VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2019 Bc. Juraj Haško

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

SÍŤOVÝ TESTER
NETWORK TESTER

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. Juraj Haško

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. Václav Zeman, Ph.D.

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

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

Student: Bc. Juraj Haško ***ID:*** 164276
Ročník: 2 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Síťový tester

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je návrh a realizace systému pro měření přenosových parametrů datových sítí. Jednotlivé přenosové parametry specifikujte a popište metodiky využívané pro jejich měření. Na základě rozboru navrhněte řešení, které umožní komplexní měření datových sítí z hlediska přenosových parametrů. Výstupem práce bude softwarový tester, který bude zakomponován do již existujícího systému pro testování sítí, jehož základem je Apache JMeter.

DOPORUČENÁ LITERATURA:

[1] BRANDER, S. RFC 2544 - Benchmarking Methodology for Network Interconnect Devices, IETF, 1999.

[2] ERINLE, Bayo. Performance Testing with JMeter 2.9. Packt Publishing Ltd, 2013.

Termín zadání: 1.2.2019 ***Termín odevzdání:*** 16.5.2019

Vedoucí práce: doc. Ing. Václav Zeman, Ph.D.
Konzultant:

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

ABSTRAKT

Práca sa zaoberá problematikou testovania dátových sietí. Cieľom práce je navrhnúť metodiku pre komplexné meranie prenosových parametrov sietí a návrh koncepcie testeru a jeho realizácia pomocou rozšírenia existujúceho programu JMeter.

KĽÚČOVÉ SLOVÁ

tester, metodika, meranie, priepustnosť, testovanie, sieť

ABSTRACT

The thesis deals with data network testing. The aim of the thesis is to design a methodology for the comprehensive measurement of network transmission parameters and design of the tester concept and realisation by helping to extend the existing JMeter program.

KEYWORDS

tester, methodology, measurement, throughput, testing, network

HAŠKO, Juraj. *Síťový tester*. Brno, 2018, 60 s. Diplomová práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: doc, Ing. Václav Zeman, Ph.D.

VYHLÁSENIE

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

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

Brno ....

podpis autora(-ky)

POĎAKOVANIE

Rád by som sa poďakoval vedúcemu diplomovej práce pánovi Ing. Václavovi Zemanovi, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Brno ....

podpis autora(-ky)

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

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

POĎAKOVANIE

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

Brno ....

podpis autora(-ky)

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OBSAH

ZOZNAM OBRÁZKOV

ZOZNAM TABULIEK

ZOZNAM VÝPISOV

ÚVOD

Z dôvodu neustále zvyšujúceho sa množstva dát, ktoré sú prenášané sieťami je potrebné testovať siete a ich jednotlivé prvky. Pre testovanie parametrov sietí sa využívajú rôzne metodiky testovania a odlišné programy, ktoré dokážu testovať parametre sietí.

Cieľom tejto práce je definovať prenosové parametre, analyzovať metodiky pre meranie prenosových parametrov siete, navrhnúť metodiku pre komplexné meranie prenosových parametrov, návrh koncepcie testeru a realizovať tester, ktorý bude merať tieto parametre rozšírením existujúceho nástroja Apache JMeter.

V práci sú jednotlivé prenosové parametre ako šírka pásma, priepustnosť a strátovosť definované a popísané spôsoby ich merania. Pomocou odporučení RFC sú popísané postupy meraní prenosových parametrov a tieto odporučenia sú porovnané.

Súčasťou práce je návrh metodiky merania prenosových parametrov a návrh koncepcie testeru. Pri návrhu metodiky merania sú využívané niektoré postupy z odporučení RFC pre meranie priepustnosti siete. Návrh koncepcie testeru pozostáva z dvoch hlavných programov JMeter a Iperf. Tieto dva programy budú spolupracovať. Iperf slúži pre realizáciu merania, JMeter pre spustenie merania, ovládanie Iperfu a vyhodnotenie merania.

Výstupom práce je sofwérový tester pre meranie prenosových parametrov siete. Tester poskytuje možnosť automatizovaného merania prenosových parametrov siete.

1 MODEL TCP/IP

V tejto práci budem pracovať v dátových sieťach podľa modelu TCP/IP. Momentálne je najrozšírenejšia a najpoužívanejšia sada TCP/IP. V minulosti niektoré firmy používali vo svojich produktoch sadu IPX/SPX, ktorá však nebola pre globálnu sieť perspektívna. Vývoj siete prebiehal začiatkom 70. rokov.

Tento protokol je štandardná množina protokolov (TCP, UDP, IP, ICMP, RIP...) vyvinutá v roku 1969 americkým ministerstvom obrany, ako prostriedok na komunikáciu medzi rôznymi typmi počítačových zariadení a počítačových sietí. Teda je to komunikačný protokol. Jeho využitie spočíva najmä na „budovanie“ rozsiahlej siete na väčšie vzdialenosti, ale taktiež je vhodný na vytvorenie jednoduchej – lokálnej siete. Zahŕňa vlastný prenos paketov v počítačovej sieti – o to sa stará protokol IP, rozhranie pre rýchle, ale nespoľahlivé odosielanie dát – protokol UDP a protokol logického kanálu TCP. Najdôležitejšia vlastnosť architektúry TCP je rýchle a ľahké pripojenie rôznych zariadení do siete postavených na rôznych technológiách.

Veľkou nevýhodou TCP/IP architektúry je bezpečnosť. Hlavnou príčinou tejto chyby je nešifrovaný prenos dát. Posielané dáta môžu byť ukradnuté a zneužité. Túto dieru v bezpečnosti ošetruje šifrovaním aplikačná vrstva OSI modelu. Ďalej je tu veľký problém s nedostatkom IP adries. IPv4 protokol vyhradil iba 32 bitový adresný priestor pre sieťovú adresu. V priebehu 90-tych rokoch výrazne vzrástol počet používateľov internetu, začali sa používať rôzne aplikácie založené na internetovej komunikácii, a preto nastal rýchly úbytok IPv4 adries. Riešením je nový protokol IPv6, ktorý by mal nahradiť súčasný IPv4 protokol a môže vyriešiť jeho nedostatky. Samotný IPv6 protokol v sebe obsahuje bezpečnostné prvky. Táto adresa má dĺžku 128 bitov a zapisuje sa ako 8 skupín po 4 hexadecimálnych číslicach.

1.1 Štrutkúra modelu TCP/IP

Problematika komunikácie je z pohľadu tejto sady rozdelená do 4 vrstiev (systém OSI má 7 vrstiev): vrstva sieťového rozhrania, sieťová vrstva, transportná vrstva a aplikačná vrstva.

1.1.1 Vrstva sieťového rozhrania

Nie je špecifikovaná touto sadou, pretože je závislá na použitej prenosovej technológií. Zaisťuje príjem paketov do alebo zo siete.

Obr. 1.1: Model TCP/IP

1.1.2 Sieťová vrstva

Zaisťuje smerovanie paketov po sieti, zjednoťuje rôzne typy sietí na úrovni smerovania. Poskytuje nespojovú a nespoľahlivú službu. Funkcie vrstvy sú realizované protokolmi IP, ICMP, ARP, ARP, OSPF a IGMP.

1.1.3 Transportná vrstva

Realizuje a zaisťuje komunikáciu koncových uzlov. Multiplexuje a demultiplexuje dátový tok od a k jednotlivým aplikáciam. S entitami aplikačnej vrstvy komunikuje cez prístupové body. Z hľadiska spojenia poskytuje dve služby: spojovo orientovanú (TCP) a nespojovo orientovanú (UDP).

Protokol TCP

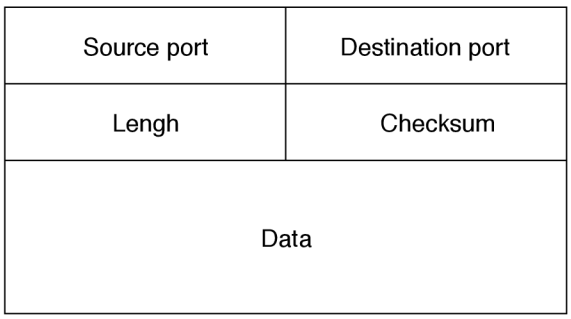
TCP je spojovo orientovaný, spoľahlivý protokol ktorý sa v TCP/IP modeli radí medzi aplikačnú a sieťovú vrstvu a poskytuje medziprocesnú sieťovú komunikáciu aplikácií založenú na toku oktetov. Veľkosť TCP hlavičky je od 20 do 60 bajtov v závislosti či sa použije pole options. Veľká hlavička síce pridáva réžiu navyše , avšak jednotlivé polia zabezpečujú jeho spolahlivosť. Nasledujúca tabuľka predstavuje TCP segment.

Keďže je TCP spojovo orientovaný protokol, je nevyhnutné aby pred samotným prenosom vytvoril spojenie. Tento proces sa nazýva Three way handshake.

Protokol UDP

Protokol UDP podobne ako protokol TCP poskytuje medziprocesnú sieťovú komunikáciu a nachádza sa na rovnakej vrstve ako TCP. Narozdiel od TCP však neposky-

tuje kontrolu toku ani potvrdenia o prijatých paketoch. UDP ale poskytuje kontrolu chybovosti, narozdiel od TCP však chybný paket len zahodí. Protokol UDP nevytvára spojenie. Protokol UDP je teda možné definovať ako nespojový, nespoľahlivý transportný protokol. Narozdiel od TCP je však jeho hlavička menšia a pri UDP komunikácii nedochádza k takej vysokej réžii ako v prípade TCP. Jeho vlastnosti ho robia vhodným na použitie vo VoIP a všade tam kde je dôležitá rýchlosť, jednoduchosť a nie je potrebné zabezpečiť potvrdenia. UDP pakety sa nazývajú aj datagramy a majú fixnú dĺžku 8 bajtov.



Obr. 1.2: Hlavička UDP datagramu

Hlavička UDP datagramu:

- **Source port / Destination port -** rovnako ako v prípade TCP definuje zdrojový a cieľový port.
- **Length -** pole definuje dĺžku celého datagramu, čiže hlavičku + dáta. Maximálna veľkosť je 65535 bajtov, avšak celková veľkosť je menšia keďže UDP datagram je zapúzdrený v IP a celková veľkosť IP datagramu je 65535 bajtov.
- **Checksum -** pole s kontrolným súčtom kontroluje chyby v celom datagrame (hlavička + data).

Protokoly TCP aj UDP následne predávajú svoje hlavičky jednému z protokolov sieťovej vrstvy. Najpoužívanejším je protokol IP.

1.1.4 Aplikačná vrstva

Ide o softvérovo riešenú vrstvu. Je to jediná sieťová vrstva, ku ktorej má priamy prístup užívateľ. Predpisuje v akom formáte a ako majú byť dáta preberané/predávané od aplikačných programov, tzn. dáta, ktoré chceme predať inej stanici sú prevádzané prostredníctvom aplikačných programov do počítačovej podoby. Prostredníctvom tejto vrstvy môžu užívatelia alebo aplikácie „vidieť“ výsledky služieb zaisťovaných všetkými predchádzajúcimi vrstvami. Aplikačná vrstva definuje spôsob, akým aplikácie komunikujú so sieťou, napríklad databázové systémy, elektronická pošta nebo programy pre emuláciu terminálov. Využíva služby nižších vrstiev a vďaka tomu je

„izolovaná“ od problémov sieťových technických prostriedkov a na rozdiel od ostatných vrstiev nezaisťuje služby pre vyššiu vrstvu (žiadnu už nemá). Dátovými jednotkami, prenášanými aplikačnou vrstvou sú APDU (Application Layer Protocol Data Unit).

Obsahuje protokoly najčastejšie používaných služieb, napríklad SMTP, FTP, TELNET, DNS, DHCP, BOOTP [12].

2 PRENOSOVÉ PARAMETRE DÁTOVÝCH SIETÍ

2.1 Kvalita služby QoS

Keďže architektúra siete založená na IP nie je určená na prenosy v reálnom čase, je nutné riešiť problematiku garancie služieb QoS v týchto sieťach. Dátové toky je nutné rozdeliť a implementovať vhodné QoS mechanizmy na uprednostnenie časovo citlivých aplikácií pred aplikáciami, ktoré nie sú až tak citlivé na oneskorenie a stratovosť paketov. Keby všetky siete mali neobmedzene zdroje, nevznikalo by preťaženie. Preťaženie vzniká, keď dopyt na sieťové zdroje presahuje dostupné kapacity. Zvýšenie kapacity sieti by bolo finančne a časovo náročné, využívajú sa QoS mechanizmy pre zabezpečenie kvality služieb. QoS, je definovaná nasledovne: Schopnosť siete poskytovať lepšie a zvláštne služby pre skupiny užívateľov a aplikácií na úkor ostatných užívateľov a aplikácií . Pomocou QoS mechanizmov môžeme poskytovať špeciálne služby a aplikácie užívateľom bez toho, aby sme zvyšovali kapacity sieti[13].

Kvalita služieb QoS sa zaoberá problematikou zabezpečenia rýchlej a dostatočne kvalitnej výmeny dát v aplikáciach, ktoré ju vyžadujú. Sú to napríklad aplikáciami sú hovory, videokonferencie, všeobecne systémy prenosu dát v reálnom čase. Táto problematika býva riešená väčšinou na vyšších vrstvách. [21].

Služby QoE môžme chápať ako subjektívny názor používateľa siete, kde sú jeho potreby rozhodujúce pri nastavení kvality jednotlivých služieb. Ide v podstate o vytváranie akéhosi kompromisu, kedy poskytovateľ služby chce čo najvýhodnejšie poskytnúť službu zákazníkovi tak, aby spĺňala jeho očakávania. Poskytovateľovi ide o zisk a tomu prispôsobí aj kvalitu služby [21].

