VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2017 Tomáš Kolmačka

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

POČÍTAČOVÁ APLIKACE PRO MĚŘENÍ SÍŤOVÝCH PARAMETRŮ V SÍTI
COMPUTER APPLICATION FOR MEASURING PARAMETERS OF NETWORK

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Tomáš Kolmačka**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. David Grenar**
SUPERVISOR

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Student: Tomáš Kolmačka ***ID:*** 174214
Ročník: 3 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

Počítačová aplikace pro měření síťových parametrů v síti

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte odbornou literaturu a elektronické zdroje vztahující se k problematice provozu v TCP/IP sítích a k jejich klíčovým parametrům využívaných v provozu TCP/IP sítích. Zaměřte se na vhodné diagnostické metody pro měření v TCP/IP. Vytvořte počítačovou aplikaci pro měření zadaných parametrů v režimu klient a server, přičemž server bude schopen z naměřených hodnot automaticky generovat grafy. Vytvořte simulační modely dle kterých bude aplikace provádět měření. Porovnejte možnosti využití teoretických předpokladů a výstupu ze simulací.

DOPORUČENÁ LITERATURA:

[1] HUNT, Craig. TCP/IP network administration. Beijing: O'Reilly & Associates, 2002. ISBN 0-596-00297-1.

[2] DOVROLIS, Constantinos. Passive and active network measurement: 6th International Workshop, PAM 2005, Boston, MA, USA, March 31-April 1, 2005 : proceedings. New York: Springer, c2005. ISBN 3540255206.

Termín zadání: 1.2.2017 ***Termín odevzdání:*** 8.6.2017

Vedoucí práce: Ing. David Grenar
Konzultant:

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

ABSTRAKT

Bakalářská práce se zabývá základní teorií síťové architektury TCP/IP, která je využívána v praxi a stanovuje klíčové parametry datových a multimediálních služeb využívajících tyto sítě. Dále se práce zabývá hlavně metodikou měření jednotlivých parametrů a rozebírá různé metody měření a jejich přesnost. V praktické části bakalářské práce je naprogramována počítačová aplikace Network Meter pracující v režimu klient-server, která umožňuje měření vybraných parametrů mezi klientem a serverem s pomocí transportních protokolů TCP a UDP. Při použití protokolu TCP umožňuje aplikace měřit zpoždění, rychlost odesílání dat a rychlost stahování dat. Při použití druhého transportního protokolu, tedy UDP, umožňuje aplikace měřit kolísání zpoždění a ztrátovost paketů. S měřící aplikací Network Meter bylo provedeno testovací měření na připravené domácí síti. Na stejné síti bylo provedeno měření i pomocí aplikace jPerf, aby bylo možné porovnat vytvořenou aplikaci s jinými aplikacemi. Při porovnávacím měření byla měřena rychlost stahování dat.

KLÍČOVÁ SLOVA

Architektura TCP/IP, doporučení k měření, metodika měření, měření v síti, parametry sítě, RFC 2544, standard měření, měřící aplikace, aplikace pro měření parametrů sítí

ABSTRACT

This bachelor’s thesis deals with the basic theory of network architecture TCP/IP which is used in practice and sets out the key parameters for data services and multimedia services using these networks. The thesis mainly deals with the methodology of measuring individual parameters and analyzes various methods of measurement and the accuracy of these methods. In the practical part of the bachelor's thesis a client-server mode application called Network Meter was programmed. This application allows the measurement of selected parameters between client and server using transport protocols TCP and UDP. When using TCP, the application can measure delay, data upload speed and data download speed. When using the second transport protocol, namely UDP, the application can measure jitter and packet loss. With the measuring application Network Meter test measurements were performed in the prepared home network. In the same network measurements were made using jPerf to compare the created application with other applications. In the comparison measurement download speed was measured.

KEYWORDS

TCP/IP architecture, recommendation for measurement, measurement methodology, network measurement, network parameters, RFC 2544, measurement standard, measurement application, application for network measurement

KOLMAČKA, Tomáš. Počítačová aplikace pro měření síťových parametrů v síti. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 45 s. Vedoucí bakalářské práce Ing. David Grenar.

PROHLÁŠENÍ

Prohlašuji, že svoji bakalářskou práci na téma Počítačová aplikace pro měření síťových parametrů v síti jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne ....

(podpis autora)

OBSAH

SEZNAM OBRÁZKŮ

ÚVOD

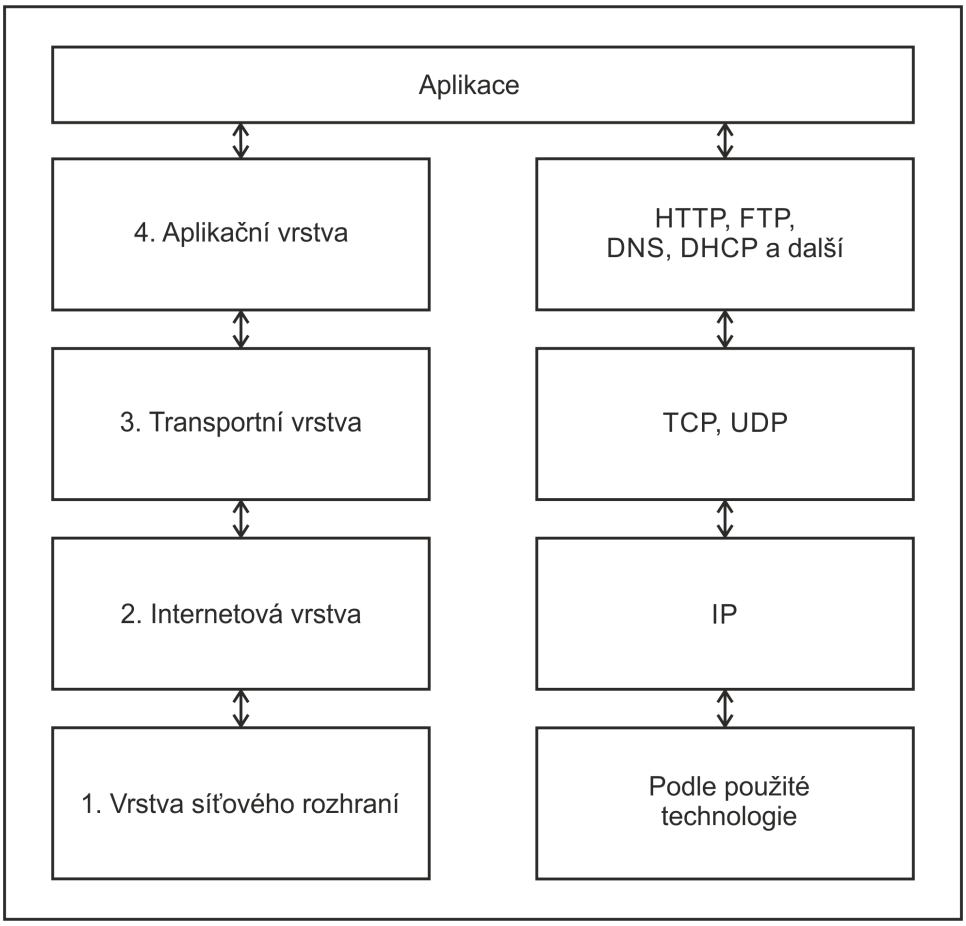
Počítačové sítě se stále rozrůstají a vyvíjí se nové síťové technologie. Stále více se využívá různých měření na těchto sítích podstatných pro správce těchto sítí, kteří používají výsledky k diagnostice sítě nebo zjištění, zda síť vyhovuje z hlediska jejího využití. O tato měření nebo spíše o programy, které provádějí takováto měření, se v posledních letech začínají čím dál více zajímat i domácí uživatelé, kteří si touto cestou chtějí ověřit, zda od poskytovatele dostávají opravdu dohodnuté a zaplacené služby.

Proto se tato bakalářská práce zabývá problematikou provozu v TCP/IP sítích a zvláště pak měřením klíčových parametrů, které mají vliv na přenos datových jednotek. K pochopení problematiky a určení klíčových parametrů sítí je zapotřebí seznámení se s protokoly tzv. rodiny TCP/IP, tj. skupina protokolů, které jsou dnes využívány v největší celosvětové síti Internet. Prozkoumáním architektury této rodiny protokolů, tj. seznámení se s charakteristikou jednotlivých síťových vrstev, získáme informace potřebné k zjištění důležitých parametrů těchto sítí. K jednotlivým vrstvám jsou přiřazeny protokoly a služby, které je využívají. Práce je zaměřena zejména na využití síťových parametrů v praxi a na faktory ovlivňující tyto parametry. Budou určeny klíčové parametry sítí, které budeme měřit. Dále se bude práce zaměřovat na datové a multimediální služby, u kterých budou zjištěny parametry zásadní pro jejich správný a kvalitní provoz. Pro jednotlivé služby a parametry budou stanoveny vhodné diagnostické metody měření. Zaměřeno bude i na to, aby jednotlivé měřící metody byly v souladu s mezinárodními standardy a doporučeními.

Cílem práce bude stanovit metody měření určitých parametrů, klíčových pro jednotlivé datové a multimediální služby. Dále bude cílem implementovat některé metody do počítačové aplikace, která bude pracovat v režimu klient a server. Aplikace bude umožňovat měření zpoždění, rychlosti stahování a rychlosti odesílání dat mezi klientem a serverem. Na straně klienta budou po dokončení měření zobrazeny jednotlivé výsledky. Pro aplikaci se jeví jako vhodný programovací jazyk Java. U tohoto jazyku je využito toho, že je téměř multiplatformní. To umožňuje spuštění testovací aplikace na řadě operačních systémů.

1 SÍŤOVÁ ARCHITEKTURA TCP/IP

Síťová architektura TCP/IP slouží ke komunikaci v počítačových sítích. V celosvětové síti Internet se používá právě tento model architektury vycházející z komunikační, dokonale propracované, architektury ISO/OSI. Architektura TCP/IP byla vytvořena již především pro praktické využití a má čtyřvrstvý model obsahující tyto vrstvy: vrstvu síťového rozhraní, vrstvu síťovou, vrstvu transportní a nejvyšší vrstvu, vrstvu aplikační (graficky zobrazuje Obrázek 1.1).



Obrázek 1.1 Model TCP/IP a protokoly jeho vrstev [2]

1.1 Vrstva síťového rozhraní

Tato vrstva se zabývá komunikací mezi bezprostředními sousedy, zajišťuje přístup k přenosovému médiu. To znamená, že nám specifikuje používané signály (elektrické, elektromagnetické nebo optické), typ média (např.: kroucená dvojlinka, optický kabel),

tvary konektorů (např.: RJ-45, BNC, SC), modulaci, kódování a synchronizaci. Tato vrstva není v modelu TCP/IP blíže specifikována, protože je zcela závislá na použité technologii přenosu.

1.2 Síťová vrstva

Vyšší vrstva, označovaná také jako IP vrstva, díky značnému využívání protokolu IP (Internet Protocol). Vrstva není závislá na přenosové technologii a používá se k adresaci a doručování paketů od příjemce k odesilateli přes mezilehlá zařízení (směrovače). Síťová vrstva se musí vyrovnávat s odlišnostmi jednotlivých dílčích sítí. Některé technologie síťového rozhraní mají limity velikosti zprávy, která může být naráz poslána, proto se síťová vrstva musí vypořádat s fragmentací a defragmentací těchto zpráv. Na této vrstvě můžeme také najít služební protokol ICMP (Internet Control Message Protocol), který slouží k hlášení chyb.

1.2.1 Protokol IP

Protokol IP umožňuje spojení lokálních sítí (LAN) do jedné celosvětové sítě Internet, ve které přenáší data mezi příjemcem a odesílatelem. Data procházející sítí jsou směrována přes směrovače (routery) až k příjemci. Každý směrovač řeší samostatně směrování k dalšímu směrovači. V IP protokolu je každému rozhraní přidělena alespoň jedna IP adresa. V současné době jsou využívány dvě verze IP protokolu 4 a 6.

V protokolu verze 4 je IP adresa čtyřbajtová. Tato adresa je unikátní a univerzální pro konkrétní rozhraní daného zařízení. Souhrn všech adres označován jako adresní prostor obsahuje teoreticky cca 4 miliardy adres. O přidělování IP adres se stará organizace IANA (Internet Assigned Numbers Authority). Adresy IPv4 jsou dvousložkové. To znamená, že jsou tvořeny adresou sítě, ve které se zařízení nachází, a adresou zařízení. Jestliže bity vyhrazené pro adresu sítě nahradíme binární „1“ a ostatní bity nahradíme binární „0“, získáme masku sítě. Pomocí masky sítě lze určit kompletní adresu sítě. Masku sítě je také možno zapsat za lomítko za adresu jako délku prefixu, která vyjadřuje počet binárních jedniček zápisu masky.

U IPv4 je adresní prostor rozdělen do tříd A až E podle rozsahu a masky sítě. To se ale ukázalo jako problémové, a proto bylo zavedeno beztřídní adresování. Nicméně je stále nutné znát třídní rozdělení, protože se používá v některých protokolech. Kvůli neefektivnímu rozdělení adres se přistoupilo k podsíťování. V případě podsíťování do adresy vstupuje třetí složka, tj. adresa podsítě, která vznikne rozdělením adresy zařízení (adresa sítě se nemění).

