



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

ANALYZÁTOR BEZDRÁTOVÉHO KOMUNIKAČNÍHO SPOJE

WIRELESS COMMUNICATION LINK ANALYZER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jakub Dostálek

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Jílek, Ph.D.

BRNO 2022



Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Jakub Dostálek

ID: 221292

Ročník: 3

Akademický rok: 2021/22

NÁZEV TÉMATU:

Analyzátor bezdrátového komunikačního spoje

POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce je návrh, implementace a ověření funkčnosti analyzátoru komunikačních parametrů bezdrátového spoje. Práce je zaměřena především na měření parametrů významných pro mobilní robotiku (ztrátovost, zpoždění, kapacita front, atd.) v závislosti na pracovních podmínkách (vytížení, vzdálenost, atd.). Měřicí systém bude určen pro komunikační moduly disponujícími rozhraním Ethernet a podporujícími protokol IPv4.

1. Proveďte rešerši již existujících řešení a dostupných nástrojů pro realizaci.
2. Popište významné charakteristiky přenosové trasy a vhodný způsob grafického vyjádření jejich vzájemných závislostí v souvislosti s pracovními podmínkami.
3. Specifikujte požadované parametry analyzátoru pro uvažovanou aplikaci.
4. Navrhněte analyzátor splňující parametry z předchozího bodu.
5. Proveďte realizaci analyzátoru a ověřte jeho vlastnosti a správnou funkci na přenosovém spoji o známých parametrech.
6. Implementujte automatizované zpracování a vyhodnocení naměřených dat (post-processing).
7. Proměřte dostupný bezdrátový spoj v konkrétní aplikaci a na základě výsledků měření doporučte vhodné nastavení.

DOPORUČENÁ LITERATURA:

COWLEY, John. Communications and Networking: An Introduction. 2nd ed. London: Springer-Verlag, 2013. ISBN 978-1-4471-4356-7.

Termín zadání: 7.2.2022

Termín odevzdání: 23.5.2022

Vedoucí práce: Ing. Tomáš Jílek, Ph.D.

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Cílem této bakalářské práce je navrhnout a realizovat analyzátor bezdrátového komunikačního spoje v mobilní robotice, využívající protokol IPv4 a rozhraní Ethernet a z naměřených dat vytvořit správně zvolené a vyjádřené charakteristiky, podle kterých se bude moci bezdrátový komunikační kanál dále optimalizovat pro nejlepší přenos dat. Pro splnění těchto požadavků byla provedena stručná rešerše již existujících řešení, byla diskutována jejich použitelnost a byly definovány významné parametry bezdrátového přenosového kanálu, které budou dané charakteristiky tvořit. Dále byl proveden vlastní návrh a realizace analyzátoru bezdrátového komunikačního spoje pro dříve specifikované požadované parametry, ověření jeho funkčnosti na přenosovém spoji o známých parametrech a zautomatizování měření a vyhodnocování dat.

Klíčová slova

bezdrátový přenosový kanál, měřicí systém, bezdrátová komunikace, přenosová rychlost, kapacita kanálu, jitter, zpoždění, ztráta packetů

Abstract

The aim of this bachelor thesis is to design and implement a wireless communication analyzer in mobile robotics, using IPv4 protocol and Ethernet interface and to create correctly selected and expressed characteristics from the measured data, according to which the wireless communication channel can be further optimized for the best data transmission. To meet these requirements, a brief search of existing solutions was performed, their applicability was discussed and important parameters of the wireless transmission channel were defined, which will form the given characteristics. Furthermore, the actual design and implementation of a wireless communication link analyzer for previously specified required parameters, verification of its functionality on the transmission link with known parameters and automation of measurement and data evaluation was performed.

Keywords

wireless link, measuring system, wireless communication, bit rate, channel capacity, jitter, latency, packet loss

Bibliografická citace

DOSTÁLEK, Jakub. *Analyzátor bezdrátového komunikačního spoje*. Brno, 2022. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/142700>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Tomáš Jílek.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	Jakub Dostálek
VUT ID studenta:	221292
Typ práce:	Bakalářská práce
Akademický rok:	2021/22
Téma závěrečné práce:	Analyzátor bezdrátového komunikačního spoje

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné 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 závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 23. května 2022

podpis autora

Obsah

SEZNAM OBRÁZKŮ	8
SEZNAM TABULEK.....	9
ÚVOD	10
1. REŠERŠE JIŽ EXISTUJÍCÍCH ŘEŠENÍ A NÁSTROJŮ.....	11
1.1 SOLARWINDS NETWORK BANDWIDTH ANALYZER PACK [1]	11
1.2 IPERF [29]	13
1.2.1 <i>iPerf2</i> [30]	14
1.2.2 <i>iPerf3</i> [32]	16
1.3 NETCPS [2].....	19
1.4 NETPERF [3]	20
1.5 RIPE NCC TEST TRAFIC MEASUREMENT SERVICE (TTM) [4].....	21
1.6 UDPMON [5]	22
1.7 PERFSOAR [33]	24
1.8 SOUHRN VÝSLEDKŮ REŠERŠE	26
2. VÝZNAMNÉ PARAMETRY A CHARAKTERISTIKY PŘENOSOVÉ TRASY A JEJICH VHODNÉ GRAFICKÉ VYJÁDŘENÍ V RŮZNÝCH ZÁVISLOSTECH.....	27
2.1 ŠÍRKA PÁSMA (BANDWIDTH, GROSS BIT RATE)	27
2.1.1 <i>Kapacita kanálu (Channel Capacity, Net Bit Rate)</i>	28
2.1.2 <i>Maximální přenosová rychlost (Available Bandwidth, Throughput)</i>	28
2.1.3 <i>Dosažitelná přenosová rychlost (Achievable Bandwidth, Goodput)</i>	28
2.2 ZPOŽDĚNÍ (LATENCY)	29
2.2.1 <i>Obousměrné zpoždění (Round-trip Time – RTT, Round-trip Delay – RTD)</i>	29
2.2.2 <i>Jednosměrné zpoždění (One-way Delay, End-to-end Delay)</i>	29
2.2.3 <i>Jitter (IP Packet Delay Variation – IPDV)</i>	30
2.3 ZTRÁTA PACKETŮ (PACKET LOSS)	30
2.3.1 <i>Jednosměrná ztráta [20]</i>	31
2.4 PŘESKUPENÍ PACKETŮ (PACKET RE-ORDERING)	31
2.5 VHODNÉ GRAFICKÉ VYJÁDŘENÍ VÝZNAMNÝCH CHARAKTERISTIK PŘENOSOVÉ TRASY V RŮZNÝCH ZÁVISLOSTECH.....	32
3. POŽADOVANÉ PARAMETRY ANALYZÁTORU PRO UVAŽOVANOU APLIKACI	33
3.1 HARDWARE.....	33
3.2 SOFTWARE	34
4. NÁVRH ANALYZÁTORU PRO POŽADOVANÉ PARAMETRY.....	35
4.1 SOFTWARE	35
4.1.1 <i>Měřicí software a operační systém</i>	35
4.1.2 <i>Časová synchronizace</i>	35
NTP [36].....	35
PTP [37].....	36
4.1.3 <i>Dálkový přístup [39]</i>	37

4.2	HARDWARE.....	38
4.2.1	<i>Klient měřicího systému</i>	38
4.2.2	<i>Server měřicího systému</i>	39
5.	REALIZACE ANALYZÁTORU A OVĚŘENÍ JEHO VLASTNOSTÍ A SPRÁVNÉ FUNKCE	
	41	
5.1	KLIENT MĚŘICÍHO SYTÉMU	41
5.2	SERVER MĚŘICÍHO SYSTÉMU	42
5.3	OVĚŘENÍ FUNKČNOSTI MĚŘICÍHO SYSTÉMU	44
6.	IMPLEMENTACE AUTOMATIZOVANÉHO ZPRACOVÁNÍ A ZOBRAZENÍ	
	NAMĚŘENÝCH DAT	49
7.	MĚŘENÍ NEZNÁMÉHO BEZDRÁTOVÉHO SPOJE A DOPORUČENÍ VHODNÉHO	
	NASTAVENÍ.....	50
7.1	MĚŘENÍ NA ROUTERECH MIKROTIK.....	50
7.1.1	<i>Plán měření</i>	50
7.1.2	<i>Průběh měření</i>	52
7.1.3	<i>Výsledky měření</i>	53
7.2	MĚŘENÍ NA RADIOMODEMECH	57
7.2.1	<i>Plán měření</i>	57
7.2.2	<i>Průběh měření</i>	58
7.2.3	<i>Výsledky měření</i>	59
8.	ZÁVĚR.....	63
	LITERATURA.....	64
	SEZNAM ZKRATEK	68

SEZNAM OBRÁZKŮ

1.1	Příklad výstupu aplikace NetFlow [20].....	13
1.2	Příklad výstupu měřicího softwaru iPerf3 [21]	19
1.3	Příklad výstupu měřicího softwaru NetCPS [22]	19
1.4	Příklad výstupu měřicího softwaru Netperf [23].....	21
1.5	Příklad výstupu měřicího softwaru UDPMon [19].....	24
1.6	Architektura programu perfSONAR [34].....	26
4.1	Výstup softwaru pro časovou synchronizaci ptp4l.....	37
4.2	Klient měřicího systému (notebook) připojený k řídicí stanici	39
4.3	Server měřicího systému (Raspberry Pi) připojený k přijímači mobilního robota	40
5.1	Graf závislosti přenosové rychlosti při příjmu dat na přenosové rychlosti při odesílání dat	46
5.2	Graf závislosti jitteru na přenosové rychlosti při odesílání dat	46
5.3	Graf závislosti ztráty packetů na přenosové rychlosti při odesílání dat.....	47
5.4	Graf závislosti jednocestného zpoždění na přenosové rychlosti při odesílání dat.....	47
5.5	Schéma zapojení pro ověření funkčnosti měřicího systému.....	48
7.1	Mapa oblasti, ve které probíhalo měření	50
7.2	Schéma zapojení pro měření na routerech MikroTik	53
7.3	Graf porovnání výsledků měření (klient -> server).....	54
7.4	Graf porovnání výsledků měření (server -> klient).....	54
7.5	Graf porovnání výsledků měření (klient -> server).....	55
7.6	Graf porovnání výsledků měření (server -> klient).....	55
7.7	Graf porovnání výsledků měření (klient -> server).....	59
7.8	Graf porovnání výsledků měření (server -> klient).....	59
7.9	Graf porovnání výsledků měření (klient -> server).....	60
7.10	Graf porovnání výsledků měření (server -> klient).....	60

SEZNAM TABULEK

1.1	Srovnání měřicích softwarů iPerf2 a iPerf3 [31].....	17
7.1	Plán měření pro různé vzdálenosti přijímače mobilního robota MikroTik od řídicí stanice a konfigurace routeru MikroTik.....	51
7.2	Maximální dosažitelná přenosová rychlost v Mb/s při příjmu dat bez ztráty packetů v různých vzdálenostech přijímače mobilního robota od řídicí stanice a vysílacích výkonech (klient -> server).....	56
7.3	Maximální dosažitelná přenosová rychlost při příjmu dat bez ztráty packetů v různých vzdálenostech přijímače mobilního robota od řídicí stanice a vysílacích výkonech (server -> klient).....	56
7.4	Plán měření pro různé vzdálenosti přijímače mobilního robota od řídicí stanice a konfigurace radiomodemů.....	58

ÚVOD

Komunikace je v mobilní robotice jedna z nejdůležitějších oblastí, protože zpravidla probíhá přes větší vzdálenosti a za pohybu robota, a proto není kabelové spojení využíváno. Kabel by se totiž mohl zachytávat za různé objekty při pohybu robota a mohl by se poškodit vlivy prostředí nebo počasí. Proto se využívá bezdrátové komunikace.

Kvalita bezdrátové komunikace je mnohem náchylnější na rušení a více technologicky omezená než komunikace drátová, a proto bude velmi důležité přesné a kvalitní měření parametrů bezdrátového přenosového kanálu v závislosti na různých pracovních podmínkách, zvolení důležitých charakteristik a zobrazování těchto charakteristik z naměřených dat, aby bylo poté možné upravit konfiguraci této komunikace a tím ji optimalizovat pro dané prostředí a pracovní podmínky.

Důležité bude vhodné zvolení měřicího systému, který využívá rozhraní Ethernet, podporuje protokol IPv4, pracuje na principu klient – server a bude ho ideálně možno dostat na nějaký vestavěný systém, který bude moci být v případě potřeby upevněn alespoň na mobilním robotu, který bude fungovat jako odpovídací server a bude se k němu moci vzdáleně připojit a ovládat ho (u klienta nebude vyžadováno využití vestavného systému, protože ten se pohybovat nebude). Tím pádem bude muset být možno měření pomocí měřicího softwaru zautomatizovat a dostat z něj nezpracovaná data, nejlépe ve formě textového souboru, které se pak budou dít zpracovávat do různých charakteristik např. pomocí programu MATLAB.

Bude zapotřebí pečlivě vyhledat už existující řešení a rozhodnout u nich o jejich vhodnosti použití pro tuto bakalářskou práci. V případě, že žádné již existující řešení nebude vhodné, bude nevyhnutelné naprogramování řešení vlastního. Dále se bude muset řešit problematika časové synchronizace obou konců měřicího systému, pro správné měření jednosměrného zpoždění a specifikovat požadované parametry analyzátoru pro uvažovanou aplikaci. Po jeho návrhu a realizaci se bude muset ověřit jeho funkčnost na přenosovém spoji o známých parametrech.

1. REŠERŠE JIŽ EXISTUJÍCÍCH ŘEŠENÍ A NÁSTROJŮ

Měření parametrů bezdrátového přenosového kanálu je v dnešním čím dál tím víc digitalizovaném světě, kde se využití bezdrátové, ale i celkové elektronické komunikace zvyšuje každým dnem, velmi využívaná věc, pro kterou existují stovky softwarových řešení.

S některými může dojít k setkání skoro každý den, ať už je to jednoduchý prohlížečový měřič rychlosti internetu (<https://www.speedtest.net/> od společnosti Ookla, Test rychlosti internetu ve vyhledávači Google, <https://rychlost.cz/speedtest/> a další stovky stránek jim podobným) pokud je zapotřebí zjistit hodnotu rychlosti nahrávání, stahování anebo odezvy, jestli poskytovatel internetu dodržuje smluvní hodnoty nebo jestli není v síti někde problém, až po sofistikované měřicí softwary, které dokážou změřit, krom výše uvedené hodnoty rychlosti nahrávání, stahování a odezvy, ještě velký počet dalších, pro tuto práci také důležitých parametrů, jejichž způsob měření se dá podrobně konfigurovat a zautomatizovat. Tyto parametry budou podrobně diskutovány v kapitole dva, protože pro účely mobilní robotiky, kde roboty většinou využívají uzavřené LAN sítě bez připojení k internetu, nebude něco, jako je prohlížečový test rychlosti internetu, stačit.

Níže je uvedena stručná rešerše softwarových řešení, které jsou uvedené buď pro úplnost anebo splňují následující kritéria této bakalářské práce:

- měření parametrů bezdrátového přenosového kanálu důležitých pro mobilní robotiku,
- komunikace přes síťovou technologii Ethernet,
- univerzalita a použití komunikačních modulů umožňující přenášet síťové datagramy protokolu IPv4.

1.1 SolarWinds Network Bandwidth Analyzer Pack [1]

Tento a vlastně jakékoliv produkty od společnosti SolarWinds jsou zaměřeny na velmi rozlehlé sítě, které vyžadují správu a dohled. Pro účely objektivní rešerše je v rešerši zahrnut i tento typ měřicího softwaru, který sice řeší měření parametrů i bezdrátového přenosového kanálu, ale svým objemem a zaměřením nebyl určen pro jednoduchá měření z bodu do bodu, kterých bude využito v této bakalářské práci.

Všechny produkty jsou na oficiálních webových stránkách firmy SolarWinds (<https://www.solarwinds.com/>) detailně popsány a vysvětlené i např. formou tří až čtyřminutového videa, které krásně a stručně shrne a ukáže, co vlastně daný program umí a dokáže. Software, který tato firma nabízí už vlastně není ani tak měřicí, ale spíše monitorovací a využívá svých měření pro zlepšení a optimalizaci sítě.

Většina služeb a balíčků, které si může uživatel stáhnout a nainstalovat jsou dostupné v rámci předplatného, jehož cena začíná na jednotkách eur (složitější služby stojí i

několik stovek eur) za jednotlivou službu, ale asi jako u všech firem s úspěšným marketingem je u nich možnost třicetidenní zkušební doby, kdy budou všechny funkce zpřístupněny a zákazník je může vyzkoušet a rozhodnout se o jejich zakoupení.

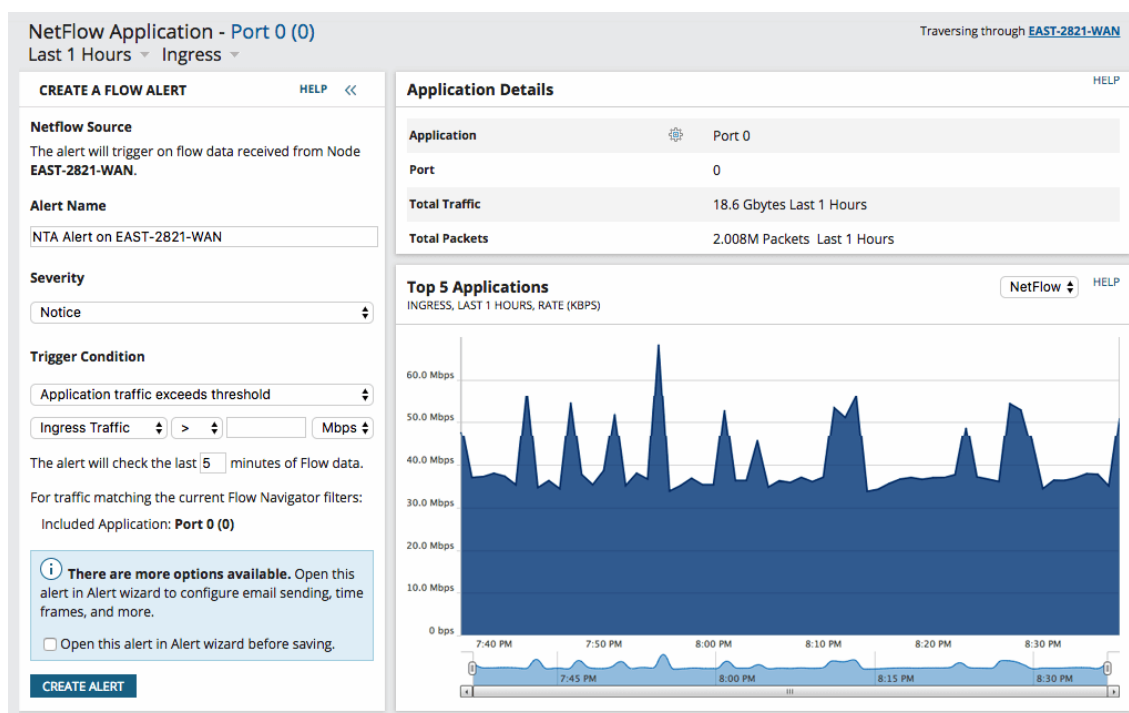
Krom těchto hrazených balíčků nabízí tato společnost i několik nástrojů zdarma a to:

- kontrolování přidělených povolení pro uživatele sítě,
- tvorbu a monitorování SQL (Structured Query Language) systémů,
- monitorování dění na síti,
- měření a úprava pohybu po síti,
- analýzu síťových cest,
- řízení vzdálených ploch a kontrolování logů.

Placené produkty pak řeší ještě širší objem problémů jako:

- monitorování výkonu sítě,
- správa konfigurace sítě,
- správa IP adres,
- sledování uživatelských zařízení spravování kvality sítě a internetových hovorů,
- správa automatizace sítě,
- analýzu kapacity kanálu (balíčků),
- kontrolu IP adres (balíčků),
- mapování topologie sítě a spoustu dalších nástrojů pro síťové inženýry.

Z toho jde usoudit, že tyto produkty by sice dokázaly splnit požadavky na měření parametrů bezdrátového přenosového kanálu i v mobilní robotice, ale protože jsou dimenzovány pro obrovské firmy a jejich sítě (a bezdrátová komunikace v mobilní robotice anebo alespoň při ověřování funkčnosti navrženého analyzátoru, bude většinou probíhat z bodu A do bodu B bez nějakých dlouhých přeskoků v malé síti LAN) a měření parametrů je jen zlomek toho, co tento program dokáže, je pro účely této bakalářské práce nevhodný, už jen díky tomu, že jak se dozvíme níže, na světě existují malé, více zaměřené, jednodušší open-source měřicí softwary bez grafického rozhraní, které místo už zpracovaných dat dají data nezpracovaná, ze kterých je pak možno vytvořit charakteristiky a zpracovávat data přesně podle potřeb a požadavků a navíc můžou být zdarma.



Obrázek 1.1 Příklad výstupu aplikace NetFlow [20]

1.2 iPerf [29]

iPerf byl multiplatformní nástroj pro aktivní měření a ladění výkonu IP sítí. Dokázal produkovat standardizovaná měření pro jakoukoli síť, což svědčilo o jeho univerzálnosti. Toto pak vedlo k jeho velké popularitě mezi podobnými nástroji. Využíval principu server – klient a dokázal vytvářet datové proudy pro měření dosažitelné přenosové rychlosti mezi dvěma konci v jednom nebo obou směrech. Dokázal vyprodukovat zprávu o množství přenesených dat, přenosové rychlosti, zpoždění, ztrátě a přeskupení paketů a dalších parametřů v předem definovaných časových intervalech.

Podporoval ladění různých parametřů souvisejících s časováním, protokoly a vyrovnávací paměti. Datové proudy využívaly buď UDP nebo TCP protokol a bylo umožněno specifikovat velikost datagramu. Byl to open-source měřicí software (což umožňovalo uživateli podrobně prozkoumat metodiku měření) napsaný v jazyce C, běžel na různých platformách včetně Linuxu, Unixu a Windows.

Byla to kompatibilní reimplementace programu ttcp (Test TCP – starší obdoba iPerf pro Linux, existuje mnoho derivací tohoto programu jako nttcp, nuttcp a nttcp, ale všechny jsou si velmi podobné, a proto je tu uveden jen iPerf), který byl vyvinut v Národním centru pro superpočítačové aplikace na Illinoiské univerzitě týmem DAST (Distributed Applications Support Team) NLANR (National Laboratory for Applied Network Research), jenž byl rozpuštěn 31. prosince 2006. V dnešní době existují dvě

verze tohoto měřicího softwaru, a to iPerf2 a iPerf3, které jsou v současné době aktivně vyvíjeny nezávisle na sobě.

