Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Měření výkonnosti síťových zařízení

Plzeň 2013 Jaroslav Pouzar

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 8. května 2013

Jaroslav Pouzar

Abstract

One of the main objectives of this work is to find and compare existing tools or applications which are commonly used for testing the performance of network devices and to analyze their usage in possible testing scenarios. There are also discussed the issues of designing testing scenarios using these tools. This theses continues by performing the tests previously designed and finally links to the sum up capabilites of measuring tools already available. Based on the results and experience gained I am going to build my own testing utility. This application should meet suggested measuring tool requirements and provide suitable results for evaluating the appropriateness of deployment home or office networking devices.

Abstrakt

Jedním ze základních cílů této práce je vyhledat a porovnat existující nástroje a aplikace, které se běžně používají pro testování výkonu síťových zařízení a analyzovat jejich použití v možných testovacích scénářích. Práce se dále věnuje problematice navrhování testovacích scénářů pomocí těchto nástrojů. Na základě dosažených výsledků a získaných zkušeností s testováním se chystám vytvořit vlastní testovací nástroj. Funkcionalita tohoto nástroje by měla vycházet ze získaných zkušeností s prací s dostupnými nástroji. Vytvořený program by měl podporovat rozhodování při posouzení vhodnosti nasazení síťových zařízení do konkrétních provozních prostředí.

Obsah

1 Úvod **1**

2 Teoretická část **2**
2.1 Seznámení s dostupnými testovacími nástroji 2
2.2 Možnosti zjištěných testovacích nástrojů 2
2.3 Popis vybraných testovacích nástrojů 5
2.4 Vybraná testovaná zařízení 10
2.5 Návrh a provedení testovacích scénářů 11
2.6 Úprava zaměření testovacích scénářů 16

3 Realizační část **19**
3.1 Návrh testovací aplikace 19
3.2 Realizace testovací aplikace 20
3.3 Návrh testovacích scénářů pro aplikaci 25
3.4 Realizace testů na vybraných zařízeních 30
3.5 Porovnání aplikace s vybranými existujícími nástroji 41

4 Závěr **42**
4.1 Posouzení vhodného nasazení síťových prvků 42
4.2 Zhodnocení získaných informací a přínosů aplikace 43

A Naměřené hodnoty **47**

B Uživatelská dokumentace **54**
B.1 Prerekvizity 54
B.2 Ovládání aplikace 54

1 Úvod

Téma této práce vzniklo na základě potřeb ověření skutečného výkonu domácích síťových zařízení. Některá tato zařízení vykazují při standardních konfiguracích velmi významný pokles přenosové rychlosti nebo zvýšenou latenci odpovědí na požadavky odeslané do internetu. Předmětem této práce je proto vývoj vlastní testovací aplikace určené k měření parametrů datových toků přes síťová zařízení nasazená převážně v domácích podmínkách nebo na nekritických segmentech počítačové sítě.

Abych získal potřebné znalosti a zkušenosti z této problematiky, musím se nutně nejprve zabývat průzkumem dostupných a používaných měřících nástrojů, kterými lze parametry síťové infrastruktury měřit. Následně se budu zabývat navrhováním testovacích scénářů se zapojením vybraných zařízení a provedením testů s využitím dostupných nástrojů. Naměřené hodnoty budou poté vyhodnoceny a interpretovány. Diskutována bude i informační hodnota provedených testů.

Po vyhodnocení získaných poznatků budou formulovány jasné požadavky na vyvíjenou aplikaci. Pro aplikaci bude nutné navrhnout vhodné testovací scénáře a provést měření s vyhodnocením výsledků.

2 Teoretická část

2.1 Seznámení s dostupnými testovacími nástroji

Vyhledávání jsem zahájil průzkumem více či méně známých internetových magazínů [Schön(2012)], [Higgins(2010a)], kde autoři recenzují nová síťová zařízení určená pro použití v domácím prostředí. Zde jsem se zajímal o metodiky a nástroje, jaké autoři recenzí při testování a měření používají.

Nebylo v mých silách se detailně seznámit se všemi dostupnými programy, protože (jak jsem zjistil) nástrojů zaměřených na generování/měření síťového provozu jsou k dispozici desítky. Při procházení dokumentací a vlastních testech jednotlivých programů jsem sledoval, jestli jsou placené nebo neplacené, jakou mají výsledky vypovídající hodnotu, jestli poskytují výstup výsledků zpracovaný do grafické podoby (tj. pro uživatele přehlednější), jak aktivní je komunita, která nástroj vyvíjí. Často jsem se setkal s velice zajímavými projekty, které mě oslovily netradiční nabídkou funkcí (například generování provozu dle statistických modelů v programu NetSpec [Jonkman(2012)]), ale jejich vývoj bohužel skončil před několika lety. Závěrem mého průzkumu tak bylo zjištění, že profesionální nástroje zabývající se touto problematikou jsou již téměř výhradně placené - např. IxChariot [Ixia(2012)]. Tyto nástroje nabízejí detailnější konfiguraci protokolů na vyšších vrstvách, poskytují propracované grafické uživatelské prostředí, disponují již předpřipravenými testovacími scénáři a umožňují podrobnější analýzu a zpracování přeneseného datového toku.

2.2 Možnosti zjištěných testovacích nástrojů

Po průzkumu dostupných testovacích nástrojů jsem sestavil přehled důležitých a užitečných testů měření síťových parametrů:

Měření maximální propustnosti – metodika testu se odvíjí od použitého protokolu 4.vrstvy ISO/OSI modelu. Při tomto testu je generován maximální možný provoz, který je následně přenášen přes testované zařízení. U protokolu TCP[1] se měří skutečně dosažená rychlost přenosu, kterou regulují TCP congestion avoidance algoritmy [Allman et al.(2009)Allman, Paxson„ Blanton], jež svými zásahy zamezují navázané relaci v přetížení přenosových kapacit dostupné linky. Protokol TCP jako takový zajišťuje pak spolehlivost doručení odeslaných zpráv. UDP[2] protokol naopak nebrání odesílateli odeslat větší množství dat, než síťová infrastruktura mezi komunikujícími stanicemi skutečně zvládne přenést. Proto je nutné během testu na straně příjemce detekovat případnou ztrátu odeslaných datagramů a jako maximální propustnost síťové cesty mezi 2 zařízeními bereme takový maximální datový proud, jenž ztratí po cestě jen minimální procento odeslaných datových jednotek resp. úspěšně přenese všechny pakety, které odesílatel odeslal. Nástrojem pro testování maximální kapacity linky v případě použití obou ze zmíněných protokolů může být program iperf.

Replikace reálného provozu – síťový provoz je generován síťovými zařízeními resp. aplikacemi, programy a procesy, jež pracují na pozadí běžícího operačního systému nebo přímo interagují s uživatelem a potřebují si pro svou činnost vyměňovat data s jiným zařízením. Protokoly zprostředkovávající takovou komunikaci mají různou podobu a náročnost na připojenou linku. To samé platí o síťových tocích, jež jsou velmi specifické a mohou se například lišit ve velikosti datových jednotek či frekvenci vyměňování řídících a datových zpráv. Replikace provozu funguje na principu odchycení uskutečněné síťové komunikace. Tu uložíme do souboru standardizovaného formátu a můžeme upravit hlavičky jednotlivých protokolů a komunikaci se stejnou či upravenou rychlostí zopakovat a vyslat znovu na síť. Jedním z dostupných programů, kterým lze takové přehrání provozu uskutečnit, je Tcpreplay. Tento nástroj funguje na principu injekce paketů [Wikipedia(2013)] mezi TCP/IP zásobník a vrstvu OS [Tcpreplay(2013)]. Smysl testu pak spočívá v pozorování chování síťových zařízení a datových toků, kdy se testuje, zda je konkrétní síťová infrastruktura schopna přenést určitý objem komunikace v daném čase a to bez pozorovaných ztrát.

[1] Transmission Control Protocol
[2] User Datagram Protocol

Měření latence – latenci označujeme jako zpoždění (ve většině případů udáváno v milisekundách), které uplyne mezi odesláním paketu z jednoho síťového zařízení a příjmem téhož paketu druhým zařízením. Tehdy mluvíme o „one-way“ latenci neboli jednosměrném zpoždění. V praxi je však užitečnější znát dobu, za kterou se nám vrátí odpověď od odeslání požadavku. Takový časový úsek bychom označili jako „two-way“(obousměrnou) latenci, obecně známou jako RTT[3]. Pro měření RTT používáme sadu protokolů z rodiny ICMP[4].

Sledování kvality – Jitter, pojem někdy známý spíše jako „Packet Delay Variation“ je jedním z ukazatelů QoS[5] a v souvislosti s ním mluvíme o kolísání latencí sítě. Síť s konstantní dobou odezvy má hodnotu *jitter* rovnou 0 a vypovídá o stavu, kdy časová prodleva mezi libovolnými 2 odeslanými pakety je shodná s časovou prodlevou, s jakou jsou tyto pakety doručeny příjemci. V souvislosti s odchylkou mluvíme o disperzi paketů nebo naopak o shlukování. [Demichelis – Chimento(2002)Demichelis, Chimento] Výskyt těchto jevů je velmi nežádoucí například během přenosu hlasových služeb.

Příklad **rozptylu**: V čase t_0 odešle stanice **A** paket P1 stanici **B**, v čase t_0+10ms odesílá stanice **A** paket P2 stanici **B**. Stanice **B** v čase t_0+20ms přijímá paket P1 v čase t_0+35ms paket P2. Mezi odesláním P1 a P2 uplynulo 10ms, stanice **B** registruje rozdíl 35-20=15ms.

Příklad **shlukování**: V čase t_0 odešle stanice **A** paket P1 stanici **B**, v čase t_0+10ms odesílá stanice **A** paket P2 stanici **B**. Stanice **B** v čase t_0+20ms přijímá paket P1 v čase t_0+25ms paket P2. Mezi odesláním P1 a P2 uplynulo 10ms, stanice **B** registruje rozdíl pouze 25-20=5ms.

Měření spolehlivosti – v souvislosti s již od návrhu nespolehlivým přenosem dat na bázi protokolu UDP mluvíme o ztrátovosti paketů. Klient v praxi nemůže ovlivnit rychlost, jakou k němu server odesílá proud – „stream“ UDP datagramů (například televizní vysílání). Záleží tedy na síťových prvcích a kapacitách přenosových linek po cestě od serveru ke klientovi, zda každý jednotlivý paket dosáhne svého cíle. V praxi může nastat, že zdroj UDP streamu vysílá datagramy tak rychle, že směrovač na trase nemusí být schopen všechny pakety v reálném čase směrovat a začne určitou část provozu

[3]Round Trip Time
[4]Internet Control Message Protocol
[5]Quality of Service

zahazovat. Takový jev má pak velmi nepříznivý výsledek na kvalitu služby, kterou server poskytuje. V případě televizního vysílání může například docházet k výpadkům obrazu/zvuku. Jako měřící nástroj ztrátovosti paketů lze využít například zmíněný iperf.

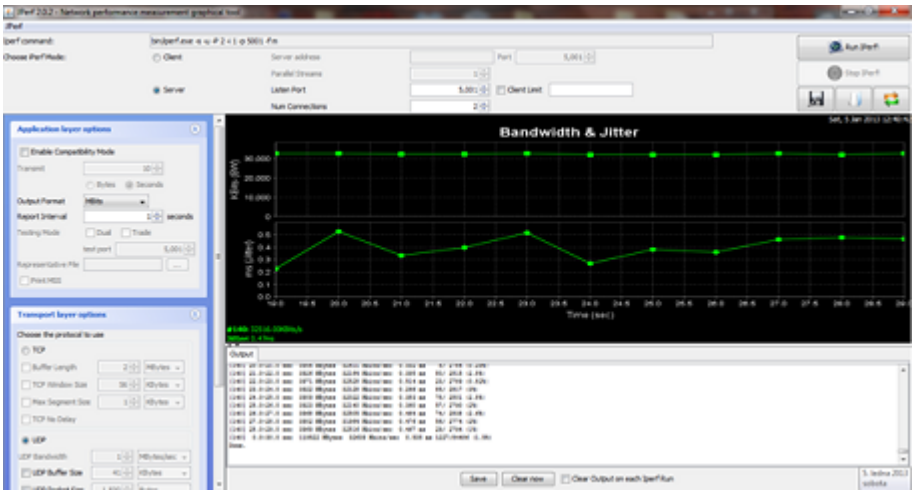
Měření maximálního počtu spojení – snahou tohoto testu je zjistit, kolika relacím je umožněno ve velmi krátkém časovém intervalu navázat spojení přes testované zařízení. Test provádějí 2 stanice, kdy jedna figuruje jako klient, který generuje a odesílá požadavky ze zadaného rozsahu portů na server, jenž obratem odpovídá, aby potvrdil klientovi, že je spojení funkční. Pro tento test se používá z důvodů rychlosti protokolu UDP, proto si nesmíme zaměňovat výše zmíněný pojem relace s TCP relací. Relacemi v tomto kontextu rozumíme například záznamy v NAT[6] tabulce směrovače.

2.3 Popis vybraných testovacích nástrojů

Iperf - Velmi často zmiňovaný a používaný CLI nástroj a to zejména díky tomu, že je zdarma a poskytuje možnosti základního testování sítě. Program podporuje klient-server architekturu a je kromě měření maximálního datového toku schopen evidovat ztrátovost paketů při UDP proudu, měřit jitter a generovat/zachytávat multicastový provoz. Program umožňuje zápis výsledků testů do textového souboru, ale nepodporuje grafickou vizualizaci získaných výsledků. Poslední verze (2.0.5) byla vydána a uveřejněna na oficiálních stránkách projektu na portále v červenci 2010. [Sourceforge(2010)]

JPerf – Jedná se o nástroj s grafickým ovládacím rozhraním, které má za úkol zpřehlednit a usnadnit uživateli ovládání výše zmíněné utility iperf, neboť právě ta je jádrem této aplikace. GUI zajišťuje spuštění nástroje iperf se zadanými parametry a následně zobrazuje výsledky získané z měření do grafu umístěného v rámci okna aplikace a umožňuje ukládat testovací konfigurace pro budoucí opětovné použití. Jde o multiplatformní nástroj napsaný v jazyce JAVA.

