

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

NÁSTROJE PRO TESTOVÁNÍ PROPUSTNOSTI SÍTĚ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PAVOL LOFFAY

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

NÁSTROJE PRO TESTOVÁNÍ PROPUSTNOSTI SÍTĚ

COMPARISON OF OPEN-SOURCE SOFTWARE TOOLS FOR NETWORK DIAGNOSTICS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PAVOL LOFFAY

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETR MATOUŠEK, Ph.D.

BRNO 2013

Abstrakt

Cílem této práce je vytvořit rešerši existujících open source nástrojů, které jsou zaměřeny na diagnostiku síťového provozu pomocí měření propustnosti a dalších základních parametrů. Práce se věnuje vytvoření metodiky pro testování a vzájemné porovnání nástrojů. Výsledkem práce je doporučení pro běžného uživatele spočívající v doporučení správného nástroje pro měření příslušného parametru síťového provozu.

Abstract

The aim of this work is to make a research of existing open source tools, which are concerned with the diagnosis of the network transmission in the form of measuring of throughput and other basic parameters. The work is devoted to the construction of methodics for testing and comparing devices between themselves. The result of the work is recommendation to an ordinary customer the right application for measuring of given network transmission parameter.

Klíčová slova

propustnost, zpoždění paketů, rozptyl zpoždění, ztráta paketů, změna poradí paketů

Keywords

throughput, delay, jitter, delay variation, packet loss, packet reordering

Citace

Pavol Loffay: Nástroje pro testování propustnosti
sítě, bakalářská práce, Brno, FIT VUT v Brně, 2013

Nástroje pro testování propustnosti sítě

Prohlášení

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Petra Matouška, Ph.D.

.....

Pavol Loffay
14. května 2013

Poděkování

Týmto by som chcel poďakovať vedúcemu mojej práce pánovi Ing. Petrovi Matouškovi, Ph.D. za jeho cenné rady a nemalé množstvo času, ktoré mi venoval pri tvorbe tejto práce.

© Pavol Loffay, 2013.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Merateľné parametre sieťového prenosu	5
2.1	Jednosmerné oneskorenie (delay)	5
2.2	Rozptyl oneskorenia (jitter)	6
2.3	Priepustnosť (throughput)	8
2.3.1	Meranie TCP priepustnosti podľa RFC 6494	8
2.4	Strata paketov	9
2.5	Zmena poradia paketov	9
2.6	Prínos pre našu prácu	10
3	Metodika testovania nástrojov	11
3.1	Výber sledovaných parametrov sieťového prenosu	11
3.2	Výber ďalších funkcionálnych vlastností	12
3.3	Metodika testovania	12
3.4	Zhrnutie	13
4	Prehľad testovaných nástrojov	14
4.1	Nástroj Iperf	14
4.1.1	Architektúra	14
4.1.2	Softvérové nároky	15
4.1.3	Popis vybraných parametrov	15
4.1.4	Ukážka testov	16
4.1.5	Nekorektné správanie	17
4.1.6	Zhodnotenie	17
4.2	Nástroj Netperf	18
4.2.1	Architektúra	18
4.2.2	Softvérové nároky	18
4.2.3	Popis vybraných parametrov	19
4.2.4	Ukážka testov	19
4.2.5	Nekorektné správanie	20
4.2.6	Zhodnotenie	21
4.3	Nástroj BWCTL	21
4.3.1	Architektúra	21
4.3.2	Softvérové nároky	22
4.3.3	Popis vybraných parametrov	22
4.3.4	Ukážka testov	23
4.3.5	Nekorektné správanie	24

4.3.6	Zhodnotenie	24
4.4	Nástroj OWAMP	24
4.4.1	Architektúra	24
4.4.2	Softvérové nároky	24
4.4.3	Popis vybraných parametrov	25
4.4.4	Ukážka testov	25
4.4.5	Nekorektné správanie	26
4.4.6	Protokol OWAMP	26
4.4.7	Zhodnotenie	26
4.5	Nástroj Thrulay	27
4.5.1	Architektúra	27
4.5.2	Softvérové nároky	27
4.5.3	Popis vybraných parametrov	28
4.5.4	Ukážka testov	28
4.5.5	Nekorektné správanie	29
4.5.6	Zhodnotenie	29
4.6	Nástroj Nuttcp	29
4.6.1	Architektúra	30
4.6.2	Softvérové nároky	30
4.6.3	Popis vybraných parametrov	30
4.6.4	Ukážka testov	31
4.6.5	Nekorektné správanie	31
4.6.6	Zhodnotenie	31
4.7	Nástroj BWPing	32
4.7.1	Architektúra	32
4.7.2	Softvérové nároky	32
4.7.3	Popis vybraných parametrov	33
4.7.4	Ukážka testov	33
4.7.5	Nekorektné správanie	33
4.7.6	Zhodnotenie	33
4.8	Celkové vyhodnotenie	34
5	Testovanie na reálnej sieti	36
5.1	Metodika testovania	36
5.1.1	Vstupné podmienky	37
5.1.2	Rozmiestnenie	37
5.1.3	Testy	38
5.1.4	Štatistická úprava výsledkov	39
5.2	Analýza výsledkov	39
5.2.1	TCP Priepustnosť	39
5.2.2	UDP Priepustnosť	42
5.2.3	Strata paketov	44
5.2.4	Jednosmerné oneskorenie	46
5.3	Záver s doporučením	46
6	Záver	48
A	Obsah CD disku	51

Kapitola 1

Úvod

Počítačové siete sú v dnešnej dobe natoľko rozšírené, že ich denne priamo či nepriamo využívame v práci alebo bežnom živote. Preto je nevyhnutné vedieť správne diagnostikovať chyby prenosu alebo parametre siete, za ktoré platíme svojmu poskytovateľovi pripojenia. Bežní užívatelia pri problémoch s pripojením väčšinou nevedia diagnostikovať príčinu, preto využívajú služby vyškolených odborníkov.

Táto práca sa zameriava na možnosti testovania sieťového prenosu pre bežného užívateľa. Pre účely testovania sme vybrali voľne dostupné open source nástroje, aby ich mohla použiť najširšia skupina užívateľov. Zameriame sa na meranie parametrov už skonvergovanej siete, ktorej topológia nie je známa. Tým pádom k sieti budeme pristupovať ako k čiernej skrinke.

Pri problémoch so sieťovým prenosom mnohokrát dochádza u samotných užívateľov. Sú spôsobené zlou konfiguráciou sieťových rozhraní, chybou operačného systému alebo problémom spôsobeným iným softvérom. Riešenie týchto chýb nie je predmetom našej práce a budeme predpokladať, že sieťové rozhrania koncových staníc sú korektne nastavené a sieťová komunikácia prebieha správne.

Hlavným cieľom práce je vyvinúť doporučenie pre koncového užívateľa, ktorej predmetom je otestovanie vybraných parametrov sieťového prenosu. Správnosť výsledkov samozrejme závisí od dokonalosti použitých nástrojov. Je dôležité si uvedomiť, že výsledky môžu byť ovplyvnené aj inými faktormi, ktoré ovplyvňujú sieťovú komunikáciu. Pričom užívateľ by mal získať výsledky, ktoré bude schopný porovnať s hodnotami uvedenými v SLA (Service Level Agreement) od poskytovateľa konektivity. Aby sme sa dostali k vytvoreniu doporučenia na testovanie parametrov sieťového prenosu, je nutné vykonať prieskum a testovanie. Každá uvedená kapitola bude bližším krokom k danému cieľu.

V prvej časti práce budú popísané parametre sieťového prenosu, ktoré sa dajú merať. Nezameriame sa len na parametre prenosu, ale zavedieme požiadavky aj na funkcionálne vlastnosti nástrojov. Vytvoríme metodiku, ktorej účelom bude určiť, či nástroj implementuje testovanie týchto parametrov. Obsahom metodiky bude taktiež overenie funkcionálnych vlastností nástrojov. Táto metodika bude následne použitá v ďalšej kapitole.

Ďalšia časť práce sa venuje jednotlivým nástrojom. Budú uvedené softvérové nároky, popis parametrov, ukážky testov s príkladmi spustenia, ale aj chyby s nekorektným správaním. Výsledkom tejto kapitoly bude prehľadné zhrnutie, ktoré ukáže, aké parametre sa dajú s jednotlivými nástrojmi merať.

Tretia kapitola bude zameraná na testovanie reálnej siete. Merania budú prebiehať z viacerých miest voči jednému koncovému bodu. Týmto spôsobom otestujeme parametre prenosu na rozličných sieťach. Testovania budú vykonané opakovane a v rôznych časoch, aby

sme získali štatisticky správne informácie. Výsledkom tejto kapitoly bude zhodnotenie nástrojov, ich použiteľnosť s ohľadom na správnosť získaných informácií z testovania. Taktiež bude uvedené doporučenie pre koncového užívateľa, aký nástroj má použiť na meranie daného parametra sieťového prenosu. Záver tejto podkapitoly je hlavným výsledkom práce, čo predstavuje zistenie reálnej použiteľnosti testovaných nástrojov.

Kapitola 2

Merateľné parametre sieťového prenosu

V dnešnej dobe má skoro každá domácnosť prístup na internet. Väčšinou platí určitý poplatok za prístup so špecifickou prenosovou rýchlosťou. Zmluva s poskytovateľom služieb niekedy obsahuje aj ďalšie špecifické parametre.

Preto je dôležité vedieť, ktoré parametre sieťového prenosu sa dajú merať, aby sme mohli overiť ich hodnoty uvedené v SLA. Základným merateľným parametrom je priepustnosť a oneskorenie. Pre bežného užívateľa vlastnosti ako stratovosť paketov alebo zmena poradia nie je podstatná z dôvodu neznalosti sieťového prenosu.

V nasledujúcich podkapitolách si uvedieme jednotlivé merateľné parametre, ktoré sú predmetom našej práce. Taktiež ukážeme konkrétne metodiky, ktoré sú však na veľmi abstraktnej úrovni a zatiaľ používajú použitie jednotlivých protokolov a typov paketov viď [17].

Tvorbe štandardov na meranie parametrov siete sa venuje skupina *IPPM (IP Performance Metrics Working Group)*. Vytvorila množstvo dokumentov RFC, ktoré sa venujú jednotlivým metrikám. Metodiky uvedené v tejto práci budú prevažne od tejto skupiny.

2.1 Jednosmerné oneskorenie (delay)

Medzi základné merateľné parametre patrí oneskorenie paketu. Je to čas potrebný na prenesenie paketu z jedného uzla na druhý. Tento čas môžeme rozdeliť na dve základné zložky, ktorými sú: čas prenosu elektromagnetického signálu a čas potrebný na spracovanie paketu sieťovými zariadeniami. Kde patrí zdržanie v rade, klasifikácia atď.

Tento parameter prenosu je veľmi dôležitý, pretože niektoré aplikácie vyžadujú dodržanie oneskorenia do určitej hraničnej hodnoty. Ak nie je čas dodržaný, aplikácia nemôže korektne pracovať. Môžeme tu zaradiť službu VoIP a podobné typy služieb pracujúcich v reálnom čase. Pomocou oneskorenia sa dá ľahko predikovať zahltenie siete. Pri tomto použití je nutné poznať oneskorenie na nezaťaženej linke, ktorá je predmetom testovania, aby sme mohli hodnoty porovnávať.

Pre meranie času potrebného na prenesenie paketu z jednej koncovej stanice na druhú je potrebné na testovaných zariadeniach v dostatočnej miere synchronizovať čas. Túto službu poskytuje *NTP (Network Time Protocol)*. Ak by sme sa rozhodli odmerať *RTT (Round Trip Time)*, tak sa bez synchronizovaného času zaobídeme. Pri meraní RTT mohol byť paket poslaný dvoma rôznymi cestami, ktoré môžu mať iný čas oneskorenia. Ak by sme teda čas RTT rozdelili na polovice, nedostaneme korektnú hodnotu jednosmerného oneskorenia.

Taktiež platí, že oneskorenie môže byť rôzne z dôvodu rôznej zaťaženia linky oboma smermi [4].

Metodika pre meranie jednosmerného oneskorenia je nasledujúca:

1. Synchronizácia času zdroja a cieľa.
2. Výber IP adresy oboch koncových staníc a určenie veľkosti posielených paketov.
3. Príprava cieľa na príjem paketov.
4. Zdroj začne posielať pakety s časom odoslania a nastavenej veľkosti.
5. Cieľ prijíma pakety. Pre každý odpočítava čas v správe s jeho aktuálnym. S výsledkov rozdielu počíta aritmetický priemer.
6. Výsledný priemer je čas oneskorenia. Čím viac paketov bolo prijatých, tým dostávame presnejšie informácie o oneskorení na linke v danom smere. Meranie pozostávajúce z jedného preneseného paketu môže poskytnúť veľmi nepresné výsledky.

Z metodiky je zrejmé, že aplikácia musí byť typu klient – server. Čo je oproti meraniu RTT pomocou nástroja *Ping* značná nevýhoda. Mnoho služieb používaných v dnešných dátových sieťach zaťažujú linku asymetricky, z čoho usudzujeme, že použiteľnosť zistenej hodnoty RTT je obmedzená. Nezistíme, v akom smere dochádza k radikálnemu oneskoreniu, ktoré býva predmetom testovania. Medzi služby, ktoré zaťažujú linku jedným smerom patrí FTP. Smer, v ktorom sa prenášajú dáta, je viac vyťažený, ako opačný, v ktorom sa posielať TCP potvrdenia.

2.2 Rozptyl oneskorenia (jitter)

Po popísaní oneskorenia paketu v kapitole 2.1 si môžeme zaviesť ďalšiu vlastnosť, ktorá slúži na diagnostiku sieťového prenosu. Je to rozptyl oneskorenia paketov. Často sa označuje ako *jitter*. Tento pojem môže byť nejasný, pretože sa používa vo viacerých oblastiach. V prvom rade sa často používa v zmysle zmeny oneskorenia signálu oproti synchronizačnému.

Motivácia pre určenie rozptylu oneskorenia paketov je podobná ako v kapitole 2.1. Aplikácie pracujúce v reálnom čase sú najcitlivejšie na zmenu oneskorenia paketov. Tento parameter napríklad ovplyvňuje veľkosť prijímacích a odosielených zásobníkov.

Rozptyl oneskorenia je možné merať viacerými spôsobmi. Najabstraktnejšia metodika vyberá z toku paketov pomocou výberovej funkcie dva s už vypočítaným jednosmerným oneskorením. V ďalšom kroku vypočíta ich rozdiel, čo je rozptyl oneskorenia [11].

Celý popis metodiky v krokoch:

1. Synchronizácia času zdroja a cieľa.
2. Výber IP adresy oboch koncových staníc.
3. Príprava cieľa na príjem paketov.
4. Zdroj posiela pakety s časom odoslania.
5. Cieľ prijíma pakety, pokiaľ výberová funkcia neidentifikuje prvý paket. Následne prevedie výpočet jednosmerného oneskorenia.

6. Cieľ prijíma pakety, pokiaľ výberová funkcia neidentifikuje druhý paket. Následne prevedie výpočet jednosmerného oneskorenia.
7. Odpočíta výsledok dvoch získaných jednosmerných oneskorení a získa výsledok rozptylu oneskorenia. Algoritmus pokračuje na 5. bode, kým nezíska dostatočný počet meraní na získanie štatisticky správnych informácií.

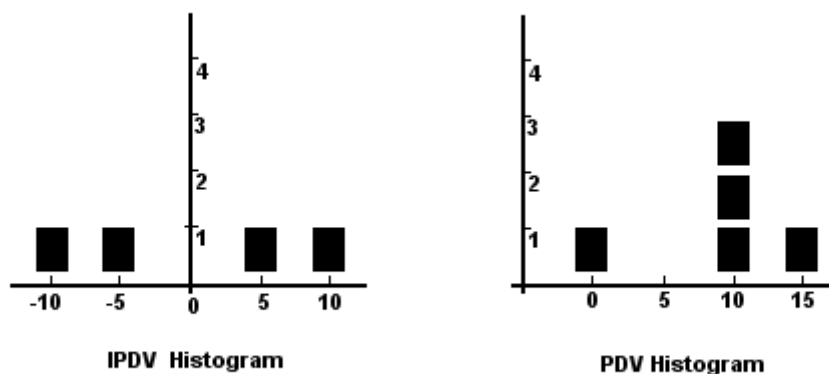
Pre reálnu implementáciu bolo nutné algoritmus upraviť. V ďalších RFC dokumentoch od skupiny IPPM došlo k upresneniu algoritmu. V dnešných implementáciách na zistenie rozptylu oneskorenia figurujú dva algoritmy. Prvý má názov rozptyl oneskorenia paketov *PDV* (*Packet Delay Variation*). Druhý sa volá rozptyl medzi paketového oneskorenia *IPDV* (*Inter Packet Delay Variaton*) viď [8].