1.2.1 iPerf2 [30]

Pro tento měřicí software (<https://sourceforge.net/projects/iperf2/>), který je od roku 2014 vyvíjen Robertem Josephem McMahonem a skupinou dalších lidí, platí vše, co bylo řečeno o originálním programu iPerf, který napsali Mark Gates a Alex Warshavsky. Je to vlastně jeho pokračování. Dále se jedná o vícevláknový design, který se přizpůsobuje počtu CPU nebo jader v systému. Podporuje jednosměrný, plně duplexní (stejný socket) a obousměrný provoz a podporuje vícenásobné simultánní toky provozu vícesměrový provoz včetně spojení SSM (source specific multicast).

Mezi jeho další funkce patří možnost:

- nastavení požadované hodnoty přenosové rychlosti (odesílání i příjmu), odesílaných packetů za sekundu nebo počtu bytů k přenosu,
- zobrazení rozšířeného výstupu měření programu,
- nastavení formátu (násobku) zobrazovaných jednotek přenosových rychlostí nebo objemu dat,
- skrytí IP adresu ve výstupu měření programu,
- provádět kontroly délky linkové (druhé) vrstvy ISO/OSI u přijatých packetů UDP (vyžaduje systémy, které podporují sockety packetů, např. Linux),
- nastavení velikost sdílené paměti mezi přenosovým vláknem (vlákny) a reportérovým vláknem (aby nedocházelo ke sporům o mutex lock),
- uložit výstup měření nebo chybové zprávy do zadaného souboru,
- nastavení (nebo náhodná generace) klíče a jeho životnosti (výchozí je 20 s), pro připojení k serveru (pouze pro TCP),
- nastavení klientských nebo serverových portů pro odesílání nebo příjem (výchozí 5001) s volitelným rozsahem,
- sčítání vláken provozu na základě IP adresy (výchozí je zdrojová IP adresa),
- nastavení výstupu pouze na součet sestav,
- nastavení času v sekundách pro naslouchání novým spojení (výchozí je nekonečno) a pro přijímání nebo odesílání dat (výchozí je 10 s),
- nastavení velikosti TCP okna (vyrovnávací paměť socketu),
- vyžádání plánovače v reálném čase (pokud je podporován),
- vazby na hosta, IP nebo multicastovou adresu, volitelný port nebo zařízení,
- nezasiílání extra zpráv pro použití se staršími verzemi,
- nastavení maximální velikosti segmentu TCP (pomocí TCP_MAXSEG),
- nastavení TCP bez prodlevy (deaktivování Nagelova algoritmu),
- vytisknutí informací o verzi,

- vyloučit z výstupu informace o spojení, datech, multicastu, nastavení a serveru,
- výstup programu v CSV,
- nastavení výchozího algoritmu řízení zahlcení, který se má použít pro nová připojení (platformy musí podporovat setsockopt TCP_CONGESTION),
- nastavení serveru, pro zpracování pouze jednoho klienta,
- výpisu histogramů zpoždění pro UDP, zápisů pro TCP anebo pro oboje s nastavitelnou šířkou intervalů (výchozí 1 ms), celkového počtu intervalů (výchozí je 1000) a intervalu spolehlivosti mezi 0 – 100 % (výchozí dolní je 5 %, horní 95 % a 3σ 99,7 %),
- nastavení přijímacího rozhraní monitorovacího zařízení TAP,
- nastavení TCP_WINDOW_CLAMP na socketu v bytech,
- nastavení pole IP_TOS (byte) socketu pro reverzní nebo plně duplexní provoz,
- spuštění serveru v režimu daemon,
- odstranění služby,
- povolit příjem IPv6 nastavením domény a socketu na AF_INET6,
- provádění TCP bounceback testů (výchozí velikost je 100 B) souběžně plně duplexně s volitelným zpožděním v milisekundách mezi čtením a zápisem (výchozí je bez zpoždění), naplánování intervalu odesílání v milisekundách (výchozí je bez prodlevy, bez argumentu je interval 1 s) a bez vypnutí zpoždění TCP_QUICKACK,
- nastavení intervalu burstů v sekundách (výchozí je jedna sekunda) a jejich velikosti v bytech (výchozí je 1 MB),
- provádět testy TCP připojení (bez datového přenosu) a nastavovat počet pokusů,
- nastavení rychlosti, která se má použít při řazení do front na úrovni socketu využívající fair-queueing, v bajtech nebo bitech za sekundu na platformách podporujících SO_MAX_PACING_RATE,
- inkrementovat IP adresu nebo port zdroje nebo cíle,
- nastavit mezeru mezi packety v sekundách pro packety, bursty a framy,
- odeslat izochronní provoz ve framech za sekundu a obsahem definovaným střední hodnotou a směrodatnou odchylkou v packetech nebo bitech za sekundu pomocí logaritmického normálního rozdělení (výchozí je 60:20 m,0),
- omezit provoz pouze na místní síť (pomocí SO_DONTROUTE),
- omezit rychlost zápisu TCP podle vzorkovaného obousměrného zpoždění,
- zakázat synchronizaci vláken TCP,
- nevypisovat závěrečnou zprávu serveru,

- měření dočasné doby odesílání klienta,
- nastavení TCP_NOTSENT_LOWAT na socketu a použití zápisů založených na událostech na select(),
- měření zpoždění mezi zápisem a čtením,
- nastavení času v sekundách pro zpoždění po připojení TCP a před zápisem socketu, u UDP je to zpoždění mezi spuštěním přenosového vlákna a prvním zápisem,
- nastavení času a data startu Tx packetu,
- nastavení přenosu dat ze souboru nebo příkazového řádku,
- nastavení počtu paralelních klientských vláken ke spuštění,
- nastavení TTL (Time to live),
- detektovat verzi softwaru druhého konce před odesláním dat.

Tento výčet funkcí tohoto měřicího softwaru jen podtrhuje jeho obrovské možnosti při využití pro analýzu bezdrátového přenosového spoje. Asi jeho největší výhodou, která všem jeho konkurentům chybí je měření jednosměrného zpoždění. Manuál pro tento program je obsáhlý, přehledný a jsou v něm uvedeny jednotlivé příklady pro různé typy testů, které jsou dopodrobna vysvětleny. Na YouTube kanálu Roberta Josepha McMahona (<https://www.youtube.com/channel/UCaqlH3a442xaZ9humrxRHGQ>) jsou také nahrány demonstrační videa tohoto měřicího softwaru, ve kterých i poskytuje krátký komentář ohledně funkcí programu.

Stažení tohoto programu je zdarma a jeho instalace je jednoduchá. Provádí se buď přes konzoli (např. příkazem `apt-get install`) anebo manuálně ze staženého balíčku. Díky všem těmto výše uvedeným skutečnostem bylo rozhodnuto k realizaci této bakalářské práce použít zrovna tento měřicí software.

1.2.2 iPerf3 [32]

Tento měřicí software (<https://software.es.net/iperf/index.html#>) je redesignem původní verze iPerf potom, co se ESnet v roce 2010 rozhodl, že je potřeba nový, jednodušší nástroj. Je to nová implementace od nuly, s cílem menšího a jednoduššího základu kódu a verze knihovny funkcí, kterou lze použít v jiných programech. Obsahuje také řadu funkcí nalezených v jiných nástrojích, jako je nuttcp a Netperf, ale v původním iPerf chyběly. Patří mezi ně například zero-copy mód a volitelný výstup v JSON. Toto ale znamená, že iPerf3 není zpětně kompatibilní s původním iPerf a tím pádem iPerf2, i když obsahuje část jeho původního kódu.

Jeho primární vývoj probíhá na operačních systémech CentOS Linux, FreeBSD a macOS. V tuto chvíli jsou to jediné oficiálně podporované platformy, nicméně se objevily zprávy o úspěchu s operačními systémy NetBSD, OpenBSD, Windows, Solaris, Android, iOS a dalšími distribucemi Linuxu. Je primárně vyvinut společností ESnet / Lawrence Berkeley National Laboratory a je vydán pod licencí BSD se třemi klauzulemi.

Pro lepší pochopení rozdílů mezi měřicím softwarem iPerf2 a iPerf3 jsou dané rozdíly uvedené v následující tabulce:

Tabulka 1.1 Srovnání měřicích softwarů iPerf2 a iPerf3 [31]

Vlastnost	iPerf2	iPerf3
Typ generovaných dat		
TCP	Ano	Ano
UDP	Ano	Ano
SCTP	Ne	Ano
IPv4	Ano	Ano
IPv6	Ano	Ano
Multicast	Ano	Ne
Pouze připojení TCP	Ano	Ne
Kontroly linkové vrstvy	Ano	Ne
Výstup		
Formát pro člověka	Ano	Ano
JSON	Ne	Ano
CSV	Ano	Ne
Skrytí IPv4 adresy	Ano	Ne
Hlášení serveru na straně klienta	Ne	Ano
Profily vygenerovaných dat		
Omezení míry fair queue	Ano	Ano
Omezení rychlosti zápisu	Ano	Ano
Omezení rychlosti čtení (TCP)	Ano	Ne
Bursty	Ano	Ano
Izochronní TCP/UDP	Ano	Ne
Reverzní provoz	Ano	Ano
Obousměrný provoz	Ano	Ano
Plně duplexní provoz	Ano	Ne
TCP bounceback	Ano	Ne
Provoz s nízkým pracovním cyklem se serverovými statistikami	Ano	Ne
TCP_NOTSENT_LOWAT se select()	Ano	Ne
TCP algoritmus řízení přetížení	Ano	Ne
Měření dočasné doby odesílání klienta	Ano	Ne
Podpora Markovova řetězce	Ne	Ne
Metriky		
Přenosová rychlost	Ano	Ano
UDP packety	Ano	Ano
UDP jitter	Ano	Ano

Zpoždění UDP packetů	Ano	Ne
Zpoždění frameů/burstů TCP/UDP	Ano	Ne
Zpoždění mezi zápisem a čtením TCP	Ano	Ne
Síťový výkon	Ano	Ne
Byty ve frontách	Ano	Ne
TCP CWND (Congestion Window)	Ano	Ne
TCP opakování	Ano	Ano
Obousměrné zpoždění TCP	Ano	Ano
Select() histogramy	Ano	Ne
UDP packety za sekundu	Ano	Ne
Histogramy zpoždění	Ano	Ne
Doby připojení TCP	Ano	Ne
Odezva TCP za interval	Ano	Ne
Výstup pouze součtu sestav	Ano	Ne
Jiné		
Vícevláknové provedení	Ano	Ne
Paralelní technika	Vlákna	Procesy
Plánování v reálném čase	Ano	Ne
Nastavení doby funkce serveru	Ano	Ne
Podpora virtuálního rozhraní TAP (pouze příjem)	Ano	Ne
Afinita CPU	Ne	Ano
Zero copy	Ne	Ano
IPv6 Flow label	Ne	Ano
Přeskok prvních vzorků za určitý čas v sekundách	Ne	Ano
Inkrementace IP adresy nebo portu zdroje nebo cíle	Ano	Ne
Vazba na zařízení nebo rozhraní	Ano	Ano
Vazba na zdrojový port	Ano	Ne
Naplánování času zahájení vysílání	Ano	Ne
Odložení času zahájení vysílání	Ano	Ne
Uživatelské heslo	Ne	Ano
Přístupovací klíče TCP	Ano	Ne
Bezstavové UDP	Ano	Ne
Python framework (asyncio flows)	Ano	Ne
Testování 100G Wi-Fi	Ano	-
Škálování na 1000+ vláken	Ano	-

Tento software bylo na začátku v plánu použít pro měření parametrů bezdrátového kanálu, protože víceméně splňuje hrubé požadavky na tyto měření, ale po zhodnocení jednotlivých funkcí, bylo rozhodnuto proti tomu. Hlavní důvod, díky kterému bylo takto rozhodnuto, byla absence funkce měření jednocestného zpoždění. Pro dosažení

požadavků by se musel tento program kombinovat s dalšími nástroji dostupnými pro operační systém Linux, což není žádoucí, a tím by odpadla jednoduchost používání a jakékoliv měření by se zkomplikovalo. Musí se vyzdvihnout možnost stažení zdarma.

```

Command Prompt
E:\PC-USR-DIR\Desktop\iperf-3.1.3-win64>iperf3 -c 172.16.0.20 -R
Connecting to host 172.16.0.20, port 5201
Reverse mode, remote host 172.16.0.20 is sending
[ 4] local 172.16.0.5 port 53763 connected to 172.16.0.20 port 5201
[ ID] Interval           Transfer             Bandwidth
[ 4]  0.00-1.00    sec   60.7 MBytes       509 Mbits/sec
[ 4]  1.00-2.00    sec   62.6 MBytes       526 Mbits/sec
[ 4]  2.00-3.00    sec   45.4 MBytes       381 Mbits/sec
[ 4]  3.00-4.00    sec   33.1 MBytes       277 Mbits/sec
[ 4]  4.00-5.00    sec   60.9 MBytes       511 Mbits/sec
[ 4]  5.00-6.00    sec   33.4 MBytes       280 Mbits/sec
[ 4]  6.00-7.00    sec   58.6 MBytes       492 Mbits/sec
[ 4]  7.00-8.00    sec   26.6 MBytes       223 Mbits/sec
[ 4]  8.00-9.00    sec   48.0 MBytes       402 Mbits/sec
[ 4]  9.00-10.00   sec   57.3 MBytes       481 Mbits/sec

-----
[ ID] Interval           Transfer             Bandwidth             Retr
[ 4]  0.00-10.00    sec   487 MBytes       408 Mbits/sec         0
[ 4]  0.00-10.00    sec   487 MBytes       408 Mbits/sec         0

iperf Done.
E:\PC-USR-DIR\Desktop\iperf-3.1.3-win64>

```

Obrázek 1.2 Příklad výstupu měřicího softwaru iPerf3 [21]

1.3 NetCPS [2]

Tento měřicí software (<https://www.netchain.com/netcps/>) byl navržen pro opravu chyb se switchy a používá se pro měření efektivního výkonu sítě využívající protokoly TCP/IP. Funguje na principu klient – server a při spuštění pošle komunikačním kanálem 100 Mb. Čas pro měření přenosové rychlosti je měřen s přesností na tisícinu sekundy. Výhodou je, že je to freeware a zdrojový kód je volně dostupný. Je to malý a velmi jednoduchý program z roku 1998, který nemá kromě měření přenosové rychlosti žádné jiné funkce, a proto nedokáže plně pokrýt požadavky této bakalářské práce.

```

C:\WINNT\System32\cmd.exe - netcps -s
C:\TingApps.Org>netcps -s
NetCPS 1.0 - Entering server mode. Press ^C to quit
Waiting for new connection...
Client connected from 192.168.1.100
--> CPS 1266688.00 KPS: 1237.00 MPS: 1.21
Avg CPS 1190292.25 KPS: 1162.39 MPS: 1.14
Peak CPS 1288192.00 KPS: 1258.00 MPS: 1.23
Client disconnected. 104857600 Kb transferred in 88.09 seconds.

```

Obrázek 1.3 Příklad výstupu měřicího softwaru NetCPS [22]

1.4 Netperf [3]

Tento měřicí software z roku 2015 psaný v programovacím jazyce C byl původně vyvinutý společností HP (<https://hewlettpackard.github.io/netperf/>). Používá se hlavně k testování přenosové rychlosti mezi dvěma body v síti a podporuje unixové doménové sockety a protokoly TCP, SCTP, DLPI (Data Link Provider Interface) a UDP.

Alespoň některá verze tohoto programu funguje na operačních systémech Windows a Unix (všechny hlavní varianty). Můžou to být i ostatní známé operační systémy. Tento měřicí software je open-source, tudíž je jeho stažení a instalace je zdarma.

Využívá principu server – klient a při provedení testu se komunikace bohužel odpojí a jsou zobrazeny výsledky. Pro opakované testy je potřeba spojení znovu navodit.

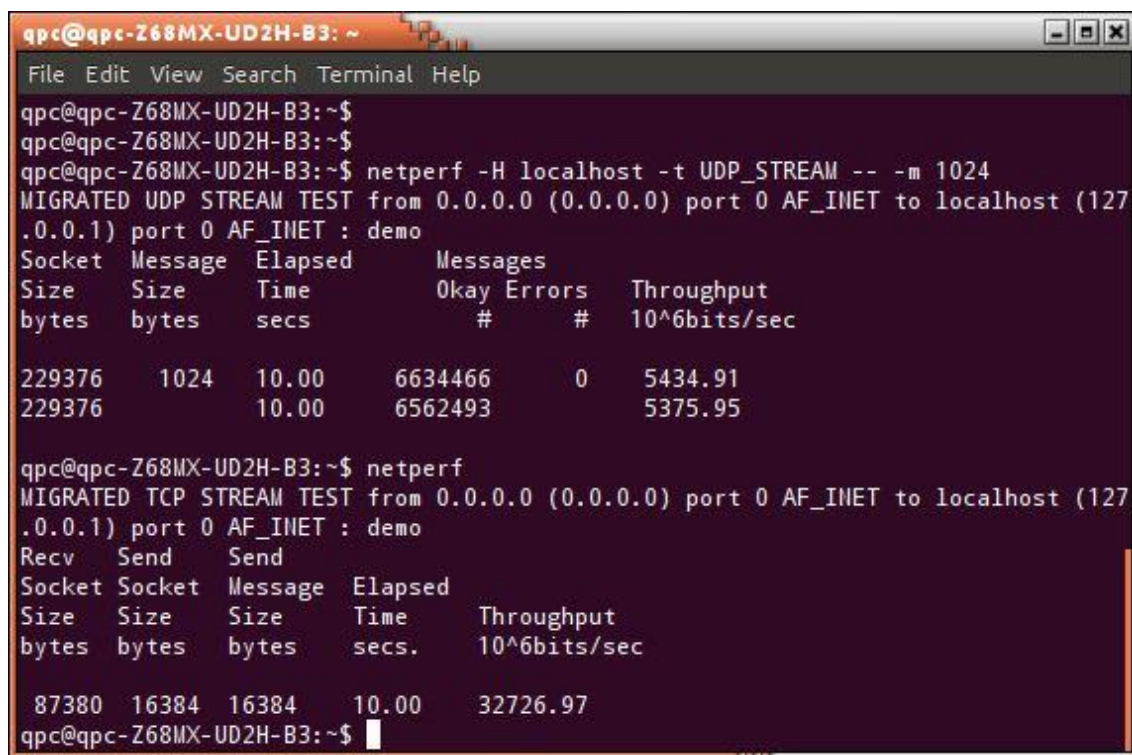
Funkce tohoto programu zahrnují:

- možnost změny velikostí vyrovnávacích pamětí používaných při příjmu a odesílání dat,
- nastavení intervalů mezi odesíláním jednotlivých packetů,
- vyhodnocení využití procesoru pro měření,
- zobrazení dat pro debugování,
- automatický převod výsledků měření na předem daný násobek,
- možnost výběru protokolu IPv4 nebo IPv6,
- výpočty spolehlivosti komunikace a opakování měření, dokud se nedosáhne požadované spolehlivosti,
- měření zpoždění,
- pevné stanovení délky jedné iterace testu,
- pevné stanovení hodnoty využití kapacity CPU systému, na kterém běží (program toto dělá automaticky),
- provádění omezených měření bez potřeby spouštět odpovídací server na vzdáleném systému,
- nastavení offsetu pro vyrovnávací paměti,
- manuální nastavení portu pro komunikační kanál, na kterém se provádí měření,
- nastavení časové prodlevy před prováděním testu,
- přiřazení instance programu k CPU,
- nastavení množství vytisklých dat o testu,
- nastavení počtu vyrovnávacích pamětí v kruhu (Netperf nepoužívá pro nepřetržité odesílání a příjem dat jen jednu vyrovnávací paměť, ale několik, které kruhově střídá).

Kromě testů přenosové rychlosti sítě jsou dále dostupné testy pro měření zpoždění a každý test automaticky měří i ztrátu packetů. Tyto testy mají spoustu nastavení skrze argumenty, které např. umožňují měřit nejen zpoždění komunikace a přenosovou rychlost jen při jednom datovém toku, ale také třeba jak rychle se dokáže navázat

spojení mezi dvěma body v síti a agregovanou přenosovou rychlost, kdy neprobíhá přenos jen jedním datovým tokem [11].

Netperf má mnoho specializovaných využití, které člověku se zkušenostmi a znalostmi ohledně sítí umožní provádět testy zaměřené na přesný problém. Spolu s rozsáhlou a extrémně bohatou dokumentací, kde je detailně popsán celý program a jeho funkce, je to velmi užitečný nástroj, který si svým způsobem zachovává svoji jednoduchost. Při praktickém použití se ale iPerf2 ukázal jako mnohem sofistikovanější a vyspělejší měřicí software. Jeho používání je intuitivnější, má více funkcí a jeho podpora a údržba je častější a aktuálnější, a právě proto byl iPerf2 zvolen místo tohoto programu.



```
qpc@qpc-Z68MX-UD2H-B3: ~  
File Edit View Search Terminal Help  
qpc@qpc-Z68MX-UD2H-B3:~$  
qpc@qpc-Z68MX-UD2H-B3:~$  
qpc@qpc-Z68MX-UD2H-B3:~$ netperf -H localhost -t UDP_STREAM -- -m 1024  
MIGRATED UDP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to localhost (127  
.0.0.1) port 0 AF_INET : demo  
Socket Message Elapsed Messages  
Size Size Time Okay Errors Throughput  
bytes bytes secs # # 10^6bits/sec  
229376 1024 10.00 6634466 0 5434.91  
229376 10.00 6562493 5375.95  
  
qpc@qpc-Z68MX-UD2H-B3:~$ netperf  
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to localhost (127  
.0.0.1) port 0 AF_INET : demo  
Recv Send Send  
Socket Socket Message Elapsed  
Size Size Size Time Throughput  
bytes bytes bytes secs. 10^6bits/sec  
87380 16384 16384 10.00 32726.97  
qpc@qpc-Z68MX-UD2H-B3:~$
```

Obrázek 1.4 Příklad výstupu měřicího softwaru Netperf [23]

1.5 RIPE NCC Test Traffic Measurement Service (TTM) [4]

TTM je služba, nejen software, která měří klíčové parametry komunikačního kanálu mezi dvěma body v síti. Byla vyvinuta v roce 2010 mezinárodní nezávislou neziskovou společností RIPE NCC (<https://www.ripe.net/analyse/archived-projects/ttm>), která funguje jako regionální internetový registr a poskytuje globální internetové zdroje a s nimi spojené služby [26].

