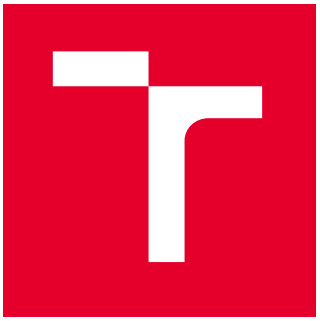


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

MULTIMEDIÁLNÍ PŘENOSY V IP SÍTÍCH

MULTIMEDIA TRANSMISSIONS IN IP NETWORKS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — **Bc. Tomáš Kejík**

AUTHOR

VEDOUCÍ PRÁCE — **Ing. David Grenar**

SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Tomáš Kejík ***ID:*** 211258

Ročník: 2 ***Akademický rok:*** 2022/23

NÁZEV TÉMATU:

Multimediální přenosy v IP sítích

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je provést teoretický rozbor problematiky multimediálních přenosů v IP sítích a vliv jejich klíčových přenosových parametrů. Důraz bude kladen na využití nástrojů s otevřeným zdrojovým kódem.
Výstupem práce bude vytvoření funkční testovací sady nástrojů pro měření jednotlivých přenosových síťových parametrů. Sada nástrojů bude využívat skriptovacího jazyku Bash. Součásti řešení práce bude integrace do uceleného uživatelského rozhraní v konzolovém prostředí.

DOPORUČENÁ LITERATURA:

[1] SPORTACK, Mark A. Směrování v sítích IP: [autorizovaný výukový průvodce: samostudium: kompletní zdroj informací o směrování a protokolech v sítích IP]. Brno: Computer Press, 2004. Cisco systems. ISBN 80-25-0127-4.
[2] LAFATA, Pavel a Jiří VODRÁŽKA. Optické přístupové sítě a přípojky FTTx. Praha: České vysoké učení technické v Praze, 2014. ISBN 978-800-1054-635.

Termín zadání: 6.2.2023 ***Termín odevzdání:*** 19.5.2023

Vedoucí práce: Ing. David Grenar

prof. Ing. Jiří Mišurec, CSc.
předseda rady studijního programu

ABSTRAKT

Tématem diplomové práce jsou „Multimediální přenosy v IP sítích". Cílem práce bylo provést teoretický rozbor problematiky multimediálních přenosů v IP sítích a vytvořit základní testovací sadu pro generování a měření síťového přenosu. Teoretická část je věnována základním typům přenosu IP videa a způsobům vysílání v IP sítích. Dále jsou popsány protokoly pro řízení nebo distribuci multimediálního obsahu. Praktická část se zabývá tvorbou skriptu, jenž byl napsán v jazyce Bash. Testovací skript obsahuje sadu funkcí, které jsou tvořeny různými síťovými nástroji. Pro automatickou a pohodlnou instalaci všech potřebných nástrojů byl vytvořen instalační Bash skript. Za účelem generování multimediálního a datového přenosu byly použity příkazy vlc a nc. Úprava multimediálních dat byla řešena nástrojem ffmpeg a k grafickému zobrazení změřených dat jsou využity nástroje smag a youplot. V neposlední řadě je také porovnávána účinnost vybraných kompresních formátů v závislosti na rychlosti přehrávání jednodeskového počítače Raspberry Pi 4B.

KLÍČOVÁ SLOVA

Bash, datový přenos, IP síť, monitorování sítě, multimediální přenos, RTP, testovací sada.

ABSTRACT

The topic of the master's thesis is "Multimedia transmissions in IP networks". The goal of the thesis was to perform a theoretical analysis of multimedia transmissions in IP networks and to create a basic test set for generating and measuring network traffic. The theoretical part is devoted to the basic types of IP video transmission and the methods of transmitting in IP networks. Furthermore, protocols for controlling or distributing multimedia content are described. The practical part deals with creating the script written in the Bash language. The test script contains a set of functions that are created by different network tools. A Bash installation script has been created to install all the necessary tools automatically and conveniently. In order to generate multimedia and data traffic, the vlc and nc commands were used. The multimedia data editing was handled by the ffmpeg tool and to graphically display the measured data the smag and youplot tools were used. Additionally, the efficiency of the selected compression formats depending on the playback speed of the Raspberry Pi 4B single-board computer was also compared.

KEYWORDS

Bash, data transmission, IP network, network monitoring, multimedia transmission, RTP, test set.

KEJÍK, Tomáš. *Multimediální přenosy v IP sítích*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2023, 94 s. Diplomová práce. Vedoucí práce: Ing. David Grenar

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Bc. Tomáš Kejík
VUT ID autora:	211258
Typ práce:	Diplomová práce
Akademický rok:	2022/23
Téma závěrečné práce:	Multimediální přenosy v IP sítích

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/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Davidu Grenarovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Dále bych rád poděkoval rodině za velkou podporu a zejména za dlouholetou trpělivost, kterou se mnou měla.

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů

Úvod

Život bez Internetu si již málokdo dokázal představit. Tento technický vynález se stal běžnou součástí našich životů. Každým rokem se zvyšují nároky na kvalitu síťového připojení. V dnešní době jsou klasické satelitní nebo kabelové televizní přenosy postupně nahrazovány přenosy přes IP sítě. IP sítě mají vybudovanou strukturu po celém svět a nabízí tak široké možnosti multimediálních služeb.

Hlavním cílem této práce je teoretický rozbor problematiky multimediálních přenosů v IP sítích, charakterizace základních typů multimediálních přenosů a vytvoření základního testovacího scénáře. Testovací sada je napsána v jazyce Bash a umožňuje automatická spouštění různých typů přenosů tak, aby to, co nejvíce odpovídalo reálné implementaci. Pro tyto účely byly použity různé síťové nástroje, které nabízí širokou škálu možností a vlastností. Pomocí těchto generátorů mohou být uskutečněny a následně otestovány přenosové parametry, jež zásadně ovlivňují kvalitu doručování.

Úvodní kapitola práce je věnována multimediálním službám v IP sítích. V úvodu kapitoly jsou popsány základní informace týkající se této problematiky, požadavky na kvalitní přenos přes nespolehlivou IP síť, dále současné i budoucí trendy v této oblasti. Následuje podkapitola s názvem Internet Protocol, která se týká kritického fungování v rámci Internetu. Na ni navazuje stěžejní část, a to samotná definice různých druhů multimediálních přenosů. Dále jsou porovnávány vlastnosti těchto přenosů. Druhá teoretická kapitola popisuje základní typy přenosů paketů, důraz je kladen na teoretický rozbor vícesměrového vysílaní a jeho náležitostí. Pro zajištění správného fungování při multicastovém vysílání jsou důležité protokoly, jež zajišťují tvorbu distribučních přenosových stromů, a dále protokol umožňující registraci účastníků do skupiny. Třetí kapitola pojednává o distribuci multimediálního obsahu. Jsou rozebrány dílčí protokoly, které umožňují řízení nebo přenos multimediálních dat. Čtvrtá kapitola obsahuje teoretický rozbor klíčových přenosových parametrů, možné příčiny vzniku chyb a parametry ovlivňující bezdrátový přenos. Poslední teoretická kapitola se nazývá Digitální videa. Tato kapitola se věnuje přehledu běžně používaných rozlišení u videa a popisuje kompresní formáty. Následující kapitola se již týká praktické části práce. Nejdříve jsou popsány základní hardwarové specifikace jednodeskového počítače, který je základním zařízením pro tuto práci. Další kapitola představuje síťovou topologii a síťové nástroje, které byly použity pro tvorbu základní testovací sady. Hlavním výstupem práce je vytvořený diplomový skript. Popsaná je tvorba a fungování jednotlivých funkcí umožňující nejen multimediální nebo datový přenos, ale také funkce sloužící k analýze výsledků měření.

1 Multimediální služby v IP sítích

Multimediální služby jsou přenášeny ke koncovým zákazníkům různými způsoby. V dnešní době je nejdominantnějším typem přenosu přenos přes IP síť. Předpokládá se, že tomu tak bude i v blízké budoucnosti. Vzhledem k neustálému zvyšování rychlosti datového spojení, vylepšování komprese videa či rozvoji daného softwaru, mají koncoví uživatelé daleko větší požitek z přenosu. V dnešní době jednotlivé trendy společně s požadavky na dané služby ovlivňují nejen rozvoj síťových technologií, ale i vlastnosti samotného síťového přenosu. Digitální video je přesně načasovaný nepřetržitý tok informací o určité přenosové rychlosti. V IP sítích je přenášena spousta různých dat z velkého množství rozdílných zdrojů na jednom přenosovém kanále. Se všemi těmito datovými proudy dohromady je Internet časovaný sběr informací, který je následně rozdělen do samostatných paketů. IP a samotné video nevytváří spolehlivé spojení z důvodu upřednostnění samotné rychlosti přenosu před spolehlivým přenosem dat. Přenos multimediálního obsahu přes IP síť i přes tyto zmíněné faktory nadále expanduje, jelikož širokopásmového připojení dosahují téměř všechny domácnosti vyspělých zemích. Poskytovatelé mohou tyto sítě využívat pro přenos videoobsahu, aniž by museli budovat další sítě. IP síť zjednodušuje spouštění nových služeb, jako je například video na vyžádání (VoD neboli Video on Demand). Samotné náklady na IP sítě nadále klesají kvůli velkému objemu každoročně nově vyrobených zařízení a existenci celosvětových standardů. IP sítě jsou vybudovány po celém světě a počet uživatelů s vysokorychlostním připojením stále roste rychlým tempem. S rostoucím očekáváním diváků ohledně kvality a dostupnosti videoobsahu se dá očekávat, že kapacita IP sítí se bude neustále zvyšovat. [1]

1.1 Internet Protocol (IP)

Internetový protokol poskytuje mechanismus pro směrování toků paketů mezi zařízeními připojenými v síti. IP je běžný protokol používaný v celém Internetu. Bez tohoto protokolu by neexistoval způsob, jak by jedno zařízení posílalo data druhému zařízení. Základní jednotkou informace je paket, který je směrován IP sítí. IP pakety jsou používány v různých technologiích jako jsou např. Ethernet, bezdrátové či optické sítě. V IP sítích existuje řada různých videoslužeb. Patří k nim služby, které jsou náročné na snímkovou frekvenci (webové kamery) a dále služby náročné jako jsou televizní přenosy s vysokým rozlišením HD. IP technologie je široce rozšířená a velké množství videotechnologií může využívat přenosy v IP síti. Počet aplikací pro IP síť je rozsáhlý. Jednotlivým aplikacím byla přidělena čísla portů spravovaných danou organizací. IP podporuje mnoho různých zařízení s řadou odlišných softwarových operačních systémů, které lze různě konfigurovat. Internetový protokol je také

velmi flexibilní, protože není vázán na konkrétní fyzické telekomunikační technologie. IP přenos přes Ethernetové rozhraní je dominantní technologií v lokálních sítí. IP přenos funguje i tam, kde je kombinováno více technologií dohromady. Existuje řada důvodů proč je výhodné přenášet video přes IP sítě.

Výhodou IP sítě je její flexibilita, nízká cena a neuvěřitelné pokrytí po celém světě. IP technologie mají velmi nízké požadavky na hardware jednotlivých komponentů. Prakticky všechny počítače a notebooky jsou vybaveny Ethernet porty. Základní IP software je velmi levný či často zdarma. Počítače a jejich operační systémy obsahují mnoho zabudovaných IP služeb. Nízké náklady na IP sítě jsou pro provozovatele vysílání velkým přínosem.

Přenosy v IP sítích přináší i jisté nevýhody. Přenášený multimediální obsah přes Internet musí být ochráněn před možným nelegálním kopírováním či sdílením. Dalším problémem jsou velmi vysoké nároky na síťové prvky, které musí přenášet i další datový provoz. V tomto případě je nutné stanovit prioritu síťového provozu. Posledním problémem může být přístup k legálnímu multimediálnímu obsahu. Mnoho vlastníků obsahu má samostatné licenční podmínky pro různé formy distribuce. [1]

1.2 Typy přenosu IP videa

V současnosti existuje mnoho různých způsobů, kterými jsou IP sítě využívány k doručování multimediálního obsahu až ke koncovým účastníkům. Internet se stal klíčovou platformou pro distribuci médií. Ta otevřela nové cesty pro objevování televizního obsahu na jakémkoliv zařízení.

Jak již z názvu vyplývá, termínem **IPTV** se rozumí služba poskytující přenos přes internetový protokol IP. IPTV je někdy zaměňována s pojmem **Internetová televize**. I když obě dvě služby jsou založené na stejném principu, liší se zejména ve způsobu doručování videa. Internetová televize využívá k poskytování videa veřejný Internet, naopak IPTV uplatňuje zabezpečené privátní sítě k poskytování videoobsahu koncovým uživatelům. Tyto privátní sítě jsou spravovány a provozovány poskytovatelem služby. Sítě vlastněné a řízené telekomunikačními operátory nejsou přístupné běžným uživatelům Internetu a nachází se v pevně stanovených geografických oblastech. Naopak Internet žádná taková omezení nemá a k daným službám lze přistupovat odkudkoliv. Pokud je video přenášeno přes veřejný Internet, některé IP pakety se mohou při průchodu různými sítěmi zpozdit, nebo mohou být úplně zahozeny. V případě IPTV koncoví účastníci sledují přenášený obsah, aniž by si vybírali z kolekce videosouborů. Obsah na vyžádání (VoD) lze přehrát přes zařízení připojené k Internetu. Přenos přes Internet se využívá i v případě přehrávání internetových videí. Na rozdíl od tradičních televizních služeb, při nichž jsou videoprogramy vysílány podle předem stanoveného vysílacího plánu, **VoD** poskytuje koncovým uživatelům

IPTV možnosti vybírat, stahovat a zobrazovat obsah dle uvážení. VoD aplikace obvykle obsahuje knihovnu různého multimediálního obsahu. Uživatelé, kteří přistupují přes širokopásmové připojení do těchto knihoven, spotřebovávají poměrně značnou šířku pásma. [1, 2]

V rámci **mobilní televize** jsou aktuální videoslužby dostupné přímo v reálném čase, nebo ke stažení. V případě mobilní televize je možné kombinovat IP přenos s digitálním vysíláním v rámci mobilní sítě. Služby mobilní televize lze považovat za rozšíření služeb IPTV. Telekomunikační norma definuje fyzickou vrstvu, linkovou vrstvu distribučního systému a také MPEG-2 transportní tok jako formát videoobsahu. Mezi nejdůležitější parametry v rámci přenosu patří audio kvalita a obzvláště důležitou částí je samotná synchronizace videa se zvukem, při níž dochází ke snížení obrazové kvality vlivem pomalejšího datového připojení. Video je možné vysílat, či přijímat bezdrátovým způsobem. [3, 4]

1.2.1 Televize přes internetový protokol

Televize přes internetový protokol (IPTV) poskytuje služby digitální televize přes IP protokol náhradou za pozemní vysílání. Tyto služby se týkají multicastového vysílání, videa na vyžádání (VoD), přenosu hlasové služby (VoIP) a také různých webových služeb. Pro správnou funkčnost služeb jako např. VoIP a VoD, musí být zajištěna kvalita služby (QoS) prostřednictvím širokopásmového připojení. Televizní vysílání je souvislý tok pořadů, filmů a seriálů. Doručování probíhá přes privátní síť poskytovatele služeb. Všechny televizní kanály mají jednotný formát obsahu (jednu metodu komprese a přibližně stejnou přenosovou rychlost). Uživatelé mají možnost zapnout nejen živé vysílání, ale mají zde rozšiřující možnost, že si uživatel může zvolit, jaký pořad bude sledovat (v kombinaci s digitálním videorekordérem umožňuje časový posun videoobsahu, lze si také obsah ukládat pro pozdější zhlédnutí). Zákazníci mohou také využívat funkce na libovolné přetočení pořadu, nebo si mohou dle potřeby vysílání zastavit. V IPTV se hojně využívá multicastové vysílání. Multicastové vysílání umožňuje jednotlivým operátorům šetřit přenosovou šířku pásma. Místo poskytování každého vysílání každému uživateli je možné přenášet pouze obsah, který požadoval.

Komponenty IPTV:

- zdroje IPTV s VoD databází a jinými programy,
- vysokorychlostní připojení, které je tvořeno optickou páteřní sítí zajišťující funkce multicastu a QoS,
- vysokorychlostní přístupové sítě metalické nebo optické,
- uživatelské zařízení IPTV, jako je digitální televize nebo počítač s vysokým rozlišením.

Uživatelské funkce IPTV:

- výběr televizního kanálu (rychlý výběr nebo krátká doba změny kanálu),
- úložiště (TV programy jsou na místním úložném zařízení),
- QoS (musí být zajištěna dostatečná velikost šířky pásma pro televizní kvalitu standardní či s vysokým rozlišením),
- nízká cena pro koncové uživatele. [3, 5]

1.2.2 Internetová televize

Při přenosu internetového videa je používán veřejný Internet. Tyto přenosy se nazývají Over-the-top (OTT). Kvalita těchto služeb může kolísat v závislosti na dostupné šířce pásma síťového připojení. K těmto službám lze přistupovat přes klasický webový prohlížeč. Jakmile si uživatel vybere daný videoobsah, dojde k přenosu z určených serverů do zařízení uživatele.

Hlavní vlastnosti:

- různě trvající videoobsah,
- různé formáty obsahu (komprese videa, správa práv a rozlišení videa),
- přenos veřejným Internetem. [5]

1.2.3 Srovnání jednotlivých typů přenosu

Sítě pro přenos IP videa lze rozdělit do dvou kategorií a to veřejné a soukromé. Nejdůležitější veřejnou sítí je Internet. Přenos veřejnou sítí využívají služby: video na vyžádání, Internetová televize a nebo také Internetové video. V případě těchto služeb ("best effort") dochází ke zpoždění a zahazování paketů. Soukromé sítě existují v mnoha různých konfigurací, od malých point-to-point až po velké systémy, které pokrývají připojení milionů uživatelů. Každý uživatel musí být připojen přímo k síti, aby bylo možné přijímat dané služby. IPTV systém je jedinečný v tom, že vyžaduje přenos přes privátní síť a umožňuje tak operátorům ovlivňovat kritické přenosové parametry. Tyto kvalitativní parametry zásadně ovlivňují výslednou kvalitu videopřenosu. Z praktického hlediska je přenos přes privátní síť velmi ekonomickým řešením. V případě IPTV by většina poskytovatelů připojení nebyla schopna zajistit vhodné parametry pro přenos tak velkého objemu dat.