Technika IPDV v očíslovanom toku paketov počíta rozptyl oneskorenia zo vzťahu $IPDV(i) = D(i) - D(i - 1)$. $D(i)$ znamená jednosmerné oneskorenie paketu i . Algoritmus PDV v prvom kroku zistí najmenšie jednosmerné oneskorenie z celého toku paketov, ktoré bude odčítat od jednotlivých jednosmerných oneskorení. Vzťah má nasledujúcu podobu: $PDV(i) = D(i) - D(min)$, $D(min)$ označuje najmenšie jednosmerné oneskorenie z celého toku.

Tabuľka 2.1 obsahuje náhodnú vzorku hodnôt jednosmerného oneskorenia. Znak U znamená neznámu hodnotu. Rozptyl oneskorenia IPDV produkuje záporné a neznáme hodnoty, pričom PDV len kladné. Histogram na obrázku 2.1 názorne zobrazuje rozptyl oneskorenia pre jednotlivé algoritmy.

Paket	1	2	3	4	5
Jednosmerné oneskorenie [ms]	20	10	20	25	20
IPDV	U	-10	10	5	-5
PDV	10	0	10	15	10

Tabuľka 2.1: Rozptyl oneskorenia IPDV a PDV.



Obrázek 2.1: Histogram IPDV a PDV.

Je dokázané, že stredná hodnota IPDV sa blíži alebo je rovná nule, viď [8]. Preto, ak chceme zistiť rozptyl oneskorenia, ktorý by sme chceli porovnať s hodnotou uvedenou v SLA, musíme použiť algoritmus PDV.

Nevýhodou algoritmu IPDV je, že v toku paketov s veľkou stratovosťou určí hodnotu rozptylu za neurčitú. Preto je jeho použitie v tomto prípade obmedzené. Aj pre všetky nevýhody IPDV sa používa v protokole RTP v mierne upravenej forme. Je to kvôli schopnosti rýchlejšie reagovať a ovplyvniť veľkosti zásobníkov dát [18].

Pre meranie tohto parametra sieťového prenosu je taktiež nutná dostatočná synchronizácia času. Problémy spojené s meraním sú rovnaké ako pri jednosmernom oneskorení. Pre správnu implementáciu je taktiež nutné použiť architektúru klient – server.

2.3 Priepustnosť (throughput)

Tento parameter sieťového prenosu je veľmi dôležitý a často býva uvedený SLA od poskytovateľa pripojenia. Podľa tohto parametra sa riadi väčšina bežných užívateľov pri výbere pripojenia k internetu. Priepustnosť môžeme definovať ako maximálnu rýchlosť prenosu daného objemu dát, pri ktorej nedochádza k strate alebo zahadzovania rámcov z daného toku za jednotku času [6].

Nástroje, ktoré dokážu odmerať tento parameter prenosu, musia byť architektúry klient – server. Klient generuje a následne posiela dáta na adresu servera, ktorý slúži ako prijímajúca stanica, kde sa vytvárajú štatistiky o prenose.

2.3.1 Meranie TCP priepustnosti podľa RFC 6494

Dokument RFC 6494 [9] od skupiny IPPM uvádza, že pre koncového užívateľa nie je dôležitá informácia o výkone siete na druhej alebo tretej vrstve internetového modelu TCP/IP, ale reálne dosiahnuteľný výkon s použitím transportného protokolu TCP [9]. Je to zapríčinené tým, že väčšina služieb, ktoré využívajú koncoví užívatelia, pracuje s transportným protokolom TCP. Meranie TCP priepustnosti je zložitejší proces. Dokument [9] popisuje celú metodiku merania. Pre našu prácu je dôležité si uvedomiť len základný princíp merania. Metodika pre svoju činnosť potrebuje zistiť viaceré parametre sieťového prenosu:

- Odmerať RTT.
- Zistiť veľkosť PMTU.
- Odmerať priepustnosť siete na L3 vrstve.

Meranie RTT môže byť prevedené spôsobom, že sa odmerá čas medzi poslaním prvého SYN paketu až po príjem nadväzujúcej odpovede SYN+ACK. Priepustnosť siete môžeme odmerať nástrojom Iperf v UDP móde, alebo použiť metodiku z RFC 2544 [7].

Pojem *PMTU* (*Path Maximum Transmission Unit*) znamená veľkosť maximálnej prenosovej jednotky z jednej koncovej stanice na druhú, ktorá nebude v prípade použitia sieťového protokolu IPv4 fragmentovaná. Proces zisťovania PMTU sa nazýva *Path MTU Discovery* a je popísaný v dokumente RFC 1191 [15] pre IPv4 a v RFC 1981 [14] pre IPv6.

Transportný protokol pracuje v troch režimoch: pomalý štart, predchádzanie zahltenia a zotavenie po výpadku. Uvedená metodika sa venuje meraniu výkonu práve v druhej fáze – predchádzanie zahltenia, kedy je prenos vo vyváženom stave. Rýchlosť v tejto fáze dosahuje optimálne hodnoty.

2.4 Strata paketov

Dôležitý parameter sieťového prenosu je strata paketov. Dochádza k nemu najmä pri preťažení linky, keď sa zaplnia vstupné rady sieťových zariadení. K zahadzovaniu dochádza tak tiež pri poškodených rámcoch.

Tento parameter je možné odmerať pomocou transportného protokolu UDP. Protokol TCP implementuje spoľahlivý prenos, takže nie je možné zistiť z vyššej vrstvy modelu OSI, že došlo k strate paketu.

Zistenie straty paketov je dôležité pre správnu funkčnosť sieťového prenosu. Ak prenos prebieha pod transportným protokolom TCP, dochádza automaticky k znovu odoslaniu paketu. Pri strate paketov nesúcich streaming videa alebo iné multimediálne dáta bežiace v reálnom čase pod protokolom UDP nedochádza k znovu odoslaniu. Tým pádom sa zhoršuje kvalita obrazu alebo zvuku. Preto je nutné pri prenosoch v reálnom čase dodržiavať hraničné hodnoty straty a oneskorenia paketov, aby bol obraz a zvuk pre užívateľa dostatočne kvalitný.

Metodika na zistenie straty paketu, prevzatá z [5]:

1. Synchronizácia času zdroja a cieľa.
2. Výber IP adresy oboch koncových staníc.
3. Príprava cieľa na príjem paketov.
4. Zdroj pošle paket s časom odoslania.
5. Cieľ prijme paket a označí ho ako doručený.
6. Ak cieľ neprijme paket v rozumnej časovej perióde, označí ho za stratený. Algoritmus sa opakuje od bodu 4. pre vopred určený počet paketov.

2.5 Zmena poradia paketov

K zmene poradia paketov môže dochádzať napríklad dôsledkom rôznej cesty jednotlivých paketov alebo zmenou poradia v zásobníkoch smerovačov [16]. Multimediálne prenosy v reálnom čase sú príkladom, kedy je podstatné doručovanie datagramov v správnom poradí.

Nástroje, ktoré dokážu identifikovať zmenu poradia paketov, pracujú s transportným protokolom UDP. Použitie TCP nie je možné, tak isto ako v prípade diagnostiky straty paketov, pretože implementuje spoľahlivý prenos.

Aby bolo možné identifikovať zmenu poradia, je nutné, aby každý UDP datagram obsahoval sekvenčné číslo, ktoré jednoznačne určuje poradie. Ak sa prijme paket so sekvenčným číslom menším ako posledný prijatý, došlo k zmene poradia.

2.6 Prínos pre našu prácu

Hlbšia znalosť parametrov siete, ktoré charakterizujú výkonnosť, je pre našu prácu veľmi dôležitá. Testovanie nástrojov na diagnostiku sieťového prenosu vyžaduje vytvoriť metódu, ktorej obsahom bude sledovanie týchto parametrov. Aby sme chápali princípom merania, bolo nutné tieto metodiky uviesť. Taktiež lepšie pochopíme architektúru testovaných nástrojov a ich softvérové nároky, ako je napríklad synchronizácia času. Naštudovanie metodiky malo aj nemalý prínos z dôvodu tvorby metodiky samotnej. Takto sme zistili, akým spôsobom sa tvorí a všetky náležitosti s tým spojené. Znalosti použijeme pri tvorbe metodiky na testovanie nástrojov.

Kapitola 3

Metodika testovania nástrojov

Pred samotným testovaním nástrojov je nutné sa zamyslieť, ako túto činnosť budeme robiť. Je dôležité, aby nástroje boli otestované jednotným spôsobom. Takto zaručíme, že ich budeme môcť medzi sebou jednoznačne porovnať.

Väčšina nástrojov bola vyvinutá za účelom odmerať špecifický parameter sieťového prenosu alebo súbor parametrov. Našou úlohou je zvoliť množinu parametrov, ktoré budú skúmané u každého nástroja. Samotné parametre na diagnostiku sú najpodstatnejšie, je však nutné sa zamyslieť nad vlastnosťami nástrojov, ktoré sú podstatné pri testovaní a použiteľnosti pre bežného užívateľa. Medzi tieto doplnkové vlastnosti patrí napríklad podpora rôznych operačných systémov a sieťového protokolu IPv6.

3.1 Výber sledovaných parametrov sieťového prenosu

Merateľné parametre sieťového prenosu uvedené v kapitole 2 budú predmetom testovania u každého nástroja na diagnostiku sieťovej komunikácie. Tieto parametre patria medzi základné vlastnosti sieťového prenosu. Ak by sme sa rozhodli bližšie špecifikovať výkonové vlastnosti siete, museli by sme testovanie spúšťať pre rôzne veľkosti datagramov, pretože malé datagramy ovplyvňujú výkon procesora a naopak veľké priepustnosť aktívnych sieťových zariadení. Hlavným cieľom našej práce je vytvoriť doporučenie pre bežného užívateľa na testovanie parametrov sieťového prenosu, takže musíme vybrať súbor parametrov, ktoré budú reflektovať požiadavky týchto užívateľov.

Zoznam vybraných parametrov:

- TCP priepustnosť
- UDP priepustnosť
- Oneskorenie
- Rozptyl oneskorenia
- Zmena poradia paketov
- Strata paketov

Do zoznamu boli zaradené aj špecifickejšie parametre ako je rozptyl oneskorenia a zmena poradia paketov. Budú sledované, ak by sa rozhodli testovať parametre siete aj skúsenejší užívatelia.

3.2 Výber ďalších funkcionálnych vlastností

Ďalšie parametre alebo vlastnosti nástrojov, ktoré budeme testovať, by mali obsahovať vlastnosti, ktoré robia nástroj použiteľný na rôznych operačných systémoch, alebo umožňujú použitie sieťového protokolu IPv6. Taktiež je vhodné sledovať, či sa daný nástroj ďalej vyvíja pre podporu do budúcnosti.

Do testu podpory operačných systémov patria: Windows, Linux a FreeBSD. Bežní užívatelia sa budú predovšetkým zaujímať o podporu operačného systému Windows a Linux.

Ďalšou sledovanou vlastnosťou je podpora sieťového protokolu IPv6 a správne pracovanie, ak klientská časť nástroja bude umiestnená za preklad adries.

Zoznam prídavných sledovaných vlastností nástrojov:

- Podpora operačného systému Windows.
- Podpora operačného systému Linux.
- Podpora operačného systému FreeBSD.
- Podpora sieťového protokolu IPv6.
- Klient umiestnený za NAT – om.

3.3 Metodika testovania

Po vyčlenení parametrov a vlastností, ktoré budeme u každého nástroja testovať, je nutné zostaviť systematickú metodiku. Hlavnou úlohou bude jednoznačne určiť, či nástroj umožňuje alebo implementuje testovanie vyčleneného parametru sieťového prenosu, alebo či spĺňa dodatočnú funkcionálnu vlastnosť z kapitoly 3.2. Pri testovaní parametrov z kapitoly 3.1 nie je cieľom metodiky prehlásiť, či sú získané výsledky merania správne. V tomto kroku sa nám jedná len o dokázanie funkcionality.

Testovanie bude prebiehať na skonvergovanej sieti, ktorej topológia nie je známa. Dôležité je poznamenať, že prenos môže ovplyvňovať nesprávne nastavenie sieťových rozhraní, alebo nastavenie operačného systému, ktorý riadi prenos. Čiže predpokladáme správne nastavenie sieťových rozhraní koncových staníc, dostatočný výkon a správne fungovanie softvérového vybavenia.

Metodika je rozdelená na dve časti: prvá má za úlohu otestovať parametre sieťového prenosu z kapitoly 3.1 a druhá požadované funkcionálne vlastnosti z kapitoly 3.2. Keďže parametrov na testovanie je veľa, v metodike nebudú uvedené konkrétne. Overenie podpory sledovaných vlastností nástroja by mohlo byť urobené na základe naštudovania manuálových stránok. To však považujeme za nepostačujúce a vlastnosť budeme reálne testovať. Metodika na meranie parametrov z kapitoly 3.1 má nasledujúci tvar:

1. Výber nástroja, ktorý bude predmetom testovania.
2. Výber *parametru* sieťového prenosu, ktorý chceme testovať.
3. Preskúmanie manuálových stránok nástroja za účelom zistenie, či nástroj implementuje test požadovaného *parametra*. Ak áno, metodika pokračuje na ďalší bod. Ak nie, prehlásime, že nástroj daný *parameter* nie je schopný odmerať.
4. Výber dvoch koncových staníc a určenie IP adries ich sieťových rozhraní.

5. Inštalácia nástroja na zvolených počítačoch.
6. Spustenie nástroja s nastavením, aby vykonal test daného *parametra*.
7. Ak test skončil úspešne a nástroj zobrazil výsledok v ľubovoľnej forme, môžeme prehlásiť, že nástroj implementuje meranie zvoleného *parametra*. Ak výsledok nebol zobrazený, prehlásime, že nástroj testovanie *parametra* neimplementuje.

Testovanie požadovaných vlastností nástrojov z kapitoly 3.2 je rozdelené na viacero častí. Prvá sa zameriava na testovanie podpory operačných systémov, druhá na otestovania sieťového protokolu IPv6 a posledná bude mať za cieľ overiť podporu klienta za prekladom adries.

Testovanie podpory operačného systému:

1. Výber nástroja a operačného systému, ktorý je predmetom testovania.
2. Inštalácia nástroja na vybranom operačnom systéme.
3. Ak sa inštalácia nepodarí, prehlásime nevhodnosť nástroja na daný operačný systém.
4. Odmeranie ľubovoľného parametra sieťového prenosu.
5. Ak sa meranie úspešne dokončilo, nástroj podporuje operačný systém. V opačnom prípade prehlásime nevhodnosť nástroja na daný operačný systém.

Podporu sieťového protokolu IPv6 otestujeme spôsobom, že vyberieme jeden z už otestovaných parametrov, ktorý nástroj podporuje pre IPv4, a následne spustíme test s adresou IPv6. Prípadne použijeme prepínač na voľbu protokolu IPv6. Samozrejme, musíme zabezpečiť IPv6 konektivitu oboch testovacích staníc.

Podporu umiestnenia klienta za preklad adries otestujeme spôsobom, že klientskú časť nástroja umiestnime za NAT a spustíme test pre už overený parameter sieťového prenosu.

3.4 Zhrnutie

Vytvorená metodika nám poskytuje ucelenú formu, pomocou ktorej bude možné nástroje systematickým spôsobom testovať. Tvorba metodiky nás donútila k presnejšiemu určeniu cieľa, ktorý obsahuje výber parametrov sieťového prenosu, ktoré budeme sledovať. V kapitole 4 budeme postupne popisovať vybrané nástroje a testovať ich vlastnosti podľa uvedenej metodiky.

Kapitola 4

Prehľad testovaných nástrojov

V tejto kapitole sa budeme venovať popisu vybraných softvérových nástrojov, ktoré nám umožňujú merať rôzne parametre siete. Boli vybraté tie najznámejšie utility, ktoré boli v čase tvorby tejto práce dostupné.

U každého nástroja budú uvedené softvérové nároky, popis parametrov a ukážka vybraných testov. Keďže sa nezameriavame na špecifické testovanie každého nástroja, popísaný obsah bude predmetom nami sledovaných parametrov uvedených v kapitole 3. Z tejto kapitoly budeme aplikovať aj metodiku testovania. Pri každom nástroji bude uvedená záverečná kapitola s tabuľkou obsahujúcou zoznam parametrov, ktoré sa dajú s nástrojom merať.

Na záver v kapitole 4.8 bude vyhodnotenie všetkých testovaných nástrojov. Sledované parametre budú uvedené v tabuľkách pre prehľadnejší výber správneho nástroja na dané testovanie.

4.1 Nástroj Iperf

Prvý z testovaných nástrojov je Iperf. Bol vyvinutý ako moderná alternatíva pre meranie maximálnej priepustnosti pod transportným protokolom TCP a UDP. Pri spustení v UDP móde je schopný odmerať rozptyl oneskorenia, stratu a výmenu poradia paketov.

K tejto konzolovej aplikácii bolo vytvorené grafické rozhranie s názvom *Jperf*¹ implementované v jazyku Java. Pre účely testovania sme zvolili verziu 2.0.5².

Existuje nová implementácia s názvom *Iperf3*, ktorá však nie je spätne kompatibilná. Jej cieľom je dosiahnuť jednoduchú implementáciu so zameraním na knižnice tak, aby ju mohli využívať iné programy.