Tato služba umožňuje komplexně a nepřetržitě sledovat konektivitu sítě k ostatním částem. Byla navržena pro nepřetržité monitorování a měření parametrů komunikace uživatelské sítě s internetem. To možná nezní jako něco, co by bylo blízko měření

parametrů bezdrátového přenosového kanálu sítě LAN pro mobilní robotiku, ale stojí za zmínku, protože stejný princip by se mohl aplikovat, pokud by byl pro toto použití požadavek na nepřetržité měření a vyhodnocování těchto parametrů.

TTM používá vyhrazená měřicí zařízení (testovací boxy), která generují malé množství provozu, takže významně nezatěžuje vaši síťovou infrastrukturu. TTM také nesleduje žádná data v síti, čímž je zajištěno soukromí. Zařízení pro měření testovacích boxů obsluhují pracovníci RIPE NCC jako černou skříňku. Jsou to vlastně předkonfigurované počítače s anténou.

Měření, která nejsou navržena na principu klient – server, jsou jednosměrná a mohou analyzovat asymetrické efekty. Měření se provádějí na síťové úrovni a jsou nezávislé na aplikacích. Výsledky měření jsou prezentovány na zabezpečeném webu a zahrnují zpoždění, ztrátu packetů, jitter a přenosovou rychlost s možností grafického zpracování a úplné statistické analýzy dat. Další výhodou je i zpracování dlouhodobých trendů všech těchto parametrů a další síťové analýzy, např. pro DNS (Domain Name System) a hlášení neočekávaných změn např. pomocí mailu.

Tato služba byla bohužel ukončena v roce 2014, takže je pro tuto bakalářskou práci irelevantní, ale jako myšlenka a princip měření z dlouhodobého a nepřetržitého hlediska stojí minimálně za zmínku.

1.6 UDPMon [5]

UDPMon je měřicí software vyvinutý v roce 2006 Richardem Hughes-Jonesem na Manchesterské univerzitě (<https://www.hep.manchester.ac.uk/u/rich/net/>) pro operační systém Linux a dá se použít pro měření jak výkonnosti switchů, tak i parametrů komunikačních kanálů v sítích LAN a WAN (Wide Area Network) pomocí packetů UDP.

Tento program umožňuje měření zpoždění, přenosové rychlosti, ztráty packetů, přeskupení packetů a jitteru, spolu se zatížením CPU a vytvářet z těchto dat histogramy. Využívá principu klient – server, a proto je potřeba mít dvě měřicí stanice. Jednu na jednom konci pro provádění měření a druhou na druhém konci, která ji bude odpovídat. [27]

I když je to program vyvinutý pro operační systém Linux, není zahrnutý v „oficiálních“ knihovnách softwaru, který si může člověk rovnou stáhnout a nainstalovat prostřednictvím vyhledávače aplikací nebo terminálu (např. pomocí příkazu `apt-get install`), ale musí být stažen z internetu a manuálně nainstalován. Je to tím, že tento měřicí software už přes deset let neobdržel žádnou aktualizaci (naposledy v roce 2011, i když je na jeho webové stránce zaznamenána aktivita v roce 2021). Tato instalace ale není pro člověka, co má základní zkušenosti s operačním systémem Linux nijak náročná.

UDPMon umožňuje díky argumentům upravit a nastavit vlastnosti měření, které zahrnují:

- manuální volbu portu, pro použití při měření (výchozí je 14233),
- množství vytisknutých informací v konzoli,
- spuštění ve funkci daemon,
- bity IP precedence v packetech odpovědí,
- DSCP (differentiated services code point) QoS (Quality of Service) bity v packetech odpovědí,
- velikost vyrovnávacích pamětí pro odesílání a příjem,
- IP ToS (Type of Service) bity v packetech odpovědí,
- šířku sloupce histogramu,
- koncové hodnoty délky odpovědi,
- inkrement odpovědi pro každou iteraci testu,
- počet iterací testu,
- minimální hodnoty histogramu,
- délku odesílaných packetů,
- interval mezi jednotlivými iteracemi testu,
- zobrazení histogramů,
- hodnotu prodlevy mezi odesíláním jednotlivých dávek packetů,
- konečné hodnoty čekací doby,
- inkrement čekací doby mezi odesíláním packetů,
- počet odesílaných packetů,
- počet odesílaných dávek packetů,
- nastavení intervalu mezi odesíláním packetů,
- nastavení počtu packetů, ve kterých server vrátí informace o příchodu a časování.

Dokumentace k měřicímu softwaru je přehledná, nabízí informace k instalaci i k jednotlivým testům spolu s příklady, jak jednotlivé testy spustit a manipulovat s výslednými daty. Přitom je dokumentace velmi stručná, takže se v ní nezkušený uživatel neztratí a není zahlcen informacemi. UDPMon má kombinaci testů a měření parametrů, které jsou potřebné pro tuto bakalářskou práci, a přitom v sobě nemá zbytečně moc specializovaných možností, které by ve výsledku nebyly využity a jen by to znehodnocovalo jeho jednoduchost.

Bohužel, v praxi se ukázalo, že díky stáří tohoto programu je jeho použití na novém hardwaru a softwaru obtížné a vyžaduje virtuální simulaci staršího operačního systému anebo určitou změnu v kódu pro přizpůsobení moderním překladačům. I po těchto krocích se ale objevovaly při měření chyby a virtualizace zanašela do měření nežádoucí parazitní zpoždění. Proto se jako měřicí software použije mnohem modernější a sofistikovanější iPerf2.

TCPMon je obdoba UDPMon, který ale využívá pro měření místo paketů UDP packety TCP. S UDPMon sdílí i autora, webovou stránku, podporu operačního systému a většinu funkcí. Do vývoje tohoto měřicího softwaru ale nebylo investováno tolik času, a proto je rozdělen do několika částí, které zprostředkovávají jednotlivá měření a je potřeba je zvlášť stáhnout a instalovat. Tím pádem i většina nedostatků a chyb nebyla opravena. Chybí i celá dokumentace. Toto jsou problémy, kterými trpí i další programy, umístěné na této webové stránce. UDPMon je nejvíce dokončený a udržovaný. [6]

```

response len bytes; Req len; loop_count; num_bad; time/frame (tod); ave time us;
min time; max time; num timeouts;
64; 64; 2000; 0; 127.79; 126.270; 120.77; 1193.952; 0
72; 64; 2000; 0; 127.812; 126.308; 121.49; 144.857; 0
80; 64; 2000; 0; 128.723; 127.209; 122.88; 158.274; 0
88; 64; 2000; 0; 234.72; 233.444; 123.31; 4839.224; 0
...
1496; 64; 2000; 0; 300.382; 298.813; 293.61; 317.982; 0

```

Obrázek 1.5 Příklad výstupu měřicího softwaru UDPMon [19]

1.7 perfSONAR [33]

Tento nástroj je kolekce open-source softwaru pro provádění a sdílení síťových měření. Obecně platí, že tyto nástroje nejsou volány přímo, ale k jejich provedení se použijí příkazy pscheduler z plánovací vrstvy. Mezi výchozí nástroje patří:

- owamp – Sada nástrojů primárně používaných pro měření ztráty paketů a jednosměrného zpoždění. Zahrnuje příkaz owping pro jednotlivé krátkodobé testy a příkaz powstream pro dlouhodobé testy na pozadí.
- twamp – Nástroj primárně používaný pro měření ztráty paketů a obousměrného zpoždění. Má zvýšenou přesnost oproti nástrojům jako ping bez stejných požadavků na synchronizaci hodin jako owamp. Klientský nástroj se jmenuje twping a lze jej použít ke spuštění proti serverům twamp. Je možno použít dodaný server twampd, ale mnoho routerů také přichází s implementací serverů twamp od dodavatele, proti kterým lze testovat.
- iPerf3.
- iPerf2.
- nuttcp – Další nástroj pro měření přenosové rychlosti s některými užitečnými možnostmi, které v jiných nástrojích nenajdete.
- traceroute – Klasický nástroj pro sledování paketů používaný při identifikaci síťových cest.
- tracepath – Další nástroj pro sledování cesty, který také měří MTU (Maximum transmission unit) cesty.
- paris-traceroute – Nástroj pro sledování paketů, který se pokouší identifikovat cesty v přítomnosti mechanismů vyrovnávacích zatížení.

- ping – Klasická utilita pro určení dosažitelnosti, obousměrného zpoždění a základní ztráty paketů.

Plánovací vrstva se skládá z jediného nástroje s názvem pScheduler, která je zodpovědná za nalezení časových úseků pro spouštění nástrojů a zároveň předcházení konfliktům v plánování, které by negativně ovlivnily výsledky, spouštění nástrojů a shromažďování výsledků a v případě potřeby odeslání výsledků do archivační vrstvy.

Archivační vrstva se skládá z jediné komponenty s názvem esmond, která ukládá informace o měření. Lze jej nainstalovat na každého hostitele měření, pokud splňují hardwarové požadavky, nebo může jediná centrální instance ukládat výsledky z více hostitelů měření.

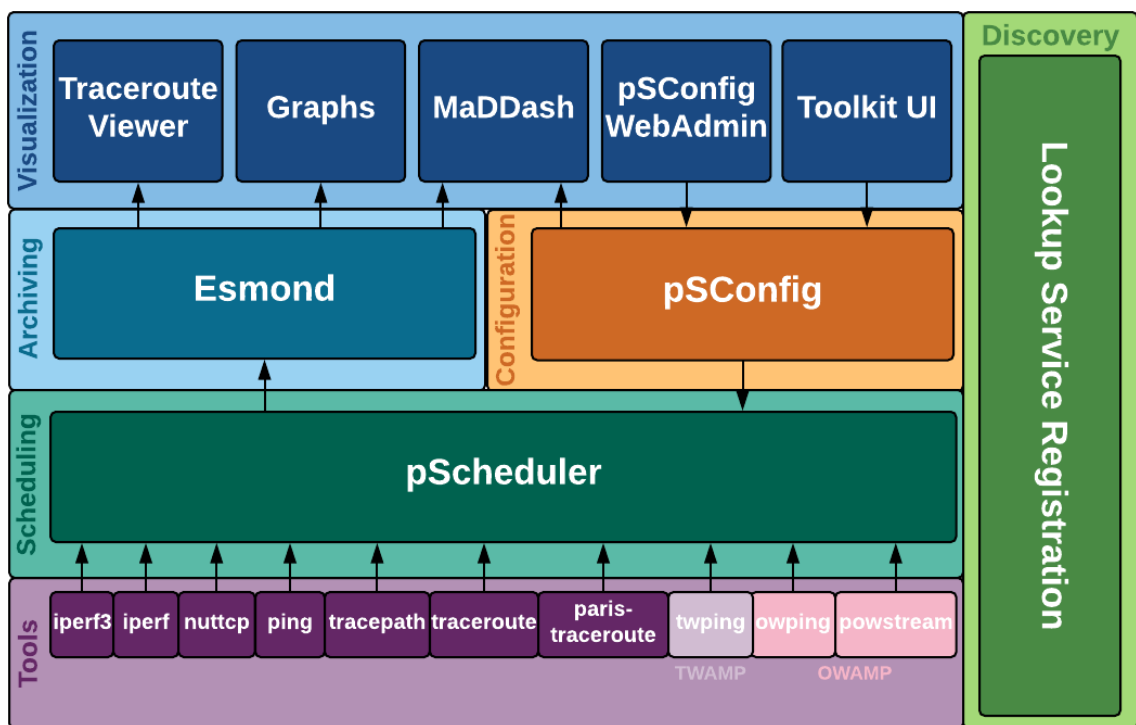
Konfigurační vrstva je místo, kde jsou definována požadovaná měření spolu s pokyny, kam je uložit. Primární komponenta na této vrstvě se nazývá pSConfig, což je šablonový rámec pro popis a konfiguraci topologie úloh.

Tento program také obsahuje komponenty pro vizualizaci dat. Tyto komponenty jsou primárním způsobem, jak většina operátorů analyzuje a identifikuje problémy se sítí. Primární nástroje jsou sada grafů, které zobrazují různá měření v průběhu času a poskytují užitečné informace o zúčastněných hostitelích a MaDDash, která se pravidelně dotazuje archivační vrstvy na měření a zobrazuje řídicí panel indikující výkon každého vzhledem k sadě definovaných prahových hodnot. Může také odesílat upozornění na základě vzorů v řídicím panelu.

Kromě zobrazení výsledků jsou k dispozici také grafická rozhraní pro konfiguraci komponent Toolkit GUI (Graphical user interface), které umožňuje definovat úkoly pro místního agenta pSConfig pScheduler a pSConfig Web Admin, webová aplikace pro definování vzdálených šablon, které mohou číst agenti pSConfig.

Každý uzel může spouštět komponentu zvanou Lookup Service (LS) Registration Daemon, která registruje jeho existenci ve veřejné anebo soukromé vyhledávací službě. Registrační daemon shromažďuje informace o každé vrstvě a také o hostiteli, na kterém běží. Tyto informace se pak použijí na více místech, aby pomohly ladit problémy a najít hostitele, se kterými lze testovat při vytváření nových konfigurací.

Tento program je další ukázkou předdimenzovaného měřicího systému, který je už ale zdarma a využívá měřicí softwary, o kterých už byla zmínka. Je tu spíše uveden, aby poskytl přehled o dalších nástrojích, které sice existují, ale jsou zastíněny nástroji jako iPerf2, a proto nejsou v rešerši samostatně uvedeny. Toto řešení není pro tuto bakalářskou práci příliš vhodné, protože by pro samotné měření měl stačit iPerf2 a následné zpracování a vizualizace dat už se bude řešit individuálně, pro možnost větší variability v zobrazování charakteristik.



Obrázek 1.6 Architektura programu perfSONAR [34]

1.8 Souhrn výsledků řešení

Závěrem může být řečeno, že softwarových řešení je vážně mnoho, ale většina je spíše cílená na rozsáhlé sítě velkých korporací. Jednoduchých konzolových programů, které změní pár základních parametrů je už mnohem méně a většinou jsou už zastaralé. Pro tuto bakalářskou práci by se dal použít prakticky jen iPerf2. Ostatní jsou buďto předimenzované a zbytečně složité anebo jsou až moc jednoduché a neměří všechny požadované parametry.

2. VÝZNAMNÉ PARAMETRY A CHARAKTERISTIKY PŘENOSOVÉ TRASY A JEJICH VHODNÉ GRAFICKÉ VYJÁDŘENÍ V RŮZNÝCH ZÁVISLOSTECH

Protože má bezdrátový přenos nespočet použití, tak to, jaký parametr a z něj odvozená charakteristika je důležitá, bude záviset na konkrétním využití. To znamená, že ty stejné parametry a charakteristiky u např. bezdrátového připojení notebooku studenta k internetu přes fakultní Wi-Fi nebudou tak důležité, jako u bezdrátové komunikace robotického ramena se vzdáleným řídicím systémem.

Toto rozdělení není jen na průmyslové a osobní využití. Např. výše zmíněný student může na notebooku stahovat několika gigabytový software, kde asi nejvíc ocení co nejvyšší rychlost stahování dat z internetu, zatímco pokud by na stejném notebooku chtěl hrát nějakou počítačovou hru náročnou na reakční dobu, ocenil by spíše nižší odezvu než vyšší rychlost připojení.

Stejný princip se promítne na tzv. průmyslovém využití. Např. některé roboty mohou pracovat i s vyšším zpožděním a ztráta pár packetů jim nezpůsobuje žádné problémy při fungování, zatímco jiné roboty budou při vyšších zpožděních a ztrátě packetů nepoužitelné.

Různé parametry, a to hlavně přenosová rychlost, mají v anglickém jazyce, ale i v češtině, různé označení (např. bandwidth, bit rate, throughput, atd...), které se mezi sebou pletou a zaměňují. Každý autor si každý parametr označuje různě, a proto je důležité pro úplnost a přehlednost uvést i ty parametry, které nejsou tak důležité a nebudou měřeny, ale vychází z nich ty parametry, které jsou důležité a budou měřeny.

2.1 Šířka pásma (Bandwidth, Gross Bit Rate)

Nejnámější definicí šířky pásma je asi rozdíl mezi horní a dolní frekvencí v nějakém nepřetržitém frekvenčním pásmu (jinými slovy frekvenční rozsah) a uvádí se v hertzech. Toto ale není ten parametr, který je ten důležitý. On sám název není ani tak správný, překlad slova z angličtiny sice správný je, ale v anglickém jazyce slovo bandwidth označuje jednak šířku pásma signálu v hertzech [24], ale v rámci výpočetní techniky i rychlost přenosu dat přes danou cestu v b/s, tzv. **přenosová rychlost**. [7] Tyto dva pojmy mají mezi sebou definovaný vztah pomocí Shannonovy-Hartleyovy věty [8]:

$$C = B \cdot \log_2(1 + SNR), \quad (2.1)$$

kde je:

C	kapacita kanálu (teoretická maximální rychlost přenosu) [b/s]
B	šířka pásma kanálu [Hz]
SNR	poměr signálu a šumu

2.1.1 Kapacita kanálu (Channel Capacity, Net Bit Rate)

Je dána použitým standardem fyzické vrstvy ISO/OSI (např. Ethernet, IEEE 802.11, atd.). [9] Logicky jde odvodit, že maximální přenosová rychlost přenosového kanálu nebo sítě bude určena jeho nejslabším, respektive nejpomalejším, článkem. [10] Např. pokud se bude systém skládat z počítače připojeného k routeru pomocí kabelu Ethernet, který zvládá 100 Mb/s (alespoň standard Fast Ethernet) a tento router bude komunikovat s druhým počítačem pomocí Wi-Fi standardu IEEE 802.11g, který zvládá 54 Mb/s [7], bude teoretická maximální přenosová rychlost omezena na 54 Mb/s. Rychlost přenosu dat v rámci zařízení přes sběrnice nebude uvažována, protože ta je zpravidla mnohonásobně vyšší, než přenosová rychlost bezdrátové komunikace, a proto ji nemůže nijak limitovat. Díky ní se může např. rozhodovat o optimálním nastavení velikosti vyrovnávací paměti TCP. [19] Tento parametr nebude nijak měřen a je zde uveden jen pro úplnost.

2.1.2 Maximální přenosová rychlost (Available Bandwidth, Throughput)

Tento parametr je odvozen z kapacity kanálu a udává maximální přenosovou rychlost, která není omezena už jen fyzickou vrstvou ISO/OSI, ale i agregací kapacity kanálu bezdrátového přenosu, protože na síti nemusí a v drtivé většině ani nepobíhá pouze jedna komunikace, ale pobíhá jich zároveň několik. [11] Pro každou komunikaci je pak v síti vyčleněno pásmo, které může využívat. [25] Dá se odvodit z kapacity kanálu a využití kanálu (agregovaná kapacita spotřebovaná po cestě). [19] Tento parametr pro tuto bakalářskou práci měřen nebude, protože bude stejný jako následující parametr, protože k agregaci kapacity kanálu bezdrátového přenosu docházet nebude a je zde uveden je pro úplnost.

2.1.3 Dosažitelná přenosová rychlost (Achievable Bandwidth, Goodput)

Tento parametr zase navazuje na ten předchozí – maximální přenosovou rychlost. Bude vždy menší než maximální přenosová rychlost, protože se od ní musí odečíst věci jako režijní byty protokolů (tzv. protokolová režie), limitace ostatních ISO/OSI vrstev jako je např. řízení toku transportní vrstvy, opakovaný přenos poškozených nebo ztracených paketů způsobený bitovými chybami anebo „zahazováním“ paketů při přetížení daného síťového prvku. [12] Příkladem je spousta.

Je důležité si uvědomit, že nemalá kapacita přenosu dat je použita pro řízení dané komunikace, kódování, adresaci, ověřování správnosti atd. a že tyto informace nejsou pro uživatele užitečné, i když stále zabírají kapacitu kanálu. Proto je tento parametr pro uživatele asi nejdůležitější, protože vypovídá o skutečné magnitudě přenesených čistých dat. [19] Vždy bude platit kapacita kanálu > maximální přenosová rychlost > opravdová přenosová rychlost.

Dále můžeme rozlišit **přenosovou rychlost odesílání (upload) a přenosovou rychlost příjmu (download)**. V mobilní robotice to není až zase tak důležitý parametr,

většinou zprostředkovává hodnoty pro osu x, ale i tak musí být dosažitelná přenosová rychlost dostatečná pro spolehlivé řízení robota a přenos dat mezi ním a ústřednou. V literatuře je velmi často nesprávně označována jako throughput, bandwidth, atd., a proto je důležité tyto pojmy rozlišovat.

2.2 Zpoždění (Latency)

Obecná definice zpoždění je doba mezi okamžikem, kdy první část (např. první bit) objektu (např. packet) projde pozorovací pozicí (např. tam, kde se karta síťového rozhraní hostitele připojí k drátu) a čas, kdy poslední část (např. poslední bit) tohoto objektu nebo souvisejícího objektu (např. packet odpovědi) projde druhým (může to být stejný bod) pozorovacím bodem. [18]

Největší faktor, který zpoždění ovlivňuje, je logicky vzdálenost mezi body A a B. Jak daleko musí informace urazit, aby se dostala do místa určení. Zpoždění je v mobilní robotice velmi nežádoucí. Rychlost přenosu dat sice může být vysoká, ale pokud je potřeba, aby robot reagoval bez prodlevy a zpoždění bude také vysoké, nebudou se data dostávat k robotu včas. Zpoždění lze měřit jednosměrně nebo obousměrně.

2.2.1 Obousměrné zpoždění (Round-trip Time – RTT, Round-trip Delay – RTD)

Pro měření tohoto parametru existuje asi ve všech známějších operačních systémech příkaz obdobný příkazu u operačního systému Windows `ping` (nebo `tracert`), pomocí kterého může být poslán proud packetů na zadanou IP adresu. Systém pak čeká na odpověď se zpožděním z kontaktovaného serveru a výsledek vypíše rovnou do konzole.