[6]Network Address Translation

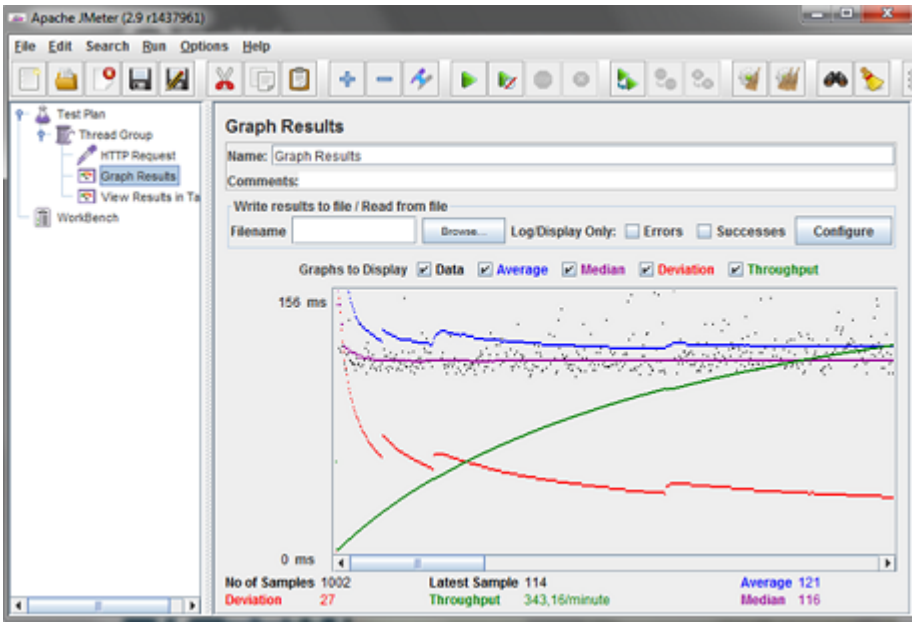


Obrázek 2.1: Měřící nástroj JPerf při běhu

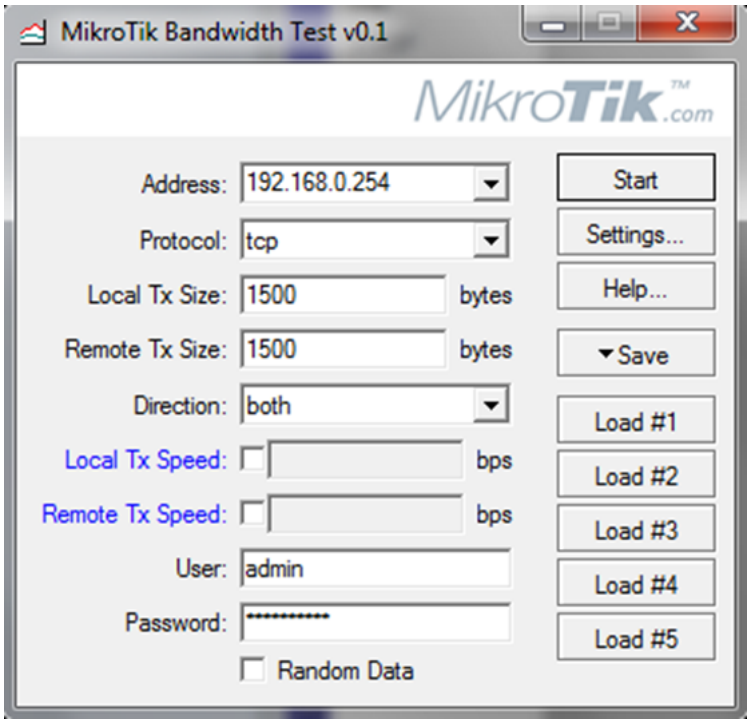
JMeter – nástroj určený pro testování aplikací, nikoliv pro měření parametrů přenosů. Původním zaměřením byl zacílen na webové aplikace, později se rozrostl o další protokoly (FTP[7], LDAP[8], databáze, shell skripty, poštovní protokoly). V případě provedení testu ve scénáři, kde je síťová infrastruktura nejslabším článkem tj. výkon serveru a klienta převyšuje kapacitu spojovací linky (včetně aktivních zařízení v cestě), lze využít tento nástroj jako simulátor generující reálný síťový provoz a naměřené hodnoty datového toku bychom za určitých okolností mohli považovat jako hodnoty prahové – na daném spoji maximální. Reálným síťovým provozem rozumíme takový, který bychom byli schopni zachytit při běžném provozu na testované lince. Zmíněn je zde, protože jak se později v realizační části ukáže, bude výsledná aplikace vycházet z původní idey tohoto nástroje.

Btest – proprietární nástroj integrovaný do každého zařízení výrobce MikroTik. To je tak schopné provádět jednoduchý zátěžový test pro zvolený konkrétní zvolený směr (download x upload) nebo oba směry najednou. Podporuje UDP i TCP relace a velkou výhodou je možnost provedení tohoto testu mezi jakýmikoliv dvěma MikroTik zařízeními, neboť všechny tyto prvky disponují nástrojem btest (každé zařízení může působit jako klient nebo server).

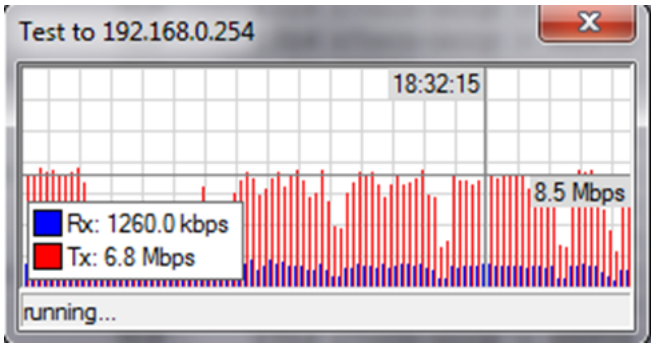
[7]File Transfer Protocol
[8]Lightweight Directory Access Protocol



Obrázek 2.2: Vizualizace hodnot jednoduchého http testu v grafickém prostředí programu JMeter



Obrázek 2.3: MikroTik Bandwidth Test – konfigurační rozhraní



Obrázek 2.4: MikroTik Bandwidth Test – vizualizace naměřených hodnot

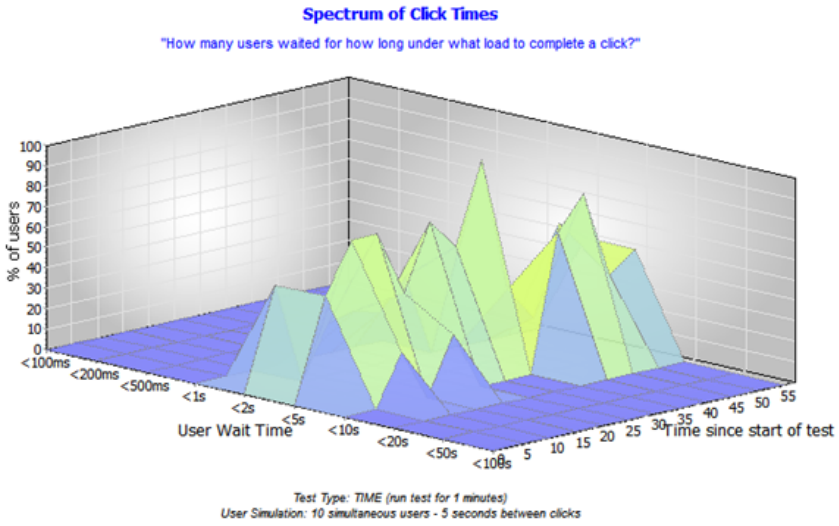
Max-session-tool – na obsluhu velmi jednoduchý CLI nástroj, používaný například v testech domácích zařízení. Program má serverovou a klientskou část, přičemž klient ve velmi krátkém časovém úseku odesílá UDP pakety ze širokého rozsahu portů na IP adresu protějšku, který je v topologii za testovaným síťovým prvkem. Vstupními parametry je na serverové straně IP adresa a první číslo portu, na kterém má server naslouchat a na klientovi zadáváme IP adresu, port serveru a první číslo portu, ze kterého klient začne odesílat. Princip měření hodnoty je takový, že při vypršení timeoutu n-tého spojení určíme maximální počet konkurentních spojení, které je zařízení stojící v cestě schopno obsloužit, jako hodnotu n-1. [Higgins(2010b)] Zdrojové a cílové porty se během testu inkrementálně o jedničku zvyšují. Během tohoto měření by neměla být připojena do topologie žádná další aktivní síťová zařízení a z hostitelských zařízení by neměla odcházet žádná další komunikace.

Webserver stress tool 7 - tento nástroj je, jak již samotný název napovídá, primárně určen pro simulaci reálného prostředí a testování/záznam chování webových serverů pod zátěží. Program nahrazuje aktivity uživatelského webového prohlížeče a zaznamenává údaje o délce vyřízení požadavku. Pomocí tohoto nástroje jsem simuloval aktivity typu prohlížení webu a za tímto účelem použil pouze takové weby, o kterých se domnívám, že mají natolik silnou hw strukturu, že mohu vyloučit pokles výkonnosti serveru na základě mého testování. Typicky www.seznam.cz, www.idnes.cz, www.novinky.cz, www.blesk.cz atd.

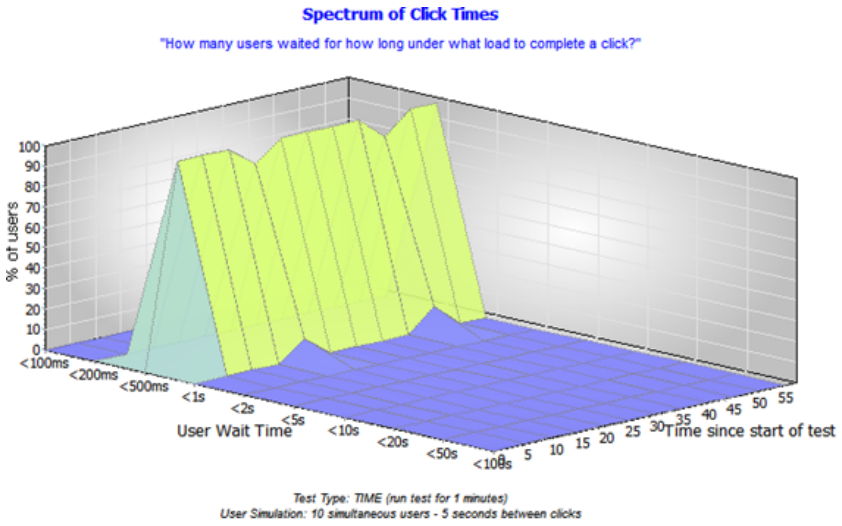
Při konfiguraci testu se vyplňuje seznam URL, jež mají být navštěvovány/proklikávány virtuálními uživateli, volí se testovací prohlížeč a nastavují parametry jako četnost kliků, délka testu. Plná verze programu je po-

skytována za úplatu, zatímco verze „Free trial“ nabízí totožné funkce, ale pouze omezené množství virtuálních uživatelů na 10 a frekvenci kliknutí 1 uživatele nejvíce 1 krát za 5 sekund. [Paessler(2013)] Tato omezení mě však nelimitují, protože v cílovém prostředí jen zřídkakdy bude více jak 10 uživatelů souběžně prohlížet webové stránky a četnost kliku v prohlížeči 1x za 5s na jednoho uživatele se mi jeví jako dostačující (720 zobrazených stránek za hodinu).

Výstupem z provedených měření je obsáhlý HTML report s celkovými statistikami, kolik dat bylo kterým uživatelem přeneseno a jaký objem síťového provozu byl po dobu testu spotřebován. Jistě nejzajímavějším údajem je však graf zobrazující dobu vyřízení požadavku(doba od kliku na odkaz po načtení stránky) v průběhu běhu testu. Jako ilustrační příklad přikládám obrázky 2.5 a 2.6



Obrázek 2.5: 10 uživatelů, 1 web: http://www.idnes.cz, každý uživatel 1 zobrazení za 5 sekund



Obrázek 2.6: 10 uživatelů, 1 web: http://m.idnes.cz – mobilní verze portálu idnes.cz, každý uživatel 1 zobrazení stránky za 5 sekund

IxChariot - profesionální, placený nástroj určený pro testování parametrů spojení mezi 2 síťovými zařízeními. Podle dokumentace a specifikace [Jonkman(2012)] se jedná o ideální nástroj testování s velkou mírou komplexity testů. Umožňuje volbu generování provozu na úrovni aplikačních protokolů. Bohužel jsem nemohl vyzkoušet ani testovací verzi tohoto sw, neboť i k tomu je nutné mít aktivní účet na portálu firmy, který je však podmíněn zakoupením alespoň jednoho kusu HW od této společnosti nebo jejích partnerů.

2.4 Vybraná testovaná zařízení

Směrovače:

- Asus WL500G
- Cisco 2801
- Edimax BR6204WG
- Linksys WRT150N
- Mikrotik RouterBoard RB532

Přepínače:

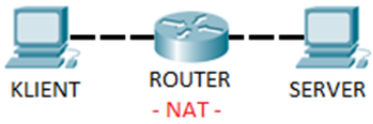
- Ovislink FSH8PS
- Cisco 3560

2.5 Návrh a provedení testovacích scénářů

Po úvodním průzkumu vybraných nástrojů jsem sestavil několik obecných testovacích scénářů pro nástroj iperf. Motivem byla snaha se s nástrojem blíže seznámit a získat z provedených měření data, která bych mohl později porovnat s výsledky jiných (podobně zaměřených) programů. Obrázky zapojení síťových zařízení jsem vypracovával v simulačním programu Cisco Packet Tracer 5.3.3, jehož použití je mi jako studentovi síťové akademie při ZČU k dispozici.

1)

Testované zařízení:	Směrovač
Režim zařízení:	NAT
Sledované parametry:	Rychlost TCP přenosu, server => klient
Schéma scénáře:	Obrázek 2.7



Obrázek 2.7:

2)

Testované zařízení:	Směrovač
Režim zařízení:	ROUTING
Sledované parametry:	Rychlost TCP přenosu, server => klient
Schéma scénáře:	Obrázek 2.8

Obrázek 2.8:

3)

Testované zařízení:	Směrovač
Režim zařízení:	BRIDGE
Sledované parametry:	Rychlost TCP přenosu, server => klient
Schéma scénáře:	Obrázek 2.9

Obrázek 2.9:

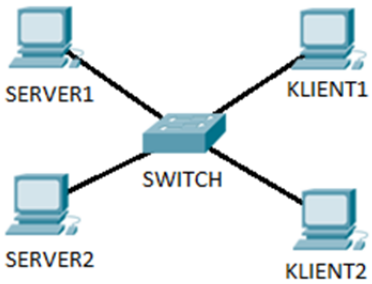
4)

Testované zařízení:	Směrovač
Režim zařízení:	ROUTING
Sledované parametry:	ztrátovost UDP paketů
Schéma scénáře:	Obrázek 2.10

Obrázek 2.10:

5)

Testované zařízení:	přepínač nebo LAN rozhraní směrovačů
Režim zařízení:	nedefinováno
Sledované parametry:	Rychlost TCP přenosu, server => klient
Schéma scénáře:	Obrázek 2.11



Obrázek 2.11:

Režim	TCP window size	Užitečná rychlost
NAT	56kB	69,6 Mb/s
	112kB	93,5 Mb/s
ROUTING	56kB	69,9 Mb/s
	112kB	93,8 Mb/s
BRIDGE	56kB	70,3 Mb/s
	112kB	94,2 Mb/s

Tabulka 2.1: Srovnání přenosových rychlostí jednotlivých režimů

QoS	UDP packet size	Ztrátovost
Vypnuto	1500Byte	0 %
	6000Byte	15,2 %
Zapnuto	1500Byte	0 %
	6000Byte	43,4 %

Tabulka 2.2: Vliv funkce QoS na ztrátovost UDP paketů

Komunikace	TCP window size	Užitečná rychlost
Klient1-Server1	56kB	72,6 Mb/s
	112kB	91,5 Mb/s
Klient2-Server2	56kB	71,4 Mb/s
	112kB	93,2 Mb/s

Tabulka 2.3: Přenosové rychlosti v závislosti na velikosti TCP window size

Vyhodnocení provedených testů

Testovací scénáře 1,2,3,4 jsem realizoval se směrovačem RouterBoard RB532. Pro scénář č.5 jsem použil síťový přepínač Ovislink FSH8PS. Získaná data jsou v tabulkách 2.1, 2.2 a 2.3.

Z naměřených hodnot v tabulce 2.1 soudím, že změna režimu zařízení nemá významný vliv na přenosovou rychlost testovacího vzorku dat. Jako mnohem zásadnější se jeví velikost parametru *TCP window size* neboli množství dat, které může jedna strana odeslat bez čekání na potvrzení druhé strany [Jacobson et al.(1992)Jacobson, Braden„ Borman]. Po zvětšení parametru z 56kB na 112kB můžeme pozorovat zvýšení rychlosti téměř až k praktickému maximu spojení typu FastEthernet.

