré sú napísané v programovacom jazyku C. Najbežnejšie sa využíva na parsovanie a serializáciu dát z XML, či HTML. Ukážku využitia LXML uvádzam vo výpise kódu 2.4.

Výpis 2.4: Ukážka využitia knižnice LXML

```
>>> root = etree.XML('<html><head/><body><p>Hello<br/>World</p></body></html>')
>>> etree.tostring(root) # default: method = 'xml'
b'<html><head/><body><p>Hello<br/>World</p></body></html>'
>>> etree.tostring(root, method='xml') # same as above
b'<html><head/><body><p>Hello<br/>World</p></body></html>'
>>> etree.tostring(root, method='html')
b'<html><head></head><body><p>Hello<br>World</p></body></html>'
>>> print(etree.tostring(root, method='html', pretty_print=True))
<html>
<head></head>
<body><p>Hello<br>World</p></body>
</html>
>>> etree.tostring(root, method='text')
b'HelloWorld'
```

BeautifulSoup4

Knižnica BeautifulSoup4 [14], slúži ako nadstavba pri parsovaní zdrojového kódu stránky na webe, či XML súborov. Uľahčuje vyhľadávanie elementov, vyťahovanie správnych tabuliek, paragrafov uchovávajúcich informácie, ktoré chceme zo stránky použiť. Ukážku môžete vidieť na výpise 2.5.

Výpis 2.5: Ukážka využitia knižnice BeautifulSoup4

```
>>> html_doc = """
<html><head><title>The Dormouse's story</title></head>
<body><p class="title"><b>The Dormouse's story</b></p>
<p class="story">Once upon a~time there was a~princess
<a href="http://e.cz/lada" class="princess" id="link">Lada</a
  >
and she lived in castle.</p>
<p class="story">...</p>
"""
>>> from bs4 import BeautifulSoup
soup = BeautifulSoup(html_doc, 'html.parser')
>>> print(soup.prettify())
<html>
  <head>
    <title>
      The Dormouse's story
    </title>
  </head>
  <body>
    <p class="title">
      <b>
        The Dormouse's story
      </b>
    </p>
    <p class="story">
      Once upon a~time there was princess
      <a class="princess" href="http://e.cz/lada" id="link">
        Lada
      </a>
      and she lived in castle.
    </p>
    <p class="story">
      ...
    </p>
  </body>
</html>
```

Geocoder

Knižnica Geocoder slúži na uľahčenie práce s poskytovateľmi služby Geocodingu [2]. Jedná sa napríklad o poskytovateľov ArcGIS, Bing, Google, MapQuest, MaxMind, Tamu a mnoho ďalších. Služby, ktoré táto knižnica podporuje sú Geocoding, Reverse Geocoding, Timezone, Elevation a ďalšie.

Výpis 2.6: Ukážka využitia knižnice Geocoder

```
>>> import geocoder
>>> geocoding = geocoder.google('Technicka 12, Brno, Ceska
    Republika')
>>> geocoding.json
{'encoding': 'utf-8', 'status_code': 200, 'county': u'Brno-
    City District', 'lng': 16.5746466, 'city': u'Brno', '
    location': 'Technicka 12, Brno, Ceska Republika', 'lat':
    49.2269971}
>>> reverse_geocoding = geocoder.google([49.15, 13.14],
    method='reverse')
>>> print reverse_geocoding
<[OK] Google - Reverse [Unnamed Road, 93470 Lohberg, Germany
    ]>
>>> elevation = geocoder.google([49.166436, 20.184462],
    method='elevation')
>>> elevation.elevation
2390.83740234375
>>> elevation.meters
2390.8
```

3 PROGRAM VYTVORENÝ NA DOLOVANIE DÁT Z WEBOVÝCH STRÁNOK

Na webových stránkach siete PlanetLab [8] má každý pripojený uzol do tejto siete zverejnený názov inštitúcie, ktorá ho spravuje, URL adresu danej inštitúcie a GPS súradnice, kde je fyzicky umiestnená, tak ako ilustruje obrázok 3.1. Keďže nie je možné zo stránok zoznam staníc s týmito informáciami vyexportovať, bolo treba vymyslieť spôsob, ako tento zoznam jednoducho vytvoriť. Jednou z možností bolo vytvoriť ho ručne, čo by však znamenalo preklikať sa cez približne 1000 staníc a dvojnásobné množstvo odkazov. Toto riešenie by bolo časovo neefektívne, a navyše by mohlo jednoducho dôjsť ku chybám pri prepise údajov do textového súboru. Ďalšou možnosťou ako vytvoriť tento zoznam bolo napísať program, ktorý prehľadá webové stránky za nás a požadované údaje uloží do výstupného súboru.

S ohľadom na praktické využitie aktuálneho zoznamu staníc v budúcnosti, ktoré ponúka druhá spomínaná možnosť, sa táto práca zaoberá práve touto možnosťou. Analyzoval som preto spôsoby vytvorenia takéhoto programu. Po preskúmaní dostupných nástrojov som navrhol dva scenáre, ako pristúpiť k danej problematike. Prvou možnosťou bolo použitie knižnice Selenium, pomocou ktorej je možné ovládať webový prehliadač, a takto postupne získať reálne načítané dáta. Druhou možnosťou bolo využitie knižnice Requests, a pomocou nej posielat GET požiadavky priamo na server. Odpovede na GET požiadavky obsahujú HTML kód, ktorý je možné spracovať pomocou knižnice BeautifulSoup4. Na základe oboch scenárov som vytvoril dva samostatné programy pracujúce na týchto princípoch. Podrobnejšie opisujem tieto programy v ďalších podkapitolách.

3.1 Varianta cez webový prehliadač

Program `planetlab_crawler_selenium.py` som napísal v programovacom jazyku Python pomocou nástroja Selenium WebDriver. Program sa spúšťa pomocou príkazu 3.1 s argumentami `username` a `password`, do ktorých je nutné vložiť užívateľské meno a heslo, ktoré užívateľ používa na prihlásenie sa na webovú stránku `https://www.planet-lab.org/`.

Výpis 3.1: Spustenie programu na dolovanie dát

```
python planetlab_crawler_selenium.py username password
```

Details	
Full name	Czech Educational and Scientific Network
Abbreviated name	CESNET
URL	http://www.ces.net
Latitude	50.102
Longitude	14.3916
Login base	cesnetple
Max slices	10
Enabled	1
Peer	PlanetLabEurope (PLE)

Obr. 3.1: Informácie o organizácii na webe siete PlanetLab

Po spustení programu sa naštartuje webový prehliadač s čistým užívateľským profilom. Do prehliadača sa zavedie WebDriver, čím program získa kontrolu nad prehliadačom. Program zavelí, aby sa načítala webová stránka `https://www.planet-lab.org/`. Keď je stránka správne načítaná, vyhledá sa na nej prihlasovací formulár, viz obrázok 3.2. Po preštudovaní HTML kódu je jasné, že môžeme vyhledať element `edit-name`, do ktorého sa vloží užívateľské meno a element `edit-pass`, do ktorého sa vloží heslo. Vyhledať element je možné cez jeho Id, Name, XPath, CSS selector a ďalšími spôsobmi. Na webe PlanetLabu bolo jednoduché použiť Id, ako zobrazujem na ukážke 3.2, pretože sa na stránke nevyskytlo žiadne Id viackrát. Po vložení prihlasovacích údajov klikne WebDriver na tlačítko `Log in`, ktoré je vyhledané pomocou XPath.

Výpis 3.2: Vyhľadanie elementov na stránke pomocou WebDriveru

```
driver.find_element_by_id("edit-name").send_keys(username)
driver.find_element_by_id("edit-pass").send_keys(password)
driver.find_element_by_xpath("//form[@id='planetlab-login-form']
    /div/input").click()
```

Keď sme úspešne prihlásení, môže program prehľadať web tak, ako by ho prehľadával človek. Ukážka vyhľadania žiadaných informácií z elementov a parsovanie

```

<h2 class="title">PlanetLab login</h2>
<div class="content">
  <form id="planetlab-login-form"
    action="https://www.planet-lab.org
    /node?destination=node" method="post">
    <div>
      <div class="form-item">
        <label for="edit-name"></label>
        <input id="edit-name" class="form-text required"
          maxlength="60" name="edit[name]" size="25"
          value="" type="text"></input>
      </div>
      <div class="form-item">
        <label for="edit-pass"></label>
        <input id="edit-pass" class="form-text required"
          name="edit[pass]" size="25" type="password"
          ></input>
      </div>
      <input class="form-submit" name="op" value="Log in"
        type="submit"></input>
      <input id="edit-planetlab-login-block"
        name="edit[form_id]" value="planetlab_login_block"
        type="hidden"></input>
    </div>
  </form>
</div>

```

The rendered form on the right shows the following structure:

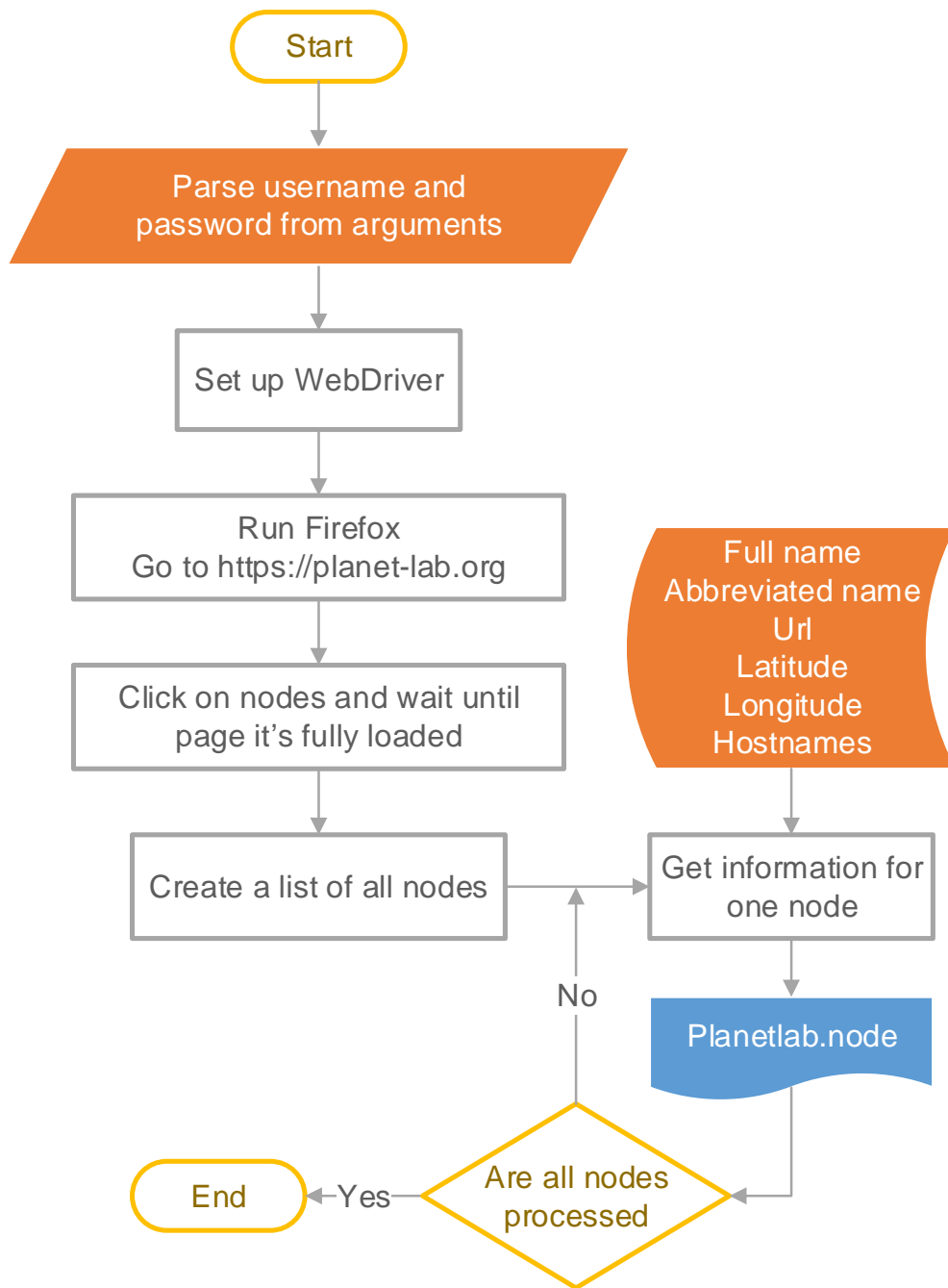
- PlanetLab login** (Title)
- E-mail: *** (Label) with a text input field.
- Password: *** (Label) with a password input field.
- Log in** (Submit button)
- [Forgot your password?](#) (Link)
- [Create an account](#) (Link)

Obr. 3.2: Prihlasovacie okno do webu <https://www.planet-lab.org/>

informácií z webovej stránky, je zobrazená na výpise kódu 3.3. Postupne sa program dostane k pre nás potrebným informáciám o všetkých staniach siete PlanetLab v Európe a tie uloží do výstupného súboru `planetlab.node`. Vývojový diagram programu je zobrazený na obrázku 3.3. Ukážkový výstup programu je zachytený na výpise kódu 3.4. Po úspešnom behu programu máme vytvorený zoznam staníc, ktoré sú súčasťou PLE (PlanetLab Europe).