V zásadě může být obousměrné zpoždění složeno z měření jednosměrných zpoždění v obou směrech. To však vyžaduje provést jednotlivá měření blízko sebe a umět vybrat vhodná dvě měření, která se mají sečíst. Vzhledem k tomu, že obousměrné zpoždění se obvykle měří snadněji než jednosměrná zpoždění, obousměrné zpoždění se obvykle měří přímo tak, že se zaznamená čas, kdy je packet odeslán (často je tento čas zaznamenán v packetu samotném), a porovnává se s časem, kdy je packet s odpovědí přijat zpět z cíle. To vyžaduje, aby byl cíl připraven přijmout a odpovědět (tj. odeslat packet zpět do zdroje) na testovací packet. [14]

Na druhou stranu je často snazší měřit obousměrné zpoždění než jednosměrné zpoždění, protože je zapotřebí pouze jeden pozorovací bod a hodiny. Mnoho aplikací také závisí hlavně na obousměrném zpoždění. [19]

2.2.2 Jednosměrné zpoždění (One-way Delay, End-to-end Delay)

Jednosměrná zpoždění jsou důležitá, protože dnešní komunikace často využívají asymetrické cesty v síti nebo mají různou kvalitu služeb v obou směrech. I symetrické cesty mohou mít různé vlastnosti kvůli asymetrickému řazení přenesených dat do fronty. [19]

Také funkčnost aplikace může většinou záviset na zpoždění v jednom směru. [13] Např. výkon přenosu souborů může více záviset na výkonu ve směru toku dat nebo ve hře může být důležitější dostat požadavek na místo určení (server, na kterém hra běží) dříve než jiný hráč než získat odpověď.

Jednosměrné zpoždění může být většinou aproximováno jako polovina obousměrného zpoždění. Jednosměrné zpoždění se obvykle měří pomocí časového označení packetu při jeho vstupu do sítě a porovnáním tohoto časového označení s časem, kdy je packet přijat v cíli. To předpokládá, že hodiny na obou koncích jsou úzce synchronizovány. Pro přesnou synchronizaci (desítky μs) jsou hodiny často synchronizovány s GPS (Global Positioning System) anebo pomocí protokolu PTP. Pokud packet není přijat v cíli v určité době, pak je jednosměrné zpoždění nedefinované (dá se označit jako nekonečné) a packet je považován za ztracený. [19] Tím pádem lze říct, že se ztrátou packetů narůstá zpoždění.

2.2.3 Jitter (IP Packet Delay Variation – IPDV)

Tento parametr v je datové komunikaci definován jako kolísání (variance) ve velikosti jednosměrného zpoždění packetů při průchodu komunikačním kanálem. Je znázorněn samostatně, protože i když je obtížné přesně měřit jednosměrné zpoždění, je docela snadné měřit změny jednosměrného zpoždění pozorováním rozestupů mezi příchody packetů odeslaných v pevném intervalu. [19] Jitter je obecně velmi důležitý při určování velikosti vyrovnávací paměti přehrávání pro aplikace vyžadující pravidelné doručování packetů (např. hlasová komunikace nebo video). [17]

2.3 Ztráta packetů (Packet Loss)

Ztrátou packetů se označuje jev, kdy jeden nebo více odeslaných packetů v síti ztratí, a proto nedorazí do svého určeného cíle a nejsou přijaty. Ztráta packetů může být způsobena zhoršením kvality signálu v síti, zahazením packetů při přehlcení sítě, poškozením packetů rušením nebo selháním síťového hardwaru, ovladačů nebo směrování (hlavně u bezdrátové komunikace). Může být způsobena i záměrně při řízení provozu sítě (zamezení zahlcení u TCP protokolu). [15] Tyto problémy bývají spojeny i s vysokým zpožděním, a proto je ztráta packetů s vysokým zpožděním úzce spojena (většinou se při vysokém zpoždění packety ztrácejí).

V komunikaci při přenosu dat je to asi ten největší problém, protože některé aplikace potom nemusí dobře anebo vůbec fungovat, pokud hodnota ztráty packetů překročí určitou mez. Čím vyšší je hodnota ztráty packetů, tím obtížnější je pro protokoly transportní vrstvy udržet velké přenosové rychlosti. A pokud je potřeba podporovat procesy s velmi velkým zpožděním, citlivost aplikací a protokolů transportní vrstvy na hodnotu ztráty packetů se stává zvláště důležitou. [19] Tato citlivost na ztrátu jednotlivých packetů, její frekvenci a vzory těchto ztrát pochopitelně závisí na samotné aplikaci. Např. u VoIP (Voice over Internet Protocol) nemá občasná

ztráta packetu žádný vliv na kvalitu konverzace, ale u přenosu souborů může jediný ztracený packet zapříčinit ztrátu části souboru. Např. u streamovaných audiovizuálních médií má ztráta packetů za následek sníženou kvalitu zvuku a obrazu, která se může projevit u obrazu tzv. „glitchováním“ (krátkodobým rozpixelováním obrazu) a u zvuku tím, že může znít „roboticky“. U datových přenosů může ztráta packetů způsobit vážné zhoršení dosažitelné přenosové rychlosti. [15]

Charakteristiku ztráty packetů si můžeme představit jako binární, kde 0 představuje úspěšný přenos ze zdroje do cíle a 1 označuje ztracený packet. [19] Protokol TCP garantuje spolehlivý přenos packetů. V případě ztráty packetů je komunikace buď zopakována, což má za následek pokles přenosové rychlosti a zvýšení zpoždění (protože znovu posíláme i data, které byly už jednou přijaty), anebo jsou přeposlány jen ztracené packety. Toto záleží na variantě protokolu TCP. Oproti tomu protokol UDP neumožňuje jakoukoli obnovu ztracených packetů. To už si řeší samotné aplikace využívající UDP protokol individuálně. [15]

Stejně jako u zpoždění, i ztráta packetů je řešena u všech známých operačních systémů terminálovým příkazem `obdobným příkazem terminálu operačního systému Windows ping (nebo tracert), které kromě hodnoty zpoždění, změří a vytisknou do konzole i procentuální ztrátu packetů a který packet byl ztracen.`

2.3.1 Jednosměrná ztráta [19]

V síti se cesta ze zdroje do cíle může lišit od cesty z cíle zpět ke zdroji (tzv. asymetrické cesty). I v případě symetrických cest pro oba směry může dodatečný provoz z jiných aplikací způsobit odlišné chování ve frontě v obou směrech. Obousměrná měření proto směřují vlastnosti obou směrů cesty, což může vést k zavádějícím výsledkům. (Např. dosažitelná přenosová rychlost přenosu souboru může silně záviset na ztrátě packetů na cestě ze zdroje k cíli, přičemž je do značné míry nezávislá na zpětném směru.) Z těchto důvodů jsou obecně preferována jednosměrná měření ztrát. Ztráta packetů je velmi závislá na cestě, kterou data jdou, a také na použitých protokolech a vybraných úrovních QoS.

2.4 Přeskupení packetů (packet re-ordering)

Když jsou packety odesílány přes přenosový kanál, mohou být doručeny na místo určení v nesprávném pořadí. Může existovat mnoho důvodů, proč může na cestě dojít k přeuspořádání packetů. Je také pravděpodobné, že existuje určitá prahová hodnota v dopravní zátěži, nad kterou může dojít k přeražení. Občasná výměna jednoho packetu s jeho sousedem má často malý dopad na síťové aplikace, ale opakované výskyty mohou mít významný vliv. [19] Např. u TCP protokolu to může spustit opětovné odeslání packetů, i když se neztratili, což zase sníží dosažitelnou přenosovou rychlost. Aplikace využívající UDP protokol musí toto řešit individuálně. Statistické vlastnosti událostí změny pořadí packetů jsou tedy nejzajímavější pro určení výkonu aplikace. [16]

2.5 Vhodné grafické vyjádření významných charakteristik přenosové trasy v různých závislostech

Bylo rozhodnuto, že se pomocí výše uvedených parametrů do grafů vynesou následující charakteristiky, které přispějí k analýze bezdrátového přenosového kanálu.

XY bodové grafy se spojnicemi:

- závislost na vzdálenosti od řídicí stanice
 - závislosti přenosové rychlosti příjmu dat na přenosové rychlosti odesílání dat při dané konfiguraci routeru
 - závislosti ztráty packetů na přenosové rychlosti odesílání dat při dané konfiguraci routeru
 - závislosti jednosměrného zpoždění na přenosové rychlosti odesílání dat při dané konfiguraci routeru
 - závislosti jitteru na přenosové rychlosti odesílání při dané konfiguraci routeru

Sloupcové grafy:

- shrnutí a porovnání výsledků z XY bodových grafů se spojnicemi

Tabulky:

- velikost přenosové rychlosti příjmu dat při různých síťových výkonech v různých vzdálenostech od řídicí stanice

3. POŽADOVANÉ PARAMETRY ANALYZÁTORU PRO UVAŽOVANOU APLIKACI

Pro aplikaci analyzátoru bezdrátového komunikačního spoje v mobilní robotice bude důležité, aby splňoval několik předem daných kritérií. Kromě nízké pořizovací ceny, bude zejména důležité, aby dokázal:

- měření provádět pomocí UDP packetů,
- změřit všechny požadované parametry,
- změřit všechny požadované parametry v dostatečném rozsahu,
- změřit všechny požadované parametry s dostatečnou přesností,
- být měřicí systém implementován na malý vestavný systém a tím pádem by mohl být připevněn na mobilní robotu a pohybovat se s ním,
- mít proces měření zautomatizován,
- mít výstup naměřených dat do textového souboru, který bude zpracovatelný v softwaru pro analýzu dat,
- být připojen ke komunikačním modulům disponujícími rozhraním Ethernet,
- podporovat protokol IPv4.

3.1 Hardware

Pro hardware může být předpokládáno, že přenosové rychlosti v mobilní robotice, nebudou nijak závratné. Můžou být v desítkách Mb/s, a proto bude pro kabeláž stačit minimálně standard Fast Ethernet.

Tento požadavek pak bude také podmiňovat hardware, na kterém měřicí software poběží. Z rešerše je patrné, že pro měření parametrů bezdrátového komunikačního kanálu je zapotřebí využití principu server – klient a tzn. použití dvou počítačů. Klient měřicího systému bude moci vykonávat prakticky jakýkoli počítač, hlavní u něj bude, aby disponoval rozhraním alespoň Fast Ethernet pro připojení k routeru, dokázal zvládnout operační systém na bázi Linuxu a daly se z něj extrahovat naměřená data. Všechny rozhraní Ethernet by měli být schopny podporovat časovou synchronizaci pro její nevyhnutelné použití.

Server měřicího systému už bude mít požadavky vyšší. Měl by být pevně spojen přímo s mobilním robotem, a proto to nesmí být žádný velký stolní počítač nebo notebook. Uvažován je spíše levný jednodeskový počítač s malým výkonem, který je možno po delší dobu napájet z akumulátoru. Toto se potom bude muset zohlednit ve výběru softwaru, aby výkon takového malého počítače stačil. Dále bude požadováno to, co u klienta měřicího systému. Bude muset disponovat rozhraním alespoň Fast Ethernet pro připojení k robotu, dokázat zvládnout operační systém na bázi Linuxu a mít možnost se z něj extrahovat naměřená data.

3.2 Software

Měřicí software bude muset umět změřit několik nezbytně důležitých parametrů. Jsou jimi přenosová rychlost, jednosměrné zpoždění, jitter, ztrátu packetů a přeskupení packetů, a to vše obousměrně. Proto je zde logické požadovat určitou funkci reverzace měření, aby se dali všechny měření provádět z jednoho konce a nemuselo se ručně přepínat na obou koncích mezi serverovým a klientským režimem.

Dále musí být měřicí systém schopen pro měření využívat UDP packety, aby byla zřetelná skutečná ztráta packetů a jednosměrné zpoždění. [35] Rozsah jednotlivých měření můžeme uvážit od stovek kb/s až po desítky Mb/s u přenosové rychlosti. Jednosměrné zpoždění se bude pohybovat v rozmezí od jednotek až po stovky milisekund. Jitter se bude pohybovat v rozmezí od desetin až po jednotky milisekund. Rozmezí ztráty packetů je logicky 0 – 100 %. Přeskupování packetů by mělo být velmi vzácné a když už k němu nastane, budou to jednotky packetů. Při nenalezení měřicího softwaru, který by splňoval tyto požadavky, by se muselo přejít k vlastnímu naprogramování měřicího softwaru, který by už všechny tyto parametry byl schopen změřit v požadovaném rozsahu, přesnosti a při použití UDP packetů.

Jedinou přesnost měření, kterou můžeme ovlivnit je správné měření jednosměrného zpoždění. U toho bude nesmírně důležitá co nejpřesnější synchronizace hodin obou konců. Toto se může projevit ve výběru operačního systému, který poběží na počítačích měřicího systému.

Pokud bude měřicí software a další použité programy fungovat v konzoli, nebude s výstupem do textového souboru žádný problém, tím pádem nebude problém asi s automatizací těchto měření a ukládání naměřených dat do textového souboru pomocí jednoduchého bash scriptu a nastavení služeb, které se po startu měřicího systému spustí. Pokud bude měřicí systém využívat své vlastní grafické prostředí, může být automatizované měření a ukládání naměřených dat do souboru problematické.

Výběr operačního systému bude záviset na tom, jestli na něm zvolený měřicí software poběží. Dále musí podporovat použití časové synchronizace, ale nebýt příliš náročný při použití ve vestavném systému. Přitom ale nesmí být moc primitivní, aby v něm bylo možné vše jednoduše nakonfigurovat pro zautomatizované provádění měření. Schopnost vzdáleného přístupu k jinému zařízení bude taky velmi důležitá, aby byla změna konfigurace měřicího softwaru na serveru měřicího systému možná provádět z dálky.

4. NÁVRH ANALYZÁTORU PRO POŽADOVANÉ PARAMETRY

4.1 Software

4.1.1 Měřicí software a operační systém

Pokud porovnáme řešerši již existujících nástrojů s výše uvedenými požadovanými parametry, tak se jako jasná a jediná volba ukazuje měřicí software iPerf2. Tento software je multiplatformní, což nám dává možnost výběru různých operačních systémů. Primárně je ale vyvíjen na operační systém Linux, a proto bude zvolen pro realizaci tohoto měřicího systému. Rovnou se tím předejde nečekaným chybám a problémům při případné nekompatibilitě s jiným operačním systémem.

Test měření s jedním koncem na operačním systému Linux a druhým koncem na operačním systému Windows proběhl, ale výsledky nebyly uspokojivé. Vyskytlo se až příliš mnoho problémů s kompatibilitou, které znehodnotili výsledky měření a jejichž řešení by vyžadovalo hodně úsilí a času, který není k dispozici.

4.1.2 Časová synchronizace

NTP [36]

Výběr měřicího softwaru iPerf2 otevírá problém časové synchronizace obou konců, pro korektní měření jednosměrného zpoždění. Časová synchronizace se normálně provádí pomocí protokolu NTP většinou přes internet, protože hodiny na počítačích jsou i s vlastním napájením hodně nestabilní. Počítač komunikuje s NTP serverem pomocí UDP packetů přes port 123, který slouží jako reference přesného času a žádá ho, aby mu tento čas poskytl. NTP server poté pošle počítači přesný čas, dle kterého si aktualizuje svůj lokální čas. Využívá upravenou verzi Marzullova algoritmu, k výběru přesných časových serverů a je navržen tak, aby zmírnil účinky proměnného zpoždění sítě.

Tento protokol přišel do aktivního používání v roce 1985 a za jeho vývoj a design je zodpovědný David Lennox Mills. Za tuto dlouhou dobu je už NTP server a klient stálíci v každém operačním systému pod různě nazvanými implementacemi (např. Windows Time u operačního systému Windows).

NTP může obvykle udržovat čas v rozmezí desítek milisekund přes internet a v sítích LAN může dosáhnout za ideálních podmínek přesnosti lepší než 1 ms. Asymetrické trasy a zahlcení sítě mohou způsobit chyby 100 ms nebo více. Pokud je ale požadavek na přesnost měření jednosměrného zpoždění v jednotkách až desítkách milisekund, je patrné, že bude tento způsob synchronizace nedostatečný a bude se muset využít přesnějšího řešení.

PTP [37]

Řádově přesnější řešení je využití protokolu PTP. V sítích LAN dokáže dosáhnout přesnosti pod 1 ms, což už je pro tyto měření přijatelná chybovost a implementace pro Linux běží v konzoli. PTP se v současnosti používá k synchronizaci finančních transakcí, přenosů věží mobilních telefonů, podmořských akustických polí a jako kvalitní substitute GPS u sítí, které vyžadují přesné načasování, ale nemají přístup k signálům satelitní navigace.

Původní verze IEEE 1588-2002, byla zveřejněna v roce 2002. Druhá verze IEEE 1588-2008, zavedena v roce 2008, zlepšila přesnost a robustnost, ale není zpětně kompatibilní s původní verzí z roku 2002. Nejnovější verze IEEE 1588-2019, zavedena v listopadu 2019, obsahuje zpětně kompatibilní vylepšení publikace z roku 2008. PTP využívá pro komunikaci datagramy UDP na portech 319 a 320.

Architektura systému distribuce času se skládá z jednoho nebo více síťových segmentů a jednoho nebo více hodin. Hodiny jsou buď zdrojem (master) nebo cílem (slave) synchronizační reference. Pro každý síťový segment v systému je vybrán synchronizační master. Hlavní časová reference se nazývá grandmaster. Grandmaster přenáší synchronizační informace do hodin umístěných v jeho síťovém segmentu, které je pak dokážou distribuovat do dalších síťových segmentů. V případě této bakalářské práce bude princip zjednodušený. Hodiny budou pouze dvě (jeden master, respektive grandmaster a jeden slave) v jednom síťovém segmentu.

Pro zvolení nejlepších hodin jako master (respektive grandmaster) se využívá algoritmu BMC, který je zvolí z následujících vlastností:

- Identifikátor – Univerzálně jedinečný číselný identifikátor hodin, který je obvykle konstruován na základě MAC adresy zařízení.
- Kvalita – PTP se pokouší kvantifikovat kvalitu hodin na základě očekávané odchylky časování, technologie použité k implementaci hodin nebo umístění ve schématu vrstvy hodin.
- Priorita – Administrativně přiřazená nápověda k prioritě, která pomáhá vybrat grandmaster pro doménu PTP.
- Rozptyl – Odhad hodin o jejich stabilitě na základě pozorování jejich výkonu oproti referenci PTP.

Aktuální master hodin vysílá informace o vlastnostech hodin v pravidelných intervalech. Hodiny, které se považují za lepší master, přenesou tuto informaci, aby vyvolaly změnu. Jakmile aktuální master rozpozná lepší hodiny, přestane vysílat a lepší hodiny převzou funkci master. Algoritmus BMC bere v úvahu pouze vlastní deklarovanou kvalitu hodin a nebere v úvahu kvalitu síťového spojení.

Synchronizační algoritmus je oproti NTP mnohem přesnější, protože bere v potaz i dobu, kterou synchronizační zpráva stráví v každém zařízení. Další věc, která přispívá k větší přesnosti tohoto protokolu je ta, že není pouze softwarově zaměřen, ale dokáže použít pro přesnou časovou referenci i externí zdroj přesného času, takže se největší

nepřesností stane jen nestabilita hodin v daném zařízení. Ostatní zdroje nepřesností jako zpoždění přerušení, plánování operačního systému atd. bohužel zůstávají, ale jejich efekt je dostatečně malý, aby mohl být zanedbán při přesnosti synchronizace na setiny milisekundy.

Softwarových implementací tohoto protokolu existuje spousta, ale pro operační systémy na bázi Linuxu se nabízí program PTPd anebo ptp4l. Tyto softwary jsou si velmi podobné, ale PTPd dokáže podporovat časovou synchronizaci z operačního systému na bázi Linuxu na operační systém Windows (operačnímu systému Windows 10 byl přidán PTP klient). V praxi se ale ukázalo, že je ptp4l mnohem spolehlivější, a proto bude pro časovou synchronizaci využito zrovna tohoto programu. [38]

```

pi@raspberrypi: ~
ptp4l[141715.688]: master offset -11171 s2 freq +10957 path delay 311349
ptp4l[141716.688]: master offset -5260 s2 freq +11543 path delay 311349
ptp4l[141717.688]: master offset 1879 s2 freq +12259 path delay 311349
ptp4l[141718.688]: master offset 13536 s2 freq +13438 path delay 311349
ptp4l[141719.689]: master offset 1806 s2 freq +12267 path delay 316108
ptp4l[141720.689]: master offset 21418 s2 freq +14249 path delay 313278
ptp4l[141721.689]: master offset -3318 s2 freq +11772 path delay 309502
ptp4l[141722.689]: master offset 20939 s2 freq +14219 path delay 308451
ptp4l[141723.689]: master offset -8663 s2 freq +11250 path delay 305297
ptp4l[141724.689]: master offset 18720 s2 freq +14007 path delay 305297
ptp4l[141725.689]: master offset -6127 s2 freq +11516 path delay 305042
ptp4l[141726.690]: master offset 31571 s2 freq +15318 path delay 305297
ptp4l[141727.690]: master offset -2023 s2 freq +11956 path delay 304720
ptp4l[141728.690]: master offset -2076 s2 freq +11949 path delay 303315
ptp4l[141729.690]: master offset 52267 s2 freq +17435 path delay 302015
ptp4l[141730.690]: master offset -13093 s2 freq +10886 path delay 300559
ptp4l[141731.690]: master offset -11501 s2 freq +11034 path delay 301530
ptp4l[141732.690]: master offset -7937 s2 freq +11382 path delay 300371
ptp4l[141733.690]: master offset 30087 s2 freq +15215 path delay 299558
ptp4l[141734.690]: master offset 8221 s2 freq +13037 path delay 297738
ptp4l[141735.691]: master offset 39489 s2 freq +16203 path delay 295673
ptp4l[141736.691]: master offset -22943 s2 freq +9937 path delay 292625
ptp4l[141737.691]: master offset 24057 s2 freq +14661 path delay 291363
ptp4l[141738.691]: master offset 52997 s2 freq +17608 path delay 281832
ptp4l[141739.691]: master offset 726 s2 freq +12381 path delay 291363

```

Obrázek 4.1 Výstup softwaru pro časovou synchronizaci ptp4l

4.1.3 Dálkový přístup [39]

Pro dálkový přístup k serveru měřicího systému a k jeho ovládání je mnoho kandidátů, ale existuje jen jeden takto zavedený a vyzkoušený standard a tím je SSH. Je to šifrovací síťový protokol pro bezpečný provoz síťových služeb přes nezabezpečenou síť, a jeho nejvyhledávanějšími aplikacemi jsou vzdálené přihlášení k zařízení a jeho ovládání přes příkazový řádek.

Tento protokol je založen na architektuře klient – server a funguje na třech principech. Transportní vrstva zajišťuje autentizaci serveru pro důvěrnost a integritu, protokol pro autentizaci uživatele ověřuje uživatele na serveru a spojovací protokol multiplexuje šifrované spojení do více logických komunikačních kanálů.