Ak chceme merať kvalitu parametrov siete, musíme brať veľký ohľad na to akým spôsobom sa sieť využíva. Ak sa sieť používa na prenos napríklad internetovú televíziu (Real-time applications), tak požiadavky sú oveľa vyššie ako pri prenose textových súborov (dát). Z tohto dôvodu delíme služby na VoIP, dátové služby a služby obrazového charakteru [21].

2.1.1 Oneskorenie

Vyjadruje čas, ktorý uplynie od okamžiku, kedy element správy vstúpi do systému do okamžiku, kedy sa objaví reakcia na výstupe zo systému. Je to parameter, ktorý vyjadruje ako dlho skutočne trvá prenos po danej trase. Oneskorenie je vždy uvažované ako čas a je predovšetkým závislý na dĺžke trasy, veľkosti prenášanej správy, šírke pásma daného kanálu a taktiež zaťažení siete. Rýchlosť šírenia signálu je limitovaná rýchlosťou svetla. Oneskorenie signálu v jednej lokalite môže byť približne 1 ms, prenos medzi kontinentmi teda môže trvať až 100 ms. Závažnosť vplyvu oneskorenia

je závislá na veľkosti správy. Pri prenose malej správy nehrá rýchlosť prenosu takú významnú úlohu ako oneskorenie. Naopak pri prenose veľkých súborov očakávame, že prenos bude trvať niekoľko sekúnd, tak oneskorenie nie je také dôležité [12] [21].

Oneskorenie môžeme definovať ako časový úsek medzi okamžikom, kedy bola nejaká akcia inicializovaná a momentom, kedy je detekovaný efekt tejto akcie. V doporučení RFC 1242 je definované oneskorenie pre dva odlišné typy zariadení. Pre zariadenie typu „store forward“ sa uvádza ako časový interval medzi prijatím posledného bitu rámca na vstupnom porte a okamžikom, kedy sa objaví prvý bit odpovedajúceho rámca na výstupnom porte. Pri zariadení typu „forwarding device“ sa jedná o interval medzi detekciou prvého bitu rámca na vstupnom porte a okamžikom, kedy sa objaví prvý bit odpovedajúceho rámca na výstupnom porte [21].

Ďalším veľmi dôležitým parametrom je obojsmerné oneskorenie (Round Trip Time). Táto hodnota je rovná viac ako dvojnásobku oneskorenia trasy v jednom smere alebo Round Trip Delay, ktorý zahŕňa dobu prenosu správy zo zdrojového koncového uzlu siete k cieľovému uzlu, dobu spracovania správy, vygenerovania odozvy a prenosu naspäť k zdrojovému uzlu [21].

2.1.2 Kolísanie oneskorenia

Kolísanie oneskorenia môžme definovať ako rozdiel jednosmerného oneskorenia jedného paketu a jednosmerným oneskorením druhého. Ide o rozdiel medzi najväčším a najmenším oneskorením, ktoré pakety dosiahnu (delay jitter). Táto hodnota by nemala byť vyššia, ako je maximálne oneskorenie, čo je dokonca aj matematicky nemožné [21].

2.1.3 Šírka pásma

Šírka pásma opisuje maximálnu rýchlosť prenosu dát siete alebo internetového pripojenia. Určuje, koľko dát je možné odoslať cez konkrétne spojenie v danom čase. Napríklad gigabitové ethernetové pripojenie má šírku pásma 1000 Mb/s. Pripojenie k internetu cez káblový modem môže poskytnúť 25 Mb/s šírky pásma.

Kým šírka pásma sa používa na popis rýchlosti siete, nemeria to, ako rýchlo sa bity dát pohybujú z jedného miesta na druhé. Keďže dátové pakety prechádzajú cez elektronické alebo optické káble , rýchlosť každého preneseného bitu je zanedbateľná. Namiesto toho šírka pásma meria, koľko dát môže prúdiť cez konkrétne pripojenie naraz.

Dáta často pretekajú cez viacero sieťových pripojení, čo znamená, že spojenie s najmenšou šírkou pásma pôsobí ako prekážka. Všeobecne platí, že internetová chrbtica a prepojenia medzi servermi majú najväčšiu šírku pásma, takže zriedka sú

najužším miestom siete. Namiesto toho bežne býva najužším miestom siete pripojenie koncového uživateľa .

Rozdiel medzi priepustnosťou a šírkou pásma je, že šírka pásma je vnímaná ako teoretická hodnota určená charakteristikou prenosových spojov a ulov, je teda väčšia ako priepustnosť[21].

2.1.4 Priepustnosť

Priepustnosť siete je zvyčajne vyjadrená ako priemer nameraných hodnot a meraný v bitoch za sekundu, alebo v niektorých prípadoch ako dátové pakety za sekundu. Priepustnosť je dôležitým ukazovateľom výkonu a kvality sieťového pripojenia. Vysoký pomer neúspešného doručenia správ v konečnom dôsledku povedie k nižšej priepustnosti a zníženému výkonu.

Sieťové zariadenia komunikujú prostredníctvom výmeny dátových paketov. Priepustnosť udáva úroveň úspešného doručenia paketov z jedného miesta v sieti do druhého. Pokles paketov na trase znižuje priepustnosť a kvalitu sieťových pripojení. Priepustnosť má veľmi reálne dôsledky pre webové služby. Nízka priepustnosť alebo vysoká strata paketov má veľký vpliv na real-time aplikácie.

Priepustnosť siete je ovplyvnená viacerými faktormi. Patria sem atribúty, ako je výkonová náročnosť fyzického hardvéru vrátane káblov a smerovačov. Preťaženie siete a strata paketov môžu mať tiež vplyv na priepustnosť [21].

Táto veličina je definovaná ako objem prenesených dát d za daný čas t.

$$C = \frac{d}{t} \ \ [b/s] \ , \tag{2.1}$$

kde C je priepustnosť, d sú prijaté rámce a t je čas.

2.1.5 Chybovosť

Chybovosťou p(-) je uvažovaná veličina, ktorá popisuje kvalitu fyzického spoja, bitová chybovosť *BER* (Bit Error Rate). Na rozdiel od spomenutých veličín nepracuje s celými dátovými jednotkami, ale iba s jednotlivými prenášanými bitmi. Bitová chybovosť je definovaná ako pomer chybne prenesených bitov b_{ch} k počtu všetkých prenesených bitov b_c.

$$p = \frac{b_{ch}}{b_c} \ \ [-] \ . \tag{2.2}$$

Chybovosť udáva aké množstvo bitov alebo paketov bolo prenesených s nejakou chybou. Existujú dva druhy chybovosti a to bitová BER (bit error rate), a paketová chybovosť (packet error rate). Bitová chybovosť je dôležitá predovšetkým pre digitálne prenosy s nepretržitým dátovým tokom [18] [21].

2.1.6 Stratovosť

Stratovosťou dátovej jednotky L (%) percento stratených alebo zahodených dátových jednotiek l zo všetkých dátových jednotiek s. Strata dátovej jednotky môže nastať z dôvodu zahltenia smerovača a nemožnosti smerovača obslúžiť v jednom okamžiku veľké množstvo požiadavkov.

$$L = \frac{l}{s} * 100 \quad [\%] \ . \tag{2.3}$$

Stratovosť určíme odoslaním zvoleného počtu rámcov, pri zachovaní konštantnej rýchlosti, testovanému zariadeniu. Musíme byť schopní rozpoznať počet rámcov, ktoré sú testovaným zariadením úspešne prenesené. Podľa RFC 1242 je stratovosť definovaná ako počet odoslaných rámcov, ku prijatým. Keďže však pri prenose môže dôjsť ku stratám (Packet loss), nie sú príjemcovi doručené všetky rámce [21].

3 METODIKY MERANIA PRENOSOVÝCH PARAMETROV

3.1 Odporučenie RFC 2544

RFC 2544 je metodológia, založená organizáciou Internet Engineering Task Force (IETF), ktorá popisuje metodológiu merania na potvrdenie výkonnostných kritérií Ethernet smerovačov. Štandard definuje nezávislé ohodnotenie výkonnosti na základe testov priepustnosti, straty, oneskorenia a špičkovej záťaže overujúc špecifické časti dokumentácie SLA. Metodológia definuje veľkosť rámca, dĺžku trvania a počet opakovaní testovania. Po ukončení testovacej kampane, výsledky poslúžia ako meradlo výkonnosti smerovača, resp. siete.

Na zabezpečenie, že Ethernetová sieť je schopná prenášať viaceré služby (Video prenosy, VoIP a iné), RFC 2544 testy využívajú predpísané veľkosti paketov (64, 128, 256, 512, 1024, 1280 a 1518), aby vedeli simulovať rôznorodú prevádzku. Malé veľkosti rámcov umožňujú vyššiu rámcovú priepustnosť, ale spôsobuje preťaženia sieťových prvkov, keďže je potreba poslať ich vyšší počet. Testy RFC 2544 sú vhodné na meranie siete, t.j. viacerých prepojených zariadení, pričom boli pôvodne vytvárané pre testovanie individuálneho zariadenia, ktoré nemali potrebnú až takú náročnú prípravu. V prípade testov na meranie oneskorenia pri 20-tich opakovaniach pre 7 rôznych veľkostiach rámcov a doby merania 2 minút je minimálny čas na test 4,6 hodiny. Preto sa odporúča na následne procesy využiť automatizáciu, ktorá zjednoduší priebeh spomínaných testov.

Je teda pravdepodobné, že niektorí operátori spúšťajú len testy, ktoré sa pre nich zdajú zaujímavé, prípadne merajú len pri najvyššej a najnižšej hodnote veľkosti rámca. V každom prípade je treba si uvedomiť, že záťažové testy sú pre operátora podstatné, lebo po zavedení zariadení do prevádzky už nie je možné korektne vyhodnotiť výkon siete resp. zariadenia bez izolácie testovaného systému. To môže spôsobiť obmedzenia používateľov a ich služieb v sieti [1] [7].

3.1.1 Testované zariadenie

RFC popisuje zapojenie testovaného zariadenia DUT - device under test a teóriu nad prepojovaním sietí. Pri vývoji, návrhu a ladení zariadenia sa používa komplementárny prístup, kedy analyzujeme jednotlivé moduly a prvky vnútri zariadenia a ich prácu s dátami.

V praxi potom pri testoch považujeme testované zariadenia za čiernu skrinku. Nie je podstatná vnútorná štruktúra, zapojenie, prevedenie alebo prvky, z ktorých

je vyrobené. Rovnako sa nezaujímame o softvérovú stránku zariadení, ako sa v ňom dáta spracovávajú a posielajú ďalej.

Pre testovanie zariadení môžeme použiť tri varianty zapojenia testovaného zariadenia - DUT a zariadení posielajúcimi a prijímajúcimi zaslané testovacie dáta. Ideálnym a jednoduchým riešením je použitie testovacieho zariadenia s vysielacím aj prijímacím portom. Zasielanie testovacích dát prebieha z vysielacieho portu testovacieho zariadenia do DUT a z neho do prijímacieho portu testovacieho zariadenia. Možno potom rýchlo a elegantne vyhodnotiť vlastnosti, charakteristiky a výkon DUT. Možnou variantou merania je oddelenie vysielacieho a prijímacieho portu na dve nezávislé testovacie zariadenia. Pre vyhodnotenie testu je potom nutné mať stavové informácie. Pri testovaní viacnásobných sieťových zariadení môžu byť DUT zapojené v sérii [7].

3.1.2 Meranie priepustnosti

Test priepustnosti určuje maximálny počet rámcov za sekundu, ktorý môže byť prenesený úsešne a bez chyby. Po získaní údaju o maximálnom počtu prenesených rámcov za sekundu, je porovnaný celkový počet odoslaných a prijatých rámcov. Ak došlo k stráte pri niektorom z rámcov, prenosová rýchlosť vydelíme dvoma a test je opakovaný. Ak počas tohto merania nenastane žiadna strata rámaca, prenosová rýchlosť bude zvýšená o polovicu rozdielu oproti predchádzajúcemu procesu. Tento postup sa opakuje až do hodnoty rýchlosti rýchlosti, pri ktorej nedochádza k žiadnej strate. Test priepustnosti musí byť vykonaný pre každú z veľkostí rámca. Čas, po ktorý sú trvá prenos rámcov, musí byť minimálne 60 sekúnd. Výsledky môžu byť interpretované v počte rámcov za sekundu (f/s) alebo v bitoch za sekundu (b/s) [7].

3.1.3 Back-to-Back Test

Tento test hodnotí vyrovnávaciu pamäť spínača. Meria maximálny počet rámcov prijatých v plnej rýchlosti až kým nedôjde k strate rámca. V nosičoch ethernetových sietí je toto meranie užitočné ako potvrdenie prebytočnej informačnej rýchlosti (EIR), ako je definované v mnohých SLA.

Burst Back-to-Back rámcov je prenášaný cez sieť s minimálnou medzirámcovou medzerou. Ak nastane strata rámca, dĺžka burstu bude skrátená. Ak nenastane pri prenose chyba, dĺžka burstu bude zvýšená. Každý test by mal trvať minimálne 2 sekundy a mal by sa opakovať najmenej 50- krát. Priemerné hodnoty sú zaznamenané pre každú veľkosť rámca. Táto hodnota by mala byť zaznamenaná v správe [7].