Výpis 3.3: Parsovanie webovej stránky

```
toggle_area_nodes = driver.find_elements_by_xpath("//div[@id='toggle-area-node']/table/tbody/tr[16]/td/table/tbody/tr[*]/td/a")
list_of_nodes = [i.text.strip() for i in toggle_area_nodes]
driver.find_element_by_xpath("//div[@id='toggle-area-node']/table/tbody/tr[15]/td/a").click()
full_name = driver.find_element_by_xpath("//div[@id='toggle-area-site']/form/table/tbody/tr/td").text
abbreviated_name = driver.find_element_by_xpath("//div[@id='toggle-area-site']/form/table/tbody/tr[2]/td").text
url = driver.find_element_by_xpath("//div[@id='toggle-area-site']/form/table/tbody/tr[3]/td").text
latitude = driver.find_element_by_xpath("//div[@id='toggle-area-site']/form/table/tbody/tr[4]/td").text
longitude = driver.find_element_by_xpath("//div[@id='toggle-area-site']/form/table/tbody/tr[5]/td").text
```



Obr. 3.3: Vývojový diagram dolovania informácií z webu cez webový prehliadač

Výpis 3.4: Časť výstupného súboru planetlab.node

```
1 147-179.surfsnel.dsl.internl.net 52.22 4.53 http://www.cwi.nl
   CWI
2 aguila1.lsi.upc.edu 41.3897 2.11273 http://www.upc.edu
   Universitat Politecnica de Catalunya
3 aguila2.lsi.upc.edu 41.3897 2.11273 http://www.upc.edu
   Universitat Politecnica de Catalunya
4 ait05.us.es 37.42 -5.9 http://www.us.es University of Sevilla
5 ait21.us.es 37.42 -5.9 http://www.us.es University of Sevilla
6 aladdin.planetlab.extranet.uni-passau.de 48.58 13.47 http://
   www.fmi.uni-passau.de/lehrstuehle/demeer/ University of
   Passau
7 anateus.ipv6.lip6.fr 48.8473 2.35749 http://www.lip6.fr
   Universite Pierre et Marie Curie
8 dannan.disy.inf.uni-konstanz.de 47.6894 9.18783 http://www.
   uni-konstanz.de University of Konstanz
9 dev-res-plab-01.tid.es 41.414 2.22087 www.tid.es Telefonica I
   +D, Barcelona
10 dev-res-plab-02.tid.es 41.414 2.22087 www.tid.es Telefonica I
   +D, Barcelona
```

3.2 Varianta bez prehliadača

Program `planetlab_list_creator.py` som vytvoril ako implementáciu riešenia úlohy vytvorenia zoznamu staníc siete PlanetLab bez nutnosti využitia webového prehliadača. Na jeho spracovanie som zvolil programovací jazyk Python. V tomto programe som použil knižnice `Requests`, `BeautifulSoup4` a `Geocoder`, ktoré som popísal v kapitole 2.2.

Program môžeme spustiť s viacerými vstupnými parametrami, ako je možné vidieť na tomto príklade spustenia programu 3.5.

Výpis 3.5: Spustenie programu na dolovanie dát bez prehliadača

```
python planetlab_list_creator.py -u uzivatel@domena.com
-p heslo -f . -s 50000
```

Všetky použiteľné vstupné parametre sú zobrazené v tabuľke 3.1. Vyžadovanými

Tab. 3.1: Vstupné parametre programu `planetlab_list_creator.py`

Použité argumenty	Význam argumentov
<code>-u username</code>	užívateľské meno
<code>-p password</code>	užívateľské heslo
<code>-e eu</code>	zmena serveru na <code>planet-lab.eu</code>
<code>-f file_path</code>	cesta kam chceme uložiť zoznam staníc
<code>-s id_start</code>	číslovanie staníc začne na danom čísle
<code>-t timeout</code>	ako dlho má program čakať, keď sa vyskytne chyba
<code>-r repetition</code>	počet opakovaní požiadavkov v prípade chyby

parametrami sú „-u“ užívateľské meno a „-p“ užívateľské heslo, keďže bez týchto údajov sa nie je možné autentizovať na webovej stránke <https://www.planet-lab.org> a bez autentifikácie sa nie je možné dostať k údajom, ktoré zisťujeme. Po spustení programu už nie je od užívateľa vyžadovaná žiadna interakcia. Po autentifikácii sa stiahnu všetky DNS názvy staníc a postupne sa k nim parsujú dáta. Následne sa z webu stiahnu hodnoty IP adresa, DNS, URL, Full name, Latitude a Longitude. Id prvej stanice je vygenerované podľa užívateľských preferencií pri použití argumentu „-s“. Bez použitia tohto argumentu sa použije Id 030001. Následne program zo získaných súradníc pomocou knižnice Geocoder, ktorú som opísal v kapitole 2.2, získa adresu, a teda údaje do kategórií Country, Region a City. Kvôli štatistickým údajom som potreboval určiť, na akých kontinentoch sú stanice umiestnené. Tento údaj mi ale reverzná geolokácia neposkytovala, a tak som napísal funkciu `get_continent`. Tá porovná medzinárodný kód krajiny [1], ktorý som získal z geolokácie, so slovníkom 3.6, a tak zistí, na ktorom kontinente sa krajina nachádza. Funkcia je zobrazená na výpise kódu 3.7. IP adresa je na webe uložená spravidla na stanicach, ktoré sú súčasťou PlanetLab Central (PLC). V prípade, ak program získa IP adresu z webu, tak ju porovná s adresou, ktorú získa z prekladu doménového mena. Podľa toho ako túto situáciu vyhodnotí, určí hodnotu stĺpca IP Flag vo výstupnom súbore. Ak sa IP adresa z webu líši od IP adresy z prekladu, IP flag sa nastaví na „diffIPfromDns“. Keď IP adresa nie je dostupná na webe, uloží sa do IP Flag hodnota „unknown“. V prípade, že nebolo možné preložiť doménové meno na IP adresu, uloží sa do IP Flag hodnota „noIPfromDns“. Stĺpce vo výstupnom súbore sú popísané v tabuľke 3.2. Ukážka výstupu programu je zobrazená na výpise kódu 3.8. Pre lepšiu predstavu ako program pracuje je na obrázku 3.4 zobrazený vývojový diagram.

Výpis 3.6: Ukážka slovníku použitého na určenie kontinentu.

```
continents = {
    'AF': {'name': 'Africa', 'countries': ['DZ', 'AO', 'BJ',
        'BW', 'BF', 'BI', 'CM', 'CV', 'CF', 'TD', '...']},
    'NA': {'name': 'North America', 'countries': ['AI', 'AG',
        'AW', 'BS', 'BB', 'BZ', 'BM', 'VG', 'CA', '...']},
    'OC': {'name': 'Oceania', 'countries': ['AS', 'AU', 'CK',
        'FJ', 'PF', 'GU', 'KI', 'MH', 'FM', 'NR', '...']},
    'AN': {'name': 'Antarctica', 'countries': ['AQ', 'BV', '
        TF', 'HM', 'GS']},
    'AS': {'name': 'Asia', 'countries': ['AF', 'AM', 'AZ', '
        BH', 'BD', 'BT', 'IO', 'BN', 'KH', 'CN', '...']},
    'EU': {'name': 'Europe', 'countries': ['AX', 'AL', 'AD',
        'AT', 'BY', 'BE', 'BA', 'BG', 'HR', 'CZ', '...']},
    'SA': {'name': 'South America', 'countries': ['AR', 'BO',
        'BR', 'CL', 'CO', 'EC', 'FK', 'GF', 'GY', '...']}}
```

Výpis 3.7: Funkcia určujúca, na ktorom kontinente sa nachádza žiadaná krajina

```
def get_continent(country):
    """
    Determine on which continent is country located.
    :param country: Short name od country
    :return: continent where county is located
    """
    for continent, val in continents.iteritems():
        for names, lists in val.iteritems():
            if names == 'countries':
                if country in lists:
                    return continent
```

Tab. 3.2: Výstup programu planetlab_list_creator.py

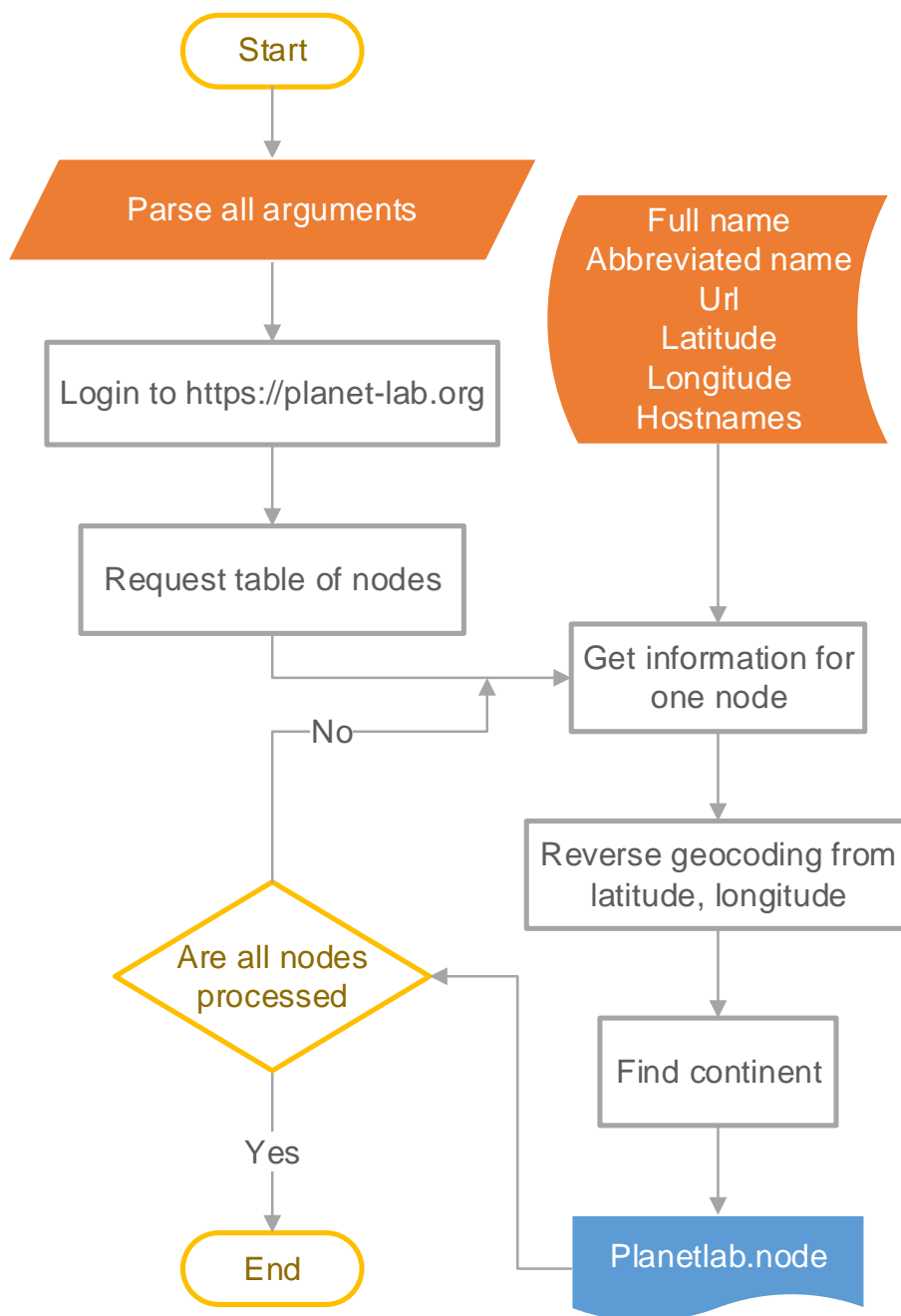
Názov stĺpca	Popis stĺpca
ID	ID stanice
IP	IP adresa
DNS	DNS názov stanice
Continent	kontinent, na ktorom sa stanica nachádza
Country	krajina, kde je stanica umiestnená
Region	región, kde je stanica umiestnená
City	mesto
URL	URL adresa organizácie
Full_name	názov inštitúcie
Latitude	zemepisná šírka
Longitude	zemepisná dĺžka
IP Flag	označenie, či je IP adresa z DNS rovnaká ako na webe