SSH bylo navrženo na operačních systémech podobných Unixu jako náhrada za Telnet a za nezabezpečené vzdálené protokoly Unix shell, jako je rsh (Remote Shell) a související protokoly rlogin a rexec, které všechny používají nezabezpečený přenos autentizačních tokenů v prostém textu.

Poprvé ho navrhl v roce 1995 finský počítačový vědec Tatu Ylönen. Následný vývoj sady protokolů probíhal v několika vývojářských skupinách, přičemž vzniklo několik variant implementace. Specifikace protokolu rozlišuje dvě hlavní verze, označované jako SSH-1 a SSH-2. Implementace jsou distribuovány pro všechny typy běžně používaných operačních systémů, včetně vestavných systémů. SSH má nespočet využití, ale zde bude používáno pouze pro dálkové ovládání serveru měřicího systému při měření a případný přenos textového souboru s detailnějším výstupem měření do klientského zařízení pomocí SCP (secure copy) pro vyhodnocení výsledků.

4.2 Hardware

4.2.1 Klient měřicího systému

Pro funkci klienta může být použit jakýkoliv počítač, který dokáže podporovat operační systém na bázi Linuxu a disponuje rozhraním alespoň Fast Ethernet. Při měření na routerech nebo řídicí stanicích bude výhodné, mít klienta měřicího systému už připravený a nakonfigurovaný, aby jej stačilo jen přinést a připojit k danému routeru nebo řídicí stanici. Proto bude spíše logické využití notebooku než stolního počítače.

Pokud je předpokládáno, že se bude router nebo řídicí stanice také pohybovat, může být notebook bez problémů substituován nějakým vestavným systémem, který se bude stejně jako server měřicího systému ovládat přes SSH, protože by bylo složité a někdy i nemožné vyřešit upevnění tohoto hardwaru k pohyblivému se routeru nebo řídicí stanici. Problém by byl i s jeho napájením, protože by baterie notebooku nemusela při dlouhodobém měření stačit. Pro měření v této bakalářské práci se ale router nebo řídicí stanice pohybovat nebude a může být umístěna blízko zdroje elektrické energie, aby ji z něj bylo možno napájet stejně jako notebook.

Protože požadavky nejsou nijak limitující, byl pro zprostředkování klienta vybrán 11 let starý notebook od společnosti HP. Disponuje dvoujádrovým procesorem Intel Pentium B950 s frekvencí 2,1 GHz, 4 GB paměti RAM DDR3 a rozhraním Fast Ethernet. Tyto specifikace jsou více než dostačující pro chod operačního systému na bázi Linuxu, měřicího softwaru iPerf2, ptp4l a SSH zároveň.

Jako operační systém si díky výběru notebooku můžeme zvolit Ubuntu verzi Desktop. Doporučené systémové požadavky jsou dvoujádrový procesor s frekvencí 2

GHz nebo lepší, 4 GB paměti RAM a pak věci, které jsou na notebooku vždy, jako USB port nebo CD (compact disk) mechanika, dostatečná paměť na pevném disku a internetové připojení. Z předchozího odstavce je zřejmé, že tyto požadavky dokáže notebook splnit. Ubuntu je uživatelsky přátelský operační systém na bázi Linuxu, přesněji operačním systémem Debian, s grafickým rozhraním GNOME (GNU Network Object Model Environment), což velmi usnadní provádění měření a konfiguraci notebooku samotného. Navíc je open-source a jeho stažení je zdarma. [40]

Další výhodou je, že měřicí software iPerf2, ptp4l a SSH jsou obsaženy v softwarových balíčcích pro tento operační systém, což usnadní jejich instalaci a aktualizaci, protože odpadne manuální proces stahování a instalace. Pro spojení s routerem nebo řídicí stanicí bude využito kabelu standardu alespoň Fast Ethernet.



Obrázek 4.2 Klient měřicího systému (notebook) připojený k řídicí stanici

4.2.2 Server měřicího systému

Pro funkci serveru měřicího systému se už bude muset použít malý jednodeskový počítač, který nezabírá moc místa, aby nebyl problém s upevněním na mobilní robot a bude moci být napájen z baterie pro mobilní robot. Dále bude muset disponovat alespoň rozhraním Fast Ethernet.

Projektů, které by splňovaly tyto požadavky je spousta a jsou si velmi podobné. Ten nejznámější a nejzavedenější z nich je bezpochyby Raspberry Pi. Jak už bylo řečeno, projekty jemu podobné, jsou prakticky stejné, takže se nemá cenu zabývat jejich srovnáváním a rovnou vybrat vhodného kandidáta z rodiny Raspberry Pi.

Nejvhodnější se ukázal nejnovější model klasického Raspberry Pi a to Raspberry Pi 4 Model B, který disponuje čtyřjádrovým procesorem Broadcom BCM2711, Cortex-A72 (ARM v8) 64-bit SoC s frekvencí 1,5 GHz, pamětí RAM LPDDR4-3200 s kapacitou buď 1, 2, 4 nebo 8 GB (4 GB budou zcela dostačující) a rozhraním Gigabit Ethernet. K napájení tohoto modelu je potřeba napětí 5 V a proudu minimálně 3 A, což je lehce realizovatelné. Jeho cena je také příznivá a pohybuje se od 990 do 2250 korun dle kapacity paměti RAM. [28]

Výběr operačního systému je u Raspberry Pi velmi jasný. Nejlepším kandidátem je oficiální operační systém na bázi Linuxu pro Raspberry Pi Raspberry pi OS (dříve Raspbian), jenž je založený na operačním systému Debian. Vyšel v roce 2012 a je vysoce optimalizovaný pro zařízení z rodiny Raspberry Pi s ARMovými procesory. Navíc je open-source a jeho stažení je zdarma. Jako grafické rozhraní používá modifikovaný LXDE (Lightweight X11 (X Window System) Desktop Environment) PIXEL. V roce 2022 byla vydána i 64bitová verze, ale použití 32bitové verze nijak neovlivní výkon ani průběh měření. [41]

Stejně jako u operačního systému na bázi Linuxu Ubuntu, je měřicí software iPerf2, ptp4l a SSH obsažen v softwarových balíčcích pro tento operační systém (SSH je rovnou předinstalováno, stačí ho jen povolit v konfiguraci Raspberry Pi), což usnadní jejich instalaci a aktualizaci, protože odpadne manuální proces stahování a instalace. Pro spojení s routerem bude využito kabelu standardu alespoň Fast Ethernet.



Obrázek 4.3 Server měřicího systému (Raspberry Pi) připojený k přijímači mobilního robota

5. REALIZACE ANALYZÁTORU A OVĚŘENÍ JEHO VLASTNOSTÍ A SPRÁVNÉ FUNKCE

5.1 Klient měřicího systému

Na notebook, který bude zprostředkovávat klienta měřicího systému, byl nainstalován operační systém na bázi Linuxu Ubuntu dostupný na jeho oficiálních stránkách <https://ubuntu.com/download/desktop>. Přesněji Desktop verze 20.04 LTS, vydána 23. dubna 2020. LTS je zkratka anglické fráze „long-term support“, což označuje verzi operačního systému, která má zaručenu dlouhodobou podporu prostřednictvím aktualizací. Tato podpora je u tohoto operačního systému dlouhá 5 let, tzn. do dubna 2025 (pro bezpečnostní aktualizace je ještě o 5 let prodloužená, tzn. 10 let, do roku 2030). Tento fakt dokáže zaručit snadnou replikovatelnost řešení analyzátoru. V průběhu tvorby této bakalářské práce byla 21. dubna 2022 vydána nová verze operačního systému Ubuntu a to 22.04 LTS (což znamená podporu do dubna 2027 a prodloužené bezpečnostní aktualizace do roku 2032), ale přechod na ni už nedává z časového hlediska smysl. [40] Instalace proběhla standardně. [48]

Nejprve se musí zajistit aktualizace knihoven dostupných balíčků softwaru a softwaru, který už je na operačním systému Ubuntu nainstalovaný. To se provede otevřením příkazového řádku a zadáním příkazů `sudo apt update` a `sudo apt upgrade`, přičemž se vždy při vyzvání musí stahování souborů potvrdit zadáním písmene `y` a stisknutím klávesy Enter (to platí i u dalším ovládání přes příkazový řádek).

Příkaz `sudo` vykonává příkaz s administrátorským oprávněním [43] a příkaz `apt` spolupracuje s hlavními knihovny, pro instalaci a odstraňování softwaru na operačním systému na bázi Linuxu Debian a jiných distribucích na Debianu založených. Příkaz `apt` zjednodušuje proces správy softwaru na počítačových systémech podobných Unixu automatizací získávání, konfigurace a instalace softwarových balíčků, a to buď z předkompilovaných souborů, nebo kompilací zdrojového kódu, čímž odpadá manuální kompilace a instalace. [42]

Dále se nainstaluje měřicí software `iperf2` pomocí příkazu `sudo apt install iperf`. Tento příkaz by měl nainstalovat nejnovější verzi tohoto měřicího softwaru, což se dá zjistit příkazem `iperf -v`. Pokud ale není nejnovější verze tohoto měřicího softwaru zahrnuta v knihovně softwaru, je ji možno stáhnout z oficiální webové stránky projektu `iperf2` <https://sourceforge.net/projects/iperf2/>, extrahovat soubory ze stáhnutého tar.gz balíčku, pomocí konzolového příkazu `cd` se dostat do extrahovaného souboru, nakonfigurovat měřicí software pomocí příkazu `./configure`, zkompilovat měřicí software pomocí příkazu `make` a nakonec měřicí software nainstalovat pomocí příkazu `make install`.

Instalace měřicího softwaru iperf2 ať už pomocí příkazu `sudo apt install iperf` nebo pomocí manuálního postupu uvedeného výše bude vyžadovat přítomnost GCC C++ kompilátoru a nástroje Automake. Pokud nejsou na operačním systému přítomny, mohou se tyto nástroje nainstalovat pomocí konzolových příkazů `sudo apt install gcc` pro GCC kompilátor, `sudo apt install gcc-c++` pro GCC kompilátor C++ verzi a `sudo apt install automake` pro nástroj Automake. Všechny tyto nástroje by měli být zahrnuty a tím pádem by se měli všechny dohromady nainstalovat příkazem `sudo apt install build-essential`. [30]

Dále je zapotřebí nainstalovat software pro časovou synchronizaci ptp4l. Toto se provede příkazem `sudo apt install linuxptp`. [38] Pokud na operačním systému chybí nainstalovaný software pro dálkový přístup, nainstaluje se pomocí příkazu `sudo apt install openssh-client`. Toto nainstaluje jen klientskou část SSH, protože se nepředpokládá využití serverové části SSH na klientovi měřicího systému, pokud by byl ale SSH server potřeba, dá se v případě jeho nepřítomnosti na operačním systému doinstalovat pomocí příkazu `sudo apt install openssh-server`. [44]

Další konfigurace nástrojů a softwaru už není zapotřebí. Tímto se dá považovat klient měřicího systému za připravený k použití a může se přejít k instalaci a přípravě serveru měřicího systému.

5.2 Server měřicího systému

Na Raspberry Pi 4 Model B, které bude zprostředkovávat server měřicího systému, byl nainstalován operační systém na bázi Linuxu Raspberry Pi OS. Přesněji Raspberry Pi OS (operating system) with desktop (toto znamená, že instalace obsahuje grafické rozhraní) 32-bit, vydán 28. ledna 2022. V průběhu tvorby této bakalářské práce byla 4. dubna 2022 vydána nová verze operačního systému Raspberry Pi OS s novým Linuxovým jádrem 5.15.30, ale přechod na ni už nedává z časového hlediska smysl. Instalace se provádí nahráním operačního systému Raspberry Pi OS na microSD kartu o minimální kapacitě 8 GB (pro Raspberry Pi OS Lite je minimální kapacity microSD karty 4 GB).

Nahrání operačního systému na microSD kartu se provádí pomocí oficiálního open source softwaru Raspberry Pi Imager, který se dá zdarma stáhnout a nainstalovat z oficiálních webových stránek Raspberry Pi <https://www.raspberrypi.com/software/>. Tato webová stránka obsahuje odkazy na nejnovější verze imageru pro operační systémy Windows, macOS a Ubuntu x86. Na operačních systémech na bázi Linuxu se dá také nainstalovat pomocí příkazu `sudo apt install rpi-imager`, pokud je zahrnut v knihovněch.

Dále je potřeba počítač s čtečkou microSD karet nebo s čtečkou SD karet a mít převodník z microSD karty na SD kartu. Po vložení microSD karty do čtečky se může

spustit Raspberry Pi Imager a zvolí se správný typ a verze operačního systému (Raspberry Pi Imager podporuje i jiné verze Raspberry Pi OS a jiné operační systémy pro Raspberry Pi), který chceme nahrát na microSD kartu. V tomto případě je to výchozí možnost. Poté se zvolí microSD karta, na kterou chceme operační systém nahrát a může se spustit zápis, který trvá pár minut dle velikosti operačního systému. Pak už stačí jen microSD kartu vytáhnout z čtečky karet a vložit ji do slotu pro microSD kartu na Raspberry Pi. Po startu Raspberry Pi by měla jít vidět prázdná plocha grafického rozhraní. [45]

První se ale musí vyřešit napájení, obrazovka a myš. Napájení Raspberry Pi jde vyřešit mnoha způsoby. U ověření jeho funkčnosti bude ale nejjednodušší napájet ho ze zásuvky. Protože ale Raspberry Pi 4 Model B pro své napájení potřebuje napětí 5 V a ne 230 V, budeme muset pro jeho napájení použít vhodný adaptér, který dokáže splnit i jeho proudové nároky (3 A). Existuje adaptér oficiálně vyráběný pro Raspberry Pi 4 Model B, který stojí asi 280 Kč. [47] Toto je nejspolehlivější a nejjednodušší řešení pro tuto situaci a následnou konfiguraci.

Pro připojení obrazovky používá Raspberry Pi 4 Model B Micro HDMI konektor. Nejjednodušší je sehnat obrazovku s HDMI konektorem a propojit je kabelem s jedním koncem HDMI a druhým koncem Micro HDMI. Jde ale také sehnat převodník HDMI na Micro HDMI a kabel s jedním koncem HDMI a druhým koncem VGA (Video Graphics Array) nebo DVI (Digital Visual Interface). Tím pádem půjde k Raspberry Pi připojit jakákoliv standardní i stará obrazovka. Pro myš a klávesnici postačí standardní rozhraní USB. [28]

Po připojení Raspberry Pi k napájení se okamžitě spustí. Nejprve bude potřeba připojit ho k internetu. To se dá provést buď drátově pomocí kabelu Ethernet anebo bezdrátově k Wi-Fi, protože síťová karta Raspberry Pi umožňuje bezdrátové připojení. Pak se provede stejná instalace softwaru, jako u klienta měřicího systému. Jediná změna bude u SSH, kterou by měl operační systém už obsahovat. To se ale musí nejdříve povolit v konfiguraci Raspberry Pi. Provede se to kliknutím na ikonu Raspberry Pi v levém horním rohu obrazovky (obdobu nabídky Start u operačního systému Windows), pak se myš najede na kolonku Preferences. Na pravé straně se objeví další menu v něm se klikne na kolonku Raspberry Pi Configuration. Otevře se okno se čtyřmi záložkami. Stačí kliknout na záložku Interfaces a u SSH překliknout z Disabled na Enabled a pak volbu potvrdit kliknutím na tlačítko OK v pravém dolním rohu okna.

Existuje mnoho dalších možností, jak povolit SSH jako je přidání souboru `ssh` do adresáře `boot`, použití nástroje `raspi-config` pomocí příkazu `sudo raspi-config` nebo nástroje `systemctl` pomocí příkazů `sudo systemctl enable ssh` a `sudo systemctl start ssh`. Tímto se dá považovat klient měřicího systému za připravený k použití a může se přejít k ověření funkčnosti. [46]

5.3 Ověření funkčnosti měřicího systému

Pro ověření funkčnosti měřicího systému bude stačit, když se oba dva konce připojí přes jakýkoliv router, u kterého známe jeho parametry (v tomto případě TP-LINK Archer VR300) nebo rovnou napřímo kabely standardu alespoň Fast Ethernet a Raspberry Pi se připojí k napájení. První je zapotřebí spustit časovou synchronizaci mezi oběma konci. Všechna obsluha bude probíhat na klientovi měřicího systému. Spustí se terminál a pomocí SSH se vzdáleně připojí k Raspberry Pi pomocí příkazu `ssh <uživatelské jméno vzdáleného systému>@<IP adresa vzdáleného systému>` (např. `ssh pi@10.0.20.1`). Pokud IP adresa není předem známá, dá se zjistit příkazem `ip addr` na serveru měřicího systému anebo na routeru, pokud je k němu server měřicího systému připojen. Bude následovat výzva k zadání hesla vzdáleného systému (výchozí heslo k Raspberry Pi je „raspberry“). Po zadání hesla by měl být klient měřicího systému úspěšně vzdáleně připojen k serveru měřicího systému. [46]

Poté se spustí software pro časovou synchronizaci v režimu slave pomocí příkazu `sudo ptp4l -P -S -i <název rozhraní Ethernet> -s -m`. Jednotlivé argumenty znamenají:

- P – režim peer-to-peer, který je přesnější než režim end-to-end, ale jde použít pouze při použití dvou systémů
- S – softwarové časové značkování, pro přesnější hardwarové značkování by musela být k notebooku připojena přesnější časová reference, což je pro tuto aplikaci přehnaně přesné
- i – volba rozhraní Ethernet pro časovou synchronizaci, tento název se dá nalézt pomocí příkazu `ip addr` (většinou je název rozhraní „eth0“)
- s – volba režimu slave
- m – zajistí výpis průběhu časové synchronizace do konzole

Tento software pro časovou synchronizaci používá jako výchozí protokol IPv4. Server měřicího systému teď bude čekat, až najde na síti systém, který by mohl použít jako referenci k časové synchronizaci. [38]

V novém terminálu provedeme stejné přihlášení ke vzdálenému systému, jaké bylo provedeno výše. V tomto terminálu poběží server měřicího softwaru `iperf2`, který se spustí pomocí příkazu `iperf -s -u -e -f m >><název souboru, do kterého se má uložit výstup>`. Jednotlivé argumenty znamenají:

- s – volba serverového režimu
- u – volba packetů UDP pro měření (jako výchozí se používají TCP)
- e – volba rozšířeného výstupu (jinak budou ve výstupu chybět údaje o jednosměrném zpoždění)

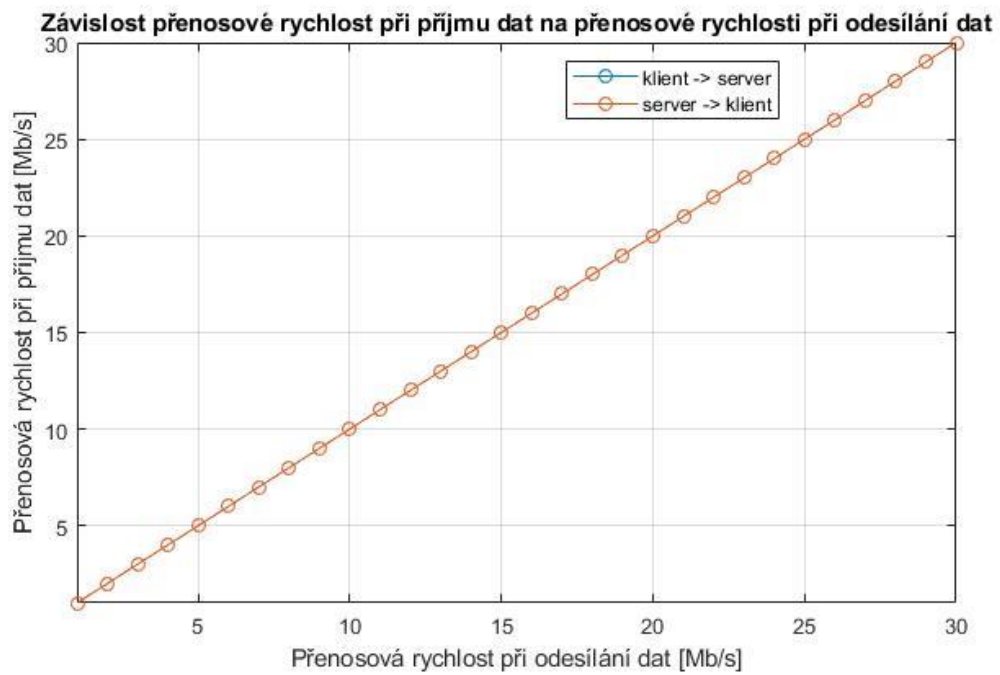
- f – volba formátu dat výstupu, argument m znamená, že budou číselné hodnoty přenosových rychlostí ve výstupu v megabitech (k by bylo pro kilobity, K by bylo pro kibibit, atd.)
- >> – volba ukládání výstupu měřicího softwaru do zadaného souboru, pokud název souboru neexistuje, vytvoří se, pokud není zadaná cesta k souboru, použije se domovský adresář (pouze jedno > by soubor při opakovaném použití příkazu vždy přepsalo, což je zde nežádoucí) [30]

Tímto je server měřicího systému připraven na měření. V dalším terminálové okně spustíme software pro časovou synchronizaci pomocí příkazu `sudo ptp4l -P -S -i <název rozhraní Ethernet> -m`. Název rozhraní Ethernet na tomto konkrétním notebooku je „eno1“. Notebook by se měl rozeznat jako nejlepší master pro časovou synchronizaci a po pár sekundách by se podle něj měl začít řídit server měřicího systému. Časová synchronizace by ze zkušenosti měla běžet alespoň 30 minut, aby se odchylka hodin dostala pod 1 ms. Toto je velká, ale nevyhnutelná nevýhoda měření jednosměrného zpoždění. [38]