3.1.4 Meranie stratovosti paketov

Tento test meria odozvy siete pri preťažení siete. Je to kritický indikátor schopnosti siete podporovať aplikácie, ktoré bežia v reálnom čase. Tieto služby sa môžu rýchlo stať neužitočné v momente, keď strata rámcov prestane byť kontrolovaná. Testovací prístroj vysiela prevádzku pri maximálnej rýchlosti linky a potom meria, či sieť zahodila niektoré rámce. Ak áno, hodnoty sú zaznamenané a test sa opakuje pri nižšej rýchlosti (o 10 a menej percent). Tento test sa opakuje až kým tri po sebe idúce testy budú uskutočnené bez straty rámca, následne sa vytvorí graf z týmito hodnotami. Výsledky sú prezentované ako percentá rámcov, ktoré boli zahodené. Takže percentá indikujú zmenu medzi ponúkaným zaťažením (odoslané rámce) a skutočným zaťažením (prijaté rámce). Aj tento test musí byť realizovaný pre všetky veľkosti rámcov [7].

3.1.5 Meranie oneskorenia

Meria čas, ktorý je potrebný pre rámec aby prešiel od odosielajúceho zariadenia cez sieť k prijímaciemu zariadeniu (end-to-end testovanie). Test môže byť tiež nakonfigurovaný aby meral čas prenášania rámcu z pôvodného zariadenia ku koncovému a naspäť. Ak sa oneskorenie mení od rámca k rámcu tak to spôsobí problémy v aplikáciach pracujúcich v reálnom čase. Test začína meraním a porovnaním priepustnosti pre každú veľkosť rámca aby sa zaistilo, že rámce nebudú vynechané. To spôsobí, že všetky vyrovnávacie pamäte zariadenia budú naplnené, teda meranie prebehne v najhorších možných podmienkach. Druhým krokom je pre skúšobný prístroj poslať prevádzku po dobu 120 sekúnd. V strednom bode prenosu musí byť rámec s časovým razítkom a keď je prijatý naspäť k testovaciemu zariadeniu, zmeria sa oneskorenie. Prenos by mal pokračovať ďalej do konca. Toto meranie je potrebné uskutočniť 20- krát pre každú veľkosť rámca a výsledky by mali byť zaznamenané ako priemerné hodnoty [7].

3.1.6 Zotavenie po preťažení

Tento test slúži k zisteniu doby, potrebnej pre celkové zotavenie testovaného zariadenia, potom čo dôjde k jeho preťaženiu. K tomu potrebujeme najskôr určiť priepustnosť testovaného zariadenia pri použití daného prenosového média. Potom začneme odosielať rámce s rýchlosťou 110 percent priepustnosti po čas aspoň 60 sekúnd. Tým dôjde k zahlteniu testovaného zariadenia, ktoré začne zahadzovať prijaté rámce. V časový okamžik A znížime rýchlosť odosielania na 50 percent priepustnosti a pokračujeme v odosielaní dát. Vo chvíli, kedy zistíme, že rámce už nie sú zahadzované, zapíšeme čas B. Rozdielom časov B-A dostaneme požadovaný čas pre zotavenie

systému po preťažení. Test by mal byť opäť niekoľkokrát zopakovaný a priemerná hodnota bude určená ako výsledok [7].

3.1.7 Výhody testov podľa RFC 2544

Metodika merania definuje jednotlivé parametre, ktoré je treba sledovať, definuje aj postup, ako tieto parametre merať. Výrobcovia zverejňovali z testov iba výsledky, ktoré robili dobrý dojem pre zákazníkov a vynechávali tie, u ktorých to tak nebolo. Pre relevantné vzájomné porovnanie sieťových prvkov od rôznych výrobcov je v RFC 2544 predpísané, akú podobu majú mať výsledky a aké namerané parametre majú byť obsahom takéhoto výstupu.

3.1.8 Nevýhody testov podľa RFC 2544

Nevýhody plynú najmä z dôvodu, že testy RFC 2544 boli navrhnuté pre laboratórne prostredie a tak je neprípustné aby sme napríklad zaťažili daný testovaný prvok na 100%.

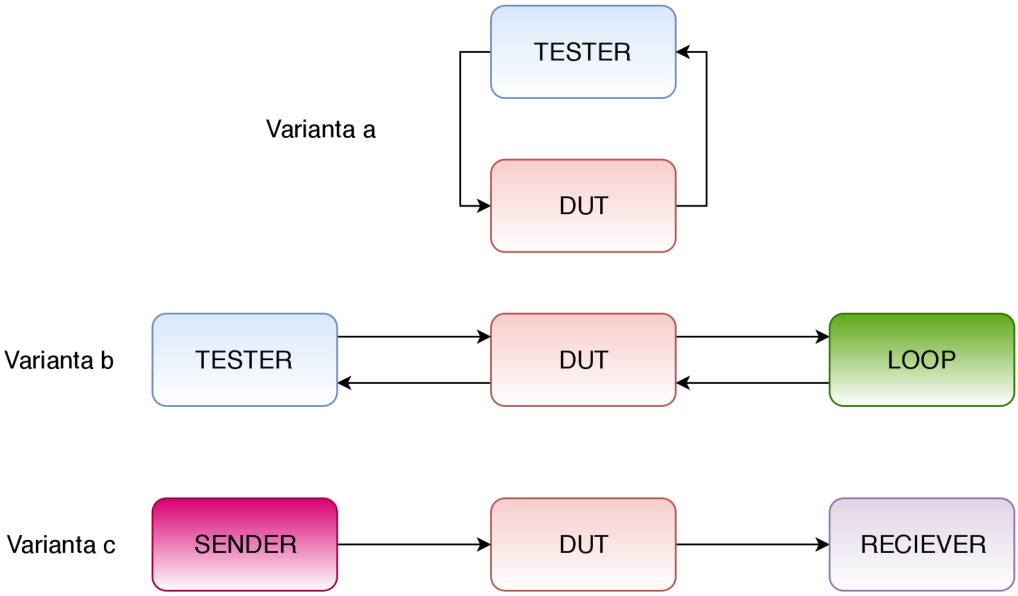
Taktiež nevýhodou je pomerne dlhá doba testovanie, ktorá môže byť až niekoľko hodín. V testoch nie sú zahrnuté všetky dôležité parametre ako napr. kolísanie v oneskorení jednotlivých dátových jednotkách. Neumožňuje testovanie paralelných dátových tokov (prenos videa, hlasových a dátových služieb). Operátor tak musí v sieti zaistiť ako sa bude zachádzať s jednotlivými dátovými tokmi.

3.1.9 Základná metodika testov

Spôsob zapojenia testovanej jednotky DUT vyplýva z vlastností DUT a z vlastností analyzátora.

Na obrázku **Obr. 3.1** zo zapojenia variantu A vyplýva, že informácie o vysielanom dátovom toku má aj prijímacia strana a preto sa dajú výsledky jednoducho analyzovať. Je to vhodné pre premeriavanie parametrov prvkov v rovnakej lokalite. Ak máme iba jeden analyzátor, ako vo variante b, a meranie prebieha v geograficky rozsiahlom prostredí, využijeme zakončenie dátového okruhu pomocou hardwarového zariadenia vo funkcií slučky, ktorá prichádzajúci tok odošle znovu k pôvodnému zdroju dát. Nevýhodou je, že prenos je ovplyvnený v oboch smeroch a tak dochádza ku skresleniu výsledkov.

V prípade, že potrebujeme premerať iba jeden smer toku, varianta c, musíme realizovať zdroj dát a analyzátor dvoma odlišnými zariadeniami a taktiež prenášať doplnkové informácie o dátovom toku zo zdroja do analyzátoru. Týmto meraním dosahujeme nepresnejšie výsledky a využíva sa pri meraní dátových okruhov s asymetrickými prenosovými parametrami v oboch smeroch [1].



Obr. 3.1: Možnosti zapojenia DUT pri testoch

3.1.10 Ďalšie nastavenie testov

Meraný dátový okruh alebo kanál musí byť správne nakonfigurovaný a konfigurácia DUT sa nesmie meniť. Tester by mal ignorovať všetky netestované rámce. Na prijímacej strane sa kontroluje správna veľkosť a integrita prijatých rámcov. Každý 100. vyslaný rámec by mal byť typu broadcast a podobne aj SNMP [1].

3.1.11 Typy a veľkosti rámcov

Veľkosť dátového rámca môže mať veľký vplyv na výsledné parametre prenosu, preto je nutné ich veľkosť presne špecifikovať. Počas merania sa majú použiť rámce, ktorých veľkosť nie je väčšia ani menšia než dovoľuje daná technológia alebo rozhranie. Odporučenie špecifikuje veľkosti dátových rámcov napríklad pre technológiu Token ring alebo FDDI. Odporučenú veľkosť rámcov pre technológiu Ethernet je možné vidieť v tabuľke 3.1 [1].

Prenos jedného rámca zahrňuje:

- 8 Byteov preambule rámce,
- N Byteov rámec(N je medzi 64 až 1518),
- 12 Byteov medzi-rámcová medzera.

Preambule dátového rámca sa používala pre synchronizáciu prijímača na dátový rámec. Medzi-rámcová medzera bola časový úsek potrebný pre regeneráciu elektronických obvodov pred príjmom ďalšieho dátového rámca.

Tab. 3.1: Odporučené veľkosti rámcov pre technológiu Ethernet

Veľkosť [B]	Rýchlosť,f/s [f/s]
64	14880
128	8445
256	4528
512	2349
1024	1197
1280	961
1518	812

3.1.12 Priebeh merania

1. Pred začatím sekvencie merania sa vyšlú informácie pre nastavenie smerovacej tabuľky.
2. Čakanie 2 sekundy, kedy sa DUT správne nastaví na základe prijatých informcií.
3. Vyslanie Learning frame na výstupný port DUT pre zaistenie správneho nastavenia CAM (Connect Addressable Memory) tabuľky.
4. Spustenie a sledovanie priebehu sekvencie merania v čase 60 s.
5. Čakanie 2 s na posledný testovací rámec.
6. Čakanie 5 s na ustálenie stavu DUT.

CAM tabuľka obsahuje priradenie portu sieťového prvku a zoznamu MAC adries iných sieťových prvkov, ktoré sú dostupné prostredníctvom tohto portu.

Dĺžku merania je v prípade potreby možno skrátiť z hodnoty 60 s na kratší čas.

Ako už bolo povedané, RFC 2544 je určené pre laboratórne merania, napriek tomu pre potreby merania na živej sieti sú vyčlenené organizáciou IANA adresy protokolu IPv4 v rozsahu od 198. 18. 0. 0 do 198. 19. 255. 255.. Cieľom tohto opatrenia je umožniť reálne meranie s minimalizáciou rizika, že test ovplyvní ostatné toky v reálnej sieti operátora [1].

Testy špecifikované v RFC 2544:

- Throughput (priepustnosť).
- Latency (oneskorenie).
- Frame Loss Rate (stratovosť rámcov).
- Back-to-Back frames (zaťažiteľnosť).
- System Recovery (zotavenie po preťažení).
- Reset (zotavenie po reštarte).

3.1.13 Záver k RFC 2544

Základnou úlohou RFC 2544 bolo vytvoriť metodiku pre overenie parametrov sieťových prvkov a umožniť tak vzájomné porovnávanie a overovanie ich parametrov.

Testy podľa RFC 2544 sú nahradzované modernejšími testami, ktoré sú zakotvené v nových standardizačných dokumentoch. Metodika RFC 2544 ja však ich súčasťou [1].

3.2 Odporučenie RFC 6349

Meranie touto metodikou je určené pre použitie v bežných sieťach, pre meranie prenosových parametrov prevažne u zákazníkov, ktorí si chcú zmerať parametre, ktoré im boli garantované poskytovateľom sieťového pripojenia [3].

3.2.1 Meranie cesty MTU

Implementácia TCP by mala používať Path MTU Discovery techniques (PMTUD). PDMTUD sa spolieha na správy ICMP „need to frag“ aby sa naučil cestu MTU. Keď zariadenie má paket, ktorý chce poslať a nemá fragmentovaný bit v hlavičke IP a paket je väčší ako MTU (maximálna prenosová jednotka) ďalšieho skoku, paket je zahodený a zariadenie pošle ICMP „need to frag“ správu späť k hosťovi, ktorý odoslal paket, táto správa obsahuje ďalší skok MTU, ktorý PMTUD používa pre svoje nastavenie sa. Vzhľadom k tomu, že veľa sietí má zakázané ICMP táto technika nie je vždy spoľahlivá [3].

3.2.2 Meranie Round-Trip Time

Round-Trip Time je uplynulý čas medzi taktovaním prvého zaslaného bitu TCP segmentu s obdržaním posledného bitu príslušného TCP potvrdenia. Malo by byť realizované mimo špičkových hodín za účelom získania spoľahlivého údaju o oneskorení siete. Inak môže dôjsť k ďalšiemu oneskoreniu v dôsledku ukladania dát do vyrovnávacej pamäte. Taktiež, ak sú vzorkovacie hodnoty RTT mimo daného testovacieho intervalu, najnižšia nameraná hodnota bude použitá ako východzia. Tak môžeme najpresnejšie odhadnúť skutočný vnútorný RTT.

Pri meraní je potrebné použiť testovacie zariadenie na každom konci siete tak, že paketový to môže byť meraný v oboch smeroch (end to end). Správanie paketov TCP môžeme zachytiť pomocou testovacích prvkov Iperfu alebo iných testovacích aplikácií. Po spustení niekoľkých meraní môžu byť zaznamenané údaje analyzované pre odhad RTT. Je dôležité vedieť, že by sme sa mali vyhnúť výsledkom založených na SYN - > SYN – ACK na začiatku TCP relácie, pretože firewall by mohol spomaliť

3-way handshake spojenia, taktiež na strane odosielateľa. Ostermannov Linuxový nástroj s argumentami *-l -r* môže extrahovať výsledky RTT priamo zo snímkov paketov.