Nově je v praxi pozvolna nasazován IP protokol verze 6, protože u verze 4 byl

vyčerpán adresový prostor. Adresa nové verze protokolu IP je tvořena šestnácti bajty z důvodu stále se zvětšujícího počtu zařízení, proto je adresní prostor navýšen a pohybuje se v řádu 10^{38}. Dále pak IPv6 přináší zjednodušení formátu záhlaví, redukci směrovacích tabulek směrovačů na globální úrovni a jednotné adresní schéma pro celý Internet i vnitřní sítě. Kromě výhod má IPv6 i jisté nevýhody a to například, že IPv4 a IPv6 nejsou mezi sebou kompatibilní, tudíž se komunikační aspekty musí řešit dvakrát (IPv4 se stále používá) a při komunikaci se musí stanovit, zda se bude používat verze 4 nebo verze 6.

IP protokol podporuje několik služebních protokolů. Mezi jeden z nejdůležitějších patří ICMP protokol (Internet Control Message Protocol), který bude rozebrán níže. Mezi služební protokoly patří také IGMP protokol (Internet Group Management Protocol), sloužící pro dopravu adresných oběžníků. Dále jde o protokoly ARP (Address Resolution Protocol) a RARP (Reverse Address Resolution Protocol), které se starají o zjištění buď MAC adresy (Media Access Control – jednoznačný identifikátor síťového rozhraní) síťového rozhraní pomocí známé IP adresy (ARP) nebo IP adresy pomocí známé MAC adresy (RARP). Tyto dva protokoly jsou brány jako samostatné protokoly, protože nejsou na protokolu IP závislé.

1.2.2 Protokol ICMP

Protokol ICMP je služební protokol a to znamená, že nepřenáší uživatelská data. Je integrovaný do IP protokolu obou verzí (jak do verze 4, tak do verze 6), protože nedisponuje žádnými signalizačními funkcemi. ICMP protokol se používá pro signalizaci mimořádných situací vzniklých v sítích využívajících IP protokol. Datové pakety ICMP protokolu jsou zabaleny do IP protokolu.

Mimořádné situace se dělí do dvou kategorií, a to hlášení chyb a dotazování. Signalizací je spousta, ale v praxi většinou konkrétní implementace TCP/IP podporuje jen potřebné signalizace. Na některých směrovačích je ICMP signalizace z bezpečnostních důvodů zakázána a pakety obsahující tyto zprávy jsou zahazovány [1].

Mezi základní signalizaci z kategorie dotazování, která je ve většině případů podporována, patří žádost o odezvu a odpověď. To slouží k ověření dostupnosti určitého zařízení, na které je odeslána ICMP zpráva „Žádost o odpověď“. Pokud tato zpráva k testovanému zařízení dorazí, je povinno na ni odpovědět. Zmíněnou komunikaci nejčastěji využívá aplikace ping pro ověření dostupnosti určitého zařízení.

1.3 Transportní vrstva

Transportní vrstva se nachází mezi vrstvou síťovou a aplikační. Vrstva je zodpovědná

za zprostředkování služeb ze síťové vrstvy pro aplikační vrstvu. Komunikace na transportní vrstvě je vždy mezi konkrétními procesy (process-to-process communication). Síťová vrstva se stará o doručení zprávy na zařízení, ale už se nestará o doručení zprávy přesné aplikaci. O doručení zprávy pro danou aplikaci (proces) se stará transportní vrstva.

Při adresaci konkrétní aplikace se používá, tzn. číslo portu, které je dvoubajtové a může nabývat hodnot od 0 do 65535. Tato adresace se používá u protokolů pracujících na transportní vrstvě, proto se ještě u těchto čísel specifikuje za lomítkem, zda jde o protokol TCP (např.: 20/TCP) nebo zda jde o protokol UDP (např.: 80/UDP). Touto adresou portu je přesně určena aplikace na zařízení.

1.3.1 Protokol TCP

Protokol TCP (Transmission Control Protocol) je jedním z důležitých protokolů používaných při přenosu dat v TCP/IP sítích. Protokol zajišťuje spojovaný a spolehlivý přenos dat mezi párem konkrétních aplikací. Tzn., že pomocí IP protokolu jsou data doručena skrz síť od zařízení odesilatele k zařízení příjemce, kde jsou pomocí TCP předána určité aplikaci, které mají být doručena.

Protokol TCP je spojovaná služba, proto je mezi dvěma aplikacemi navázáno spojení a je vytvořen virtuální okruh, který je plně duplexní (data jsou přenášena oběma směry současně a nejsou na sobě nijak závislá). Spolehlivý přenos dat je zajištěn číslováním přenášených bajtů, které jsou nazývány jako segmenty. Po přijetí segmentu se odesílá zpráva o jeho přijetí. Pokud je nějaký segment ztracen nebo poškozen, přijímač si ho vyžádá znovu. Integrita přenášených segmentů je zabezpečena kontrolním součtem.

1.3.2 Protokol UDP

Opakem TCP protokolu je protokol UDP (User Datagram Protocol). Ten je nespojovaný a nespolehlivý, ale dokáže přenášet data sítí velmi rychle. Je to hlavně díky tomu, že je nespojovaný, tzn., že nemusí navazovat s příjemcem spojení před odesíláním dat. Dále pak nepotřebuje tak velkou režii v hlavičce, čímž je ušetřeno místo pro data. Jedná se o nespolehlivý protokol, tudíž nejsou posílány zprávy odesílateli o tom, že všechny datagramy došly, jako tomu bylo u protokolu TCP. Při adresaci aplikací na zařízeních se opět využívá čísla portu jako tomu je u TCP protokolu.

Protokol UDP se využívá hlavně u aplikací, které potřebují rychlý přenos dat a nevadí jim, že některé datagramy nedojdou. Protokol UDP se tedy používá například při komunikaci v reálném čase (protokol RTP – Real-time Transport Protocol), tedy

např. při doručování zvukových a obrazových dat. Zajímavou vlastností UDP protokolu oproti TCP protokolu je, že adresátem datagramu nemusí být jen jednoznačná IP adresa, ale adresátem může být skupina stanic nebo oběžník.

1.4 Aplikační vrstva

Prostřednictvím aplikační vrstvy komunikují jednotlivé aplikace s transportní vrstvou. Aplikační vrstva obsahuje velké množství protokolů pro poskytování různých služeb například protokol HTTP (přenos webových stránek) nebo protokol SMTP (přenos elektronické pošty). Tato vrstva bývá často implementována až v aplikaci, která ji využívá pro svoje funkce, ale není to pravidlem. Vrstva také zahrnuje relační a prezentační služby a funkce.

1.4.1 Datové služby

Mezi datové služby můžeme zařadit službu FTP (File Transfer Protocol), která nám zajišťuje bezpečný přenos dat z jednoho zařízení na druhé v rámci přenesení všech dat bez poškození. Pro zabezpečení přenášených dat lze využít službu SFTP, která přenášená data šifruje.

Dále mezi datové služby patří například služba HTTP (Hypertext Transfer Protocol), která přenáší webové stránky a další data mezi uživatelem a serverem. Funguje na principu dotaz a odpověď. Tzn., že uživatel posílá na server (skrze prohlížeč) dotaz na určitou stránku nebo dokument a server, pokud tuto stránku nebo dokument má, odesílá zpět uživateli základní informace o dokumentu/stránce a samotná data. Pokud soubor nenajde nebo nastane nějaká chyba, odesílá zpět uživateli chybovou hlášku.

1.4.2 Multimediální streamovaný obsah

Multimediální služby jsou služby, které pracují s daty v reálném čase, proto není vhodné, aby používaly pro přenos na transportní vrstvě protokol TCP, jako ho používají datové služby. Vhodnější protokol pro multimediální služby je protokol UDP. Ten ale neobsahuje metody jakým způsobem přenášet multimediální obsah ani nezaručuje přenos všech dat. Proto musí být tyto metody řešeny v rámci protokolů na aplikační vrstvě. Protokol používaný pro přenos dat v reálném čase je protokol RTP (Real-time Transport Protocol). Mezi multimediální služby patří například VoIP (Voice over Internet Protocol), IPTV (Internet Protocol Television) a další.

2 PARAMETRY SÍTÍ

2.1 Propustnost

Propustnost nám udává, jak rychle jsme schopni přenést informace přes danou trasu sítě v reálných podmínkách daných její technickou specifikací. Propustnost se udává v násobcích bitů za jednotku času (sekundu). Tato hodnota je nižší než teoretická šířka pásma. Hodnotu také omezuje formát zpráv, charakter komunikace nebo konstrukce a vytížení všech zařízení na trase.

2.2 Rychlost přenosu

Tento parametr nám udává teoretickou hodnotu rychlosti při přenosu, kterou jsme schopni přenášet informace přes danou síť. Rychlost přenosu vyplývá z použitých technických prvků v síti. Tyto prvky mají určenou rychlost přenosu danou výrobcem. Naproti rychlosti přenosu stojí výše zmíněná propustnost, která udává rychlost přenosu změřenou na dané síti.

2.2.1 Rozptyl

Udává rozdíl mezi maximální a minimální hodnotou přenosové rychlosti naměřenou v určitém časovém intervalu.

2.3 Ztrátovost paketů

Pokud při přenosu nastane zahlcení směrovačů a ty začnou zahazovat některé pakety, vzniká tak ztrátovost paketů. Ta je u přenosového protokolu TCP řešena vyžádáním si takového paketu znovu. Tím se eliminuje ztrátovost a vzniká tak pouze větší zpoždění. Toto řešení je ovšem nešťastné pro aplikace, které pracují v reálném čase, které potřebují nejnižší možné zpoždění. Ty proto používají přenosový protokol UDP, který nemá mechanizmy pro znovu poslání ztraceného paketu. Tím vzniká ztrátovost paketů, ale aplikace pracující v reálném čase většinou tolerují do jisté míry to, že některé pakety nedorazí.

2.4 Zpoždění

Důležitým parametrem sítí je také zpoždění, označované jako latence nebo v angličtině delay. Udává čas (v sekundách), jak dlouho skutečně trvá přenos po dané trase. Zpoždění je závislé především na délce trasy, velikosti přenášené zprávy, šířce pásma daného kanálu a také na zatížení trasy.

Zpoždění uváděné v publikaci [2] je většinou 1 ms v lokální síti, ve velkých sítích napříč kontinenty může být zpoždění i 100 ms. Zpoždění má zásadní vliv, pokud je přenášen malý počet dat. S rostoucím počtem přenášených dat vliv zpoždění klesá a začíná mít větší vliv maximální přenosová rychlost (šířka pásma).

Zpoždění můžeme měřit jako jednosměrné nebo obousměrné (RTT – round-trip-time), tj. zpoždění trasy k cíli i zpět. Tato hodnota je rovna minimálně dvojnásobku jednosměrného zpoždění. V reálných podmínkách je tato hodnota vyšší, protože se do ní připočítává i čas, který cílové zařízení potřebuje k odpovědi na vyslanou zprávu, u níž je měřeno zpoždění.

2.5 Kolísání zpoždění

Kolísání zpoždění neboli jitter udává, jak moc se mění zpoždění mezi dvěma po sobě jdoucími příchozími pakety ze stejného proudu. Toto zpoždění mezi příchozími pakety vzniká v souvislosti se zahlcením sítě a řešením frontových mechanizmů v rámci zajišťování služeb na směrovačích. Kolísání zpoždění je hlavní problém u hlasových a video služeb, které podle dokumentu [6] a článku [11] potřebují, aby byl vysílán a přijímán jeden paket každých 20 ms. Při nadměrném kolísání zpoždění začne vypadávat zvuk u hlasových služeb a u video služeb se začne trhat obraz. Aplikace a zařícení jsou navrženy tak, aby dokázaly tolerovat i větší kolísání zpoždění než 20 ms a to díky speciálním algoritmům a ukládáním toku dat do vyrovnávací paměti, ovšem i tak by kolísání zpoždění nemělo přesáhnout 50 ms.

2.6 Kvalita služeb

Hlavním úkolem kvality služeb (QoS – Quality of Service) je zajištění dostatečně kvalitních prostředků sítě pro aplikace, které je vyžadují ke své správné funkčnosti. Patří sem hlavně aplikace pracující v reálném čase, různé hovory, streamovaný obsah a další. Pro zajištění kvality služeb se při komunikaci v síti QoS definují tři základní modely a to služby typu Best-effort, integrované služby (IntServ – Integrated Services) a diferencované služby (DiffServ – Differentiated Services), které obsahují rozdílné

typy mechanizmů pro zajištění kvality služeb v síti.

V modelu Best-effort služby nejsou nijak ovlivňovány a v síti není nijak definována kvalita služeb, tím pádem se data snaží sítí přenést bez zpoždění, co nejrychleji a není řešena datová priorita nebo zda je data možné doručit.