Výpis 3.8: Výstup programu planetlab_list_creator.py

```

1 # ID IP DNS continent country region city url full_name
   latitude longitude IP Flag
2 050000 129.240.228.138 planetlab2.simula.no EU NO Oslo Oslo
   http://www.simula.no Simula Research Laboratory 59.93
   10.75 diffIPfromDns
3 050001 203.230.7.202 planetlab2.kreonet.net AS KR unknown
   Nonsan http://www.kreonet2.re.kr KREONET at KISTI-DAEJON
   36.2 127.26 identicalIP
4 050002 168.188.48.26 datacomngn.cnu.ac.kr AS KR unknown
   Eumseong http://www.cnu.ac.kr Chungnam National University
   37 127.5 noIPfromDns
5 050003 133.1.172.60 int-pl2.ise.eng.osaka-u.ac.jp AS JP
   unknown Suita-shi http://www.ist.osaka-u.ac.jp Osaka
   University 34.81 135.52 noIPfromDns
6 050004 133.1.172.58 int-pl1.ise.eng.osaka-u.ac.jp AS JP
   unknown Suita-shi http://www.ist.osaka-u.ac.jp Osaka
   University 34.81 135.52 noIPfromDns
7 050005 202.180.37.19 planetlab-02.kyushu.jgn2.jp unknown
   unknown unknown unknown NICT JGN2 Fukuoka unknown
   unknown noIPfromDns
8 050006 152.8.246.199 csplanet2.ncat.edu NA US Guilford County
   Greensboro http://www.eng.ncat.edu North Carolina AT
   State University 36.07 -79.77 diffIPfromDns

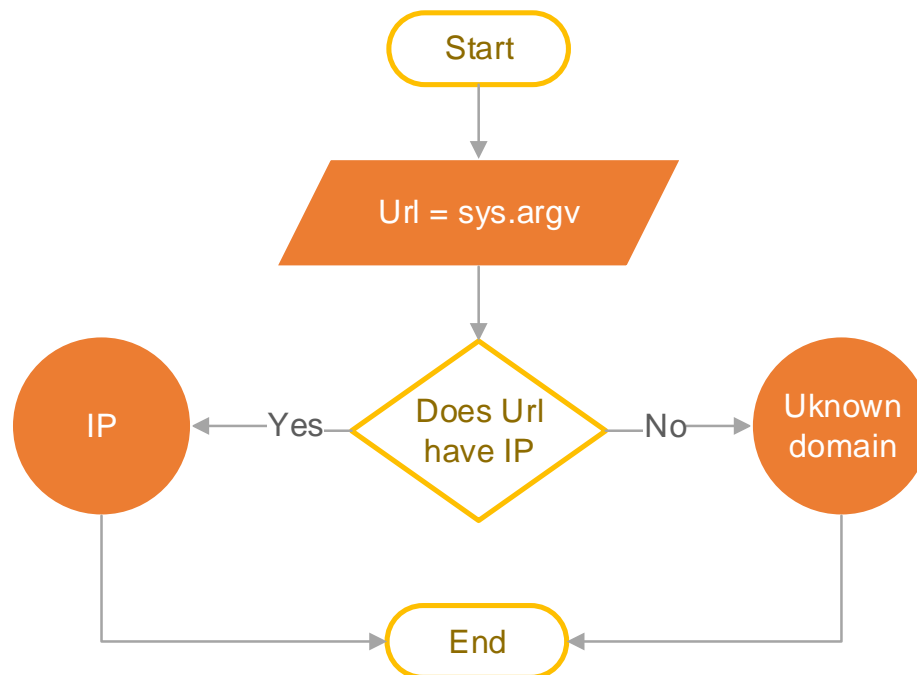
```



Obr. 3.4: Vývojový diagram dolovania informácií z webu pomocou knižnice Requests

4 MERANIE PARAMETROV SIETE PLANET-LAB

Meranie parametrov siete prebiehalo z 50 staníc siete PlanetLab, ale kvôli vymazaniu Slice sa mi podarilo dostať výsledky len z 29 staníc. Stanice som vybral na základe predchádzajúcich meraní, kde som zistil, ktoré stanice sú dostupné online a nedochádza u nich k vážnejším výpadkom spojenia. Tabuľka 4.1 zobrazuje najstabilnejšie stanice z testovacích meraní dostupnosti. Dostupnosť uzlov bola overovaná pomocou programu ping a merania času nutného na pripojenie k serveru pomocou služby SSH. Na meranie prenosovej rýchlosti som využil program Rsync [17]. Tieto programy som implementoval do skriptov, ktoré som napísal v jazyku Bash [11]. Uzly siete sú identifikované doménovým menom, ale užívatelia pripojení k uzlu nemajú v základnej konfigurácii PlanetLabu žiaden nástroj na overenie doménového mena (napr. nslookup, dig, či host). Z tohto dôvodu som vytvoril skript `getHostByName.py` v jazyku Python, ktorý overí správnosť prekladu DNS mena na IP adresu. Vývojový diagram tohto programu je zobrazený na obrázku 4.1.



Obr. 4.1: Vývojový diagram prekladu mena na IP adresu

Na prípravu meracieho systému som na svojom počítači vytvoril súborovú štruktúru obsahujúcu zložky `measurments`, `results`, `logs` a `upload_file`. V zložke `measur-`

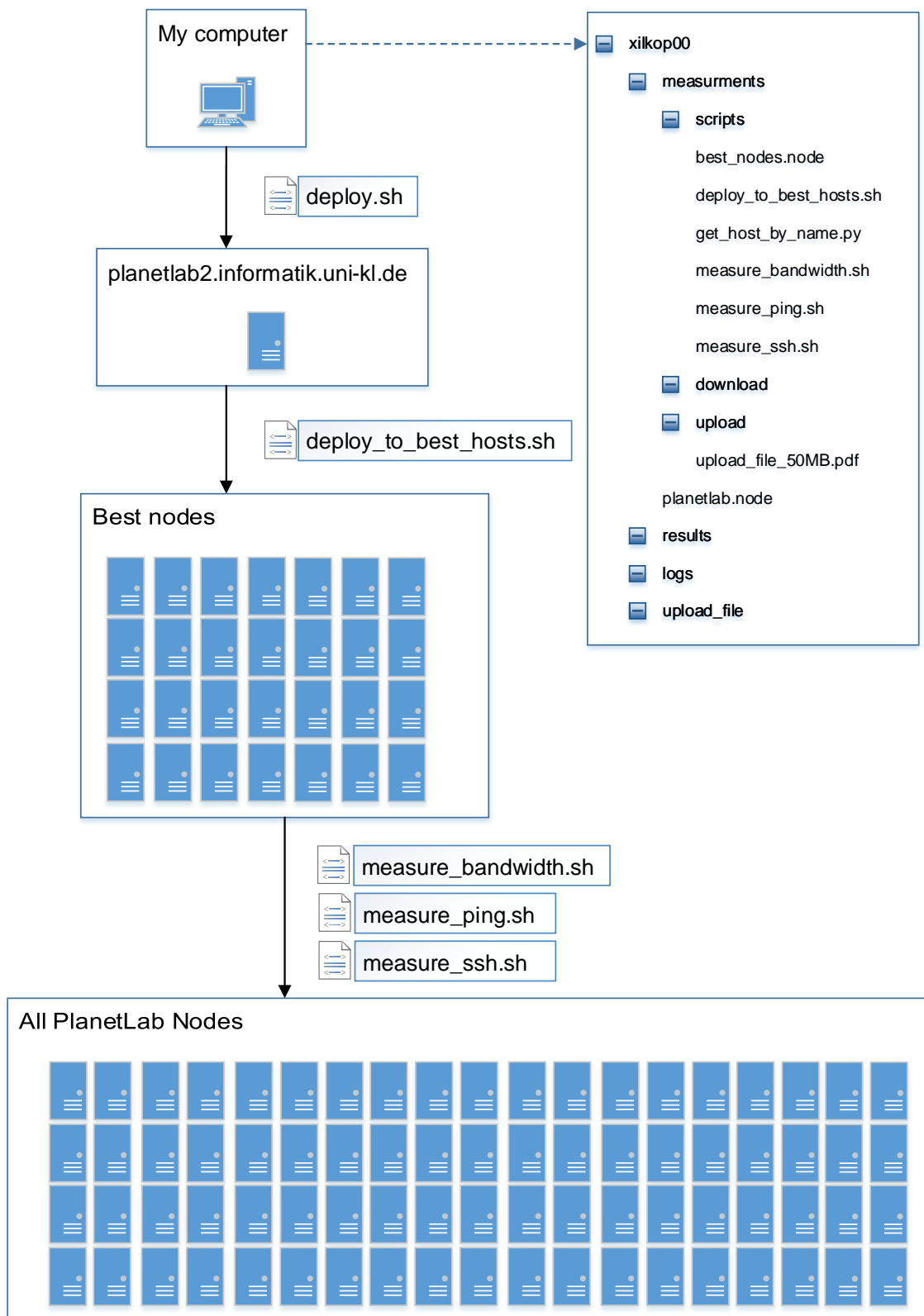
ments sa nachádzajú konkrétne skripty vykonávajúce meranie a pomocné zložky, či súbory. Zvyšné zložky sú prázdne, budú do nich ukladané namerané údaje, chybové výpisy, či odoslané súbory z iných serverov. Celú zložku `xilkop00` pomocou skriptu `deploy.sh` odošlem na server, ktorý dlhodobo dával najlepšie výsledky, teda `planetlab2.informatik.uni-kl.de`. Pomocou skriptu `deploy_to_best_hosts.sh` z tohto serveru odošlem stromovú štruktúru na stanice, z ktorých bude prebiehať meranie. Celá štruktúra s priebehom šírenia je zobrazená na obrázku 4.2.

Tab. 4.1: Čiastočný zoznam staníc, z ktorých bolo vykonávané meranie

Názov stanice	Krajina	Zemepisná šírka	Zemepisná dĺžka
planetlab3.hiit.fi	FI	60.1	25
onelab1.pl.sophia.inria.fr	FR	43.6156	7.06874
lim-planetlab-2.univ-reunion.fr	RE	-20.905	55.5
ple1.det.uvigo.es	ES	42.1692	-8.68458
planetlab1.cs.uoi.gr	GR	39.6182	20.8386
plab3.ple.silweb.pl	PL	50.2901	18.6773
planetlab1.net.in.tum.de	DE	48.1493	11.5669
planetlab1.cs.uit.no	NO	69.6813	18.977
ple1.hpca.ual.es	ES	36.83	-2.40454
iraplab1.iralab.uni-karlsruhe.de	DE	49.015	8.405
pl1.tailab.eu	FR	48.9408	2.30674
planetlab1.u-strasbg.fr	FR	48.5237	7.73833
icnalplabs1.epfl.ch	CH	46.5203	6.5656
planetlab2.ci.pwr.wroc.pl	PL	51.1	16.93
planetlab1.tlm.unavarra.es	ES	42.7993	-1.63544
onelab3.info.ucl.ac.be	BE	50.6833	4.61667
planetlab2.di.fct.unl.pt	PT	38.672	-9.18423
planetlab-coffee.ait.ie	IE	53.4187	-7.90557

4.1 Odozva na SSH

SSH, takzvaný Secure Shell je server–klient aplikácia, pomocou ktorej sa môžeme pripájať na vzdialené servery [10]. Kladnou stránkou SSH je vytváranie šifrovaného spojenia. Na meranie času nutného na pripojenie k serveru som napísal skript `ssh_latency.sh`, ktorého hlavná časť programového kódu je zobrazená vo výpise kódu 4.1. Program na vstupe načíta zoznam staníc `planetlab.node` a vyčíta z neho DNS názov uzlu, ktorého dostupnosť bude merať. Pomocou skriptu `getHostByName.py`