ICMP ping môže byť taktiež použitý pre odhad RTT, za predpokladu, že sa berie do úvahy veľkosť paketu. Niektoré obmedzenia s ICMP pingom môžu zahŕňať odlišnosti v milisekundách. Taktiež rýchlosť ICMP je často obmedzená alebo zdržaná vo frontách vyrovnávacích pamätí. ICMP nemusí fungovať, ak sa zmena klasifikácie QoS vykonáva v každom skoku. ICMP nie je tak spoľahlivé a presné ako in-band meranie [3].

3.2.3 Meranie úzkeho hrdla

Tieto merania by mali byť spustené v oboch smeroch, najmä v nesymetrických sieťach (napríklad ADSL). Majú byť vykonávané v rôznych intervaloch počas pracovného dňa alebo počas týždňa. Testovanie v rôznych časových intervaloch poskytnú lepšiu charakteristiku priepustnosti TCP a lepšiu vnútornú diagnostiku. Testy šírky pásma by mali produkovať výstupy o dosiahnutej priepustnosti počas celej dĺžky trvania testu [3].

3.2.4 Meranie TCP priepustnosti

Počet pokusov a výber medzi jedným alebo viacnásobným TCP spojením bude založený na tom, aký máme zámer pri testovaní. Jediný test by mohol byť postačujúci pre zmeranie dosiahnuteľnej priepustnosti. Je však potrebné vedieť, že v IP sieti môžu byť použité rôzne techniky riadenia prevádzky a že niektoré z týchto techník je možné testovať iba ako viacnásobné spojenie. Je odporúčané uskutočňovať testy za rôznych podmienok vo všetkých smeroch v rôznych časoch dňa [3].

3.2.5 TCP Metriky

Táto metodika sa zameriava na priepustnosť TCP a poskytuje 3 základné metriky, ktoré môžu byť použité pre lepšie pochopenie výsledkov. Je známe, že zložitosť a nepredvídateľnosť TCP je problémom pre vývoj kompletných metrík.

Transfer Time Ratio: je to pomer medzi reálnym časom prenosu a ideálnym časom prenosu.

Efektivita TCP: ide o pomer medzi bytmi, ktoré museli byť poslané znovu voči celkovým preneseným bytom.

Buffer Delay: reprezentuje nárast RTT počas testovania TCP priepustnosti voči základnému RTT [3].

$$avgRTT = \frac{nRTT}{t} \quad [-] \text{ [3]}, \tag{3.1}$$

kde $avgRTT$ je priemer RTT počas prenosu, $avgRTT$ počet RTT počas prenosu a t celkový čas prenosu v sekundách.

$$bd = \frac{avgRTT \text{ - } dRTT}{dRTT} \times 100 \quad [\%] \text{ [3]}, \tag{3.2}$$

kde bd je buffer delay, $avgRTT$ počet RTT počas prenosu a $dRTT$ východzia hodnota RTT.

3.2.6 Uskutočňovanie TCP testov priepustnosti

Existuje niekoľko nástrojov, ktoré sa používajú pre toto meranie a jedným z najčastejšie používaných je Iperf. Pre meranie týmto nástrojom potrebujeme mať k dispozícií dve koncové stanice,na ktorých bude Iperf nainštalovaný, jedna z nich je nastavená ako klient, druhá ako server. Je možné nastaviť veľkosť prenášaného okna užívateľom. Priepustnosť môže byť meraná v jednom alebo v oboch smeroch. Pre dosiahnutie čo najpresnejších údajov by mal test trvať po dobu aspoň 30 sekúnd a opakovať by sa mal v rôznych časoch počas dňa [3].

3.2.7 Záver k RFC 6349

Meranie prenosových parametrov sietí touto metodikou prináša pomerne presné výsledky meraní z dôvodu opakovania testov v rôznych časoch dňa a týždňa. Tým sa vytvorí pomerne presný obraz o tom ako je v jednotlivých časoch vyťažená sieť a aké sú jej vlastnosti v týchto intervaloch.

3.3 Odporučenie ITU-T Y.1564

Toto odporučenie definuje metodiku pre posudzovanie správnej konfigurácie telekomunikačnej siete využívajúcej technológiu Ethernet a definuje metodiku pre posudzovanie parametrov prenosu služieb, ktoré takýto typ siete využívajú. Metodika vznikla z dôvodu aby aj poskytovatelia služieb mali štandardizovaný spôsob overenia parametrov svojich služieb. Vylepšenie proti RFC 2544 prišlo napríklad v podobe generovania a analýzy viacerých paralelných dátových tokov [9].

3.3.1 Kvalita služby a zmluva SLA

Vznikol nový vzťah medzi poskytovateľom služieb a zákazníkom, ktorý je upravený prostredníctvom zmluvy SLA (*Service Level Agreement*), teda Zmluvy o garantovanej úrovni služieb. Tento dokument obsahuje všetky podmienky, za ktorých sa uskutočnia dátové prenosy. Schopnosť poskytovať služby na určitej rovni sa posudzuje pomocou parametrov QoS (*Quality of Service*) [9]. Zmluva SLA obsahuje tieto časti:

- **Obchodná-** popis vzťahu dodávateľ- odberateľ
- **Služby-** špecifikácia služieb
- **Technická-** výpis parametrov a ich limít
- **Procesná-** spôsob vykazovania parametrov a súvisiace procesy

Metodika sa zameriava na overenie parametrov dátových tokov služieb. Taktiež sa zaoberá parametrami od sieťovej po transportnú vrstvu RM-OSI. Sledujú sa parametre ako stratovosť, oneskorenie, kolísanie veľkosti oneskorenia a priepustnosť, ktorá je definovaná iným spôsobom ako v RFC 2544 [9].
Zavádzajú sa tu podmienky pre testovanie paralelných dátových tokov, testovanie prioritizácie medzi tokmi a urýchľuje sa meranie tým, že jednotlivé merania sa neuskutočňujú iba sekvenčne za sebou. Dĺžka merania je 2 hodiny, avšak môžeme ju skrátiť iba na 2 minúty alebo predĺžiť až na 24 hodín. Meranie prebieha pre oba smery súčasne alebo pre každý smer zvlášť [9]. Overovanie parametrov prebieha v dvoch fázach:

Kontrola nastavenia sieťovej konfigurácie služieb

Postupne sa merajú parametre každej služby a overuje správnosť všetkých parametrov SLA. Prenosová rýchlosť sa nastavuje pomocou Ramp testu, ktorá stupňovite narastá a test trvá 1 až 10 s. V každom z krokov sú overované parametre SLA [9]. Definujú sa dve dôležité hraničné hodnoty prenosovej rýchlosti- CIR a EIR:

- CIR (*Committed Information Rate*) je maximálna prenosová rýchlosť dátového toku služby. Do tejto hodnoty sú prenosové parametre dátového toku garantované v definovaných medziach
- EIR (*Excess Information Rate*) je maximálna prenosová rýchlosť, kde už nie sú garantované všetky parametre prenosu.

Pomocou týchto dvoch hraničných rýchlostí je možné definovať tri oblasti:

- garantované pásmo- dáta sú prenášané vždy za dodržaní SLA parametrov
- Best Efford pásmo- dáta sú prenášané v prípade, že existuje v sieti voľná prenosová kapacita. Parametre SLA nemusia byť dodržané
- Dropped pásmo- dáta nie sú nikdy prenášané, testuje sa iba do hranice CIR+ EIR+ 25

Kontrola nastavenia parametrov kvality služieb QoS

Vďaka tomu, že sú služby generované parelelne, je možné overiť parametre v prostredí, ktoré sa bude veľmi blízke bežnému prostrediu. Služby sú generované súčasne do ich hodnoty prenosovej rýchlosti CIR a sú overované parametre všetkých služieb. Cieľom je napríklad overenie mechanizmu pre prioritizáciu alebo obmedzovanie konkrétnych dátových tokov v sieťových prvkoch pri ich paralelnom prenose, táto časť v testoch podľa RFC 2544 chýbala [9].

3.3.2 Záver k Y.1564

Odporúčanie ITU-T Y.1564 definuje metodiku testovania, ktorá sa môže použiť pri posudzovaní správnej konfigurácie a výkonu ethernetovej siete na poskytovanie služieb. Táto metodika testovania mimo prevádzky bola vytvorená tak, aby poskytovatelia služieb mohli mať štandardný spôsob merania výkonu služieb založených na princípe ethernetovej sieti. Nasadením Y. 1564 sa merajú prenosové parametre a porovnajú sa s očakávanými hodnotami pre každú službu. To zaisťuje, že sa nachádzajú namerané hodnoty v rámci svojho rozsahu, alebo v rámci prahových hodnôt definovaných pre garantovanú záťaž siete. [9]

4 NÁSTROJE PRE MERANIE PRENOSOVÝCH PARAMETROV

K zmeraniu prenosových parametrov dátových sietí môžme využiť rônze softvérové nástroje. Zrejmou výhodou týchto nástrojov alebo aplikácií je, že sú väčšinou ľahko dostupné, často opensourceové a pre užívateľa komfortné z pohľadu ovládatelnosti.

4.1 Iperf

Iperf je veľmi rozšírený sieťový testovací nástroj používaný na vytváranie TCP a UDP dátových tokov a na meranie priepustnosti siete, cez ktorú prechádzajú. Je voľne dostupný (open source) a funguje na rôznych platformách vrátane OS Linux, OS Unix alebo OS Windows. Za jeho podporu zodpovedá Národné Laboratórium pre Aplikovaný Sieťový Výskum v Spojených štátoch amerických.

Iperf je moderný nástroj na meranie priepustnosti siete, napísaný v C++. Dovoľuje používateľovi nastaviť rôzne parametre, ktoré môžu byť použité na testovanie siete alebo na optimalizovanie a ladenie siete. Funguje na princípe klient - server a môže merať priepustnosť medzi dvoma koncami buď jednosmerne alebo obojsmerne. Pri testovaní kapacity UDP umožňuje používateľovi nastaviť veľkosť datagramu a poskytne výsledky pre priepustnosť, kolísanie oneskorenia a stratovosť paketov. Pri testovaní TCP kapacity, Iperf meria priepustnosť uložených dát v pakete.

Ide o open sourceový softvér, ktorý je určený pre platformy Linux, Unix a Windows (natívne alebo vnútri Cygwin). Je voľne dostupný na stránke `https://iperf.fr/iperf-download.php` [14].

Pomocou Jperf je možné rozšíriť Iperf o grafické rozhranie a taktiež umožniť užívateľovi nastavovať parametre pri meraní bez použitia príkazového riadku.

Nevýhodou je skutočnosť, že pri meraní prenosových parametrov pomocou tejto utility dochádza k zahlteniu siete, preto je potrebné deaktivovať firewall. Iperf je možné použiť aj pre útok DOS (tzv. hladovanie šírky pásma).

Pri zisťovaní maximálnej šírky pásma odošle prvý paket s určitou veľkosťou a ak úspešne prejde k príjemcovi, nasledujúci odoslaný paket zvýši a takto postupuje až dokým veľkosť prenášaného paketu dosiahne maximálnu možnú úroveň, pri ktorej nedôjde k zahodeniu paketu.

Iperf meria tieto parametre:

- Šírka pásma
- Stratovosť paketov
- Kolísanie oneskorenia

- Schopnosť multicastu

Maximálna veľkosť segmentu *MSS* je najväčšie množstvo dát v bajtoch, ktoré môže počítač dodať v jednom nefragmentovanom TCP segmente. Môže byť počítaná ako maximálna prenášaná jednotka *MTU* zmenšená o hlavičky TCP a IP. Maximálna prenášaná jednotka je najväčšie množstvo dát, ktoré môže byť prenesené v jednom rámci. Pre nastavenie veľkosti *MTU* zadáme na strane klienta príkaz *iperf -c IP adresa serveru -M veľkosť -m*. Štandartné veľkosti *MTU* pre rôzne topológie sietí:

- Ethernet- 1500 B, používané v LAN sieťach
- PPPoE- 1492 B, používané v ADSL linkách
- Token Ring- 17914 B, stará technológia vyvinutá IBM
- Dial-up- 576 B

Všeobecne platí, že väčšia *MTU* prinesie väčšie využitie šírky pásma.

Iperf je dostupný pre najznámejšie operačné systémy, jeho výkonnosť je veľmi podobná reálnemu prostrediu. Dokáže vytvárať UDP aj TCP dátové toky, následne ich aj merať a má aj grafické používateľské prostredie. Funguje na báze klient - server, takže musí byť prítomný na oboch koncových zariadeniach.

4.2 JMeter

Apache JMeter je Java open source softvér, ktorý bol prvýkrát vyvinutý Stefanom Mazzoccim z Apache Software Foundation, je určený na testovanie správania a merania výkonnosti sietí. Pomocou nástroja JMeter je možné analyzovať a merať výkonnosť webových aplikácií alebo rôznych služieb.

Apache JMeter je bezplatný open source nástroj na testovanie výkonnosti platformy okolo 90. rokov. Je vyspelý, robustný, prenosný a vysoko rozšíriteľný. Má veľkú užívateľskú základňu a ponúka množstvo pluginov na pomoc pri testovaní.