Model integrovaných služeb zajišťuje metody řízení služeb a jejich garanci, které musí být implementovány ve všech zařízeních v síti (koncové stanice, směrovače a další). Přenos dat probíhá tak, že aplikace vyšle síti požadavek pro přenos dat, na který dostane odpověď, zda v daném okamžiku síť disponuje dostatečnými prostředky na to, aby byl přenos dat uskutečněn. Pokud je přenos v tento daný okamžik zamítnut má aplikace požadující přenos dat dvě možnosti. Buď s odesláním dat počkat a po chvíli zkusit, zda se prostředky uvolnily, anebo snížit požadavky pro přenos a znovu odeslat požadavek pro přenos dat. Pokud síť disponuje dostatečnými prostředky a požadavek pro přenos byl přijat, tak jsou všechna zařízení, přes která bude přenos probíhat, obeznámena s touto rezervací prostředků, aby ji nemohla dále přidělovat jiným zařízením. K seznamování zařízení s rezervací prostředků slouží rezervační protokoly, mezi které patří nejčastěji využívaný protokol RSVP (Resource reSerVation Protocol), jenž s sebou nese velkou režii a tím pádem i jistou zátěž pro síť.

Model diferencovaných služeb na rozdíl od modelu integrovaných služeb nezajišťuje prostředky pro přenos dat, ale jednotlivé služby dělí podle jejich požadavků na přenos a poté služby přiřazuje do určitých tříd, které jsou stanoveny předem a musí být známé všem zařízením v síti tak, aby koncová zařízení mohla značkovat odesílané pakety určitou třídou a směrovače na základě značky v paketu jej mohly upřednostnit nebo přesunout do pozadí kvůli jeho menší prioritě.

Kvalita služeb zajištěná modelem integrovaných služeb dokáže velmi přesně nastavit garanci služeb, ale nevýhodou je náročnost na kapacitu sítě kvůli velké režii protokolů, které rezervují prostředky pro přenos na jednotlivých směrovačích. Podle materiálu [6] se tato metoda využívá v podnikové síti nebo může být použita na okrajích sítí na rozdíl od diferencovaných služeb, u kterých je uváděno použití například v rozsáhlých sítích nebo v jádrech sítí. Model diferencovaných služeb oproti modelu integrovaných služeb nese výhodu v jednodušším řízení přístupu, protože pakety jsou značkovány koncovými zařízeními, a také v tom, že není potřeba protokolu, který zajišťuje rezervaci prostředků, čímž se eliminuje zátěž sítě vzniklá režií tohoto rezervačního protokolu.

3 TESTOVACÍ DOPORUČENÍ A TESTY

Při měření parametrů sítí by měla být dodržována jistá doporučení, která jsou uvedena v dokumentech RFC (Request for Comments). Tyto dokumenty se zabývají standardy dnešních sítí. Dříve se zabývaly předchůdcem internetu, tj. sítí ARPANET [10]. Pro nás jsou zajímavá RFC doporučení, která se zabývají měřením parametrů v síti.

Aby bylo při měření dosaženo nejpřesnějších výsledků, musíme zohledňovat řízení provozu, kvůli kterému můžou být omezovány některé nebo všechny služby. Také je potřeba zohlednit, zda na měřené síti neprobíhá monitorování provozu, které by při překročení limitů omezilo nebo zastavilo měřící datový tok. Jednou z dalších věcí, na kterou by měl být brán zřetel je dostupnost služeb na portech. Pokud je na některém portu podporována nějaká služba, neznamená to, že tato služba musí být podporována i na jiném dalším portu. Dále by při měření parametrů neměl probíhat žádný jiný přenos v této měřené síti, aby nedocházelo k ovlivňování výsledků jinými přenosy.

3.1 RFC 1242

Dokument RFC 1242 vydaný skupinou BMWG (Benchmarking Methodology Working Group) ze sdružení IETF (Internet Engineering Task Force) v roce 1991 se zabývá terminologií a definuje základní pojmy v oblasti výkonnostních testů sítí, které se používají v dalších dokumentech navazujících na tento dokument [8].

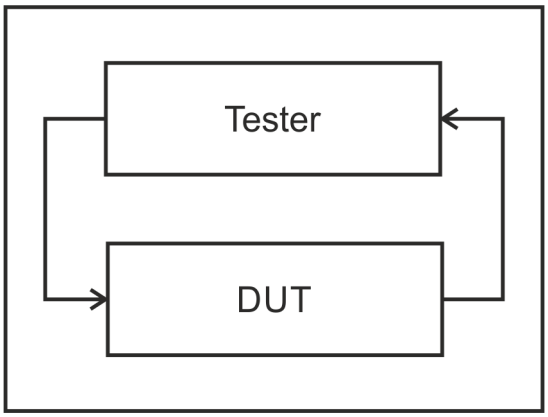
3.2 RFC 2544

Jedná se opět o dokument vydaný skupinou BMWG v roce 1999, který specifikuje nastavení měřených zařízení DUT (Device Under Testing) a měřící metody pro parametry potřebné k vyhodnocení měření na zařízení DUT.

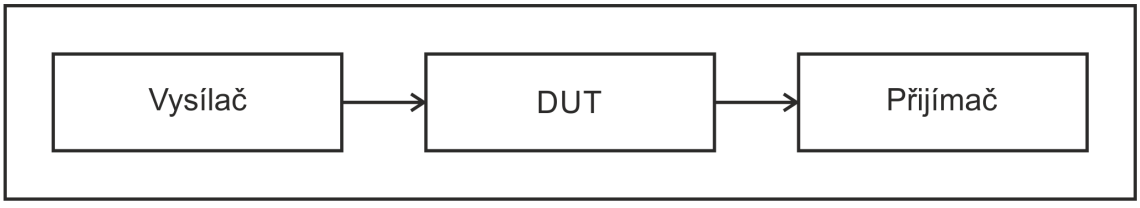
Doporučení dělí požadavky na nastavení zařízení a měření parametrů do tří skupin podle toho, zda musí být splněny (skupina MUST), zda jsou pouze doporučeny (skupina SHOULD), anebo na ty, které jsou volitelné (skupina MAY). Pokud při testu nejsou splněny všechny podmínky MUST je zařízení vyhodnoceno jako nekompatibilní s RFC specifikací. Pokud jsou splněny všechny podmínky MUST a všechny podmínky SHOULD, pak je zařízení vyhodnoceno jako plně kompatibilní se specifikací RFC. A pokud nastane případ, že zařízení splňuje všechny podmínky MUST, ale některé z podmínek SHOULD nesplňuje, je testované zařízení označeno jako podmínečně

kompatibilní se specifikací RFC.

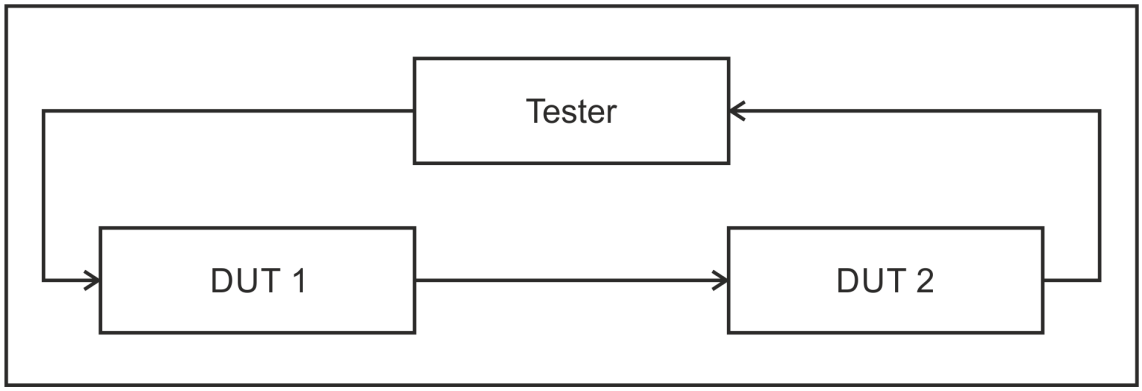
Doporučení obsahující sérii testů, které se provádějí pro standardní délky rámců 64 B, 128 B, 256 B, 512 B, 1024 B, 1280 B a 1518 B, stanovuje tři zapojení měřeného zařízení a testovacího zařízení vhodné pro tyto testy. Při prvním zapojení (Obrázek 3.1) pracuje testovací zařízení jako vysílač i přijímač. U tohoto zapojení je jednoduché vyhodnocení, protože je známo, která data byla poslána a je tedy možné zkontrolovat, zda nenastala chyba. U druhého zapojení (Obrázek 3.2) se využívá dvou testovacích zařízení. Jedno plní funkci vysílače a druhé plní funkci přijímače. Toto zapojení není příliš vhodné, protože některé testy s ním nelze uskutečnit, např. test propustnosti. Třetí zapojení (Obrázek 3.3) se nejvíce podobá reálnému zapojení zařízení v síti. V zapojení je tester plnící funkci přijímače i vysílače, ale cílovému DUT, jsou předřazeny další prvky, které jsou při měření měřeny také. Reálným příkladem tohoto zapojení může být měření zařízení, které se vůči testeru nachází ve vedlejší nebo nadřazené síti. Měřící data tam musí projít přes různá další zařízení stojící v cestě až k cílovému zařízení.



Obrázek 3.1 Tester jako vysílač i přijímač [9]



Obrázek 3.2 Samostatné zařízení jako vysílač i přijímač [9]



Obrázek 3.3 Reálné zapojení v praxi [9]

Před začátkem testování je nutností, aby na měřených zařízeních byla nastavena běžná uživatelská konfigurace. Po dobu testování nesmí být nastavená konfigurace na začátku změněna. Pokud existuje více konfigurací pro uživatele, je prováděno měření pro všechny konfigurace.

3.3 Měření propustnosti

Propustnost, jak bylo popsáno výše, je rychlost, jakou jsou rámce schopny být přenášeny přes danou trasu v síti. Propustnost je měřena ve dvou směrech, a to od testovaného zařízení k testeru (upload, upstream), to znamená, že testované zařízení bude odesílat data na testovací zařízení nebo je možno propustnost měřit od testovacího k testovanému zařízení (download, downstream) tak, že testované zařízení bude stahovat data od testovacího.

Při měření propustnosti podle RFC 2544 je postupováno tak, že na měřené zařízení je poslán určitý počet paketů určitou rychlostí. Podle zvoleného zapojení jsou tyto pakety spočítány buď na testovacím, nebo testovaném zařízení. Pokud je počet přijatých paketů stejný jako počet odeslaných paketů, tzn., žádné pakety nebyly po cestě zahozeny, je rychlost a počet paketů zvýšena a měření se opakuje, dokud nejsou pakety zahazovány. Tím je zajištěno, že propustnost sítě již není dostačující a její hodnota je určena z předešlého měření, při kterém ještě nebyly pakety zahazovány.

U většiny webových nebo volně stažitelných programů měřících propustnost sítě je využíváno metody měření, při které klient stahuje dostatečně velký soubor ze serveru (download) nebo posílá soubor na server (upload). Při tomto měření je počítán počet odeslaných nebo přijatých dat za určitý interval a na základě toho je stanovena přenosová rychlost. Soubor sloužící k tomuto měření musí být tak velký, aby bylo možné provést dostatečně velký počet měření. Zprůměrováním těchto měření je dána výsledná propustnost použité trasy od klienta k serveru v jednom nebo druhém směru.

Pro dosažení větší přesnosti měření je možné použít těchto souborů více o různých velikostech a při velmi rychlém stažení nebo odeslání souboru opakovat toto měření se souborem s větší velikostí, u kterého by mělo měření trvat déle.

3.4 Měření ztrátovosti paketů

Ztrátovost paketů musí být měřena s použitím transportního protokolu UDP, protože při použití protokolu TCP, který zaručuje spolehlivý přenos dat, ztrátovost paketů vzniká. Ale pokud se některé pakety ztratí, jsou odeslány znovu, a tak se ztrátovost paketů projeví tím, že se transformuje do zpoždění a zvětší zpoždění vzniklé při přenosu. Protokol UDP je nespolehlivý, neobsahuje mechanizmy ke kontrole doručených paketů, tudíž je u něj žádoucí měření ztrátovosti paketů.

Měření je prováděno tak, že je odeslán určitý počet paketů na testované zařízení, kde je spočítán počet přijatých paketů. Počet ztracených paketů je zjištěn pouhým odečtením těchto hodnot, ale pro větší přehlednost a využitelnost se ztrátovost paketů *ZP* udává v procentech, tudíž se pro výpočet použije vztah [9]:

$$ZP = \frac{OP - PP}{OP} \cdot 100 \text{ [\%]}, \tag{3.1}$$

kde OP je počet odeslaných paketů, PP je počet přijatých paketů a ZP reprezentuje počet ztracených paketů udávaný v procentech.

Pro první měření ztrátovosti paketů by měla být použita rychlost přenosu odpovídající 100 % maximální přenosové rychlosti pro danou velikost paketu (nutno nejprve provést test propustnosti). Dále se postup měření opakuje s tím, že je snižována maximální rychlost přenosu po 10 % (rychlost přenosu lze snižovat i po méně procentech, ale maximálně po 10 %) dokud není změřena dvakrát po sobě nulová ztrátovost paketů.