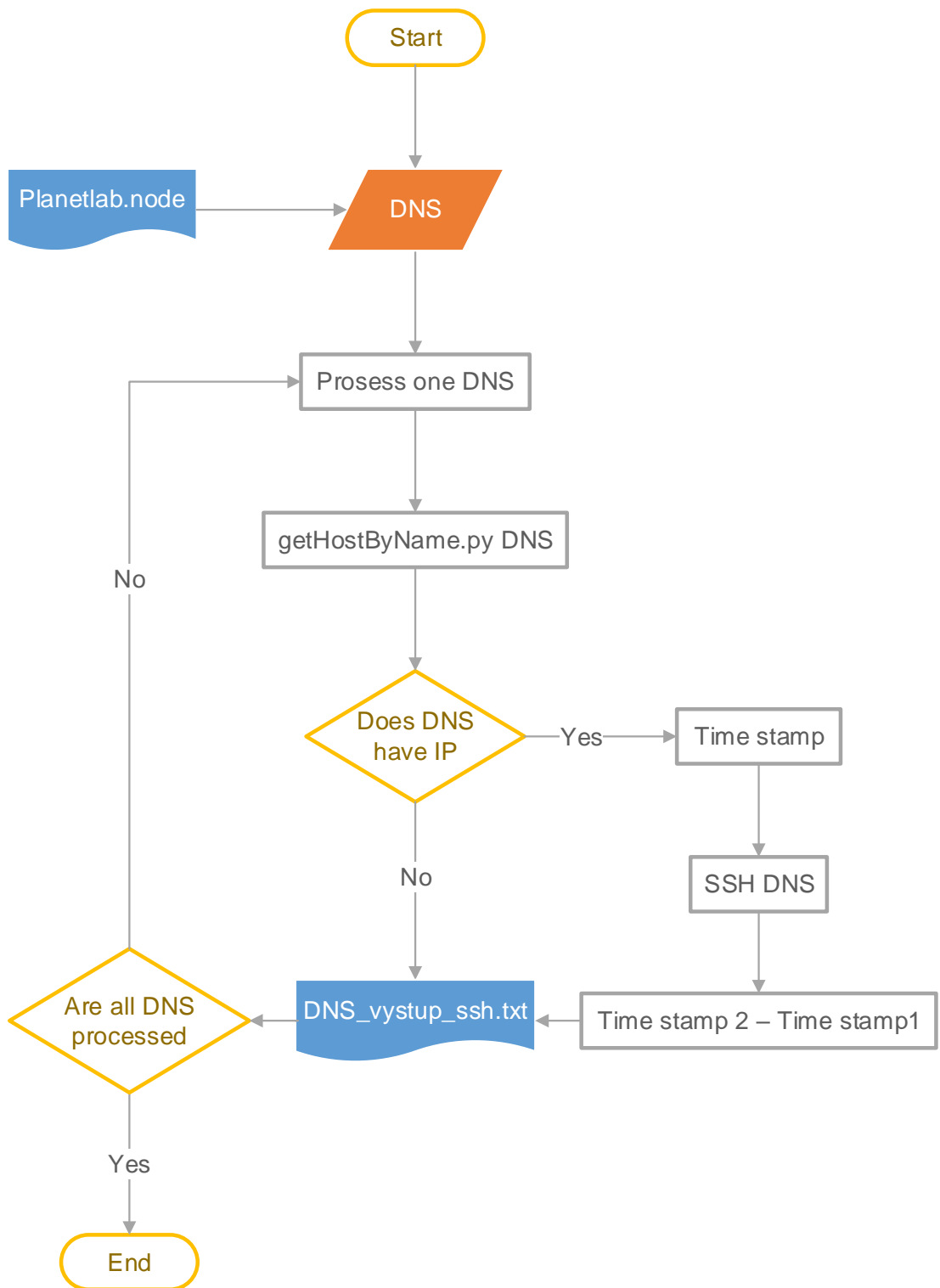
Obr. 4.2: Příprava na meranie prenosových parametrov

sa overí, či pre dané doménové meno existuje IP adresa. Keď nie, zapíše sa do výstupného súboru "\$hostname"_vystup_ssh.txt DNS názov uzlu spolu s textom „UnknownDomainName“. "\$hostname" v názve súboru je nahradené menom stanice, na ktorej prebieha meranie. Keď pre dané DNS meno existuje IP adresa, uloží sa časová značka do premennej T a zavolá sa SSH s argumentami, ktoré určujú nastavenie spojenia. Význam argumentov je uvedený v tabuľke 4.2. Keď je spojenie nadviazané, na vzdialenom serveri sa spustí príkaz *exit* a spojenie sa ukončí. Po ukončení spojenia sa uloží nová časová značka a vypočíta sa rozdiel od časovej značky T. Tento rozdiel v milisekundách je uložený do výstupného súboru spolu s DNS názvom. V prípade, že ku spojeniu na server nedôjde do 30 sekúnd, do výstupného súboru sa k DNS názvu vypíše „Timeout“. Chybové výpisy pri SSH spojení sú presmerované do výstupného súboru "\$hostname"_ssh_err_debug, aby bolo možné ladiť prípadné chybové stavy. Vývojový diagram je zobrazený na obrázku 4.3. Meranie prebiehalo každých 6 hodín automaticky pomocou plánovača úloh Cron. Ukážkový výstup z troch meraní je zobrazený vo výpise kódu 4.2.

Výpis 4.1: Meranie času nutného na pripojenie k SSH serveru

```
# time stamp -> ssh connect
T="$(date +%s%N)"
ssh      -vvv
        -o BatchMode=yes
        -o ConnectTimeout=30
        -o StrictHostKeyChecking=no
        -o UserKnownHostsFile=/dev/null
        -l cesnet_feec
        -i /home/cesnet_feec/.ssh/xilkop00_planetlab $foo
        exit 2>> "$logs"'ssh/'"$foo"_ssh_err_debug ||
        timeout=1

# Nanoseconds
T="$(($(date +%s%N)-T))"
# Milliseconds
M="$((T/1000000))"
```



Obr. 4.3: Vývojový diagram merania doby odozvy na SSH

Tab. 4.2: Použité argumenty pri pripájaní na server pomocou SSH

Použité argumenty SSH	Význam argumentov
[-v verbose]	úroveň ladiacich výpisov
[-i identity_file]	cesta k súkromnému kľúču
[-l login_name]	prihlasovacie meno
[-o option]	doplňkové nastavenia
BatchMode=yes	automatické overenie verejným kľúčom
ConnectTimeout=20	dostupnosť serveru sa testuje 20 sekúnd
StrictHostKeyChecking=no	automatické prijatie hostiteľského kľúča
UserKnownHostsFile=/dev/null	prijatý kľúč bude uložený do /dev/null

Výpis 4.2: Časť výstupného súboru "\$hostname"_vystup_ssh.txt

```

1  itchy.comlab.bth.se UnknownDomainName UnknownDomainName
   UnknownDomainName
2  lim-planetlab-1.univ-reunion.fr Timeout Timeout Timeout
3  lim-planetlab-2.univ-reunion.fr 3858 2921 3463
4  marie.iet.unipi.it Timeout Timeout Timeout Timeout
5  mars.planetlab.haw-hamburg.de 555 1443 635
6  medea.inf.uth.gr Timeout Timeout Timeout
7  merkur.planetlab.haw-hamburg.de 698 610 581
8  mimas.ipv6.lip6.fr Timeout Timeout Timeout
9  node1.planetlab.uni-luebeck.de Timeout Timeout Timeout
10 node2pl.planet-lab.telecom-lille1.eu 1187 915 1064
11 node1pl.planet-lab.telecom-lille1.eu Timeout Timeout Timeout
12 node2.planetlab.uni-luebeck.de 625 600 642

```

4.2 Odozva na ping

Ping je aplikácia, ktorá zasiela ICMP správy „Echo Request“ a očakáva odpoveď „Echo Reply“. Na základe tejto informácie určí, či je cieľový uzol dostupný a ako dlho paketom trvá, kým sa správy dostanú k cieľu a späť [9]. Pomocou nástroja ping v Bash skripte [6] som vytvoril program, ktorý na všetky uzly siete Planet-Lab, vyčítané zo zdrojového súboru Planetlab.node, postupne zasiela 5 požiadaviek a testuje, či na tieto požiadavky príde očakávaná odpoveď. V prípade správnej odpovede zapíše program do výstupného súboru k DNS názvu uzlu hodnotu

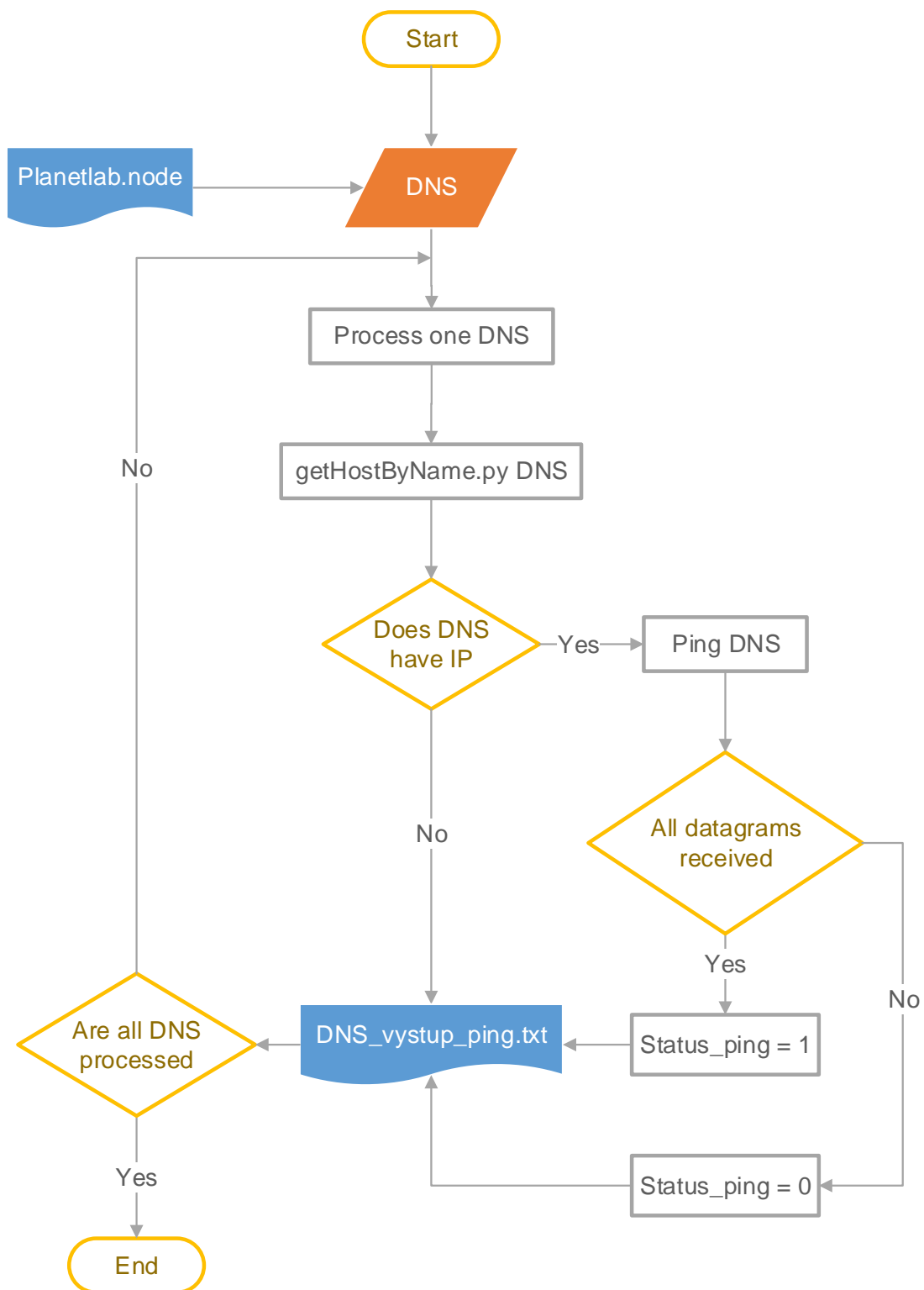
status_ping = 0. V prípade, že na niektorú z požiadaviek nepríde odpoveď, vyhodnotí sa status_ping = 1. Vývojový diagram testu dostupnosti je zobrazený na obrázku 4.4.