V IP sítích termín QoS vyjadřuje mechanismus přiřazení různým službám určité priority přenosu. Řízení QoS pomáhá zajistit nepřetržitý kvalitní multimediální přenos. QoS lze nastavit v hlavičce IP protokolu, což následně umožňuje směrovačům udělit prioritu konkrétnímu paketu. Tento mechanismus se používá k upřednostnění videopřenosu před dalšími síťovými provozy. Pomocí tohoto mechanismu je možné snížit ztrátovost paketů či zpoždění paketů. Ve veřejné síti by tato funkce byla bezcenná, protože by nebylo možné kontrolovat jednotlivé uživatele, zda nemají

nastavenou vysokou prioritu pro svůj přenos. IPVoD, Internetová televize a přenos videa přes Internet fungují v sítích, v nichž není zajištěna priorita přenosu. Naopak při přenosu IPTV je tento mechanismus využíván.

V IP sítích existují čtyři režimy komunikace (unicast, multicast, broadcast a anycast). Každá z výše uvedených služeb pracuje v jiném režimu. Pro IPVoD službu i internetové video se využívá unicastové vysílání. Je to jediný způsob, jak uživatelům udělit náhodný přístup k obsahu. Internetová televize používá replikovanou unicastovou komunikaci, jelikož multicast není ve veřejné síti podporován. Obsah je kopírován specializovanými reflektujícími servery, které jsou v síti různě rozmístěny. IPTV služba využívá multicastového přeposílání obsahu.

Síťový protokol je sada pravidel, jež řídí přenos dat mezi aplikacemi. Hlavní rozdíly mezi protokoly jsou dány především jejich implementací do reálného provozu. Některé protokoly jsou využívány pro přenos dat v reálném čase, jiné naopak mají funkci kontrolních mechanismů, apod. Jedním z nejvíce používaných protokolů je Hyper Text Transfer Protocol (HTTP). Uvedený protokol se využívá pro služby VoD, internetovou TV nebo internetová videa. Tento protokol rozděluje obrazové informace do bloků dat. Při odesílání dat server odesílá pakety bez ohledu na zaslané potvrzení od webového prohlížeče klienta. Jedná se však o nespolehlivý přenos dat. Důraz je kladen především na rychlost přenosu, což má za následek přirozenější načasování. Pakety jsou zasílány dle obsahu videa a zvuku. Následně si webový prohlížeč uživatele stáhne blok dat dříve, než je potřeba. Nevýhodou progresivního přístupu může být počáteční zpoždění. Pokud klient stáhne velký blok dat, může nastat časová prodleva, než dojde k uložení do vyrovnávací paměti. Real-Time Transport Protocol (RTP) je protokol určený pro doručování multimediálních dat v reálném čase. Protokol RTP využívá protokol UDP. Tato kombinace se běžně používá pro IPTV službu, i pro jiné typy doručování. [1]

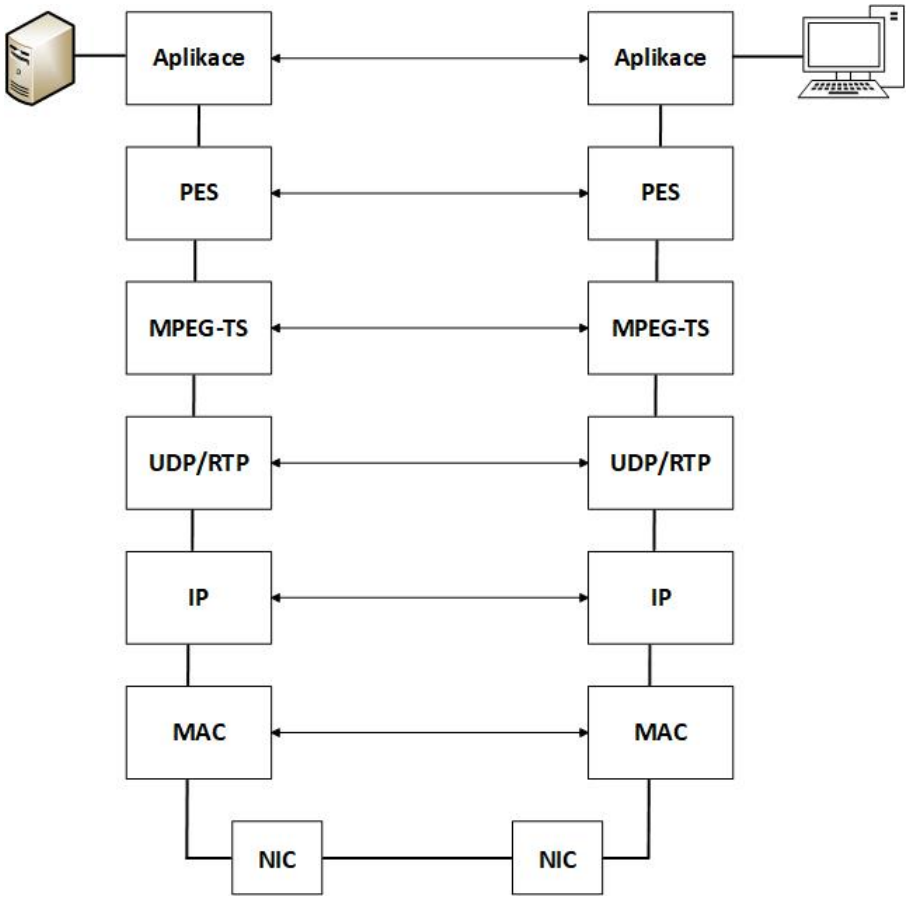
1.3 Architektura přenosu

IPTV systém lze rozdělit do několika vrstev. Každá vrstva vykonává určitou funkci potřebnou pro síťovou komunikaci. Hierarchicky vyšší vrstva požaduje služby vrstvy nižší. Přenos probíhá od fyzické vrstvy.

Fyzická vrstva reprezentuje proud jednotlivých bitů, které jsou přenášeny prostřednictvím přenosového média. Linková vrstva je druhou vrstvou v pořadí. Na této vrstvě probíhá komunikace mezi zařízeními v lokální síti (koordinace přístupu k fyzickému kanálu). Další vrstva se označuje jako IP vrstva. Na této vrstvě dochází ke směrování paketů mezi dílčí sítě. Transportní vrstva je implementována až v koncových zařízeních (softwarová implementace). Tato vrstva je zodpovědná za přenos datagramů (UDP/RTP) mezi odesílatelem a příjemcem. Relační vrstva zajišťuje

vytvoření spojení pro přenos jednotlivých složek multimediálního obsahu pro programový kanál (např. pro MPEG). Vrstva paketového elementárního toku (PES) mapuje multimediální složky na transportní toky. Aplikační vrstva (aplikace) zajišťuje správu informačního rozhraní v rámci komunikace mezi zařízením a koncovým účastníkem.

Obrázek 1.1 znázorňuje jednotlivé vrstvy IPTV systému. První tři vrstvy jsou důležité především pro správné fungování síťové komunikace. Čtvrtá vrstva má na starosti přenos příslušných datagramů. Vrstva MPEG transport stream (MPEG-TS) kombinuje více typů paketizovaných elementárních proudů (jednotlivé PES pro audio, video a data) do výsledného transportního streamu pro programový kanál. Vrstva paketového elementárního toku (PES) mapuje a přiřazuje jednotlivé složky (jednotlivé elementární streamy videa, audia a dat) multimediálního přenosu na konkrétní paketizované elementární proudy. Samotná aplikace zpřístupňuje výsledný videoobsah koncovému uživateli. [6]



Obr. 1.1: Vrstvy IPTV [6]

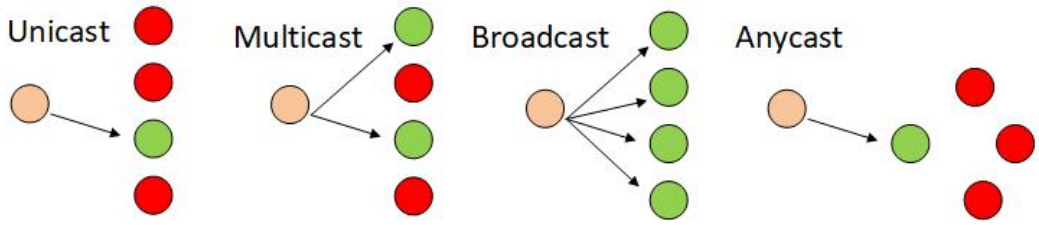
2 Typy přenosů v IP sítích

V IP sítích rozlišujeme čtyři typy přenosů paketů:

- unicast - jeden uzel komunikuje přímo s druhým uzlem,
- multicast - přenos ze zdrojového uzlu k vybrané skupině příjemců,
- broadcast - zdrojový uzel komunikuje se všemi uzly v síti,
- anycast - komunikace probíhá s nejbližším uzlem, který spadá do patřičné skupiny.

V tradičních IP sítích je paket obvykle odesílán zdrojem do jednoho cíle (unicast). Naproti tomu multicast byl vyvinut pro výhodnou komunikaci mezi zdrojem a vícero příjemci. Multicastové vysílání je vhodné použít především k multimediálnímu přenosu. Oproti unicastovému vysílání má tento typ vysílání několik výhod. Velmi zefektivňuje celkovou komunikaci (šetří šířku pásma) a snižuje přetížení sítě (nedochází k takovému zahlcení a ztrátovosti paketů na směrovačích).

Pakety je také možné odesílat na všechna zařízení v síti (broadcast), což se vzhledem k povaze tohoto přenosu nedoporučuje. Směrovače, které obdrží paket s broadcastovým typem cílové adresy, mají ve výchozím nastavení zakázané tyto pakety směrovat dále. Posledním výše zmíněným typem doručování paketů je anycast. Anycast umožňuje více zařízením sdílet stejnou IP adresu. Komunikace následně probíhá se zařízením, jež je nejblíže nebo nejvíce dostupné. [7, 8]



Obr. 2.1: Typy vysílání [9]

2.1 Multicastové vysílání

IP multicast je technologie, která umožňuje šetřit šířku pásma. Poskytuje jediný datový tok velkému množství příjemců. Nahrazením kopií jediným datovým proudem informací pro všechny příjemce dokáže snížit celkovou zátěž pro všechna účastnická zařízení. V rámci vícesměrového vysílání jsou směrovače zodpovědné za kopírování a distribuci obsahu všem hostitelům, kteří naslouchají na konkrétní skupině. Směrovače využívají Protocol Independent Multicast (PIM) k vytváření distribučních

stromů pro přenos obsahu vícesměrového vysílání, což má za následek nejefektivnější doručování dat více příjemcům. IP multicast nabízí výhody týkající se zachování šířky pásma. V aplikacích s velkou šířkou pásma, jako je MPEG video, může být výhodný především v situaci s menším počtem příjemců, jinak by videoproudy spotřebovaly značnou část přenosové šířky pásma. Vícesměrové vysílání je také výhodné i pro aplikace, jež spotřebovávají malou šířku pásma, a to i v případě, že je obsah přenášen velkému množství koncových uživatelů. Kromě toho je multicast jedinou alternativou, která není všesměrová (pro situace vyžadující odesílání dat vícero koncovým účastníkům). U aplikací s nízkou šířkou pásma existuje alternativa, jež je nazývaná replikace dat ze zdroje. V případě této alternativy může dojít ke zhoršení výkonu aplikace a to především z důvodu vzniku latence a proměnného zpoždění. K vyřešení tohoto problému by bylo nutné nasadit drahé servery pro správu replikace a distribuci dat. Tato řešení navíc vedou k vícero přenosům stejného obsahu, což spotřebovává velké množství dostupných prostředků. U většiny aplikací s velkou šířkou pásma mají tyto problémy za následek, že vícesměrové vysílání je fakticky jedinou vhodnou volbou. Mezi další aplikace, které využívají multicastového vysílání, patří:

- firemní komunikace,
- interaktivní hraní,
- sledovací systémy,
- přenos digitálního videa (IPTV).

Přenos realizovaný pomocí IP multicastu využívá rozsahu adres třídy D (224.0.0.0 až 239.255.255.255). V rámci rozsahu adres této třídy existují adresy, které jsou institucí Internet Assigned Numbers Authority (IANA) vyhrazeny pro konkrétní účely. Vyhrazené adresy jsou určené např. známým protokolům a aplikacím vícesměrového vysílání. [9]

2.1.1 Distribuční stromy

Směrovače s podporou vícesměrového vysílání vytvářejí logické distribuční stromy, jež určují přenosovou cestu vícesměrového vysílání za účelem doručení obsahu všem příjemcům. Tvorba distribučních stromů je velice složitá a zpravidla ji provádí multicastový směrovací protokol. Cílem těchto protokolů je vytvořit co nejvýhodnější strom. Existují mechanismy pro vytváření a udržování distribučních stromů. Pro tuto tvorbu existují různá kritéria (celkové zpoždění, cena, šířka pásma). Různé multicastové algoritmy používají různé techniky pro vytvoření distribučního stromu. Nově vzniklé stromy tvoří topologii, která neobsahuje žádné smyčky.

Stromy se dělí na dva základní druhy:

- sdílený strom (Shared Tree),

- strom nejkratší cesty (SPT).

Členové skupin se mohou kdykoliv připojit či odpojit, proto musí být distribuční stromy dynamicky aktualizovány. Pokud aktivní přijímače na konkrétní větvi přestanou vyžadovat provoz, směrovače jsou schopné tuto větev oříznout a zastavit tak směrování do této části. Následně jsou zprávy posílány pouze tam, kde se strom větví. Pokud přijímač začne žádat o příjem multicastového obsahu, tak směrovač opět upraví distribuční strom a znovu začne směrovat provoz do této části sítě.

Nejjednodušší formou multicastového distribučního stromu je **strom nejkratší cesty** (SPT). Kořenem je vždy první směrovač připojený k vysílači. Tento kořen distribuuje data nejkratší cestou až ke koncovým příjemcům. Pro každý zdroj dat musí existovat distribuční strom. Pro SPT se používá zápis (S,G), v němž S je IP adresa zdroje a G je IP adresa cílové skupiny. Podle definice SPT obsahují optimální cesty mezi zdrojem a příjemci z hlediska počtu skoků, což má za následek minimální zpoždění pro distribuci multicastového provozu. Nicméně je nutné udržovat informaci o cestě pro kterýkoliv zdroj na každém směrovači. V rozsáhlých sítích s mnoha zdroji mohou tyto informace zahlcovat směrovače, což může způsobovat výrazné zvýšení paměťových prostředků.

Sdílené stromy (Shared tree) používají jeden společný kořen (root) umístěný ve vybraném bodě sítě. Strom existuje pouze jeden, a to pro všechny zdroje dat. Sdílený kořen se nazývá Rendezvous Point (RP). Data jsou odesílána nejprve na tento kořen, který se následně stará o jejich distribuci prostřednictvím jediného stromu. Veškerý multicastový přenos je tedy posílán přes tento bod. Vzhledem k principu tohoto mechanismu jsou na kořenový směrovač kladeny vysoké výpočetní nároky. Multicastový přenos nemusí probíhat vždy efektivně z hlediska počtu skoků a zpoždění, jelikož může v síti existovat i výhodnější přenosová cesta mezi zdrojem a příjemci. Tento problém vyžaduje vhodné umístění RP. Technika sdíleného stromu je výhodná z hlediska minimalizace paměťových požadavků na směrovače. [7, 10]

2.1.2 Druhy multicastového přenosu

Multicastový přenos se dělí na dva základní druhy:

- Any Source Multicast (ASM),
- Source Specific Multicast (SSM).

Vícesměrové vysílání z jakéhokoliv zdroje (ASM) může používat obě techniky distribuce stromů (sdílený strom a strom nejkratší cesty). ASM je používán především ve chvíli, kdy není pevně stanovené, kdo bude vysílač a kdo přijímač. Zprávy vícesměrového vysílání obvykle začínají ze sdíleného stromu (pomocí RP), ale následně se přepínají do stromu nejkratší cesty, aby byla zajištěna vyšší účinnost

sítě. ASM má značné nevýhody. Nemůže si určovat, od kterých zdrojů dat bude data přijímat a od kterých naopak ne, což je podstatné z bezpečnostního hlediska.

Vícesměrové vysílání se specifickým zdrojem (SSM) vyžaduje přesně specifikovaný zdroj dat. Tato metoda se používá pro službu IPTV. Technika SSM má několik výhod. Jednou z nich může být optimalizované využití šířky pásma, kdy přijímače mohou vyžadovat provoz pouze z explicitně známých zdrojů, čímž se uvolní značná šířka pásma. Další výhodou je i snížení rizika v podobě eliminace útoků z neznámých zdrojů. [7, 10, 11]

2.1.3 Internet Group Management Protocol (IGMP)

Tato kapitola pojednává o velmi důležité části multicastového přenosu dat. Internet Group Management Protocol (IGMP) je protokol používaný k dynamické registraci hostitelů. V multicastovém prostředí se informace o členství vyměňují mezi příjemcem a nejbližším směrovačem podporujícím multicast. IGMP je používán hostitelskými zařízeními k připojení či k odchodu ze skupiny (IGMP Join/Leave) v tom případě, pokud komunikace probíhá přes protokol Internet Protocol version 4 (IPv4). Hostitelé vytvářejí členství ve skupinách odesíláním zpráv (Membership Report) jejich nejbližším multicastovým routerům. Takovéto směrovače monitorují jednotlivé IGMP zprávy. Směrovače se periodicky dotazují (Membership Query), aby zjistily, které skupiny v dané podsíti jsou aktivní a které naopak ne. Existují tři verze protokolu IGMP (IGMPv1, IGMPv2 a IGMPv3). Každá z těchto verzí přináší vylepšení oproti té předchozí.