Měření by mělo být zobrazeno v grafu, ve kterém musí být na ose X rychlost přenosu paketů v procentech vůči maximální přenosové rychlosti pro určitou velikost paketu. Na ose Y musí být zobrazena ztrátovost paketů v procentech.

3.5 Měření zpoždění

Pro měření zpoždění je nutné nejprve změřit propustnost mezi testovacím a testovaným zařízením kvůli zjištění rychlosti, kterou jsou posílány pakety, pomocí kterých bude měřeno zpoždění. Měřící pakety s určitou velikostí a rychlostí, která byla změřena při

testu propustnosti, jsou posílány na testované zařízení. Proud těchto paketů by měl trvat nejméně 120 s a každých 60 s by měla být odeslána značka, která určuje uplynutí oněch 60 s. Čas, za který byly všechny měřící pakety odeslány, je označen jako časové razítko A. Na straně přijímače je nutné zachytit čas přijetí všech odeslaných měřících paketů, označený jako časové razítko B. Výsledného zpoždění je dosaženo tak, že odečteme časové razítko A od časového razítka B. Tento test zpoždění musí být opakován více než 20krát a z naměřených hodnot zpoždění je stanoven průměr, který odpovídá výslednému zpoždění.

Při měření zpoždění je možno využít metod pro změření doby odezvy, které nedají přesnou dobu zpoždění, ale jako prvotní odhad jsou dostatečné. Jak bylo uvedeno výše, signalizační protokol ICMP pracující na síťové vrstvě dokáže poslat dotaz na dostupnost jistého zařízení. Sledováním odeslané zprávy a příchodem odpovědi lze určit zpoždění mezi testovaným a testovacím zařízením. Nevýhoda této metody je v tom, že v některých sítích je protokol ICMP z bezpečnostních důvodů zakázán. Další problém při využití této metody může nastat v sítích využívajících QoS, kde zprávy protokolu ICMP mohou mít nastavenou nízkou prioritu, čímž se zvětší zpoždění, a tak nebude naměřena korektní hodnota zpoždění.

Podstatně lepší a přesnější metoda zjišťování zpoždění sítě je metoda použitá v práci [12], ve které se pro zjišťování zpoždění sítě využívá protokol HTTP pracující na aplikační vrstvě. Měření probíhá tak, že je zaznamenán čas A před tím, než klient naváže HTTP spojení se serverem. Po navázání spojení klienta se serverem posílá klient požadavek na server a ten posílá zpět odpověď. Po přijetí této odpovědi je zaznamenán čas B a vypočteno zpoždění tím, že od zaznamenaného času B je odečten čas A.

3.6 Měření kolísání zpoždění

Internetový článek [11] popisuje tři přesné metody měření kolísání zpoždění, které jsou založeny na doporučení RFC 4689. U těchto tří metod závisí výsledná hodnota kolísání zpoždění na čtyřech časových parametrech. A to na čase odeslání prvního paketu ve dvojici, čase přijetí prvního paketu ve dvojici, čase odeslání druhého paketu ve dvojici a na čase přijetí druhého paketu z dvojice.

3.6.1 Metoda stejného intervalu vysílání

Při této metodě (Inter-arrival histogram) jsou předdefinovány dva ze čtyř parametrů, které jsou potřebné pro měření a stanovení výsledků. Pakety jsou vysílány v přesně stanoveném intervalu a je tedy měřena pouze doba, za kterou je paket přijat na druhém zařízení (inter-arrival time). Rozdíl časového intervalu mezi pakety se označuje jako

packet-to-packet jitter. Hodnoty intervalu mezi pakety jsou měřeny po určitých časových periodách a následně jsou zobrazeny formou histogramu.

Tato metoda má jedno kritické místo a pár nedostatků, které se týkají přesnosti. Kritické pro tuto metodu je právě nutnost vysílání paketů ve zcela přesných intervalech. Proto lze tuto metodu využít, jen pokud v síti bude konstantní a periodická zátěž s fixní délkou intervalu mezi pakety. Pokud hardware umožňuje měnit velikost paketů s tím, že zachová stále stejný interval vysílání paketů za sebou, mohou být použity pakety různých velikostí. Problém týkající se přesnosti metody nastává v okamžiku, když dojde ke ztrátě paketu (je zahozen nebo poškozen). Tím se velice zvyšuje interval času mezi příchozími pakety a tím je dosažena nepřesná hodnota. Proto je nutné se snažit, aby při měření byla ztrátovost co nejnižší, nejlépe nulová. Stejně tak bude měření nepřesné, pokud nebudou pakety přicházet v takovém pořadí, v jakém byly vyslány.

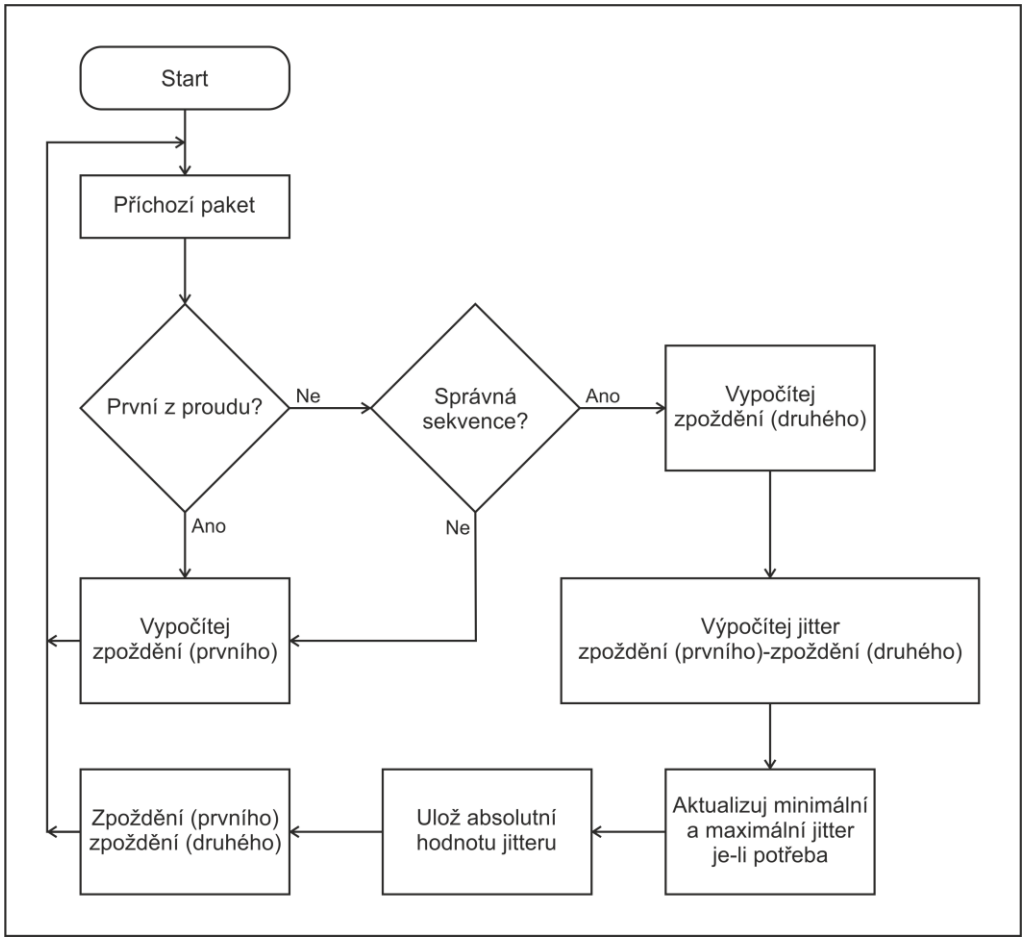
3.6.2 Metoda zachyť vše a zpracuj

Druhá metoda měření (Capture and post-process method) kolísání zpoždění spočívá v tom, že jsou nejdříve zachyceny všechny pakety a z nich je poté určeno výsledné kolísání zpoždění. Většina testovacích zařízení sloužících k zachycení příchozích a odchozích dat jednotlivé pakety značkuje a přidává k nim tak informace jako například čas odeslání a čas přijetí paketu nebo sekvenční číslo, ze kterého lze určit, jak pakety byly přijímány nebo odesílány, ztrátovost paketů, či doručení paketů v správném pořadí.

Limitující pro tuto metodu je velikost paměti pro zachycená data (buffer), která může být velmi rychle vyčerpána, pokud jsou data odesílána a přijímána velkou rychlostí nebo pokud je potřeba data zachytávat delší dobu, než jsou paměti schopny udržet. Nevýhodou této metody je to, že se nejedná o metodu v reálném čase (všechna data jsou nejdříve zachycena a pak se teprve vyhodnocují), to zabraňuje možnosti reagovat v reálném čase na výsledky měření kolísání zpoždění.

3.6.3 Metoda měření kolísání zpoždění v reálném čase

V článku [11] lze najít také metodu pro měření kolísání zpoždění v reálném čase (True, real-time jitter measurement method), která splňuje požadavky průmyslového standardu MEF (Metro Ethernet Forum) 10 z roku 2004. Následující obrázek (Obrázek 3.4) znázorňuje, jak takový test probíhá.



Obrázek 3.4 Diagram pro měření kolísání zpoždění v reálném čase [11]

Pokud je příchozí paket první z proudu dat, potom je vypočteno jeho zpoždění, které je uloženo pro další výpočty. Pokud příchozí paket není první z datového proudu, potom je vyžadováno ověření, zda je paket ze stejné sekvence. Jestliže není ze stejné sekvence, je výsledek zpoždění prvního paketu zahozen a nahrazen výsledným zpožděním tohoto nového paketu. Tímto ověřením je zpřesněno měření, protože eliminuje ztracené pakety nebo ty pakety, které nepřišly ve správném pořadí, jak měly přijít. Pokud je přijat první paket a druhý paket je ze správné sekvence, vypočítá se zpoždění druhého paketu, uloží se a následuje výpočet kolísání zpoždění jako rozdíl zpoždění současného a předešlého paketu. Pokud je potřeba, aktualizuje se záznam minimálního a maximálního kolísání zpoždění a uloží se absolutní hodnota a kolísání zpoždění. Absolutní hodnoty se nepřepisují, ale ukládají se všechny. Z těchto hodnot je možné buď už při měření, anebo až po skončení měření, počítat například průměrnou hodnotu kolísání zpoždění. Nakonec je zpoždění aktuálního paketu uloženo jako zpoždění předešlého a čeká se na další příchozí paket.

3.7 Měření doby obnovy po přetížení

Doba obnovy systému po přetížení se podle dokumentu RFC 2544 [9] měří tak, že na testované zařízení jsou poslány pakety s rychlostí 110 % maximální přenosové rychlosti změřené při měření propustnosti. Tyto pakety jsou posílány minimálně 60 s. Po této době je zaznamenán čas (časové razítko A) a zároveň je zmenšena rychlost posílaných paketů na 50 % maximální přenosové rychlosti. Jakmile je ztrátovost paketů nulová, je zaznamenán čas podruhé (časové razítko B) a odečtením časového razítka A od časového razítka B je získána měřená doba obnovy systému po přetížení. Test by měl proběhnout několikrát a výsledná doba obnovy bude vypočtena jako průměr z naměřených hodnot.

4 POČÍTAČOVÁ APLIKACE

V praktické části bakalářské práce byla vytvořena počítačová aplikace Network Meter pro měření zadaných parametrů TCP/IP sítí skládající se ze serverové a klientské části. Mezi měřené parametry patří měření zpoždění, rychlosti stahování (download) a odesílání (upload) dat při použití transportního protokolu TCP. Dále počítačová aplikace umí měřit ztrátovost paketů (packet loss) a kolísání zpoždění mezi pakety (jitter) při použití transportního protokolu UDP.