V tabulce 2.2 lze pozorovat vliv QoS funkce na protékající datové přenosy. Dle výsledků je pro standardní velikosti UDP datagramů (tj. do 1500Byte) výkon směrovače dostatečný, neboť ani při rychlosti 30Mbps nezahodí jediný datagram. Při použití tzv. jumbo paketů směrovač již nestíhá směrovat všechny pakety a dochází ke ztrátám. Po zapnutí jednoduché QoS, která pakety pouze značkuje na základě zdrojové adresy dochází ke znatelnému zvýšení ztrátovosti.

Výsledky získané z testovacího scénaře č.5 potvrdily příčinu nižších přenosových rychlostí u předchozích scénářů. Snížení hodnoty parametru *TCP window size* se projevilo poklesem přenosové rychlosti. Naměřené hodnoty jsou umístěny do tabulky 2.3.

2.6 Úprava zaměření testovacích scénářů

Zhodnocení získaných poznatků

Provedené testy jsem využil k seznámení se s metodikou testování a konfigurací MikroTik zařízení. Při navrhování dalších testovacích scénářů se budu snažit brát v úvahu, že primární provozní oblastí síťového zařízení je prostředí v domácnostech nebo menších firmách. Takové zařízení by mělo být schopno:

- Přenášet rychle a spolehlivě data mezi zařízeními v rámci LAN
- Zachovat kvalitu hlasového hovoru (VOIP) i během vyššího zatížení
- Zahazovat minimální množství UDP provozu (např. online hry, stream pro TV)
- Pracovat v režimu NAT bez znatelné ztráty výkonu

Návrh skupiny testovacích protokolů

Provoz generovaný testovací aplikací by měl být svým složením podobný reálnému provozu, abychom byli schopni odhalit případné chyby nebo zhoršenou kvalitu síťových služeb, které mají pro uživatele skutečný význam. Pro další testování jsem proto vybral několik nejpoužívanějších protokolů. Každý z těchto protokolů je charakteristický svým chováním, velikostí komunikačních jednotek a frekvencí komunikace, náročností na specifické parametry přenosové soustavy a interaktivitou s lidskou činností.

Při výběru testovaných protokolů jsem vycházel ze studie [Schulze – Mochalski(2009)Schulze, Mochalski] německé společnosti Ipoque GmbH, která se specializuje na monitorování a optimalizace datových toků dnešního internetu. Zmíněná studie analyzuje zastoupení použití známých internetových protokolů v 8 vybraných regionech světa za rok 2008 a 2009. Celkový vzorek analyzovaných přenesených dat čítá velikost 1,3 PetaByte. Zaujala mne zde tabulka znázorňující podíly jednotlivých tříd internetových protokolů na celkovém přeneseném objemu. Z ní vyplývá, že nejvíce dat je přeneseno P2P[9]

[9]Peer to Peer

protokoly. Jak se ale v práci můžeme dále dočíst, pouze 15-20% uživatelů připojených k internetu využívá k datovým přenosům skupiny těchto protokolů.

Protocol Class	Southern Africa	South America	Eastern Europe	Northern Africa	Germany	Southern Europe	Middle East	South-western Europe
P2P	65,77%	65,21%	69,95%	42,51%	52,79%	55,12%	44,77%	54,46%
Web	20,93%	18,17%	16,23%	32,65%	25,78%	25,11%	34,49%	23,29%
Streaming	5,83%	7,81%	7,34%	8,72%	7,17%	9,55%	4,64%	10,14%
VoIP	1,21%	0,84%	0,03%	1,12%	0,86%	0,67%	0,79%	1,67%
IM	0,04%	0,06%	0,00%	0,02%	0,16%	0,03%	0,50%	0,08%
Tunnel	0,16%	0,10%	-	-	-	0,09%	2,74%	-
Standard	1,31%	0,49%	-	0,89%	4,89%	0,52%	1,83%	1,23%
Gaming	-	0,04%	-	-	0,52%	0,05%	0,15%	-
Unknown	4,76%	7,29%	6,45%	14,09%	7,84%	8,86%	10,09%	9,13%

Obrázek 2.12: Procentuální podíl komunikačních protokolů na celkovém objemu přenesených dat

Druhou nejpoužívanější sadou protokolů je webová sada a tedy hlavně HTTP. Varianta zabezpečené komunikace HTTPS tvořila dle studie podíl jednotek procent. Následuje přenos audio a video médií a hlasových služeb. Tabulku uzavírají takzvané standardní protokoly - FTP, SMTP, DNS, POP, IMAP.

Pod touto prací jsou podepsáni pánové Hendrik Schulze a Klaus Mochalski. [9]

Vybrané protokoly, jejichž kvalita komunikace bude předmětem měření vyvíjené aplikace, jsou:

- HTTP – prohlížení webu
- FTP – přenos souborů
- TELNET – interaktivní komunikace

- UDP stream – hry, rádio, televize

Zvolené parametry jsou pak v jednotlivých testech:

- HTTP – doba potřebná k načtení stránky
- FTP – rychlost přenosu (oběma směry)
- TELNET – prodlevy v odezvě, výpadky spojení
- UDP stream – ztrátovost paketů

Testovací nástroj by měl ideálně obsahovat grafické rozhraní.

Během průzkumu a práce s testovacími nástroji jsem neobjevil takový program, který by byl zdarma dostupný a vyhovoval výše navrhnutým požadavkům. O jejich naplnění se proto budu snažit v rámci vyvíjené aplikace v realizační části bakalářské práce.

3 Realizační část

3.1 Návrh testovací aplikace

Požadavky na aplikaci

Požadavky na funkcionality měřící aplikace vychází ze závěrů předchozích provedených měření. Hlavním motivem tvorby vlastní aplikace je vytvoření takového nástroje, který umožňuje přenášet data s využitím skutečných a v praxi používaných protokolů aplikační vrstvy ISO/OSI modelu. Cílem není sestrojit generátor náhodných dat ani aplikaci na reprodukci odchyceného provozu.

Síťová architektura a topologie zapojení

Schéma síťové komunikace při běhu více instancí aplikace v rámci sítě bude realizováno ve formátu klient-server a to v poměru 1:N. Jeden server bude schopen najednou vyřizovat požadavky od více klientů. Klienti budou mít na starost konfiguraci, spouštění a zaznamenávání výsledků testů – měření. Před spuštěním testu musí být navázáno řídící spojení mezi klientem a serverem. To poskytne klientovi možnost požádat server o spuštění služeb (daemonů) před samotným zahájením testu. Server má naopak možnost klientovi oznámit případný chybový stav požadované služby, na který musí být klient schopen reagovat např. nezahájením testu určitého protokolu nebo kompletně celé testové sady. Klient bude spouštět v jednom okamžiku najednou vždy celou sadu testů, přičemž každý test (protokol) poběží ve vlastním vlákně. V případě, kdy klient provádí více iterací skupiny testů, čeká se před zahájením následující iterace vždy na dokončení všech testů aktuální iterace.

Podporované protokoly

Aplikace bude využívat nejpoužívanější protokoly aplikační vrstvy, aby generovaný síťový tok odpovídal skutečným datovým přenosům. U každého protokolu budou zvoleny sledované parametry, které vypovídají o kvalitě služby. U každého protokolu bude vhodně zvolena odpovídající metrika měření. Měřící nástroj bude schopen generovat a obsluhovat požadavky postavené na protokolu HTTP, FTP, TELNET a zachytávat proud UDP paketů.

Programovací jazyk

Za účelem multiplatformního nasazení aplikace jsem se rozhodl o jejím vývoji v jazyce JAVA bez využití utilit třetích stran ve formě různých binárních souborů. Veškerou funkcionalitu aplikace budou pokrývat JAVA balíky a knihovny.

Grafické rozhraní

Pro uživatelsky přívětivější ovládání by měla aplikace disponovat grafickým rozhraním, kde bude možné nastavit všechny parametry a sledovat průběžné výsledky měření.

Formát výstupu a uložení naměřených dat

Naměřené hodnoty a výsledky testů musí být možno uložit pro případnou analýzu, porovnání či další zpracování. Při práci s jinými měřícími nástroji jsem ocenil možnost sledovat aktuálně naměřené hodnoty. Ideálním stavem je umístění grafu generovaného v reálném čase (nebo periodicky se obnovujícím) do okna aplikace.

3.2 Realizace testovací aplikace

Vývojové prostředí

Vývojovým prostředím pro JAVA aplikaci jsem si zvolil NetBeans. Framework použitý pro vývoj grafického rozhraní nástroje je SWING.

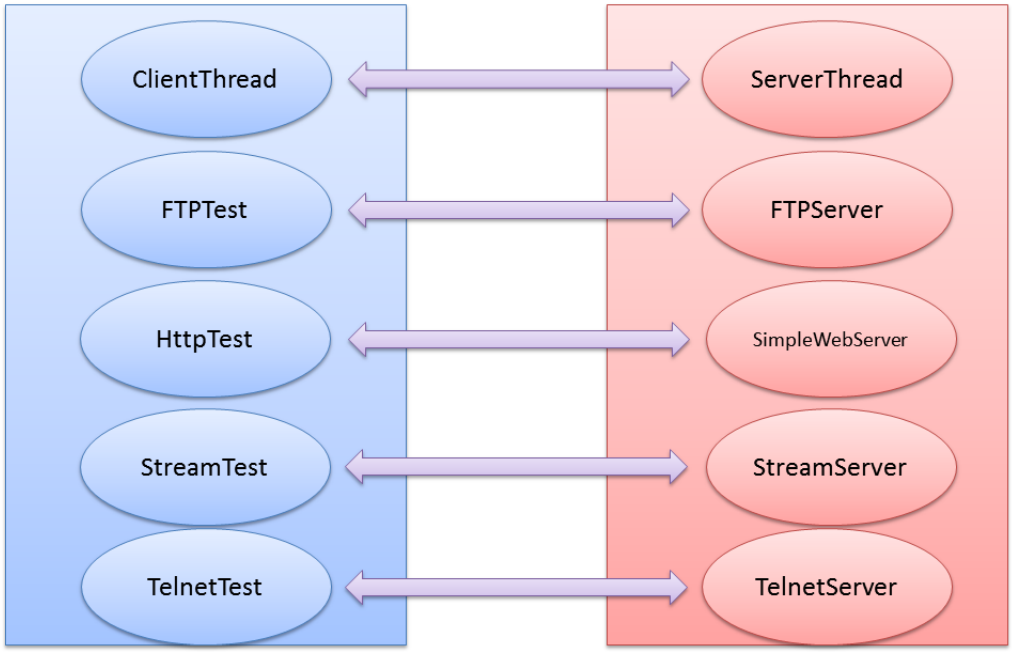
Architektura a vzhled aplikace

Na počátku vývoje testovací aplikace jsem se musel rozhodnout, zda budu vytvářet 2 aplikace, kdy jedna bude plnit funkci serveru a druhá klienta nebo pouze jednu aplikaci, kdy bude uživatel moci zvolit roli vhodným přepínačem. Pro jednodušší distribuci, obsluhu, mobilitu a na základě osobních pozitivních zkušeností s druhou zmíněnou možností jsem vybral variantu s volbou klient/server až po spuštění aplikace. Každá spustitelná verze aplikace tak může v testovacích scénářích plnit obě dvě role (ne v jeden okamžik).

Při rozdělení na klientskou a serverovou část bylo třeba zvolit, která část bude mít řídící funkci, pomocí níž bude uživatel konfigurovat parametry testu

a sbírat výsledky. Zároveň mi nepřišlo vhodné, aby uživatel musel provádět konfiguraci plánovaného testu na obou stranách síťové topologie.

Pro lepší simulaci reálného prostředí a věrnějších charakteristik datových toků jsem musel zajistit, aby všechny druhy testů běžely současně. Každý test ve svém vlákně. Za účelem udržení homogenity testu během jednotlivých iterací jsem pak musel implementovat vláknovou synchronizaci. Zvoleným synchronizačním prvkem je bariéra, která slouží k tomu, že každé vlákno po skončení iterace čeká, až budou hotovy všechny testy. Po dosažení bariéry všemi vlákny jsou testy puštěny do dalšího cyklu.



Obrázek 3.1: Znázornění komunikace klientských vláken (modře) se serverovou částí aplikace (červeně)

Konfiguračnímu rozhraní jsem chtěl dát uživatelsky příjemnou podobu, a proto je aplikace plně ovladatelná z grafického prostředí bez nutnosti dohledávat v dokumentaci potřebné parametry či přepínače. Možnou nevýhodou ovládání skrze grafické rozhraní však může být v některých konfiguračních položkách nedostatečný rozsah hodnot pro náročnější testovací scénáře.

Popis metodik jednotlivých testů

Pro popis funkcionality a možností aplikace označuji jako klientskou stranu

aplikace tu instanci spuštěného programu, ve které bylo před spuštěním zvolen režim „klient“. Tato aplikace poskytuje konfigurační a výstupní rozhraní prováděných testů a sestavuje konfigurační zprávu, která obsahuje požadavky na spuštění konkrétních služeb s nakonfigurovanými parametry a je odesílána serverové straně. Analogicky platí totéž pro serverovou stranu aplikace a volbu „server“. Toto označení má za účel zjednodušit orientaci v popisu testovacích scénářů.

FTP test

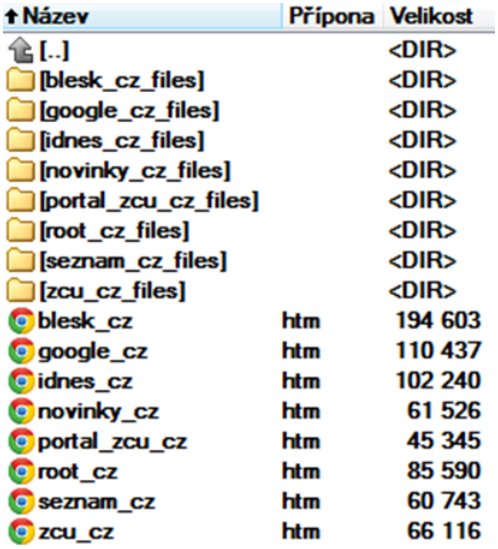
Tento typ testu využívá protokol FTP, s jehož pomocí můžeme zjistit skutečnou přenosovou kapacitu síťového spojení. Serverová strana aplikace spouští za pomoci externí knihovny FTP server jako službu na klientovi předem známém portu. Pokud se serverové straně aplikace nepodaří službu rozběhnout nebo se například nezdaří otevřít předem domluvený TCP port, je o této skutečnosti klientovi odeslána zpráva a uživatel je o této skutečnosti upozorněn výpisem v konzoli v serverové i klientské straně aplikace. V případě, kdy se serveru podaří službu úspěšně spustit, je následně nakonfigurován testovací účet služby, jehož přihlašovací údaje jsou všem klientům předem známy. Klientská část aplikace se za pomoci externí knihovny připojí na spuštěný FTP server a do kořenového adresáře zahájí přenos vzorového souboru. Vzorový soubor (sample file) je soubor vybraný v konfigurační záložce testů v konfigurační možnosti *Sample file path*. Po úspěšném odeslání souboru spočítá aplikace průměrnou přenosovou rychlost dat v kbps a tu uloží mezi získané výsledky do databázového souboru a bezprostředně zahajuje stahovací fázi testu. Během této fáze klient stahuje původně odeslaný vzorový soubor. Doba, jenž uplyne během datového přenosu je opět použita pro výpočet průměrné rychlosti a výsledek je uložen do tabulky s výsledky. V případě, kdy FTP server současně obsluhuje několik klientských testovacích aplikací, je žádoucí, aby každý klient použil jiný název přenášeného souboru a to z toho důvodu, aby klienti během testu nemohli navzájem přepisovat tentýž soubor. Stažené a nahrané soubory nejsou během měření ani po jeho skončení odstraněny ze souborového systému. Lze tak později například ověřit, že se soubor podařilo přenést bez chyby. Aplikace komunikuje protokolem FTP v pasivním režimu.