IGMPv1 poskytuje hostitelům jednoduchý mechanismus, jak informovat nejbližší multicastový směrovač o tom, že chtějí přijmout příslušnou skupinu (zpráva IGMP report). Kromě toho IGMPv1 poskytuje také mechanismus pro směrovače, aby se mohly dotazovat, zda jsou v dané podsíti nějací hostitelé a zda mají zájem o příjem vícesměrového vysílání. Dotazy IGMP, jak již bylo zmíněno výše, jsou přenášeny v pravidelných intervalech. Výchozí nastavení tohoto intervalu činí šedesát sekund. Naneštěstí tato verze neposkytuje žádný mechanismus pro opuštění skupiny. Takže i poté, co poslední zainteresovaný hostitel přestane odesílat reporty, směrovač i nadále vysílá.

IGMPv2 funguje stejně jako předchozí verze, tedy IGMPv1, umožňuje ale navíc explicitní funkci pro opuštění skupiny (Leave Group). IGMPv2 také vylepšuje koncept reportu tím, že umožňuje směrovači odesílat dotazy pro členy určité skupiny (Group-Specific Query). Tyto zprávy výrazně zlepšují rychlost zastavení multicastové komunikace vysílaní jedné ze skupin. Po zastavení je následně možné implicitně začít vysílat data pro druhou skupinu. Příklad komunikace: V síti se nachází hostitel, který aktuálně přijímá provoz vysílaný skupině 1. Hostitel nyní chce zastavit

příjem skupiny 1 a ihned poté chce začít přijímat data, jež jsou vysílána skupině 2. Pokud by byla využívaná první verze protokolu IGMP, pak by hostitel přestal vysílat reporty pro skupinu 1 a začal by přenášet reporty druhé skupině. Do doby, než vyprší časový limit členství ve skupině 1, bude hostitel přijímat síťový provoz pro obě skupiny. V případě použití druhé verze protokolu IGMP (IGMPv2), hostitel odešle zprávu typu explicitního opuštění skupině 1 a začne odesílat reporty pro skupinu 2. Za předpokladu, že v sítí nejsou žádní hostitelé, kteří chtějí přijímat provoz pro skupinu 1, směrovač přestane vysílat obsah určený pro tuto skupinu.

IGMPv3 poskytuje funkci specifikace zdroje, ze něhož si daný hostitel přeje přijímat data. Přijímače signalizují členství v multicastové skupině ve dvou režimech:

- Include - v tomto režimu je specifikován seznam zdrojů, od kterých si přeje přijímat provoz,
- Exclude - v tom režimu je specifikován seznam zdrojů, od kterých nemá být vícesměrové vysílání přijímáno.

IGMPv3 je nasazováno v případě, kdy je vyžadováno specifické vícesměrové vysílání (implementace SSM). Oproti IGMPv1 a IGMPv2 je nejnovější verze nejméně implementována, i když se s průběhem let stále více uplatňuje. IGMPv2 je nepochybně lepší a škálovatelnější než IGMPv1. Je otázkou, zda IGMPv3 poskytuje jakoukoliv významnější výhodu oproti IGMPv2. [7, 9, 12]

2.1.4 Protocol Independent Multicast (PIM)

Směrovací protokol používaný ve spojení s IP multicastem se nazývá Protocol Independent Multicast (PIM). PIM má varianty směrovacího protokolu používaného k vytvoření distribučního stromu pro vysílání od zdroje (či zdrojů) k přijímačům. Směrovač přeposílá multicastový paket, pouze pokud byl přijat na nadřazeném rozhraní ke zdroji nebo na RP (v případě techniky sdíleného stromu). Pokud paket dorazí na rozhraní, které není na příslušné cestě ke zdroji, je tento paket zahozen. Tento mechanismus se nazývá Reverse Path Forwarding (RPF) a zabraňuje vzniku smyček v topologii, což je kritický aspekt fungování multicastového vysílání. RPF se vyhýbá smyčkám tím, že nepředává duplicitní pakety. K určení přenosové cesty se využívají standardní směrovací protokoly, jako je např. Open Shortest Path First (OSPF). [7, 9, 10, 13]

PIM Dense Mode (PIM-DM)

Tento směrovací protokol byl navržen pro případ husté distribuce vícesměrového vysílání. Očekává se, že daný multicastový obsah bude vyžadovat velký počet uživatelů v podsíti. V první fázi dochází k záplavovému posílání zpráv všem hostitelům. Směrovače, které nemají ve své podsíti žádné příjemce, odesílají zprávy PIM prune

a následně jsou z této cesty odstraněny. Tento postup se periodicky opakuje každé tři minuty, což má za následek značnou režii. Zdroj začne odesílat data, každý směrovač data přijme a následně je předá svým PIM sousedům a také na všechny spoje, které jsou připojené k přijímači. Směrovač také testuje, zda data přišla po optimální přenosové trase (mechanismus RPF). PIM DM podporuje především stromy nejkratší cesty. Nepoužívá tedy RP, což umožňuje snadnější implementaci než při nasazení řídké distribuce vícesměrového vysílání. PIM-DM je vhodné použít především v případě blízké vzdálenosti mezi zdrojem a přijímačem. [7, 9, 10, 13]

PIM Sparse Mode (PIM-SM)

PIM Sparse Mode používá explicitní model spojení, kde se do multicastových skupin připojují pouze směrovače s aktivními přijímači. Oproti předchozímu mechanismu to má výrazné výhody v šetření značné režie v síti. Směrovače používají zprávy PIM Join a Prune k připojení a opuštění distribučních stromů. Zpráva PIM Join se musí zasílat opakovaně k potvrzení platnosti příjmu skupiny v dané podsíti. Směrovací protokol PIM-SM je primárně nasazen pro sdílený distribuční strom, avšak je možné ho využít i pro stromy nejkratší cesty. PIM-SM se využívá především pro rozsáhlé sítě, jelikož nedochází k zaplavení celé sítě. Tento protokol používá kontrolní bod (RP), který je stěžejním bodem pro výměnu vícesměrových dat. Každý směrovač v síti by měl mít nakonfigurované informace o RP. Směrovače, ke kterým jsou připojeny zdroje multicastového vysílání, odesílají multicastová data na RP (zpráva Pim Register). RP následně směruje tento provoz k jednotlivým příjemcům. Směrovače informují Rendezvous Point, že mají ve své podsíti aktivní příjemce (registrace příjemců do skupiny pomocí IGMP zpráv). PIM-SM je nejpoužívanějším multicastovým směrovacím protokolem. [7, 9, 10, 13]

PIM Sparse-Dense Mode

Tento režim je kombinací obou předchozích protokolů. Rozhodnutí použít řídký nebo hustý režim pro konkrétní skupinu vícesměrového vysílání závisí na tom, zda má skupina odpovídající záznam v mapovací cache paměti. Pokud v mezipaměti existuje záznam, používá tato skupina vícesměrový protokol v řídkém režimu. Pokud v paměti neexistuje záznam pro danou skupinu, pak se používá vícesměrové vysílání v hustém režimu. [7, 9, 10, 13]

PIM Source Specific Multicast (PIM-SSM)

Protokol vícesměrového vysílání pro specifický zdroj (PIM-SSM) funguje jako rozšíření pro síťový přenos, v němž jsou data předávána pouze k přijímačům, které se ke zdroji explicitně připojí. Pro tento druh přenosu se používá metoda vytváření

sdíleného distribučního stromu. Mnoho podnikových řešeních je založeno právě na protokolu PIM-SSM. Tento protokol se ukázal jako spolehlivé, rozsáhlé a efektivní řešení. Jsou zde však jistá omezení a složitosti ohledně fungování Internet Standard Multicast (ISM). ISM musí např. udržovat přehled o tom, kteří hostitelé aktivně odesílají multicastový provoz. K předávání těchto informací je využíván protokol IGMPv3. Dále je možné využívat funkce mapování, které umožňují automatické určení zdroje, i když je používán protokol IGMPv2. PIM-SSM je možné využít v IPTV službě. [7, 9, 10, 13]

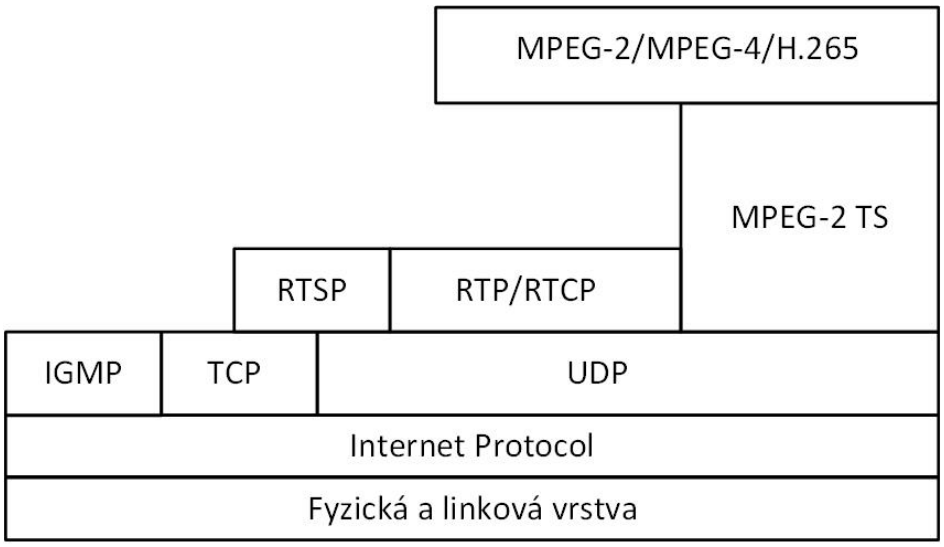
PIM Bidirectional (BIDIR-PIM)

Zcela běžně jsou distribuční stromy jednosměrné, což znamená, že data jsou posílána pouze k příjemcům. PIM Bidirectional je směrovací protokol vícesměrového vysílání, jenž umožňuje vytvoření obousměrného distribučního stromu. Nasazení BIDIR-PIM tedy uskutečňuje vytvoření jednoho obousměrného sdíleného stromu. Tento způsob je efektivní z hlediska malé obsazenosti šířky pásma (přenos many-to-many), avšak efektivita směrování může být i nižší. [7, 9, 10, 13]

3 Distribuce multimediálního obsahu

Tato kapitola se zabývá klíčovými síťovými protokoly a možnostmi přenosu multimediálního obsahu přes IP síť. Od počátku IP sítí se neustále objevují nové nápady a různé služby. Mnohé z těchto nových funkcí vyžadují standardizované způsoby výměny dat a dalších informací. Nové IPTV služby lze vnímat jako kombinace televize s vysoce interaktivními internetovými koncepty. Výsledkem je řada nových aplikací založených na doručování audiovizuálního obsahu obousměrným kontrolovaným způsobem. Podpora datových a multimediálních aplikací přes běžné infrastruktury je pravděpodobně hlavním úspěchem IP sítí. Multimediální přenosy vyžadují dostatečnou šířku pásma (propustnost) a malé zpoždění. Při přenosu videa v reálném čase vzniká ztrátovost a chybovost. V souvislosti s tímto vznikly nové protokoly, které se snaží řešit problémy ovlivňující výsledný přenos.

Na obrázku 3.1 jsou zobrazeny protokoly věnované konkrétním aplikacím. Např. protokol IGMP je určený k podpoře přepínání kanálů. Protokol RTSP poskytuje funkci videorekordéru a je využíván ve službě VoD. RTP protokol se využívá k samotnému přenosu multimediálních dat. [14, 15]



Obr. 3.1: Protokoly používané pro multimediální služby [14]

3.1 Real-Time Transport Protocol (RTP)

Real-Time Transport Protocol byl nejprve vyvinut pro podporu audio a videokonference, avšak byl zároveň záměrně navržen tak, aby do budoucna byl flexibilní a dále

rozšiřitelný. RTP je protokol, jenž umožňuje doručování multimediálních dat v reálném čase koncovému zařízení. Lze jej použít v prostředích unicastového i multicastového vysílání. Přenos může probíhat od jednoho až do mnoha tisíc koncových bodů. Mohou být přenášeny proudy zakódované prakticky jakýmkoliv současným nebo budoucím schématem kódování. RTP podporuje koncept mixerů a translátorů. Mixéry přijímají síťové proudy z různých zdrojů a jejich kombinováním se tak vytváří nový výsledný proud. To je velmi důležité z hlediska synchronizace přenosu. Mixér generuje vlastní identifikátor zdroje a také časový kód pro datový proud, který vysílá. Translátory předávají RTP pakety s neporušenou synchronizací. Mohou změnit kódování, kopírovat pakety pro přenos z multicastové typu na unicast a také poskytují filtrování dat. RTP protokol umožňuje nejlepší podporu pro výše uvedené vlastnosti. Není ideálním řešením pro každou aplikaci, ale vzhledem ke své flexibilitě je výbornou volbou pro mnoho multimediálních implementací. Porovnání vlastností protokolů User Datagram Protocol (UDP) a Transmission Control Protocol (TCP) s RTP je zajímavé v několika bodech:

- UDP poskytuje pouze jednoduchý nespolehlivý přenos.
- TCP nabízí služby typu end-to-end, ale na ztrátu a zahlcení reaguje zpomalením přenosu dat. To však není vhodným řešením pro přenos dat v reálném čase.
- Návrh RTP počítal s požadavkem na poskytování služeb v reálném čase přes nespolehlivou IP síť.
- RTP poskytuje přehled o své transportní vrstvě. To umožňuje aplikacím určit způsob přenosu informací.
- RTP nabízí řadu klíčových funkcí pro podporu přenosu dat v reálném čase. Přenos časové značky, přenos sekvenčního čísla, zdroj a identifikace užitečného zatížení, zpětná vazba na kvalitu příjmů, synchronizace a řízení přenosu.
- RTP se při přenosu svých paketů spoléhá na síťovou vrstvu. RTP nespecifikuje transportní protokol, avšak výhradně používá UDP k doručování paketů.

Parametry pro přenos v reálném čase definuje struktura uvedená v **RTP hlavičce**. Hlavička je znázorněna na obrázku 3.2 s popisem jednotlivých částí.

Version (V) - Verze RTP protokolu.

Padding (P) - Identifikuje výskyt doplňkových oktetů na konci užitečných dat. Pokud je nastaven výplňový bit, poslední oktet obsahuje oktety, které by měly být ignorovány.

Extension (X) - Pokud je nastaven rozšiřující bit, musí za pevnou hlavičkou následovat hlavička rozšíření. Obvykle je to používané jako rozšíření užitečného zatížení, avšak používá se to velmi zřídka.

CSRC count (CC) - Počet CSRC obsahuje počet identifikátorů CSRC, které následují za pevnou hlavičkou.

Marker (M) - Interpretace značky je definována profilem. Umožňuje označit hranice rámců, jež mají být označeny v toku paketů.

Payload type (PT) - Toto pole identifikuje formát užitečného RTP zatížení.

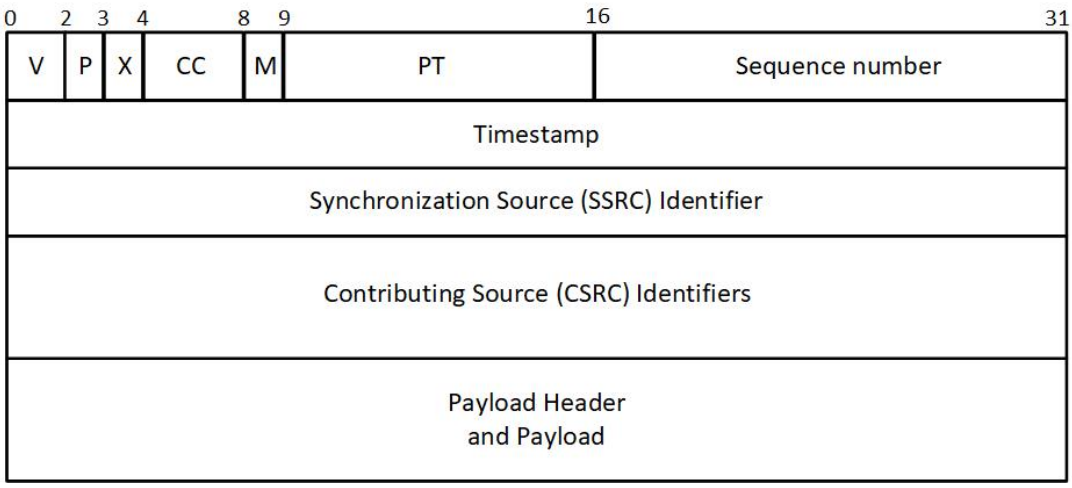
Sequence number - Počáteční hodnota tohoto pole by měla být náhodná a měla by se zvyšovat o jednu pro každý odeslaný datový paket.

Timestamp - určuje čas kódování prvního oktetu multimediálních dat v tomto paketu. Časové razítko neurčuje skutečný čas. Slouží k měření jitteru a synchronizaci. Různé RTP relace mohou mít různá časová měřítka.

Synchronization source identifier (SSRC) - jednoznačně identifikuje zdroj přenášeného proudu v užitečné zátěži. Je vybírán náhodně.

Contributing source identifier (CSRC) - seznam zdrojů přispívajících k přenosu. Může být až patnáct CSRC položek. Například pokud je počet audio proudů zkombinován do jednoho toku pomocí směšovače a každý z původních zdrojů bude mít v tomto seznamu své SSRC zapsané jako CSRC.