4.1.1 Architektúra

Tento program bol implementovaný princípom klient – server v jednom spustiteľnom súbore. Podľa parametra musíme špecifikovať, ktorý proces chceme spustiť. Klient posiela dáta a server ich prijíma, tieto úlohy sa však môžu so špecifikovaním parametrov vymeniť. Server je implementovaný konkurentným spôsobom, takže obslúži viacero klientov súčasne.

Implementácia prebehla v jazyku C a C++ s využitím rozhrania BSD soketov. Pre potreby projektu Iperf bola vyvinutá knižnica DAST v jazyku C++ vid' [12].

¹Dostupná na <http://code.google.com/p/xjperf/>

²Dostupná na <http://sourceforge.net/projects/iperf/>

4.1.2 Softvérové nároky

Iperf je multiplatformová aplikácia spustiteľná na unixových systémoch FreeBSD, Solaris, Linux a MacOS X. Taktiež je ju možné používať na Windows.

Vo väčšine distribúciach Linuxu je obsiahnutý v balíčkových repozitároch, čo značne uľahčuje inštaláciu a dostupnosť. Pre priamu kompiláciu zo zdrojových textov je nutné mať nainštalovaný gcc a knižnicu glibc. Grafická nadstavba Jperf je implementovaná v jazyku Java. Pre jej beh je nutné zabezpečiť virtuálne prostredie Java aplikácii (JRE).

4.1.3 Popis vybraných parametrov

Jednotlivé parametre nástroja sú uvedené v tabuľke 4.1. Tabuľka je rozdelená do niekoľkých častí, aby združila sémanticky podobné prepínače. Iperf je architektúry klient – server implementovaný v jednom spustiteľnom súbore. Preto je nutné pomocou prepínačov vybrať správny mód.

Parameter	Popis
Základné	
-h, –help	výpis nápovedy
-v, –version	výpis verzie
-s	mód servera
-c <doména>	mód klienta, doména špecifikuje adresu alebo doménové meno, na ktorom beží server
-D, –daemon	spustí server na pozadí
Nastavenie spojenia	
-p, –port <číslo>	špecifikácia portu
-V	použije protokol IPv6
-u, –udp	použije UDP protokol namiesto TCP
-b, –bandwidth <číslo>[KM]	použitie iba s -u, bude testovať do maximálnej priepustnosti (implicitne 1 Mbit/s)
-P, –parallel <číslo>	vytvorí paralelné testovacie spojenia
Špecifikácia výpisov	
-f, –format [kmKM]	nastaví jednotky výpisu: Kbits, Mbits, KBytes, MBytes
-i, –interval <číslo>	periodické výpisy výsledkov v sekundách
-m, –print_mss	vypíše TCP maximum segment size – MSS a MTU
Nastavenie dĺžky trvania testu	
-t, –time <číslo>	dĺžka prenosu dát v sekundách (implicitne 10 s)
-n, –num <číslo>[KM]	veľkosť bajtov na prenos (namiesto parametra -t)
Nastavenie prenášaných dát	
-F, –fileinput <cesta>	prenášané dáta zoberie zo súboru
-I, –stdin	prenášané dáta zoberie zo štandardného vstupu
Obojsmerné testovanie	
-d, –dualtest	obojsmerný test v súčasnom čase
-r, –tradeoff	obojsmerný test individuálne

Tabuľka 4.1: Vybrané prepínače nástroja Iperf.

Spustenie klientskej a serverovej časti:

```
$ iperf -c <adresa>
$ iperf -s
```

4.1.4 Ukážka testov

Pred samotným začatím testovania je potrebné spomenúť, že Iperf implicitne spúšťa testovanie TCP priepustnosti na desať sekúnd.

Obrázok 4.1 ukazuje spustenie a výpis servera bez pridaných prepínačov. Na ukážke je naviazané jedno spojenie s identifikačným číslom 4 na test TCP priepustnosti. Výstup poskytuje informácie o časovej dĺžke testu, množstve prenesených dát a nameranej priepustnosti. V tomto prípade klientská strana poskytuje tie isté informácie.

```
xloff00@merlin: ~/bp$ iperf -s
-----
Server listening on TCP port 5001
TCP window size: 64.0 KByte (default)
-----
[  4] local 147.229.176.19 port 5001 connected with 147.229.221.171 port 56372
[ ID] Interval      Transfer    Bandwidth
[  4] 0.0-10.2 sec  115 MBytes  94.2 Mbits/sec
```

Obrázek 4.1: Spustenie a výpis Iperf servera.

Pre spustenie klienta je potrebné vedieť doménové meno alebo IP adresu hostiteľa, na ktorom beží Iperf server. Zadáva sa ako parameter prepínača `-c`. Obrázok 4.2 zobrazuje spustenie klienta s parametrom, ktorý spôsobí periodický výpis nameraných dát. Ako je z obrázku vidieť, bol spustený test na TCP priepustnosť. Keďže sme nešpecifikovali dĺžku trvania, test bol spustený na desať sekúnd s periodickými výpismi každé 2 sekundy.

```
[pavol@vaio bp]$ iperf -c merlin.fit.vutbr.cz -i 2
-----
Client connecting to merlin.fit.vutbr.cz, TCP port 5001
TCP window size: 22.9 KByte (default)
-----
[  3] local 147.229.221.171 port 56533 connected with 147.229.176.19 port 5001
[ ID] Interval      Transfer    Bandwidth
[  3] 0.0- 2.0 sec  25.8 MBytes  108 Mbits/sec
[  3] 2.0- 4.0 sec  22.4 MBytes  93.8 Mbits/sec
[  3] 4.0- 6.0 sec  22.5 MBytes  94.4 Mbits/sec
[  3] 6.0- 8.0 sec  22.4 MBytes  93.8 Mbits/sec
[  3] 8.0-10.0 sec  22.5 MBytes  94.4 Mbits/sec
[  3] 0.0-10.1 sec  116 MBytes  96.2 Mbits/sec
[pavol@vaio bp]$
```

Obrázek 4.2: Spustenie a výpis Iperf klienta pre testovanie TCP priepustnosti.

Pri UDP testovaní musíme špecifikovať maximálnu priepustnosť, ktorú chceme testom dosiahnuť, pretože implicitne sa zahájí testovanie do hodnoty 1 Mbps. Pomocou parametra `-b` môžeme špecifikovať hodnotu maximálnej priepustnosti, ktorú chceme otestovať. Výstup programu pre UDP testovanie nám poskytuje tie isté informácie ako TCP test, ale je doplnený o informácie zaslané serverom. Tie zahŕňajú počet stratených paketov k celkovému počtu, rozptyl oneskorenia a počet paketov prijatých v nesprávnom poradí.

Nasledujúce spustenie klienta pre UDP testovanie ukazuje obrázok 4.3. V tomto meraní sme sa snažili overiť, či je možné dosiahnuť na linke priepustnosť 20 Mbps. Ako demonštruje výstup programu, linka nie je schopná prenosu na tejto rýchlosti. Ďalej je možné dedukovať, že dochádzalo k výraznej strate paketov. Nastáva tu situácia, v ktorej sme špecifikovali hornú hranicu testovanej priepustnosti väčšiu, ako je reálna. Z ukážky vidieť, že klient ukazuje priepustnosť 19.7 Mbps a server 3.27 Mbps. Hodnoty sa výrazne líšia, presnejšia hodnota nameranej priepustnosti je samozrejme od strany servera, pretože ten má informácie o prijatých paketoch, ktoré skutočne prišli. Z tohoto chovania môžeme dedukovať, že server neposkytol informácie o prenesených paketoch klientovi po riadiacom kanáli po skončení testu.

```
[pavol@vaio ~]$ iperf -c merlin.fit.vutbr.cz -u -b 20M
-----
Client connecting to merlin.fit.vutbr.cz, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[  3] local 192.168.1.4 port 47343 connected with 147.229.176.19 port 5001
[ ID] Interval      Transfer    Bandwidth
[  3] 0.0-10.0 sec  23.5 MBytes 19.7 Mbits/sec
[  3] Sent 16750 datagrams
[  3] Server Report:
[  3] 0.0-10.3 sec  3.99 MBytes 3.27 Mbits/sec 15.066 ms 13898/16746 (83%)
[  3] 0.0-10.3 sec  1 datagrams received out-of-order
[pavol@vaio ~]$
```

Obrázek 4.3: Spustenie a výpis Iperf klienta pre UDP test.

4.1.5 Nekorektné správanie

Na obrázku 4.3 z výstupu programu Iperf pri UDP testovaní je možné vidieť, že bol prijatý jeden paket v nesprávnom poradí. Toto chybové hlásenie sa vyskytuje vždy, ak zahájime UDP test s priepustnosťou väčšou ako je 3 Mbit/s – parameter `-b 3M`. Niekedy sa nezobrazí výstup s informáciami o UDP priepustnosti, ktoré posielala server klientovi, takže zobrazená priepustnosť je iba z klientovej strany, ktorá môže obsahovať nesprávne informácie.

4.1.6 Zhodnotenie

Pre meranie základných parametrov siete ako je priepustnosť, rozptyl oneskorenia a strata paketov je Iperf vhodný nástroj. Poskytuje veľmi jednoduché ovládanie a je dostupný pre väčšinu dnes používaných systémov.

Ako ďalšie kladne hodnotiace fakty musíme spomenúť použitie obojstranných testov a možnosť emulovať paralelne testovacie spojenia medzi jednou inštanciou klienta a servera. Taktiež schopnosť serveru obslúžiť súčasne viacerých klientov (konkurentný server). Dôležitým faktom je funkčnosť protokolu IPv6 pre budúce použitie tohto nástroja.

Za výrazný nedostatok považujeme fakt, že server pracuje iba v jednom zo zvolených módov a to TCP alebo UDP. Pre testovanie z jednej stanice pomocou TCP a UDP je potreba spustiť dve rôzne inštancie servera.

Tabuľka 4.2 poskytuje súhrnné informácie o možnosti použitia nástroja Iperf na meranie parametrov sieťového prenosu. Ďalšia tabuľka 4.3 obsahuje informácie o dodatočných vlastnostiach tohto nástroja.

TCP priep.	UDP priep.	Oneskorenie	Jitter	Zmena poradia	Strata
✓	✓		✓		✓

Tabulka 4.2: Merateľné parametre siete nástroja Iperf.

Linux	Windows	FreeBSD	IPv6	NAT
✓	✓		✓	✓

Tabulka 4.3: Ďalšie vlastnosti nástroja Iperf.

4.2 Nástroj Netperf

Ďalším vybraným nástrojom je Netperf. Je to testovací nástroj na meranie rôznych aspektov sieťového výkonu [13]. Primárne určený na jednosmerné testovanie prenosu pod protokolom TCP, UDP a SCTP. Oproti aplikácii Iperf je s ním možné testovať rôzne špecifickejšie vlastnosti sieťového prenosu. Je implementovaný ako konzolová aplikácia. Pre účely testovania bola vybraná verzia 2.6.0³.

4.2.1 Architektúra

Architektúra nástroja Netperf je založená na modeli klient – server. Nástroj je rozdelený do dvoch spustiteľných programov **netperf** a **netserver**, ktorý reprezentuje serverovú časť.

Ako u nástroja Iperf, klient generuje dátový prenos a server prijíma na danom porte. Každé spojenie medzi klientom a serverom obsahuje dva komunikačné kanály, na jednom sa prenášajú riadiace informácie a druhý slúži na prenos testovaných dát. Implementácia prebehla v jazyku C s použitím rozhrania BSD soketov.

4.2.2 Softvérové nároky

Tento nástroj je taktiež multiplatformová aplikácia spustiteľná na pomerne všetkých dostupných operačných systémoch. Patria tu unixové operačné systémy, Linux a FreeBSD, ale aj operačný systém Windows.

³Dostupná na <http://www.netperf.org/netperf/>

4.2.3 Popis vybraných parametrov

Tabuľka 4.4 obsahuje vybrané parametre, ktoré sú predmetom testovania nami skúmaných parametrov sieťového prenosu. Taktiež sú zlúčené podľa sémanticky podobných vlastností.

Parameter	Popis
Základné	
-h	výpis nápovedy
-V	výpis verzie
-D	spustí server na popredí
Nastavenie spojenia	
-p <číslo>	špecifikácia čísla portu
-4	použije IPv4 adresu, nastaví AF_INET
-6	použije IPv6 adresu, nastaví AF_INET6
-t <typ>	typ transportného protokolu, pre UDP hodnota UDP_STREAM, pre TCP TCP_STREAM (implicitné)
Špecifikácia výpisov	
-f <[GMKgmK]>	nastaví jednotky výpisu, veľké písmena umocnia jednotky na druhú a malé na desiatu
Nastavenie dĺžky trvania testu	
-l <číslo>	dĺžka prenosu dát v sekundách (implicitne 10 s)
Nastavenie prenášaných dát	
-F <cesta>	prenášane dáta zoberie zo súboru
Test špecifické	
-m <číslo>	nastavenie veľkosti poľa predávaného funkcií send, nemusí priamo ovplyvniť veľkosť posiadaného paketu, použitie pri UDP_STREAM
-s <číslo>	nastaví veľkosť prijímaného a odosiadaného poľa na strane klienta.
-S <číslo>	nastaví veľkosť prijímaného a odosiadaného poľa na strane servera.

Tabuľka 4.4: Vybrané prepínače nástroja Netperf.

Parametre sa vo všeobecnosti delia na globálne a test špecifické. Test špecifické musia byť oddelené dvoma znakmi "-". Uvádžame príkaz na základné spustenie klienta a servera:

```
$ netperf -H <adresa,protokol> -p <port> <globálne> -- <test špecifické>
$ netserver <adresa,protokol> -p <port> -D
```

Protokol sa špecifikuje číslom 4 pre IPv4 (AF_INET) a 6 pre IPv6 (AF_INET6). Je to nepovinná položka. Ak sa nezadá, použije sa AF_UNSPEC.

4.2.4 Ukážka testov

V tejto sekcii sa pozrieme na možné testy s nástrojom Netperf. Pri spustení bez parametrov upravujúcich testovanie sa zahájí implicitne prenos nad transportným protokolom TCP na dĺžku trvania desať sekúnd.

Pre úspešné testovanie je potrebné zabezpečiť beh inštancie **netserver** na hostiteľovi, na ktorý sa budú generovať dáta z klientskej časti **netperf**. Aplikácia **netserver** nevypisuje

žiadne výpisy. Pri štandardnom spustení sa spustí na pozadí. Toto chovanie môžeme potlačiť parametrom `-D`. Keďže serverová aplikácia neposkytuje žiadne samostatne výpisy, nebude uvádzať terminálové obrázky z jej behu.

Klientská časť vyžaduje jeden povinný prepínač `-H`, ktorý vyžaduje parameter doménové meno, alebo IP adresu hostiteľa, na ktorom beží server.

Prvý test môžeme vidieť na obrázku 4.4, na ktorom je uvedený výstup pri testovaní priepustnosti pod transportným protokolom TCP. Výstup aplikácie pri tejto konfigurácii poskytuje informácie o maximálne dosiahnutej priepustnosti, dĺžke trvaní testu a veľkosti zásobníkov pre príjem a odoslanie dát. Výpis taktiež poskytuje informáciu o veľkosti posielanej správy. Z obrázka 4.4 vyplýva, že bola dosiahnutá priepustnosť 93,7 Mbit/s.

```
[pavol@vaio bp]$ netperf -H 147.229.208.192 -l 90
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 147.229.208.192 () port 0 AF_INET : demo
enable_enobufs failed; getprotobyname
Recv Send Send
Socket Socket Message Elapsed
Size Size Size Time Throughput
bytes bytes bytes secs. 10^6bits/sec
87380 16384 16384 90.25 93.70
```

Obrázek 4.4: Spustenie a výpis nástroja Netperf pre TCP_STREAM.

Následujúci test z obrázku 4.5 demonštruje spustenie klienta, ktorý zaháji testovanie pod transportným protokolom UDP. Táto možnosť bola dosiahnutá prepínačom `-t UDP_STREAM`. Nápadne môžeme vidieť použitie IPv6 adresy hostiteľa, na ktorej beží serverová časť. Takto sa použil sieťový protokol IPv6 bez ďalších prídavných prepínačov.

Výstup nám poskytuje informácie o dosiahnutej priepustnosti, dĺžke trvania, ale aj počet chybné a správne prenesených správ. Ako demonštruje obrázok, posledné dva riadky obsahujú namerané výsledky, ktoré nie sú totožné. Posledný riadok je výstup nameraných údajov `netserveru`, ktorý po ukončení testu poslal dáta klientovi. Údaje nie sú totožné, pretože bolo odoslané väčšie množstvo dát, aké bolo prijaté serverom. Toto je typický fakt pri testovaní pod transportným protokolom UDP, dáta je možné rýchlejšie odoslať, ale nie všetky budú korektne prijaté. Môžeme vidieť, že počet prijatých správ na servery bol menší ako odoslaných.

```
[pavol@vaio bp]$ netperf -H 2001:67c:1220:c1a0:224:7eff:feda:2130 -t UDP_STREAM
MIGRATED UDP STREAM TEST from ::0 (::) port 0 AF_INET6 to 2001:67c:1220:c1a0:224:7eff:feda:2130 () port 0 AF_INET6 : demo
enable_enobufs failed; getprotobyname
Socket Message Elapsed Messages
Size Size Time Okay Errors Throughput
bytes bytes secs # # 10^6bits/sec
212992 65507 10.01 1802 0 94.35
229376 10.01 1782 93.30
```

Obrázek 4.5: Spustenie a výpis nástroja Netperf pre UDP_STREAM.