Výpis 4.3: Ukážka výstupu merania dostupnosti na ping

```
1 planetlab1.xeno.cl.cam.ac.uk 1 1 1 1 1
2 planetlab1.aut.ac.nz 1 1 1 1 1
3 planetlab1.informatik.uni-erlangen.de 0 0 0 0 1
4 planetlab-01.naist.jp UnknownDomainName UnknownDomainName
   UnknownDomainName UnknownDomainName UnknownDomainName
5 planetlab1.uta.edu 1 1 1 1 1
6 planetlab2.cis.upenn.edu 0 0 0 0 0
7 pl2.bit.uoit.ca 1 1 1 1 1
8 node3.planet-lab.titech.ac.jp UnknownDomainName
   UnknownDomainName UnknownDomainName UnknownDomainName
   UnknownDomainName
9 planetlab4.cs.uoregon.edu 1 1 1 1 1
10 planetlab1.netlab.uky.edu 0 0 0 0 0
11 planetlab1.cs.stevens-tech.edu 0 0 0 0 0
```

4.3 Meranie prenosovej rýchlosti

PlanetLab je živá sieť, ktorej všetky systémy sú využívané stovkami užívateľov, ktorí na stanicach robia svoj výskum. Pomocou meraní prenosovej rýchlosti som chcel zistiť mieru pripravenosti siete na väčšiu záťaž. Napísal som skript, ktorý bude vykonávať merania prenosovej rýchlosti medzi jednotlivými uzlami siete. Implementoval som program Rsync, ktorý slúži na pohodlný prenos súborov z jedného miesta, či viacerých zdrojov do cieľového umiestnenia. Rsync sa v praxi využíva najmä na zálohovanie dát na jednej stanici, či medzi viacerými stanicami. Využitie Rsync v sebe nesie viaceré výhody, ako je napríklad zachovanie oprávnení a vlastníckych informácií súborov, možnosť komprimácie dát, či možnosť prenosu šifrovaných dát cez SSH. Moja implementácia využitia Rsync na odoslanie súboru na vzdialený server je vyobrazené vo výpise kódu 4.4. Stiahnutie súboru zo vzdialeného serveru je opísané vo výpise 4.5. Vývojový diagram celého programu je na obrázku 4.5. Program načíta zo súboru Planetlab.node DNS mená stanic siete PlanetLab a postupne pre každú stanicu vykonáva merania. Skontroluje, či doménové meno je možné preložiť na IP adresu. V prípade, že to možné je, pripojí sa na stanicu a skontroluje,



Obr. 4.4: Vývojový diagram merania dostupnosti serveru pomocou programu ping

či na servery neostali súbory z meraní, ktoré mohli v minulosti zlyhať. Keď takéto súbory nájde, tak ich odstráni a pomocou Rsync preniesie pripravený 50 MB súbor z cesty `/home/cesnet_feec/xilkop00/measurements/upload/file_50mb.pdf` na vzdialenú stanicu `/home/cesnet_feec/xilkop00/upload_file/uploaded_file_$(hostname).pdf`. Po uložení súboru vypíše program Rsync vďaka použitému parametru „-- stats“ prenosové štatistiky. Jedným z údajov tejto štatistiky je práve prenosová rýchlosť v bajtoch za sekundu (B/s). Tento údaj, teda hodnotu rýchlosti odosielania si skript pripraví na uloženie. Následne skript mnou nahraný súbor zo serveru stiahne rovnakým spôsobom, akým bol na server uložený. Získam tým prenosovú rýchlosť sťahovania. Doménové meno a hodnotu nahrávania i sťahovania skript uloží do výstupného súboru a pokračuje v meraní prenosových parametrov na ďalšom serveri.

Tab. 4.3: Použité argumenty programu Rsync

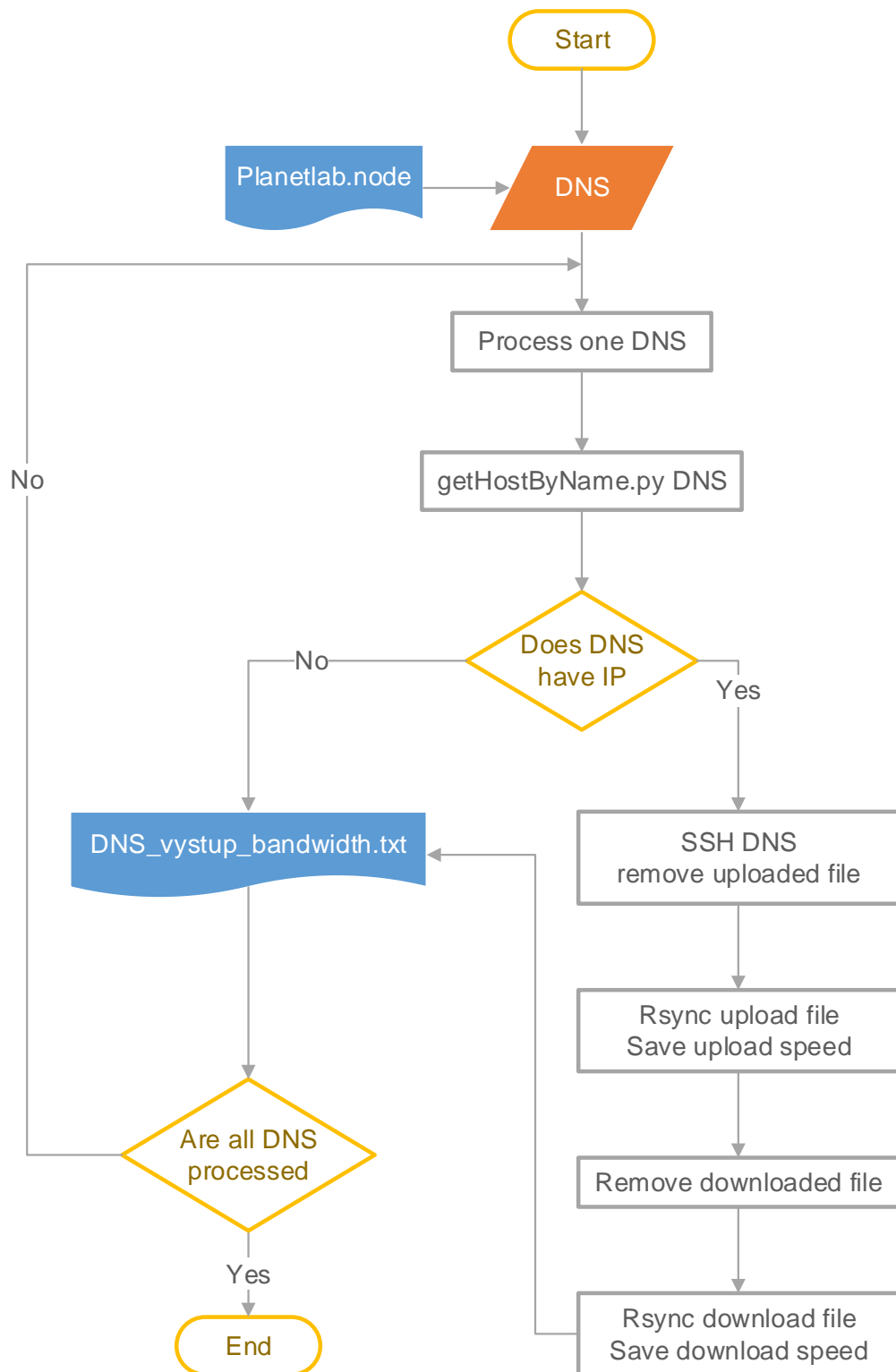
Použité argumenty programu Rsync	Význam argumentov
--stats	výpis štatistík o prenose
[-a archive]	zachovanie oprávnení a vlastníckych práv
[-v verbose]	prihlasovacie meno
[-e]	doplňkové nastavenia
-rsync-path	spustenie programu na vzdialenej stanici
-remove-source-files	odosielateľ zmaže súbor po odoslaní

Výpis 4.4: Získanie prenosovej rýchlosti pri odosielaní súboru cez Rsync

```
upload_stats_output=$(rsync --stats -ave
  "ssh -o BatchMode=yes -o StrictHostKeyChecking=no
  -o UserKnownHostsFile=/dev/null -o ConnectTimeout=10
  -i /home/cesnet_feec/.ssh/xilkop00_planetlab"
  --rsync-path="mkdir -p /home/cesnet_feec/xilkop00/
  upload_file && rsync"
  /home/cesnet_feec/xilkop00/measurements/upload/
  file_50mb.pdf
  cesnet_feec@$foo:/home/cesnet_feec/xilkop00/
  upload_file/uploaded_file_$(hostname).pdf)
upload_RETVAL=$?
[ $upload_RETVAL -eq 0 ] && upload_speed=$(echo
  $upload_stats_output | awk -F 'bytes ' '{print $9}' |
  awk '{print $1}') && echo $upload_stats_output >> /
  home/cesnet_feec/xilkop00/logs/rsync/upload_speed.txt
[ $upload_RETVAL -ne 0 ] && upload_speed="None"
```

Výpis 4.5: Získanie prenosovej rýchlosti pri prijímaní súboru cez Rsync

```
download_stats_output=$(rsync --stats -ave
  "ssh -o BatchMode=yes -o StrictHostKeyChecking=no
  -o UserKnownHostsFile=/dev/null -o ConnectTimeout=10
  -i /home/cesnet_feec/.ssh/xilkop00_planetlab"
  --remove-source-files cesnet_feec@$foo:/home/
  cesnet_feec/xilkop00/upload_file/
  uploaded_file_$(hostname).pdf
  /home/cesnet_feec/xilkop00/measurements/download/
  downloaded.pdf)
download_RETVAL=$?
[ $download_RETVAL -eq 0 ] && download_speed=$(echo
  $download_stats_output | awk -F 'bytes ' '{print $9}'
  | awk '{print $1}') && echo $download_stats_output >>
  /home/cesnet_feec/xilkop00/logs/rsync/download_speed.
  txt && rm /home/cesnet_feec/xilkop00/measurements/
  download/downloaded.pdf
[ $download_RETVAL -ne 0 ] && download_speed="None"
```

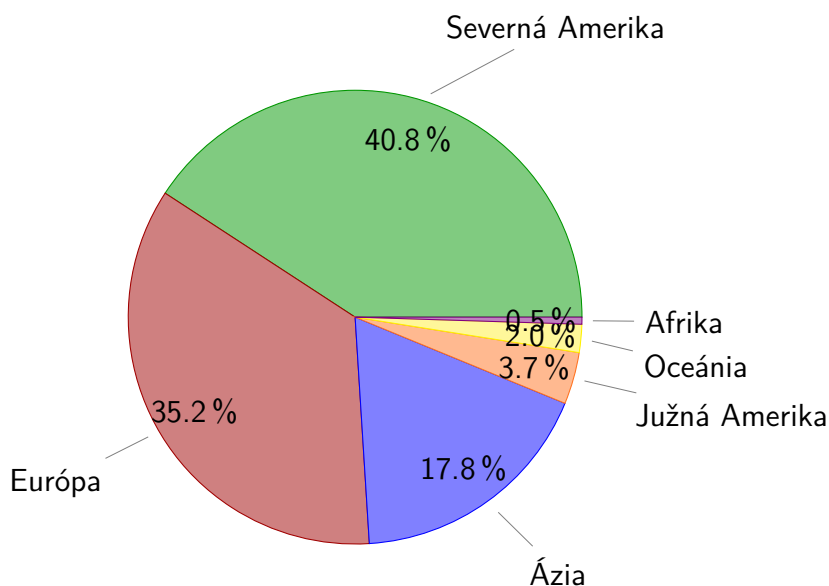


Obr. 4.5: Vývojový diagram merania prenosovej rýchlosti

5 ANALÝZA DOSIAHNUTÝCH VÝSLEDKOV

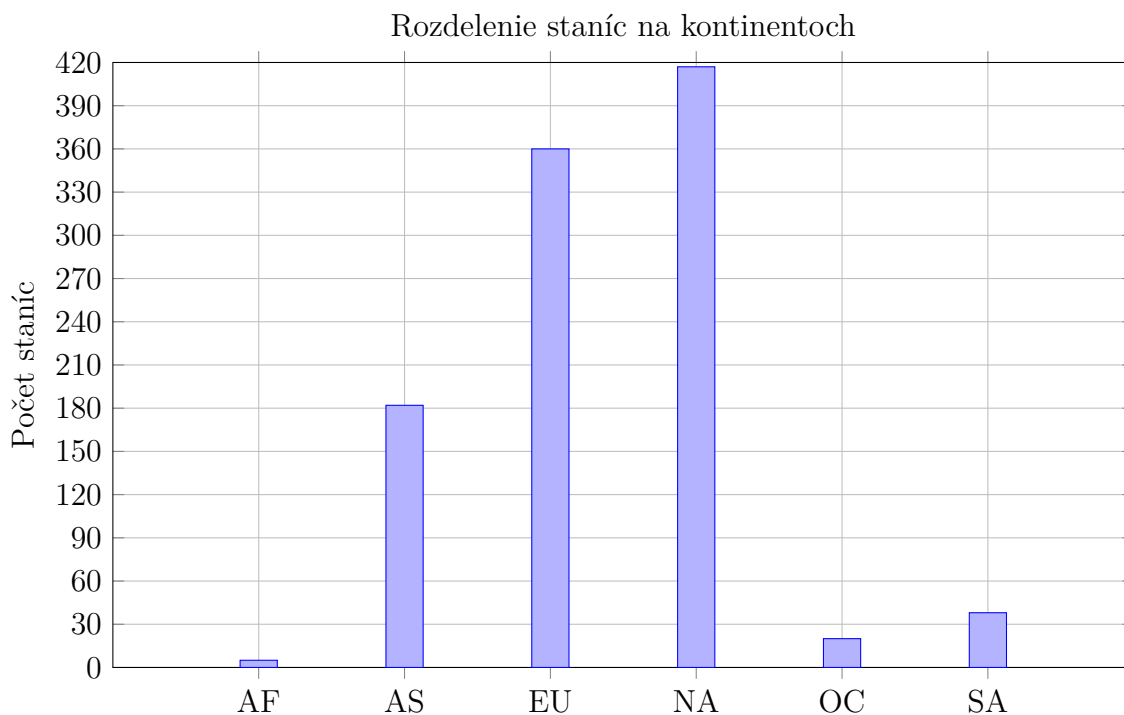
5.1 Geografická analýza staníc siete PlanetLab

Pomocou programu, ktorý je popísaný v kapitole 3.2, som vytvoril zoznam všetkých staníc siete PlanetLab na svete. Vo vytvorenom zozname môžeme pozorovať, že 417 z celkového počtu 1022 staníc sa nachádza na území Severnej Ameriky, čo tvorí až 40,8% z celkového počtu staníc. To, že najväčšie množstvo staníc je na území Severnej Ameriky odzrkadľuje fakt, že táto výskumná sieť vznikla práve na tomto kontinente a bol tu aj najväčší potenciál v jej budovaní. Celkovo 360 staníc, čo je 35,2% z celkového počtu, sa nachádza v Európe. Ďalšia početná skupina staníc sa nachádza v Ázii, a to celkovo 182 staníc, čo činí 17,8%. Pomerné rozdelenie staníc na kontinentoch som zobrazil na grafe 5.1 a 5.2.