Apache JMeter môže byť použitý pre testovanie funkčnosti a výkonnosti statických aj dynamických zdrojov (súbory, webové služby, dynamické webové jazyky, atď.). Môže byť použitý na simuláciu záťaže na server, sieť alebo objekt, prípadne pre analýzu celkového výkonu pod rôznymi typmi záťaže. Môže byť tiež použitý pre vykonávanie funkčných testov na webových stránkach, databázach, LDAP, webových službách, atď.[10].

4.2.1 Testovanie výkonnosti

Testovanie výkonnosti je typ testovania, ktorého cieľom je určiť citlivosť, spoľahlivosť, priepustnosť, interoperabilita a škálovateľnosť systému a / alebo aplikácie pod daným pracovným zaťažením. Môže byť tiež definovaný ako proces určovania rýchlosti alebo efektívnosti počítača, siete, softvérovej aplikácie, alebo zariadenia. Testovanie je možné vykonávať na softvérových aplikáciách, systémových zdrojoch, cielených aplikačných komponentoch, databázach atď.

Normálne zahŕňa automatizovaný testovací balík, pretože to umožňuje jednoduché, opakovateľné simulácie pre rôzne podmienky normálneho, špičkového a výnimočného zaťaženia. Takéto formy testovania pomôžu overiť, či systém alebo aplikácia spĺňa špecifikácie, ktoré požaduje predajca. Proces môže porovnávať aplikácie z hľadiska parametrov, ako je prenosová rýchlosť, priepustnosť, šírka pásma, efektívnosť alebo spoľahlivosť. Testovanie môže tiež pomôcť ako diagnostický nástroj pri určovaní úzkeho hrdla a jednotlivých bodov zlyhania. Často sa vykonáva v kontrolovanom prostredí a v spojení so záťažovým testovaním - proces určovania schopnosti systému alebo aplikácie zachovať určitú úroveň účinnosti za nepriaznivých podmienok [10].

Testovanie výkonu je zvyčajne spoločné úsilie všetkých zúčastnených strán. Strany zahŕňajú podnikateľské subjekty, podnikových architektov, vývojárov, testerov, DBA, administrátori systému a správcovia siete. Takáto spolupráca je nevyhnutná pre efektívne zbieranie informácií, presné a hodnotné výsledky pri vykonávaní testovania. Monitorovanie využitia siete, databázové vstupy a výstupy a čakanie, najvyššie dotazy a počty volaní, pomáha tímu nájsť úzke miesta a oblasti, ktoré si vyžadujú ďalšiu pozornosť prebiehajúceho úsilia o ladenie [10].

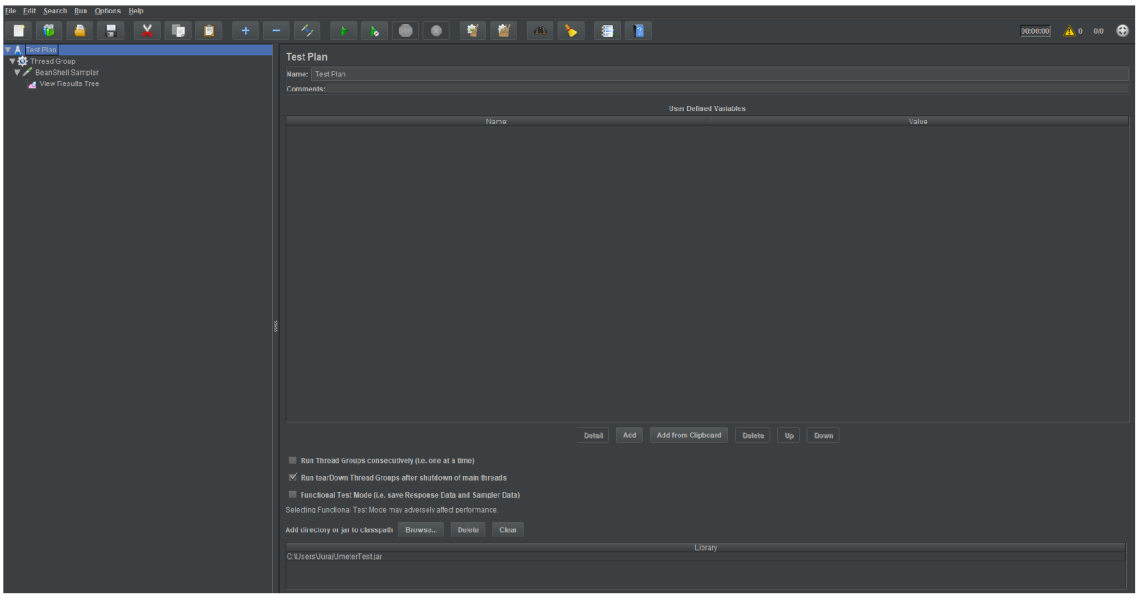
4.2.2 Prostredie

JMeter sa používa na testovanie webových aplikácií alebo aplikácií FTP. V súčasnosti sa používa na funkčné testy, test databázového servera atď.

4.2.3 Prehľad funkcií

Základné prvky štrutury pre zostavenie testovacieho plánu:

- Sampler- vytvára vzor pretvorbu samotných paketov,
- Logic controller- ovláda poradie v ako sú spracované jednotlivé samplery,
- Config element- slúži k nastaveniu konfigurácie a premenných pre ďalšie použitie v sampleroch,
- Timer- umožňuje ovládať test v čase(pozastavenie, zmena intenzity),



Obr. 4.1: Prostredie programu Apache JMeter

- Pre processor- súži k úprave samplerov pred ich spustením v danej časti testovacieho plánu,
- Post processor- aplikovaný po prebehnutí sapleru, unožňuje uživateľsky filtovať určité informácie z odpovede serveru,
- Assertions- slúži k validácií funkčnsti testovania na základe projatých odpovedí oproti odoslaným dotazom sampleru,
- Listener- zobrazuje odpoveď serveru na dotaz a výsledky testovania.

Hlavnou časťou JMetru je sampler, ktorý vykonáva jadro funkcie programu. Každý sampler predstavuje určitú funkciu nástroja, ako napríklad rôzne dotazy (HTTP). JMeter štandardne obsahuje tieto typy samplerov:

- FTP dotaz,
- HTTP dotaz,
- JDBC dotaz,
- Java dotaz,
- SOAP/XML-RPC dotaz,
- LDAP dotaz,
- Access log Sampler,
- BeanShell Samppler,
- BSF Sampler,
- TCP Sampler,
- JMS Publisher, Subscriber, Point-to-point,
- SMTP Sampler,

- OS Process Sampler.

4.2.4 Vytvorenie rozšírenia

Tento program umožňuje vytváranie vlastných funckií priamo v prostredí JMetru v sampleri alebo pomocou súborov vyexportovaných napríklad z prostredia eclipse do súborov *.jar*.

V prípade vyvárania kódu v sampleri nie je potrebné vkladať cestu k externému súboru, ale na druhej strane neponúka rozsiahle možnosti bežných vývojových prostredí.

Pre ukážku vytvorenia rozšírenia do JMetru som sa rozhodol pre vytvorenie jednoduchej kalkulačky. Samotný kód je napísaný priamo v prostredí eclipse. Následne je exportovaný ako *.jar* súbor do vybranej zložky v počítači.

Po spustení programu JMeter je potrebné v zložke Test Plan pridať cestu k vyexportovanému *.jar* súboru. Vytvorený program teda bude sučasťou JMetru. V sampleri je potrebné naimportovať vytvorenú triedu, s ktorou môže JMeter pracovať. Sampler obsahuje funkciu, ktorá naplní premenné, s ktorými bude program kalkulačky pracovať. Výsledok je zobrazený v konzolovom okne JMetru.

Spôsob, ktorým môžme naplniť premenné v sampleri môže byť následovný:

```
import testing.Calculation;
Calculation obj = new Calculation();
a=obj.sum(10,20);
System.out.println(a);
```

4.2.5 Spustenie externého programu

Pomocou vývojového prostredia v JMetri je taktiež možné spustiť externý program, čo bude potrebné pre pokračovanie v tejto práci. V tomto prípade bude spustený program Iperf, ktorý bude používaný v tejto práci.

Je potrebné v programe napísať celú cestu k požadovanému *.exe* sóboru. Pri spúšťaní programu Iperf je taktiež potrebné doplniť parameter, ktorý bude určovať či sa jedná o zapnutie Iperfu v režime klient alebo server, tento parameter bude súčasťou príkazu, ktorý bude posielaný zo sampleru. Po prebehnutí tohoto kódu bude spustený program Iperf v novom okne.

Príklad spustenia externého programu:

```
try {
      Process p = Runtime.getRuntime().exec("cmd_/iperf.exe");
      p.waitFor();
      System.out.println("DONE!");
```

```
    } catch (IOException e) {
   // TODO Auto-generated catch block
      e.printStackTrace();
  }
```

4.3 NutTCP

Je open sourcevý nástroj pre meranie výkonnosti siete. Jeho najzákladnejšie využitie je určiť priepustnosť sieťovej vrstvy TCP alebo UDP prenosom TCP segmentov alebo UDP datagramov zo zdrojového systému naprieč prepojovacou sieťou do cieľového systému. Je to nástroj s podobnými vlastnosťami ako Iperf.

4.4 Porovnanie nástrojov

Pre túto prácu je potrebné zvoliť vhodný nástroj, ktorý dokáže spolupracovať s programom JMeter. Na základe vlastností som sa rozhodol pre túto prácu využiť Iperf verzie 2. Hlavnou výhodou oproti novšej verzií Iperf3 je možnosť zmerať priepustnosť siete jednoduchšie, pri zachovaní presnosti. Program Iperf3 totiž k zmeraniu priepustnosti vyšších hodnot potrebuje vytvoriť viacnásobné spojenie, aby dokázal vysielať dostatočné množstvo dát, čo komplikuje výpočty a prácu s výsledkami.

Pri porovnaní programu Iperf s nuttcp je náročné posúdiť, ktorý je vhodnejší program pre meranie priepustnosti. Pre túto prácu som sa rozhodol využívať Iperf.

Tab. 4.1: Tabuľka porovnania nástrojov pre meranie prenosových parametrov

	iperf2.0.5	**iperf2.0.8+**	**iperf3.1.5+**	**nuttcp 8.x**
Priepustnosť TCP	áno	áno	áno	áno
Priepustnosť UDP	áno	áno	áno	áno
JSON výstup	nie	nie	áno	nie
Mulitiplatforma	áno	áno	áno	áno
Open source	áno	áno	áno	áno

5 NÁVRH METODIKY MERANIA PRENOSOVÝCH PARAMETROV

Súčasťou zadania práce je navrhnúť metodiku pre komplexné meranie prenosových parametrov dátových sietí. V tejto práci som sa rozhodol navrhnúť metodiku pre test priepustnosti UDP, ktorý bude zahŕňať meranie priepustnosti na rôznych portoch. Návrh metodiky vychádza z odporúčania RFC 2544.

5.1 Meranie priepustnosti UDP

Návrh merania priepustnosti vychádza z predpokladu, že meranie bude prebiehať v laboratórnom prostredí, čo v praxi znamená, že sieť a jednotlivé prvky siete budú počas merania využívané a zaťažované iba prenosom dát, ktoré sú využívané pre samotné meranie. Týmto obmedzíme možnosť skreslenia výsledkov merania.

5.1.1 Postup merania

Meranie priepustnosti UDP bude vychádzať z odporučenia RFC 2544. Na začiatku bude realizované meranie pomocou vysielania rámcou s danou veľkosťou 64 B ako je definované v RFC 2544, následne príde k zmeraniu UDP priepustnosti tým, že budú po dobu 10 sekúnd vysielané UDP rámce o tejto veľkosti.

V ďalšom kroku bude zvýšená veľkosť vysielaných UDP datagramov a test sa bude opakovať až po maximálnu veľkosť datagramu podľa odporúčania. Meranie pre každú veľkosť datagramov bude trvať 10 sekúnd.

Tab. 5.1: Tabuľka veľkostí datagramov

Veľkosť datagramu [B]	64	128	256	512	1024	1280	1518

Ďalšou možnosťou testu bude meranie s vlastnými hodnotami datagramov, bude teda možné určiť priepustnosť UDP pre rôzne veľkosti datagramov. Týmto môžeme získať viac informácií o priepustnosti siete. Taktiež môžme merať priepustnosť na rôznych portoch.

5.1.2 Formát výsledkov merania

Výsledky merania budú zobrazené v podobe tabuľky a následne uložené do súboru *.csv*, kde bude možné s nimi ďalej pracovať. Výsledný súbor bude obsahovať informáciu o veľkosti datagramu, s ktorými bolo meranie realizované, číslom portu, ktorý bol použitý pri meraní a nameranú priepustnosť v bitoch za sekundu (b/s).

6 NÁVRH KONCEPCIE TESTERU

Cieľom tejto práce je vytvoriť sieťový tester, ktorý bude testovať prenosové parametre dátových sietí a bude využívať existujúci OpenSourceový program Apache JMeter.

K tomuto programu je potrebné využiť možnosti niektorého z existujúcich programov pre meranie prenosových parametrov sietí, pre komplexnejšie meranie. Po porovnaní rôznych programov pre meranie týchto parametrov som sa rozhodol pre využitie programu Iperf z dôvodu vlastností, ktoré sú vhodné pre automatizáciu merania prenosoých parametrov. Základom tohoto konceptu je komunikácia medzi JMetrom a Iperfom.