HTTP test

Tento test zahrnuje simulaci skutečného datového přenosu pomocí http protokolu a současně měří uplynulý čas během přenosu všech požadovaných datových vzorků. Webová služba je spuštěna na základě požadavku v přijaté

konfigurační zprávě na domluveném portu a ukončuje se při zavření aplikace. V případě testovacího scénáře s více klienty je spuštěna pouze jedna instance webového serveru, která obsluhuje požadavky všech http klientů. Měření, zpracování a ukládání výsledků obstarává klientská strana.

Pro tento typ testu je nutné, aby měla serverová část aplikace přístup k připraveným, s aplikací dodávaným, datovým vzorkům - Obrázek 3.2. Jedná se o kopie několika titulních stránek vybraných webů včetně obrázků, skriptů a css souborů potřebných pro offline zobrazení kompletní stránky ve webovém prohlížeči. Tato data jsou poskytována webovým serverem na serverové straně aplikace během prováděného http testu.



Obrázek 3.2: Ukázka souborové struktury webové služby aplikace

Uživatel má při konfiguraci http testu možnost upravovat jeho rozsah výběrem příslušné sady vzorků z konfigurační nabídky na klientské straně aplikace.

UDP test

Reprezentantem UDP provozu je proud datagramů. Ty jsou generovány na serverové straně ve zvolené velikosti a s uživatelem zadanou frekvencí. Stejně tak lze nastavit délku trvání UDP proudu. Klientská strana naslouchá na domluveném portu, zpracovává příchozí datagramy, počítá a následně ukládá absolutní počet a po přepočtu i procento paketů, jež nedorazilo na klientův síťový interface v ohraničeném čase. Časový limit pro tento test se počítá z hodnoty zadané v konfiguračním rozhraní a připočte se +500ms.

To je doba, kdy server již nevysílá, ale klient ještě přijímá a zpracovává příchozí pakety zpožděné přenosem po síti. UDP stream má představovat multimediální audio/video služby, u nichž je kladen důraz na rychlé doručení s pokud možno co nejmenší časovou prodlevou

TELNET test

Pro potřeby měření latence linky a simulaci interaktivity zadávání příkazů na vzdálený terminál jsem navrhl test, ve kterém klientská stanice inicializuje na začátku každé iterace TCP spojení na serverem vystavený port. Pro každého připojeného klienta je na serveru vytvořeno nové vlákno, aby nedocházelo k časovým prodlevám při vyřizování odpovědí na přijaté požadavky. Požadavky generuje s uživatelem zadanou frekvencí klientská strana a server po jejich přijetí okamžitě odpovídá zprávou s jasně daným formátem. Charakteristika tohoto síťového spojení se podobá aplikačnímu telnet protokolu, odkud pochází název testu.

Při tomto testu se měří časy, které uplynou mezi odesláním požadavku a přijetím příslušné odpovědi. Hodnoty jsou pak ukládány do 2 tabulek. Do jedné se ukládají latence všech požadavků. Do druhé pak již jen průměrná hodnota zpoždění, vypočtená ze všech zpráv iterace.

Na obrázku 3.3 je zobrazena část zachycené komunikace mezi klientem a serverem. Hodnoty ve sloupci *Time* jsou časy mezi zachycením jednotlivých paketů na síťovém rozhraní stroje s aktivní klientskou částí aplikace.

Time	Source	Destination	Protocol	Length	Info
0.106286	10.0.0.11	192.168.0.10	TCP	66	58619 > 2023 [PSH, ACK] Seq=229 Ack=248 Win=259 Len=12
0.002498	192.168.0.10	10.0.0.11	TCP	67	2023 > 58619 [PSH, ACK] Seq=248 Ack=241 Win=257 Len=13
0.106518	10.0.0.11	192.168.0.10	TCP	66	58619 > 2023 [PSH, ACK] Seq=241 Ack=261 Win=259 Len=12
0.002958	192.168.0.10	10.0.0.11	TCP	67	2023 > 58619 [PSH, ACK] Seq=261 Ack=253 Win=257 Len=13
0.106313	10.0.0.11	192.168.0.10	TCP	66	58619 > 2023 [PSH, ACK] Seq=253 Ack=274 Win=258 Len=12
0.002410	192.168.0.10	10.0.0.11	TCP	67	2023 > 58619 [PSH, ACK] Seq=274 Ack=265 Win=257 Len=13
0.107260	10.0.0.11	192.168.0.10	TCP	66	58619 > 2023 [PSH, ACK] Seq=265 Ack=287 Win=258 Len=12
0.002355	192.168.0.10	10.0.0.11	TCP	67	2023 > 58619 [PSH, ACK] Seq=287 Ack=277 Win=257 Len=13
0.106913	10.0.0.11	192.168.0.10	TCP	66	58619 > 2023 [PSH, ACK] Seq=277 Ack=300 Win=258 Len=12
0.003104	192.168.0.10	10.0.0.11	TCP	67	2023 > 58619 [PSH, ACK] Seq=300 Ack=289 Win=257 Len=13

Obrázek 3.3: Ukázka zachycené komunikace mezi klientskou a serverovou částí aplikace v průběhu TELNET testu

Před začátkem měření odesílá klient serveru konfigurační zprávu. Ten tím tak dostane informaci o tom, které testy jsou požadovány a jaké služby je třeba na serveru pro klienta případně potřeba nakonfigurovat a spustit.

Ukládání a zpracování dat

Aplikace průběžně během testu zapisuje získané hodnoty z prováděných

testů do tabulek databázového souboru formátu sqlite s názvem „test.db“ za pomoci JDBC[1] spojení. Tento soubor se primárně vytváří při prvním spuštění testu s libovolnou konfigurací. Pokud již v adresáři soubor s takovým názvem existuje, tak se tento použije s tím, že před samotným vkládáním nových/naměřených hodnot jsou nad příslušnou tabulkou volány DDL[2] příkazy DROP & CREATE.

Reprodukce měření s identickou konfigurací

Při vývoji aplikace byla snaha parametrizovat délku a rozsah testu na základě vstupních dat zvolených uživatelem. Proto lze s velmi velkou mírou přesnosti vytvořit s použitím stejného HW vybavení nezávisle konfigurované testy tak, že je můžeme později za předpokladu použití stejných testovacích zařízení (tj. těch, které se zúčastní testu v roli klient nebo server, ale nejsou předmětem testování) a shodných vstupních parametrech, považovat za totožné. Dosažené výsledky těchto testů poté má smysl mezi sebou vzájemně porovnávat.

3.3 Návrh testovacích scénářů pro aplikaci

Z důvodu zachování přehlednosti v navrhnutých testovacích scénářích a výsledcích budeme používat pro označení testovaných síťových zařízení následující zkratky:

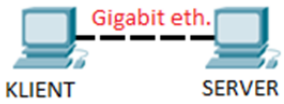
- Asus = ASUS WL500G
- Cisco = Cisco 2801
- Cisco 3560 = Cisco 3560
- Edimax = EdimaxBR6204WG
- Linksys = Linksys WRT150N
- Routerboard = RouterBoard RB532

[1] Java Database Connectivity
[2] Data Definition Language

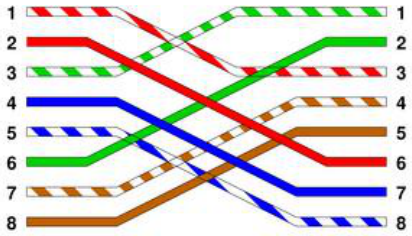
Následující testovací scénáře byly realizovány v síťové laboratoři UI505 na Západočeské univerzitě. Testovací aplikace byla spuštěna na tamních laboratorních desktopových počítačích.

Scénář 1)

Testovaná zařízení:	2x GigabitEthernet NIC (Klient & Server)
Režim zařízení:	Spojení stanic kříženým UTP kabelem
Sledované parametry:	Rychlost stahování a odesílání přes FTP protokol
Schéma scénáře:	Obrázek 3.4
Poznámka:	Zakončení síťového kabelu pro GigabitEthernet demonstruje obrázek 3.5

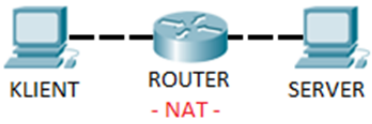


Obrázek 3.4:



Obrázek 3.5: Schéma zapojení kříženého kabelu pro GigabitEthernet zdroj: http://www.svetsiti.cz

Scénář 2)



Obrázek 3.6:

Testovaná zařízení:	Asus, Cisco, Edimax, Linksys, RouterBoard
Režim zařízení:	NAT
Sledované parametry:	Rychlost stahování a odesílání přes FTP protokol.
Schéma scénáře:	Obrázek 3.6
Poznámka:	

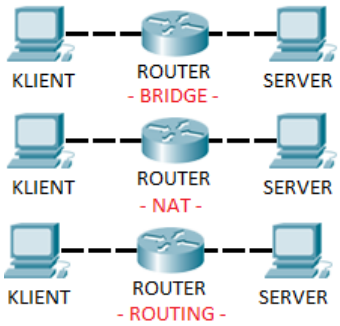
Scénář 3)

Testovaná zařízení:	Asus, Cisco, Edimax, Linksys, RouterBoard
Režim zařízení:	NAT
Sledované parametry:	Čas potřebný k přenosu všech prvků www stránek
Schéma scénáře:	Obrázek 3.7
Poznámka:	

Obrázek 3.7:

Scénář 4)

Testovaná zařízení:	Asus, Cisco, Edimax, Linksys, RouterBoard
Režimy zařízení:	NAT, BRIDGE,ROUTING
Sledované parametry:	Vliv režimu na datový tok protokolu FTP
Schéma scénáře:	Obrázek 3.8
Poznámka:	



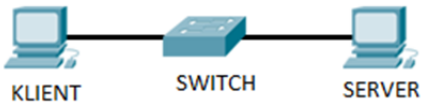
Obrázek 3.8:

Scénář 5)

Testovaná zařízení:	Asus, Cisco, Edimax, RouterBoard
Režim zařízení:	NAT
Sledované parametry:	FTP, HTTP, TELNET, STREAM
Schéma scénáře:	Obrázek 3.9
Poznámka:	Budeme pozorovat, jak budou jednotlivá zařízení reagovat na velikou zátěž tvořenou směsí datových toků s různou charakteristikou

Obrázek 3.9:

Scénář 6)



Obrázek 3.10:

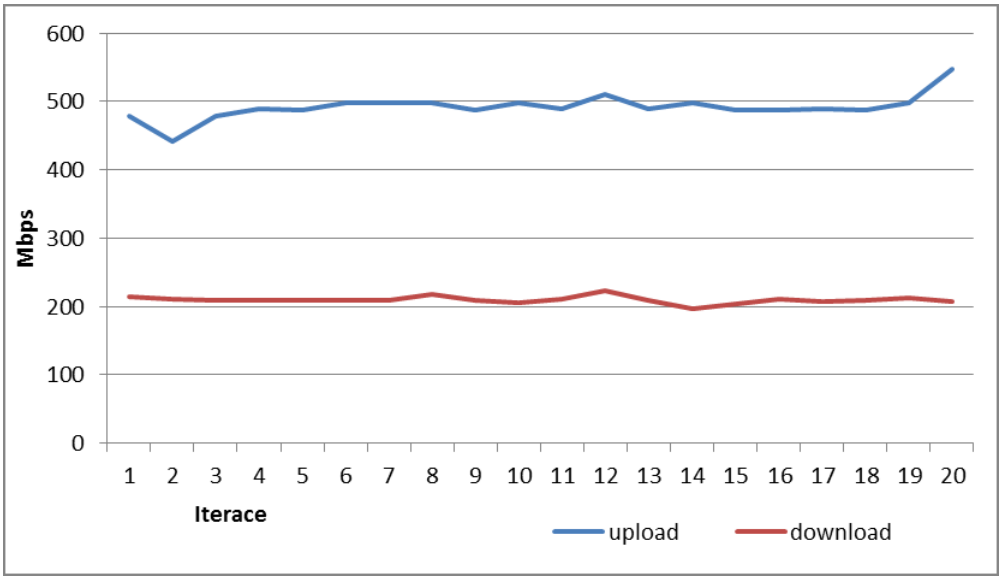
Testovaná zařízení:	Asus, Edimax, Cisco 3560
Režim zařízení:	defaultní nastavení
Sledované parametry:	přenosové rychlosti protokolu FTP
Schéma scénáře:	Obrázek 3.10
Poznámka:	

3.4 Realizace testů na vybraných zařízeních

Nyní se podrobněji podíváme na naměřené hodnoty a provedeme jejich srovnání.

Scénář 1)

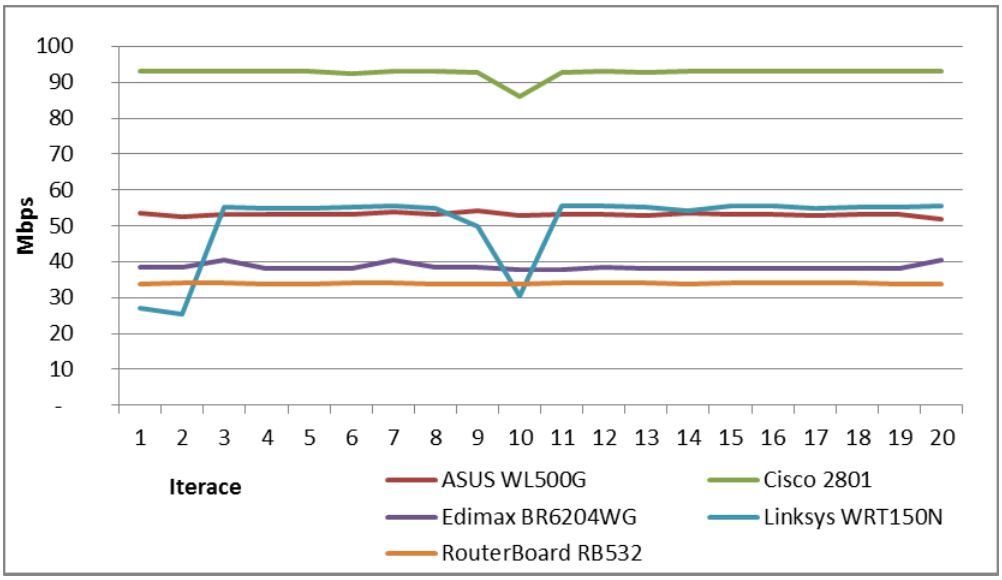
Toto měření slouží k ověření, že klientská a serverová stanice jsou pro testování výkonově dostatečné. Spojení stanic je realizováno UTP kříženým kabelem - viz obr. 3.5. Komunikace protokolu FTP tak neprochází žádným síťovým prvkem.