4.2.5 Nekorektné správanie

S nástrojom Netperf sa nám nepodarilo previesť testovania pod protokolom UDP a spojazdiť komunikáciu so sieťovým protokolom IPv6. Obe tieto vlastnosti sú v nástroji implementované a uvedené v manuálových stránkach. Testovanie pod transportným protokolom UDP sa podarilo spojazdiť pri použití rovnakých operačných systémoch na koncových staniciach. Utilita však nechcela nadviazať spojenie pomocou IPv4. Preto meranie UDP priepustnosti nepovažujeme za správne fungujúce.

4.2.6 Zhodnotenie

Medzi hlavné výhody tohto konzolového nástroja považujeme úplne ovládanie servera z aplikácie klienta. To umožňuje použitie rozličných parametrov. Pre plnohodnotné testovanie stačí spustenie jednej inštancie servera na vzdialenom hostiteľovi. Toto chovanie je veľmi vhodné, pretože nevyžaduje opakované spustenie serverovej časti pri zmene transportného protokolu. Aplikácia taktiež poskytuje veľmi zrozumiteľné výstupy, ktoré uvádzajú tie najdôležitejšie fakty.

Za nedostatok môžeme považovať nemožnosť získať informácie o zmene poradia, strate paketov a hodnote jitter pri testovaní pod transportným protokolom UDP. Pomocou filtrácie a analýzy sieťového toku bolo zistené, že parameter `-m` nastaví veľkosť dát posielených v UDP rámci, avšak nie celkovú veľkosť paketu. Ak celková veľkosť presiahne MTU, je správa fragmentovaná. Toto správanie uvedeného prepínača musíme pri testovaní brať v úvahu. Tabuľky 4.5 a 4.6 obsahujú súhrnné informácie o možnostiach testovania pomocou tohto nástroja.

TCP priep.	UDP priep.	Oneskorenie	Jitter	Zmena poradia	Strata
✓					

Tabuľka 4.5: Merateľné parametre siete nástroja Netperf.

Linux	Windows	FreeBSD	IPv6	NAT
✓	✓	✓		✓

Tabuľka 4.6: Ďalšie vlastnosti nástroja Netperf.

4.3 Nástroj BWCTL

BWCTL je terminálová aplikácia, ktorá zabezpečuje meranie priepustnosti prostredníctvom iných nástrojov. Pre svoju funkčnosť potrebuje niektorý z nástrojov na meranie sieťových parametrov. Medzi tieto nástroje patrí Iperf, Thrulay a Nuttcp. Kombináciou týchto aplikácií pri testovaní je schopný BWCTL odmerať široké spektrum sieťových parametrov. Samotná aplikácia neimplementuje žiadne testovacie techniky, avšak externe spúšťa uvedené nástroje. Takto docielime, že pomocou jednej bežiacej utility na servery budeme schopní testovať pomocou troch rozličných nástrojov.

Úlohou BWCTL bolo taktiež zaviesť prvky, ktoré konkurenčné nástroje neobsahovali. Jedná sa o podporu plánovania a zabezpečenia. To však nie je predmetom našej práce. Pre testovanie sme použili verziu 1.4⁴.

4.3.1 Architektúra

Taktiež sa jedná o aplikáciu klient–server, takže pre testovanie potrebuje spustený ďalší proces na vzdialenom stroji. Nástroj sa delí na dve samostatne spustiteľné aplikácie. Prvá slúži na inicializáciu a nastavovanie parametrov testovania, je to klient `bwctl`. Druhá slúži ako démon bežiaci na vzdialenom hostiteľovi. Jej názov je `bwctld`.

⁴Dostupná na <http://www.internet2.edu/performance/bwctl/>

Významnou funkciou je schopnosť spustiť testovanie z klienta tak, že nebude jednou z koncových staníc. Tento spôsob nám umožňuje testovanie medzi sieťovými uzlami, na ktoré nemáme prístup.

4.3.2 Softvérové nároky

Nároky na softvér tohto nástroja sú obsiahlejšie, pretože pre svoju funkčnosť potrebuje iné aplikácie. Na stanici, na ktorej budeme chcieť úspešne testovať, musí byť nainštalovaný jeden z nástrojov Iperf, Thrulay alebo Nuttcp. Pre úspešné testovanie ďalej vyžaduje, aby koncové stanice mali synchronizovaný čas pomocou NTP protokolu, čiže spustený NTP démon. Podpora NTP sa dá potlačiť parametrom `-a`, ale toto nastavenie nezaručuje správne výsledky testov. Aplikácia taktiež môže skončiť s chybovým návratovým kódom.

Samotný nástroj bol úspešne testovaný na linuxových systémoch s jadrom verzie 2.4, 2.6 a FreeBSD 4.X a 5.X. Na systéme Solaris sa nedá úspešne skompilovať Thurlay, takže jeho použitie je obmedzené. Nástroj nie je dostupný ako balíček v linuxových distribúciách, preto je potrebné kompilovanie zo zdrojových textov. Kompilácia vyžaduje GNU Make.

4.3.3 Popis vybraných parametrov

Väčšina podporovaných prepínačov je prevzatá z nástroja Iperf, ktoré sú uvedené v tabuľke 4.1. Je dôležité si overiť sémantiku daných prepínačov, pretože sa môžu líšiť. Uvedená tabuľka 4.7 obsahuje vlastné prepínače tohto nástroja. Jedná sa hlavne o viacnásobné spustenie testovania. Démon na strane servera prijíma len určité parametre, viď tabuľka 4.7 a [1].

Parameter	Popis
Základné	
-h	výpis nápovedy
-V	výpis verzie
-a syncfuzz	povolí testovanie bez NTP démona
-T <program>	určí nástroj pre testovanie, možné voľby sú Iperf, Nuttcp, Thrulay
-f [kmKM]	nastaví jednotky výpisu: Kbits, Mbits, KBytes, MBytes
Nastavenie spojenia	
-4	použije IPv4 adresu, (implicitne preferuje IPv6)
-6	vynúti použitie IPv6 adresy
-c <adresa>	adresa alebo doménové meno stanice, ktorá bude prijímať dáta
-s <adresa>	adresa alebo doménové meno stanice, ktorá bude posilať dáta
Riadenie testovania	
-I <číslo>	časový interval v sekundách v ktorom bude periodicky spúšťať testovanie
-n <číslo>	povolí spustenie určitého počtu testov (použitie s -I)
Parametre pre bwctld	
-c <adresa>	adresár s konfiguračnými súbormi
-Z	spustí server na popredí

Tabuľka 4.7: Vybrané parametre nástroja BWCTL.

Príkaz na spustenie klienta a démona na popredí:

```
$ bwctl -c <adresa>
$ bwctld -Z
```

4.3.4 Ukážka testov

Testovanie pomocou BWCTL môže byť značne jednoduché, pretože združuje viacero nástrojov, ktoré sú ovládané tým istým rozhraním. Nasledujúci test z obrázka 4.6 ukazuje meranie TCP priepustnosti. Ako je vidieť, ak pomocou parametra `-T` nešpecifikujeme použitý nástroj, spustí sa meranie pomocou nástroja Iperf pre TCP test s dĺžkou trvania desať sekúnd. Takto spustený test s parametrom `-c` spôsobí, že klient bude generovať a následne posielat dáta na server, kde beží `bwctld` démon, ktorý dáta prijme. Ak zvolíme parameter `-s`, bude prenos dát prebiehať v opačnom smere.

```
[pavol@vaio bp]$ bwctl -c eva.fit.vutbr.cz -f k
bwctl: Using tool: iperf
bwctl: 16 seconds until test results available

RECEIVER START
bwctl: exec_line: iperf -B 147.229.176.14 -s -f k -m -p 5001 -t 10
bwctl: start_tool: 3562424227.091049
-----
Server listening on TCP port 5001
Binding to local address 147.229.176.14
TCP window size: 64.0 KByte (default)
-----
[ 12] local 147.229.176.14 port 5001 connected with 147.229.221.171 port 37152
[ ID] Interval      Transfer    Bandwidth
[ 12] 0.0-10.1 sec  115712 KBytes  94116 Kbits/sec
[ 12] MSS size 1448 bytes (MTU 1500 bytes, ethernet)
bwctl: stop_exec: 3562424241.295047
RECEIVER END
```

Obrázek 4.6: Ukážka testu TCP priepustnosti s použitím nástroja BWCTL.

Využitie schopností nástroja BWCTL ukazuje test na obrázku 4.7. Každých desať sekúnd spustí testovanie s dĺžkou trvania 2 sekundy pomocou nástroja Iperf pre meranie maximálnej TCP priepustnosti.

```
eva ~/bp/bwctl/eva/bwctl> ./bwctl -c 147.229.221.171 -f m -I 10 -t 2
bwctl: Using tool: iperf
bwctl: SessionRequest: 147.229.221.171 busy. (Try -L flag)
bwctl: 9 seconds until next testing period
bwctl: 8 seconds until test results available

RECEIVER START
bwctl: exec_line: iperf -B 147.229.221.171 -s -f m -m -p 5001 -t 2
bwctl: start_tool: 3562438577.511038
-----
Server listening on TCP port 5001
Binding to local address 147.229.221.171
TCP window size: 0.08 MByte (default)
-----
[ 12] local 147.229.221.171 port 5001 connected with 147.229.176.14 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 12] 0.0- 2.1 sec   23.0 MBytes  94.1 Mbits/sec
[ 12] MSS size 1448 bytes (MTU 1500 bytes, ethernet)
bwctl: stop_exec: 3562438583.684043
RECEIVER END
bwctl: 8 seconds until test results available
```

Obrázek 4.7: Ukážka opakovaného spustenia testu pomocou BWCTL.

4.3.5 Nekorektné správanie

Pri testovaní sa vyskytovali problémy pri spustení v UDP móde. Väčšinou nemohlo byť naviazané spojenie. S prepínačom `-I` nastávala situácia, že klient v niektorých periódach nemohol naviazať spojenie s démonom na vzdialenej stanici. Vypisovaná hláška bola: `SessionRequest: host busy. (Try -L flag)`.

4.3.6 Zhodnotenie

Ako už bolo spomenuté tento nástroj pre testovanie využíva iné programy, takže nám neprináša žiadne vylepšenia a k výsledkom by sme sa dopracovali použitím utilít, ktoré využíva.

Nevýhodou tohto nástroja je komplikovaná inštalácia, ktorá vyžaduje oddelené inštalovanie ďalších nástrojov. Ďalší záporný fakt je použitie démona NTP.

Ak sa rozhodneme pre dlhodobejšie a obsiahlejšie testovanie, tento nástroj bude správnu voľbou, pretože poskytne jednotné rozhranie pre viacero nástrojov, ktoré by sme museli obsluhovať samostatne.

Tabuľky, ktoré obsahujú súhrnné informácie o možnostiach testovania parametrov siete, neuvádzame, pretože tento nástroj len spúšťa ďalšie aplikácie. Keďže každý z nástrojov bude otestovaný samostatne. Tabuľky uvedieme v príslušnej kapitole.

4.4 Nástroj OWAMP

Tento nástroj neslúži na meranie priepustnosti siete. Implementuje protokol *OWAMP* (*One Way Active Measurement Protocol*), ktorý slúži na meranie času prenosu paketu z jedného hostiteľa na druhého (jednosmerné oneskorenie). Je podobný aplikácii Ping, ktorá sa zameriava na vyhodnotenie času RTT. RTT môžeme chápať ako dvojnásobnú hodnotu času prenosu paketu z jedného hostiteľa na druhého. Týmto dospejeme k nepresnej hodnote a tento spôsob nie je správny. Kvôli tomu vznikol protokol OWAMP [19] a aplikácia s rovnakým názvom OWAMP ho implementuje.

Výstup tejto konzolovej aplikácie nám poskytuje informácie o časoch potrebných na prenesenie paketu z jedného hostiteľa na druhého v oboch smeroch. Pre účely testovania bola použitá verzia 3.3⁵.

4.4.1 Architektúra

Nástroj taktiež vychádza z modelu klient–server. Aplikácia je rozdelená do dvoch samostatne spustiteľných programov. Klientská utilita má názov `owping` a server `owampd`, ktorý sa normálne spúšťa na pozadí a poskytuje minimálne množstvo výpisov.

4.4.2 Softvérové nároky

Má podobné nároky na softvér ako aplikácia BWCTL. Oba sú vyvíjané rovnakou organizáciou. Vyžaduje synchronizovaný čas pomocou NTP protokolu, čiže spustený príslušný démon. Oproti aplikácii BWCTL sa táto požiadavka nedá potlačiť prepínačom.

Podporované operačné systémy sú FreeBSD, MacOS X, Linux a Solaris. Nástroj je dostupný ako balíček pre niektoré distribúcie Linuxu. Ak je potrebná kompilácia zo zdrojových textov, vyžaduje program GNU Make.

⁵Dostupná na <http://www.internet2.edu/performance/owamp/index.html>

4.4.3 Popis vybraných parametrov

Podobne ako BWCTL aj tento nástroj obsahuje možnosti zabezpečenia a autentizácie. To však nie je predmetom našej práce, tak sa zameriame len na prepínače súvisiace s testovaním. Tabuľka 4.8 obsahuje vybrané prepínače programov `owping` a `owampd`.

Parameter	Popis
Prepínače pre <code>owping</code>	
-h	výpis nápovedy
-c <číslo>	počet testovacích paketov, (implicitne 100)
-f	prevedie jednosmerný test smerom od vzdialeného hostiteľa
-t	prevedie jednosmerný test smerom k vzdialenému hostiteľovi
-s <číslo>	veľkosť paketu
-4	použije IPv4 protokol, (implicitne preferuje IPv6)
-6	použije IPv6 protokol
Prepínače pre <code>owampd</code>	
-Z	spustenie na popredí
-c <adresa>	cesta k priečinku obsahujúcemu konfiguračné súbory (<code>owampd.conf</code> , <code>owampd.limits</code> , ak sa nezadá berie aktuálny pracovný adresár)
-S <adresa>:port	určí adresu a port na ktorom bude prijímať spojenia

Tabuľka 4.8: Vybrané parametre nástroja OWAMP.

Klient `owping` vyžaduje jeden povinný parameter – adresu vzdialeného počítača s bežiacim procesom `owampd`. Spustenie základného testu môže vyzeráť nasledovne:

```
$ owping <adresa>:<port>
$ owampd -S <adresa>:<port> -Z
```

V prípade použitia IPv6 je potrebné uviesť port v hranatých zátvorkách. Parameter `-Z` spôsobí spustenie démona na popredí. Démon je možné spustiť dvoma spôsobmi. Prvý vyžaduje konfiguračný súbor. Cesta k nemu sa zadáva prepínačom `-c`. Druhý spôsob spustenia sme uviedli v ukážke.

V prípade spustenia s konfiguračným súborom je nutné upraviť jeden riadok v súbore `owampd.conf`. Zadáme adresu a port, na ktorom bude prijímať spojenia. Protokol OWAMP má rezervovaný port 861, ktorý vyžaduje práva super užívateľa. Pokiaľ chceme toto správanie obísť, musíme zvoliť iné číslo.

```
srcnode localhost:861
srcnode eva.fit.vutbr.cz:8611
```

4.4.4 Ukážka testov

Na nasledujúcom teste si ukážeme, ako môžeme odmerať jednosmerné latencie na ceste k vzdialenej sieťovej stanici. Obrázok 4.8 obsahuje výstup z aplikácie `owping`. Pomocou parametra `-c` bol upravený počet testovacích paketov z implicitnej hodnoty 100 na 110. Výstup programu obsahuje informácie o jednosmernom oneskorení v oboch smeroch medzi

testovanými stanicami. Medzi ďalšie informácie, ktoré poskytuje, patrí počet skokov, rozptyl oneskorenia a strata paketov.

```
[pavol@vaio conf]$ owping -c 110 ps1.ochep.ou.edu:861
Approximately 14.8 seconds until results available

--- owping statistics from [a05-0904b.kn.vutbr.cz]:38002 to [ps1.ochep.ou.edu]:49487 ---
SID:      810f28e7d47184c13d0a02b845a55d30
first:    2012-12-11T11:09:07.220
last:     2012-12-11T11:09:17.874
110 sent, 0 lost (0.000%), 0 duplicates
one-way delay min/median/max = 70.9/71/75 ms, (err=0.625 ms)
one-way jitter = 0.8 ms (P95-P50)
Hops = 18 (consistently)
no reordering

--- owping statistics from [ps1.ochep.ou.edu]:59486 to [a05-0904b.kn.vutbr.cz]:35185 ---
SID:      93e5ddabd47184c150cb6720b6e9702c
first:    2012-12-11T11:09:06.958
last:     2012-12-11T11:09:18.052
110 sent, 0 lost (0.000%), 0 duplicates
one-way delay min/median/max = 70.4/70.6/70.8 ms, (err=0.625 ms)
one-way jitter = 0.1 ms (P95-P50)
Hops = 18 (consistently)
no reordering
```

Obrázek 4.8: Ukážka spustenia nástroja Owping.