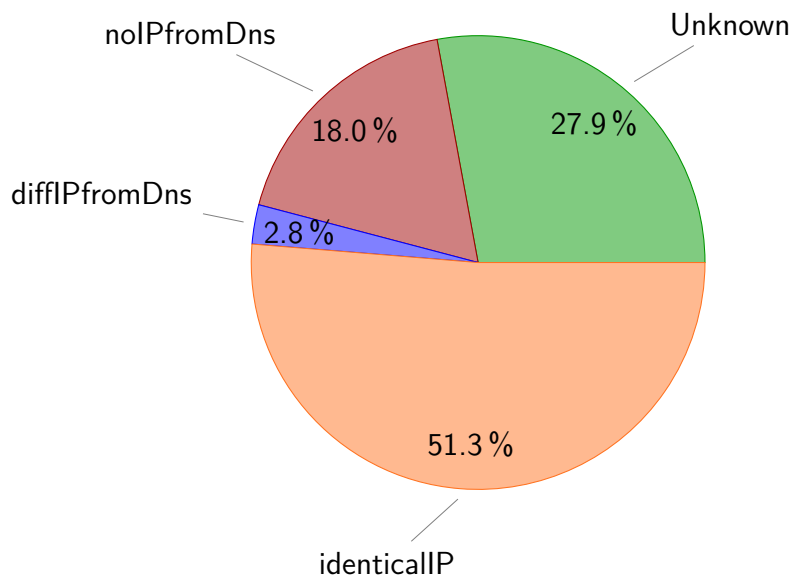
Obr. 5.1: Percentuálne rozdelenie staníc na kontinentoch

Do zoznamu staníc som ku každej stanici vkladal aj IP adresu uverejnenú na webových stránkach PlanetLabu, v tom prípade, ak bola na webových stránkach uvedená. Tú následne program porovnal s prekladom doménového mena na IP adresu. Ako môžeme vyčítať z grafu 5.4, 29 staníc má na webových stránkach chybnú uvedenú IP adresu stanice. Celkovo sa jedná len o chybovosť 2,8%, viz graf 5.3, čo by za bežných okolností bolo prípustné. Avšak, keď prihliadneme na fakt, že sa jedná o profesionálnu sieť odborných inštitúcií, túto chybovosť môžeme považovať za relatívne vysokú. Taktiež až 18,0% doménových mien, teda celkovo 184 doménových mien, nebolo možné preložiť na IP adresu. Na tieto stanice by sa preto cez SSH určite nebolo možné pripojiť pomocou doménového mena, pretože by pri pripájaní

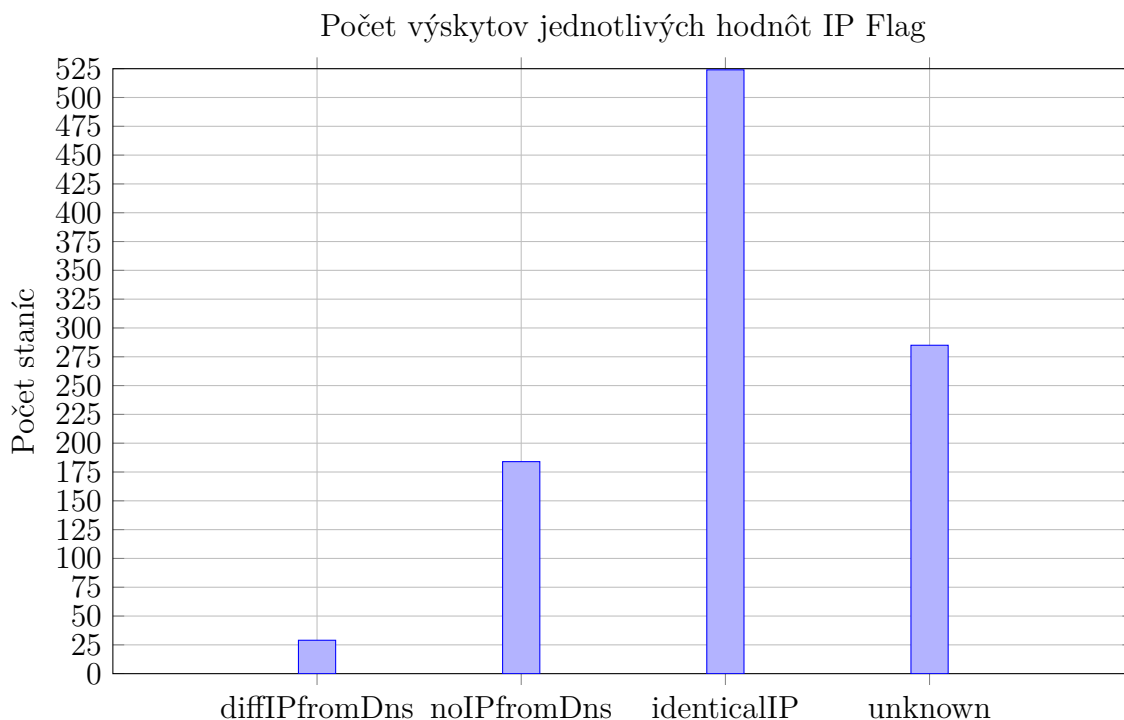


Obr. 5.2: Graf zastúpenia staníc pre každý kontinent

zlyhal preklad. 524 staníc, teda 51,3%, malo IP adresu správne vyplnenú. 285 staníc, teda 27,9%, nemalo IP adresu uverejnenú na webových stránkach, a preto som ju nemal s čím porovnávať.



Obr. 5.3: Percentuálne rozdelenie hodnôt IP Flag



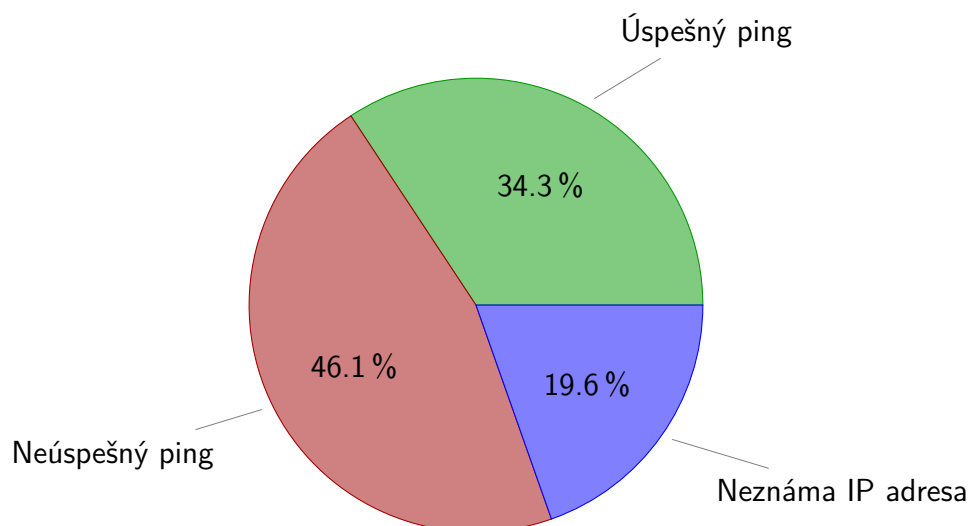
Obr. 5.4: Percentuálny graf zobrazenia hodnôt IP Flag

5.2 Výkonová analýza siete

Merania prenosových parametrov siete prebiehali v časovom rozmedzí dvoch mesiacov. Počas tohto obdobia pripadlo na jednu stanicu siete PlanetLab v priemere 3206 behov meraní dostupnosti na ping, 3110 behov dostupnosti na SSH a 2589 behov meraní prenosovej rýchlosti. Tento rozdiel v počte behov je daný odlišnou časovou náročnosťou testov. Výsledkov meraní, ktoré som obdržal zo spomínaných skriptov, som spracoval do štatistickej a grafickej podoby, ktorú môžete vidieť nižšie.

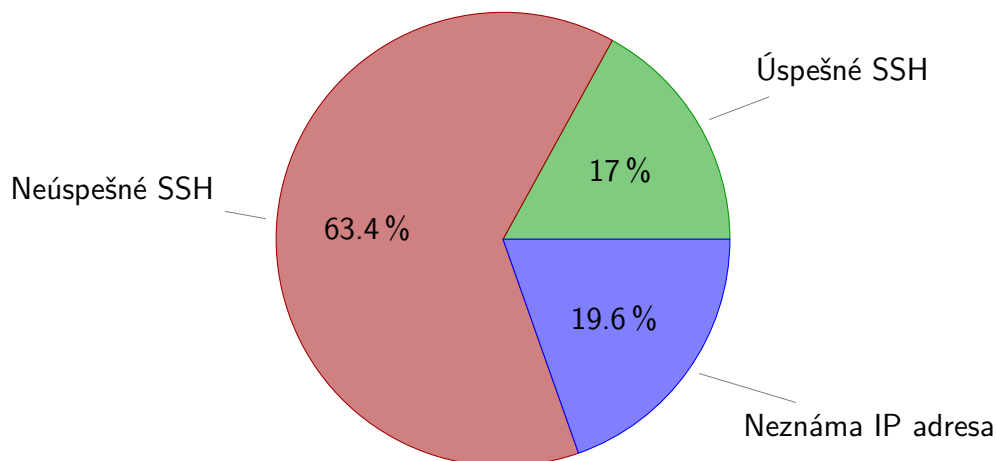
Dostupnosť uzlov

Ako som popísal v kapitole 4, overenie dostupnosti uzlov sa sústredilo na program ping a SSH. Na obrázku 5.5 je zobrazený koláčový graf, ktorý zobrazuje celkový stav dostupnosti uzlov na ping. Za dostupné stanice berieme tie, ktoré boli dostupné aspoň 90% meraní. Najväčšie percento uzlov patrí do skupiny s neúspešnou odozvou. Je ich celkom 46,1%. To mohlo byť spôsobené blokovaním ICMP paketov po ceste k danému uzlu, firewallom priamo na uzle, či z dôvodu fyzického vypnutia uzla. Do najmenej početnej skupiny so 19,6% patria uzly, kde nebol úspešný preklad doménového mena. Zvyšných 34,3% tvorili uzly, ktoré sú dostupné na ping.