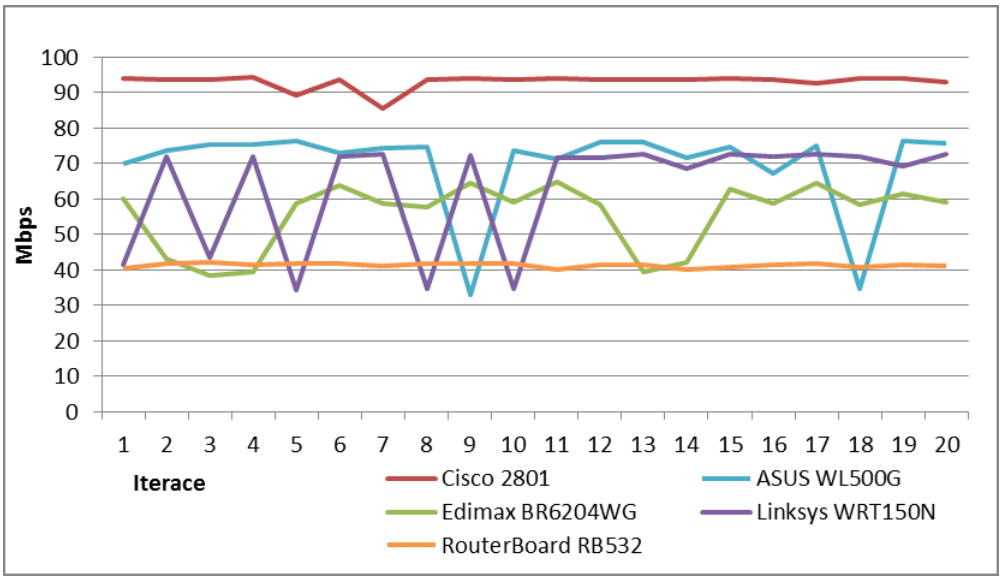
Obrázek 3.11: Graf FTP testu - rychlost stahování a odesílání dat

Scénář 2)

V tomto testu probíhala komunikace pomocí FTP protokolu skrze zařízení, jež pracovalo v režimu NAT. Použití toho režimu je typické na hraničních prvcích v domácnostech a to jednak z důvodu bezpečnostních (jedná se o přirozený firewall bránící cizím zařízením proniknout na síťové prvky v lokální síti), ale především z důvodu nedostatku volných veřejných IPv4 adres [Huston(2013)]. V průběhu měření vykazovala všechna zařízení stabilní rychlosti ve směru od serveru ke klientovi. V opačném směru byly výsledky rozkolísanější. Naměřené hodnoty udávají skutečnou průměrnou přenosovou rychlost FTP protokolu. Skutečný datový tok, který bychom naměřili na spojovacím médiu tak bude po připočtení hlaviček všech hlaviček použitých protokolů nižších vrstev samozřejmě vyšší. V testu dominoval Cisco směrovač s rychlostmi přesahujícími 90Mbps v obou směrech. Nejhůře si vedl RouterBoard s 35-45 Mbps. Ostatní zařízení se rychlostně pohybovala nad zmíněným RouterBoardem, v rozmezí 45-70Mbps. Cisco směrovači se v tomto testu žádné zařízení výkonově nepřiblížilo. Naměřené hodnoty jsou zaneseny do grafů na obrázcích 3.12 a 3.13.



Obrázek 3.12: Graf FTP testu - rychlost stahování dat



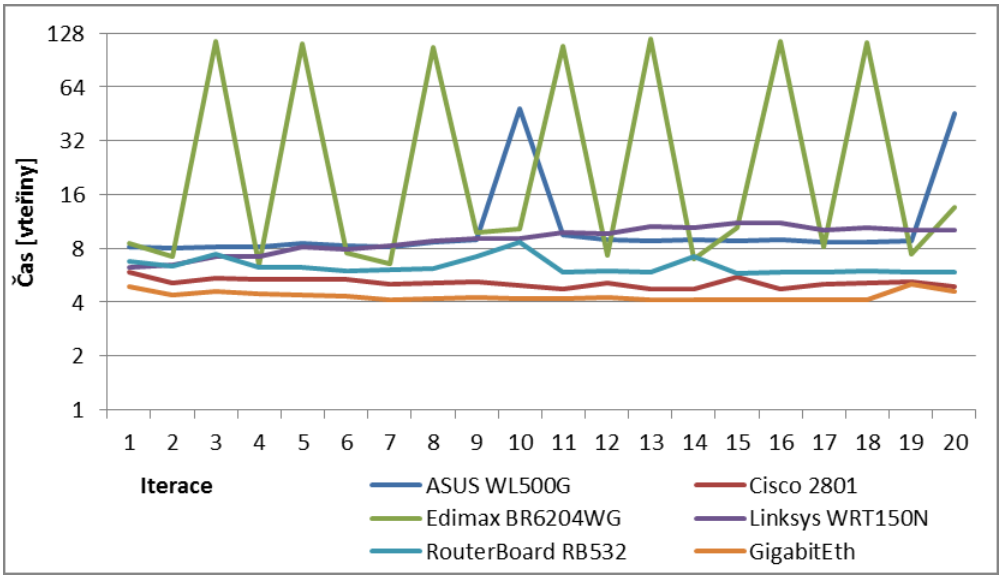
Obrázek 3.13: Graf FTP testu - rychlost odesílání dat

Scénář 3)

Po FTP testu jsem provedl s totožnými směrovači test, ve kterém klient se serverem komunikoval protokolem http a sítí se přenášely soubory tvořící dohromady repliku skutečných webových stránek, jak byly zveřejněny viz obr.3.2. Všechna zařízení opět pracovala v režimu NAT. Během testu se měřila doba, za jakou klient dostane od serveru všechny prvky webových stránek. Pro test byla zvolena kompletní sada připravených stránek. Celková velikost souborů všech 8 stránek čítá přibližně 8MB. I přes relativně malý objem přenášených dat ukázaly výsledky měření velké rozdíly v celkových časech. Nejlepších hodnot dosáhl Cisco směrovač. U něj se čas iterace pohyboval mezi 4-5 sekundami. U směrovače značky Edimax se při 3. iteraci objevil problém s navazováním TCP spojení. Klient se dostal do stavu, kdy odesílal TCP SYN pakety, na které nepřicházela odpověď. Tento stav trval desítky sekund, než přišel první SYN ACK paket od serveru a pokračovalo se v testu. Průběh tohoto testu s Edimax zařízením jsem sledoval v konzoli klientské aplikace a ve chvíli, kdy jsem zpozoroval delší nečinnost, jsem za pomoci systémového příkazu netstat zjistil, že počet navázaných TCP spojení stanice je 164. To vůbec není vysoké číslo, a proto mě toto zjištění překvapilo. Identické chování směrovače se opakovalo každou 3. iteraci a v tomto testu tak mezi ostatními směrovači naprosto propadlo.

Podobné problémové chování při navazování TCP relací jako měl smě-

rovač Edimax vykazoval i Asus, ale s frekvencí každou 10. iteraci a doba vzpamatování u něj byla cca o polovinu kratší viz obr.3.14. Síťové prvky RouterBoard a Linksys stabilně dosahovaly času iterace 6-7 respektive 7-10 sekund. Pro srovnání jsem do grafu přidal i test provedený na gigabitovém spojení 2 stanic laboratoře.

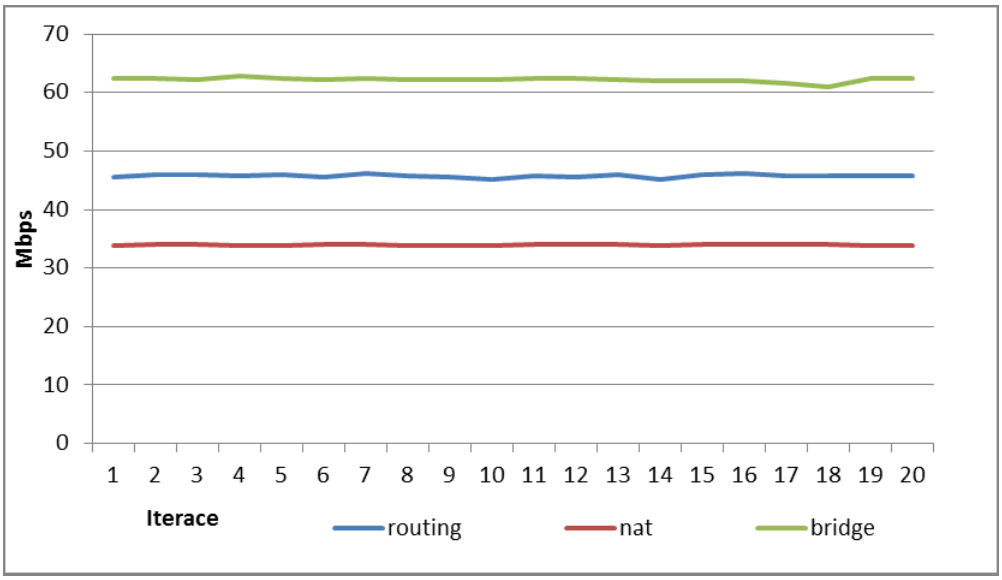


Obrázek 3.14: Graf HTTP testu - doba přenosu souborů

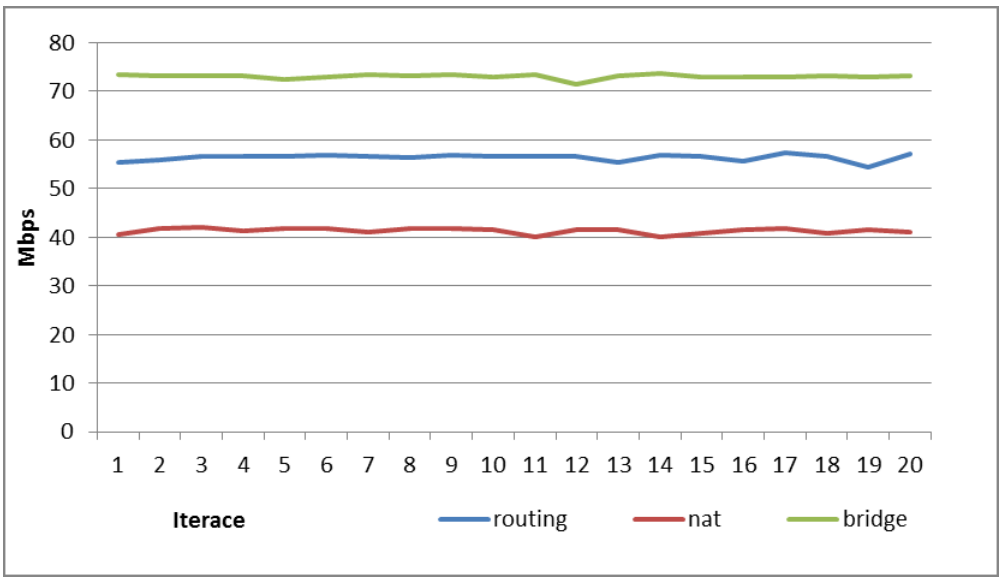
Time	Source	Destination	Protocol	Length	Info
138.336797	10.0.0.11	192.168.0.10	TCP	54	49561 > http [ACK] Seq=1 Ack=1 Win=66560 Len=0
138.336951	10.0.0.11	192.168.0.10	HTTP	243	GET /idnes_cz_files/B0E4a5d76_benesova2.jpg HTTP/1.1
138.339193	192.168.0.10	10.0.0.11	TCP	1314	[TCP segment of a reassembled PDU]
138.339395	192.168.0.10	10.0.0.11	TCP	1314	[TCP segment of a reassembled PDU]
138.339407	10.0.0.11	192.168.0.10	TCP	54	49561 > http [ACK] Seq=190 Ack=2521 Win=66560 Len=0
138.340436	192.168.0.10	10.0.0.11	TCP	1314	[TCP segment of a reassembled PDU]
138.340490	192.168.0.10	10.0.0.11	HTTP	719	HTTP/1.0 200 OK (JPEG JFIF image)
138.340521	10.0.0.11	192.168.0.10	TCP	54	49561 > http [ACK] Seq=190 Ack=4447 Win=66560 Len=0
138.342231	10.0.0.11	192.168.0.10	TCP	54	49561 > http [FIN, ACK] Seq=190 Ack=4447 Win=66560 Len=0
138.342921	192.168.0.10	10.0.0.11	TCP	60	http > 49561 [ACK] Seq=4447 Ack=191 Win=66560 Len=0
138.346014	10.0.0.11	192.168.0.10	TCP	66	49562 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1
141.355568	10.0.0.11	192.168.0.10	TCP	66	49562 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1
147.361560	10.0.0.11	192.168.0.10	TCP	62	49562 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1260 SACK_PERM=1
159.375484	10.0.0.11	192.168.0.10	TCP	66	49563 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1
162.384208	10.0.0.11	192.168.0.10	TCP	66	49563 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1
162.388277	10.0.0.11	192.168.0.10	TCP	62	49564 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1260 SACK_PERM=1
165.394980	10.0.0.11	192.168.0.10	TCP	62	49564 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1260 SACK_PERM=1
165.396297	10.0.0.11	192.168.0.10	TCP	66	49565 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1
165.398009	192.168.0.10	10.0.0.11	TCP	66	http > 49565 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1260 WS=256 SACK_PERM=1
165.398113	10.0.0.11	192.168.0.10	TCP	54	49565 > http [ACK] Seq=1 Ack=1 Win=66560 Len=0
165.398287	10.0.0.11	192.168.0.10	HTTP	238	GET /zcu_cz.htm HTTP/1.1
165.401793	192.168.0.10	10.0.0.11	TCP	1314	[TCP segment of a reassembled PDU]
165.401974	192.168.0.10	10.0.0.11	TCP	1314	[TCP segment of a reassembled PDU]
165.402001	10.0.0.11	192.168.0.10	TCP	54	49565 > http [ACK] Seq=185 Ack=2521 Win=66560 Len=0
165.403127	192.168.0.10	10.0.0.11	TCP	1314	[TCP segment of a reassembled PDU]
165.403496	192.168.0.10	10.0.0.11	TCP	1314	[TCP segment of a reassembled PDU]
165.403535	10.0.0.11	192.168.0.10	TCP	54	49565 > http [ACK] Seq=185 Ack=5041 Win=66560 Len=0
165.403612	192.168.0.10	10.0.0.11	TCP	1314	[TCP segment of a reassembled PDU]

Obrázek 3.15: Ukázka zachycené komunikace - chybějící TCP SYN pakety

Scénář 4)



Obrázek 3.16: Srovnání přenosových rychlostí směrem ke klientovi v závislosti na operačním režimu zařízení MikroTik RouterBoard RB532



Obrázek 3.17: Srovnání přenosových rychlostí směrem od klienta v závislosti na operačním režimu zařízení MikroTik RouterBoard RB532

Tímto testovacím scénářem jsem zjišťoval a porovnával výkonovou náročnost jednotlivých operačních režimů mezi 2 síťovými porty typu FastEthernet na zařízení **RouterBoard RB532**. Podle očekávání jsem dosáhl nejlepších výsledků v režimu bridge. Síťový provoz v ten moment není zatížen směrovacím rozhodováním a dalšími operacemi typickými pro 3. a vyšší vrstvu ISO/OSI modelu [MikroTik(2013)]. Největší úbytek přenosové rychlosti byl pozorovatelný při směrování s překladem adres – NAT.