Pre vytvorenie automatizovaného merania prenosových parametrov siete je potrebné vytvoriť program, ktorý bude sprostretkúvať komunikáciu medzi Apache JMetrom a Iperfom. Apache JMeter poskytuje viac možností ako rozšíriť svoje základé funkcie. Jednou z nich je vytvorenie kódu priamo v prostredi JMetru v Sampleri, bez vytvárania dotatočných súborov. Ďalšou možnosťou je vytvoriť kód samostatne v ľubovoľnom vývojovom prostredí a následne exportovať tento kód do súboru *.jar*. Z praktického dôvodu som sa rozhodol zvoliť druhú možnosť a teda vytvorený *.jar* súbor bude importovaný do Apache Jmetru.

6.1 Logika merania

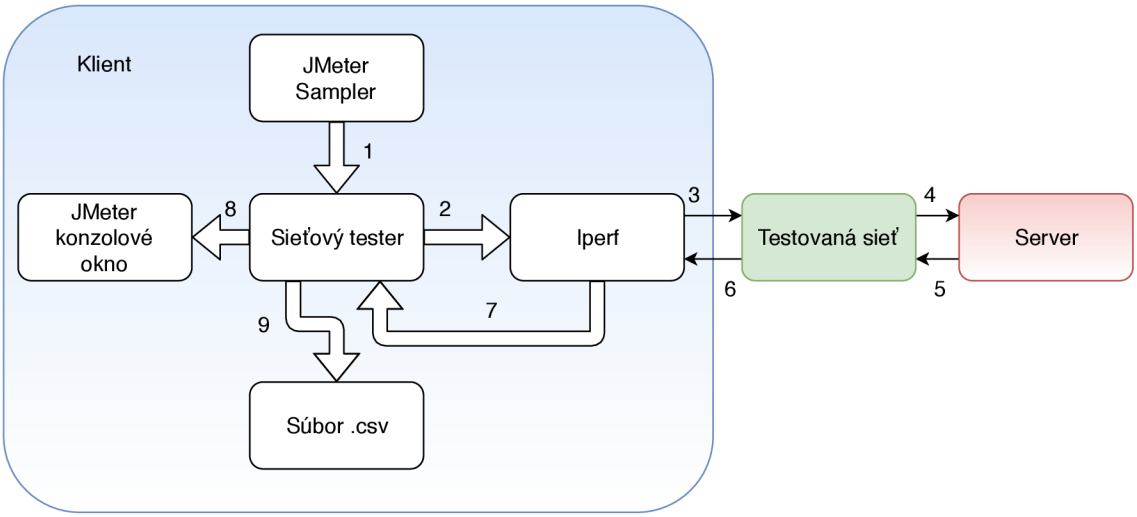
Ako už bolo uvedené v predošlých kapitolách, logika merania bude vychádzať z odporučenia RFC 2544.

Pred samotným zahájením merania bude potrebné nastaviť IP adresu serveru, voči ktorému bude realizované meranie a port, na ktorom bude meranie prebiehať.

Po zahájení merania budú posielané rámce o konkrétnej, užívateľovi známej veľkosti. Meranie priepustnosti UDP pre každú veľkosť rámca bude trvať 10 sekúnd. K tomuto meraniu bude využívaný program Iperf.

Po dokončení merania priepustnosti budú výsledky pre jednotlivé veľkosti rámcov zobrazené v *Mb/s*. Výsledok bude taktiež zahŕňať port, na ktorom bolo meranie uskutočnené, veľkosť rámcov, ktoré boli použité a celkové množstvo prenesených dát počas jednotlivých meraní.

- Sampler programu JMeter pošle príkaz pre spustenie softvérového testeru (1),
- softvérový tester pošle príkaz pre spustenie merania so zadanými parametrami do programu Iperf (2),
- Iperf zaháji meranie, vysiela datagramy do siete na zadanú IP adresu a port (3, 4),
- server posiela odpovede naspäť (5, 6),



Obr. 6.1: Návrh koncepcie testeru

- Iperf zo získaných informácií vypočíta priepustnosť siete a posiela výsledky do softvérového testeru (7),
- softvérový tester výsledky spracuje a posiela ich na zobrazenie v konzolovom okne JMetru (8),
- následne softvérový tester uloží výsledky do vytvoreného .csv súboru (9).

Logika merania spočíva v postupnom zväčšovaní datagramov UDP, ktoré slúžia pre realizáciu merania. Očakávaným výsledkom je, že čím väčšiu hodnotu budú mať prenášané datagramy, tým väčšiu priepustnosť siete nameriame. Dôvodom je, že každý paket bude obsahovať lepší pomer medzi záhlavím a objemom prenášaných dát v datagrame.

6.2 Overenie dostupnosti servera

Pred spustením merania je potrebné overiť dostupnosť strany serveru, ktorý je pre samotné testovanie potrebný. Po zadaní IP adresy bude prvá kontrola uskutočnená overením formátu zadanej adresy. Následne sa zistí dostupnosť zadanej adresy.

Ak bude formát adresy správny a adresa bude dostupná, bude následovať zadanie portu. Tu bude kontrolovaná hodnota portu, ak bude zadaná hodnota portu validná (v rozsahu 2–65 535), bude možné pristúpiť k samotnému meraniu priepustnosti.

6.3 Meranie priepustnosti UDP

Prvým krokom samotného merania je vysielanie rámcov zo strany klienta na server o veľkosti 64 B a bude zobrazený a uložený výsledok merania. Meranie pre každú

veľkosť rámca bude prebiehať v trvaní 10 sekúnd.

Ďalším krokom je postupné zvyšovanie veľkosti datagramu na hodnoty, ktoré sú uvedené v odporúčaní RFC 2544. Týmto spôsobom bude zmeraná priepustnosť UDP pre všetky potrebné veľkosti datagramov.

Pre potreby špecifického merania bude metodika rozšírená o možnosť voliť veľkosť rámcov podľa potreby užívateľa.

Výsledky merania budú zobrazené a uložené do súboru pre prípadné ďalšie spracovanie.

7 SOFTVÉROVÝ TESTER

Cieľom práce je vytvoriť softvérový tester pre meranie prenosových parametrov siete. Zdrojový kód vytvoreného rozšírenia do programu Apache JMeter je napísaný v programovacom jazyku Java.

Toto rozšírenie bude sprostretkúvať komunikáciu medzi programom Apache JMeter a Iperf. JMeter bude riadiacim prvkom merania. Výsledkom je možnosť automatizovaného merania priepustnosti siete na jednotlivých portoch podľa odporučenia RFC 2544 alebo možnosť merania s veľkosťami rámca, ktoré zadá užívateľ podľa potreby.

7.1 Princíp činnosti

Táto sekcia sa venuje popisu funkcie vytvoreného testeru. Zdrojový kód je napísaný v programovacom jazyku Java.

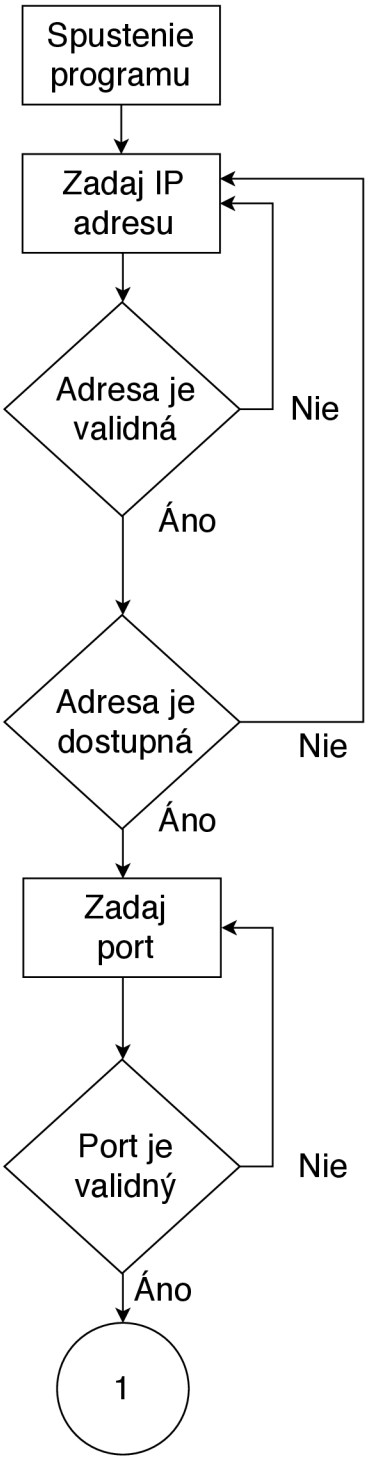
Vývojový diagram programu je na obrázkoch **Obr. 7.1** a **Obr. 7.2**.

Jednotlivé kroky merania:

1. Zadanie IP adresy servera.
2. Kontrola formátu IP adresy.
3. Kontrola dostupnosti IP adresy.
4. Zadanie čísla portu.
5. Kontrola portu.
6. Zvolenie automatického merania alebo meranie s vlastnou veľkosťou rámcov.
7. Zahájenie merania.
8. Výstup zobrazený v konzolovom okne.
9. Uloženie výsledkov merania do súboru.

Na začiatku samotného programu sú importované potrebné knižnice a definované premenné, ktoré budú využívané ďalej v programe.

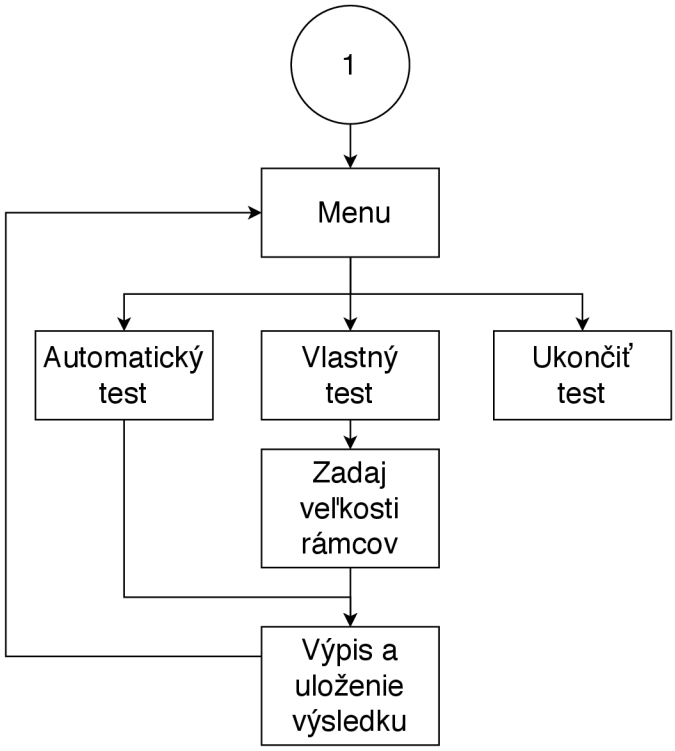
```
import java.io.*;
import java.net.InetAddress;
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.Scanner;
import java.util.regex.Pattern;
import java.sql.Timestamp;
```



Obr. 7.1: Vývojový diagram testeru.

7.1.1 Nastavenie IP adresy

Funkcie pre zadanie používanej IP adresy serveru a portu, na ktorom bude prebiehať merania sú zabezpečené funkciou Scanner. Užívateľ zadá potrebné údaje a tie sú uložené pre ďalšie spracovanie. Po zadaní IP adresy nastáva overenie formátu IP



Obr. 7.2: Vývojový diagram testeru.

adresy, teda či sa jedná o IPv4 adresu. V prípade, že IP adresa má validný formát, nastáva kontrola dostupnosti danej adresy. Ak niektorá z týchto podmienok nie je splnená, užívateľ je vyzvaný k opätovnému zadaniu IP adresy.

Fukcia pre zadanie IP adresy serveru a kontrola formátu adresy:

```
Scanner IPaddressReader = new Scanner(System.in);
  System.out.println("Enter␣server␣IP␣address␣(X.X.X.X):");
        ipAddress = IPaddressReader.nextLine();
        if (!isIpAddressVerified) {
          if (isIpValid(ipAddress) &&
          isIpReachable(ipAddress)) {
            isIpAddressVerified = true;
          } else {
            continue;
          }
        }
```

Aby mohlo byť samotné meranie uskutočnené, je potrebné overenť dostupnosť zadanej IP adresy:

```
public static boolean isIpReachable(final String ip) {
```

```
    try {
      InetAddress address = InetAddress.getByName(ip);
      if (address.isReachable(10000)) {
        return true;
      } else {
        System.out.println("IP␣address␣is␣not␣reachable!");
        return false;
      }
    } catch (Exception e) {
      e.printStackTrace();
    }
    return false;
  }
```

Pri overení valitdnosti IP adresy a portu bol používaný regex(regular expresion-regulárny výraz) definuje vzor vyhľadávania pre reťazce. Vyhľadávací vzor môže byť čokoľvek z jednoduchého znaku, pevného reťazca alebo komplexného výrazu obsahujúceho špeciálne znaky opisujúce vzor. Vzor definovaný regexom sa môže pre daný reťazec zhodovať raz alebo viackrát alebo vôbec.

7.1.2 Nastavenie portu

Po zadaní validnej a dostupnej IP adresy je užívateľ vyzvaný k zadaniu portu, tento port musí byť rovnaký, ako ten, na ktorom naslúcha server. Následuje opäť kontrola čísla portu.