Obr. 5.5: Graf dostupnosti uzlov siete PlanetLab na program ping

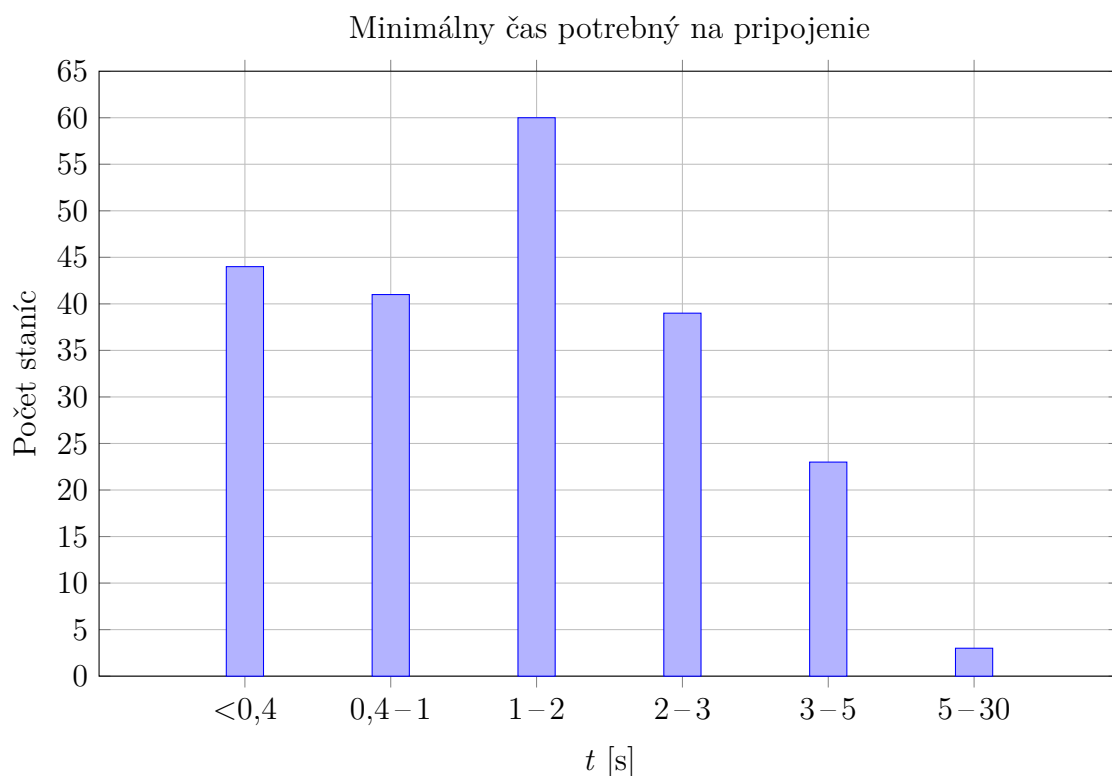
Výsledky merania dostupnosti uzlov na protokol SSH sú v porovnaní s programom ping o niečo horšie. Ako je zobrazené na obrázku 5.6, bolo sa možné vzdialene pripojiť iba na 17% z celkového počtu 1026 uzlov. V porovnaní s dostupnosťou staníc na program ping je rozdiel až 17,3%. Počet uzlov, kde nebol úspešný preklad doménového mena je v oboch prípadoch rovnaký, teda 19,6%. Počet nedostupných uzlov tvorí až 63,4% všetkých uzlov.



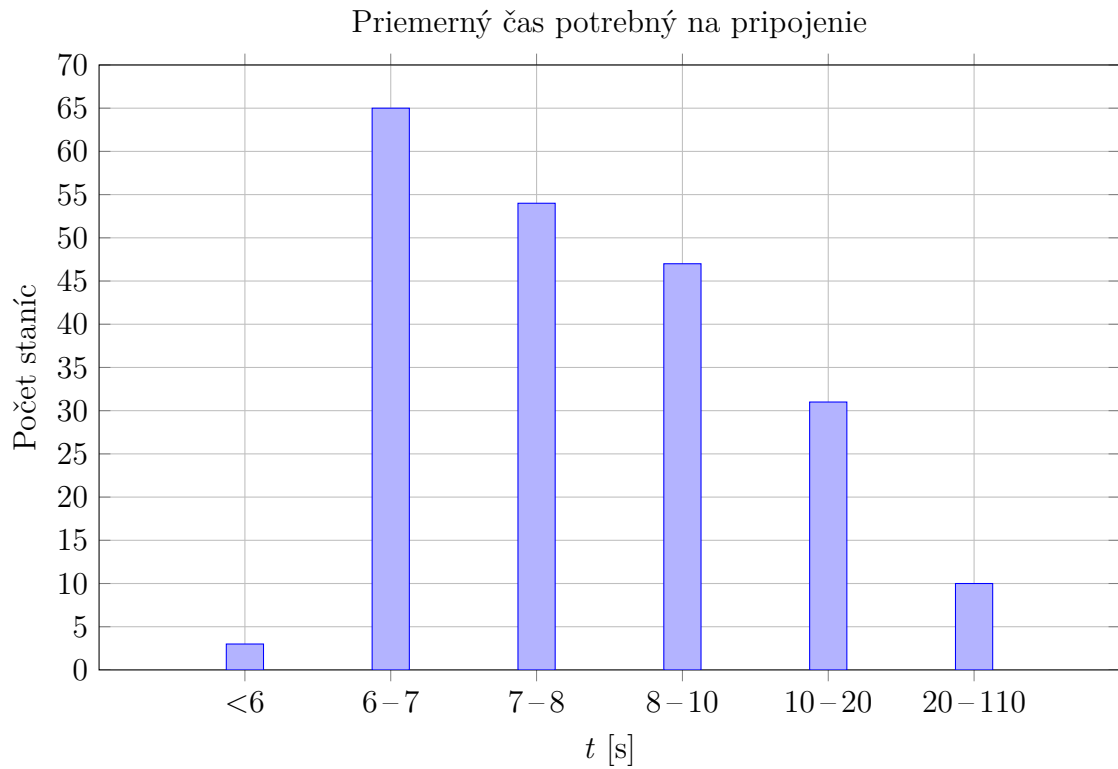
Obr. 5.6: Graf dostupnosti uzlov siete PlanetLab na protokol SSH

Z nameraných dát boli vybrané minimálne, priemerné a maximálne hodnoty doby pripojenia pre uzly, na ktoré sa bolo možné úspešne pripojiť pomocou SSH. Tieto hodnoty sú vynesené v stĺpcových grafoch 5.7, 5.8 a 5.9. Z najkratších časov v grafe 5.7 môžeme usúdiť, že na väčšinu uzlov, na ktoré je možné sa pripojiť, sa

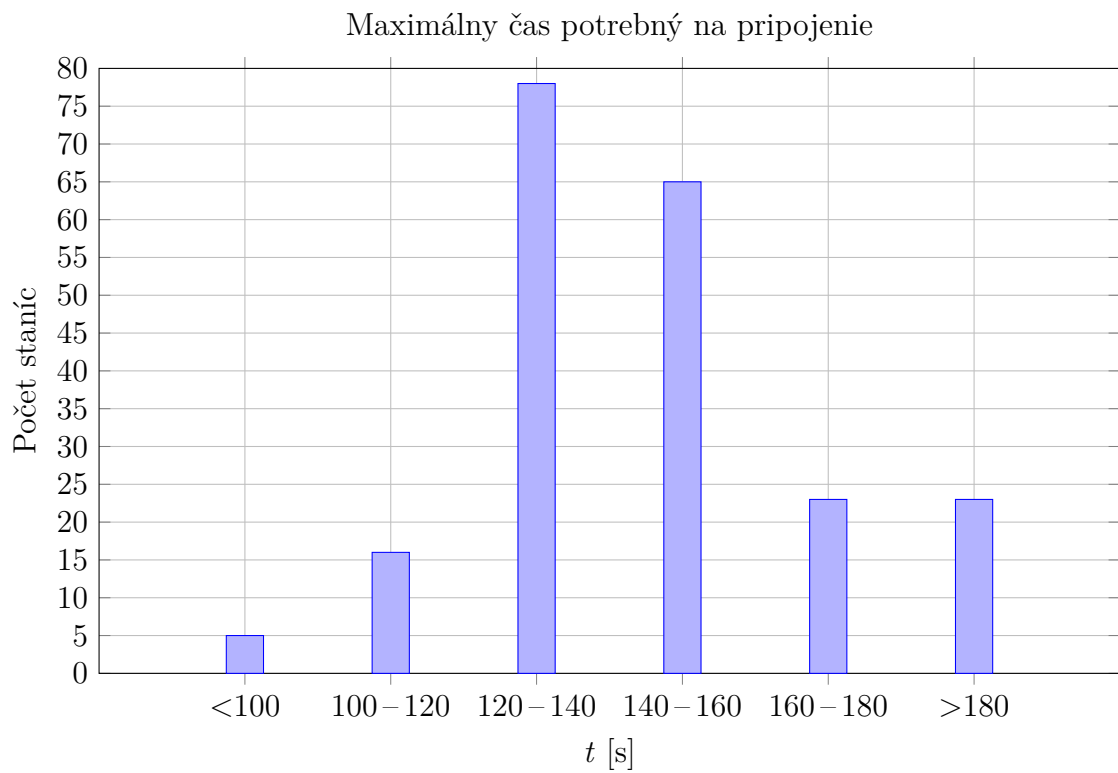
v najlepšom prípade pripojíme do 2 sekúnd. Pri pohľade na graf priemerných časov 5.8 potrebných na pripojenie vidíme, že nutný čas na pripojenie poskočil o niečo vyššie. Väčšina staníc sa ale pripojí približne do 10 sekúnd, presná priemerná hodnota je 9,96 sekundy. Pri vytváraní spojenia sme do argumentu SSH zadali hodnotu „timeout=30“, čo v praxi znamená, že keď pripojenie neprebehne do 30 sekúnd, tak sa SSH spojenie ukončí. Za týchto okolností sa prejavila istá anomália. Pri množstve meraní prebiehalo pripájanie na vzdialenú stanicu rádovo dlhšie ako 30 sekúnd. Pri testovaní skriptu, pomocou ktorého som sa na stanice pripájal, som danú situáciu nevedel navodiť, takže nebolo možné zozbierať ladiace výpisy. Dá sa ale dedukovať, že stanica, na ktorú sa vytváralo spojenie bola nad mieru vyťažená, a SSH server tak nemal dostatočný priestor na obsluhu požiadavkov. Pri zoskupení najhorších časov je znateľné, že ani 100 sekúnd nebolo poväčšine dost na vytvorenie spojenia. Z tohto grafu 5.9 tiež môžeme vyčítať, že z celkového počtu 1026 uzlov sa pri každom meraní bolo možné pripojiť celkovo na 233 uzlov.



Obr. 5.7: Najkratší nameraný čas, za ktorý sa bolo možné pripojiť ku stanici



Obr. 5.8: Priemerný čas potrebný na pripojenie ku stanici



Obr. 5.9: Najdlhší nameraný čas, za ktorý sa bolo možné pripojiť ku stanici

Prenosová rýchlosť

Meranie prenosovej rýchlosti medzi stanicami siete PlanetLab som realizoval pomocou skriptu 4.3, ktorý prenáša súbor o veľkosti 50 MB z jednej stanice na druhú. Pri prvých meraniach som použil súbor o veľkosti 350 MB a testoval meranie na všetky stanice každú hodinu. Na niekoľkých stanicach vznikol problém so swapovaním pamäte na disk, preto som sa rozhodol zmenšiť súbor, s ktorým pracujem. Od tej chvíle nenastal žiaden ďalší problém s pamäťou.

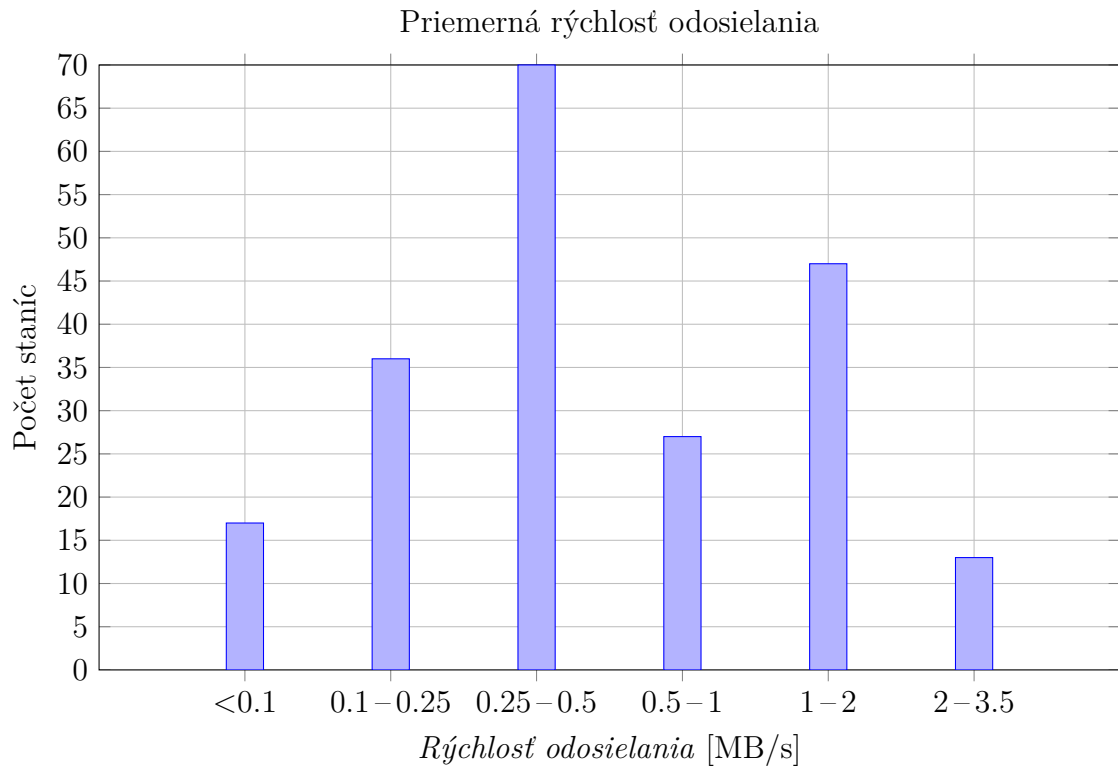
Z nameraných dát som vybral minimálnu a maximálnu hodnotu a súčasne som vypočítal priemernú hodnotu daného merania pre každú stanicu. Následne som z týchto minimálnych, maximálnych a priemerných hodnôt vypočítal priemery, ktoré uvádzam v tabuľke 5.1. Ak porovnáme dáta v danej tabuľke, môžeme vidieť, že rýchlosť odosielania je pre minimálne hodnoty vyššia o 59,6%, pre priemerné hodnoty o 14,7% a pre maximálne hodnoty o 5,4%.