Payload - neboli užitečné zatížení, je jednou z nejdůležitějších položek RTP protokolu. Specifikace rozložení a samotná interpretace užitečného zatížení je definována pomocí profilů. [15]



Obr. 3.2: RTP paket [15]

3.2 RTP Control Protocol (RTCP)

RTP Control Protocol (RTCP) je součástí celkové implementace RTP protokolu a poskytuje zpětnou vazbu o kvalitě přenosu. Také umožňuje sledování odesílatelů v případě restartů nebo nadměrné ztráty paketů. RTCP tyto služby poskytuje všem účastníkům relace prostřednictvím periodického vysílání řídících paketů. Zpětná

vazba na kvalitu přenosu může být zajištěna adaptivním kódovacím schématem pro případnou vhodnou volbu přenosové rychlosti při přenosu dat. Dále poskytuje kritické informace pro vícesměrové vysílání. Použití kanonického názvu (CNAME) pomocí RTCP umožňuje přijímači sledovat zdroj vysílání, pokud se změní SSRC. CNAME lze také použít k přiřazení více proudů, jako je přenos hlasu a videa, k jednomu zdroji. Standard RFC 3550 vyžaduje, aby RTCP pakety byly zasílány všem účastníkům relace. Aby bylo možné RTCP škálovat na velký počet účastníků, je nutné řídit přenosovou rychlost dat. RFC 3550 poskytuje pravidla časování pro výpočet intervalu přenosu RTCP. Cílem je, aby RTCP nevyužívalo více než 5 % celkové šířky pásma. Každý účastník by měl udržovat tabulku o všech ostatních aktivních účastnících z důvodu výpočtu intervalu přenosu RTCP paketů. Translátory, které upravují RTP zprávy, se musí také starat o správu příslušných RTCP zpráv. Mixéry generují vlastní zprávy o příjmu (viz níže).

V případě RTCP protokolu existuje pět typů paketů:

- **Sender Report (SR)** - pakety typu SR jsou odesílány účastníky, kteří přijímají a odesílají data. Tento typ poskytuje statistiky o vysílání i příjmu. Pokud účastník neodeslal žádné další pakety od odeslání předchozí zprávy, je místo nich odeslána zpráva RR.
- **Receiver Report (RR)** - paket typu RR obsahuje informace o příjmu pouze od účastníků, kteří přijímají data. Pro každý ze zdrojů z nichž tento účastník přijal pakety, by měla existovat jedna zpráva o příjmu.
- **Source Description (SDES)** - položky popisu zdroje, včetně CNAME.
- **BYE** - indikace odchodu účastníka z příslušné relace.
- **APP** - funkce definované aplikací.

Začátek každého RTCP paketu je pevná hlavička podobná RTP paketu. To co následuje, jsou strukturované proměnné délky končící na 32bitové hranici. Více RTCP paketů by mělo být vždy zřetězeno tak, aby vytvořilo výsledný RTCP paket. Každý paket obsahuje zprávy o příjmu (SR nebo RR) a SDES (CNAME). RR musí být prvním paketem ve složené formě. [15]

3.3 Session Announcement Protocol (SAP)

Session Announcement Protocol (SAP) slouží k oznamování probíhajících relací v rámci vícesměrového vysílání. Tento protokol je definován v RFC 2974. Oznamovatel SAP periodicky vysílá zprávu obsahující oznámení na multicastovou adresu a port. Pomocí těchto zpráv je zajištěno, že se libovolný počet uživatelů může připojit k příslušnému multicastovému vysílání. Perioda vysílání je stanovena takovým způsobem, aby celková velikost SAP zpráv nepřesáhla stanovený limit. Rychlost odesílání zpráv je odvozena od počtu účastníků multicastového vysílání. Zpráva SAP

nenese žádné informace o počtu zúčastněných stran, pouze oznamuje, že nějaká relace existuje. Zprávy tohoto protokolu se odesílají na port 9875. [16]

3.4 Session Description Protocol (SDP)

Session Description Protocol (SDP) je ve skutečnosti formát zprávy pro popis inicializačních parametrů multimediálního toku. Původně byl vyvinut společně s protokolem SAP (Session Announcement Protocol) pro multicastové vysílání. Nicméně našel i všeobecné využití pro různé síťové protokoly. Je důležité zmínit, že není určen pro vyjednávání skutečného obsahu nebo kódování multimédií. Aplikace používají SDP k oznámení relace, pozvání účastníků a vyjednávání parametrů, jako jsou typ média, formát a další vlastnosti.

Popis relace je série polí <TYP>=<HODNOTA>. Pole <TYP> obsahuje písmeno určující daný typ. Pole <HODNOTA> závisí na použitém typu. Relace pomocí SDP protokolu obsahuje následující základní údaje: název a účel relace, čas, v němž je relace aktivní, obsažená média, informace potřebné k příjmu a informace o šířce pásma.

Popis relace:

- v - verze protokolu,
- o - identifikátor původu a relace,
- s - název relace,
- u - adresa URL popisu,
- e - emailová adresa,
- p - číslo telefonu,
- c - informace o spojení,
- b - informace o šířce pásma,
- z - nastavení časového pásma,
- k - kryptovací klíč,
- a - nula nebo více řádků atributu relace.

Popis času:

- t - čas, po který je relace aktivní,
- r - nula nebo více opakování.

Popis médií:

- m - jméno média a transportní adresa,
- média typ - audio, video, text, aplikace nebo zpráva,
- transportní protokol - UDP a RTP,
- média formát - interpretace závisí na poli transportní protokol,
- i - titulek,
- c - informace o spojení,

- b - informace o šířce pásma,
- k - kryptovací klíč,
- a - nula nebo více řádků atributu relace. [15]

3.5 Real Time Streaming Protocol (RTSP)

Real Time Streaming Protocol (RTSP) je aplikační protokol, který umožňuje klientovi řídit vysílání více multimediálních proudů odeslaných ze serveru. Multimediální toky mohou být buď uložená data, nebo živé přenosy. RTSP řídí doručovací kanály (UDP, multicast UDP a TCP) nebo doručovací mechanismy používané v RTP streamech. To si lze představit jako dálkové ovládání pro streamování videí s možností spouštění, zastavení, pozastavení a mnoha dalších operací. Jelikož RTSP je nestavovým typem protokolu (bez ohledu na to, že funguje nad transportním protokolem TCP), je stav udržován na serveru pomocí identifikátoru relace. Stavy serveru jsou např.: **Setup (rezervní zdroje), Play/Record, Pause a Teardown**. Během multimediálního přenosu může RTSP využívat buď UDP přenos, nebo několik TCP spojení. Ačkoliv RTSP nejčastěji řídí proudy odesílané přes RTP, lze k přenosu použít i jiné protokoly. Syntaxe a operace u RTSP jsou záměrně podobné jako u protokolu HTTP. To umožňuje snazší analýzu pomocí existujícího kódu a umožňuje tak přidat rozšíření HTTP do RTSP. RTSP může využívat oba transportní protokoly, které fungují na portu 554. Výměna RTSP zpráv se skládá z řady požadavků a odpovědí (zprávy Request a Response).

RTSP protokol zahrnuje následující metody:

- **DESCRIBE** slouží k vyžádání popisu prezentace či multimediálního obsahu identifikovaných příslušnou URL adresou.
- **GET_PARAMETER** tato metoda slouží k získání konkrétního parametru.
- **PAUSE** umožňuje dočasné zastavení všech multimediálních přenosů (oznámení serveru).
- **PLAY** umožňuje zahájení přenosu multimediálních dat. Poslání požadavku serveru, stejně jako v předchozím případě.
- **RECORD** spouští zaznamenávání multimediálního přenosu.
- **SETUP** tato metoda specifikuje, jakým způsobem bude multimediální přenos probíhat.
- **TEARDOWN** oznámení serveru k ukončení přenosu a celkové RTSP relace.
- **URL** definuje příslušnou RTSP adresu pro média objekt.
- **RTSP/1.0** tato položka identifikuje protokol a verzi. [15]

3.6 Standard MPEG

Ve své skutečné podobě standardy MPEG definují metody multiplexování pro přenos videa. Systémová vrstva MPEG poskytuje standardizovanou metodu poskytování integrovaných obrazových, zvukových a datových služeb. Dále definuje technologie pro kompresi videa, které dokáží překódovat digitální přenosy a snížit tak celkový objem dat přenosu. Například nekomprimovaný obrazový Standard Definition (SD) signál má bitovou rychlost 170 Mb/s a pomocí komprese je tato hodnota snížena na 2,5 - 4 Mb/s. Dalším příkladem může být obrazový HD signál, který je přenášen rychlostí 500 Mb/s v nekomprimované podobě. Pomocí komprimace je bitová rychlost snížena na 8-12 Mb/s. Datové toky zpracovávané MPEG se používají k vysílání dat souvisejících s programem.

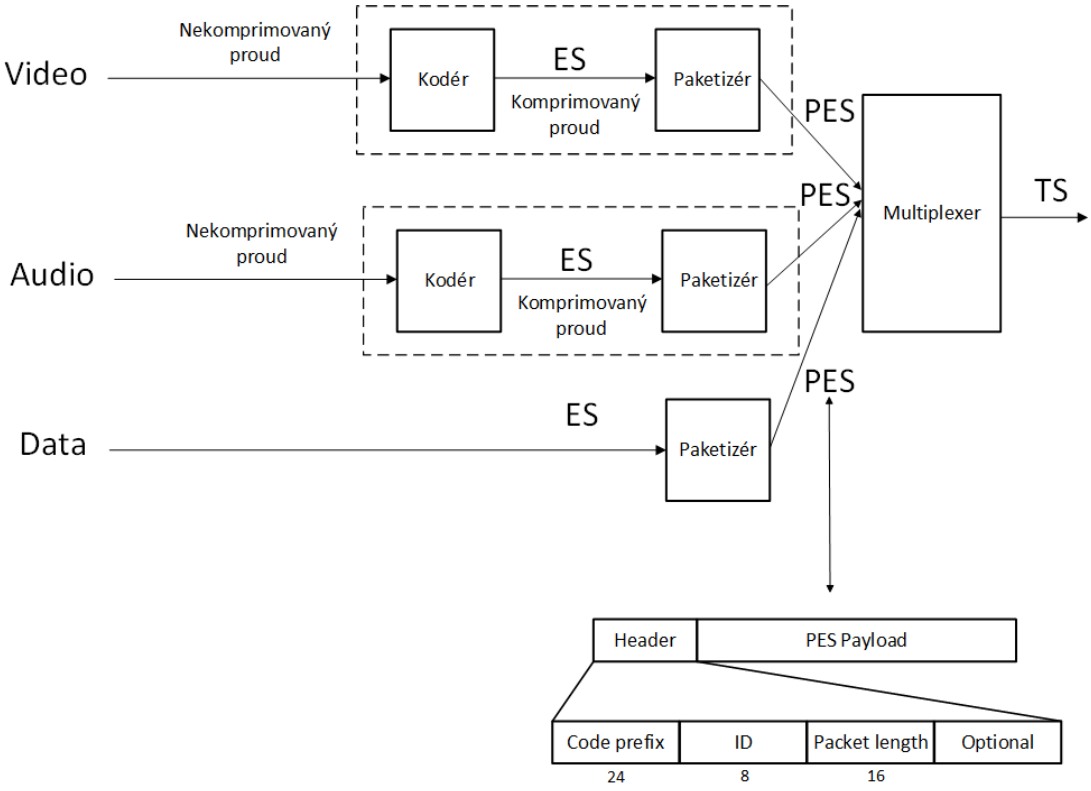
Nejzákladnější složkou pro MPEG je Elementary Stream (ES). Například televizní videoprogram obsahuje kombinaci ES pro video (obvykle jeden), jeden či více ES pro zvuk a také ES pro řídící data. Každý elementární stream generovaný na výstupu audio a videokodéru obsahuje pouze jeden typ dat. U obrazu a zvuku jsou informace ukládány do přístupových jednotek, z nichž každá představuje základní jednotku kódování. Paketizační funkce shromažďuje data jednotlivých proudů a řetězí je do jednoho výsledného. Na obrázku 3.3 je znázorněn základní princip. PES datagram je blok pevné (nebo proměnné) velikosti (o maximální velikost 65 535 bajtů). PES je obvykle organizován tak, aby obsahoval všechny elementární streamy všech přístupových jednotek.

MPEG-Program Stream (MPEG-PS) je v zásadě určen pro ukládání a načítání dat z paměťových médií. Podporuje seskupování video, audio a datových elementárních proudů, které mají společnou časovou základnu. Každý PS se skládá pouze z jednoho multimediálního obsahu. PS je používán v bezeztrátovém prostředí (např. DVD disky). Programový tok je utvořen jako skupina pevně spojených paketů PES.

MPEG-Transport Stream (MPEG-TS) je formát pro přenos komprimovaného videa. Tento formát je navržen pro poskytování dat v reálném čase přes nespolehlivé přenosové prostředí. MPEG-TS kombinuje více PES do jediného transportního toku. Multiplexovány jsou také zároveň informace o synchronizaci. Transportní tok současně segmentuje PES do menších TS paketů o pevné velikosti. Záhlaví PES obsahuje zásadní informace, díky nímž můžeme rozlišit pakety z jednotlivých proudů. Paketizované elementární streamy jsou generovány procesem paketizace. Přenášené informace těchto proudů se skládají z původních dat ES. Existují určitá omezení pro vytváření TS: první bajt PES musí být prvním bajtem TS, každý transportní paket musí obsahovat pouze data z jednoho PES paketu. Hlavní varianta MPEG-TS se nazývá Multiple Program Transport Stream (MPTS). MPTS je vytvářen kombinací

jednoprogramových transportních toků. Tento typ také obsahuje kontrolní informace potřebné ke koordinaci digitálního přenosu a jakýchkoliv dalších dat, která mají být odeslána.

Délka MPEG-TS paketu činí 188 bajtů, obsahuje i 4 bajtovou hlavičku. Klíčovým polem v hlavičce TS je 13 bitový Program Identifier (PID). Je to jednoznačný a unikátní identifikátor pro každý program. Pakety stejného ES obsahují stejné PID. Součástí transportních streamů jsou také metadata, která definují informace o tzv. Program Association Table (PAT). PAT obsahuje informace o všech programech. Každá položka v PAT odkazuje na programy uložené v Program Map Table (PMT). PMT následně odkazuje na ES programu. [7]



Obr. 3.3: Sloučení paketizovaných elementárních proudů do TS [7]

4 Přenosové parametry

Předpokladem úspěchu multimediálních služeb je subjektivní kvalita Quality of Experience (QoE) vnímaná uživatelem. Z tohoto důvodu je nutné zajistit, aby výsledná kvalita obrazu byla dostačující a bez výpadků. Přenášený video tok je závislý především na těchto kvalitativních parametrech: na přenosové rychlosti, zpoždění, jitteru a ztrátovosti paketů. Kvalita IPTV či jiné multimediální služby je zejména ovlivňována aktuálním stavem parametrů v počítačové síti. Kvalita služeb (QoS) je určena z technických vlastností sítě a dovoluje poskytovatelům následně kontrolovat a garantovat kvalitu nabízených služeb. Stanovení priorit QoS může zahrnovat nejen výše uvedené parametry, ale také např. bitovou chybovost, stanovení konstantní bitové rychlosti a určení kritických hodnot dalších parametrů, které se používají k zajištění dostatečné kvality přenosu.

Zpoždění je čas, který uplyne od odeslání paketu vysílačem až po jeho přijetí na přijímacím zařízení. Délka celkového zpoždění je součet dílčích zpoždění způsobených vysílačem (doba čekání na volný přenosový slot), zpožděním způsobeným délkou síťové trasy (doba cesty paketů sítí) a také zpožděním při zpracování paketu (směrování). Se zpožděním souvisí také parametr, který se nazývá **rozptyl zpoždění** (PDV). Rozptyl zpoždění udává rozdíl mezi maximálním a minimálním zaznamenaným zpožděním. PDV používají aplikace ke stanovení velikosti vyrovnávací paměti na přijímací straně.

Jitter (kolísání zpoždění) je nežádoucí náhodná hodnota. Představuje variabilitu v doručování paketů (změna zpoždění při přenosu). Změny v přenosové trase, způsobu přijímání, zpracování a přeposílání paketů ovlivňují hodnotu tohoto parametru. K eliminaci nežádoucích vlivů jitteru se používá vyrovnávací paměť. Do vyrovnávací paměti jsou pakety dočasně ukládány za účelem vytvoření rezervy. Je nutné zvolit vhodnou velikost, jinak při velkých změnách zpoždění může dojít k přetečení nebo podtečení této paměti.

Přenosová rychlost udává množství informace za jednotku času. Možnosti současných videoaplikací, jsou ovlivněny požadovanou přenosovou rychlostí. Přenosová rychlost při přenosu digitálního obrazu závisí na druhu použité komprese a použitém obrazovém formátu.

Propustnost udává kapacitu přenosového kanálu a je výchozím parametrem pro nabídku poskytovaných služeb. Tento parametr je reálnou hodnotou.

Šířka pásma je teoretická hodnota, které není možné z fyzikálního hlediska v praxi dosáhnout. Zvyšování šířky pásma má za následek snižování zpoždění.

Změna pořadí paketů vzniká v důsledku existence zpoždění i vlivem směrovací politiky. Pakety odeslané mimo pořadí mohou způsobit další zkreslení v obrazu, protože dekodér v přijímači může očekávat jinou sekvenci dat. Pakety mimo pořadí

je možné měřit. Udává se jako poměr mezi počtem paketů, jež mají byt přijaty k celkovému počtu opravdu přijatých paketů.

Ztrátovost paketů je průměrný počet zahozených paketů vyjádřený v % vzhledem k celkovému počtu přenesených paketů. IP sítě poskytují nespolehlivou a nespojovanou službu. V případě zahlcení sítě dochází k zahazování paketů. Tato ztrátovost má velký vliv na výslednou kvalitu přenášeného obrazu, jelikož každý IP paket může obsahovat až 7 bloků datového toku MPEG-TS. Klíčem ke kvalitnímu přenosu dat je určení vysoké priority pro pakety přenášející multimediální obsah a snížení jejich ztrátovosti.