4.4.5 Nekorektné správanie

Ak sa klient nachádzal za prekladom adries, nebolo možné nadviazať spojenie s démonom. Z toho vyplýva, že aplikáciu nebude môcť používať väčšina bežných užívateľov kvôli rozšírenému použitiu prekladu adries.

4.4.6 Protokol OWAMP

Protokol OWAMP vznikol na požiadavky merania jednosmerného oneskorenia. Oproti nástroju Ping má umožňovať aj meranie stratovosti paketov. Protokol poskytuje možnosť merania zo stanice, ktorá nieje ani jednou z koncových bodov. Takto môžeme testovať oneskorenie medzi stanicami, na ktoré nemáme prístup. Medzi ďalšie vlastnosti protokolu patrí autentifikácia koncových bodov.

Pre tieto požiadavky bolo nutné protokol rozdeliť na dve časti: ovládaciú a testovaciu. Prvá má za účel nadviazať spojenie. Tieto správy nesú parametre merania a údaje pre autentifikáciu. Správy určené pre testovanie obsahujú časové razítka a sekvenčné číslo. Kvôli bezpečnostným vlastnostiam protokolu môžu obsahovať aj ďalšie údaje, napríklad pre overenie identity. Testovacie pakety môžu mať ľubovoľne nastavenú veľkosť.

4.4.7 Zhodnotenie

Nástroj hodnotíme veľmi kladne. Síce nám neposkytuje funkcionality v podobe merania priepustnosti, ale dokáže odmerať rozptyl oneskorenia, stratu a zmenu poradia paketov. Hlavnou úlohou je určenie jednosmerných latencií na linke medzi testovanými zariadeniami. Takto môžeme zistiť, že latencie sa môžu na daných smeroch líšiť, taktiež aj počet skokov a iné parametre. Nástroj je jednoduchý na ovládanie a dobre odladený. Tabuľka 4.9 obsahuje

súhrnné informácie o možnostiach použitia tohto nástroja pri diagnostikovaní siete. Tabuľka 4.10 ukazuje ďalšie vlastnosti tohto nástroja.

Nevýhodu vidíme v nutnosti použitia démona NTP na testovaných staniciach, čo je však pre meranie jednosmerného oneskorenia nevyhnutná súčasť správnej implementácie. Ak sa klient nachádza za prekladom adres, výrazne znižuje použiteľnosť nástroja kvôli implementácii, ktorá toto rozmiestnenie nedovoľuje.

TCP priep.	UDP priep.	Oneskorenie	Jitter	Zmena poradia	Strata
		✓	✓	✓	✓

Tabuľka 4.9: Merateľné parametre siete pomocou nástroja OWAMP.

Linux	Windows	FreeBSD	IPv6	NAT
✓		✓	✓	

Tabuľka 4.10: Ďalšie vlastnosti nástroja OWAMP.

4.5 Nástroj Thrulay

Ako ďalší nástroj pre diagnostiku a testovanie parametrov sieťového prenosu si uvedieme Thrulay. Tento projekt bol pôvodne založený Stanislavom Shanulov, ktorý implementoval jeho pôvodnú verziu. Neskôr sa vývoja ujala organizácia Internet2, ku ktorej sa pridal aj pôvodný autor. Druhou vývojovou vetvou je nástroj Thrulay-ng, ktorý vznikol za podpory projektu *Google Summer of Code*. Oba tieto projekty sú úzko zviazané a podporované organizáciou Internet2 pod vedením Jeff W. Boote. Tieto nástroje sú veľmi podobné a poskytujú takmer také isté možnosti testovania. Jediný markantný rozdiel je, že nástroj od organizácie Internet2 poskytuje pri UDP testovaní informácie o oneskorení a rozptyle oneskorenia.

Pre podobnosť nástrojov bude následná charakteristika zhodná pre obidva. Jedná sa o konzolovú aplikáciu napísanú v jazyku C. Primárne testuje TCP priepustnosť a RTT. S použitím protokolu UDP umožňuje otestovať oneskorenie a jeho rozptyl, stratu, duplikáciu a zmenu poradia paketov. Pre účely testovania bola vybraná verzia od organizácie Internet2 s číslom 0.9⁶.

4.5.1 Architektúra

Architektúra tohto nástroja je taktiež typu klient – server. Klientská časť má názov **thrulay** a serverová **thrulayd**.

4.5.2 Softvérové nároky

Ako bolo uvedené, tento nástroj bol vyvinutý organizáciou Internet2. Oproti nástrojom BWCTL a OWAMP nepotrebuje pre svoju činnosť aktívny NTP démon na synchronizáciu času. V aktuálne vybratej verzii sú podporované systémy Linux, BSD, Solaris a Mac OS X. Nástroj nie je dostupný v balíčkových repozitároch, preto je nutná kompilácia zo zdrojových textov.

⁶Dostupná na <http://e2epi.internet2.edu/thrulay/>

4.5.3 Popis vybraných parametrov

Parametre tohto nástroja sú rozdelené do dvoch skupín podľa aplikácii, ktoré ovládajú. Prepínače nájdeme v tabuľke 4.11.

Parameter	Popis
Prepínače pre thrulay	
-p <číslo>	špecifikácia čísla portu
-t <číslo>	dĺžka trvania testu (implicitne 60 s)
-u <číslo>[kMGT]	UDP test so špecifikovanou priepustnosťou v bitoch za sekundu, (k znamená 1000, M 10 ⁵)
-m <číslo>	počet TCP tokov (implicitne 1)
-i <číslo>	interval výpisov priebežných výsledkov, ak sa zadá 0, vypíše len výsledok (implicitne 1 s)
Prepínače pre thrulayd	
-p <číslo>	určí prijímajúci port (implicitne 5003)
-a <adresa/maska>	prijme spojenia iba z uvedenej adresy
-d	program sa spustí na popredí a bude vypisovať informácie o testoch na štandardný chybový výstup

Tabuľka 4.11: Vybrané parametre nástroja Thrulay.

Klientská časť nástroja Thrulay vyžaduje jeden povinný parameter – adresu alebo doménové meno stroja, na ktorom beží serverová časť `thrulayd`. Testovanie sa implicitne spúšťa na 60 sekúnd. Toto chovanie môžeme upraviť parametrom `-t`, ako je uvedené nižšie. Ak chceme spustiť démona na popredí, použijeme prepínač `-d`. Toto nastavenie zapne ladiace výpisy, z ktorých sa dozvieme namerané údaje, ktoré implicitne zobrazuje iba klient.

```
$ thrulay -t <čas> <adresa>
$ thrulayd -d
```

4.5.4 Ukážka testov

V tejto sekcii si uvedieme dva majoritné testy, ktoré sa dajú s týmto nástrojom vykonať. Prvý je test TCP priepustnosti, ako demonštruje ukážka 4.9. Oproti iným nástrojom nám poskytuje informácie o RTT a rozptyle oneskorenia. Väčšina nástrojov je schopná tieto charakteristiky odmerať pomocou protokolu UDP. V tom je tento nástroj výnimočný.

```
[pavol@pavol-VPCF11M1E thrulay-0.9]$ src/thrulay -t 20 -i 0 sec6net-mv1.fit.vutbr.cz
# local window = 262142B; remote window = 262142B
# block size = 65536B
# MTU: 1500B; MSS: 1448B; Topology guess: Ethernet (or PPP)
# MTU = getsockopt(IP_MTU); MSS = getsockopt(TCP_MAXSEG)
# test duration = 20s; intermediate reporting disabled
# delay (median) and jitter (interquartile spread of delay) are reported in ms
#(ID) begin, s    end, s    Mb/s    RTT delay,ms    jitter
#(**)    0.000    20.001    94.105    21.897    5.673
```

Obrázek 4.9: Ukážka spustenia nástroja Thrulay pre meranie TCP priepustnosti.

Ďalší test na obrázku 4.10 ukazuje meranie pomocou transportného protokolu UDP. Výsledok tohto merania nám poskytuje informácie o oneskorení a jeho rozptylu, ďalej strate, duplikácii a zmene poradia paketov. Posledné dva parametre boli pri základnom meraní vždy nulové. V ďalšej kapitole zameranej na merania zistíme, či sa výsledky zmenia.

```
[pavol@pavol-VPCF11M1E thrulay-0.9]$ src/thrulay -t 10 -u 100M sec6net-mv1.fit.vutbr.cz
Delay:      8.943ms
Loss:       0.000%
Jitter:     3.982ms
Duplication: 0.000%
Reordering: 0.000%
```

Obrázek 4.10: Ukážka spustenia nástroja Thrulay pre meranie pomocou protokolu UDP.

4.5.5 Nekorektné správanie

Neschopnosť naviazať spojenie pomocou sieťového protokolu IPv6.

4.5.6 Zhodnotenie

Nástroj hodnotíme veľmi kladne pre jeho jednoduché ovládanie a inštaláciu, ktorá nevyžaduje ďalšie podporné aplikácie. Sklamala nás neschopnosť testovania priepustnosti pod protokolom UDP. Testovanie pod protokolom UDP sa zameriava na zistenie jednosmerného oneskorenia pri vyťaženosť linky na danej priepustnosti. Toto meranie oneskorenia môže byť zavádzajúce, pretože v niektorých meraniach sme dostali záporné hodnoty, čo môže byť spôsobené rozdielnym časom na koncových staniciach. Preto nedoporučujeme testovanie oneskorenia pomocou tohto nástroja. Tabuľky 4.12 a 4.13 poskytujú informácie o možnostiach testovania tohto nástroja.

TCP priep.	UDP priep.	Oneskorenie	Jitter	Zmena poradia	Strata
✓				✓	✓

Tabulka 4.12: Merateľné parametre siete pomocou nástroja Thrulay.

Linux	Windows	FreeBSD	IPv6	NAT
✓		✓		✓

Tabulka 4.13: Ďalšie vlastnosti nástroja Thrulay.

4.6 Nástroj Nuttcp

V 80. rokoch 20. storočia so vznikom protokolu TCP bol implementovaný nástroj Ttcp pre meranie jeho priepustnosti. Ttcp bol vtedy zaradený medzi štandardné utility systému BSD. Od tej doby vzniklo viacero projektov, ktoré sú založené na tomto pôvodnom nástroji. Medzi jeho známe rozšírenia patrí Nttcp, v ktorom sú implementované rozširujúce možnosti testovania. Z Nttcp vychádza nástroj Nuttcp, ktorému sa budeme podrobnejšie venovať. Z jeho predchodcov si zachoval vlastnosti ako jednoduchosť ovládania a implementáciu v jednom zdrojovom texte. Pre výber tohto nástroja sme sa rozhodli, pretože

je v súčasnej dobe stále vyvíjaný a poskytuje najlepšie možnosti testovania. Taktiež sa nachádza v mnohých balíčkových repozitároch distribúcie Linuxu.

Nuttcp je nástroj vyvinutý na meranie TCP a UDP priepustnosti. Autori ho priamo porovnávajú s riešením Iperf. Podľa ich názoru je to najlepší dostupný nástroj pre svoju jednoduchosť, ľahkosť použitia a schopnosti merania [3]. Skladá sa z jedného spustiteľného súboru, ktorý implementuje klienta a démona súčasne. Pre účely testovania bola vybratá verzia 7.2.1⁷.

4.6.1 Architektúra

Taktiež sa jedná o aplikáciu typu klient – server. Obe strany sú implementované v jednom spustiteľnom súbore. Použitím prepínača sa vyberie zvolená strana. Komunikácia prebieha na dvoch portoch, z nich jeden je komunikačný (5000) a druhý určený na prenos testovaných dát (5001).

4.6.2 Softvérové nároky

Nuttcp pre svoju činnosť nepotrebuje žiadne doplnkové aplikácie, napríklad kvôli synchronizácii času. Medzi podporované systémy patrí Linux, FreeBSD, Solaris a Windows. Nástroj môžeme nájsť vo väčšine balíčkových repozitárov. Výhodná je kompilácia zo zdrojových textov, pretože sa nástroj stále vyvíja. Pre úspešnú kompiláciu je potrebný prekladač jazyka C a knižnica glibc. Nasledujúca ukážka demonštruje jeden z príkladov kompilácie.

```
$ cc -O3 -o nuttcp nuttcp-7.2.1.c
```

4.6.3 Popis vybraných parametrov

Tabuľka 4.14 poskytuje výpis najdôležitejších prepínačov. Aj napriek tomu, že je klient a server implementovaný v jednom spustiteľnom súbore, prepínače sme rozdelili do dvoch skupín na ovládanie klienta a démona.

Parameter	Popis
Spoločné prepínače	
-p <číslo>	číslo portu
-v	doplnkové výpisy
-4	použitie IPv4
-6	použitie IPv6
Prepínače pre klienta	
-r	prevedie testovanie od servera ku klientovi
-u	použitie protokolu UDP, testuje priepustnosť do 1 Mbps
-R <číslo>[MG]	určí maximálnu testovanú priepustnosť
-T <číslo>[mh]	dĺžka trvania testu, m – minúty, h – hodiny, bez značky sekundy
-i <číslo>	časový interval výpisov merania
Prepínače pre démona	
-S	spustí démona
-nofork	program sa spustí na popredí

Tabuľka 4.14: Vybrané parametre nástroja Nuttcp.

⁷Dostupná na <http://lcp.nrl.navy.mil/nuttcp/>

Pre správne spustenie klienta stačí zadať jeden argument: IP adresu alebo doménové meno stanice kde je spustená serverová časť. Niekedy je nutné pre nadviazanie spojenia použiť prepínač určujúci sieťový protokol. Druhý príkaz spustí démona na popredí.

```
$ nuttcp <adresa>
$ nuttcp -S --nofork
```

4.6.4 Ukážka testov

Prvý test na obrázku 4.11 obsahuje výsledok z merania TCP priepustnosti. Poskytuje informácie o veľkosti prenesených dát za časovú jednotku a dosiahnutú priepustnosť. Výpis obsahuje taktiež informácie o RTT v milisekundách a vyťaženie CPU lokálnej (TX) a koncovkej (RX) stanice. Taktiež údaj o znovu poslaných paketoch. Vypísaná hodnota RTT je v porovnaní s aplikáciou Ping veľmi podobná, preto ju môžeme považovať za správnu. Test sa implicitne spustí na 10 sekúnd. Toto chovanie môžeme zmeniť použitím prepínača `-T` viď 4.14.

```
[pavol@pavol-VPCF11M1E nuttcp]$ ./nuttcp sec6net-mv1.fit.vutbr.cz
103.9971 MB / 10.08 sec = 86.5818 Mbps 0 %TX 9 %RX 25 retrans 0.93 msRTT
```

Obrázek 4.11: Ukážka spustenia nástroja Nuttcp pre meranie TCP priepustnosti.

Nasledujúci obrázok 4.12 demonštruje výpis nástroja pre meranie UDP priepustnosti. Pri spustení bol použitý parameter, ktorý spôsobil použitie UDP protokolu a špecifikovanie maximálnej testovanej priepustnosti, ktorá bola v tomto prípade 100 Mbps. Oproti výstupu z merania TCP priepustnosti obsahuje informácie o celkovom počte poslaných a zahodených paketov. Z toho je vypočítaná stratovosť paketov.

```
[pavol@pavol-VPCF11M1E nuttcp]$ ./nuttcp -u -R 100M sec6net-mv1.fit.vutbr.cz
106.5078 MB / 10.02 sec = 89.1384 Mbps 4 %TX 9 %RX 5654 / 114718 drop/pkt 4.93 %loss
```

Obrázek 4.12: Ukážka spustenia nástroja Nuttcp pre meranie UDP priepustnosti.

4.6.5 Nekorektné správanie

Neschopnosť spustiť testovanie pod protokolom IPv6. Klientská časť aplikácie niekedy nebola schopná nadviazať spojenie. Túto chybu je možné potlačiť použitím prepínača `-4`, ktorý vynúti použitie sieťového protokolu IPv4.

4.6.6 Zhodnotenie

Hlavnou prednosťou tohto nástroja je jednoduchosť vo všetkých smeroch. Je implementovaný v jedinom súbore so zdrojovým textom, čo uľahčuje a urýchľuje kompiláciu. Uživatelské rozhranie v podobe prepínačov je tiež veľmi intuitívne a jednoduché. Výpisy sú zrozumiteľné a poskytujú len tie najdôležitejšie informácie. Aj keď nástroj nedokáže odmerať oneskorenie a jeho rozptyl, poskytuje základnú funkcionálnu pre meranie priepustnosti.

Pri testovaní UDP priepustnosti klientská časť vyťažovala CPU na maximum. Toto správanie považujeme za nedostatok implementácie.