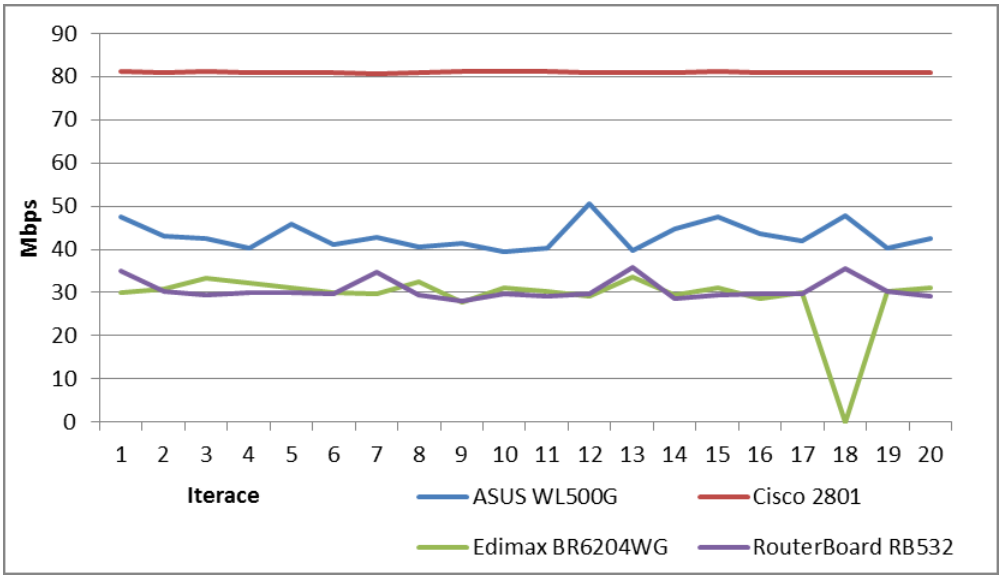
Scénář 5)

Testovací scénář, ve kterém jsou během iterace v jeden okamžik paralelně spuštěny všechny testy, má vystavit testované zařízení maximálnímu zatížení. Můžeme pak sledovat charakteristiky datových toků použitých síťových protokolů. Výsledkem provedených testů je následující rozbor: **Cisco 2801** – rychlost přenosu dat FTP protokolem byla snížena ve směru od serveru ke klientovi kvůli současnému sdílení kapacity spoje s proudem UDP datagramů, jehož objem činil cca 4Mbps. I přesto se přenosové rychlosti obou těchto testů v součtu přibližovaly maxima FastEthernet spojení. Na testu http protokolu se zátěž projevila prodloužením iterace v průměru o 71,3% viz tab. A.6 oproti samostatně běžícímu http testu. Stále se však jedná o výrazně nejlepší hodnoty mezi ostatními zařízeními. Po celou dobu měření se latence simulovaného telnet spojení pohybovala na úrovni jednotek ms.

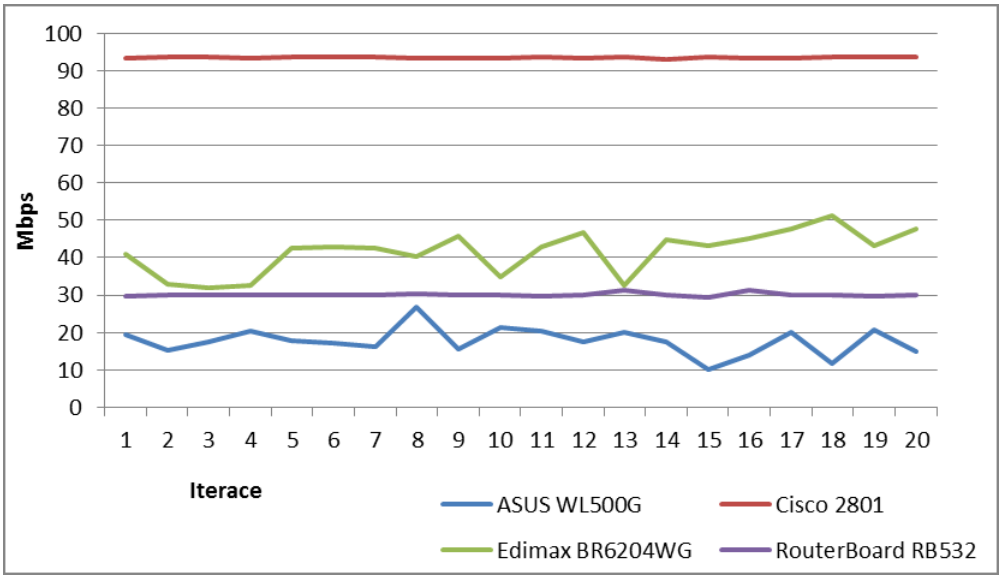
Edimax BR6204WG – Toto zařízení vykázalo druhý nejlepší výsledek v rychlosti odesílání dat protokolem FTP. V opačném směru toku dat ale došlo v 18.iteraci k přerušení navázané relace a stahování bylo zrušeno. U http testu se opět se projevily problémy s navazováním TCP relací jako v případě samostatného testu viz obr.3.15. Během měření se zařízení potýkalo s kolísavými odezvami telnet spojení v rozmezí 1-12ms a s drobnými ztrátami paketů UDP proudu v řádu jednotek procent.

ASUS WL-500G – Směrovač si vedl velmi dobře v rychlosti stahování při FTP testu s naměřenými hodnotami mezi 40 a 50Mbps viz obr. 3.18. Naopak v odesílání testovacího vzorku za ostatním zařízeními zaostával s rychlostmi nepřesahujícími 20Mbps viz obr 3.19. Časový úsek potřebný pro přenos všech dat pomocí http protokolu byl v případě tohoto plného zatížení o 139% delší. Latence se pohybovala v řádech stovek milisekund a počet ztracených datagramových jednotek protokolu UDP se kolísavě vyšplhal až přes 10% viz obr. 3.22

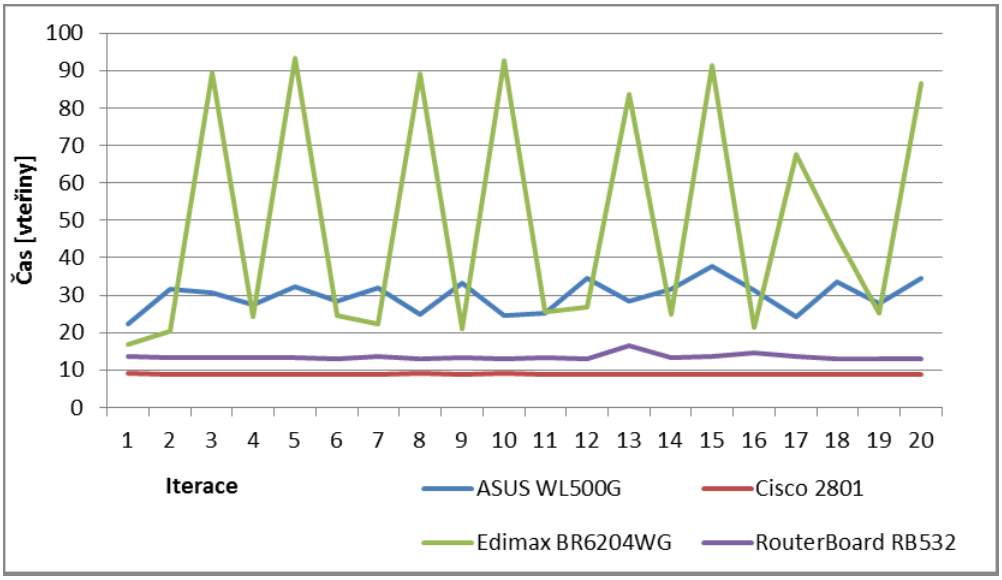
RouterBoard RB532 – RouterBoard stabilně dosahoval při FTP testu přenosových rychlostí okolo 30Mbps. V http testu trvala iterace konstantně přibližně 13 sekund, což znamená nárůst proti nezatíženému testu o 112%. Stále je to však druhý nejlepší výsledek. Odezvy telnet relace byly během testu s několika málo výjimkami 5-6ms a ztrátovost paketů se pohybovala v řádu jednotek.



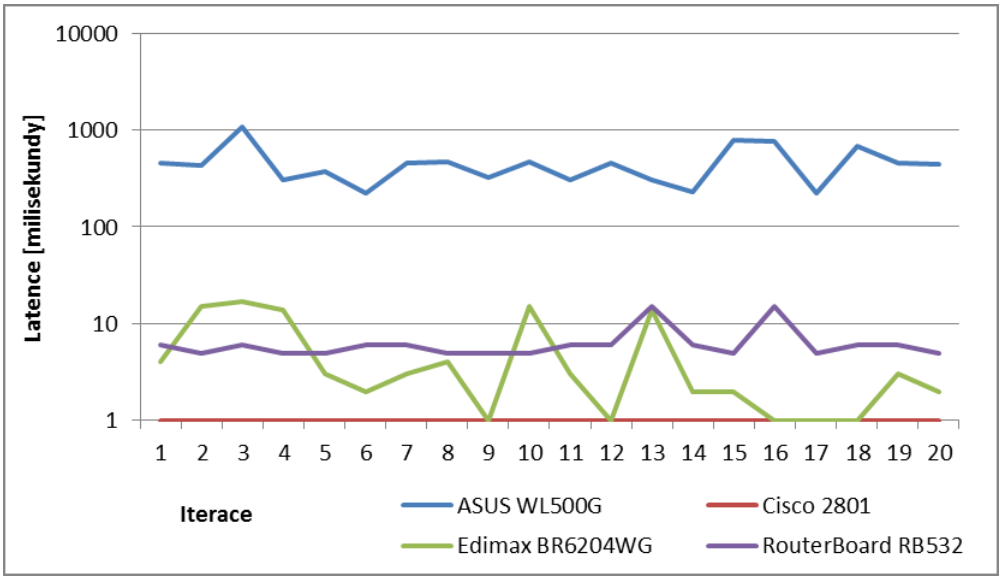
Obrázek 3.18: Graf FTP testu při současném běhu všech testů - rychlost stahování dat



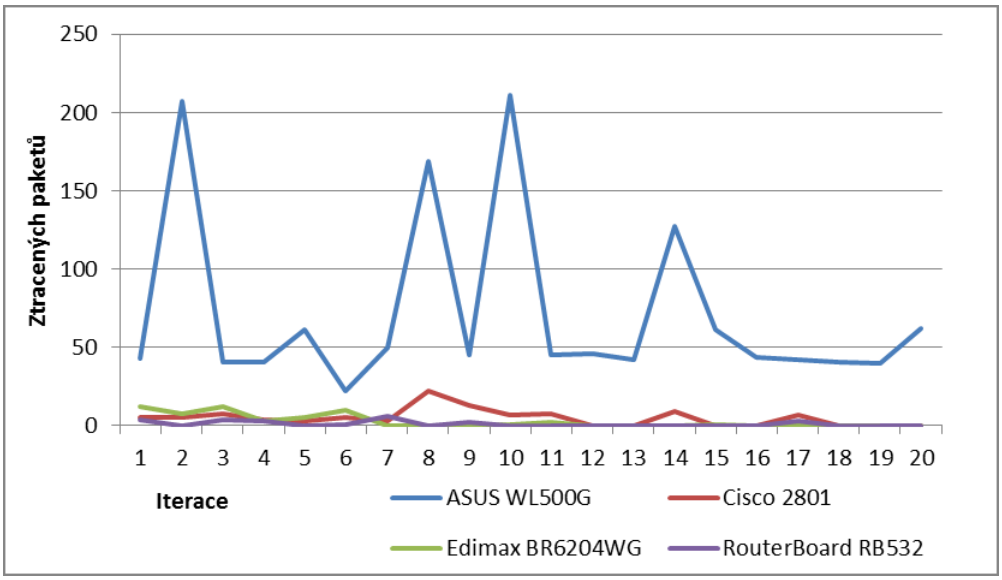
Obrázek 3.19: Graf FTP testu při současném běhu všech testů - rychlost odesílání dat



Obrázek 3.20: Graf HTTP testu při současném běhu všech testů - doba přenosu souborů



Obrázek 3.21: Graf TELNET testu při současném běhu všech testů - průměrná odezva TCP spojení

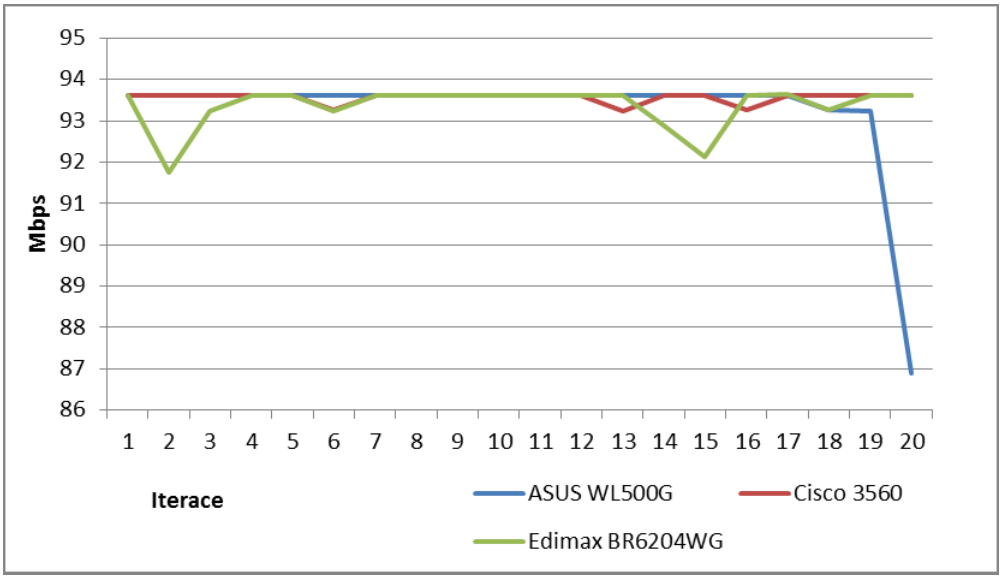


Obrázek 3.22: Graf UDP stream testu při současném běhu všech testů - počet nepřijatých paketů

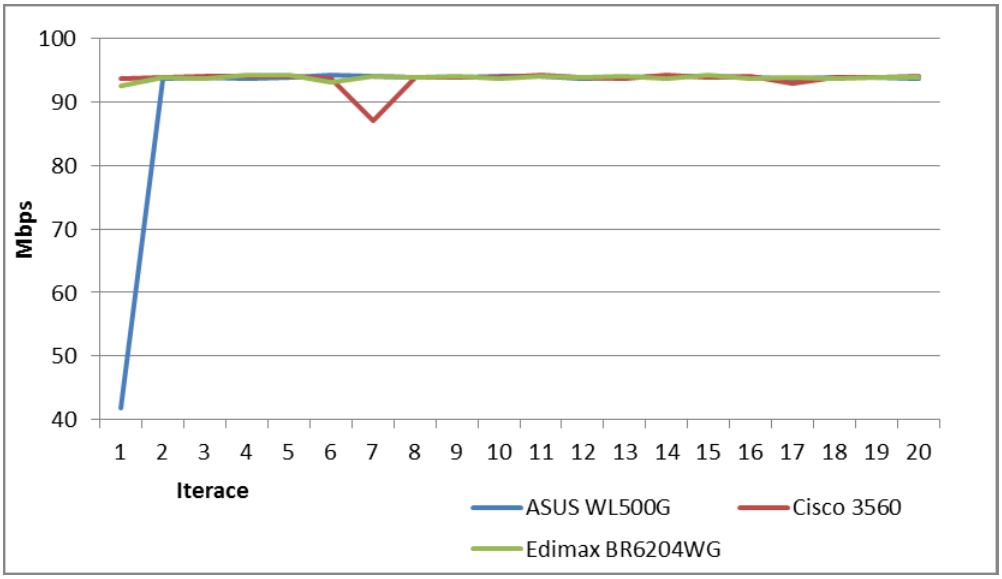
Scénář 6)

Výsledky tohoto scénáře na obrázcích 3.23 a 3.24 demonstrují, že parametry datového přenosu v rámci LAN rozhraní směrovačů jsou srovnatelné s výsledky při použití přepínače.

Tabulky s naměřenými hodnotami jsou přiloženy v příloze.