Během přenosu multimédií prostřednictvím IP sítě mohou při zobrazování vznikat různé deformace zvuku a videa. Možné příčiny vzniku těchto chyb jsou uvedeny v tabulce 4.1. [6, 17]

Typ	**Deformace**	**Možná příčina**
Deformace zvuku	Ztráta synchronizace obrazu a zvuku	Ztráta synchronizace, chyba zařízení, špatné multiplexování
Deformace zvuku	Šum ve zvuku	Ztráta paketů
Deformace zvuku	Ztráta zvuku	Ztráta paketů
Deformace videa	Kostičkování obrazu	Ztráta paketů, malá přenosová rychlost
Deformace videa	Trhání obrazu	Ztráta paketů, vysoký jitter, chybný systémový čas, ztráta synchronizace
Deformace videa	Zamrznutí obrazu	Hlavně ztráta paketů, vysoký jitter, chybný systémový čas, ztráta synchronizace
Deformace videa	Žádný obraz	Výpadek přijímače, přílišná ztráta paketů, ztráta synchronizace
Deformace videa	Rozostření	Chyba kodéru
Deformace videa	Šum v obraze	Chyba kodéru

Tab. 4.1: Deformace multimediálních dat [17]

4.1 Bezdrátový přenos

V rámci síťového přenosu lze koncovým zařízením přenášet multimediální služby bezdrátovým způsobem. Vysokorychlostní bezdrátová IEEE 802.11n LAN síť může být jednou ze základních infrastruktur pro přenos IPTV služby. [3]

Kvalita bezdrátového signálu může být ovlivněna stavem bezdrátového kanálu. Základní parametry ovlivňující tento druh přenosu jsou např. síla signálu (RSSI), objem datového provozu, rušení nebo šum. Tyto parametry by měly být analyzovány, aby se předešlo případné ztrátě výkonu.

Parametry ovlivňující kvalitu bezdrátového přenosu:

- Fyzická rychlost - rychlost, kterou jsou bity skutečně odesílány. Fyzická rychlost souvisí s použitou modulací a použitým standardem IEEE 802.11.
- Síla signálu (Received Signal Strength Indication - RSSI) - určuje bezdrátový signál na příslušném místě (mezi stanicí a směrovačem). Tento parametr indikuje úroveň přenášeného signálu.
- Efektivní propustnost - je maximální množství dat, které lze bezdrátově přenést za jednotku času, závisí na použitém standardu 802.11.
- Rychlost přenosu dat uživatelům - závisí na aktivitách samotného uživatele.
- Velikost šumu - signalizuje velikost rušení okolních zařízení provozujících stejné rádiové pásmo jako lokální bezdrátová síť. Vysoká hodnota šumu znamená nižší poměr odstupu signálu k šumu, což může vést ke snížení datového objemu pro přijímací zařízení.
- Bezdrátové technologie v okolí - určují velikost rušení pro ostatní technologie a také ovlivňují samotné přenosy ostatních zařízení. [18]

5 Digitální videa

Rozlišení obrazového monitoru a videa se může značně lišit. Rozlišení obrazovky nebo displeje se vztahuje k celkovému počtu obrazových prvků (pixelů), které lze zahrnout do fyzické velikosti. Rozlišení videa se vztahuje k celkovému počtu pixelů v jednom snímku videa. Skutečné video je často zmenšeno (nebo převzorkováno), aby odpovídalo fyzické velikosti obrazovky. Pokud je velikost obrazovky mnohem větší než rozlišení videa, může být kvalita videa snížena. Naopak kvalita videa zůstává nedotčena, když je video s vysokým rozlišením mapováno na malé obrazovce (části videa mohou být oříznuty). I když se rozlišení televizní obrazovky v průběhu let zlepšilo, rozlišení statických snímků je stále na vyšší úrovni. Například rozlišení Ultra High Definition (UHD) obrazovky 3840x2160 je zhruba dvakrát nižší než u 16 megapixelového digitální fotoapáratu. [4]

5.1 Rozlišení videa

Přehled běžných rozlišení videa je uveden v tabulce 5.1. Rozlišení SD bylo z velké části nahrazeno vysokým rozlišením HD. Rozlišení HD 720p, Full HD 1080i a 1080p jsou zdaleka nejpoužívanějšími formáty. Obraz UHD-1 nabízí čtyřnásobný počet pixelů a dvojnásobné horizontální i vertikální rozlišení než běžný obrazový formát 1080p. Rozšiřující formát UHD-2 obsahuje čtyřnásobný počet pixelů oproti UHD-1. Rozdíl v kvalitě videa mezi formáty UHD-1 a UHD-2 není významný, avšak rozdíl mezi UHD-1 a 1080p je značný. Nově vznikající videa stále cílí na rozlišení pod 1080p z důvodu zpětné kompatibility a také z hlediska přenosových parametrů v IP sítích (menší nároky na přenos). [4]

Formát	**Rozlišení (px)**
UHD-2 4320p (progresivní)	7680x4320 (16:9)
UHD-1 extra široké (progresivní)	5120x2160 (21:9)
UHD-1 2160p (progresivní)	3840x2160 (16:9)
Full HD 1080p (progresivní)	1920x1080 (16:9)
Full HD 1080i (prokládané)	1920x1080 (16:9)
HD 720p (progresivní)	1280x720 (16:9)
SD	720x480, 720x576

Tab. 5.1: Běžné rozlišení videa [4]

HD video může obsahovat 720 nebo 1080 vertikálních progresivních řádků (720p a 1080p), 1080 prokládaných řádků (1080i) a je schopno vykreslit 16:9 obraz (se čtvercovými pixely) s výstupem jednoho nebo více kanálů digitálního zvuku. Pro

UHD a nové standardy kódování videa, jako je H.265/HEVC, je obraz zobrazován progresivním způsobem. Ačkoliv je poměr stran aktuálně specifikován na 16:9, mnoho monitorů podporuje poměr stran 21:9. U progresivního skenování jsou všechny aktivní řádky zobrazeny v každém snímku videa, zatímco u prokládaného zobrazení jsou liché a sudé řádky zobrazeny po sobě jdoucích snímcích s poloviční snímkovou frekvencí. Nevýhodou prokládání je, že horizontální rozlišení je sníženo na polovinu a video je často filtrováno, aby se zabránilo blikání a vykreslování pohybových artefaktů. Formát prokládaného videa je používán pro satelitní televizní vysílání, zatímco progresivní formát se používá při přenosu přes IP sítě. S používaným formátem videa je uváděna také snímková frekvence. Pro video 1080p se snímkovou frekvencí 60 Hz nebo 60 snímků za sekundu (FPS) se uvádí jako 1080p60. Pro UHD obraz jsou specifikovány 24, 50, 60 a 120 fps. Formáty digitálního videa jsou definovány z hlediska horizontálního rozlišení. Tato rozlišení se často zapisují jako násobky základní hodnoty pixelů pomocí přípony "K". 2K obraz tedy má 2048, formát 4K obsahuje 4096x2160 pixelů a formát 8K obsahuje 8192x4320 pixelů.

Obraz UHD-2 s rozlišením 7680x4320 obsahuje 33 milionů nebo 16krát pixelů obrazu 1080p. Toto rozlišení je stále mnohem nižší než rozlišovací schopnost lidského oka. Snímková frekvence progresivního vykreslování je 60 Hz, což odpovídá přenosové rychlosti v rozmezí 500 Mbit/s do 6,6 Gbit/s. Příkladem může být přenos IP sítí, kdy bylo video komprimováno z celkové přenosové rychlosti 24 Gbit/s na hodnotu asi 100 Mbit/s a přenosová rychlost zvuku byla snížena z hodnoty 28 Mbit/s na hodnotu 7 Mbit/s. [4]

5.2 Kompresní formáty

Přenosové rychlosti v počítačové síti se stále zvyšují. Vysokorychlostní připojení v rámci přístupových sítí je již nutností. Také nároky na paměťová média jsou větší než kdykoliv předtím. Je tedy zřejmé, že bez komprese multimediálních dat se již neobejdeme. Vývojáři jsou nuceni vynakládat nemalé úsilí na zlepšení daných kompresních formátů. Komprese videa přináší dvě důležité výhody. První výhodou je ušetření části paměťového média, ať už se jedná o osobní nebo webové úložiště. Pomocí kompresních formátů dochází ke znatelnému snížení celkového objemu dat. Uložit nekomprimované video by v některých případech bylo zcela nemožné. Druhá výhoda tkví v samotné kompresi videa (bitové rychlosti) a audia. Komprese videa umožňuje efektivnější využití přenosových prostředků. Pokud je k dispozici přenosové médium s velkou šířkou pásma poskytující vysokou propustnost dat, pak je výhodnější najednou přenášet několik komprimovaných videí vyšší kvality než vysílat pouze jediný nekomprimovaný tok s nízkým rozlišením. Kompresní formáty umožňují přenos kvalitního multimediálního obsahu přes běžná přenosová média, aniž by

došlo k vyčerpání celé šířky pásma. Pro nadcházející roky díky pokračujícímu vývoji se kompresní formáty stanou nezbytnou součástí multimediálních služeb. [19]

5.2.1 H.264/Advanced Video Coding (AVC)

Tento typ kompresního formátu byl představen v roce 2003. H.264 přinesl významné zvýšení kompresního poměru a umožňuje tím ušetřit až 50 % bitové rychlosti ve srovnání s předchozími standardy. Tento standard může zvýšit odolnost vůči chybám pomocí flexibilního kódování s reorganizací kódových dat. Zvýšení účinnosti kódování a flexibilita souvisí se zvýšením složitosti samotné struktury kódování (větší složitost algoritmů). H.264 může podporovat různé interaktivní (videotelefonie) a neinteraktivní aplikace (broadcastové vysílání, streamování, ukládání nebo VoD aplikace). Využívá vlastností předchozích standardů a přidává mnoho dalších kódovacích nástrojů a technik, které zvyšují účinnost komprese. Stejně jako předchozí formáty používá následující základní principy komprese videa:

- transformaci pro redukci prostorové korelace,
- kvantování pro řízení datového toku,
- pohybově kompenzovanou predikci pro snížení časové korelace,
- entropické kódování pro snížení statistické korelace.

Sám přináší nové vlastnosti:

- adaptivní intra-snímkovou predikci,
- transformaci malé velikosti bloku s celočíselnou přesností,
- variabilní velikost bloků,
- čtvrtinovou přesnost pro kompenzaci pohybu,
- vylepšené entropické kódování. [20]

5.2.2 H.265/High Efficiency Video Coding (HEVC)

HEVC představuje nejnovější standard kódování videa uveřejněný v roce 2013. Je nástupcem formátu AVC a oproti tomuto formátu snižuje datový tok o polovinu při zachování stejné obrazové kvality (bitové rychlosti). Existují tři profily: hlavní profil, hlavní profil 10 a hlavní statický obraz. Profily definují sady kódovacích nástrojů pro vytvoření výsledného datového toku. [20]

HEVC bitový tok se skládá ze sekvence datových jednotek nazývaných jednotky síťové abstraktní vrstvy NAL (Network abstraction layer). Každý snímek se skládá z tzv. slice, které obsahují makrobloky. Makroblok obsahuje veškeré informace o oblasti s příslušnou velikostí pixelů. Některé jednotky nesou informace na úrovni celé kódované videosekvence. Další jednotky mohou nést kódované vzorky právě ve formě slice nebo obsahují volitelné informace, jež jsou nápomocné při dekódování videa. Postup je následující. Do kodéru je přiveden vstupní obraz. Tento obraz je kodérem

převeden na bitový tok (sekvence jednotek NAL). Následně jsou tyto jednotky dekódovány dekodérem. Kodér i dekodér ukládají obraz do vyrovnávací paměti. Dříve uložené obrazy zůstávají případně využitelné z hlediska predikčního schéma. [21]

5.2.3 AOMedia Video 1 (AV1)

AOMedia Video 1 (AV1) byl vydán v roce 2018 a je nástupcem formátu VP9. Ve srovnání s tímto formátem je kompresní formát AV1 účinnější. Umožňuje snížení bitové rychlosti o více než 30 %, což má za následek snížení nároků na přenosovou šířku pásma a paměťová úložiště. Vyšší účinnost komprese je vykoupena složitější a výpočetně náročnější strukturou kodéru. Zrychlení kódování videa při zachování samotné kvality je velmi náročné. Optimalizovaný kodér, nazývající se Libaom AV1, nabízí skvělé řešení pro VoD aplikace, jelikož kódovací proces zabere více času. Samotný princip je založen na blokově frekvenční transformaci. [22]

Obr. 5.1: AV1 Libaom kodér [22]

5.2.4 Video Predictor version 9 (VP9)

V roce 2011 firma Google zahájila vývoj bezplatného kodeku VP9 (dokončen v roce 2013) se zaměřením na videa s vysokým rozlišením (HD a UHD). Tento vývoj probíhal paralelně s vývojem formátu HEVC. Formát byl vytvářen s vědomím, že bude využíván u služby Youtube. Jedná se o blokově orientovaný formát s vylepšením velikosti bloku na 64x64 pixelů. Cílem bylo vytvořit účinný kompresní formát pro snížení velkého množství dat, aby nedocházelo ke zhoršení kvality videa. Z tohoto důvodu v počáteční fázi kódování dochází k predikci (krokům inter-snímku a intra-snímku) následovanou transformací a kvantizací. V konečné fázi dochází k entropickému kódování. Intra-snímková predikce používá ke generování sousední vzorky aktuálního bloku (uvnitř aktuálního snímku). Inter-snímková predikce používá ke generování dříve zakódované snímky. Uvnitř předchozího snímku dochází k hledání takového bloku, který je nejvíce podobný tomu aktuálnímu. V rámci této predikce také dochází k odhadu pohybu. [23]

6 Testovací model přístupové sítě

V této kapitole je již popsána praktická část. Kapitola je členěna do jednotlivých podkapitol. Jednotlivé podkapitoly popisují dílčí části, které společně vytváří funkční síťový celek. Nejdříve je popsán hardware jednodeskového počítače Raspberry Pi, jenž má funkci serveru a uživatelské stanice. Další podkapitola se týká parametrů síťové topologie popisující strukturu dané sítě. Následuje výčet a rozbor síťových nástrojů využitých pro testovací sadu. Jako poslední je popsán vytvořený instalační skript, jenž slouží k uživatelsky snadné instalaci všech potřebných nástrojů. Tyto nástroje tvoří základní funkcionalitu testovací měřicí sady, která je stěžejním výstupem této diplomové práce.

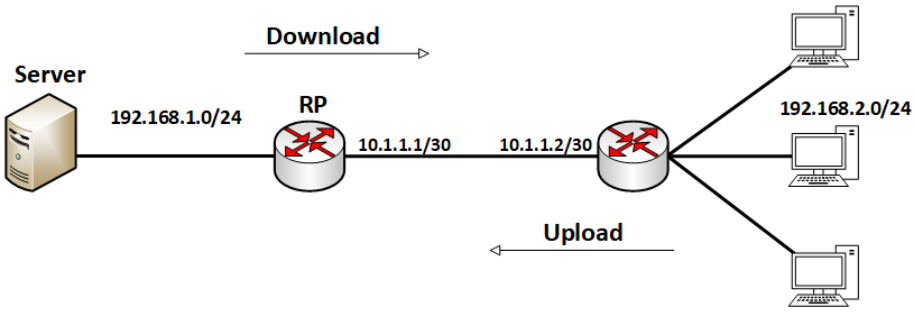
6.1 Hardware jednodeskového počítače

K účelu této práce byl vybrán jednodeskový počítač Raspberry Pi model 4B. Konstrukce desky plošného spoje je podána v minimalistické velikosti odpovídající velikosti např. osobního průkazu. Raspberry PI disponuje dvěma obrazovými výstupy micro High-Definition Multimedia Interface (HDMI). Pomocí těchto obrazových výstupů je možné tento typ počítače využívat za účelem přehrávání až 4K rozlišení či přenosu multimediálního obsahu. Zvukový výstup je řešen pomocí čtyřpólového 3,5 mm jack konektoru. Napájecí konektor typu USB-C umožňuje vyšší výkon při napájení. K dispozici jsou celkem čtyři USB konektory (2x USB 3.0 konektor a 2x USB 2.0 konektor) umožňující připojení vnějších periférií. Ze síťového hlediska je velmi důležité, že se zde nachází plnohodnotný gigabitový Ethernet port. V jeho bohaté výbavě dále nechybí ani moduly pro Wi-Fi 802.11b/g/n/ac s podporou jak 2.4 GHz, tak 5 GHz a Bluetooth 5.0. Jeho základ představuje procesorová architektura typu ARM. Tato architektura se vyznačuje nízkou spotřebou. Deska modelu 4B je osazena 64 bitovým čtyřjádrovým procesorem ARM v8 Broadcom BCM2711 s taktovací frekvencí 1,5 GHz. Procesor obsahuje jádro Cortex-A72, které umožňuje provádět 64/32 bitové instrukce mimo pořadí. Procesor BCM2711 obsahuje hardwarovou podporu pro dekódování High Efficiency Video Coding (HEVC) 4K60FPS videa a dekódování AVC 1080p60FPS či 1080p30FPS videa. Velikost použité Random Access Memory (RAM) paměti odpovídá 8 GB (LPDDR4-2400 SDRAM). Na všech počítačích je nainstalován operační systém Linux Ubuntu ve verzi 22.04.1 (Jammy Jellyfish).

6.2 Síťová topologie

Pro testování funkčnosti měřicí sady byla vytvořena fyzická topologie, která je znázorněná na obrázku 6.1. Stanicím a routerům byly nastaveny IPv4 adresy. Dále z důvodu ušetření režie byl na rozhraních deaktivován protokol IPv6. Aby na žádném ze zařízeních nedocházelo k blokování přenosu, byla nastavena příslušná firewallová pravidla (příkaz *sudo ufw*). Jednotlivé prvky byly propojeny strukturovanou kabeláží UTP Cat 5.e. Tato kabeláž umožňuje propustnost až 1 Gbit/s. Počítačová síť je sestavena z tzv. Self-Office Home-Office (SOHO) zařízení od firmy Mikrotik. Tato síť poskytuje propustnost 100 Mbit/s, a to v obou směrech přenosu. Zvolená propustnost je pro praktické testování dostačující. Mezi jednotlivými zařízeními je zajištěna komunikace pomocí dynamického směrovacího protokolu OSPF. Topologie obsahuje celkově 2 routery. Součástí topologie jsou celkově čtyři jednodeskové počítače. Stanice nacházející se v podsíti 192.168.1.0/24 má roli serveru a generuje síťový obsah do podsítě 192.168.2.0/24, v níž se nachází tři koncové stanice. V tomto případě se jedná o přenos ve směru download. Přenos dat může probíhat i v opačném směru (upload).