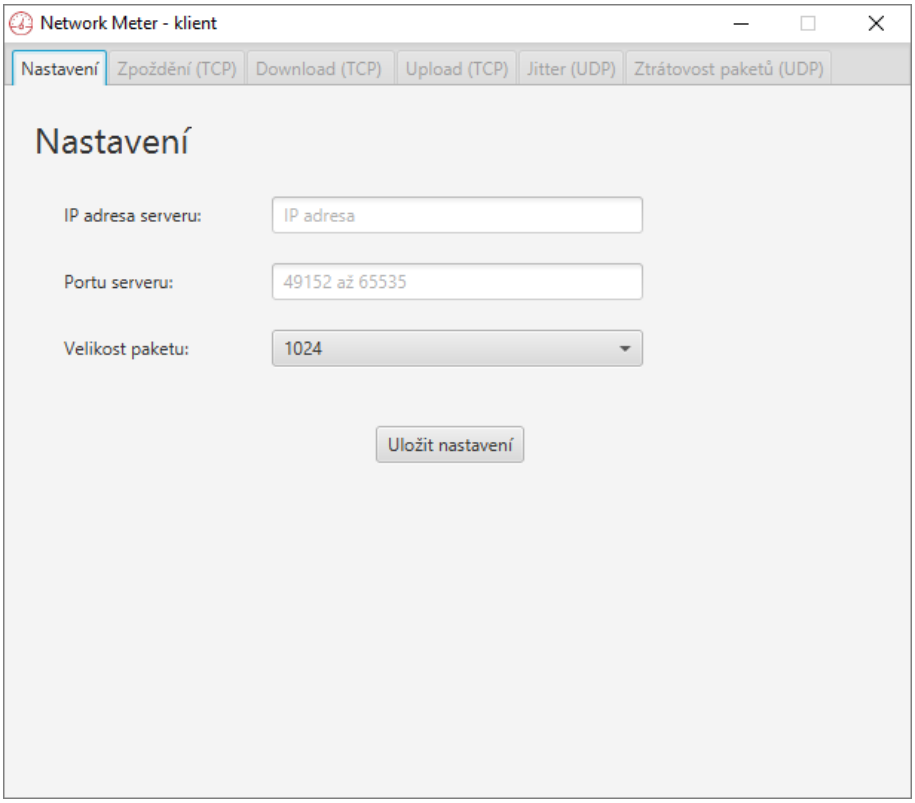
Pro vytvoření aplikace bylo využito objektově orientovaného programovacího jazyka Java. Mezi jeho hlavní přednosti patří přenositelnost na různé operační systémy, robustnost, bezpečnost a jednoduchost [13]. Pro spuštění aplikace je nutné, aby byl na zařízení nainstalován JRE (Java Runtime Environment), což je označení pro softwarový balíček potřebný ke spuštění aplikací napsaných v jazyce Java. Tento balíček je složen z JVM (Java Virtual Machine) a aplikačního rozhraní. JVM je virtuální stroj, na kterém jsou spouštěny Java aplikace. Aby bylo možné programovat aplikace v Javě, je zapotřebí také JDK (Java Development Kit) obsahující nástroje pro programátory.

Java také poskytuje velmi dobrou podporu vícevláknových aplikací, což umožnuje vykonávání několika operací současně. Více vláken se využívá zejména u aplikací s grafickým uživatelským rozhraním, ve kterých se jedno vlákno stará o grafické uživatelské rozhraní a další vlákna vykonávají ostatní operace. Kdyby tomu tak nebylo, grafické rozhraní by se zastavilo a čekalo by, až budou dokončeny ostatní operace. Zastavené nereagující grafické rozhraní aplikace z pohledu uživatele nevypadá esteticky. Uživatel neví, jestli aplikace vykonává operace nebo se její činnost některou chybou zastavila a není možné tuto chybu zobrazit uživateli. Proto je potřeba využít více vláken, která umožní, aby grafické rozhraní aplikace reagovalo, i když budou vykonávány uživatelem vybrané operace.

Celá aplikace je rozdělena na dvě části. Serverovou část běžící na serveru, vůči kterému je prováděno měření a klientskou část, která běží na klientském zařízení a generuje grafy z naměřených hodnot. Serverová část aplikace využívá pro komunikaci s uživatelem textového rozhraní konzole. Klientská část aplikace pro měření a generování grafů je vytvořena na platformě JavaFX, která je určena k vytváření interaktivního grafického rozhraní. Celé aplikační rozhraní JavaFX bylo importováno do Javy, a tak se zápis JavaFX programů neliší od těch klasických psaných v Javě. Aby mohla být využita JavaFX, je nutno do aplikace zahrnout knihovnu JavaFX.

4.1 Popis grafického a textového rozhraní aplikace

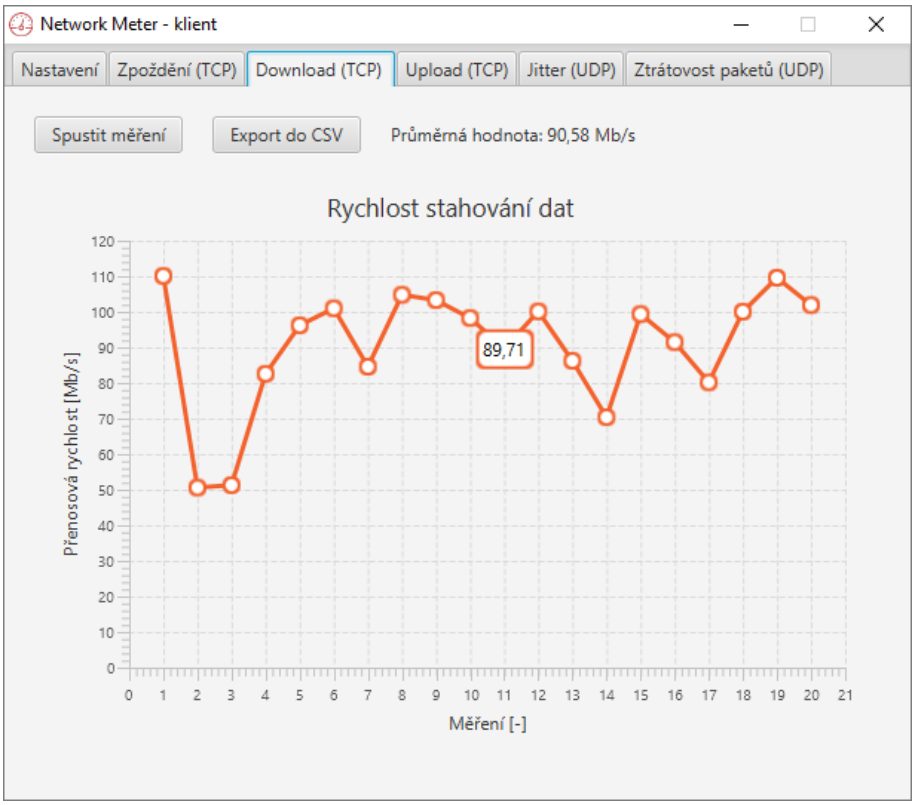
Grafické rozhraní klientské aplikace obsahuje hlavní okno programu, ve kterém je navigace mezi metodami pro měření parametrů sítí a nastavením parametrů programu řešena pomocí šesti záložek. Záložka „Nastavení“ (Obrázek 4.1) se otevře v okně aplikace po spuštění programu a umožňuje nastavit IP adresu a port serveru, vůči kterému budou probíhat měření. Dále je možné nastavit velikost odesílaného paketu na hodnoty 1518 B, 1280 B, 1024 B, 512 B nebo 256 B. Defaultně je nastavena velikost paketu na 1024 B. Tyto údaje je nutné zadat a uložit tlačítkem „Uložit nastavení“ pro odemknutí záložek s měřeními. Při ukládání nastavení je kontrolováno, jestli je správně zadaná IP adresa a číslo portu. Pokud tyto údaje nejsou správně zadány, zobrazí se chybová hláška a je nutné tyto údaje vyplnit znovu. Zadávání správného čísla je také ošetřeno tím, že do textového pole určeného pro číslo portu lze zadat pouze čísla. Aby mohlo být spuštěno měření, musí být server také spuštěný a musí na něm běžet serverová aplikace.



Obrázek 4.1 Grafické rozhraní klientské části aplikace – nastavení

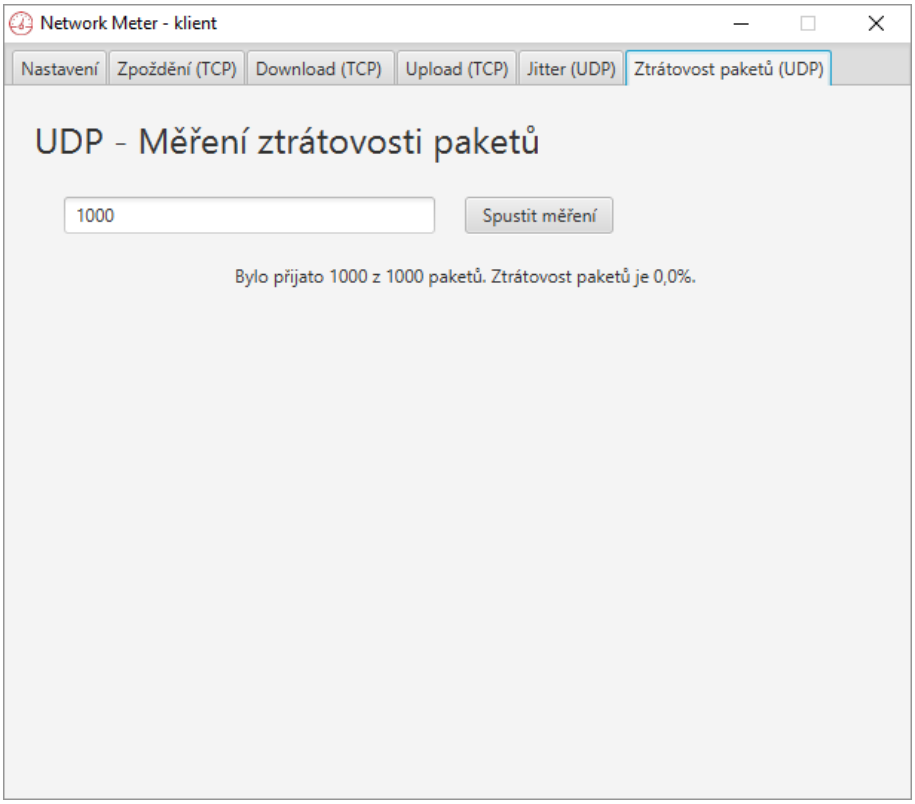
Další záložky jsou určeny pro měření konkrétního parametru sítě. Všechny záložky pro měření vypadají stejně (Obrázek 4.2), liší se pouze záložka určená pro měření ztrátovosti paketů (Obrázek 4.3). Na každé záložce se nachází tlačítko „Spustit měření“,

které příslušné měření zahajuje. Dále se na záložce nachází graf, ve kterém jsou znázorňovány naměřené hodnoty. Graf má název a jednotky na osách podle zvoleného měření. Na ose X je vždy číslo měření bez jednotky a na ose Y je podle měření jednotka buď Mb/s nebo ms. Po najetí kurzoru na uzel v grafu se zobrazí obdélníček s naměřenou hodnotou zaokrouhlenou na dvě desetinná místa. Z grafu lze exportovat naměřené hodnoty do formátu CSV (Comma-separated values) pomocí tlačítka „Export do CSV“. Po kliknutí na toto tlačítko se zobrazí okno pro výběr souboru, ve kterém uživatel vybere buď existující soubor nebo vytvoří nový soubor, do kterého se uloží naměřené hodnoty. Vedle tlačítek je zobrazena průměrná hodnota vypočtená z naměřených hodnot po dokončení měření. Průměrná hodnota je vypočítána jako aritmetický průměr.



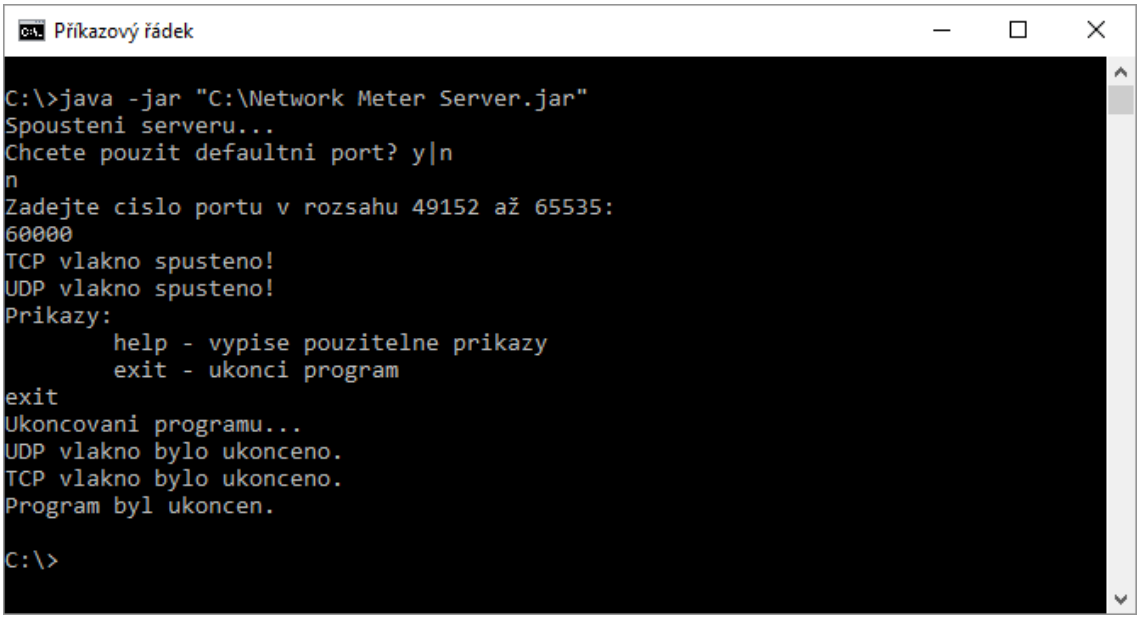
Obrázek 4.2 Grafické rozhraní klientské části aplikace – měření

Na dalším obrázku (Obrázek 4.3) je vidět záložka pro měření ztrátovosti paketů, lišící se svým vzhledem od ostatních záložek. Nachází se na ní textové pole pro zadání počtu odeslaných paketů na server. Do tohoto textového pole je možné zadat počet paketů pouze jako číslo, a to v rozsahu od 1 do 2 147 483 647. Dále se na záložce nachází tlačítko „Spustit měření“ pro spuštění měření a textový řádek, do kterého jsou vypisovány zprávy o měření. Po dokončení měření se zde objeví výsledek měření. Ve výsledku je uvedeno, kolik paketů bylo přijato, odesláno a jaká je ztrátovost paketů v procentech.