Obrázek 3.23: Graf FTP testu - rychlost stahování dat



Obrázek 3.24: Graf FTP testu - rychlost odesílání dat

3.5 Porovnání aplikace s vybranými existujícími nástroji

Iperf Nejprve bych se chtěl věnovat srovnání testovací aplikace s testovacím nástrojem iperf. FTP test realizované měřící aplikace má plnit funkci detekce maximálního datového toku. Jinými slovy výstupem je informace o užitečném zatížení TCP protokolu (tzv. payload) během přenosu. Iperf umožňuje nastavit velikost TCP window size, podporuje vícevláknovou komunikaci mezi klientem a serverem a je na rozdíl od měřící aplikace schopen přenášet data v multicast síťovém režimu. I přes širší paletu možností jež lze v nástroji iperf nakonfigurovat, se domnívám, že je pro srovnání výkonnosti 2 zařízení výhodnější vycházet z měření parametrů protokolu, který bude v praxi používán a jehož případné chyby v přenosu mají přímý vliv na uživatelské pohodlí. Jako důkaz mohou posloužit rozdílné výsledky z testovacího scénáře viz. 3.16 a 3.17 proti hodnotám v tabulce 2.1. Pro kompletní analýzu a sledování charakteristik datových toků bych tak doporučil zkombinovat výhody obou zmiňovaných nástrojů.

JMeter Účel použití tohoto nástroje je test výkonnosti běžící služby na serveru. JMeter nepodporuje spouštění instancí testovaných služeb, jako se děje u realizované měřící aplikace (FTP server, web server, UDP stream). Dalším významným rozdílem je, že jde pouze o klienta, který nabízí podstatně více možností testování včetně podpory více paralelních vláken. Použít však tento nástroj za účelem měření výkonnosti síťového zařízení lze pouze za předpokladu, kdy máme jistotu, že je testované zařízení nejslabším článkem v řetězci zařízení(jejich síťových rozhraní) a volných kapacit přenosových linek mezi klientem a serverem.

4 Závěr

4.1 Posouzení vhodného nasazení síťových prvků

Na základě výsledků realizovaných měření vytvořenou testovací aplikací můžeme diskutovat nad vhodným nasazením zařízení do provozního prostředí.

Z výsledků vyplývá, že směrovač **Cisco 2801** je výkonově dostatečný ve všech testovaných směrech. Nebál bych se tak jeho nasazení do role hraničního prvku pro větší síťovou infrastrukturu čítající klidně i desítky síťových prvků a uživatelů. Domnívám se tak na základě výsledků provedeného zátěžového testu, při kterém se výrazně nezhoršil žádný ze sledovaných parametrů datových toků.

Směrovač **ASUS WL500G** bych nedoporučil připojovat u ISP(zkratka), jenž nabízí připojení do internetu o rychlosti v řádu desítek Mbps. Pokud bychom tak totiž učinili a směrovač zatížili přenosy aplikačních protokolů, byli bychom nepříjemně překvapeni zhoršením odezvy do internetu, delší dobou načítání webových stránek při běžném prohlížení a možnými výpadky hlasových a video služeb.

U zařízení **Edimax BR6204WG** jsem objevil poměrně závažný problém s nízkým limitem navázaných TCP spojení. Nedoporučoval bych proto, aby tento síťový prvek fungoval jako výchozí brána do internetu pro více současně pracujících uživatelů. V takovém případě může popsaný jev všechny potrápit.

Síťový prvek **RouterBoard RB532** patří mezi starší a v současnosti již nepodporované modely výrobce Mikrotik. Jeho nasazení do domácího provozu však nic nebrání. V testech zařízení nedosahovalo nejlepších hodnot, zato ale stabilních a bez výkyvů. S menší pravděpodobností tak může nastat situace, kdy jeden uživatel zatíží náročnými datovými přenosy tento prvek do takové míry, aby významně ovlivnil ostatní připojené uživatele v lokální síti.

4.2 Zhodnocení získaných informací a přínosů aplikace

Během práce na tomto projektu jsem se seznámil s testovacími nástroji používanými pro měření parametrů počítačových sítí. Vyzkoušel jsem si navrhnout a provést několik testovacích scénářů s několika vybranými nástroji a na základě výsledků provedených měření navrhnout vlastní měřící aplikaci. Abych mohl navrženou aplikaci realizovat, musel jsem se seznámit s frameworkem SWING a naučit se pracovat s vlákny v prostředí JAVA.

Vytvořená aplikace nabízí kompaktní nástroj pro simulaci datových toků vybraných protokolů a disponuje jednoduchým ovládáním. Může tak snadno sloužit širšímu publiku v pomoci při základní diagnostice síťových zařízení. Aplikaci proto z tohoto hlediska hodnotím jako přínosnou a využitelnou. Popis ovládání se nachází v příloze a dokumentace pak na přiloženém CD.

Seznam zkratek a slovních výrazů

TCP	Transmission Control Protocol, protokol 4.vrstvy ISO/OSI modelu
UDP	User Datagram Protocol, protokol 4.vrstvy ISO/OSI modelu
Latence	Zpoždění ke kterému dojde při přenosu dat sítí
RTT	Round Trip Time, čas mezi odesláním signálu a přijetím potvrzení
ICMP	Internet Control Message Protocol
Qos	Quality of Service
NAT	Network Address Translation
P2P	Peer to peer, typ síťové architektury

Literatura

[Allman et al.(2009)Allman, Paxson„ Blanton] ALLMAN, M. – PAXSON, V. – BLANTON, E. TCP Congestion Control. RFC 5681 (Draft Standard), September 2009. Dostupné z: http://www.ietf.org/rfc/rfc5681.txt.

[Demichelis – Chimento(2002)Demichelis, Chimento] DEMICHELIS, C. – CHIMENTO, P. IP Packet Delay Variation Metric for IP Performance Metrics (IPPM). RFC 3393 (Proposed Standard), November 2002. Dostupné z: http://www.ietf.org/rfc/rfc3393.txt.

[Higgins(2010a)] HIGGINS, T. How We Test Hardware Routers - Revision 3. http://www.smallnetbuilder.com/lanwan/lanwan-howto/31103-how-we-test-hardware-routers-revision-3, 2010a. [Online; navštíveno 8-12-2012].

[Higgins(2010b)] HIGGINS, T. Instructions for the Matrix21 Router Session Test Tool. http://jdsworld.com/wp-content/uploads/2010/09/router_session_test_tool_instructions.pdf, 2010b. [Online; navštíveno 8-12-2012].

[Huston(2013)] HUSTON, G. IPv4 Address Report. http://www.potaroo.net/tools/ipv4/, 2013. [Online; navštíveno 14-3-2013].

[Ixia(2012)] IXIA. IxChariot Specifications. http://www.ixchariot.com/products/datasheets/ixchariot.html, 2012. [Online; navštíveno 12-12-2012].

[Jacobson et al.(1992)Jacobson, Braden„ Borman] JACOBSON, V. – BRADEN, R. – BORMAN, D. TCP Extensions for High Performance. RFC 1323 (Proposed Standard), May 1992. Dostupné z: http://www.ietf.org/rfc/rfc1323.txt.

[Jonkman(2012)] JONKMAN, R. NetSpec. http://www.ittc.ku.edu/netspec/, 2012. [Online; navštíveno 25-10-2012].

[MikroTik(2013)] MIKROTIK. Packet Flow Chart. http://wiki.mikrotik.com/wiki/Manual:Packet_Flow, 2013. [Online; navštíveno 23-3-2013].

[Paessler(2013)] PAESSLER. Features of Webserver Stress Tool. http://www.paessler.com/webstress/features, 2013. [Online; navštíveno 5-1-2013].

[Schön(2012)] SCHöN, O. Kvalitní router s WiFi je základ domácí pohody, otestovali jsme jich hned pět. http://tech.ihned.cz/, 2012. [Online; navštíveno 10-11-2012].

[Schulze – Mochalski(2009)Schulze, Mochalski] SCHULZE, H. – MOCHALSKI, K. Internet Study 2008/2009. http://www.ipoque.com/sites/default/files/mediafiles/documents/internet-study-2008-2009.pdf, 2009. [Online; navštíveno 17-12-2012].

[Sourceforge(2010)] SOURCEFORGE. Iperf. http://sourceforge.net/projects/iperf/files/, 2010. [Online; navštíveno 25-10-2012].

[Tcpreplay(2013)] TCPREPLAY. Tcpreplay FAQ. http://tcpreplay.synfin.net/wiki/FAQ#RunningTcpreplay, 2013. [Online; navštíveno 30-1-2013].

[Wikipedia(2013)] WIKIPEDIA. Packet Injection. http://en.wikipedia.org/wiki/Packet_injection, 2013. [Online; navštíveno 4-1-2013].

A Naměřené hodnoty

iterace	Download	Upload
1	214,39	478,02
2	210,49	440,94
3	208,53	478,02
4	208,53	488,46
5	208,53	487,79
6	208,53	498,66
7	208,65	498,66
8	218,45	498,66
9	208,53	487,13
10	204,82	498,66
11	210,36	488,46
12	222,66	510,03
13	208,53	488,46
14	196,08	498,66
15	202,97	487,79
16	210,49	487,79
17	206,60	488,46
18	208,65	487,79
19	212,36	498,66
20	206,72	546,63

Tabulka A.1: Testovací scénář č.1. Hodnoty jsou v Mbps

	ASUS		Edimax		RouterBoard		Cisco		Linksys	
i.	down	up	down	up	down	up	down	up	down	up
1	53,73	69,94	38,36	60,19	33,88	40,53	93,24	93,99	27,05	41,33
2	52,73	73,81	38,36	43,02	34,03	41,71	93,24	93,70	25,57	72,05
3	53,22	75,38	40,38	38,31	34,08	42,10	93,26	93,67	55,26	43,42
4	53,10	75,46	38,30	39,39	33,78	41,40	93,24	94,24	55,00	72,01
5	53,34	76,47	38,23	58,79	33,78	41,81	93,26	89,19	54,87	34,50
6	53,10	73,04	38,23	63,86	33,98	41,75	92,49	93,70	55,27	71,99
7	53,85	74,42	40,38	58,64	34,13	40,98	93,24	85,70	55,54	72,70
8	53,09	74,68	38,42	57,62	33,93	41,73	93,26	93,75	54,88	34,73
9	54,10	33,08	38,42	64,54	33,88	41,72	92,87	93,87	49,87	72,21
10	52,85	73,82	37,66	59,16	33,93	41,65	86,23	93,85	30,54	34,64
11	53,22	71,23	37,85	64,73	34,03	39,98	92,85	93,90	55,68	71,78
12	53,10	76,16	38,35	58,34	34,03	41,61	93,24	93,70	55,54	71,72
13	52,84	76,08	38,16	39,41	33,98	41,54	92,87	93,80	55,14	72,62
14	53,73	71,78	38,23	42,19	33,88	40,05	93,26	93,67	54,36	68,48
15	53,10	74,59	38,16	62,79	34,03	40,90	93,24	93,92	55,54	72,61
16	53,22	67,36	38,30	58,87	34,03	41,63	93,24	93,78	55,54	72,10
17	52,97	74,94	38,10	64,46	33,98	41,75	93,22	92,51	55,02	72,62
18	53,09	34,65	38,17	58,38	34,03	40,77	93,24	94,10	55,13	71,94
19	53,22	76,40	38,23	61,41	33,88	41,62	93,26	93,90	55,40	69,30
20	51,90	75,87	40,46	59,06	33,88	41,18	93,24	93,06	55,54	72,55

Tabulka A.2: Testovací scénář č.2. Hodnoty jsou v Mbps

iterace	ASUS	Cisco	Edimax	Linksys	RouterBoard	GigabitEth
1	8,10	5,92	8,54	6,23	6,76	4,87
2	8,07	5,12	7,15	6,45	6,32	4,42
3	8,21	5,42	115,32	7,21	7,38	4,58
4	8,16	5,34	6,51	7,21	6,23	4,46
5	8,51	5,35	112,13	8,19	6,26	4,42
6	8,26	5,38	7,59	7,93	6,03	4,34
7	8,21	5,03	6,52	8,34	6,10	4,15
8	8,68	5,10	105,73	8,87	6,14	4,20
9	8,91	5,20	9,82	9,07	7,16	4,23
10	48,14	4,96	10,22	9,08	8,63	4,18
11	9,57	4,76	108,60	9,80	5,88	4,17
12	8,94	5,15	7,35	9,71	5,98	4,28
13	8,79	4,75	117,95	10,54	5,92	4,14
14	8,93	4,75	6,98	10,44	7,15	4,11
15	8,85	5,56	10,38	11,10	5,84	4,12
16	8,90	4,76	115,78	11,13	5,89	4,15
17	8,69	5,01	8,12	10,18	5,93	4,12
18	8,73	5,14	112,51	10,37	5,96	4,15
19	8,87	5,18	7,43	10,08	5,85	5,07
20	45,27	4,92	13,50	10,07	5,90	4,57

Tabulka A.3: Testovací scénář č.3. Hodnoty jsou sekundy

	BRIDGE		NAT		ROUTING	
iterace	down	up	down	up	down	up
1	62,50	73,52	33,88	40,53	45,60	55,27
2	62,50	73,15	34,03	41,71	45,96	55,98
3	62,17	73,29	34,08	42,10	45,87	56,51
4	62,84	73,18	33,78	41,40	45,70	56,57
5	62,32	72,49	33,78	41,81	45,97	56,74
6	62,17	73,04	33,98	41,75	45,60	56,78
7	62,33	73,38	34,13	40,98	46,25	56,70
8	62,16	73,25	33,93	41,73	45,78	56,42
9	62,16	73,34	33,88	41,72	45,60	56,93
10	62,16	72,92	33,93	41,65	45,15	56,62
11	62,32	73,42	34,03	39,98	45,69	56,56
12	62,34	71,42	34,03	41,61	45,60	56,64
13	62,16	73,10	33,98	41,54	45,87	55,43
14	61,98	73,59	33,88	40,05	45,15	56,78
15	61,99	72,99	34,03	40,90	45,87	56,58
16	61,99	72,94	34,03	41,63	46,15	55,66
17	61,66	73,01	33,98	41,75	45,78	57,39
18	61,02	73,20	34,03	40,77	45,78	56,59
19	62,33	72,92	33,88	41,62	45,70	54,29
20	62,50	73,06	33,88	41,18	45,70	57,05

Tabulka A.4: Testovací scénář č.4. Hodnoty jsou v Mbps

	ASUS		Cisco		Edimax		RouterBoard	
it.	down	up	down	up	down	up	down	up
1	47,50	19,56	81,10	93,29	30,07	41,05	35,14	29,77
2	43,14	15,37	80,82	93,67	30,84	32,97	30,18	29,92
3	42,57	17,63	81,10	93,67	33,45	31,82	29,38	30,15
4	40,33	20,30	80,82	93,29	32,32	32,46	30,07	30,15
5	45,71	17,82	80,82	93,67	31,14	42,66	30,01	30,06
6	41,20	17,31	80,80	93,67	30,11	42,89	29,78	30,02
7	42,89	16,22	80,51	93,67	29,84	42,50	34,85	30,15
8	40,69	26,78	80,80	93,31	32,41	40,40	29,50	30,34
9	41,50	15,54	81,10	93,31	27,88	45,90	28,10	29,89
10	39,50	21,34	81,10	93,29	30,97	34,87	29,58	29,98
11	40,26	20,30	81,10	93,67	30,19	42,74	29,20	29,60
12	50,54	17,46	80,82	93,29	29,26	46,74	29,84	29,86
13	39,63	20,07	80,80	93,67	33,74	32,69	35,81	31,29
14	44,74	17,58	80,80	92,92	29,46	44,65	28,50	30,01
15	47,61	10,05	81,11	93,67	31,01	43,22	29,47	29,23
16	43,62	14,02	80,82	93,29	28,47	45,26	29,78	31,37
17	41,88	20,20	80,82	93,29	30,03	47,61	29,71	29,86
18	47,90	11,78	80,82	93,67	0,00	51,22	35,60	29,98
19	40,40	20,54	80,80	93,67	30,27	43,14	30,40	29,82
20	42,42	14,86	80,80	93,65	31,10	47,81	29,29	29,94