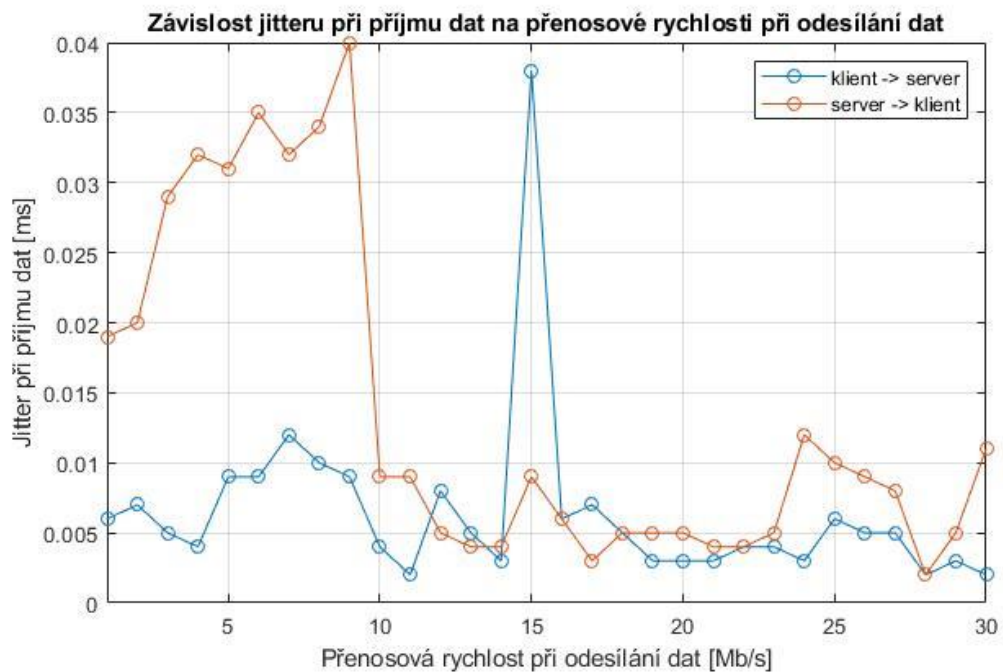
V poslední terminálu se spustí měřicí software pomocí příkazu `iperf -e -u -f m -r -b <hodnota požadované přenosové rychlosti v příslušném formátu> -c <IP adresa vzdáleného systému> >><název souboru, do kterého se má uložit výstup>`. Jednotlivé argumenty znamenají:

- r – volba provádění dalšího reverzního měření (aby se nemusel měřicí software spouštět na obou koncích v opačných režimech, značně to urychlí měření)
- b – volba hodnoty požadované přenosové rychlosti pro provedení měření, např. pro 1 Mb/s se zadá „1m“
- c – volba klientského režimu

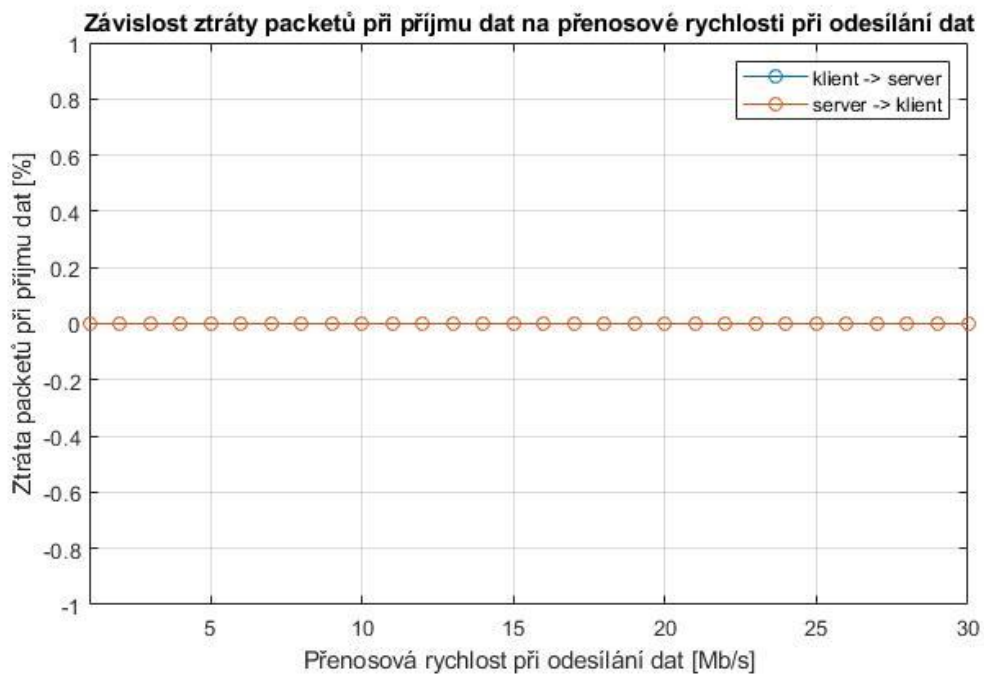
Měřicí software provede měření na zadané přenosové rychlosti po výchozí dobu 10 s. Tato doba se dá změnit pomocí argumentu `-t <doba měření v sekundách>`, ale bylo rozhodnuto, že měření dlouhé 10 s je dostatečné pro omezení nepřesností, které mohou na bezdrátové síti nastat (výsledky měření sám měřicí software průměruje). Měření bylo provedeno při přenosových rychlostech od 1 Mb/s do 30 Mb/s s krokem 1 Mb/s. [30] Pro budoucí obsáhlejší měření bude využito jednoduchého bash scriptu pro automatické provádění měření. Výsledky měření byly zpracovány v programu MATLAB pomocí skriptu, který bude popsán v následující kapitole. Toto jsou výsledné grafy:



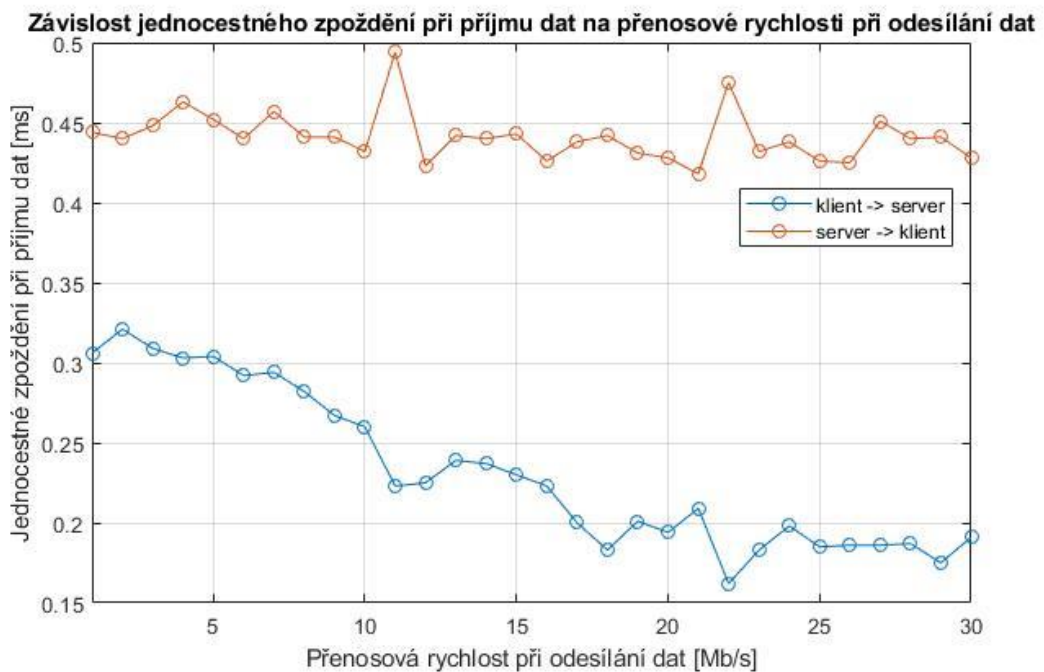
Obrázek 5.1 Graf závislosti přenosové rychlosti při příjmu dat na přenosové rychlosti při odesílání dat



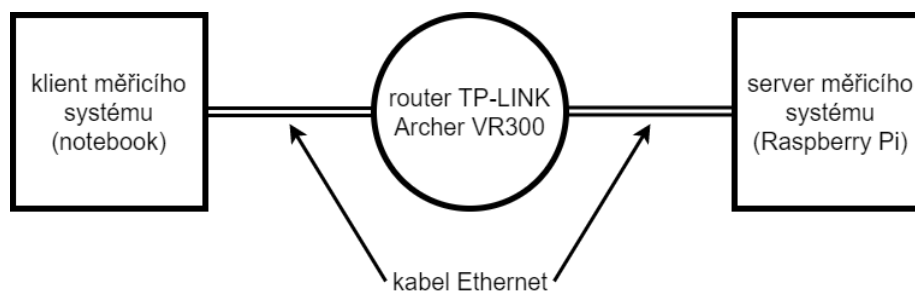
Obrázek 5.2 Graf závislosti jitteru na přenosové rychlosti při odesílání dat



Obrázek 5.3 Graf závislosti ztráty packetů na přenosové rychlosti při odesílání dat



Obrázek 5.4 Graf závislosti jednocestného zpoždění na přenosové rychlosti při odesílání dat



Obrázek 5.5 Schéma zapojení pro ověření funkčnosti měřicího systému

Z grafů lze vyčíst, že měření probíhá správně. Protože je tento vcelku moderní router určený pro domácnost, vytváření silné Wi-Fi sítě pro pokrytí celého domu a není nijak nakonfigurován pro omezování přenosových rychlostí, není vidět u přenosových rychlostí od 1 do 30 Mb/s žádná saturace a průběh závislosti přenosových rychlostí při příjmu dat na přenosových rychlostech při odesílání dat je lineární ($x = y$). Saturace by šla vidět až u přenosových rychlostí nad 100 Mb/s, protože by se projevila hranice fyzické vrstvy kabelu standardu Fast Ethernet použitého pro připojení k routeru (Fast Ethernet dokáže zvládnout přenosové rychlosti do 100 Mb/s). Tím pádem je ztráta paketů nulová. Jitter a jednocestné zpoždění je ve velmi malých hodnotách díky prakticky přímému spojení. U spojení klient \rightarrow server jde vidět postupné snižování jednocestného zpoždění s rostoucí přenosovou rychlostí při odesílání dat. Z toho lze odvodit, že přenosovému kanálu se v tomto směru špatně přenášejí data při malých přenosových rychlostech. K přeskupení paketů nedošlo. Tímto proběhlo ověření funkčnosti úspěšně a lze přejít k měření neznámého bezdrátového spoje.

6. IMPLEMENTACE AUTOMATIZOVANÉHO ZPRACOVÁNÍ A ZOBRAZENÍ NAMĚŘENÝCH DAT

Automatizované zpracování a zobrazení naměřených dat probíhá pomocí MATLAB skriptu (viz příloha), který se upravuje dle charakteru měření. Uživatel musí vložit seznam souborů s výstupy z měření. Ty by měly mít v názvu udány podmínky měření, např. „test-client-0-nv2-b-20.txt“, kde 0 značí vzdálenost od řídicí stanice, nv2 značí použitý wireless protokol, b značí použitý standard Wi-Fi (band) a 20 značí použitou šířku kanálu (channel width). Na oddělovačích a pořadí konfigurace prakticky nezáleží (definuje se libovolný oddělovač), ale musí být konzistentní ve všech názvech souborů, stejně jako text před podmínkami měření.

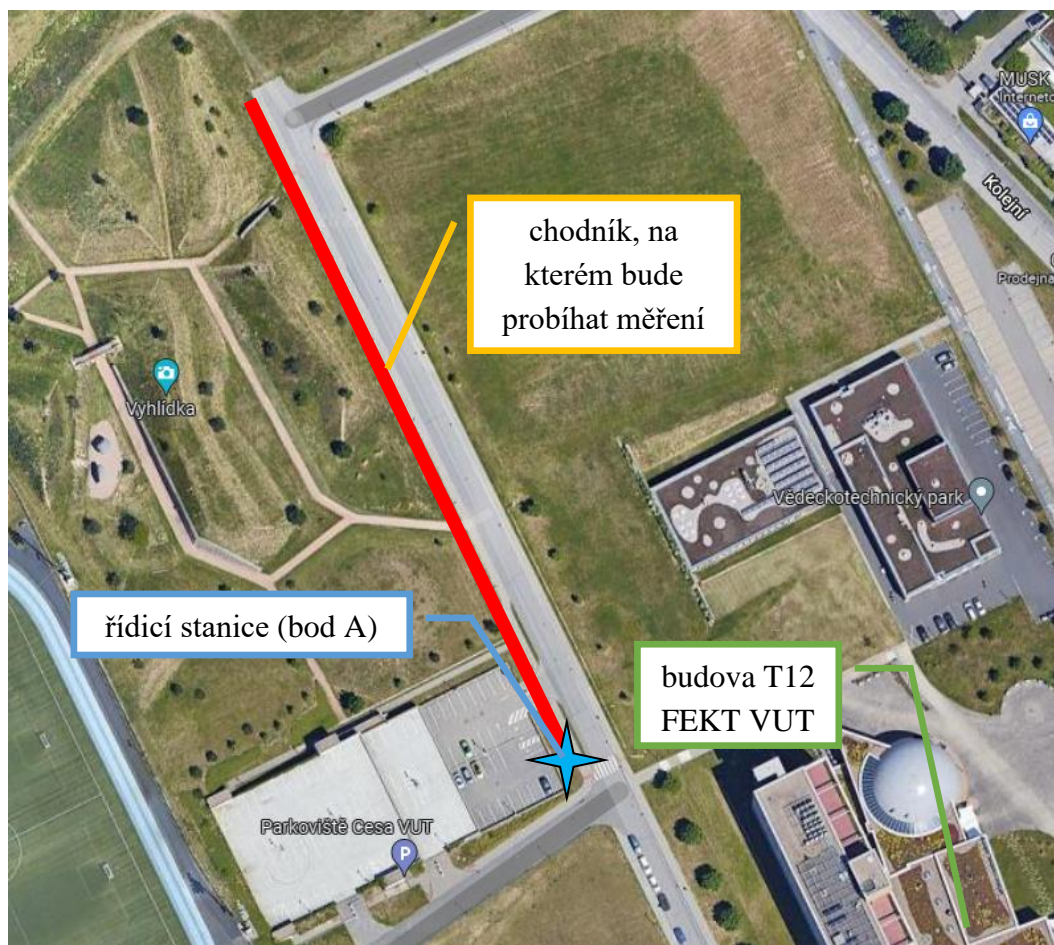
For smyčka pak pro každý soubor v seznamu provádí načítání dat do matice pomocí `readmatrix`, extrahování čistých dat o měření pomocí `contain` z této matice a jejich rozřazování podle směru komunikace. Dále se provádí vykreslování grafů, které se dají dle volby uživatele rozdělit např. dle směru komunikace, použitého wireless protokolu, atd. Legendu a popisy grafů musí uživatel individuálně měnit dle charakteru měření. Všechny vygenerované grafy jsou k nalezení v elektronické příloze bakalářské práce. Pokud by byly v příloze na konci bakalářské práce, počet stran by narostl o nejméně 56 (grafů je 112 a pro jejich čitelnost by musely být dva grafy na jednu stránku), což by bylo velmi nepraktické (zdvojnásobila by se tloušťka výtisku).

7. MĚŘENÍ NEZNÁMÉHO BEZDRÁTOVÉHO SPOJE A DOPORUČENÍ VHODNÉHO NASTAVENÍ

7.1 Měření na routerech MikroTik

7.1.1 Plán měření

Měření bude probíhat venku na chodníku, který se nachází mezi parkovištěm CESA VUT a budovou T12 FEKT VUT, aby se předešlo rušení způsobenou sítí eduroam v budově T12 FEKT VUT. Řídicí stanice s klientem měřicího systému bude umístěna před parkovištěm CESA VUT, aby z něj bylo zajištěno napájení ze zásuvky přivedeno pomocí prodlužovacího kabelu. Napájení by mohlo být při pohybu řídicí stanice snadno vyřešeno bateriemi, v tomto případě bude ale výhodnější použití prodlužovacího kabelu. Přijímač mobilního robotu se bude umísťovat na chodník (znázorněn v mapě červenou čarou) do určitých vzdáleností od řídicí stanice směrem k vyhlídce, čímž se zajistí téměř stejná výška od země řídicí stanice a přijímače mobilního robota.



Obrázek 7.1 Mapa oblasti, ve které probíhalo měření

Řídicí stanice, kterou zprostředkovává router MikroTik, se umístí na vhodné místo před parkoviště CESA VUT a zapojí se do elektřiny. K řídicí stanici se připojí kabely standardu alespoň Fast Ethernet klient měřicího systému a notebook s operačním systémem Windows a nainstalovaným softwarem WinBox (nejnovější verze 32 nebo 64bitová je dostupná zdarma ke stažení z oficiálních stránek MikroTik <https://mt.lv/winbox64>), pomocí kterého se bude měnit konfigurace řídicí stanice. Budou se měřit parametry bezdrátového přenosového kanálu v závislosti na různých konfiguracích řídicí stanice a různých vzdálenostech od řídicí stanice.

Bylo rozhodnuto, že se proměří parametry bezdrátového přenosového kanálu u bezdrátových protokolů nv2 a 802.11, pásmech 2GHz-only-b, g a n a šířek pásma u všech pásem 20 MHz a u n ještě 10 MHz, 5 MHz a 20/40 Ce MHz s pevným vysílacím výkonem 17 dBm. Poté se proměří účinek změny vysílacího výkonu na maximální dosažitelnou přenosovou rychlost s nulovou ztrátou packetů v různých vzdálenostech přijímače mobilního robota od řídicí stanice. Pokusnými měřeními byla určena maximální vzdálenost 100 m, do které se dokáží systémy dokomunikovat. Bylo rozhodnuto, že se budou měření provádět s krokem 25 m, tak aby se dosáhlo dostatečné množství dat pro vyhodnocení, ale aby měření byly časově zvládnutelné. Měření tedy budou probíhat ve vzdálenostech 100 m, 75 m, 50 m, 25 m a < 1 m. Toto bude simulovat různé vzdálenosti, kterých může mobilní robot při pohybu dosáhnout.

Tabulka 7.1 Plán měření pro různé vzdálenosti přijímače mobilního robota MikroTik od řídicí stanice a konfigurace routeru MikroTik

vzdálenost od řídicí stanice [m]	konfigurace routerů												
	wireless protokol												
	nv2						802.11						
	band 2GHz-only-												
	b	g	n				b	g	n				
	channel width [MHz]												
20	20	20	10	5	20/40 Ce	20	20	20	10	5	20/40 Ce		
< 1													
25													
50													
75													
100													

Přijímač mobilního robota zprostředkovaným sesterským routerem MicroTik se připojí kabelem standardu alespoň Fast Ethernet k Raspberry Pi a jejich napájení se bude řešit pomocí baterií (12 V a 6, 7 Ah). Napájení pro Raspberry Pi 4 Model B bude muset ještě procházet přes napěťový měnič, aby se převedlo z 12 V na 5 V. Nakonec se proměří závislost vzdálenosti doku komunikování a v ní maximální dosažitelné přenosové rychlosti na nastaveném výkonu sítě ve stejných vzdálenostech, které byly určeny pro předchozí měření a na jedné výchozí konfiguraci řídicí stanice.

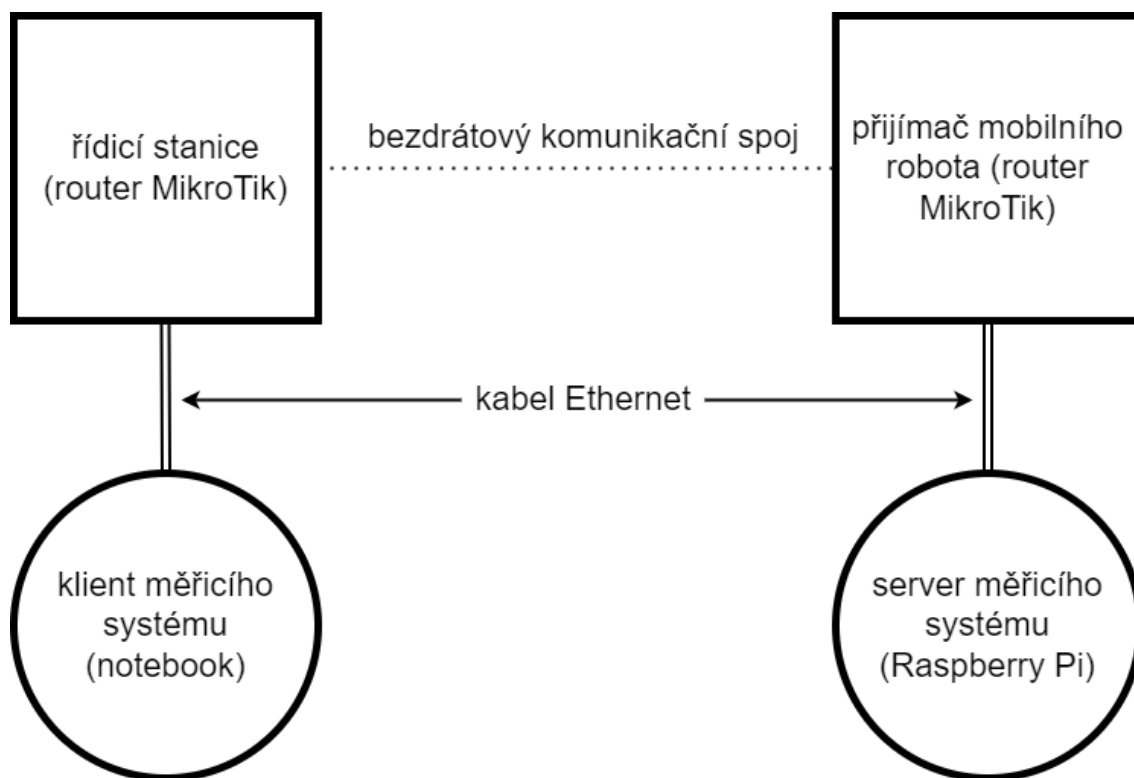
7.1.2 Průběh měření

Průběh měření bude identický jako u ověřování funkčnosti analyzátoru v kapitole 5. Jediné odlišnosti budou změny konfigurace řídicí stanice, změna vzdálenosti přijímací stanice mobilního robota od řídicí stanice a způsob provádění měření na klientovi měřicího systému. Přijímací stanice se serverem měřicího systému se umístí do příslušné vzdálenosti, ve které bude probíhat měření. Poté se budou provádět měření v příslušných konfiguracích řídicí stanice v dané vzdálenosti. Po změření parametrů bezdrátového přenosového kanálu ve všech konfiguracích řídicí stanice se přesune přijímací stanice se serverem měřicího systému do další vzdálenosti a měření se zopakuje, dokud nebudou proměřeny všechny možné kombinace. Pro zrychlení a zjednodušení měření se pro jejich provádění využije jednoduchého bash skriptu:

```
#!/bin/bash
ip=<IP adresa vzdáleného systému>
file="<název souboru, do kterého se má uložit výstup>"
for i in {<dolní hranice i>..<horní hranice i>..<krok i>}
do
iperf -e -u -f m -b $i"m" -c $ip -r >>$file
wait
done
```

Skript se vytvoří v jakémkoli textovém editoru a uloží se nejlépe do domovského adresáře (pro lepší přístup v terminálu) s příponou .sh. U tohoto souboru se ještě musí povolit jeho spouštění. To se provede tak, že se pravým kliknutím na soubor otevře vyskakovací menu a klikne se položku Properties (nebo označením souboru a stisknutím klávesové kombinace Alt+Enter). Otevře se okno s vlastnostmi souboru a vybere se záložka Permissions. V této záložce se musí zaškrtnou možnost Execute: Allow executing file as program. Jiná možnost je otevřít terminál v adresáři, který obsahuje daný skript (pokud je uložen v domovském adresáři, není co řešit) a zadat příkaz `chmod +x <jméno skriptu>.sh`.