Obrázek 4.3 Grafické rozhraní klientské části aplikace – měření ztrátovosti paketů

Serverová část je vytvořena jako konzolová aplikace, tzn. komunikuje s uživatelem prostřednictvím textového rozhraní (Obrázek 4.4). Po spuštění se serverová část aplikace dotazuje uživatele, jestli si chce zvolit vlastní nebo defaultní číslo portu, na kterém bude měření probíhat. Defaultně je nastaven port 55555, ale uživateli je umožněno nastavit si vlastní číslo portu, protože může nastat taková situace, při které bude defaultní port využíván jinou aplikací nebo bude blokován firewallem. Číslo portu lze volit pouze z dynamického nebo soukromého rozsahu 49152 až 65535 [14]. Po nastavení vhodného portu aplikace podá prostřednictvím textového rozhraní hlášení o tom, jestli je připravena na měření pomocí transportních protokolů TCP a UDP. Pokud se nevyskytne nějaký problém, je v tomto stavu serverová aplikace připravena k měření. Textové rozhraní konzole nabízí možnost zadání některých příkazů. Uživatel může zadat příkaz „exit“ pro ukončení serverové části aplikace nebo příkaz „help“ pro vypsání použitelných příkazů a informací o tom, k čemu příkazy slouží. Po zadání příkazu „exit“ spustí aplikace metody pro ukončování a na konzoli je vepsána textová zpráva o ukončování programu. Aplikace začne postupně vypínat obě vlákna využívaná pro měření, což může několik sekund trvat maximálně však deset sekund.



Obrázek 4.4 Textové rozhraní serverové části aplikace

4.2 Vývoj počítačové aplikace

Měřící aplikace se skládá ze serverové a klientské části. Vnitřně jsou tyto části ještě rozděleny na další dvě podle toho, který transportní protokol se využívá k měření.

V podkapitole o RFC 2544 je rozebíráno zapojení testeru a DUT. Z tohoto úhlu pohledu při měření zpoždění, rychlosti stahování a odesílání dat pomocí transportního protokolu TCP aplikace pracuje v zapojení, ve kterém je tester klientská část a ta přijímá nebo vysílá data k serveru čili DUT. Při měření kolísání zpoždění a ztrátovosti paketů obsahuje tester vysílač i přijímač a přes DUT data jen prochází zpět k testeru. Klientská část aplikace má tedy roli testeru a serverová část aplikace má roli DUT.

4.2.1 Serverová část

Serverová část je jednodušší než klientská, protože jen přijímá nebo posílá data. V případě měření za použití transportního protokolu TCP přijímá zprávy od klienta a podle toho přijímá nebo posílá data, případně potvrzuje, že byla data přijata. V případě použití přenosového protokolu UDP slouží serverová část k přeposílání přijatých paketů od klienta a zase nazpět do klientské části.

Serverová část je tvořena třemi třídami, třídou Main, TCPCommunicationThread a UDPCommunicationThread.Třída Main je prováděna v hlavním vlákně programu, zajišťuje komunikaci s uživatelem pomocí textového rozhraní a vytváří nebo ukončuje dvě vedlejší vlákna starající se o komunikaci s klientskou částí aplikace. První vlákno

vytvořené podle třídy TCPCommunicationThread slouží k měření parametru při použití transportního protokolu TCP. Vlákno se stará o vytvoření tzv. TCP socketu, ke kterému se klient připojuje při provádění měření. Jakmile se klient připojí, čeká vlákno na zprávu s informací o tom, které měření bude vykonáváno a podle ní pak provádí příslušnou funkci. Vlákno může dostat tři druhy zpráv „delay“, „download“ a „upload“. Druhé vlákno vytvořené podle třídy UDPCommunicationThread slouží k měření parametrů při použití transportního protokolu UDP. Vlákno čeká, jestli přijde nějaký UDP paket. Jakmile nějaký paket přijde, otevře ho, zjistí adresu odesilatele a odešle ho zpět tam, odkud přišel.

4.2.2 Klientská část

Klientská část aplikace je složitější a obsahuje metody pro správu grafického rozhraní, vyhodnocování měření, generování grafů z měření a jejich export. Klientskou část aplikace tvoří několikrát více tříd než serverovou část.

Základní třídy pro aplikaci s grafickým rozhraním tvořeným pomocí knihovny JavaFX jsou Main vytvářející okno a načítající grafické rozhraní aplikace ze souboru ApplicationView.fxml, ve kterém je uložena šablona vzhledu okna a prvky grafického rozhraní. S tímto šablonovým souborem je provázána potřebná třída ApplicationController starající se pomocí metod o události vzniklé interakcí uživatele a grafického rozhraní.

Třídy DataContainer, UDPDataContainer a TCPDataContainer jsou použity jako kontejnery pro předávání dat mezi ostatními třídami. Obsahují tedy metody pro nastavování a vracení jednotlivých proměnných. Funkční metody jsou obsaženy v těchto třídách pouze minimálně. Třída DialogThrower je využita při ošetření výjimek a umožňuje jednoduše vyvolat dva druhy vyskakovacích oken, a to informativní nebo chybové. Pokud v průběhu vykonávání programu nastane nějaká chyba, je zobrazena ve vyskakovacím okně i s příslušnými informacemi přímo definujícími chybu. Vyskakovacích oken je také využito při exportování naměřených dat z grafu do souboru CSV, kdy se objeví vyskakovací okno při úspěšném uložení souboru. Třída MeasurementLineChart představuje graf měření. Uchovává nastavení jednotlivých grafů a jejich data. Umožňuje přidávání hodnot, vykreslování a mazání grafu v grafickém rozhraní. Dále pak počítá aritmetický průmět z naměřených hodnot uvedených v grafu zobrazovaném uživateli po skončení měření. Třída HoveredThresholdNode slouží k tomu, aby mohla být uživateli zobrazena hodnota při najetí kurzorem na uzel grafu. Přidává tedy do grafu další popisky jednotlivých uzlů, které se zobrazí, pokud na ně uživatel najede kurzorem myši. Tato třída také kontroluje to, jestli uživatel najel kurzorem myši na příslušné místo pro zobrazení popisku.

Dále jsou součástí aplikace třídy TCPMeasurement, TCPReceiverThread a TCPTransmitterThread sloužící k měření při použití transportního protokolu TCP a třídy UDPMeasurement, UDPReceiverThread a UDPTransmitterThread sloužící k měření při použití transportního protokolu UDP. Kombinace tříd pro měření byly voleny tak, aby bylo možné a jednoduché případné doimplementování dalších měřících metod.

Nejdříve byla použita kombinace tříd bez použití kontejneru. Třída pro zapouzdření měření byla v obou případech velice podobná. Byla využívána abstraktní třída, ze které byly odvozeny jednotlivé třídy pro měření. Při tomto způsobu musela být pro všechny měřící metody vytvořena nová třída, což bylo posouzeno jako neefektivní. Efektivnějšího řešení bylo dosaženo využitím čtyř tříd, z nichž jedna byla využívána jako kontejner, druhá byla třída pro zapouzdření a zbylé dvě v sobě nesly metody pro přijímání nebo odesílání při měření.

4.3 Implementované měřící metody

Do aplikace byly implementovány měřící metody určené k měření důležitých parametrů pro datové služby sítě, tj. rychlost stahování a odesílání dat při použití protokolu TCP. Dále pak metody měření parametrů pro multimediální služby sítě, tj. ztrátovosti paketů a kolísání zpoždění. Byla implementována také metoda pro měření zpoždění mezi serverem a klientem.

Měření rychlosti odesílání dat probíhá tak, že je nejdříve navázáno TCP spojení se serverovou částí aplikace a pak je vyslána zpráva „upload“. Po přijetí zprávy se server přichystá na přijímání datových paketů. Poté začne klientská část aplikace posílat data na server, přičemž se počítá počet odeslaných bitů za daný čas. Aplikace generuje náhodná data o velikosti 2 Gb souboru, což by mělo být dostačující pro téměř všechna měření. Po uběhnutí jedné sekundy se spočítá počet odeslaných bitů za sekundu, počítadlo bitů se vynuluje a hodnota se zanese do grafu. Toto počítání probíhá, dokud se neodešlou veškerá data nebo dokud se nezměří potřebný počet vzorků.

Měření rychlosti stahování dat probíhá podobně jako měření rychlosti odesílání dat. Je navázáno TCP spojení serverové a klientské části aplikace. Serverové části je odeslána zpráva „download“, po níž se server připraví na měření rychlosti stahování dat. Klient se po odeslání zprávy přichystá na přijímání dat a server po obdržení zprávy začne vysílat data z náhodně vygenerovaného 2 Gb souboru. Klient stopuje čas a počítá, kolik bitů přijal. Po jedné sekundě je vypočítán počet bitů za sekundu, je vynulováno počítadlo a hodnota je zanesena do grafu. Měření opět probíhá, dokud se nepřenesou všechna data nebo dokud se nezměří potřebný počet vzorků.

Pro měření zpoždění byla zvolena a implementována metoda měření podobná měřící metodě zmíněné v kapitole 3.5 využívající k měření zpoždění protokol HTTP. Stejně jako u metody měření využívající protokol HTTP, je měřen čas od vyslání požadavku až po příchod odpovědi. Měření zpoždění v aplikaci mezi serverem a klientem probíhá tak, že se stopuje čas získání odpovědi (response) generované serverem na žádost (request) generovanou klientem. Klient zaznamená čas a začne navazovat TCP spojení se serverem. Po navázání spojení se odesílá zpráva „request“ na server. Ten ji zpracuje a odesílá jako odpověď zprávu „response“. Po přijetí této zprávy klient opět zaznamená čas, vypočítá zpoždění, zanese je do grafu a ukončuje TCP spojení. Tento postup se opakuje, dokud není provedený příslušný počet měření. Mezi jednotlivými měřeními aplikace vždy vyčká půl sekundy.

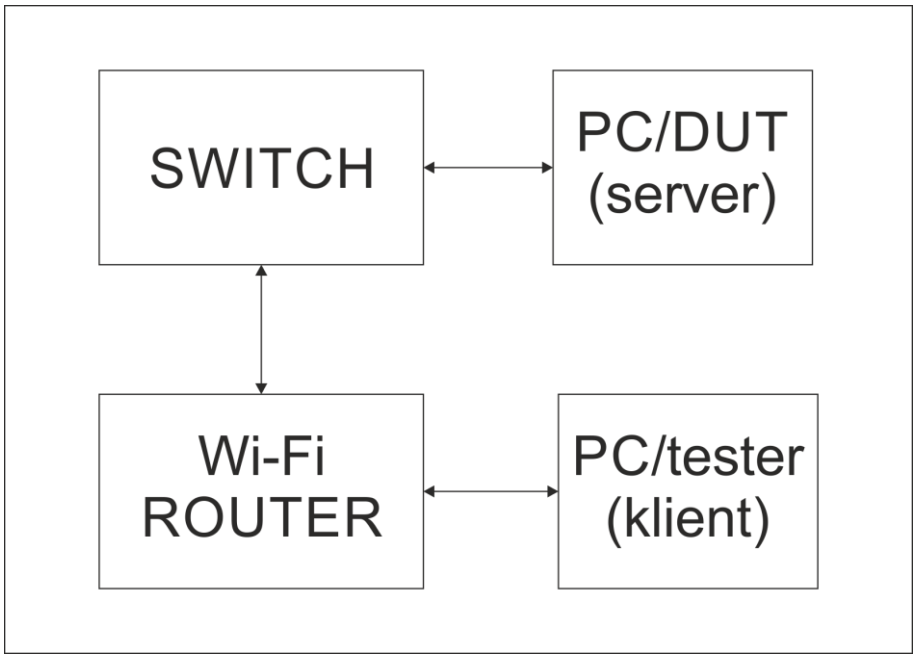
Při měření ztrátovosti paketů je na server posílán uživatelem zadaný počet paketů. Server tyto pakety přijme a odešle je zase zpět klientovi, který zná počet všech paketů, a počítá, kolik paketů se mu vrátilo. Klient má nastavený určitý čas (5 sekund), ve kterém by mu měl dojít paket. Tento čas se po každém přijatém paketu resetuje. Pokud zpoždění paketů přesáhne tento čas, je měření vyhodnoceno jako dokončené a začne se vypočítávat výsledek měření. Ten je pak uživateli zobrazen ve stavovém řádku na záložce s měřením jak v paketech, tak v procentech.