Tabulka A.5: Testovací scénář č.5. - FTP test - Hodnoty jsou v Mbps

iterace	ASUS	Cisco	Edimax	RouterBoard
1	22,15	9,00	16,93	13,55
2	31,77	8,75	20,24	13,44
3	30,56	8,83	89,38	13,37
4	27,55	8,74	24,24	13,32
5	32,31	8,74	93,16	13,37
6	28,46	8,85	24,65	13,10
7	32,08	8,80	22,15	13,63
8	24,78	9,14	89,23	13,03
9	33,22	8,83	20,95	13,29
10	24,64	8,99	92,50	12,98
11	25,20	8,83	25,37	13,33
12	34,36	8,69	26,71	13,01
13	28,25	8,71	83,81	16,56
14	31,47	8,80	24,73	13,18
15	37,77	8,74	91,21	13,48
16	31,31	8,72	21,19	14,57
17	24,29	8,86	67,46	13,67
18	33,45	8,74	45,62	13,10
19	27,74	8,69	25,12	12,89
20	34,51	8,66	86,48	13,07

Tabulka A.6: Testovací scénář č.5. - HTTP test - Hodnoty jsou sekundy

	Telnet				UDP stream			
it.	ASUS	Cisco	Edimax	RB	ASUS	Cisco	Edimax	RB
1	462	1	4	6	43	5	12	4
2	426	1	15	5	207	5	8	0
3	1083	1	17	6	41	8	12	4
4	302	1	14	5	41	4	3	3
5	374	1	3	5	61	3	5	0
6	221	1	2	6	22	5	10	1
7	462	1	3	6	50	3	0	6
8	468	1	4	5	169	22	0	0
9	320	1	1	5	45	13	0	2
10	467	1	15	5	211	7	1	0
11	302	1	3	6	45	8	2	0
12	452	1	1	6	46	0	0	0
13	303	1	14	15	42	0	0	0
14	229	1	2	6	127	9	0	0
15	782	1	2	5	61	0	1	0
16	773	1	1	15	44	0	0	0
17	221	1	1	5	42	7	0	3
18	685	1	1	6	41	0	0	0
19	462	1	3	6	40	0	0	0
20	444	1	2	5	62	0	0	0

Tabulka A.7: Testovací scénář č.5. - TELNET a UDP test - Hodnoty jsou v milisekundách u TELNET testu a počet u UDP stream testu

B Uživatelská dokumentace

B.1 Prerekvizity

Aplikace byla vyvíjena a testována v prostředí OS Windows 7, ale její běh je díky 100%-nímu složení z JAVA balíků možný i v dalších prostředích. Spuštění nástroje se provede zadáním příkazu *java -jar Simulant.jar* do příkazového řádku nebo spuštěním skriptu **start.bat**, který se nachází ve stejném adresáři.

Před spuštěním programu je vhodné zkontrolovat nastavení brány firewall a zde případně povolit výjimku pro aplikaci a jí využívané porty. Na serverové straně to jsou defaultně porty: **21,80,2023,4444**. Na klientovi pak pro UDP stream port **5555**.

B.2 Ovládání aplikace

Na obrázku B.1 se nachází okno aplikace po spuštění.

V případě, že se jedná o serverovou instanci, nemusíme již kromě čísla portu nic nastavovat a aplikaci spustíme tlačítkem **Start** v pravém horním rohu. Uživatel provádí volbu režimu klient/server pomocí přepínače v levém horním rohu.

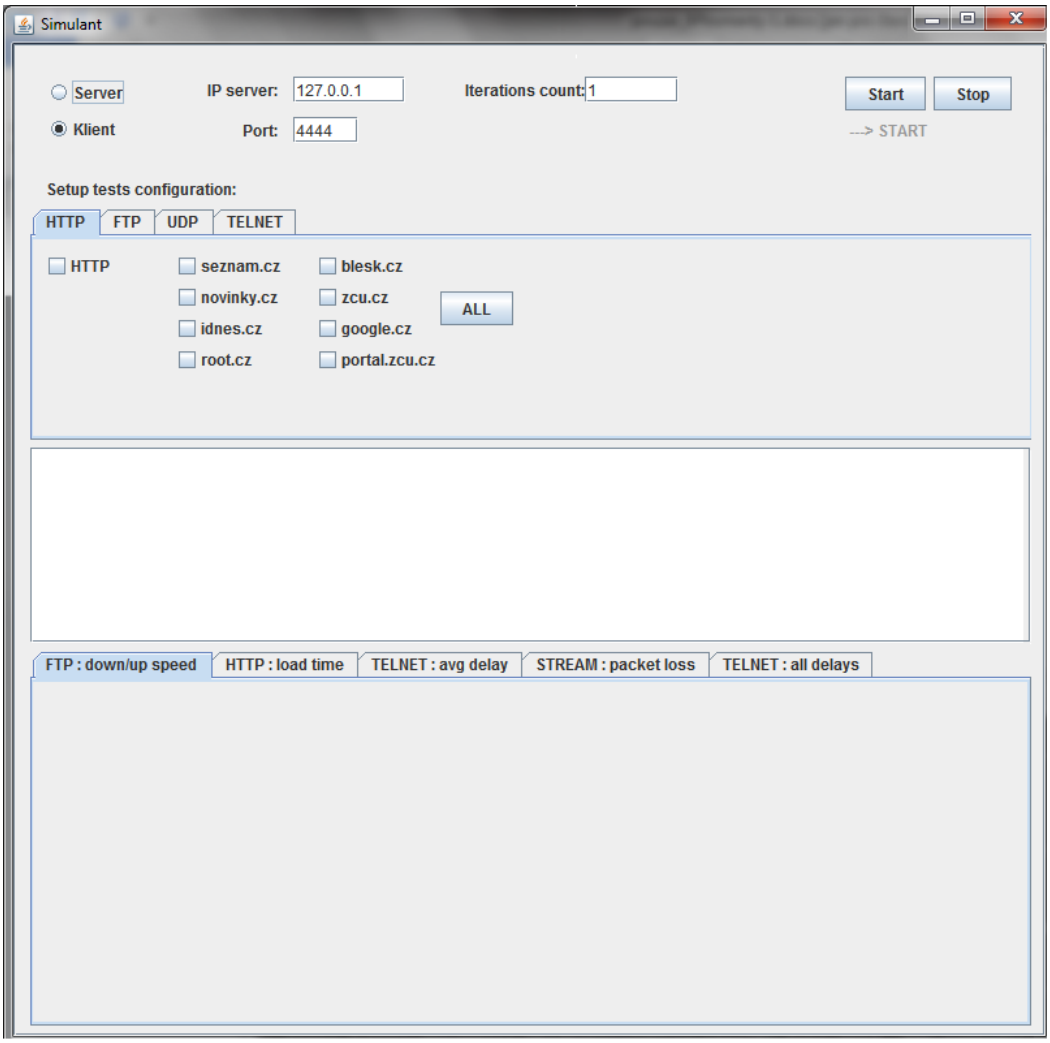
Na klientské straně se provádí volba vybraných testů zaškrtnutím políčka vedle názvu testu. Pod jednotlivými záložkami se pak nacházejí konfigurační rozhraní testů. Metodiky testů jsou popsány v kapitole 3.2. Před spuštěním klienta nastavujeme počet iterací testu a IP adresu + komunikační port serveru.

Informace o průběhu měření lze sledovat po celou jeho dobu. Výstupem je textová konzole a v případě klienta i pravidelně aktualizované grafy ve spodní části aplikace. Běžící klient je na obrázku B.2. Protější serverová strana pak na obrázku B.3.

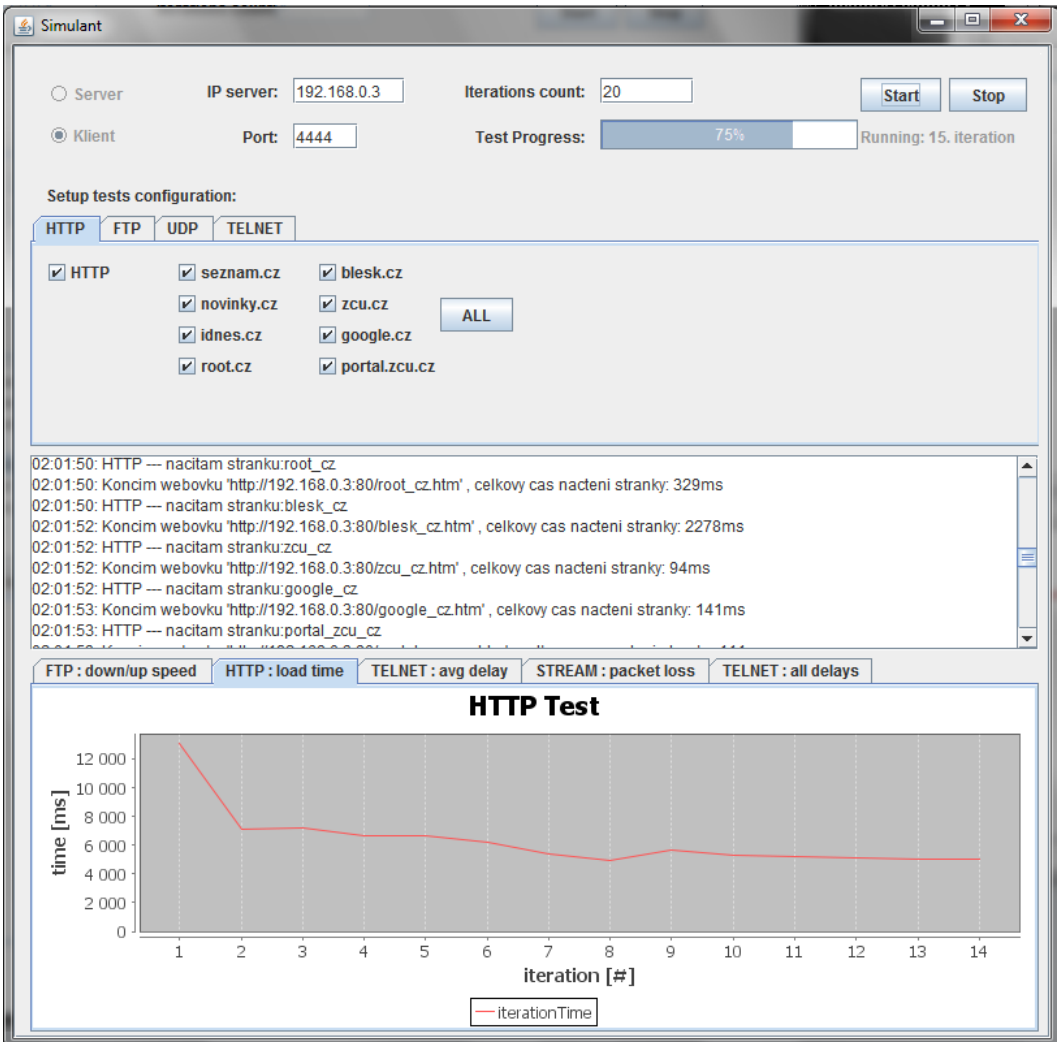
Zastavení běžícího testu provede uživatel tlačítkem **Stop** v pravém horním rohu. Po jeho stisknutí se nastaví aktuálně běžící iterace jako poslední

a měření je ukončeno. Rychlé ukončení aplikace lze provést zavřením okna, klávesovou zkratkou ALT+F4 nebo CTRL+C v okně příkazového řádku se spuštěnou aplikací.

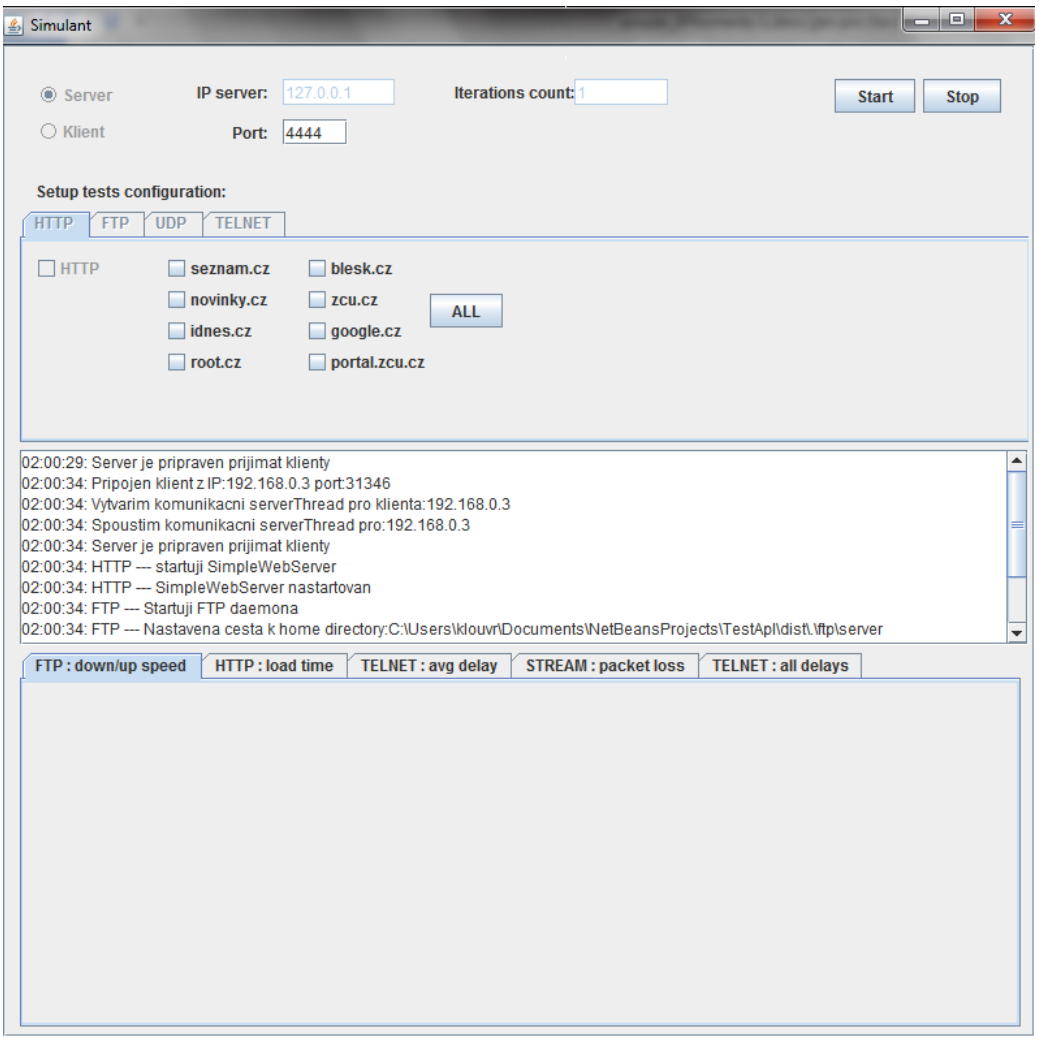
Soubor s výsledky měření *test.db* je vytvořen v adresáři na úrovni spouštěcího JAR souboru aplikace. Po skončení měření je možné soubor prohlížet např. freeware programem SQLite Database Browser. Ke stažení na adrese: `http://sourceforge.net/projects/sqlitebrowser/`



Obrázek B.1: Aplikace po jejím spuštění



Obrázek B.2: Aplikace - klient v průběhu měření



Obrázek B.3: Aplikace - běžící serverová instance programu