Tab. 5.1: Priemerné hodnoty jednotlivých meraní

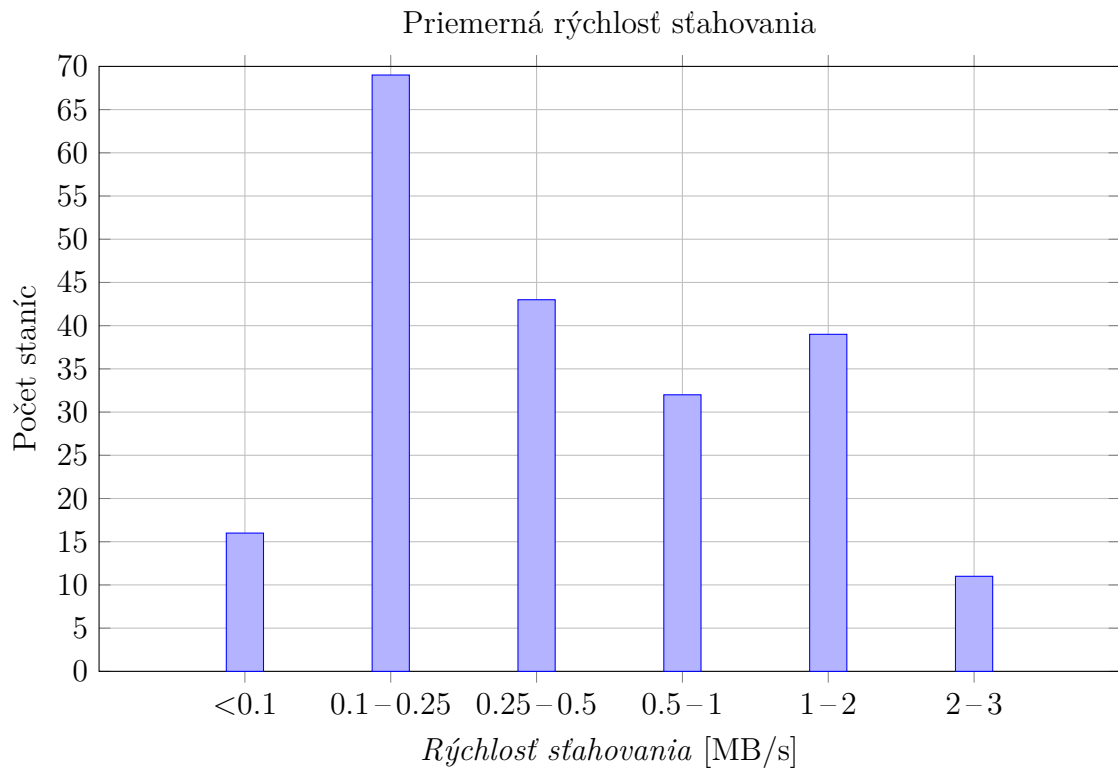
Typ prenosu	Hodnoty [MB/s]		
	Minimálne	Priemerné	Maximálne
Stahovanie	0.001406	0.61	3.788
Odosielanie	0.002244	0.6994	3.994

Taktiež som tieto priemerné namerané dáta zobrazil v stĺpcových grafoch, ktoré je možné vidieť nižšie. Graf 5.10 ukazuje závislosť priemernej rýchlosti odosielania na počet staníc. Na najväčšie množstvo staníc sa priemerná rýchlosť odosielania pohybovala okolo 0,25 až 0,5 MB/s. Maximálna priemerná rýchlosť odosielania dostala na 2 až 3,5 MB/s.

Graf 5.11 popisuje závislosť priemernej rýchlosti sťahovania na počet staníc. Pre najväčšie množstvo staníc som zaznamenal rýchlosť sťahovania v rozmedzí 0,1 až 0,25 MB/s. Taktiež môžeme z tohto grafu vyčítať maximálnu priemernú rýchlosť sťahovania a teda hodnotu 2 až 3 MB/s.



Obr. 5.10: Priemerná rýchlosť odosielania na server v sieti PlanetLab



Obr. 5.11: Priemerná rýchlosť sťahovania zo serveru v sieti PlanetLab

6 ZÁVER

V úvode tejto práce som sa zoznámil s experimentálnu sieťou PlanetLab. Opísal som stručný vývoj tejto siete a predstavil pár projektov, ktoré sú v rámci siete realizované. V zadaní práce bolo zostaviť zoznam staníc siete PlanetLab nachádzajúcich sa v Európe, ako aj najšť efektívny a účinný spôsob merania parametrov z tejto siete, konkrétne dostupnosť na ping a dobu odozvy na SSH. V diplomovej práci som sa zameril na overovanie dostupnosti uzlov z viacerých staníc v rámci Európy. Navyše som vytvoril program na meranie prenosovej rýchlosti, a taktiež som po konzultácií s vedúcim práce doc. Ing. Danom Komosným Ph.D. vytvoril globálny zoznam staníc a neobmedzil som sa iba na Európu. Ďalšou z hlavných úloh bola následná interpretácia získaných dát a jej prehľadné zobrazenie v grafickej podobe. Všetky vytvorené programy sa nachádzajú na priloženom DVD.

Na automatizované vytvorenie zoznamu staníc som napísal program v jazyku Python s názvom `planetlab_crawler_selenium.py`. Tento program sa prihlási na webové stránky siete PlanetLab a informácie z nej spracuje do výstupného súboru `planetlab.node`. Informácie, ktoré sa ukladajú do výstupného súboru, zahŕňajú napríklad DNS názov stanice, IP adresu, geografickú polohu, ako aj zemepisnú šírku a dĺžku. Nevýhodou daného riešenia je nutnosť využitia grafického užívateľského prostredia. Preto som neskôr vytvoril ďalší program v jazyku Python `planetlab_list_creator.py`, ktorý na správne fungovanie nepotrebuje grafické rozhranie.

Čo sa týka jednotlivých meraní, ako prvé som vytvoril programy v jazyku Bash na overenie dostupnosti pomocou programu ping (`measure_ping.sh`) a overenie dostupnosti na protokol SSH (`measure_ssh.sh`), ktorý zmeria časové oneskorenie. Meranie dostupnosti na ping ukázalo, že v sieti sa nachádza 34,3% dostupných staníc, a naopak 46,1% staníc je nedostupných. 19,6% staníc nebolo možné overiť z dôvodu neúspešného prekladu doménového mena, a teda neznámej IP adresy.

Pri meraní dostupnosti na protokol SSH som meral čas nutný na pripojenie sa na jednotlivé uzly. Následná analýza nameraných dát demonštrovala, že na 816 uzlov sa nebolo možné pripojiť vôbec. U 177 uzlov nebolo možné preložiť doménové meno na IP adresu a na zvyšných 639 sa nebolo možné pripojiť do 30 sekúnd. Zároveň sa bolo možné pripojiť pri každom meraní na 210 uzlov. V priemere sa čas nutný na pripojenie pohybuje okolo 6 až 10 sekúnd.

Na meranie prenosovej rýchlosti, teda rýchlosti sťahovania a odosielania dát, som vytvoril program `measure_bandwidth`. Meranie prebiehalo odosielaním a prijímaním súboru o veľkosti 50 MB. Zo získaných dát som pre každú stanicu našiel minimálnu, maximálnu a priemernú hodnotu. Z týchto hodnôt som vypočítal priemery. Z meraní sa ukázalo, že prenosová rýchlosť odosielania bola vždy vyššia oproti rýchlosti

stahovania. Zatiaľ čo priemerná rýchlosť odosielania sa pohybovala v rozmedzí 0,25 až 0,5 MB/s, priemerná rýchlosť stahovania bola medzi 0,1 až 0,25 MB/s.

LITERATÚRA

- [1] Agency, I. . M.: Country Codes – ISO 3166. [online], May 2016, [cit. 3. 5. 2016]. URL: <http://www.iso.org/iso/home/standards/country_codes.htm>
- [2] Carriere, D.: Geocoder: Simple, Consistent. [online], 2013, [cit. 10. 5. 2016]. URL: <<https://geocoder.readthedocs.io/>>
- [3] Fiuczynski, M. E.: *User's Guide*. Naposledy uprvené 02. 1. 2006 [cit. 8. 5. 2014]. URL: <<https://www.planet-lab.org/doc/guides/user>>
- [4] Google: Test Automation for Web Applications. [online], naposledy uprvené 10. 2. 2015 [cit. 7. 12. 2015]. URL: <http://www.seleniumhq.org/docs/01_introducing_selenium.jsp>
- [5] ILKO, P.: *Geografická pozice serverů sítě PlanetLab*. Bakalárska práce, Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikac, 2014.
- [6] Mendel, C.: *Advanced bash scripting guide 5.3: an in-depth exploration of the art of shell scripting*. Lexington: Lulu, 2008, ISBN 978-143-5752-191.
- [7] Peterson, L.; Anderson, T.; Culler, D.; aj.: A Blueprint for Introducing Disruptive Technology into the Internet. In *Proceedings of HotNets-I*, Princeton, New Jersey, October 2002, str. 7.
- [8] PlanetLab: An open platform for developing, deploying, and accessing planetary-scale services. [online], 2007, [cit. 10. 5. 2016]. URL: <<https://www.planet-lab.org/>>
- [9] Postel, J.: Internet Control Message Protocol, RFC 792. Internet Engineering Task Force, September 1981. URL: <<http://www.ietf.org/rfc.html>>
- [10] Provos, N.; Friedl, M.: OpenSSH Manual pages. [online], July 2008, [cit. 3. 4. 2014]. URL: <<http://www.openssh.com/manual.html>>
- [11] Ptáček, L.: *Linux : dokumentační projekt*. Brno: Computer Press, 2007, ISBN 978-80-251-1525-1.
- [12] Python Software Foundation: urllib2 — extensible library for opening URLs. [online], naposledy uprvené 7. 5. 2016 [cit. 10. 5. 2016]. URL: <<https://docs.python.org/2/library/urllib2.html>>
- [13] Reitz, K.: Requests: HTTP for Humans. [online], naposledy uprvené 2015 [cit. 7. 12. 2015]. URL: <<http://docs.python-requests.org/en/latest>>

- [14] Richardson, L.: Beautiful Soup. [online], naposledy upravené 29.9.2015 [cit. 7.12.2015]. URL: <<http://www.crummy.com/software/BeautifulSoup/>>
- [15] Rita, P.: *TCP/IP v kostce. 2. vyd.* České Budějovice: Kopp, 2009, ISBN 978-80-7232-388-3, 620 s.
- [16] Stephan Richter: lxml - XML and HTML with Python. [online], naposledy upravené 5.5.2016 [cit. 10.5.2016]. URL: <<http://lxml.de/>>
- [17] Wayne Davison, P. M., Andrew Tridgell: rsync(1) - Linux man page. [online], [cit. 16.5.2016]. URL: <<http://linux.die.net/man/1/rsync>>

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

DNS	Domain Name System
DSPS	Distributed Stream Processing Systems
GUI	Graphical User Interface,
GPS	Global Positioning System
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
MIT	Massachusetts Institute of Technology
PLC	PlanetLab Central
PLE	PlanetLab Europe
QoS	Quality of Service
URL	Uniform Resource Locator
VServer	Virtual Server