V rámci této topologie bylo nakonfigurováno vícesměrové vysílání. Pro zajištění správného fungování multicastového přenosu byly použity protokoly IGMPv2 a PIM. IGMPv2 byl nasazen do obou podsíti. Pokud nějaká stanice měla zájem přijímat multicastový obsah, zaregistrovala se pomocí tohoto protokolu do příslušné skupiny a mohla tak přijímat daný obsah. Přenosová cesta vícesměrového vysílání byla typu sdíleného stromu (Shared Tree). V síti existoval pouze jediný logický distribuční strom. Jako sdílený kořen stromu (RP) byl určen hraniční router serverové stanice (IPv4 adresa 10.1.1.1), který se staral o veškerý vícesměrový přenos. Protokol PIM k určení distribučního stromu využil veškerých směrovacích informací zajištěných protokolem OSPF. Multimediální přenos byl zprostředkováván protokolem RTP/MPEG Transport Stream.



Obr. 6.1: Síťová topologie

6.3 Síťové nástroje

V této podkapitole jsou popsány jednotlivé open-source nástroje, jež zprostředkovávají jednotlivé funkce síťového přenosu. Pro multimediální přenosy byl vybrán software VideoLAN media player (VLC). Pro generování datového přenosu byl zvolen program Netcat a nástroj příkazového řádku Ping. K zachytávání a analýze dat byly nainstalovány aplikace Wireshark a dále jeho terminálová verze TShark. Vykreslení přenášených dat v reálném čase probíhá pomocí programu Smag.

Příkaz **ifconfig** slouží ke konfiguraci a zobrazení parametrů síťového rozhraní. Tento příkaz je také velmi užitečný při ladění celého systému. Je možné ho získat z balíčku *net-tools*. Příkaz **ethtool** je síťovým nástrojem operačního systému Linux. Používá se ke konfiguraci síťových rozhraní nebo také k vyhledávání různých informací (údajů o ovladačích, apod.).

VideoLAN client (VLC) media player představuje svobodný multimediální program, který nabízí širokou škálu možností. VLC je k dispozici ve dvou verzích, pro Linux a Windows. Tento program obsahuje funkce pro přehrání různých formátů videa či audia. Podporuje různé kompresní formáty a také multimediální protokoly, jež umožňují přenášet tento druh dat přes síťovou strukturu. Dále poskytuje pokročilé ovládání, např. synchronizaci videa a titulků.

FFmpeg je multimediální framework poskytující funkce jako kódování, dekódování, transkódování, multiplexování nebo demultiplexování multimediálních dat a v neposlední řadě jejich streamování či přehrávání. Tento framework je možné nasadit napříč různými operačními systémy. FFmpeg lze použít pro operace úpravy videoobsahu (komprese, odstranění šumu, konverze barev, změna snímkové frekvence a další). FFmpeg obsahuje velké množství různých knihoven, jež nabízejí rozhraní pro různé kodeky (např. H.264/AVC, H.265/HEVC, AV1 a VP9). Pro zabalení videoobsahu nabízí rozsáhlou podporu kontejnerů jako jsou avi, mp4, mp3, wma, mkv, ts a mnoho dalších. Existuje nástroj příkazového řádku vycházející z knihovny FFmpeg, nazývá se **ffprobe**. Jedná se o analyzátor multimediálního proudu. Pomocí tohoto nástroje lze extrahovat různé informace o multimediálním obsahu. Dále existuje nástroj příkazového řádku, který se nazývá **ffplay**. Jak již z názvu vyplývá, slouží k přehrávání videí.

Netcat (unixový příkaz nc) je obslužný nástroj pro čtení a zápis pro TCP či UDP přenos. Multiplatformní nástroj Netcat je možné nainstalovat na různých typech operačních systémů. Netcat nabízí několik funkcí: skenování portů, sledování provozu na daném portu a přenos dat.

Wireshark je multiplatformní software, který slouží k zachytávání a analýze síťového provozu. Tento nástroj umožňuje grafickou reprezentaci jednotlivých parametrů. Nabízí mnoho funkcí pro analýzu komunikačních protokolů. V grafickém

rozhraní lze data různě filtrovat a zobrazovat. Wireshark také zahrnuje terminálovou verzi nazývanou **TShark**. TShark umožňuje dle příslušných přepínačů nastavit parametry pro záznam měření. Pomocí něj lze také číst jednotlivé pcap soubory, nebo provádět různé převody mezi druhy souborových formátů. V neposlední řadě podporuje funkci pro generování paketů.

Ping je síťový nástroj pro testování funkčnosti spojení mezi dvěma zařízeními. Ping periodicky zasílá data a následně očekává odezvu od protistrany. Pomocí tohoto mechanismu lze určit obousměrné zpoždění, tzv. Round Trip Time (RTT).

Smag je nástroj příkazového řádku pro vytváření grafů v reálném čase. Pokud pracujeme pouze v textovém prostředí a potřebujeme odstranit různé problémy síťového charakteru pak zobrazení dat pomocí tohoto programu je ideálním řešením. Smag umožňuje vykreslení dat pro zřetězené příkazy, které vytvoří jeden číselný výstup. Tento výstup je poté vykreslen na ose y.

Youplot je nástroj příkazového řádku vykreslující síťová data ze zvoleného souboru. Tento nástroj se využívá v kombinaci s dalšími linuxovými příkazy, aby bylo dosaženo požadovaného grafického výstupu.

6.4 Instalační skript

Pro pohodlnou uživatelskou instalaci všech potřebných nástrojů použitých ve vytvořené testovací sadě, byl vytvořen instalační skript. Tento skript je rozdělen do dvou částí. Tyto části jsou logicky děleny pomocí dvou nadefinovaných funkcí. Funkce se nazývají *fst_inst()* a *inst_menu()*.

Výpis 6.1: Úvodní instalace

```
fst_inst(){
sudo apt update && sudo apt upgrade
mkdir /home/$dir/DP
mkdir /home/$dir/DP/videos
mkdir /home/$dir/DP/files
mkdir /home/$dir/DP/measurement
sudo mkdir /home/capture
finder1=$(find $(pwd) -name dp.sh)
finder2=$(find $(pwd) -name installation.sh)
cp $finder1 /home/$dir/DP
cp $finder2 /home/$dir/DP
... inst_menu }
```

Funkce *fst_inst()* je volána pouze před první instalací zadáním příkazu ***source installation.sh; fst_inst***. Její podoba je uvedena ve výpisu 6.1. Hned na začátku

se nachází dva příkazy *update* a *upgrade*. Uvedené příkazy provedou kompletní aktualizaci operačního systému Linux. Následují příkazy *mkdir*, které vytvoří potřebnou adresářovou strukturu. Bez těchto složek by testovací sada nefungovala správně. V domovském adresáři je vytvořena výchozí složka pojmenovaná výstupem proměnné *$USER* (jméno uživatele). Skripty *dp.sh* a *installation.sh* jsou nalezeny pomocí příkazu *find* v příslušné části počítače a příkazem *cp* jsou zkopírovány do nově vytvořené struktury. Poslední řádek zavolá hlavní menu nazvané *inst_menu*.

Hlavní menu pro instalaci dílčích nástrojů (výpis 6.2) lze také spustit zadáním příkazu ve tvaru ***source installation.sh; inst_menu***. Uživatel má na výběr, který nástroj si bude chtít nainstalovat. Pro kompletní funkčnost testovací sady by však neměl vynechat ani jeden. Součástí jednotlivých voleb jsou příkazy pro instalaci daných nástrojů nebo také příkazy pro instalaci podpůrných programů třetích stran. Princip je jednoduchý. Uživatel si zvolí číslo volby. Vytvořená logika je stejná jako v případě uživatelského menu uvedeného v testovací sadě. Zadaná hodnota je přes příkaz *read* uložena do pole *array*. Toto pole je vyčítáno pomocí cyklu *for*. Dále je tato hodnota předána do podmínky *case*, v níž dochází k větvení všech voleb. Podmínka *case* je jednodušší forma podmínky *if elif else*. Volba **)* slouží pro ošetření situace, při níž uživatel zadá volbu mimo rozsah. V tomto případě se uživateli znovu zobrazí úvodní menu. Ukončení *case* a *for* cyklu značí *esac* a *done*. Přehled jednotlivých voleb pro instalaci nástrojů je uveden níže:

- volba 1 - *ifconfig* a *ethtool*,
- volba 2 - *netcat* a *jot*,
- volba 3 - *vlc* a *ffmpeg* (+podpůrné programy),
- volba 4 - *tshark*,
- volba 5 - *youplot*,
- volba 6 - *smag*,
- volba 7 - ukončení instalace.

Výpis 6.2: Úryvek instalačního menu

```
inst_menu(){
...
read -p "Enter the value: 1, 2, 3, 4, 5, 6 or 7:" -a array
for dataSh in "${array[@]}"
case $dataSh in
1)
sudo apt install net-tools #ifconfig
sudo apt install ethtool
break
;; ... }
```

Zvolením třetí volby (viz výpis 6.3) jsou nainstalovány programy *vlc* a *ffmpeg*, které se využívají při odesílání a zpracování multimediálních dat. Po dokončení úvodní instalace dvou zmíněných nástrojů dále dochází k instalaci podpůrných programů pro kompilaci a běh FFmpeg frameworku. Nejprve dojde k instalaci potřebných balíčků, jež mohou být po dokončení celé instalace v případě potřeby odstraněny. Příkazem *mkdir -p <název složky>* dojde k vytvoření výchozí složky v domovském adresáři. Následuje kompilace a instalace knihoven třetích stran. Tyto knihovny jsou potřebné pro správné fungování v rámci komprese multimediálních dat. Pro správné fungování uvedených kompresních formátů jsou důležité především knihovny *libx264*, *libx265*, *libvpx*, *libfdk-aac* a *libaom*. Většinu knihoven je možné nainstalovat pomocí správce balíčků. Balíčky, které se nenachází v tomto správci, je nutné nainstalovat pomocí zdrojových kódů z GitHub serveru, nebo také z privátních serverů daných projektů. V závěru volby jsou uvedeny příkazy ke zkompilování nově nainstalovaných knihoven s programem FFmpeg.

Výpis 6.3: Úryvek třetí volby instalačního skriptu

```
3)
...
sudo apt-get update -qq && sudo apt-get -y install \
cmake build-essential \
...
sudo apt install libunistring-dev libaom-dev libdav1d-dev
mkdir -p ~/ffmpeg_sources ~/bin
sudo apt-get install libx264-dev
sudo apt-get install libx265-dev libnuma-dev
sudo apt-get install libvpx-dev
sudo apt-get install libfdk-aac-dev
...
PATH="$HOME/bin:$PATH" PKG_CONFIG_PATH="$HOME/ \
ffmpeg_build/lib/pkgconfig" ./configure \
...
--enable-libaom \
--enable-libvpx \
--enable-libx264 \
--enable-libx265 \
...
break
;;
```

7 Testovací sada

Testovací sada byla vytvořena ve skriptovacím jazyce Bash. Tento jazyk představuje základní rozhraní pro komunikaci uživatele s operačním systémem počítače. Příkazy jsou vykonávány sekvenčně. K reprezentaci této sady byl zhotoven jeden velký skript nazývaný *dp.sh*. Součástí tohoto souboru jsou veškeré síťové nástroje. Tyto nástroje jsou implementovány pomocí dílčích funkcí. Funkce je úsek kódu, který lze volat opakovaně. Namísto vypisování stejného kódu je výhodnější napsat kód jednou a poté až tuto funkci volat dle potřeby. Volání funkce probíhá napsáním příkazu ve formě jejího názvu (bez závorek). Na začátku celého skriptu se nachází znaky *#!*, které značí, že se jedná o skript, za nímž následuje cesta (*bin/bash*) k interpretu příkazů. Následuje řádek s definicí proměnné *dir=$(echo $USER)*. Příkaz *echo $USER* vypíše jméno aktuálně přihlášeného uživatele. Uživatel musí mít vytvořenou potřebnou adresářovou strukturu (pomocí skriptu *installation.sh*). Hodnota uložená v *$dir* slouží k určení výchozí složky pro všechny operace. Dále již následují funkce měřicí sady definované v příslušných funkcí. V níže uvedených podkapitolách jsou uvedeny stěžejní části naprogramovaných funkcí diplomového skriptu.

7.1 Metodika měření

Spuštění testovací sady probíhá zadáním příkazu *source dp.sh; menu*. Po zadání tohoto příkazu se uživateli spustí ucelené uživatelské rozhraní. Ten má následně dvě možnosti, zda se chce vydat cestou síťového přenosu nebo měřením dat. Metodika měření je graficky reprezentována pomocí stromové struktury uvedené na obrázku 7.1.

Pro síťový přenos byly vytvořeny generátory multimediálního a datového přenosu. **Multimediální generátor** je dvojího typu, pro unicastový a multicastový přenos. Před zahájením samotného přenosu může dojít k úpravě vlastností videoobsahu. Tato sada nabízí různé varianty, jež využívají vlastností nástroje **FFmpeg**. V této části se nachází volba pro zobrazení metadat videa. Úprava dat poté může probíhat dle těchto informací. Uživatel si zde vybírá ze tří variant. V jedné z nich může dojít ke změně formátu kódování. Účinnější kompresní formát dokáže zmenšit bitový tok i na polovinu, což ve finále značně snižuje přenosové nároky. Další snižování požadavků na přenos lze také zajistit úpravou snímkové frekvence nebo změnou rozlišení videa. Koncový uživatel si také může zvolit, jak dlouhý segment videa bude chtít použít. Další popis se již týká generátoru. V rámci síťového přenosu si lze zvolit povinné parametry: cílové IP adresy a porty. Dále je uživatel dotazován, zda chce uskutečnit hromadný přenos nebo přenos pouze jednoho videa. Aby bylo

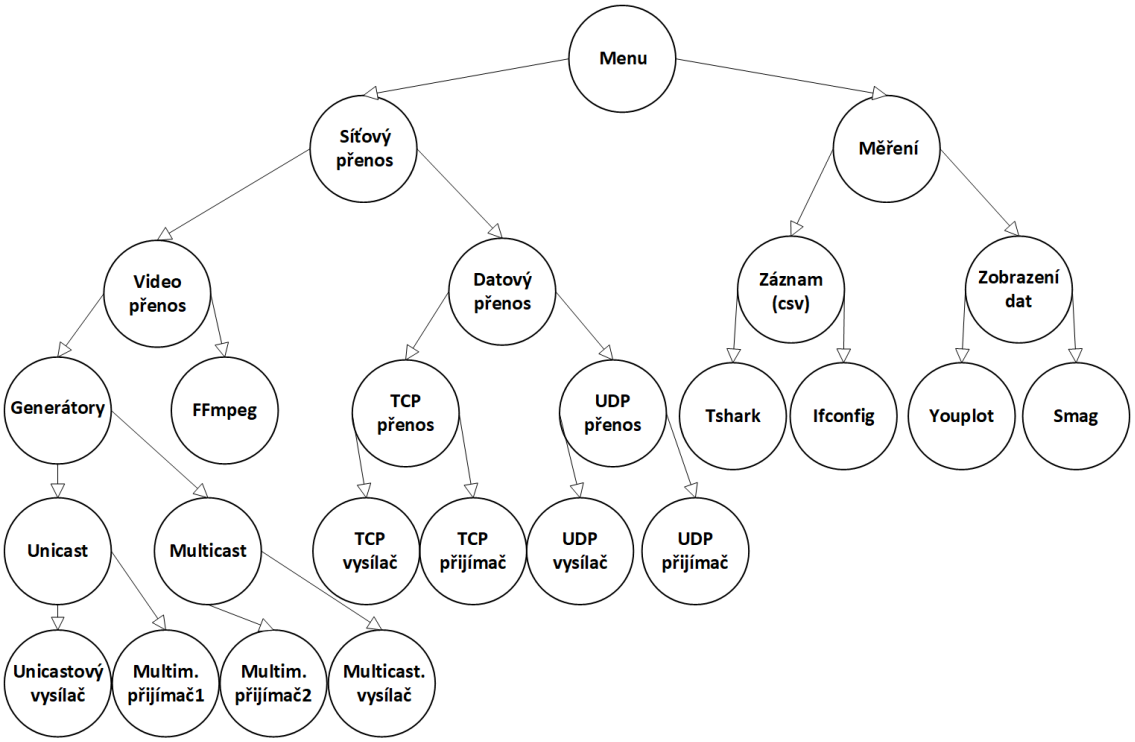
možné generátor využívat, jak v grafickém tak v konzolovém prostředí, byla naprogramována možnost pro výběr rozhraní (grafického nebo konzolového). V závislosti na zvoleném typu generátoru existuje výběr režimu generování. Zvolením režimu si uživatel určí, jakým způsobem chce posílat multimediální data. Streamování může probíhat až na tři koncové stanice zároveň. Pro situaci, kdy chce koncová stanice přijímat videoobsah, byl vytvořen multimediální přijímač. Na obrázku 7.1 je přijímač rozdělen na dvě části, avšak v diplomovém skriptu se nachází pouze jedna funkce. K tomuto kroku došlo z důvodu dodržování pravidel pro tvorbu stromové struktury. Po spuštění přijímače je uživatel vyzván, aby zadal IP adresu a port. Na této adrese a portu bude naslouchat a přijímat multimediální data. **Datový generátor** nabízí dvě možnosti přenosu. Přenosy jsou rozlišeny dle použitého transportního protokolu. TCP protokol nabízí spolehlivé a spojované odesílání, zatímco UDP upřednostňuje rychlost před spolehlivým přenosem dat. Poté záleží už jen na uživateli, jaké vlastnosti bude preferovat, zda zvolí spolehlivost před rychlostí a naopak. Uživatel zadá jméno předlohy, IP adresu a port. Daná předloha by měla obsahovat požadovanou bitovou rychlost. Uživatel si sám zvolí z jakého sloupce budou tyto data vyčítána. Nástrojem *jot* jsou vytvořeny soubory požadované velikosti a ty jsou poté přenášeny ke koncovému účastníkovi. Na přijímací straně se musí nacházet nástroj, který bude naslouchat na daném portu a bude odesílat odpovídající signalizační zprávy, tak aby byl zajištěn přenosový mechanismus TCP protokolu. Pro UDP příjem také dochází k naslouchání na požadovaném portu, avšak obsah datového souboru je ihned zobrazován v otevřeném oknu terminálu. Po spuštění přijímacího skriptu je uživatel vyzván k zadání portu. Po zadání této volby se uživatel může rozhodnout, zda chce ve firewallu povolit komunikaci na zadaném portu, či nikoliv. Postup je pro oba přijímače stejný.