TCP priep.	UDP priep.	Oneskorenie	Jitter	Zmena poradia	Strata
✓	✓				✓

Tabulka 4.15: Merateľné parametre siete pomocou nástroja Nuttcp.

Linux	Windows	FreeBSD	IPv6	NAT
✓	✓	✓		✓

Tabulka 4.16: Ďalšie vlastnosti nástroja Nuttcp.

4.7 Nástroj BWPing

Tento nástroj sme vybrali do našej práce pre jeho jedinečnosť implementácie testovania. Služi na meranie priepustnosti a času RTT. Testovanie prebieha pomocou protokolu ICMP. Program posieľa správy ICMP Echo Request a čaká na doručenie Echo Replay [2]. Týmto mechanizmom zisťovania priepustnosti sa líši od všetkých klasických nástrojov. Je to excelentné riešenie, ktoré nepotrebuje druhú koncovú stanicu so spusteným procesom tejto aplikácie. Táto implementácia má však svoje nedostatky. Ak sú po ceste filtrované ICMP správy, tento mechanizmus nefunguje. Meranie môže ovplyvniť taktiež aplikovaná QoS na meranej linke. Pre účely testovania bola vybratá verzia 1.7⁸.

4.7.1 Architektúra

Kvôli implementácii merania pomocou správ ICMP táto konzolová aplikácia vyžaduje iba klientskú časť. Aplikácia bola rozdelená do dvoch spustiteľných súborov, ktoré sa líšia použitím sieťového protokolu IPv4 a IPv6. Pre IPv4 je vyčlenený `bwping` a pre IPv6 `bwping6`. Je implementovaný v jazyku C s použitím BSD soketov typu RAW.

4.7.2 Softvérové nároky

Nástroj je dostupný v zdrojových textoch, takže je potrebná priama kompilácia. Pre túto činnosť je potrebný prekladač jazyka C a knižnica glibc. Dôležité je poznamenať, že pre spustenie musíme mať práva super užívateľa kvôli práci so soketmi typu RAW.

⁸Dostupná na <http://bwping.sourceforge.net/index.php>

4.7.3 Popis vybraných parametrov

Pre základné spustenie nástroj potrebuje tri povinné prepínače: prenosovú rýchlosť, veľkosť paketu a celkový objem prenesených dát. Tabuľka 4.17 obsahuje výpis prepínačov.

Parameter	Popis
-b <číslo>	prenosová rýchlosť v kbps
-s <číslo>	veľkosť paketu v bajtoch
-v <číslo>	objem poslaných dát v bajtoch
-r <číslo>	interval výpisov (implicitne vypnuté)
-B <adresa>	nastaví adresu odchádzajúcich paketov

Tabuľka 4.17: Vybrané parametre nástroja BWPing.

Nasledujúca ukážka demonštruje spustenie nástroja. Pre spustenie sú všetky parametre povinné.

```
$ bwping -b <číslo> -s <číslo> -v <číslo> <adresa>
```

4.7.4 Ukážka testov

Na nasledujúcej ukážke si uvedieme spustenie s výstupom merania. Keďže pracuje so správami ICMP, nie je možné vybrať transportný protokol. Testovala sa maximálna priepustnosť 90 Mbps s ICMP paketmi o veľkosti 61 KB a celkové množstvo poslaných dát bolo 9 MB. Výstup nám poskytuje informácie o počte poslaných a prijatých paketov, dosiahnutej priepustnosti a čase RTT, ktorý bol nameraný pri dosiahnutí nameranej priepustnosti.

```
[pavol@pavol-VPCF11M1E bwping-1.7]$ sudo ./bwping -b 10000 -s 61000 -v 9000000 sec6net-mv1.fit.vutbr.cz
Target: sec6net-mv1.fit.vutbr.cz (147.229.9.75), transfer speed: 10000 kbps, packet size: 61000 bytes, traffic volume: 9000000 bytes
Total: pkts sent/rcvd: 148/148, volume rcvd: 9028000 bytes, time: 8 sec, speed: 9028 kbps, rtt min/max/average: 23/23/23 ms
```

Obrázek 4.13: Ukážka spustenia nástroja BWPing.

4.7.5 Nekorektné správanie

Implementačné chyby tohto nástroja neboli zaznamenané, jedná sa o jednoduchú utilitu, ktorá je dobre odladená. Ak pri testovaní nastavíme príliš veľké hodnoty prepínačov, vypíše hlášku `bwping: sendto() failed: No buffer space available`. Toto je jediné zistené nekorektné správanie, ktoré môže zneprijemňovať testovanie.

4.7.6 Zhodnotenie

Prednosťou tohto nástroja je, že nepotrebuje pre svoju činnosť serverovú časť. Týmto spôsobom sme schopní odmerať priepustnosť voči ľubovoľnej stanici, na ktorú nemáme prístup.

Nevýhodou je použitie protokolu ICMP. Mnohí poskytovatelia pripojenia tieto správy filtrujú, tak nie je možné úspešne testovanie. Taktiež podpora kvality služieb môže ovplyvňovať výsledky. Hodnoty nameranej priepustnosti nikdy nezodpovedali reálne dostupnej. Taktiež uvedené problémy s ICMP protokolom znižujú jeho použiteľnosť.

Nástroj odporúčame len na experimentálne účely, alebo použitie vo vlastnej sieti na overenie konektivity. Tabuľky 4.18 a 4.19 obsahujú súhrn možností testovania s týmto nástrojom. Nameranú priepustnosť týmto nástrojom sme označili, že patrí pod protokol UDP.

TCP priep.	UDP priep.	Oneskorenie	Jitter	Zmena poradia	Strata
	✓				✓

Tabuľka 4.18: Merateľné parametre siete pomocou nástroja BWPing.

Linux	Windows	FreeBSD	IPv6	NAT
✓		✓	✓	✓

Tabuľka 4.19: Ďalšie vlastnosti nástroja BWPing.

4.8 Celkové vyhodnotenie

Po podrobnom otestovaní vybraných open source nástrojov, ktoré tvoria väčšinu dostupných aplikácií s týmto zameraním, je nutné ich medzi sebou porovnať. Každý z nástrojov je špecifický a ponúka rôzne možnosti testovania. Tabuľka 4.20 obsahuje informácie o možnosti použitia jednotlivých utilít na meranie špecifických parametrov sieťového prenosu, ktoré sme vybrali v kapitole 3.1. Naša metodika pre hodnotenie nástrojov skúma aj iné vlastnosti nástrojov. Tabuľka 4.21 obsahuje tieto funkcionálne vlastnosti uvedené v kapitole 3.2. Medzi vlastnosti sme pridali fakt, či je nástroj aktívne vyvíjaný. Ak bola posledná verzia vydaná v roku 2012, vývoj pokračuje.

Z výsledkov z tabuľky 4.20 je zrejmé, že každá utilita poskytuje iné možnosti testovania. Väčšina dokáže odmerať TCP alebo UDP priepustnosť. Nástroje, ktoré umožňujú testovanie pod transportným protokolom UDP, využívajú jeho vlastnosti a dokážu odmerať stratovosť, zmenu poradia a rozptyl oneskorenia paketov.

Testovania jednosmerného oneskorenia medzi dvoma stanicami je možné len v prípade, ak majú dostatočne synchronizovaný čas. Inak sú výsledky nepresné. Tento parameter dokáže odmerať OWAMP a Thrulay. Avšak Thrulay nevyžaduje synchronizáciu času, takže ho do tejto skupiny neradíme. Ostatné nástroje, ktoré nie sú schopné odmerať jednosmerné oneskorenie, poskytujú údaj o RTT pri nameranej priepustnosti.

Nástroj BWCTL sme do súhrnných tabuliek neuvádzali, pretože pre účely testovania používa Iperf, Nuttcp a Thrulay. Ak by sme schopnosti týchto troch nástrojov spojili, dokázali by odmerať takmer všetky parametre sieťového prenosu, ktoré sledujeme. Zložitosť testovania s týmto nástrojom je kvôli synchronizácii času a zdĺhavej inštalácii vysoká. Pre naše účely je BWCTL nevhodný nástroj.

Nástroj	TCP priep.	UDP priep.	Oneskorenie	Jitter	Zmena poradia	Strata
Iperf	✓	✓		✓		✓
Netperf	✓					
OWAMP			✓	✓	✓	✓
Thrulay	✓				✓	✓
Nuttcp	✓	✓				✓
BWPing		✓				✓

Tabulka 4.20: Možnosti nástrojov testovať parametre sieťového prenosu.

Všetky testované nástroje sú určené pre unixové operačné systémy. Podpora Windows je zabezpečené prostredníctvom Cygwin⁹. Utility, ktoré sú prispôsobené na Windows, je možné stiahnuť preložené v binárnej forme.

Ako je vidieť z tabuľky 4.21, všetky aplikácie okrem OWAMP správne fungujú s prekladom adres. Podpora IPv6 bola uvedená v manuáloch u všetkých nástrojov. Reálne testovanie ukázalo skutočnú funkcionálnosť. Projekty všetkých uvedených nástrojov aktívne pokračujú, čo je dôležité pre budúce využitie.

Nástroj	Linux	Windows	FreeBSD	IPv6	NAT	Aktívny vývoj
Iperf	✓	✓		✓	✓	✓
Netperf	✓	✓	✓		✓	✓
OWAMP	✓		✓	✓		✓
Thrulay	✓		✓	✓	✓	✓
Nuttcp	✓	✓	✓		✓	✓
BWPing	✓		✓	✓	✓	✓

Tabulka 4.21: Ďalšie vlastnosti testovaných nástrojov.

Výsledky tejto kapitoly nám pomohli zistiť, ktoré parametre sieťového prenosu sa dajú odmerať jednotlivými nástrojmi. V ďalšej kapitole vytvoríme metodiku, pomocou ktorej utility otestujeme na reálnej sieti. Z tabuľky 4.20 vyberieme parametre sieťového prenosu, ktoré budeme na reálnej sieti testovať.

⁹Dostupné na <http://www.cygwin.com/>

Kapitola 5

Testovanie na reálnej sieti

Po preskúmaní nástrojov sme ich boli schopní medzi sebou porovnať na úrovni funkcionality. Ďalším krokom tejto práce je testovanie na reálnej sieti. Následná analýza výsledkov meraní nám umožní vyhodnotiť použiteľnosť nástrojov s ohľadom na správnosť výsledkov. Cieľom tejto kapitoly bude určiť nástroj vhodný na meranie vybraného parametra sieťového prenosu.

Prvý krok spočíva vo vytvorení metodiky testovania. Tejto téme sa venuje kapitola 5.1. Keďže výsledky merania budú v číselnej podobe a testovanie ovplyvňuje veľké množstvo nezávislých faktorov, je nutné merania uskutočniť opakovane. Tomuto problému sa venuje kapitola 5.1.4. Po zhotovení metodiky a štatistickej úprave výsledkov budeme schopní výsledky vyhodnotiť v kapitole 5.2.

5.1 Metodika testovania

Metodika testovania nástrojov na reálnej sieti musí zahŕňať viacero faktorov. V prvok kroku je potrebné vyčleniť parametre sieťového prenosu, ktoré budeme merať. Keďže nástroje poskytujú rozličné testovacie možnosti, parametre je nutné vybrať tak, aby boli merateľné väčšinou nástrojov. Z tabuľky 4.20 môžeme vidieť, že tu patrí priepustnosť oboch transportných protokolov a strata paketov. K týmto vlastnostiam pridáme jednosmerné oneskorenie, pretože meranie tohto parametra poskytuje iba nástroj OWAMP. Získaná hodnota nás bude zaujímať s porovnaním s hodnotou RTT získanou pomocou utility Ping. Ďalej je nutné zaistiť, aby testovanie každého nástroja bolo rovnaké. Tým myslíme dĺžku trvania testu a čas, v ktorom bol spustený. Kvôli rôznej vyťažnosti liniek internetu v čase.

Parametre sieťového prenosu, ktoré budeme merať:

- TCP priepustnosť.
- UDP priepustnosť.
- Strata paketov.
- Jednosmerné oneskorenie.

5.1.1 Vstupné podmienky

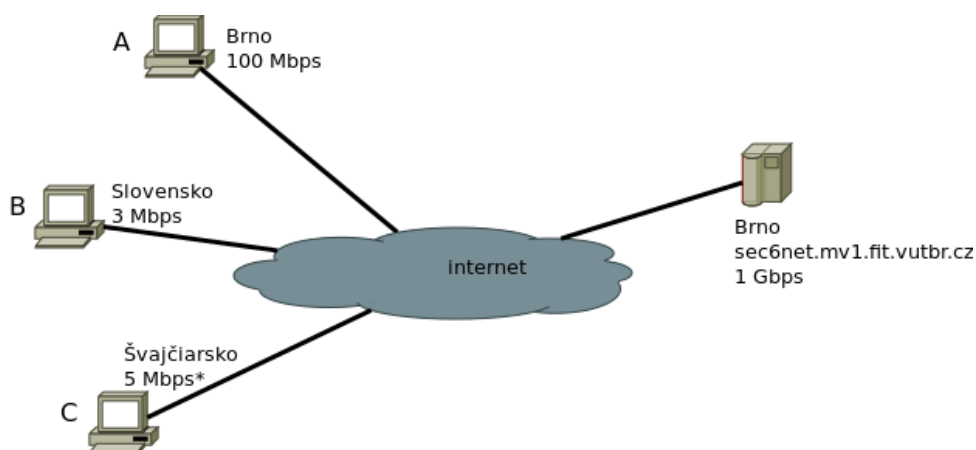
Keďže testovanie neprebíha v laboratórnych podmienkach, ale na reálnej sieti cez internet, rôzne dátové toky môžu ovplyvňovať namerané hodnoty. V neposlednom rade merania môžu ovplyvniť aj iné procesy bežiacie na koncových staniciach, obsluhu hardvérových prerušení a podobné príčiny. Preto budeme merania robiť v priebehu celého dňa s rozstupom 6 hodín. Takto uskutočníme celkovo 4 merania za 24 hodín. Prvé zahájime v čase 00:00.

Na koncových staniciach, medzi ktorými prebiehalo testovanie, budú povolené len základné služby, aby testovanie bolo čo najmenej ovplyvnené inými procesmi. Server, na ktorom budú spustené procesy démonov, má operačný systém Red Hat Enterprise Linux Server 6.3 (Santiago). Stanica, z ktorej budú spúšťané klientské časti utilít, disponuje operačným systémom Ubuntu 12.10.

5.1.2 Rozmiestnenie

Testovanie bude prebiehať vždy voči stanici s adresou `sec6net-mv1.fit.vutbr.cz` s gigabitovým pripojením do internetu. Tento školský server je umiestnený v Brne. Stanica, z ktorej bude spúšťané meranie prostredníctvom klientských častí nástrojov, má sieťovú kartu s maximálnou prenosovou rýchlosťou 100 Mbps. Na obrázku 5.1 je znázornené rozmiestnenie všetkých testovacích staníc. Väčšina koncových klientských staníc sa nachádza za prekladom adries. Preto neuvádzame ich IP adresy.

Meranie voči serveru v Brne bolo robené z troch lokalít s rôznym pripojením k internetu. Myslíme tým maximálnu priepustnosť uvedenú v zmluve s ISP. Prvá lokalita je Brno, kde máme k dispozícii pripojenie s rýchlosťou 100 Mbps. Počet skokov v dobe testovania bol 7. Ďalšia stanica je umiestnená na Slovensku s pripojením 3 Mbps. Počet skokov k serveru v Brne bol 11. Posledné meranie prebiehalo zo Švajčiarska. Tam sme mali internetové pripojenie s maximálnou rýchlosťou 5 Mbps. V zmluve s poskytovateľom internetového pripojenia je špecifikovaná maximálna priepustnosť. Provider však v zmluve uvádza, že priepustnosť môže dosahovať aj veľmi nízke hodnoty, podľa lokality prípojky, pretože pripojenie je poskytované v rámci celej siete *Swisscom*¹. Počet skokov v dobe testovania bol 13. Tabuľka 5.1 obsahuje informácie o parametroch internetových pripojení koncových bodov a počet skokov k serveru v Brne.



Obrázek 5.1: Náčrt testovanej topológie.

¹Hlavný telekomunikačný provider v Švajčiarsku

Miesto	Priepustnosť uvedená v SLA [Mbps]	Počet skokov
Brno	100	7
Slovensko	3	11
Švajčiarsko	5	13

Tabulka 5.1: Parametre internetového pripojenia koncových bodov.

5.1.3 Testy

V jednotlivých podkapitolách budú uvedené podrobnejšie informácie o zvolených testoch. Pre každý test je uvedený cieľ, nastavenie v podobe dĺžky trvania alebo počtu odoslaných paketov a jednotky, v ktorých budú uvedené výsledky. Pre všetky testy okrem jednosmerného oneskorenia, viď 5.1.3, platia podmienky uvedené v 5.1.1 a rozmiestnenie v 5.1.2.

Meranie TCP a UDP priepustnosti

Cieľom tohto testovania je odmerať najväčšiu možnú priepustnosť medzi koncovými stanicami. Priepustnosť bude meraná pod transportným protokolom UDP a TCP. Dĺžka testu bude nastavená na 20 sekúnd.