Ve skriptu se definují 3 proměnné. Jsou to `ip`, kam se zadá IP adresa vzdáleného systému, `file`, kam se zadá název souboru, do kterého se má uložit výstup a `i`, která nabývá hodnot 1 – 30 s krokem 1 dle daného cyklu `for` smyčky a určuje velikost přenosové rychlosti odesílání v daném cyklu `for` smyčky. `for` smyčka provádí měření pomocí měřicího softwaru `iperf2` s přenosovými rychlostmi při odesílání dat 1 – 30 Mb/s (argument „m“) s krokem 1 Mb/s. Klíčové slovo `wait` zajišťuje provedení dalšího cyklu `for` smyčky až po skončení předchozího měření. [49]



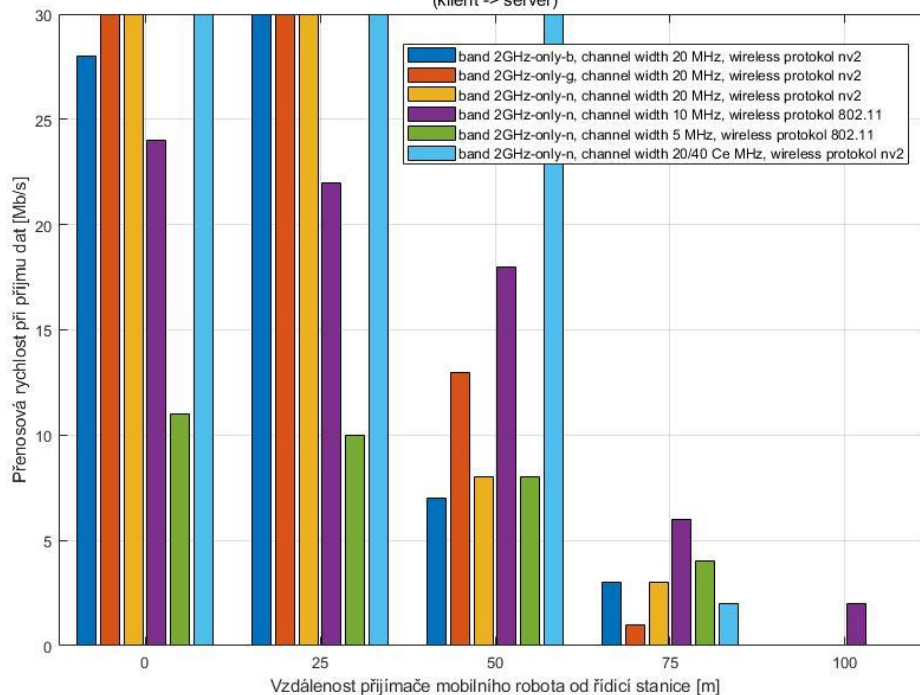
Obrázek 7.2 Schéma zapojení pro měření na routerech MikroTik

7.1.3 Výsledky měření

V grafech jsou uvedeny hodnoty přenosových rychlostí a jednocestných zpoždění, kdy ještě nedocházelo ke ztrátě packetů. Z grafů lze vyčíst, že nejlepší konfigurací pro řídicí stanici i přijímač mobilního robota je pro vzdálenosti přijímače mobilního robota do 50 m od řídicí stanice band 2GHz-only-n, channel width 20/40 Ce MHz, wireless protokol nv2. Toto odpovídá teoretickým předpokladům, protože širší kanál znamená schopnost bezdrátového komunikačního kanálu přenášet více dat za cenu dosahu. Je dobré si všimnout, že u kratších vzdáleností přijímače mobilního robota od řídicí stanice se dobře chovají i ostatní konfigurace s šířkou kanálu 20 Hz, ale jejich dosah není tak dobrý jako u šířky kanálu 20/40 Ce MHz (je to dáno také stářím Wi-Fi standardů b a g).

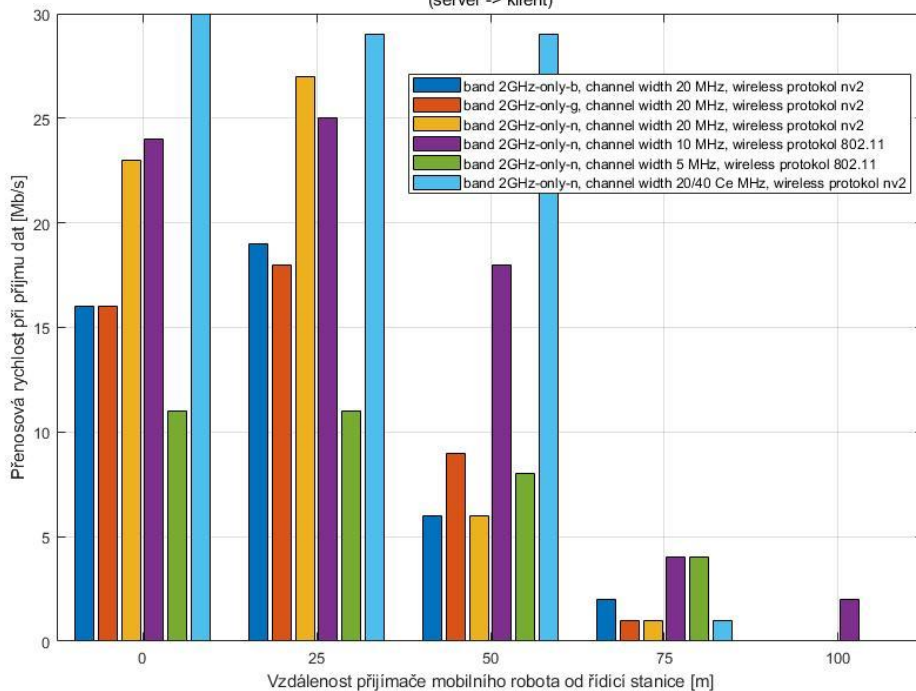
Pro vzdálenosti přijímače mobilního robota od řídicí stanice větší než 50 m, je nejvhodnější použití konfigurace pro řídicí stanici i přijímač mobilního robota band 2GHz-only-n, channel width 10 MHz, wireless protokol 802.11. Toto také odpovídá teoretickým předpokladům, protože snížením šířky kanálu se sice sníží maximální přenosová rychlost dat, ale vylepší se dosah bezdrátové komunikace. U šířky kanálu 5 MHz, se sice teoreticky dokomunikujeme dále než u šířky kanálu 10 MHz, ale přenosová rychlost je v tomto případě příliš omezená na to, aby mohla probíhat kvalitní bezdrátová komunikace. V grafech se občas objeví malá náhodná ztráta packetů, která se u bezdrátového přenosu dá očekávat, díky jeho náchylnosti k rušení.

Porovnání výsledků měření pro určení nejlepší konfigurace v dané vzdálenosti přijímače mobilního robota od řídicí stanice (klient -> server)



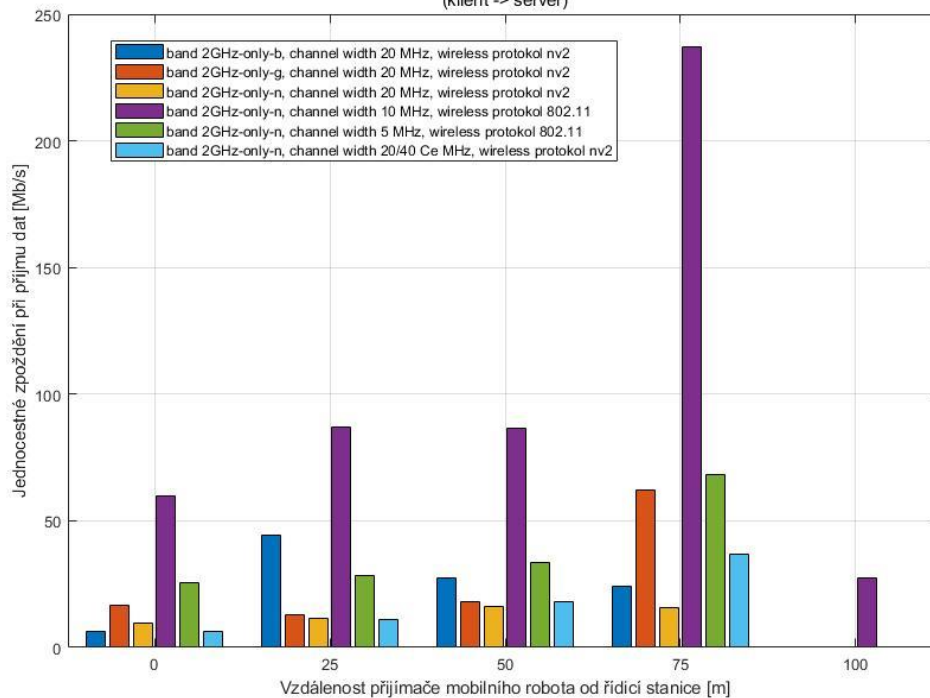
Obrázek 7.3 Graf porovnání výsledků měření (klient -> server)

Porovnání výsledků měření pro určení nejlepší konfigurace v dané vzdálenosti přijímače mobilního robota od řídicí stanice (server -> klient)



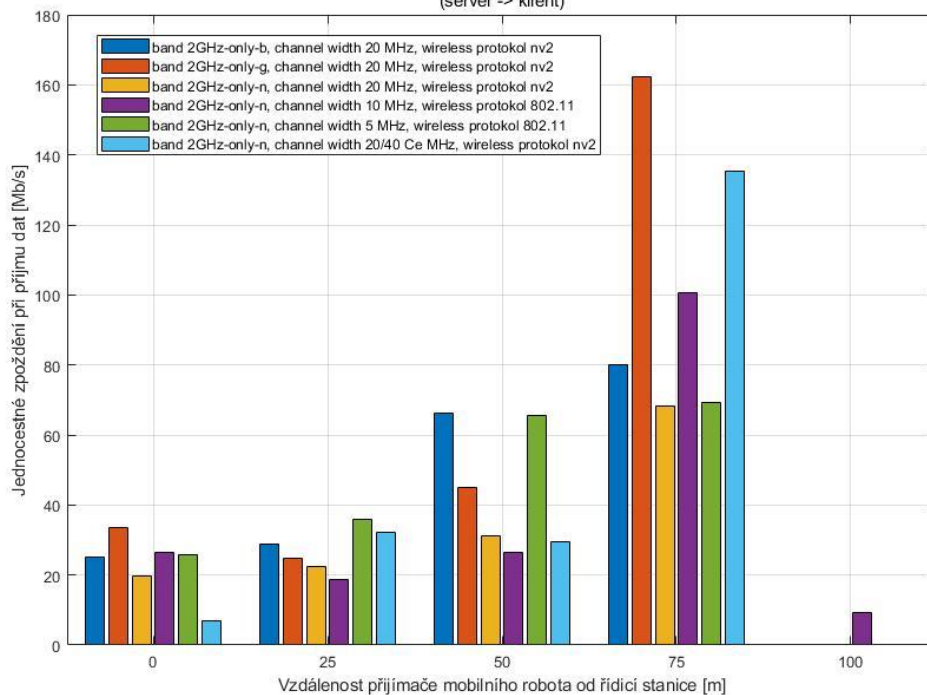
Obrázek 7.4 Graf porovnání výsledků měření (server -> klient)

Porovnání výsledků měření pro určení nejlepší konfigurace v dané vzdálenosti přijímače mobilního robota od řídicí stanice (klient -> server)



Obrázek 7.5 Graf porovnání výsledků měření (klient -> server)

Porovnání výsledků měření pro určení nejlepší konfigurace v dané vzdálenosti přijímače mobilního robota od řídicí stanice (server -> klient)



Obrázek 7.6 Graf porovnání výsledků měření (server -> klient)

U grafů jednocestného zpoždění lze vidět, že nejmenší jednocestné zpoždění má ve většině případů pro vzdálenosti přijímače mobilního robota do 50 m od řídicí stanice konfigurace routerů band 2GHz-only-n, channel width 20/40 MHz, wireless protokol nv2, do 75 m je to potom konfigurace routerů band 2GHz-only-n, channel width 20 MHz, wireless protokol nv2 a do 100 m je to konfigurace routerů band 2GHz-only-n, channel width 10 MHz, wireless protokol 802.11, jejíž jednocestné zpoždění je velmi malé, což asi zapříčinila malá přenosová rychlost odesílání dat, která se může na této konfiguraci routeru dosáhnout, při níž ještě nedochází k saturaci kanálu (inkrement přenosové rychlosti odesílání dat je příliš hrubý).

Další zajímavostí je, že šířky kanálu 20 MHz a výše se chovají lépe na wireless protokolu nv2 než na wireless protokolu 802.11. Na wireless protokolu 802.11 zase lépe fungují nižší šířky kanálu. Proto by se při potřebě delšího dosahu komunikace měl volit wireless protokol 802.11, ale pokud nejsou předpokládány velké vzdálenosti mobilního robota od řídicí stanice, je lepší zvolit wireless protokol nv2. Grafy přeskupení packetů nejsou uvedeny, protože k němu skoro vůbec nedochází. Objevuje se jen u velké ztráty packetů, která nastane jen u velkých přenosových rychlostí, a to jen velmi zřídka.

Tabulka 7.2 Maximální dosažitelná přenosová rychlost v Mb/s při příjmu dat bez ztráty packetů v různých vzdálenostech přijímače mobilního robota od řídicí stanice a vysílacích výkonech (klient -> server)

vzdálenost [m]	vysílací výkon [dBm]						
	30	25	20	15	10	5	0
< 1	30	30	30	30	30	30	30
25	26	23	30	21	15	5	4
50	11	11	10	3	2	1	-
75	1	1	1	1	1	-	-
100	1	1	1	1	-	-	-

Tabulka 7.3 Maximální dosažitelná přenosová rychlost při příjmu dat bez ztráty packetů v různých vzdálenostech přijímače mobilního robota od řídicí stanice a vysílacích výkonech (server -> klient)

vzdálenost [m]	vysílací výkon [dBm]						
	30	25	20	15	10	5	0
< 1	30	30	30	30	30	30	30
25	24	21	30	13	3	2	1
50	10	10	8	2	1	-	-
75	1	1	1	1	-	-	-
100	1	1	1	1	-	-	-

Z tabulek je patrné, že ve vzdálenosti přijímače mobilního robota od řídicí stanice < 1 m kvalita komunikace nezáleží na zvoleném vysílacím výkonu. Ve vzdálenosti přijímače mobilního robota od řídicí stanice 25 m se už projeví fakt, že více výkonu sítě není vždy lépe. Bezdrátový komunikační kanál zde funguje nejlépe při vysílacím výkonu 20 dBm. Při vzdálenostech přijímače mobilního robota od řídicí stanice vyšších než 50 m je pro dokomunikování třeba volit vysílací výkon alespoň 25 dBm.

Všechny tyto doporučení platí jak pro řídicí stanici, tak pro přijímač mobilního robota. Je ale důležité si všimnout, že i když jsou to párové routery, jejich síťové karty nejsou stejné. Řídicí stanice je ve všech stránkách výkonnější než její sesterský router, který zprostředkovával přijímač mobilního robota. Grafy jitteru nejsou zobrazeny, protože rozptyl hodnot byl příliš chaotický a z grafů by nešla poznat jakákoliv závislost. Grafy ztráty paketů nejsou zobrazeny, protože pokud jsou uvedeny již grafy přenosových rychlostí, budou mít nulovou vypovídací hodnotu.

7.2 Měření na radiomodemech

7.2.1 Plán měření

Plán měření na radiomodemech je téměř totožný s plánem měření na routerech MikroTik. Změny jsou následující:

- ke změně konfigurace radiomodemů se využívá webového rozhraní, a ne dedikovaného softwaru (jako WinBox), tudíž odpadá použití druhého notebooku s operačním systémem Windows a konfigurace se dá měnit na notebooku, který je využíván jako klient měřicího systému pomocí webového prohlížeče (zadáním IP adresy daného radiomodemu)
- konfigurace radiomodemů, které se budou proměřovat, byly zvoleny RF power PEP 20 a 30 dBm, modulation type 64QAM a 256QAM a FEC 2/3 a 5/6 (ostatní konfigurace zůstává výchozí)
- účinek změny vysílacího výkonu na maximální dosažitelnou přenosovou rychlost s nulovou ztrátou paketů v různých vzdálenostech přijímače mobilního robota od řídicí stanice se neměří
- maximální vzdálenost, do které se dokáží systémy dokomunikovat, byla určena jako 50 m a vzdálenosti přijímače mobilního robota od řídicí stanice ve kterých se bude měřit byly zvoleny jako 45 m, 30 m, 15 m a < 1 m

Tabulka 7.4 Plán měření pro různé vzdálenosti přijímače mobilního robota od řídicí stanice a konfigurace radiomodemů

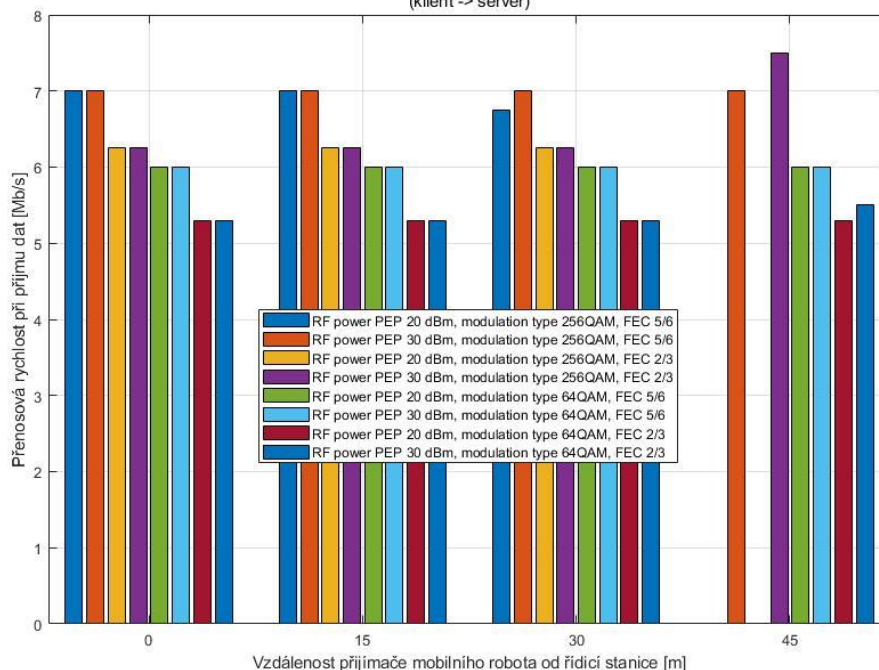
vzdálenost [m]	konfigurace radiomodemů							
	RF power PEP [dBm]							
	20				30			
	modulation type							
	64QAM		256QAM		64QAM		256QAM	
	FEC							
	2/3	5/6	2/3	5/6	2/3	5/6	2/3	5/6
< 1								
15								
30								
45								

7.2.2 Průběh měření

Průběh měření na radiomodemech je téměř totožný s průběhem měření na routerech MikroTik. Jediná změna je ve skriptu u proměnné i , která bude nabývat hodnot 1 – 10 Mb/s s krokem 0,5 Mb/s. Přenosové rychlosti odesílání dat nad 10 Mb/s není logické měřit, protože přinesou jen vyšší ztrátu packetů (bezdrátový přenosový kanál již dosáhl saturace a nedokáže přenášet více dat). Tato situace je podobná volbě horní hranice přenosové rychlosti odesílání dat 30 Mb/s u routerů MikroTik. Tam ale volbu ovlivnilo i to, aby byl vygenerovaný objem dat přehledně zpracovatelný do grafů.