Při měření kolísání zpoždění server opět jen přeposílá přijaté pakety, které jsou nyní číslovány, a posílá je zpět klientovi. Klient pak pracuje téměř podle výše uvedeného diagramu pro měření kolísání zpoždění v reálném čase na obrázku (Obrázek 3.4). Je zaznamenán čas přijetí prvního paketu a čeká se na další paket. Jakmile přijde, je zaznamenán čas jeho příchodu. Díky tomu, že jsou pakety očíslovány, existuje možnost ověření, zda se jedná o pakety po sobě jdoucí. Pokud se jedná o pakety po sobě jdoucí, je vypočteno kolísání zpoždění a je vyneseno do grafu. Pokud se nejedná o pakety po sobě jdoucí, je druhý příchozí paket brán jako první, tj. uloží se čas jeho příchodu a čeká se na další paket. Tento postup se opakuje, dokud není splněn zadaný počet měření.

4.4 Zkušební měření a porovnání s jinými aplikacemi

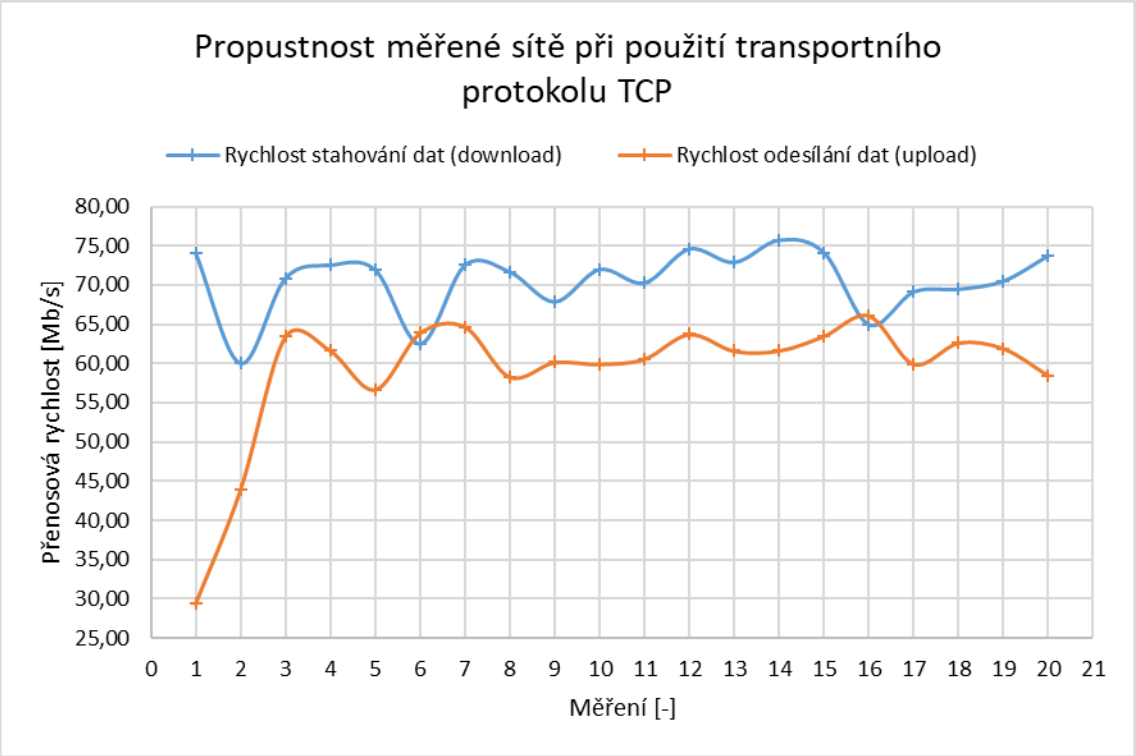
Po vytvoření měřící aplikace bylo provedeno testovací měření na předpřipravené domácí síti s topologií znázorněnou na obrázku (Obrázek 4.5). Serverová část aplikace byla spuštěna na počítači, který je připojený pomocí UTP kabelu k přepínači (switchi) s teoretickou propustností 1 Gb/s. Tento přepínač byl připojen UTP kabelem k LAN rozhraní Wi-Fi směrovače (routeru) s teoretickou přenosovou rychlostí 100 Mb/s. K Wi-Fi směrovači byl bezdrátově připojen druhý počítač, na kterém byla spuštěna klientská část aplikace. Teoretická přenosová rychlost mezi těmito zařízeními byla až

300 Mb/s. Z teoretických přenosových rychlostí vyplývá, že změřená přenosová rychlost by neměla být větší než 100 Mb/s, protože linka mezi Wi-Fi směrovačem a přepínačem má teoretickou rychlost pouze 100 Mb/s. Měření bylo provedeno při minimální zátěži sítě a bylo zaznamenáno dvacet naměřených hodnot při měření zpoždění, kolísání zpoždění, rychlosti stahování a odesílaní dat mezi klientskou a serverovou částí aplikace. Při měření ztrátovosti paketů bylo provedeno pět měření. Dále bylo měření opakováno pro stejnou síť, ale s využitím jiné měřící aplikace, aby bylo možné naměřené výsledky porovnat i s jinými aplikacemi pro měření parametrů TCP/IP sítí. Pro toto porovnání bylo zvoleno pouze měření rychlosti stahování dat ze serveru.



Obrázek 4.5 Topologie měřené domácí sítě

Při měření pomocí vytvořené aplikace v rámci bakalářské práce byly změřeny všechny parametry, které umožňuje aplikace měřit. Z některých naměřených hodnot byly sestaveny grafy. Při měření bylo naměřeno průměrné zpoždění 114,65 ms, rychlost stahování dat byla 70,57 Mb/s, rychlost odesílání dat byla 59,10 Mb/s a průměrné kolísání zpoždění bylo 1,65 ms. Dále bylo provedeno pět měření ztrátovosti paketů při odesílání 10 000 paketů. Při všech pěti měřeních byla naměřena nulová ztrátovost paketů. Z naměřených hodnot odesílání a přijímání dat byl sestaven jeden graf (Obrázek 4.6), na kterém je vidět, že rychlost stahování dat ze serveru je větší než rychlost odesílání dat na server.

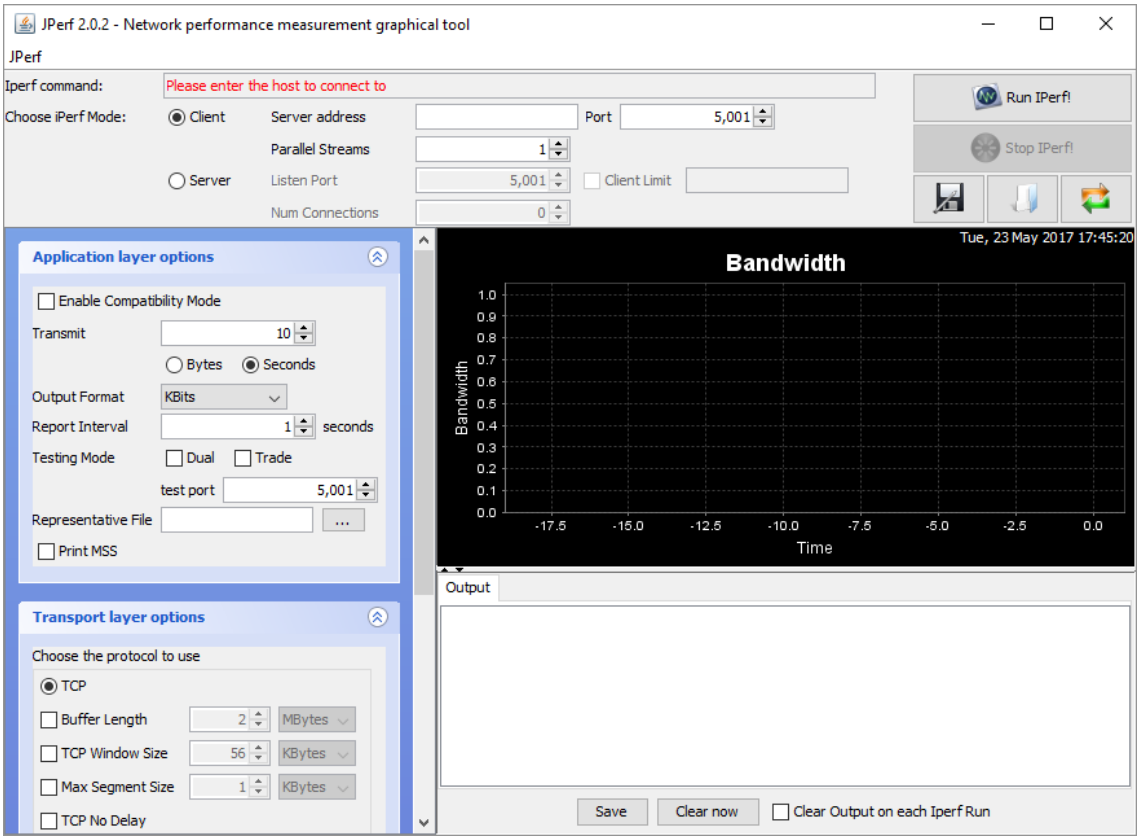


Obrázek 4.6 Graf propustnosti měřené sítě při použití transportního protokolu TCP

Jako jiná měřící aplikace byla vybrána aplikace jPerf verze 2.0.2 [15]. Tato aplikace je grafickou nadstavbou (Obrázek 4.7) pro snadné používání konzolové aplikace iPerf, pomocí které probíhají jednotlivá měření. JPerf pracuje stejně jako vytvářená aplikace v rámci bakalářské práce v režimu klient a server. Navíc je jPerf naprogramována také v programovacím jazyku Java. Aplikace je vydávána pod licencí freeware, což znamená, že plně funkční aplikace je volně k použití bez poplatků s výjimkou toho, že nesmí být nijak upravován zdrojový kód [16]. Aplikace umožnuje měření rychlosti stahování dat ze serveru pomocí transportních protokolů TCP a UDP. Při měření rychlosti stahování dat pomocí protokolu UDP aplikace měří i kolísání zpoždění.

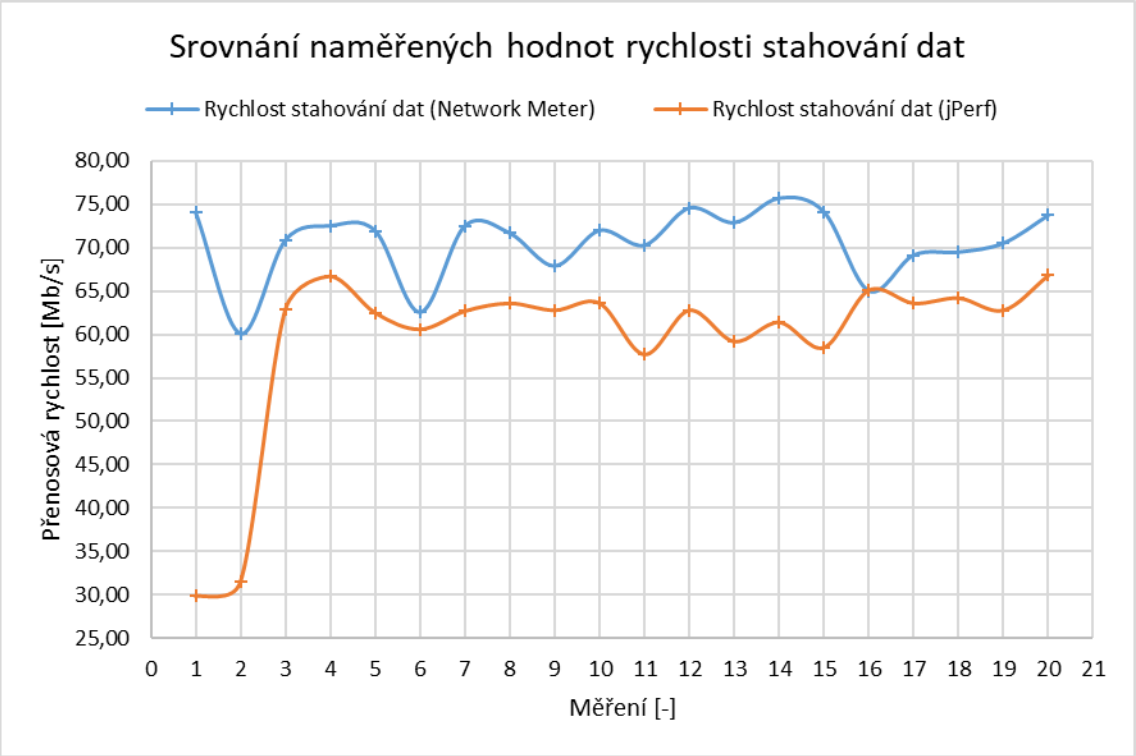
Aby bylo možné provést měření, je nutné spustit aplikaci jPerf na klientském zařízení i na serveru. Po spuštění aplikace je nutno stanovit, zda bude aplikace pracovat v režimu klient nebo server. Pokud je zvolen režim server, je třeba nastavit port, na kterém bude měření probíhat a také musí být vybrán transportní protokol, pomocí kterého se bude měřit. Nakonec je nutno server spustit tlačítkem „Run IPerf!“ Pokud je zvolen klientský režim, je nutné zadat IP adresu serveru a port, na kterém běží aplikace v režimu server. Dále je zde možné opět nastavit, jestli bude měření probíhat pomocí transportního protokolu TCP nebo UDP a kolik dat se má odeslat. Pro spuštění měření se používá také tlačítko „Run IPerf!“.