Pri testovaní maximálnej UDP priepustnosti je potrebné špecifikovať maximálnu priepustnosť. V tabuľke 5.2 sú uvedené hodnoty, ktoré boli použité pre špecifikovanie maximálnej UDP priepustnosti. Testovaná priepustnosť pre Švajčiarsko je oproti zmluve s ISP (maximálne 5 Mbps) menšia o viac než polovicu. Ako je uvedené v kapitole 5.1.2, maximálna priepustnosť pripojenia závisí od umiestnenia prípojky. V našom prípade pripojenie bolo v dobe merania pomalšie než 1 Mbps. Preto sme zvolili uvedenú hodnotu 2 Mbps. Ak zvolíme príliš veľké číslo, výstup nástrojov bude signalizovať veľké percento stratovosti paketov. Výsledná nameraná hodnota bude uvedená v jednotkách Mbps.

Miesto	Nastavená maximálna priepustnosť [Mbps]
Brno	100,00
Slovensko	3,50
Švajčiarsko	2,00

Tabulka 5.2: Hodnoty nastavenej maximálnej priepustnosti pri meraní UDP priepustnosti.

Pre meranie TCP priepustnosti sme vybrali nástroje: Iperf, Netperf, Thrulay a Nuttcp. Do testu UDP priepustnosti: Iperf, Nuttcp a BWPing.

Strata paketov

Cieľom tohto testu bude určiť percento stratených paketov k celkovému počtu prenesených pri dosiahnutej maximálnej priepustnosti. Stratu paketov je možné merať len pod transportným protokolom UDP. Nameraná hodnota bude uvedená v percentách podľa vzťahu 5.1. Hodnoty prevzaté z testovania UDP priepustnosti. Takto budeme schopní analyzovať stratovosť pri dosiahnutej maximálnej priepustnosti.

$$\frac{\text{stratené pakety}}{\text{celkový počet paketov}} * 100 [\%] \quad (5.1)$$

Pre meranie straty paketov boli zahrnuté nástroje: Iperf, Nuttcp a Thrulay. BWPing sme nezahrnuli, pretože za stratu paketu počíta nedoručený paket ICMP Echo Reply k odo-
slanému ICMP Echo Request.

Jednosmerné oneskorenie

Cieľom tohto testu je určenie rozdielu jednosmerného oneskorenia nameraného pomocou utility OWAMP a polovičnej hodnoty RTT pomocou nástroja Ping. Týmto testom sa snažíme zistiť reálnu použiteľnosť nástroja OWAMP, pretože utilita Ping je dostupná vo všetkých bežne používaných systémoch.

Pri meraní oneskorenia sme zvolili množstvo 50 testovacích paketov. Do výsledkov bude zahrnutá priemerná hodnota uvedená v milisekundách. Keďže s nástrojom OWAMP nie je možné testovať, ak sa klientská časť nachádza za prekladom adresy, bude prevedené meranie len z miesta v Brne, kde máme prístup k verejnej IP adrese.

5.1.4 Štatistická úprava výsledkov

Výsledky získané pomocou testovaných utilít majú číselnú formu. Preto je možné ich upraviť a získať informácie, ktoré nám bližšie pomôžu analyzovať výsledky. Ako bolo uvedené v 5.1.1, test ovplyvňujú viaceré nezávislé faktory. Preto je nutné testovanie vykonať opakovane.

Zo štatistického súboru získaných výsledkov z nezávislých behov spravíme aritmetický priemer podľa vzťahu 5.2. Ďalšou významnou úpravou výsledkov bude vypočítanie smerodajnej odchýlky, ktorá určuje, ako široko sú hodnoty rozložené v množine viď [10]. Tvar rovnice je uvedený v 5.3.

$$\bar{x} = \frac{1}{n} \sum_{i=0}^n x_i \quad (5.2)$$

$$s = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (5.3)$$

5.2 Analýza výsledkov

Výsledky z meraní sa nachádzajú v príslušných podkapitolách. Každý test obsahuje tabuľku s nameranými hodnotami a stĺpcový graf, ktorý prehľadne znázorňuje výsledky.

5.2.1 TCP Priepustnosť

Výsledky z merania TCP priepustnosti sú uvedené v tabuľkách 5.4 pre meranie z Brna, 5.5 pre Švajčiarsko a 5.6 pre Slovensko. Pre prehľadnosť uvádzame tabuľku 5.3, v ktorej sú údaje o priepustnosti uvedenej v SLA a nameranej.

Z výsledkov môžeme zhodnotiť, že nameraná priepustnosť z Brna a Slovenska je približne rovnaká ako uvedená hodnota v SLA. Priemerná priepustnosť zo Švajčiarska je 0.25 Mbps. Ako sme uviedli v predošlých kapitolách, tamojší poskytovateľ pripojenia uviedol, že rýchlosť je závislá na umiestnení prípojky. V našom prípade sa prípojka nachádzala

v lokalite, kde bola dostupná iba telekomunikačná sieť. Preto bola nameraná priepustnosť výrazne nižšia v porovnaní s hodnotou v zmluve.

Smerodajná odchýlka výsledkov jednotlivých nástrojov a priemerne nameranej priepustnosti je pomerne malé číslo. V prevedení na priepustnosť pre bod zo Slovenska je to 93 Kbps, čo je malá hodnota oproti priemernej 2940 Kbps. V prepočte na percentá je to 3,16 % z nameranej priepustnosti.

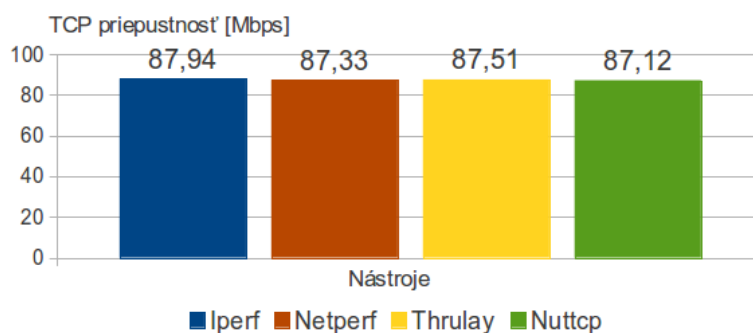
Po tejto analýze môžeme zhodnotiť, že výsledky meraní testovaných utilít sa odlišujú v malej miere. Preto je nemožné priamo rozhodnúť, ktorý nástroj je na meranie TCP priepustnosti najlepší v zmysle správnosti výsledkov. Pre meranie tohto parametra sieťového prenosu odporúčame ľubovoľný z testovaných nástrojov.

Miesto	Priemerná odmeraná TCP priepustnosť [Mbps]	Smerodajná odchýlka nástrojov od priemeru	Priepustnosť z SLA [Mbps]
Brno	87,48	0,30	100,00
Slovensko	2,94	0,093	3,00
Švajčiarsko	0,25	0,062	5,00*

Tabulka 5.3: Výsledky nameranej TCP priepustnosti.

Nástroj	Iperf [Mbps]	Netperf [Mbps]	Thrulay [Mbps]	Nuttcp [Mbps]
1.beh o 00:00	88,21	87,57	87,35	87,45
2.beh o 06:00	87,76	87,39	87,21	87,10
3.beh o 12:00	88,54	87,23	87,52	87,19
4.beh o 18:00	87,25	87,13	87,95	86,74
priemer	87,94	87,33	87,51	87,12
smerodajná odchýlka	0,30			

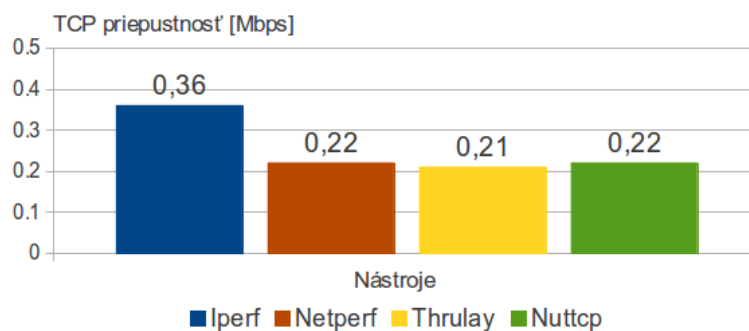
Tabulka 5.4: Odmeraná TCP priepustnosť testovaných nástrojov z Brna.



Obrázek 5.2: Graf odmeranej TCP priepustnosti testovaných nástrojov z Brna.

Nástroj	Iperf [Mbps]	Netperf [Mbps]	Thrulay [Mbps]	Nuttcp [Mbps]
1.beh o 00:00	0,36	0,22	0,21	0,22
2.beh o 06:00	0,36	0,22	0,21	0,22
3.beh o 12:00	0,41	0,21	0,21	0,22
4.beh o 18:00	0,37	0,22	0,21	0,22
priemer	0,36	0,22	0,21	0,22
smerodajná odchýlka	0,062			

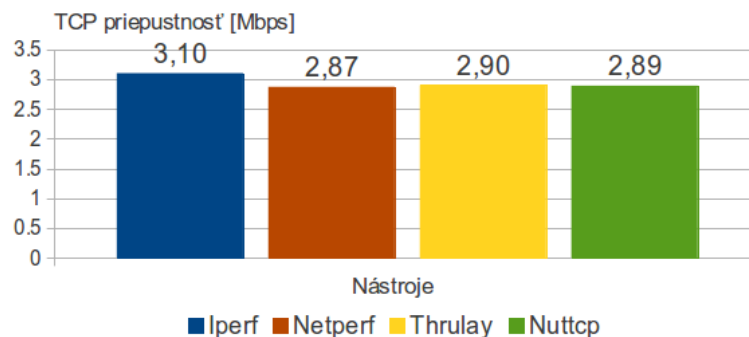
Tabulka 5.5: Odmeraná TCP priepustnosť testovaných nástrojov zo Švajčiarska.



Obrázek 5.3: Graf odmeranej TCP priepustnosti testovaných nástrojov zo Švajčiarska.

Nástroj	Iperf [Mbps]	Netperf [Mbps]	Thrulay [Mbps]	Nuttcp [Mbps]
1.beh o 00:00	3,10	2,86	2,91	2,89
2.beh o 06:00	3,11	2,89	2,91	2,90
3.beh o 12:00	3,12	2,83	2,86	2,87
4.beh o 18:00	3,09	2,89	2,91	2,89
priemer	3,10	2,87	2,90	2,89
smerodajná odchýlka	0,093			

Tabulka 5.6: Odmeraná TCP priepustnosť testovaných nástrojov zo Slovenska.



Obrázek 5.4: Graf odmeranej TCP priepustnosti testovaných nástrojov zo Slovenska.

5.2.2 UDP Priepustnosť

Výsledky z merania UDP priepustnosti sa nachádzajú v tabuľke 5.8 pre Brno, 5.10 pre Slovensko a 5.9 pre Švajčiarsko. Pre prehľadnosť je uvedená tabuľka 5.7, v ktorej sú údaje o priemerne nameranej UDP priepustnosti v porovnaní s hodnotou z SLA.

Namerané hodnoty priemernej priepustnosti sú veľmi podobné ako pri testovaní TCP priepustnosti. Oproti TCP priepustnosti by výsledky UDP priepustnosti mali mať väčšiu hodnotu, pretože v prenose odpadá režia TCP protokolu. Ako môžeme vidieť z výsledkov pre lokalitu zo Švajčiarska, vyšla UDP priepustnosť nižšia ako TCP. Keďže sa jedná o malú chybu, môže byť zapríčinená chybou merania.

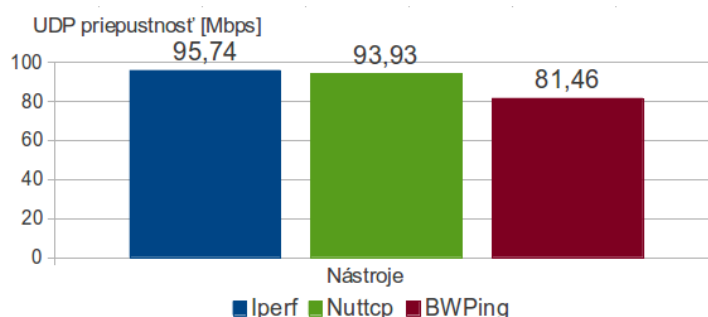
Smerodajná odchýlka je, podobne ako v prípade testu TCP priepustnosti, zanedbateľné číslo. Z tohto zistenia môžeme zhodnotiť, že pre meranie UDP priepustnosti je z množiny testovaných nástrojov vhodný ľubovoľný, okrem utility BWPing. Dosiahnutá priepustnosť s týmto nástrojom bola vždy výraznejšie nižšia v porovnaní s ostatnými. Rozdiel sa prejaví pri testovaní väčších hodnôt priepustnosti. Ako napríklad meranie z Brna, kde sme mali k dispozícii 100 Mbps prípojku.

Miesto	Priemerná odmeraná UDP priepustnosť [Mbps]	Smerodajná odchýlka nástrojov od priemeru	Priepustnosť z SLA [Mbps]
Brno	90,38	6,35	100,00
Slovensko	2,95	0,168	3,00
Švajčiarsko	0,21	0,008	5,00

Tabuľka 5.7: Výsledky nameranej UDP priepustnosti.

Nástroj	Iperf [Mbps]	Nuttcp [Mbps]	BWPing [Mbps]
1.beh o 00:00	95,71	93,94	65,83
2.beh o 06:00	95,75	93,94	84,57
3.beh o 12:00	95,78	93,94	87,75
4.beh o 18:00	95,72	93,88	87,69
priemer	95,74	93,93	81,46
smerodajná odchýlka	6,35		

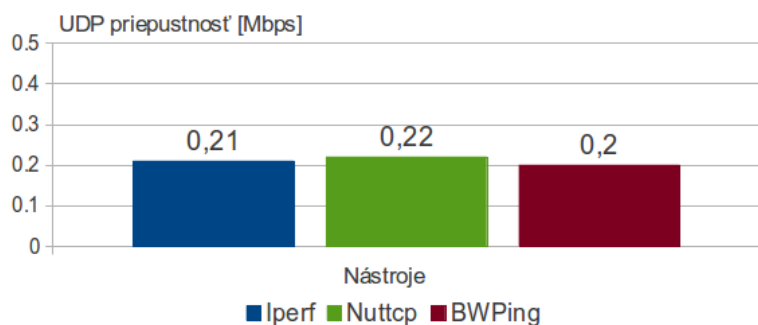
Tabuľka 5.8: Odmeraná UDP priepustnosť testovaných nástrojov z Brna.



Obrázek 5.5: Graf odmeranej UDP priepustnosti testovaných nástrojov z Brna.

Nástroj	Iperf [Mbps]	Nuttcp [Mbps]	BWPing [Mbps]
1.beh o 00:00	0,22	0,22	0,21
2.beh o 06:00	0,19	0,22	0,20
3.beh o 12:00	0,22	0,22	0,19
4.beh o 18:00	0,21	0,22	0,20
priemer	0,21	0,22	0,20
smerodajná odchýlka	0,008		

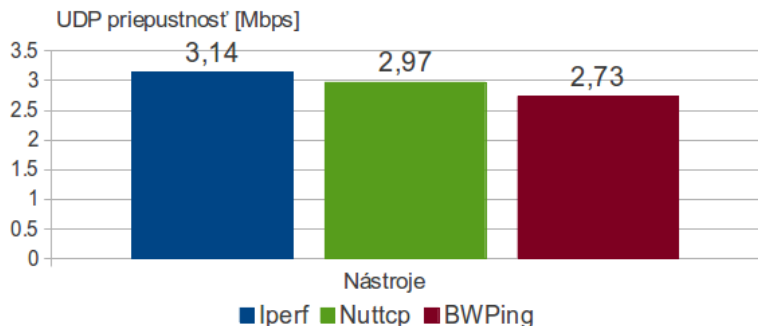
Tabulka 5.9: Odmeraná UDP priepustnosť testovaných nástrojov zo Švajčiarska.



Obrázek 5.6: Graf odmeranej UDP priepustnosti testovaných nástrojov zo Švajčiarska.

Nástroj	Iperf [Mbps]	Nuttcp [Mbps]	BWPing [Mbps]
1.beh o 00:00	3,28	3,11	2,78
2.beh o 06:00	3,10	2,93	2,95
3.beh o 12:00	3,08	2,91	3,04
4.beh o 18:00	3,10	2,94	2,13
priemer	3,14	2,97	2,73
smerodajná odchýlka	0,168		

Tabulka 5.10: Odmeraná UDP priepustnosť testovaných nástrojov zo Slovenska.



Obrázek 5.7: Graf odmeranej UDP priepustnosti testovaných nástrojov zo Slovenska.

5.2.3 Strata paketov

Výsledky stratovosti paketov sú uvedené v tabuľke 5.12 pre Brno, 5.14 pre Slovensko a 5.13 pre Švajčiarsko. Pre prehľadnosť uvádzame tabuľku 5.11, ktorá obsahuje výsledky nameranej stratovosti v percentách, odmeranej a nastavenej priepustnosti.