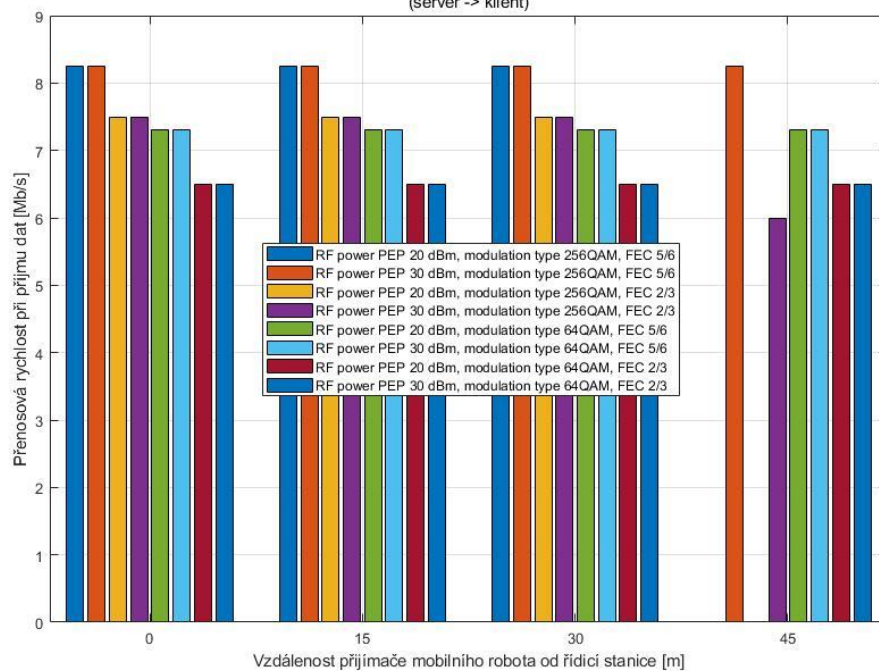
7.2.3 Výsledky měření

Porovnání výsledků měření pro určení nejlepší konfigurace v dané vzdálenosti přijímače mobilního robota od řídicí stanice (klient -> server)



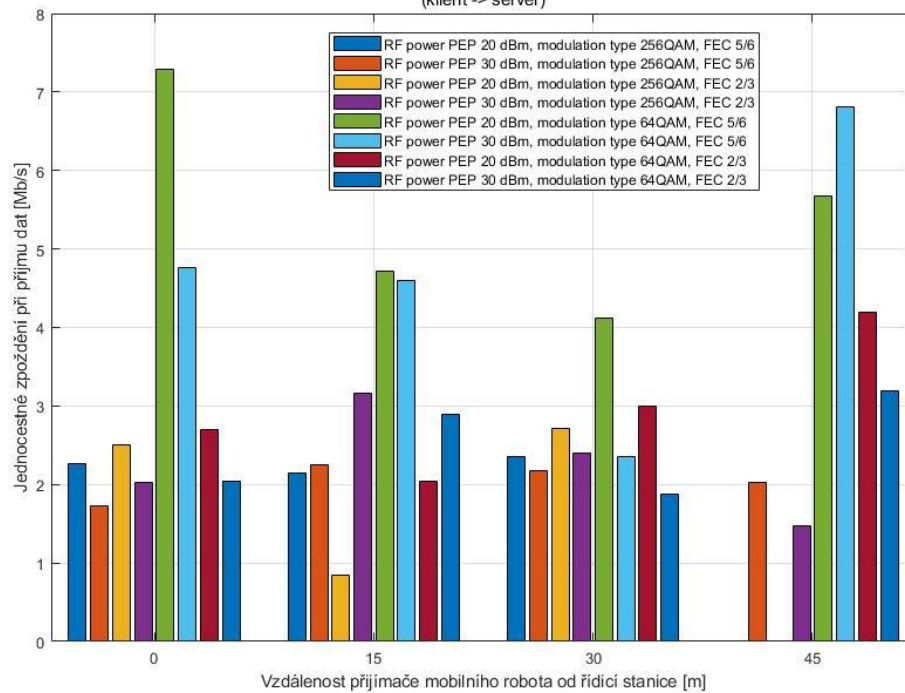
Obrázek 7.7 Graf porovnání výsledků měření (klient -> server)

Porovnání výsledků měření pro určení nejlepší konfigurace v dané vzdálenosti přijímače mobilního robota od řídicí stanice (server -> klient)



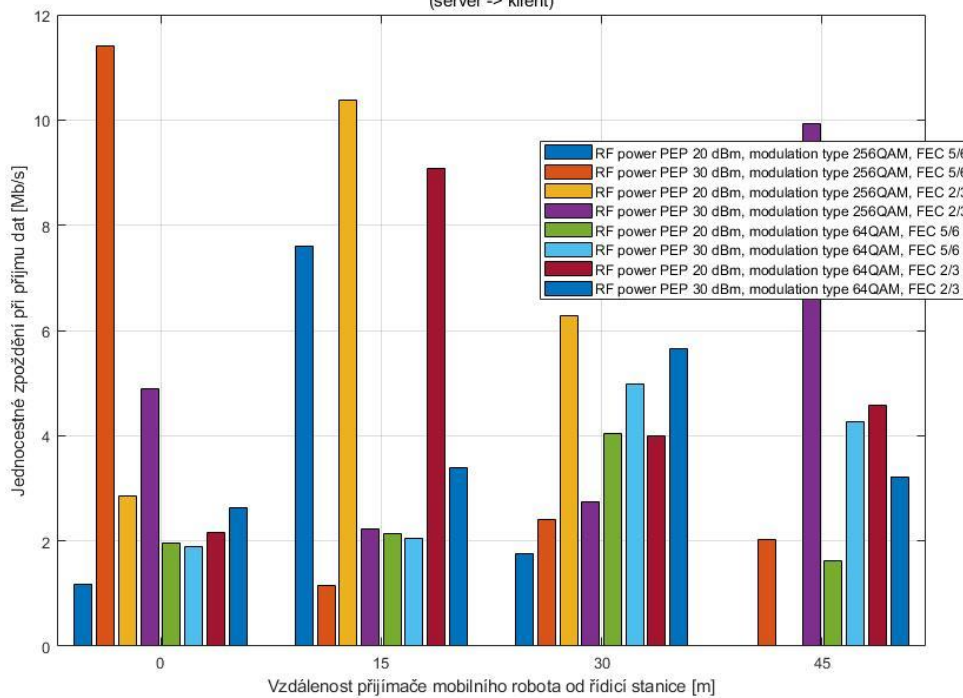
Obrázek 7.8 Graf porovnání výsledků měření (server -> klient)

Porovnání výsledků měření pro určení nejlepší konfigurace v dané vzdálenosti přijímače mobilního robota od řídicí stanice (klient -> server)



Obrázek 7.9 Graf porovnání výsledků měření (klient -> server)

Porovnání výsledků měření pro určení nejlepší konfigurace v dané vzdálenosti přijímače mobilního robota od řídicí stanice (server -> klient)



Obrázek 7.10 Graf porovnání výsledků měření (server -> klient)

V grafech jsou uvedeny hodnoty přenosových rychlostí odesílání dat, kdy došlo k saturaci bezdrátového přenosového kanálu a jednocestných zpoždění, kdy ještě nedocházelo ke ztrátě packetů. Z grafů lze vyčíst, že nejlepší konfigurací pro řídicí stanici i přijímač mobilního robota je RF power PEP 30 dBm, modulation type 256QAM, FEC 5/6. Ve vzdálenosti přijímače mobilního robota od řídicí stanice větší než 45 m, měla sice větší přenosovou rychlost odesílání dat konfigurace řídicí stanice RF power PEP 30 dBm, modulation type 256QAM, FEC 2/3, ale jen o 0,5 Mb/s, což není velký rozdíl.

Hodnoty jednocestných zpoždění měla konfigurace RF power PEP 30 dBm, modulation type 256QAM, FEC 5/6 většinou kolem 2 ms, až na případ vzdálenosti přijímače mobilního robota od řídicí stanice < 1 m, kdy vystoupala až k 12 ms, což naznačuje, že se při malých vzdálenostech přijímače mobilního robota od řídicí stanice nechová přijímač mobilního robota na této konfiguraci tak dobře, jak by mohl. Existují i konfigurace radiomodemů, při kterých bylo dosaženo menšího zpoždění, ale rozdíly mezi nimi jsou tak malé, aby změnili doporučení konfigurace radiomodemů.

Grafy jitteru nejsou zobrazeny, protože rozptyl hodnot byl příliš chaotický a z grafů by nešla poznat jakákoliv závislost. Grafy ztráty packetů nejsou zobrazeny, protože pokud jsou uvedeny již grafy přenosových rychlostí, budou mít nulovou vypovídací hodnotu. Grafy přeskupení packetů nejsou uvedeny, protože k němu skoro vůbec nedochází. Objevuje se jen u velké ztráty packetů, která nastane jen u velkých přenosových rychlostí, a to jen velmi zřídka. Radiomodem se ukázal o dost přesnější než router MikroTik a vykazoval mnohem lepší parametry i když za cenu nižších přenosových rychlostí odesílání dat a dosahu bezdrátové komunikace.

Měření kapacity bufferů

Pro radiomodem ještě proběhlo měření kapacity bufferů. Toto měření probíhá na principu odeslání burstu dat naráz, na rozdíl od pozvolného odesílání dat jako v předchozích testech. Měří se kapacita bufferů datová a kapacita bufferů packetová. Tzn., že buffery dokážou udržet buď určitý objem dat nebo packetů. Přesná velikost bufferu se dá těžko změřit. Spíše se sleduje hranice objemu odeslaných dat nebo packetů, kde se projeví ztráta packetů, ze které se pak dá odhadnout přibližná velikost bufferu.

Měření proběhlo standardně, jen se změnil argumenty v příkazu pro spuštění klienta měřicího systému. Použil se příkaz `iperf -e -u -isochronous=1:<počet packetů pro odeslání>p -l <velikost packetu>k -c <IP adresa vzdáleného systému>`. Argument `-isochronous=1:<počet packetů pro odeslání>p`, zajistí, že se vyše burst zadaného počtu packetů o zadané velikosti `l` v kB každou sekundu trvání testu.

Tím pádem se první najde hranice packetové kapacity bufferu postupným zvyšováním počtu odesílaných packetů s pevnou velikostí (např. 1 kB) a poté se najde hranice datového objemu bufferu postupným zvyšováním velikosti odesílaných packetů s pevnou hodnotou počtu odesílaných packetů (např. 10). Celé měření probíhá principem pokus omyl, dokud se neobjeví ztráta packetů, která bude signalizovat přetečení bufferu.

Ztráta packetů se objevila při odeslání více než 33 packetů o velikosti 1 kb a při odeslání více než 44 kB dat (10 packetů s velikostí 4,4 kB). Tudíž lze velikost bufferu radiomodemu odhadnout na 33 packetů a 44 kB.

8. ZÁVĚR

Z výsledků rešerše je patrné, že existuje pouze jeden vhodný měřicí software pro potřeby této bakalářské práce, a to měřicí software iPerf2. Byl navržen analyzátor bezdrátového komunikačního spoje, který pracuje na principu klient – server a k měření využívá UDP, rozhraní Ethernet, IPv4 a měří přenosovou rychlost, jitter, ztrátu packetů, jednocestné zpoždění a přeskupení packetů v obou směrech komunikace. Jako server měřicího systému se použilo Raspberry Pi 4 Model B, aby se mohl pohybovat s mobilním robotem a mohl být napájen z baterie. Pro klienta měřicího systému se použil notebook. Procesy měření byly zautomatizovány pomocí bash skriptu a vyhodnocování a zobrazování naměřených dat pomocí MATLAB skriptu. Funkčnost analyzátoru byla ověřena na domácím routeru TP-LINK Archer VR300.

Dále se proměřovaly parametry routeru MikroTik a radiomodemu v závislosti na jejich různých konfiguracích a vzdálenostech řídicí stanice od přijímače mobilního robota. Měření probíhalo venku za budovou T12 FEKT VUT před parkovištěm CESA VUT, aby se předešlo parazitním vlivům rušení sítě eduroam uvnitř a v bezprostřední blízkosti budovy T12 FEKT VUT. Vygenerovaných grafů bylo příliš a zdvojnásobily by obsah bakalářské práce, proto jsou uvedeny v elektronické příloze bakalářské práce. Výsledná data byla agregována do přehledných sloupcových grafů, ze kterých jsou patrné nejlepší konfigurace pro měření router MikroTik a radiomodem.

Pro router MikroTik byla při vzdálenostech řídicí stanice od přijímače mobilního robota do 50 m doporučena konfigurace band 2GHz-only-n, channel width 20/40 Ce MHz, wireless protokol nv2, do 75 m konfigurace band 2GHz-only-n, channel width 20 MHz, wireless protokol nv2 a do 100 m konfigurace band 2GHz-only-n, channel width 10 MHz, wireless protokol 802.11. Pro radiomodem byla doporučena konfigurace RF power PEP 30 dBm, modulation type 256QAM, FEC 5/6.

LITERATURA

- [1] PRODUCTS. *IT Management Software & Remote Monitoring Tools / SolarWinds* [online]. Austin: SolarWinds, 2021 [cit. 2021-12-30]. Dostupné z: <https://www.solarwinds.com/>
- [2] NetCPS. *NetChain Communications* [online]. San Diego: NetChain Communications, 2010, February 1998 [cit. 2021-12-30]. Dostupné z: <https://www.netchain.com/netcps/>
- [3] Care and Feeding of Netperf 2.7.X. *The Netperf Homepage* [online]. Palo Alto: Hewlett-Packard, 2015 [cit. 2021-12-30]. Dostupné z: <https://hewlettpackard.github.io/netperf/>
- [4] Test Traffic Measurement Service (TTM). *RIPE Network Coordination Centre* [online]. Amstrdam: Réseaux IP Européens Network Coordination Centre, 2010 [cit. 2021-12-30]. Dostupné z: <https://www.ripe.net/analyse/archived-projects/ttm>
- [5] UDPMon. *UDPMon Home Page* [online]. Manchester: Richard Hughes-Jones, 2021 [cit. 2021-12-30]. Dostupné z: <https://www.hep.manchester.ac.uk/u/rich/net/index.html>
- [6] TCPMon. *TCPMon Home Page* [online]. Manchester: Richard Hughes-Jones, 2011 [cit. 2021-12-30]. Dostupné z: https://www.hep.manchester.ac.uk/u/rich/Tools_Software/tcpmon.html
- [7] Bandwidth (computing) - Wikipedia. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-12-30]. Dostupné z: [https://en.wikipedia.org/wiki/Bandwidth_\(computing\)](https://en.wikipedia.org/wiki/Bandwidth_(computing))
- [8] Shannon–Hartley theorem - Wikipedia. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-12-30]. Dostupné z: https://en.wikipedia.org/wiki/Shannon%E2%80%93Hartley_theorem
- [9] Channel capacity - Wikipedia. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-12-30]. Dostupné z: https://en.wikipedia.org/wiki/Channel_capacity
- [10] Bit rate - Wikipedia. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-12-30]. Dostupné z: https://en.wikipedia.org/wiki/Bit_rate
- [11] Throughput - Wikipedia. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-12-30]. Dostupné z: <https://en.wikipedia.org/wiki/Throughput>
- [12] Goodput - Wikipedia. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-12-30]. Dostupné z: <https://en.wikipedia.org/wiki/Goodput>
- [13] End-to-end delay - Wikipedia. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-12-30]. Dostupné z: https://en.wikipedia.org/wiki/End-to-end_delay

- [14] Round-trip delay - Wikipedia. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-12-30]. Dostupné z: https://en.wikipedia.org/wiki/Round-trip_delay
- [15] Packet loss - Wikipedia. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-12-30]. Dostupné z: https://en.wikipedia.org/wiki/Packet_loss
- [16] Out-of-order delivery - Wikipedia. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-12-30]. Dostupné z: https://en.wikipedia.org/wiki/Out-of-order_delivery
- [17] MORTON, Al, Len CIAVATTONI, Gomathi RAMACHANDRAN, Stanislav SHALUNOV a Jerry PERSER. Packet Reordering Metrics. *RFC 4737* [online]. 2006 [cit. 2022-12-30]. Dostupné z: doi:10.17487/RFC4737
- [18] BRADNER, Scott, ed. Benchmarking Terminology for Network Interconnection Devices. *RFC 1242* [online]. 1991 [cit. 2022-12-30]. Dostupné z: doi:10.17487/RFC1242
- [19] HUGHES-JONES, Richard. Writeup for udpmon: A Network Diagnostic Program. *UDPMon Home Page* [online]. Manchester: The University of Manchester, 2010, October 2010 [cit. 2021-12-30]. Dostupné z: https://www.hep.manchester.ac.uk/u/rich/Tools_Software/udpmon/udpmon_v2.pdf
- [20] NetFlow Analyzer - Analyze Remote Network Bandwidth Traffic | SolarWinds. *SolarWinds* [online]. Austin: SolarWinds, 2021, Říjen 2010 [cit. 2021-12-30]. Dostupné z: <https://www.solarwinds.com/-/media/solarwinds/swdcv2/licensed-products/netflow-traffic-analyzer/images/product-screenshots/nta-create-flow-alert.aspx?rev=e9775522151848a59695e07e9efe8444>
- [21] Troubleshooting Client Speed Using IPerf - Ruijie.info. *Ruijie.info* [online]. Ruijie.info, 2021, 2021-07-16 [cit. 2021-12-30]. Dostupné z: <https://ruijie.info/wp-content/uploads/2021/07/1.png>
- [22] NetCPS. *NetChain Communications* [online]. San Diego: NetChain Communications, 2010, February 1998 [cit. 2021-12-30]. Dostupné z: <https://www.netchain.com/netcps/netcps.gif>
- [23] Netperf – networking benchmark software. *LinuxLinks* [online]. LinuxLinks, 2021 [cit. 2021-12-30]. Dostupné z: <https://i2.wp.com/www.linuxlinks.com/wp-content/uploads/2018/01/Screenshot-netperf.jpg?w=573&ssl=1>
- [24] Bandwidth (signal processing) - Wikipedia. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-12-30]. Dostupné z: [https://en.wikipedia.org/wiki/Bandwidth_\(signal_processing\)](https://en.wikipedia.org/wiki/Bandwidth_(signal_processing))
- [25] Link aggregation - Wikipedia. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-12-30]. Dostupné z: https://en.wikipedia.org/wiki/Link_aggregation

- [26] RIPE NCC - Wikipedia. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-12-30]. Dostupné z: https://en.wikipedia.org/wiki/RIPE_NCC
- [27] Client–server model - Wikipedia. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-12-30]. Dostupné z: https://en.wikipedia.org/wiki/Client%E2%80%93server_model
- [28] Raspberry Pi 4 Model B - 4GB RAM. *RPishop.cz* [online]. Roudné: RPishop.cz, 2021 [cit. 2021-12-30]. Dostupné z: <https://rpishop.cz/raspberry-pi-4b/1598-raspberry-pi-4-model-b-4gb-ram-765756931182.html>
- [29] Iperf - Wikipedia. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-05-08]. Dostupné z: <https://en.wikipedia.org/wiki/Iperf>
- [30] Manpage of IPERF. *SourceForge* [online]. San Diego (CA): Slashdot Media, 2022 [cit. 2022-05-08]. Dostupné z: <https://iperf2.sourceforge.io/iperf-manpage.html>
- [31] Iperf 2 & Iperf 3 Comparison Table. *SourceForge* [online]. San Diego (CA): Slashdot Media, 2022 [cit. 2022-05-08]. Dostupné z: <https://iperf2.sourceforge.io/IperfCompare.html>
- [32] Iperf3. *ESnet* [online]. Berkeley (CA): Lawrence Berkeley National Laboratory Energy Sciences Network, 2022 [cit. 2022-05-08]. Dostupné z: <https://software.es.net/iperf/#>
- [33] PerfSONAR. *ESnet* [online]. Berkeley (CA): Lawrence Berkeley National Laboratory Energy Sciences Network, 2022 [cit. 2022-05-08]. Dostupné z: <https://www.perfsonar.net/index.html>
- [34] Diagram. In: *PerfSONAR* [online]. Berkeley (CA): Lawrence Berkeley National Laboratory Energy Sciences Network, 2022 [cit. 2022-05-08]. Dostupné z: https://docs.perfsonar.net/_images/intro_about-arch.png
- [35] TCP vs. UDP: What's the Difference?. *Livesize* [online]. Austin: Livesize, 2022 [cit. 2022-05-08]. Dostupné z: <https://www.livesize.com/en/blog/tcp-vs-udp/>
- [36] Network Time Protocol - Wikipedia. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-05-08]. Dostupné z: https://en.wikipedia.org/wiki/Network_Time_Protocol
- [37] Precision Time Protocol - Wikipedia. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-05-08]. Dostupné z: https://en.wikipedia.org/wiki/Precision_Time_Protocol
- [38] Configuring PTP Using ptp4l. *Fedora Docs* [online]. Raleigh: Fedora Docs, 2022 [cit. 2022-05-08]. Dostupné z: https://docs.fedoraproject.org/en-US/fedora/latest/system-administrators-guide/servers/Configuring_PTP_Using_ptp4l/
- [39] Secure Shell - Wikipedia. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-05-08]. Dostupné z: https://en.wikipedia.org/wiki/Secure_Shell

- [40] Ubuntu - Wikipedia. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-05-08]. Dostupné z: <https://en.wikipedia.org/wiki/Ubuntu>
- [41] Raspberry Pi OS - Wikipedia. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-05-08]. Dostupné z: https://en.wikipedia.org/wiki/Raspberry_Pi_OS
- [42] APT (software) - Wikipedia. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-05-08]. Dostupné z: [https://en.wikipedia.org/wiki/APT_\(software\)](https://en.wikipedia.org/wiki/APT_(software))
- [43] sudo - Wikipedia. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-05-08]. Dostupné z: <https://en.wikipedia.org/wiki/Sudo>
- [44] Service - OpenSSH. *Ubuntu* [online]. London: Canonical Group Limited, 2022 [cit. 2022-05-08]. Dostupné z: <https://ubuntu.com/server/docs/service-openssh>
- [45] Raspberry Pi OS. *Raspberry Pi* [online]. Cambridge: Raspberry Pi Foundation, 2022 [cit. 2022-05-08]. Dostupné z: <https://www.raspberrypi.com/software/>
- [46] Raspberry Pi Documentation - Remote Access. *Raspberry Pi* [online]. Cambridge: Raspberry Pi Foundation, 2022 [cit. 2022-05-08]. Dostupné z: <https://www.raspberrypi.com/documentation/computers/remote-access.html>
- [47] Raspberry Pi 4 - 5V/3A černý - Napájecí adaptér. *Alza* [online]. Praha: Alza.cz, 2022 [cit. 2022-05-08]. Dostupné z: https://www.alza.cz/raspberry-pi-4-5v-3a-cerny-d5662379.htm?kampan=adw4_prislusenstvi-pro-mt_pla_all_obecnacenove-hity_cenove-hity_c_1003744___RK104fa2_598387784353_~136008501519~&gclid=CjwKCAjwj42UBhAAEiwACIhADsroZNNwvaN1Me5pKwrB0SCGeusD-lGdZxofagjHUYKXkIwrGhIQ6RoCamsQAvD_BwE
- [48] Install Ubuntu desktop. *Ubuntu* [online]. London: Canonical Group Limited, 2022 [cit. 2022-05-08]. Dostupné z: <https://ubuntu.com/tutorials/install-ubuntu-desktop#1-overview>
- [49] Bash Scripting Tutorial - Linux Tutorials - Learn Linux Configuration. *LinuxConfig* [online]. Sydney: TOSID Group Pty, 2022 [cit. 2022-05-08]. Dostupné z: <https://linuxconfig.org/bash-scripting-tutorial>

SEZNAM ZKRATEK

LAN	Local Area Network
IPv4	Internet Protocol version 4
IP	Internet Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
SCTP	Stream Control Transmission Protocol
IPv6	Internet Protocol version 6
BSD	Berkeley Software (Standard) Distribution
RIPE NCC	Réseaux IP Européens Network Coordination Centre
CPU	Central processing unit
ISO/OSI	International Organization for Standardization/Open Systems Interconnection
IEEE	Institute of Electrical and Electronics Engineers
RAM	Random-access memory
ARM	Advanced (Acorn) RISC (Reduced Instruction Set Computer) Machines
GCC	GNU Compiler Collection
JSON	JavaScript Object Notation
ESnet	Energy Sciences Network
CSV	Comma Separated Values
NTP	Network Time Protocol
PTP	Precision Time Protocol
BMC	Best Master Clock
USB	Universal Serial Bus
HP	Hewlett Packard
DDR3	Double Data Rate 3
sudo	substitute user, do
apt	advanced package tool
SD	Secure Digital
HDMI	High-Definition Multimedia Interface
VUT	Vysoké učení technické
FEKT	Fakulta elektrotechniky a komunikačních technologií
CESA	Centrum sportovních aktivit
PEP	Peak envelope power
RF	Radio Frequency
FEC	Forward error correction