V aplikaci je také možné nastavení dalších parametrů podle toho, na které vrstvě protokolu TCP/IP se uplatní. Jak je vidět na obrázku grafického rozhraní aplikace (Obrázek 4.7), parametry mohou být nastaveny na aplikační vrstvě, transportní vrstvě nebo na internetové vrstvě, jejíž nabídka není na obrázku (Obrázek 4.7) vidět. Na aplikační vrstvě lze nastavit počet vysílaných bajtů nebo dobu, po kterou se budou data vysílat. Možné je i nastavení po jakém čase budou zapisovány hodnoty do grafu. Aplikace pro měření generuje náhodné hodnoty nebo lze použít vlastní soubor s daty. Pro transportní vrstvu lze nastavit používaný protokol TCP nebo UDP. Pro oba tyto protokoly lze nastavit ještě konkrétní parametry například velikost přenášeného paketu. Pro internetovou vrstvu lze zapnout používání protokolu IP verze 6 nebo dobu platnosti paketu TTL (Time to live).



Obrázek 4.7 Grafické rozhraní aplikace jPerf

Měření rychlosti stahování dat bylo provedeno i aplikací jPerf, pomocí které bylo naměřeno opět dvacet hodnot. Porovnání naměřených hodnot vytvářenou aplikací a aplikací jPerf je možné vidět na grafu (Obrázek 4.8), kde je zobrazeno měření přenosové rychlosti stahování dat.



Obrázek 4.8 Srovnání naměřených hodnot rychlosti stahování dat

Hodnoty při měření různými aplikacemi se od sebe liší. To může být způsobeno použitím různých měřících metod nebo také provozem v sítí. Měření oběma aplikacemi neprobíhalo současně, ale nejdřív bylo provedeno měření pomocí aplikace Network Meter a potom aplikací jPerf. Jelikož v síti byl provoz sice minimální, ale nějaký v ní byl, mohl se ve chvíli měření pomocí aplikace jPerf zvětšit a ovlivnit tak naměřené hodnoty. Pokud nebudeme brát v potaz začátek přenosu (první dvě hodnoty), při kterém je rozdíl hodnot největší, je maximální rozdíl naměřených hodnot téměř 15,62 Mb/s. Průměrná rychlost stahování dat při měření aplikací jPerf byla 59,44 Mb/s a pomocí aplikace Network Metr byla 70,57 Mb/s. Rozdíl naměřených průměrných rychlostí stahování dat je tedy 11,13 Mb/s.

5 ZÁVĚR

V první kapitole bakalářské práce byla probrána základní teorie sítí a hlavně architektura TCP/IP, která se dnes v praxi využívá. Byly rozebrány jednotlivé vrstvy a základní protokoly, které pracují na těchto vrstvách. Dále byly v bakalářské práci rozebrány jednotlivé parametry sítí a byla stanovena jejich důležitost pro datové a multimediální služby. Pro měření jednotlivých parametrů byly nalezeny a popsány některé zajímavé metody měření. Práce se zabývala také standardy a doporučeními pro měření jednotlivých parametrů. Hlavně byl důraz kladen na doporučení RFC 2544, na jehož základě byly některé metody měření a vyhodnocování změřených výsledků popsány.

V praktické části byly některé metody implementovány do počítačové aplikace Network Meter typu klient/server, která zvládne měřit zpoždění, kolísání zpoždění, rychlost stahování a odesílání dat mezi klientskou a serverovou částí aplikace. Klientská část aplikace má grafické rozhraní pro komunikaci s uživatelem a provádí měření vůči serveru. V závislosti na zvoleném měření jsou z naměřených hodnot generovány příslušné grafy. To umožňuje uživateli vidět grafy z naměřených hodnot ihned po měření nebo rovnou při měření. Hodnoty z těchto grafů lze exportovat do souboru ve formátu CSV. V praktické části bylo také provedeno testovací měření aplikace Network Meter v připravené domácí síti. Z naměřených hodnot rychlosti stahování a odesílání dat byl sestaven graf (Obrázek 4.6). U ostatních parametrů byly v textu uvedeny průměrné naměřené hodnoty. Stejná síť byla změřena také měřícím programem jPerf, aby mohly být naměřené hodnoty pomocí vytvořené aplikace Network Meter porovnány s hodnotami měřenými jinou aplikací. Pro toto porovnání bylo vybráno měření přenosové rychlosti. Tento parametr byl tedy změřen oběma aplikacemi a měření bylo vyhodnoceno.

V rámci zadání byly splněny téměř všechny zadané body. Byla prostudována odborná literatura a elektronické zdroje vztahující se k problematice provozu v TCP/IP sítích a jejich klíčovým parametrům. Také byly prostudovány vhodné diagnostické metody pro měření v TCP/IP sítích. Na základě poznatků z teorie byla vytvořena počítačová aplikace pracující v režimu klient/server. Podle zadání by měl server z naměřených hodnot automaticky generovat grafy. V tomto bodu nebylo dodrženo zadání, protože grafy generuje klientská část aplikace. Aplikace je tak uživatelsky přívětivější, protože většinou bude provádět měření uživatel z klientské části aplikace a bude chtít mít grafy ihned zobrazené. Nakonec bylo také podle zadání provedeno a vyhodnoceno simulační měření na předpřipravené domácí síti při sníženém provozu.

LITERATURA

[1] KABELOVÁ, Alena a Libor DOSTÁLEK. Velký průvodce protokoly TCP/IP a systémem DNS. 5., aktualiz. vyd. Brno: Computer Press, 2008. ISBN 978-80-251-2236-5.

[2] JEŘÁBEK, Jan. Komunikační technologie. Brno: Vysoké učení technické v Brně, 2014. ISBN 978-80-214-4713-4.

[3] HUNT, Craig. TCP/IP network administration. Beijing: O'Reilly & Associates, 2002. ISBN 0-596-00297-1.

[4] DOVROLIS, Constantinos (ed.). Passive and acitve network measurement: 6th International Workshop, PAM 2005, Boston, MA, USA, March 31-April 1, 2005 : proceedings. 1st ed. Berlin: Springer-Verlag, 2005. Lecture notes in computer science, 3431. ISBN 3-540-25520-6.

[5] FOROUZAN, Behrouz A. TCP/IP protocol suite. 4th ed. Boston: McGraw-Hill Higher Education, 2010, xxxv, 979 s. ISBN 978-0-07-337604-2.

[6] ČÍKA, Petr. Multimediální služby [online]. Brno: Vysoké učení technické v Brně, 2012 [cit. 2016-11-03]. ISBN 978-80-214-4443-0. Dostupné z: https://www.vutbr.cz/www_base/priloha.php?dpid=61622

[7] VAŇKOVÁ, Jana a Michal ČERNÝ. Parametry počítačových sítí [online]. [cit. 2016-11-04]. Dostupné z: http://clanky.rvp.cz/clanek/c/G/15061/parametry-pocitacovych-siti.html/

[8] BRADNER, Scott (ed.). Request for Comments: 1242: Benchmarking Terminology for Network Interconnection Devices [online]. Cambridge: Harvard University, 1991 [cit. 2016-11-25]. Dostupné z: https://www.ietf.org/rfc/rfc1242.txt

[9] BRADNER, Scott a Jim MCQUAID (eds.). Request for Comments: 2544: Benchmarking Methodology for Network Interconnect Devices [online]. Cambridge: Harvard University, 1999 [cit. 2016-11-25]. Dostupné z: https://www.ietf.org/rfc/rfc2544.txt

[10] ŠKORPIL, Vladislav. Přístupové a transportní sítě [online]. Brno: Vysoké učení technické v Brně, 2012 [cit. 2016-11-08]. ISBN 978-80-214-4457-7. Dostupné z: https://www.vutbr.cz/www_base/priloha.php?dpid=68661

[11] Measuring jitter accurately. LightwaveOnline [online]. Jim Anuskiewicz, 2008 [cit. 2016-11-25]. Dostupné z: http://www.lightwaveonline.com/articles/2008/04/measuring-jitter-accurately-54886317.html

[12] KUNC, Jaroslav. Android aplikace pro měření uživatelské spokojenosti s mobilní datovou službou. Brno: Vysoké učení technické v Brně, 2015.

[13] SCHILDT, Herbert. Java: a beginner's guide. Sixth edition. New York: McGraw-Hill Education, 2014. ISBN 9780071809252.

[14] Service Name and Transport Protocol Port Number Registry. Internet Assigned Numbers Authority [online]. [cit. 2017-05-18]. Dostupné z: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

[15] Freeware: jPerf prověří propustnost sítě. PC World.cz [online]. Martin Buchta, 2012 [cit. 2017-05-23]. Dostupné z: http://pcworld.cz/downloads/freeware-jperf-proveri-propustnost-site-45139

[16] Freeware Definition. The Linux Information Project [online]. The Linux Information Project, 2004 [cit. 2017-05-23]. Dostupné z: http://www.linfo.org/freeware.html

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

CP	Čas počítání paketů
OP	Odeslané pakety
PP	Přijaté pakety
SP	Spočítané pakety
ZP	Ztrátovost paketů

ARP	Address Resolution Protocol
ARPANET	Advanced Research Projects Agency NETwork
BMWG	Benchmarking Methodology Working Group
BNC	Bayonet Neill Concelman connector
CSV	Comma-separated values
DHCP	Dynamic Host Configuration Protocol
DiffServ	Differentiated Services, diferencované služby
DNS	Domain Name System
DUT	Device Under Testing
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protokol
IntServ	Integrated Services, integrované služby
IP	Internet Protocol
IPTV	Internet Protocol television
IPv4	Internet Protocol verze 4
IPv6	Internet Protocol verze 6

ISO	International Organization for Standardization
OSI	Open Systems Interconnection
JDK	Java Development Kit
JRE	Java Runtime Environment
JVM	Java Virtual Machine
LAN	Local Area Network
MAC	Media Access Control
MEF	Metro Ethernet Forum
QoS	Quality of Service, kvalita služeb
RARP	Reverse Address Resolution Protocol
RFC	Request for Comments
RJ-45	Registered Jack-45
RSVP	Resource reSerVation Protocol
RTP	Real-time Transport Protocol
RTT	Round-trip time, obousměrné zpoždění
SC	Subscriber Connector
SFTP	Secure Shell File Transfer Protocol
SMTP	Simple Mail Transfer Protocol
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TTL	Time to live
UDP	User Datagram Protocol
UTP	Unshielded Twisted Pair
VoIP	Voice over Internet Protocol

A OBSAH PŘILOŽENÉHO CD

Na přiloženém CD se nachází elektronická verze této bakalářské práce a soubor data.xlsx obsahující naměřené hodnoty a grafy vytvořené z těchto hodnot. Tento soubor je možné otevřít v programu Microsoft Excel.

Dále se na přiloženém CD nachází archiv NetworkMeter.zip, který obsahuje zdrojové kódy serverové a klientské části počítačové aplikace. Tyto zdrojové kódy je možné zobrazit ve vývojovém prostředí Eclipse, které je zdarma ke stažení na webové stránce https://eclipse.org/. Aplikaci lze spustit buď v tomto vývojovém prostředí nebo pomocí spustitelných JAR souborů nacházejícími se také na přiloženém CD.

B TABULKY NAMĚŘENÝCH HODNOT

B.1 Měření pomocí aplikace Network Meter

Měření	Zpoždění	Kolísání zpoždění	Rychlost stahování dat	Rychlost odesílaní dat
[-]	[ms]	[ms]	[Mb/s]	[Mb/s]
1	120	3	74,07	29,34
2	105	1	60,06	43,91
3	115	1	70,85	63,41
4	105	0	72,53	61,63
5	104	0	71,93	56,68
6	111	2	62,52	63,88
7	107	1	72,52	64,66
8	111	0	71,67	58,29
9	112	6	67,89	60,15
10	103	0	71,99	59,90
11	113	3	70,28	60,59
12	108	1	74,58	63,72
13	108	5	72,89	61,59
14	111	0	75,71	61,63
15	107	2	74,12	63,50
16	111	0	64,99	66,10
17	107	2	69,14	59,94
18	123	1	69,48	62,63
19	206	5	70,52	61,92
20	106	0	73,73	58,48

B.2 Měření pomocí aplikace jPerf

Měření	[-]	1	2	3	4	5	6	7	8	9	10
Přenosová rychlost	[Mb/s]	29,8	31,5	62,9	66,7	62,5	60,6	62,7	63,6	62,8	63,6

Měření	[-]	11	12	13	14	15	16	17	18	19	20
Přenosová rychlost	[Mb/s]	57,7	62,8	59,2	61,4	58,5	65,1	63,6	64,2	62,8	66,8