Overenie validnosti zadaného portu zabezpečuje následovná funkcia:

```
public static boolean isPortValid(final String port) {
    if (PORT_PATTERN.matcher(port).matches()) {
      return true;
    } else {
      System.out.println("Port␣number␣is␣not␣valid!");
      return false;
    }
  }
```

7.1.3 Výber požadovaného merania

Následuje časť kódu, kde sú pomocou funkcie Switch ponúknuté možnosti uživateľovi. Prvou z nich je automatické meranie na základe odporučenia RFC 2544, čo znamená, že veľkosti rámcov sú preddefinované a uložené v Array liste. Druhou

možnosťou je nastavenie veľkosti rámcov, ktoré sú určené pre meranie ručne, teda užívateľ sám určí aké veľké budú používané rámce a koľko rôznych rámcov bude použitých. Od toho bude závisieť aj dĺžka samotného merania. Treťou možnosťou je ukončenie merania.

Po zvolení prvej možnosti je zahájené meranie prispustnosti UDP pomocou spustenia programu Iperf s konkrétnym príkazom s potrebnými parametrami pre meranie. Z výstupu programu Iperf je vyextraktovaná hodnota priepustnosti a zobrazená v konzolovom okne JMetru spoločne s údajmi o používanom porte. Tento proces sa opakuje pre všetky veľkosti rámcov.

Ukážka funkcie pre zvolenie automatického testu:

```
    case "1":
         System.out.println("Automatic testing");

         frameSizeList = Arrays.asList(64, 128, 256, 512,
                              1024, 1280, 1518);
         runCommands(r, ipAddress, port, frameSizeList);
         System.out.println("Press:\n1) For automatic
          testing\n2) For custom testing\n3) Quit");
         break;
```

Druhá možnosť umožňuje užívateľovi zvoliť veľkosti rámcov, jednotlivé hodnoty oddelí čiarkou. Pre zahájenie merania a tým ukončenie zadávania rámcov je potrebné stlačiť klávesu „S“, k čomu je užívateľ vyzvaný priamo v konzolovom okne.

7.1.4 Uloženie výsledkov merania

Výstup z programu Iperf je možné zobraziť pomocou parametrov *-y C* ako CSV výstup, čo znamená, že jednotlivé údaje sú oddelené čiarkou. To umožní jednoduchší spôsob rozdelenia výsledku pre zobrazenie požadovaných údajov. Zo zdrojového kódu programu Iperf je možné určiť, na ktorej pozícií sa nachádzajú aké parametre.

Pre zistenie, ktorý údaj čo znamená pri zjednodušenom vypise výsledku merania nám pomôže zistiť, aká je štruktúra výstupu programu Iperf, ktorú popisuje následovná časť kódu programu Iperf:

```
// UDP Reporting
        printf( reportCSV_bw_jitter_loss_format,
                timestamp,
                (stats->reserved_delay == NULL ? ",,,"
: stats->reserved_delay),
                stats->transferID,
```

```
stats->startTime,
stats->endTime,
stats->TotalLen,
speed,
```

Tento zjednodušený výstup programu Iperf je využitý práve v tomto vytvorenom rozšírení. Získané hodnoty môže užívateľ vidieť priebežne v konzolovom okne. Po ukončení merania je vytvorený súbor CSV, kde sú v tabuľke uložené namerané hodnoty spolu s veľkosťou použitých rámcov.

Parsovanie výsledku, na základe vyššie uvedeného výpisu programu Iperf, zabezpečuje následovný kód:

```
String[] parts = line.split(",");
String test = parts[8];
int result = Integer.parseInt(test);
```

Funkcia pre výpis výsledkov v konzolovom okne JMetru:

```
System.out.print("Throughput␣UDP:␣" + df.format(megaBits)
            + "␣Mb/s" + "\t" + "Frame␣size:␣" + frameSize
            + "␣B" + "\t" + "Port:␣" + port + "\n");
```

Parsovanie výsledku a prevod z *b/s* na *Mb/s* s dvomi desatinnými miestami:

```
String[] parts = line.split(",");
        int bits = Integer.parseInt(parts[8]);
        double megaBits = bits / 1048576.0;

        DecimalFormat df = new DecimalFormat("#.##");
```

Nastavenie názvu CSV súboru s časovým razítkom:

```
Timestamp timestamp = new Timestamp
(System.currentTimeMillis());
FileWriter writer = new FileWriter
("output_" + timestamp.getTime() + ".csv");
```

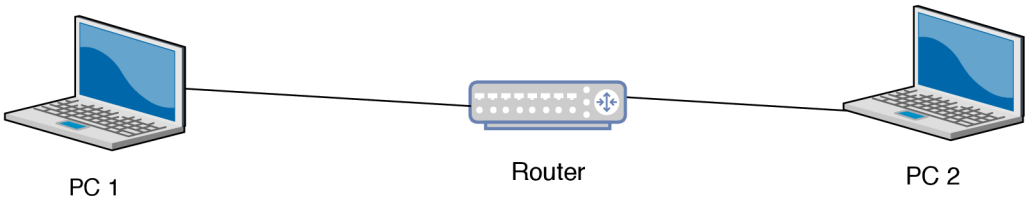
Uloženie výsledkov do CSV súboru zabezpečuje následovná časť kódu:

```
CSVwriter.writeLine(writer,
Arrays.asList(String.valueOf(frameSize),
String.valueOf(bits), String.valueOf(port)));
```

Názov súboru je rozšírený o časové razítko, čo zaistí, že pri opakovaých meraniach sa jednotlivé súbory neprepíšu, pretože každý súbor má jedinečný názov. Ďalšou výhodou takéhoto spôsobu tvorby názvu súboru je, že užívateľ nemusí pri každom meraní zadávať nový názov súboru, kam sa budú dáta ukladať.

7.2 Príprava pracoviska

K overeniu funkčnosti testera budeme potrebovať jeden počítač (PC1), na ktorom bude nainštalovaný ApacheJMeter s vytvoreným rozšírením a Iperf. Na druhom počítači (PC2) bude nainštalovaný program Iperf, ďalej budeme potrebovať zariadenie, ktoré chceme podrobiť testu, napríklad router alebo switch. Obidva počítače budú prepojené s testovaným zariadením pomocou ethernetového káblu.



Obr. 7.3: Topológia testovacej siete

Na počítači PC1 spustíme program Apache JMeter a naimportujeme doň vytvorené rozšírenie. Počítač PC2 bude využívaný ako server a teda spustíme program Iperf v režime server pomocou príkazu *iperf -s* (prípadne pomocou parametru *-p* nastavíme číslo portu).

7.2.1 Inštalácia programu Iperf

Program Iperf je voľne dostupný k stiahnutiu na stránke `https://iperf.fr/iperf-download.php`. Ja som sa z dôvodu jednoduchšieho ovládania pre meranie priepustnosti vyšších hodnôt rozhodol pre Iperf verzie 2. Dôvod je taký, že novšie verzie potrebujú väčšie množstvo nadviazaných spojení pre zmeranie priepustnosti vyšších hodnôt, čo môže mať za následok menej presné výsledky z dôvodu zaokrúhlovania výsledku a podobne.

Po stiahnutí je potrebné súbor rozbaliť do ľubovoľného priečinka. Tento postup je potrebné spraviť na oboch koncových staniciach.

7.2.2 Inštalácia programu Apache JMeter

Tento program je možné taktiež voľne dostupný k stiahnutiu na stránke `https://jmeter.apache.org/`. Tu si užívateľ vyberie konkrétnu verziu vhodnú pre jeho operačný systém. Po stiahnutí je potrebné opäť rozbaliť stiahnutý súbor. Program je tým pripravený na používanie, nie je nutná ďalšia inštalácia.

Program Apache JMeter stiahneme do koncovej stanice, ktorú chceme používať ako klient, čo znamená, z ktorej bude realizované meranie. Program spustíme otvorením súboru jmeter.bat v zložke bin.

7.2.3 Importovanie rozšírenia do JMetra

Vytvorené rozšírenie je potrebné importovať do programu Apache JMeter. Po spustení programu JMeter klikneme na ikonu Test plan, dolu na obrazovke zvolíme v Menu Add directory or jar to classpath možnosť Browse a nájdeme vytvorený súbor, následne potvrdíme „OK“.

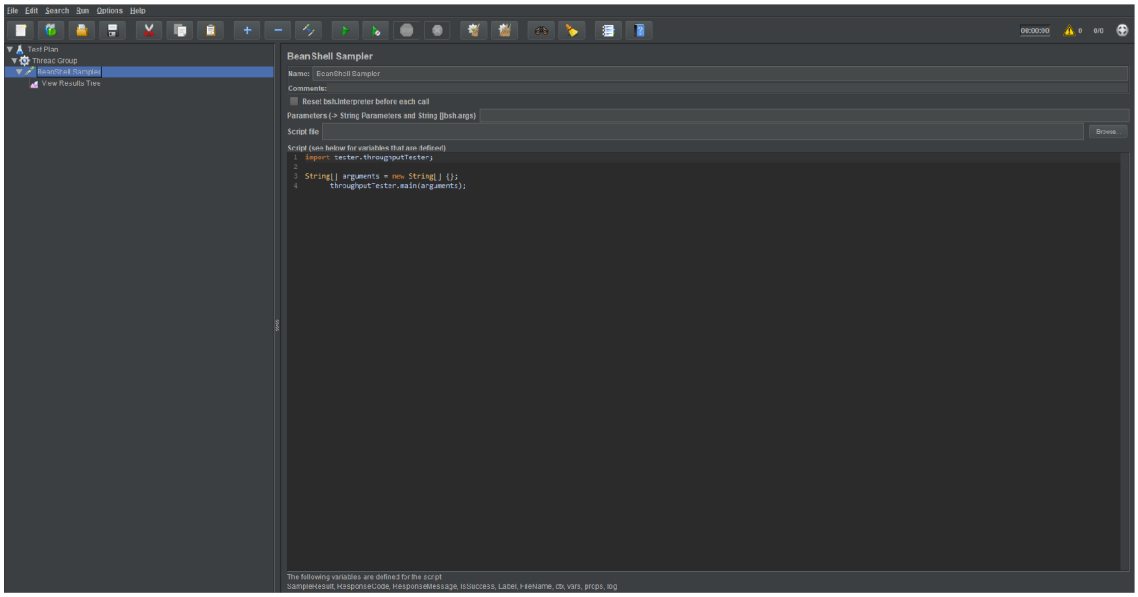
Ďalším krokom je vytvorenie nového vlákna. Kliknutím pravého tlačítka myši vľavo na ikonu *Test plan-> Add-> Threads-> Thread group*.

Potom pridáme Sampler klinkutím pravého tlačítka myši na *Thread group-> Add-> Sampler-> Bean Shell Sampler*. V časti Scripts vložíme časť kódu, ktorá nám umožní spustenie importovaného rozšírenia.

Výpis 7.1: Nastavenie pre použitie vytvoreného rozšírenia.

```
import tester.throughputTester;

String[] arguments = new String[] {};
      throughputTester.main(arguments);
```



Obr. 7.4: Nastavenie JMetru pre použitie rozšírenia.

Pre následnovné používanie programu Iperf na počítači PC1 skopírujeme obsah rozbaleného súboru Iperf do podzložky bin v programe Apache JMeter.

7.3 Testovacie meranie

Táto sekcia sa venuje overeniu fuknčnosti vytvoreného testeru.

7.3.1 Topológia testovacej siete

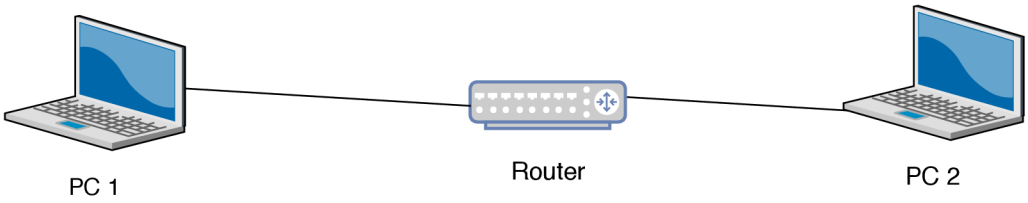
Pre overenie funkčnosti vytvoreného testeru som navrhol testovaciu topológiu siete, ktorá pozostáva z dvoch počítačov, ktoré sú pomocou sieťového kábla prepojené cez router. Na oboch použitých počítačoch bol nainštalovaný operačný systém Windows 10. Použitý router mal výrobcom udanú maximálnu priepustnosť 100 *Mb/s*.

Nastavenia PC1:

- Operačný systém Windows 10,
- vypnutý firewall,
- nainštalovaný Iperf verzie 2,
- JMeter s importovaným rozšírením.

Nastavenia PC2:

- Operačný systém Windows 10,
- vypnutý firewall,
- Iperf verzie 2 v režime server.



Obr. 7.5: Topológia testovacej siete

7.3.2 Zahájenie merania

Na počítači PC2 spustíme program Iperf v režime server s požadovaným portom, na ktorom chceme meranie uskutočniť. Napríklad pre meranie na východzom porte 5001 pomocou príkazu *iperf -s -p 5001*.

Na počítači PC1, kde je nainštalovaný program Iperf a Apache JMeter spustíme JMeter v zložke kde máme rozbalený JMeter prejdeme do podzložky bin a spustíme súbor jmeter.bat. Prejedeme do vytvoreného testovacieho prostredia (postup vytvorenia v predošlej sekcií).