Z výsledkov je zrejmé, že ak nastavená hodnota maximálnej testovanej priepustnosti je väčšia ako reálne dosiahnutá, tak sa stratovosť paketov zväčší. Preto pri testovaní UDP priepustnosti doporučujeme vhodne zvoliť maximálnu priepustnosť, aby stratovosť nebola príliš veľká.

Výsledky meraní z Brna ukazujú, že výsledná hodnota stratovosti paketov je 0,00 % pre všetky nástroje. Nastavená hodnota testovanej priepustnosti bola 100 Mbps a dosiahnutá hodnota 90,38 Mbps. Nulová stratovosť je príčinou kvalitnej akademickej siete. Meranie nepreukázalo priepustnosť 100 Mbps. Tá je len teoretická, pretože klientská stanica disponuje sieťovou kartou s maximálnou rýchlosťou 100 Mbps.

Výsledky z ďalších dvoch destinácií ukazujú porovnateľné výsledky nástrojov. Smerodajné odchýlky vyšli 0,977 pre Švajčiarsko a 1,996 pre Slovensko. Tieto hodnoty poukazujú na to, že všetky testované nástroje odmerali približne rovnakú stratovosť pri nastavenej maximálnej priepustnosti. Z tohto môžeme usúdiť, že ľubovoľný z vybraných nástrojov je vhodný na meranie stratovosti paketov.

Miesto	Priemerná stratovosť paketov [%]	Nastavená UDP priepustnosť [Mbps]	Nameraná UDP priepustnosť [Mbps]
Brno	0,00	100,00	90,38
Slovensko	13,71	3,50	2,95
Švajčiarsko	84,04	2,00	0,21

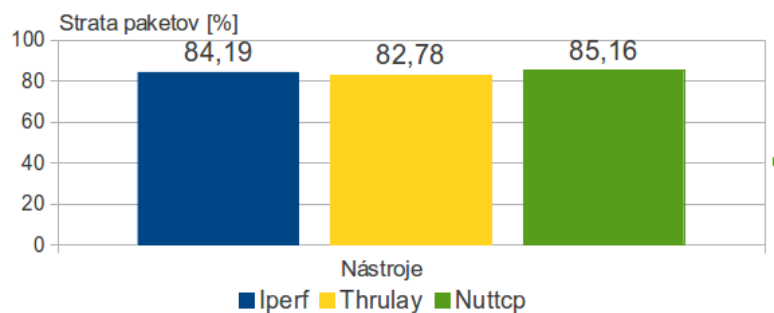
Tabuľka 5.11: Výsledky stratovosti paketov.

Nástroj	Iperf [%]	Thrulay [%]	Nuttcp [%]
1.beh o 00:00	0,00	0,00	0,00
2.beh o 06:00	0,00	0,00	0,00
3.beh o 12:00	0,00	0,00	0,00
4.beh o 18:00	0,00	0,00	0,00
priemer	0,00	0,00	0,00
smerodajná odchýlka	0,00		

Tabuľka 5.12: Odmeraná strata paketov testovaných nástrojov z Brna.

Nástroj	Iperf [%]	Thrulay [%]	Nuttcp [%]
1.beh o 00:00	83,36	82,72	85,15
2.beh o 06:00	85,41	82,69	85,15
3.beh o 12:00	83,44	82,71	85,17
4.beh o 18:00	84,56	82,98	85,16
priemer	84,19	82,78	85,16
smerodajná odchýlka	0.977		

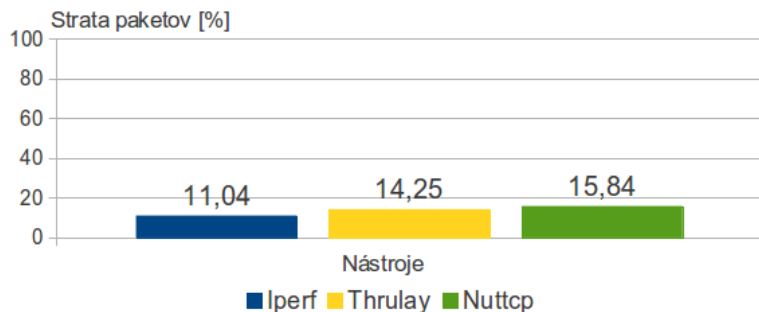
Tabulka 5.13: Odmeraná strata paketov testovaných nástrojov zo Švajčiarska.



Obrázek 5.8: Graf odmeranej stratovosti paketov zo Švajčiarska.

Nástroj	Iperf [%]	Thrulay [%]	Nuttcp [%]
1.beh o 00:00	9,90	14,04	14,27
2.beh o 06:00	11,23	14,10	16,07
3.beh o 12:00	11,79	14,78	16,92
4.beh o 18:00	11,24	14,09	16,08
priemer	11,04	14,25	15,84
smerodajná odchýlka	1,996		

Tabulka 5.14: Odmeraná strata paketov testovaných nástrojov zo Slovenska.



Obrázek 5.9: Graf odmeranej stratovosti paketov zo Slovenska.

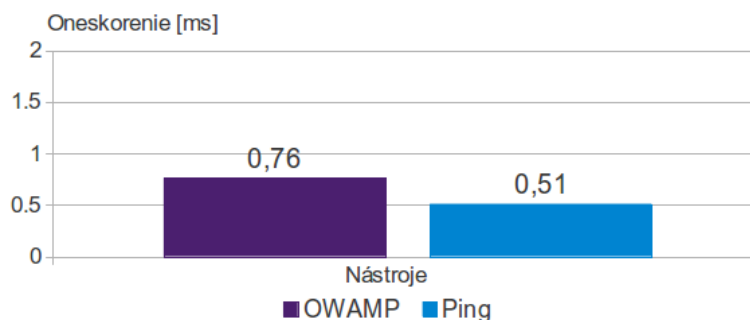
5.2.4 Jednosmerné oneskorenie

Výsledky z merania jednosmerného oneskorenia pomocou nástroja OWAMP sa nachádzajú v tabuľke 5.15. Hodnota RTT by mala byť približne rovná dvojnásobku hodnoty jednosmerného oneskorenia. Nemusí to platiť v prípade, ak sú linky v oboch smeroch rôzne zaťažené, alebo majú rozdielnú maximálnu prenosovú rýchlosť. Prípadne sa paket pošle inou cestou. Viacmenej hodnota RTT by mala byť vždy väčšia, ako hodnota jednosmerného oneskorenia.

Výsledky merania ukazujú, že hodnota získaná pomocou utility OWAMP je väčšia, ako hodnota RTT utility Ping. Aby sme mohli presne určiť, ktorý z nástrojov dáva presnejšie výsledky, je potrebné podrobnejšie a rozsiahlejšie testovanie. Toto však nie je predmetom našej práce.

Nástroj	OWAMP [ms]	Ping [ms] (RTT)
1.beh o 00:00	0,83	0,50
2.beh o 06:00	0,52	0,49
3.beh o 12:00	0,80	0,51
4.beh o 18:00	0,90	0,52
priemer	0,76	0,51

Tabuľka 5.15: Jednosmerné oneskorenie z Brna.



Obrázek 5.10: Graf jednosmerného oneskorenia z Brna.

5.3 Záver s doporučením

Výsledkom tejto kapitoly má byť doporučení pre bežného užívateľa, aby vedel, ktorý nástroj má použiť na meranie vybraného parametra sieťového prenosu. Tabuľka 4.20 obsahuje informácie o funkcionalite nástrojov pre meranie rôznych parametrov sieťového prenosu. My sme sa zamerali na overenie funkcionality nástrojov pre meranie priepustnosti pod oboma transportnými protokolmi TCP a UDP, stratu paketov a jednosmerného oneskorenia.

Merania v tejto kapitole ukázali, že všetky z nástrojov sú schopné odmerať vybrané parametre. Rozdiely vo výsledkoch boli minimálne. Preto výber najvhodnejšieho nástroja bude záležať na funkcionalite a možnostiach použitia. Taktiež treba brať v úvahu dostupnosť nástroja v balíčkových systémoch a nami nájdené nekorektné chovanie.

Pre meranie TCP priepustnosti by sme mohli zvoliť ľubovoľný z testovaných nástrojov, pretože všetky odmerali viac menej rovnaké hodnoty. Ak by sme mali vybrať nástroj, ktorý

sa jednoducho používa, je dostupný a poslúži aj na meranie ostatných parametrov, bol by to Iperf a Nuttcp. Utilita Iperf je v súčasnej dobe najrozšírenejšia. Nájdeme ju skoro vo všetkých balíčkových repozitároch. Druhý zmienený nástroj je výhodnejšie kompilovať zo zdrojových textov, pretože sa stále vyvíja.

Na merania UDP priepustnosti máme na výber tri nástroje: Iperf, Nuttcp a BWPing. Posledná utilita nebola schopná správne odmerať priepustnosť. Namerané hodnoty boli vždy menšie v porovnaní s ostatnými nástrojmi. Pre tieto fakty a problémy s ICMP správmi doporučujeme používať tento nástroj v prípade, ak nemáme prístup na druhú stanicu. Namerané hodnoty však nebudú korektné. Odporúčaný nástroj pre meranie UDP priepustnosti je Nuttcp a Iperf. Pri nástroji Iperf treba spomenúť nekorektné správanie, v ktorom klientská časť aplikácie niekedy nezobrazí namerané hodnoty serverovej časti. Toto sa prejavuje iba pri meraní UDP priepustnosti. Výsledky sa však zobrazia na štandardnom výstupe servera.

Stratovosť paketov úzko súvisí s meraním UDP priepustnosti. Pri tomto druhu teste máme k dispozícii informácie o stratovosti. Odporúčané nástroje by boli opäť Iperf a Nuttcp. Utilitu Thrulay nedoporučujeme, pretože neposkytuje informáciu o nameranej priepustnosti.

Ak by sme mali vybrať nástroj, ktorý dokáže odmerať všetky vybrané parametre sieťového prenosu, bol by to Nuttcp. Je veľmi jednoduchý na ovládanie, ale jeho aktuálna verzia sa nenachádzala v balíčkových repozitároch. Druhým najpoužiteľnejším nástrojom je Iperf. Je jednoduchý na použitie a jeho schopnosti merania sú rovnaké ako u Nuttcp. Jeho hlavnou prednosťou je dostupnosť aktuálnej verzie v balíčkových repozitároch.

Posledné meranie sa venovalo overeniu výsledkov nástroja OWAMP. Tento nástroj vždy odmeral hodnotu jednosmerného oneskorenia väčšiu oproti hodnote RTT. Výsledky sa priemerne líšili o 0.25 ms, čo je pomerne malá hodnota. Aby sme mohli rozhodnúť, ktorý nástroj poskytuje správne výsledky, vyžadovalo by to podrobnejšie merania. Treba zdôrazniť, že aplikácia Ping je dostupná vo väčšine systémov, oproti nástroju OWAMP, ktorý vyžaduje kompiláciu zo zdrojových textov a synchronizovaný čas pomocou NTP. Po zvážení týchto faktorov je jednoduchšie použiť utilitu Ping.

Kapitola 6

Záver

V našej práci sme sa zaoberali porovnávaním a testovaním open source nástrojov na meranie rôznych parametrov sieťového prenosu. Pre bližšie pochopenie, akým spôsobom sa merajú tieto parametre, sme uviedli metodiky, ktoré sa touto činnosťou zaoberajú. Tento krok nám taktiež pomohol k vytvoreniu metodiky na porovnanie nástrojov na báze funkcionality.

Hlavnou úlohou práce bolo testovanie nástrojov na reálnej sieti. Pre účely testovania bola vytvorená metodika, v ktorej sme stanovili, aké parametre prenosu budeme sledovať a za akých podmienok. Výsledky meraní sme analyzovali a nástroje medzi sebou druhý krát porovnali. V tomto kroku sme sa zamerali na rozdiely nameraných hodnôt. Výsledkom merania bolo určenie vhodného nástroja na meranie daného parametra sieťového prenosu. Pri určovaní vhodného nástroja sme brali do úvahy aj chybné správanie a ďalšie funkcionálne vlastnosti nástrojov.

Zo získaných informácií a výsledkov meraní sme vytvorili webové stránky¹. Obsahujú popisy jednotlivých utilít, ich funkcionálne schopnosti a doporučené, ktorý nástroj je vhodný na meranie daného parametra sieťového prenosu.

Z analýzy metodiky na meranie priepustnosti je jasné, že je potrebný softvér typu klient – server. To spôsobuje pre bežného užívateľa problém, pretože väčšinou nemá k dispozícii druhú koncovú stanicu s verejnou IP adresou, na ktorej by mohol nástroj nainštalovať. Toto by mohli zabezpečiť poskytovatelia pripojenia na ich serveroch.

Mnohokrát sa samotní klienti sťažujú poskytovateľovi služieb na „pomalé pripojenie“. ISP však nie je schopný odmerať priepustnosť voči klientovi. Tu nastáva priestor pre implementáciu softvéru, ktorý by bol umiestnený v aktívnom sieťovom zariadení, ktoré je pod správou ISP, ale umiestnené u klienta. Táto utilita by zabezpečovala meranie priepustnosti, straty paketov a iných parametrov potrebných pre diagnostiku pripojenia. Takto by sa poskytovateľ pripojenia mohol brániť pred sťažovaním klienta. Taktiež by to viedlo k rýchlejšej identifikácii závady. Táto aplikácia by mnohým užívateľom vyriešila problém merania parametrov sieťového prenosu.

¹Dostupné na <https://nes.fit.vutbr.cz/ansa/pmwiki.php?n=Main.Xlofffa00>

Literatura

- [1] *Bandwidth Test Controller (BWCTL): bwctld* [online]. 2009-2-23 [cit. 2012-12-5]. Dostupné na:
<<http://www.internet2.edu/performance/bwctl/bwctld.man.html>>.
- [2] *BWPing* [online]. [cit. 2012-3-24]. Dostupné na:
<<http://bwping.sourceforge.net/>>.
- [3] *Manpage of NUTTCP* [online]. [cit. 2012-3-22]. Dostupné na:
<<http://www.lcp.nrl.navy.mil/nuttcp/nuttcp.html>>.
- [4] ALMES, G., KALIDINDI, S. a ZEKAUSKAS, M. *A One – way Delay Metric for IPPM*. RFC 2679. Září 1999.
- [5] ALMESAK, G., KALIDINDI, S. a ZEKAUSKAS, M. *A One – way Packet Loss Metric for IPPM*. RFC 2680. Září 1991.
- [6] BRADNER, S. *Benchmarking Terminology for Network Interconnection Devices*. RFC 1242. Červenec 1991.
- [7] BRADNER, S. a MCQUAID. *Benchmarking Mthodology for Network Interconnect Devices*. RFC 2544. Březen 1999.
- [8] CLAISE, B. a MORTON, A. *Packet Delay Variaton Applicability Statement*. RFC 5481. Březen 2009.
- [9] CONSTANTINE, B., FORGET, G., GEIB, R. et al. *Framework for TCP Throughput Testing*. RFC 6349. Srpen 2011.
- [10] CYHELSKÝ, L. a SOUČEK, E. *Základy statistiky*. Praha: Vysoká škola finanční a správní, 2009. ISBN 978-80-7408-013-5.
- [11] DEMICHELIS, C. a CHIMENTO, P. *IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)*. RFC 3393. Listopad 2002.
- [12] GATES, M., WARSHAVSKY, A., TIRUMALA, A. et al. *DAST library* [online]. 2003-1-3 [cit. 2012-12-3]. Dostupné na:
<<http://pirlwww.lpl.arizona.edu/resources/guide/software/ipperf/lib.html>>.
- [13] JONES, R. *Care and Feeding of Netperf 2.6.X: Netperf Manual* [online]. 2012 [cit. 2012-12-5]. Dostupné na:
<<http://www.netperf.org/svn/netperf2/tags/netperf-2.6.0/doc/netperf.html>>.

- [14] McCANN, J., DEERING, S. a MOGUL, J. *Path MTU Discovery for IP version 6*. RFC 1981. Srpen 1996.
- [15] MOGUL, J. a DEERING, S. *Path MTU Discovery*. RFC 1191. Listopad 1990.
- [16] MORTON, A., CIAVATTONE, L., RAMACHANDRAN, G. et al. *Packet Reordering Metrics*. RFC 4737. Listopad 2006.
- [17] PAXSON, V., ALMES, G., MAHDAVI, J. et al. *Framework for IP Performance Metrics*. RFC 2330. Květen 1998.
- [18] SCHULZRINNE, H., CASNER, S., FREDERICK, R. et al. *RTP: A Transport Protocol for Real-Time Applications*. RFC 3550. Červenec 2003.
- [19] SHALUNOV, S., TEITELBAUM, B., KARP, A. et al. *A One-way Active Measurement Protocol (OWAMP)*. RFC 4656. Září 2006.

Příloha A

Obsah CD disku

- Zdrojový kód technickej správy v adresári **sprava**
- Skript pre spúšťanie automatizovaného testovania v adresári **test**
- Zdrojový kód webovej stránky v adresári **www**.
- Návod na použitie v súbore **readme.txt**