Měření přenosových parametrů může probíhat ve dvou různých variantách. Varianta ukládání změřených dat do csv souborů je vhodná především pro pozdější hloubkovou analýzu ve větším vzorku dat. Měření v reálném čase je výhodné, pokud uživatel potřebuje ihned odhalit příčinu konkrétního problému v dané situaci.

V rámci první varianty se uživateli naskytují dvě možnosti s využitím nástroje *tshark* nebo *ifconfig*. Spuštěním analyzátoru ***tshark*** se objevují čtyři možnosti. Uživatel může zapnout zachytávání paketů. Toto měření je následně uloženo do souboru s příponou *pcap*. Volbou číslo dva lze provést konverzi do csv formátu. Zvolením třetí možnosti jsou zobrazeny RTP statistiky přímo do terminálového okna. Poslední možnost se týká měření bajtové rychlosti. Bajtová rychlost může být dále přepočítána na bitovou rychlost s výsledným uložením do csv souboru. Zřetězením příkazu ***ifconfig*** a dalších linuxových příkazů lze vytvořit požadovaný výstup (bajtovou rychlost, bitovou rychlost, počet přenesených paketů nebo obousměrné zpoždění). Tento výstup je poté ukládán do souboru, jehož jméno je zvoleno uživatelem. V prvním sloupci

souboru je uložen čas záznamu. Čas je inkrementován s každým změřeným vzorkem cyklem *for* od hodnoty jedna.

Finální soubory csv, vytvořené ať už s pomocí *tshark* nebo variantou s *ifconfig*, lze graficky zobrazit příkazem ***youplot***. Uživatel zapnutím této volby může vidět grafické závislosti založené na hodnotách, jež jsou uložené v těchto souborech. Vykreslování v reálném čase probíhá pomocí programu **smag**. Uživatel si může zobrazit průběh stejných parametrů jako při variantě s ukládáním do souborů. Zde však nedochází k ukládání, ale k okamžitému vykreslování přímo do terminálu. Ze zobrazených grafických závislostí lze vyčíst průběh základních přenosových parametrů.



Obr. 7.1: Stromová struktura pro grafickou reprezentaci metodiky měření

7.2 Uživatelské rozhraní

Uživatelské rozhraní (obrázek 7.2) je výchozím bodem celé testovací sady. Diplomový skript by měl být spuštěn zadáním příkazu ***source dp.sh; menu***. Uživatelské rozhraní umožňuje snadné a pohodlné spuštění dané funkce. Konkrétní operace může být samozřejmě volána ihned bez načtení tohoto rozhraní, avšak uživatel musí disponovat bližší znalostí skriptu.

V úvodní části je deklarován nekonečný cyklus *while*. Tento cyklus probíhá do té doby, dokud jej sám uživatel neukončí. Pomocí ANSI kódu jsou zde definovány tři

barvy a zakončující hodnota. Tyto barvy slouží ke zpřehlednění textových výstupů. Následují příkazy *echo*, které slouží k textovému zobrazení nabídky voleb. Nabídka má následující podobu:

- volba 0 - zobrazení sítových adres,
- volba 1 - přenos dat pomocí protokolu TCP,
- volba 2 - přenos dat pomocí protokolu UDP,
- volba 3 - FFmpeg - úprava videoobsahu,
- volba 4 - unicastový multimediální přenos,
- volba 5 - multicastový multimediální přenos,
- volba 6 - přijímání TCP přenosu,
- volba 7 - přijímání UDP přenosu,
- volba 8 - multimediální přijímač,
- volba 9 - měření pomocí Tshark,
- volba 10 - zobrazení přenosových parametrů v reálném čase (smag),
- volba 11 - měření parametrů: přijaté/odeslané pakety, přijaté/odeslané bajty, přijaté/odeslané bity a obousměrné zpoždění,
- volba 12 - grafické zobrazení dat (youplot),
- volba 13 - ukončení běhu skriptu.

```
tomaske-receiver@tomaskereceiver:~/DP$ source dp.sh; menu
Open the users menu -> command: source dp.sh; menu
********-------------------------*******
Select mode to run:
0 - Display the current network interface configuration (IP and MAC addr.)
1 - Transmission data - TCP protocol
2 - Transmission data - UDP protocol
3 - FFMPEG functionality (Encoding,..)
4 - Unicast multimedia transmission
5 - Multicast multimedia transmission
6 - Receive data - TCP protocol
7 - Receive data - UDP protocol
8 - Multimedia receiver
9 - Tshark (capturing, csv conversion, byte rate, bit rate and RTP statistics)
10 - Smag - show me a graph (measurement of various network parameters)
11 - Measurement - Packet/s, Byt/s, Bit/s and Ping -> save data to csv file
12 - Youplot (create a graph for parameters from option 11)
13 - Exit this script
********-----------------------*******
Enter the value: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 or 13:
```

Obr. 7.2: Uživatelské rozhraní

Příkaz *read* přečte vstup zadaný z klávesnice. Parametr tohoto příkazu *-p* značí, že se jedná o řetězec výzvy. Výstup je poté vložen do pole *array*. Toto pole je procházeno cyklem *for*, který přečte aktuálně uložený prvek a vloží ho do proměnné *dataSh*. Tato proměnná je následně předána do podmínky *case*. Podmínka *case* obsahuje celkem třináct různých možností. Jednotlivé volby jsou porovnávány s proměnnou

dataSh, jež obsahuje číslo volby zadané z klávesnice uživatele. V případě, že se hodnota volby rovná hodnotě proměnné *dataSh*, je provedeno tělo zvolené možnosti. Po zadání hodnoty v rozsahu je otevřena nová záložka terminálového okna. V nově otevřené záložce běží nově zvolená funkce. V původním okně je stále spuštěno uživatelské menu. K otevírání záložky terminálu slouží příkaz *gnome-terminal*. Tento příkaz je definován v podobě *gnome-terminal –title=""–tab – bash -i -c ""*. Parametr *title* specifikuje název nově otevřeného prvku, parametr *tab* vytvoří novou záložku a příkaz *bash* je definován s dvěma přepínači. Přepínač *i* značí interaktivní shell a přepínač *c* spouští sekvenční provádění příkazů. Pokud by bylo zvoleno špatné číslo, provede se volba definovaná symbolem ***. Uživateli je vypsána chybová hláška a následně tak může znovu vybírat v rámci menu. Zadáním volby třináct je skript ukončen (příkaz *exit*). Úryvek uživatelského menu je uveden ve výpisu 7.1 níže.

Výpis 7.1: Nabídka uživatelského rozhraní

```
menu(){
while true do
RED="\e[31m"
GREEN="\e[32m"
BLUE="\e[34m"
END="\e[0m"
...
read -p "Enter the value: 0, 1, 2, 3, 4, 5, 6, 7, 8,
9, 10, 11, 12 or 13: " -a array
for dataSh in "${array[@]}" do
case $dataSh in
0)
echo -e "${GREEN}Option 0:Network interface \
configuration ${END}"
gnome-terminal --title="Network configuration" --tab \
-- bash -i -c "source dp.sh; net_param"
break
;; ...}
```

Volby číslo devět, deset a jedenáct mají funkcionalitu rozdělenou do více částí. Volání těchto částí musí být patřičně ošetřeno. Zmíněné volby mají téměř identickou strukturu, která se liší pouze v detailech a proto bude popsána pouze desátá volba uživatelského rozhraní (viz výpis 7.2). Proměnná *i* vyjadřuje počet průchodů cyklu. Následuje nekonečný *for* cyklus. Tento cyklus je možné ukončit příkazem *break*. V těle cyklu je inkrementována proměnná *i=$(($i+1))*. Po této inkrementaci následuje podmínka zajišťující, zda hodnota *$i* je rovna pěti. Pokud

ano, *for* cyklus je ukončen a pokračuje se v kódu definovaném za cyklem. Pokud ne, pokračuje se v provádění kódu definovaného v těle nekonečného cyklu. Uživatel je vyzván k zadání volby v rozsahu písmen a-e. Po tázacím příkazu *read* následuje podmínka *if else*. Podmínka *if* testuje, zda zadaná hodnota odpovídá písmenu a, b, c, d nebo e. V případě pokud byla vybrána jedna z těchto voleb, pokračuje se v rámci těla této *if* podmínky a pomocí vnořené podmínky *if elif elif elif else* se zjišťuje, které písmeno bylo přesně vybráno. Pokud dojde ke shodě v jednom z bloku, provede se příkaz uvedený právě uvnitř bloku. V opačném případě je zadána volba mimo rozsah, a tím pádem se provede blok *else*. V *else* bloku je dekrementována proměnná *i=$(($i-1))* tak, aby nedošlo ke snížení počtu pokusů při výběru. Nekonečný *for* cyklus je prováděn do té doby, než jsou spuštěny všechny funkce, resp. dokud se proměnná *i* nerovná hodnotě pět. Jakmile nabude této hodnoty, provede se příkaz *break* uvedený v podmínce *if [$i -eq 5]*.

Výpis 7.2: Uživatelské menu - volba číslo 10

```
... for (( ;; )) do
i=$(($i+1))
if [ $i -eq 5 ]; then
break  fi
read -p "Enter the value: a, b, c, d or e: " value
if [[ "$value" == "a" ]] || [[ "$value" == "b" ]] || \
[[ "$value" == "c" ]] || [[ "$value" == "d" ]] || \
[[ "$value" == "e" ]]; then
if [ "$value" == "a" ]; then
gnome-terminal --title="Transmission (packets/s)" --tab \
-- bash -i -c "source dp.sh; packets_measurement"
elif [ "$value" == "b" ]; then
gnome-terminal --title="Throughput (bytes/s)" --tab \
-- bash -i -c "source dp.sh; byt_measurement"
elif [ "$value" == "c" ];  then
gnome-terminal --title="Throughput (bits/s)" --tab \
-- bash -i -c "source dp.sh; bit_measurement"
elif [ "$value" == "d" ]; then
gnome-terminal --title="RTT [ms]" --tab -- bash -i \
-c "source dp.sh; ping_measurement"
else menu fi
else
i=$(($i-1)) ...
```

7.2.1 Zobrazení síťových adres

Před spuštěním měření nebo generováním přenosu je vhodné zjistit základní síťové parametry uživatelské stanice. K tomuto účelu byla vytvořena funkce *net_param()*. V těle této funkce jsou použity příkazy *ip addr show*, *ip route* a *ip link*. Prvně uvedený příkaz je zřetězen s příkazem *awk /inet/*, v němž *inet* značí, že se jedná o protokol síťové vrstvy, a tak jsou vypsány všechny IP adresy přidělené dané stanici. Pomocí příkazu *ip route* jsou do konzole zaznamenány směrovací údaje zahrnující název výchozí brány. Poslední příkaz (*ip link*) je opět spojen pomocí tzv. pipe s příkazem *awk* a *xargs*. *Xargs* nahrazuje mezery pouze jednou mezerou. Parametr tohoto příkazu *-n 2* zajistí, že po dvou slovech dochází ke skoku na nový řádek, na němž se již budou nacházet údaje o dalším z rozhraní. Výstupem těchto příkazů jsou tedy údaje o linkové MAC adrese pro všechna rozhraní.

Výpis 7.3: Zobrazení síťových parametrů

```
net_param(){
echo "IP addresses:"
ip -o addr show | awk '/inet/ {print $2":",$4","," \
broadcast: "$6}'
ip route
echo "MAC addresses:"
ip link | awk '{print $2}' | xargs -n 2
sleep 120 }
```

Obr. 7.3: Ukázka volby 0 - zobrazení síťových adres

7.3 Multimediální přenosy

Tato podkapitola pojednává o přenosu a úpravě multimediálních dat. Pro úpravu multimediálních souborů a datových proudů jsou implementovány funkce, jež využívají vlastností programu FFmpeg. Pro multimediální přenos byly vytvořeny celkem tři funkce. Dvě z nich slouží k vysílání multimediálního obsahu pro různé testovací scénáře. Aby bylo možné na straně přijímače zobrazit přenášená multimediální data a následně je i zanalyzovat, byl vytvořen přijímač, který naslouchá na dané IP adrese a portu. Přijímací funkce byla vytvořena pouze jedna, a to pro oba druhy přenosu. Multimediální proudy jsou generovány a přijímány příkazem *vlc*. Tento příkaz je využitelný jak v grafickém, tak i v konzolovém prostředí. V následujícím textu jsou uvedeny stručné ukázky tykající se multimediálního přenosu. V diplomovém skriptu *dp.sh* jsou uvedeny v kompletní podobě.

7.3.1 Úprava multimediálních dat

Ještě než uživatel zahájí videopřenos, může multimediální obsah upravit podle vlastního uvážení. K úpravě multimediálních dat byl vybrán nástroj FFmpeg. Tento nástroj nabízí širokou škálu možností a je součástí funkce nazvané *ffsub_menu()*. Po zavolání této funkce je uživateli zobrazena nabídka pro práci s videoobsahem. Úvodní podmenu je implementováno v rámci *for (;;)* cyklu. Uživatel zadá příslušnou volbu, která je předána až do podmínky *case*, při níž dochází k porovnání a k provedení operace nadefinované pro vybranou možnost. Po spuštění daných voleb je uživatel vybídnut k zadání povinných parametrů. Přehledné znázornění uživatelského podmenu:

- H.265 kódování,
- H.264 kódování,
- VP9 kódování,
- AV1 kódování,
- změna rozlišení,
- změna snímkové frekvence,
- zobrazení informací.

Před tím než proběhne jakákoliv změna klíčových vlastností videa, je vhodné pomocí volby h zobrazit aktuální vlastnosti a z nich dále vycházet. Tato volba zde není uvedena a je možné ji najít ve skriptu *dp.sh* ve funkci *ffsub_menu()*. Zobrazení základních vlastností videa je zajištěno příkazem **ffprobe** s definovanými parametry. Pro konverzi videoobsahu byly implementovány čtyři kompresní formáty: **HEVC**, **AVC**, **VP9** a **AV1**. Podoba konverzního příkazu pro HEVC formát je uvedena ve výpisu 7.4. Přepínač *-i* obsahuje název vstupního videosouboru. Následuje pře-

pínač *-vcodec* značící, že bude provedena konverze do jiného formátu. Pro převod do HEVC formátu byla použita knihovna *libx265*. Součástí tohoto příkazu je *aac* audio kodek s definovanou rychlostí 128 kbit/s. Konverze do AVC formátu je uvedena ve výpisu 7.5. Tato metoda využívá stejné přepínače, avšak je zde použita knihovna *libx264*, která zajistí převod do AVC formátu. Metoda VP9 (výpis 7.6) využívá přepínače *-c:v*, přičemž písmeno c znamená codec a písmeno v znamená video. Knihovna *libpx-vp9* určí kompresní formát VP9. Přepínač *-b:v* specifikuje průměrnou bitovou rychlost na 2 Mbit/s. K poslednímu převodu byl vybrán formát AV1 (výpis 7.7), jehož příkaz s nastaveným parametrem *-c:v libaom-av1* značí využití referenčního *libaom* kodéru. Přepínač *-crf* může nabývat hodnot od nuly až po šedesát tři, nižší hodnota znamená lepší kvalitu a větší velikost souboru. Přepínač *-b:v 0* nastaví datový tok na nulu. Pokud by tak nebylo učiněno, *crf* by spustil režim omezené kvality s výchozí bitovou rychlostí 256 kbit/s. Poslední parametr *-strict experimental* je přidán z důvodu, že je použita starší verze *libaom* kodéru. Na konci každé konverze jsou do terminálového okna vypsány tyto údaje: obrazová bitová rychlost původního a nově vzniklého videa a také celková velikost původního a nového videosouboru. Ve finále jsou ještě vypočítány rozdíly mezi těmito získanými parametry.

Výpis 7.4: FFmpeg příkaz pro konverzi do HEVC formátu

```
ffmpeg -i videos/$vname -vcodec libx265 -acodec aac \
videos/$nwname
```

Výpis 7.5: FFmpeg příkaz pro konverzi do AVC formátu

```
fmpeg -i videos/$vname -vcodec libx264 -acodec aac \
videos/$nwname
```

Výpis 7.6: FFmpeg příkaz pro konverzi do VP9 formátu

```
ffmpeg -i videos/$vname -c:v libvpx-vp9 -b:v 2M \
videos/$nwname
```

Výpis 7.7: FFmpeg příkaz pro konverzi do AV1 formátu

```
ffmpeg -i videos/$vname -c:v libaom-av1 -crf 30 -b:v 0 \
-strict experimental videos/$nwname
```

Součástí úpravy dat jsou také příkazy pro **změnu frekvence**, **rozlišení** a **změnu doby trvání** videa. Na začátku každého příkazu musí být definovány povinné údaje (název vstupního a výstupního videa atd.). Hodnota nové snímkové frekvence fps je nastavena stejnojmenným přepínačem. Změna rozlišení je zajištěna filtrem *scale*, kde dochází ke specifikování šířky a výšky videa. Tyto parametry jsou nastaveny uži-

vatelem, který stanoví finální rozlišení nově vytvořeného videa. Změna délky videa je uskutečněna definovaným přepínačem *-t*, jenž stanoví novou délku trvání videa.