Po úspešnom nastavení oboch koncových staníc je možné realizovať meranie. Na počítači PC1 spustíme meranie kliknutím na tlačidlo Run v hornej lište a v konzolovom okne sa nám zobrazí následovné menu:

Výpis 7.2: Nastavenie IP adresy a portu po spustení testu.

```
Enter IP address of server (X.X.X.X):                    1
192.168.1.108                                            2
```

```
Enter port number (set the same number on server):
5001
```

Zadáme IPv4 adresu počítača PC2 a stlačíme ENTER. Následne je potrebné zadať číslo portu, ktorý bude využitý pre meranie. Potom sa dostaneme k ponuke:

Výpis 7.3: Výber požadovaného testu.

```
Press:
1) For automatic test
2) For custom test
3) Quit
```

Z tohoto menu si vyberieme či sa bude jednať o automatizované meranie priepustnosti UDP podľa odporučenia RFC 2544, o manuálne meranie, kde zvolíme veľkosť testovacích rámcov alebo chceme meranie ukončiť.

Výstupná hodnota priepustnosti z programu Iperf je v bitoch za sekundu. Preto je potrebné túto hodnotu prepočítať na megabity za sekundu, kvôli predpokladaným vyšším hodnotám priepustnosti. Do výstupného súboru CSV sú však hodnoty priepustnosti v bitoch za sekundu.

7.3.3 Automatické meranie

Pri voľbe automatického merania sú použité rámce podľa odporúčania RFC 2544. Postupne je realizované meranie so všetkými veľkosťami rámcov, ktoré predpisuje toto odporúčanie. Dĺžka merania pre každú veľkosť rámca je 10 sekúnd.

Po ukončení merania sú výsledky zobrazené v konzolovom okne a následne exportované do CSV súboru. V tomto súbore sú uvedené hodnoty rámcov pre každé meranie, nameraná priepustnosť a port, na ktorom bolo meranie realizované.

Následne sa v konzolovom okne opäť zobrazí menu, kde užívateľ znovu môže vybrať ďalšie meranie alebo ukončiť testovanie.

Výpis 7.4: Výstup v konzolovom okne JMetru automatického merania.

```
Automatic testing
Starting test...
Throughput UDP: 3,9 Mb/s    Frame size: 64 B    Port: 5001
Throughput UDP: 12,7 Mb/s   Frame size: 128 B   Port: 5001
Throughput UDP: 31,4 Mb/s   Frame size: 256 B   Port: 5001
Throughput UDP: 65,8 Mb/s   Frame size: 512 B   Port: 5001
Throughput UDP: 72,1 Mb/s   Frame size: 1024 B  Port: 5001
Throughput UDP: 88,1 Mb/s   Frame size: 1280 B  Port: 5001
Throughput UDP: 94,6 Mb/s   Frame size: 1518 B  Port: 5001
```

Tab. 7.1: Tabuľka nameraných hodnôt pri automatickom teste

Veľkosť datagramu [*B*]	64	128	256	512	1024	1280	1518
Priepustnosť UDP [*Mb/s*]	3,9	12,7	31,4	65,8	72,1	88,1	94,6

Výpis 7.5: Ukážka výsledkov uložených do CSV súboru.

```
Frame size,Throughput,Port
64,2522245,5001
128,4729139,5001
256,8450231,5001
512,13996460,5001
1024,21270977,5001
1280,23914558,5001
1518,18191692,5001
```

7.3.4 Vlastné meranie

Pri voľbe vlastného merania je užívateľ vyzvaný k tomu, aby zadal veľkosť prvého rámca pre testovanie. Následne môže zvoliť veľkosť rámca pre ďalšie meranie alebo stlačením klávesy „S“ ukončí zadávanie rámcov a spustí meranie priepustnosti pre zvolené veľkosti rámcov.

Po ukončení merania sú výsledky zobrazené v konzolovom okne a následne exportované do CSV súboru tak isto ako pri automatickom meraní. V tomto súbore sú uvedené hodnoty rámcov pre každé meranie, nameraná priepustnosť a port, na ktorom bolo meranie realizované.

Následne sa v konzolovom okne opäť zobrazí menu, kde užívateľ znovu môže vybrať ďalšie meranie alebo ukončiť testovanie.

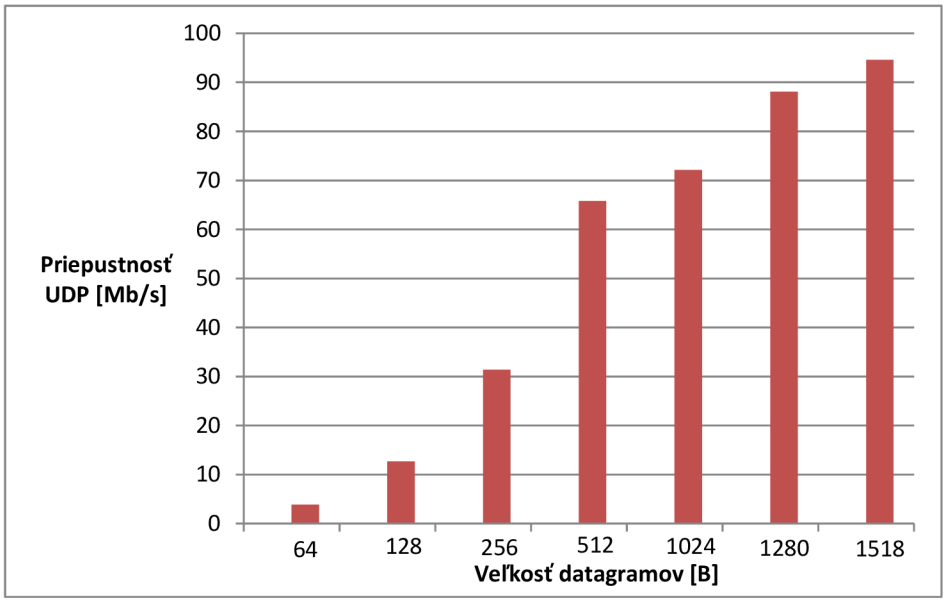
Výpis 7.6: Výstup v konzolovom okne JMetru automatického merania.

```
Custom testing
Starting test...
Throughput UDP: 4,1 Mb/s    Frame size: 69 B    Port: 5001
Throughput UDP: 65,5 Mb/s   Frame size: 512 B   Port: 5001
Throughput UDP: 98,3 Mb/s   Frame size: 1440 B  Port: 5001
```

7.3.5 Zhodnotenie nameraných hodnôt

Z výsledkov automatického testu vyplýva, že najväčšia priepustnosť UDP bola nameraná pri najväčšej veľkosti datagramu podľa odporučenia RFC 2544 (1518 *B*) a naopak najnižšia priepustnosť bola nameraná pri najnižšej veľkosti datagramu

podľa tohoto odporučenia. Maximálna priepustnosť siete je 94,6 *Mb/s*. Najnižšia priepustnosť siete pri veľkosti datagramu 64 *B* je 3,9 *Mb/s*.



Obr. 7.6: Graf nameraných hodnôt

Na základe výsledkov môžme potvrdiť, že pri vyšších veľkostiach datagramu bol dosiahnutý priaznivejší pomer riadiacich bitov k dátam a tým pádom väčšie množstvo prenesených dát a vyššia priepustnosť siete.

8 ZÁVER

Práca sa zaoberá problematikou merania prenosových parametrov dátových sietí. V prvej časti práce je popísaný sieťový model TCP/IP, ktorý je referenciou pre túto prácu. Následne sú popísané jednotlivé parametre dátových sietí a metodiky pre ich meranie. Popísané sú odporučenia RFC 2544, RFC 6349 a ITU-T Y.1564. Pre každé z odporúčaní je analyzovaná metodika merania.

Výstupom práce je navrhnutá metodika pre meranie prenosových parametrov sietí. Jedná sa o metodiku pre meranie priepustnosti UDP. Metodika pre meranie priepustnosti vychádza z odporučenia RFC 2544. Je v nej rozpísaný postup, ktorým bude realizované meranie.

Návrhu koncepcie testeru sa venuje následovná kapitola práce. Tester pozostáva z dvoch programov a to JMeter a Iperf. Tieto dva programy navzájom komunikujú pomocou vytvoreného rozšírenia. V sampleri JMetru je importované vytvorené rozšírenie, ktorý obsahuje samotnú logiku merania, ktorá je vytvorená na základe navrhnutej metodiky merania. Z tohto programu sú posielané príkazy pre meranie, ktoré uskutočňuje Iperf, z ktorého sú výsledky následne zobrazené v prostredí JMetru. Užívateľ si môže zvoliť či bude meranie uskutočnené na základe odporučenie RFC 2544 alebo zvolí možnosť vlastného merania, kde zvolí veľkosti posielaných rámcov.

Ďalším cielom práce je vytvorenie sofvérového testeru, ktorý dokáže merať prenosové parametre siete. Základom tohto testeru je už existujúci program Apache JMeter, do ktorého je importované vytvorené rozšírenie. S týmto rozšírením sme schopní zmerať priepustnosť UDP pomocou postupu, ktorý vychádza z odporučenia RFC 2544 alebo meranie môžme upraviť podľa vlastnej potreby.

Pre účel overenia správnej funkčnosti návrhu bola vytvorená testovacia topológia siete, ktorá pozostáva z dvoch staníc a jedného routera, ktorý tieto stanice spája. Na tejto topológií boli realizované testovacie merania, ktorých výsledky sú spracované do tabuľky. Z výsledkov testovacieho merania je možné zhodnotiť, že čím viac sa blížila veľkosť posielaných rámcov k maximálnej hodnote ethernetového rámca, tým väčšiu priepustnosť sme namerali. Je to spôsobené tým, že bity určené na réžiu mali menší podieľ na celkovom množstve prenesených dát a tým bol prenos efektívnejší.

LITERATÚRA

[1] BRADNER, S. *RFC 2544 – Benchmarking Methodology for Network Interconnect Devices.* IETF, 1999.

[2] CHIMENTO P. *IETF RFC 5136.* [online]. JHU Applied Physics Lab, 2008. Dostupné z URL: <hhttps://tools.ietf.org/html/rfc5136>.

[3] CONSTANTINE B. *IETF RFC 6349.* [online]. JDSU, 2011. Dostupné z URL: <http://https://tools.ietf.org/html/rfc6349,2011>.

[4] ČESKÝ TELEKOMUNIKAČNÍ ÚŘAD. *Obecná pravidla a doporučení pro využívání řízení datového provozu .* 2013.

[5] ČESKÝ TELEKOMUNIKAČNÍ ÚŘAD. *Měření datových parametrů sítí pomocí TCP protokolu.* 2014.

[6] FEIBELL, W. *Encyklopedie počítačvých sítí.* Computer press, 1996. ISBN 80-85896-67-2.

[7] IETF. *Benchmarking Methodology for Network Interconnect Devices: RFC 2544.* [online]. Network Working Group 1999. Dostupné z URL: <http://www.ietf.org/rfc/rfc2544.txt>.

[8] Emily, H. *Apache JMeter: A practical beginner's guide to automated testing and performance measurement for your websites.* Packt Publishing Ltd, 2013. ISBN 978-1-847192-95-0

[9] ITU-T group. *ITU-T Y.1564 : Ethernet service activation test methodology.* [online]. Geneva, 2011. Dostupné z URL: <http://www.itu.int/rec/T-REC-Y.1564/en.>.

[10] ERINLE, B. *Performance Testing with JMeter 2.9.* Packt Publishing Ltd, 2013.

[11] JAREŠ, P. *Diagnostika přenosových systémů a sítí využívajících technologii Ethernet.* České vysoké učení technické v Praze Fakulta elektrotechnická

[12] JEŘÁBEK, J. *Komunikační technologie.* Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací, 2013.

[13] Cavanaugh, M. *IP Telephony Self-Study: Cisco Qos Exam Certification Guide, Second Edition.* 2004. ISBN 1-58720-124-0.

[14] LEINEN, S.. *Iperf Tool.* [online]. 2014. Dostupné z URL: <http://kb.pert.geant.net/PERTKB/IperfTool>.

[15] MATHIS M., HEFFNER J. *IETF RFC 4821.* [online]. PSC, 2007. Dostupné z URL: <https://tools.ietf.org/html/rfc4821>.

[16] McCANN J. *IETF RFC 1981.* [online]. Digital Equipment Corporation, 1996. Dostupné z URL: <https://tools.ietf.org/html/rfc1981>.

[17] MOGUL J. *IETF RFC 1191.* [online]. DECWRL, 1990. Dostupné z URL: <https://tools.ietf.org/html/rfc1191>.

[18] NOVOTNÝ, V. *Architekura sítí.* Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačných technológií. Ústav telekomunikácií, 2012.

[19] STOVER, M.; LIN, P. *How to Write a plugin for JMeter Apache.* [online]. 2012. Dostupné z URL: <http://jmeter.apache.org/extending/jmeter_tutorial.pdf>.

[20] ŠTRAUCH, A.. *Iperf: měření rychlosti spojení.* [online]. 2012. Dostupné z URL: <http://www.root.cz/clanky/iperf-mereni-rychlosti-spojeni/>.

[21] WANG, Z. *Internet QoS: architectures and mechanisms fo Quality od Service.* Morgan Kaufmann, 2001.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

QoS	Quality of Services- Kvalita služieb
TCP	Transport Control Protocol
UDP	User Datagram Protocol
RTT	round-trip time- rozdiel času od odoslania prvého bitu príjemcovi do doručenia posledného bitu príslušného TCP acknowledgement
BB	Bottle-neck Bandwidth- najnižšia hodnota šírky pásma celej meranej trasy

ZOZNAM PRÍLOH

A OBSAH PRILOŽENÉHO CD

- Jar súbor vytvoreného rozšírenia *tester.jar*
- Súbor *tester.zip*, ktorý obsahuje zdrojový kód rozšírenia
- Súbor *readme.txt* s návodom na stiahnutie potrebných programov a importovanie rozšírenia
- Súbor *manual.txt* s návodom na ovládanie testera