Výpis 7.8: Změna snímkové frekvence

```
ffmpeg -i videos/$vname -filter:v fps=$fps videos/$nwname
```

Výpis 7.9: Změna rozlišení

```
ffmpeg -i videos/$vname -vf scale=$wi:$he videos/$nwname
```

Výpis 7.10: Změna délky videa

```
ffmpeg -i videos/$vname -t $newdur videos/$nwname
```

7.3.2 Unicastový přenos dat

Tato kapitola pojednává o unicastovém multimediálním generátoru. Unicastový generátor je spuštěn zavoláním funkce *unicast_sender*. Tento generátor nabízí velký počet možností. Ihned po spuštění jsou uživateli nabídnuty dvě úvodní:

- volba 1 - streamování všech videí ze složky *videos*,
- volba 2 - streamování konkrétního videa.

Po dokončení tohoto výběru následuje dotazování ohledně počtu unicastových koncových stanic, na než chce obsah přenášet. Nabídka možností vypadá následovně:

- volba a - jeden unicastový příjemce,
- volba b - dva unicastový příjemci,
- volba c - tři unicastový příjemci,
- volba d - ukončení streamování.

Po zadání této volby je tázán na výběr typu uživatelského rozhraní programu *vlc*. Dotazování je realizováno pomocí nekonečného cyklu *while* uvedeného ve výpisu 7.11. Tento výběr bude záviset především na tom, zda uživatelský počítač podporuje grafický, nebo jen textový režim. Nabídka má následující podobu:

- volba a - GUI verze (příkaz vlc),
- volba b - CLI verze (příkaz cvlc).

Po výběru uživatelského rozhraní je téměř vše hotovo a přistupuje se k samotnému přenosu. Uživatel má poslední možnosti ovlivnit parametry přenosu. Může si vybrat mezi:

- volbou a - streamování celého videa,
- volbou b - streamování určitého segmentu videa.

Výpis 7.11: Unicastový generátor - volba typu rozhraní

```
unicast_sender(){
...
while true do
echo "Select mode: "
echo "Option a - GUI version (vlc)"
echo "Option b - CLI version (cvlc)"
read -p "Set a streaming version: " option
case $option in
a)
version_vlc="vlc"
break ;;
b)
version_vlc="cvlc"
break ;;
*)
echo "Repeat the option.." ;;
esac
done
...
```

V rámci této funkce (viz výpis 7.12) dochází k větvení podmínek, do nichž jsou dále vnořeny další podmínky. První podmínka *if elif else* porovnává, zda byla zvolena volba jedna (streamování všech videí), nebo dva (jednoho videa). Je porovnávána proměnná *$allspec*, jejíž hodnotu zadal uživatel přes příkaz *read -p*. Pokud uživatel zadal hodnotu jedna, provede se blok *if [$allspec == 1]*. Pokud zadal hodnotu dva je vykonán blok *elif [$allspec == 2]*, jinak se provede blok *else*. V případě *else* je znovu zavolána funkce *unicast_sender* a uživatel musí parametry zadat od znovu.

V tělu bloku *if [$allspec == 1]* jsou implementovány podmínky pro výběr počtu koncových stanic. Porovnávaná proměnná *$val* je opět předána přes příkaz *read -p*. Pomocí podmínky *if elif elif elif else* dochází k vnoření do první podmínky a také ke zjišťování, zda byla zadána volba a, b, c nebo d. V případě zadání písmena mimo rozsah, dochází opět k opětovnému zavolání unicastové funkce a uživatel musí celý proces opakovat. Pokud však uživatel zvolí jednu ze správných možností je vyzván k doplnění dalších údajů (IP adresa a port). V závislosti na výběru volby jsou definovány konkrétní *vlc* příkazy. Tyto příkazy se liší podle nastavených parametrů (jsou znovu zadávány uživatelem), které odpovídají počtu stanic, na něž má probíhat přenos videoobsahu. Na dalším řádku se nachází *for* cyklus, který slouží k procházení složky *videos*, jež obsahuje všechna videa k přenosu. Jakmile tento cyklus načte první

video, je uživatel vyzván k určení času přenosu. Může si vybrat mezi streamováním celého videa, nebo může specifikovat konkrétní čas. Tato funkcionalita je opět zajištěna větvením podmínky *if*, která je vnořená do již vnořené podmínky (podmínka v podmínce). V případě výběru první možnosti, je pomocí *ffprobe* zjištěna doba trvání videa a tato hodnota je uložena do proměnné *dur*, jež je následně předána do příkazu *timeout*. Příkaz *timeout* je úzce spjat s příkazem *vlc*. Pokud si vybere volbu dvě a zadá přesnou hodnotu (v sekundách), dochází k předání zadané hodnoty opět do příkazu *timeout*. Pokud by uživatel zadal nesprávnou hodnotu časové volby, proběhne streamování celého videa (vykoná se blok *else*). Po konci každého přenosu je uživateli položena otázka, zda chce pokračovat, nebo zda chce ukončit proces streamování. Funkcionalita je založena na stejném principu jako v případě volby typu rozhraní. Ukončení streamování je zajištěno příkazem *exit*.

Tělo bloku *elif [$allspec == 2]* se provede, pokud se proměnná *$allspec* rovná dvěma. Programovací logika je stejná jako v prvním případě (pro volbu číslo jedna), avšak nachází se zde mírné odlišnosti vyplývající z principu. Zásadní odlišnost oproti předchozímu bloku spočívá v odstranění cyklu *for*, jelikož zde nedochází k procházení celé složky *videos*. Název přenášeného videa je dopředu specifikován. V této části již také není potřebný nekonečný tázací cyklus *while*, protože je zde vyžadován přenos pouze jednoho videa.

Výpis 7.12: Ukázka unicastového generátoru - volba 1 pro jednoho příjemce

```
unicast_sender(){
...
if [ $allspec == 1 ];
then
if [ "$val" == "a" ];
then
read -p "Destination IP address: " ip
read -p "port: " port
for str in "videos"/*;  do
echo "Option a - full video streaming "
echo "Option b - specific streaming time"
read -p  "Option for time is: " mode
read -p "Set a streaming time (in seconds): " spect
dur=$(ffprobe -i $str -show_entries format=duration \
-v quiet -of csv="p=0")
if [ "$mode" == "a" ]; then
t_param=$dur
elif [ "$mode" == "b" ]; then
t_param=$spect
else
t_param=$dur
fi
timeout $t_param $version_vlc -v file:///home/$dir \
/DP/$str :sout="#duplicate{dst=display, \
dst=rtp{dst=$ip,port=$port,mux=ts}}"
...}
```

7.3.3 Multicastový přenos dat

Multicastový přenos je spuštěn zavoláním funkce *multicast_sender*. Struktura kódu a celková implementace dílčích vlastností vychází z unicastového generátoru. Z uvedeného důvodu již struktura funkce nebude popisována. V rámci této funkce jsou definovány opět tři možnosti dle počtu koncových stanic, na něž bude obsah přenášen. Z podstaty věci vyplývá, že multicastový *vlc* příkaz bude mít jiné parametry než unicastový *vlc* příkaz. Tato část je tedy zaměřena na detailní rozbor parametrů uvedených ve *vlc* příkazu.

Multimediální streamování může současně probíhat až ke třem koncovým uživa-

telům. Nabídka těchto možností vypadá následovně:

- volba a - pouze multicastové streamování,
- volba b - jeden unicastový příjemce a jedna multicastová skupina,
- volba c - dva unicastoví příjemci a jedna multicastová skupina,
- volba d - ukončení režimu.

Po zadání volby *a* prostřednictvím příkazového řádku programu VLC je zajištěna funkce přenosu na jednu multicastovou adresu. Na začátku řádku je definován příkaz *timeout*, který zajišťuje spouštění příkazu pro předem stanovený časový limit. Hodnota času je určena uživatelem před tím v rámci nabídky voleb. Za touto částí se nachází již nachází streamovací příkaz *vlc* s parametrem *-v*. Tento příkaz je možné používat v grafické, či příkazové formě. Vybraný režim reprezentuje proměnná *$version_vlc*, ve které je uložen příkaz *clvc* (c značí textový režim) a *vlc* reprezentující grafickou verzi. Parametr *-v* značí míru výpisu informací o relaci a přenosu. Další možností je přidání parametru *-vv*, nebo například *-vvv* pro výpis většího počtu informací. Po tomto již následuje definice cesty k samotnému souboru, jež odpovídá adresářové struktuře vytvořené při instalaci. Dále jsou zadány: přenosová metoda RTP/MPEG TS, cílová adresa a port. IP adresa a port jsou definovány ihned po zavolání multicastové funkce. Parametr *dst=display* zajistí, že přenášený videoobsah je zobrazen i na streamovacím zařízení. Dále jsou definovány tyto parametry: povolení protokolu SAP, název skupiny, název streamu a metoda *mux=ts*. Metoda multiplexování ve formě transport streamu sloučí více proudů do jednoho výsledného. Po dokončení přenosu je definována pětisekundová pauza, po které dojde příkazem *clear* k vymazání *vlc* informací z terminálového okna. Volby *b* a *c* fungují na stejném principu, ale jsou zde přidány další parametry související s přenášením dat na další stanice. Ukázka multicastového generování pomocí programu VLC je uvedena ve výpisu 7.13.

Výpis 7.13: Ukázka multicastového generátoru - režimy generování

```
multicast_sender(){
...
if [ $allspec == 1 ]; then
if [ "$val" == "a" ]; then
timeout $t_param $version_vlc -v
file:///home/$dir/DP/$str
:sout="#duplicate{dst=display,dst=rtp{dst=$ip,port= \
$port,mux=ts,sap,group="$ngroup",name=$nstream}}"
clear
...
elif [ "$val" == "b" ]; then
timeout $t_param $version_vlc -v \
file:///home/$dir/DP/$str \
:sout="#duplicate{dst=display,dst=rtp{dst=$ip1, \
port=$port1,mux=ts},dst=rtp{dst=$ip2,port= \
$port2,mux=ts,sap,group="$ngroup",name=$nstream}}"
...
elif [ "$val" == "c" ]; then
timeout $t_param $version_vlc -v \
file:///home/$dir/DP/$str \
:sout="#duplicate{dst=display,dst=rtp{dst=$ip1, \
port=$port1, mux=ts},dst=rtp{dst=$ip2,port= \
$port2,mux=ts,sap,group="$ngroup",name=$nstream}, \
dst=rtp{dst=$ip3,port=$port3,mux=ts}}"
...
elif [ "$val" == "d" ]; then
exit
...
else
multicast_sender
fi ...}
```

7.3.4 Multimediální přijímač

Pro zobrazení videa u koncového účastníka byla vytvořena přijímací funkce. Tuto funkci lze použít pro oba typy vysílání. Tato část je spuštěna zavoláním funkce *multimedia_receiver()* (výpis 7.14). Uživatel je vyzván k zadání následujících parametrů:

- IP adresa,
- port.

Po zadaní parametrů jsou nabídnuty dvě následující možnosti: přidaní firewallového pravidla pro zvolený port a volba typu rozhraní. Následuje samotný příkaz *vlc*, jež umožňuje zobrazení obsahu přenášeného multimédia. Přijímač naslouchá na výše zadané IP adrese a portu.

Výpis 7.14: Úryvek funkce pro přijímač multimediálního provozu

```
multimedia_receiver(){
...
while true do
echo "Select mode: "
echo "Option a - GUI (vlc)"
echo "Option b - CLI (cvlc)"
read -p "Set a streaming version: " option
case $option in
a)
version_vlc="vlc"
break
;;
...
sleep 3
$version_vlc rtp://$ip:$port }
```

7.4 Datové přenosy

Pro datový přenos byly naprogramovány funkce, které využívají vlastností protokolů TCP a UDP. Implementace datového přenosu byla provedena pomocí příkazu *nc*. Součástí těchto funkcí je také unixový nástroj nazývající se *jot*. Tento nástroj umožňuje generování náhodných dat s možností následného uložení vytvořených dat do textového souboru. Tato část skýtá celkem čtyři funkce (2 pro vysílání a 2 pro příjem dat) uvedené ve finálním skriptu. Vzhledem k odlišnému principu přenosu dat byly vytvořeny testovací funkce pro TCP a UDP protokol. V operačním systému Ubuntu je nutné nastavit příslušná firewallová pravidla. Příkaz *sudo ufw allow <číslo portu>/<název protokolu>* povolí konkrétní komunikaci na zadaném portu.

7.4.1 Vysílač pro TCP a UDP provoz

V rámci vytvořené topologie je možné odesílat datový provoz nejen ze serveru, ale také směrem od klientů. Přenosy v opačných směrech se liší pouze zadanými IP adresami a porty, k nimž je obsah směrován.

Jako první je popisován **TCP generátor provozu**. Tento generátor je spuštěn zavoláním funkce *tcp_sender()*. Po spuštění je nutné zadat tyto parametry:

- cílovou cestu k předloze pro generování,
- číslo sloupce, z něhož má být generováno,
- oddělovač dat,
- cílovou IP adresu,
- port,
- délku trvání pauzy mezi přenosy.

Proměnná *input*, jenž byla zadána uživatelem, definuje cestu k souboru s přenosovými parametry (viz výpis 7.15). Dále je inicializována proměnná *i*, která určuje číslo vygenerovaného souboru. Poté následuje deklarace cyklu *while*. Tento cyklus čte řádky ze souboru obsahující náměr z reálné sítě. Tento náměr definuje velikost výsledného souboru, a tedy také bitovou rychlost přenosu. V těle cyklu *while* se nachází proměnná *fileSize*. V této proměnné je využita tzv. *pipe* ("|"). *Pipe* je forma přesměrování výstupu jednoho příkazu na vstup druhého. S následnou kombinací příkazu *awk* s nastaveným parametrem *-F* (pro oddělovač) je hodnota zadaného sloupce každého řádku vytisknuta do proměnné *fileSize*. Každým průchodem se také zvyšuje hodnota *i*. Příkazem *jot* je vygenerována požadovaná velikost souboru. Tento příkaz má následující parametry: *-r* značí, že budou generována náhodná data a, *-c* definuje velikost přenášených dat. Nově vzniklý soubor je uložen do příslušné složky ve formě textového souboru.

V další části je definován *for* cyklus, který slouží k načítání uložených souborů. Případně je možné odkomentovat a implementovat druhou smyčku *for*, jež značí nadefinovaná čísla pro IP adresu a port. Tato smyčka je využita pouze v případě, že se v síti nachází více příjemců čekajících na data. Cykly jsou vnořeny do sebe. To má za následek, že se jeden typ vytvořeného souboru odešle na všechny IP adresy definované právě druhým cyklem. Příkaz *nc* umožňuje otevřít TCP či UDP spojení. Podoba příkazu má například následující strukturu: *nc -vnn $ipadd $port -w1 <$fileName*. Nejdříve jsou tedy určeny sockety (IP adresa + port) koncových zařízení zadané ihned v úvodu, poté je nastaven parametr *w1*. Tento parametr určuje *timeout* příkazu. Jako poslední je definován název souboru, který bude přes dané spojení odeslán. V závěru skriptu je napsán příkaz *sleep* sloužící k uspání procesu na definovanou dobu. Příkazem *rm* po každém průchodu *while* cyklu je soubor proměnné velikosti ihned odstraněn. Tímto způsobem je zamezeno zbytečnému snižování ka-

pacity úložného média jednodeskového počítače.

Výpis 7.15: Generátor TCP provozu

```
tcp_sender(){
...
i=0
while read -r line; do
fileSize=$(echo $line | awk -F "$separator" \
'{print $'"$column"'}')
let value=$fileSize*2
echo "File size:" $value "B"
i=$(($i+1))
echo "File number:" $i
jot -r -c $fileSize >/home/$dir/DP/
files/data$i.txt
echo "Creation new file data"$i".txt"
for fileName in "files"/*.txt; do
#for j in {2..3} do
echo "Transmission:" $fileName
nc -vnn $ipadd $port -w1 < $fileName
#done
echo "$ptime second pause"
echo "------------------------------"
sleep $ptime
done
rm /home/$dir/DP/files/*.txt
done < "$input" ...}
```

UDP generátor provozu se liší od TCP funkce v zadaném parametru příkazu *nc*, v čísle portu a samotným zavoláním funkce (*udp_receiver()*). Jedná se o parametr *-u*, který upozorňuje, že pro přenos bude použit protokol UDP.

Výpis 7.16: Příkaz Netcat pro generátor UDP provozu

```
nc -u -vnn $ipadd $port -w1 < $fileName
```

7.4.2 Přijímač pro TCP a UDP provoz

Jako první je popsána funkce nazvaná *tcp_receiver()*, která slouží jako **přijímač TCP provozu**. Po zavolání této funkce je uživatel vyzván k zadání portu, na němž bude naslouchat. Dále musí stanovit počet průchodů cyklu a dobu trvání pauzy. Tyto údaje jsou předány do proměnných *port*, *end* a *ptime*. Následuje dotaz, zda

chce povolit příchozí komunikaci na zvoleném naslouchacím portu. Pokud již toto pravidlo bylo nastaveno v minulosti, stačí vložit písmeno *n* jako Ne. Tázací logika je naprogramována jednoduchým nekonečným cyklem *while* v kombinaci s podmínkou *case*, proto se nemůže stát, že uživatel zadá volbu mimo rozsah. Opět je deklarována proměnná *i*, která je opět určena k číslování výsledného souboru. Po dokončení podmínky následuje *for* cyklus, který bude proveden od jedničky až po číslo rovnající se hodnotě *$end*, jíž sám uživatel zadal. V těle cyklu dochází k inkrementaci proměnné *i*. Stejně jako na vysílací straně je použit příkaz *nc*. Tento příkaz slouží k naslouchání na daném portu. Po úspěšném přijetí je soubor na krátkou chvíli uložen a poté smazán příkazem *rm*.

Výpis 7.17: Přijímací skript TCP provozu

```
tcp_receiver(){
...
while true do
read -p "Allow port in ufw? (y/n):  " ex
case $ex in
y)
sudo ufw allow $port/tcp
break
;;
n)
break
;;
*)
echo "Repeat the option.."
;;
esac
done
echo "I'm listening on tcp port $port"
i=0
for (( i=1; i<=$end; i++ )) do
i=$(($i+1))
nc -vvlnp $port >/home/$dir/DP/files/data$i.txt
rm /home/$dir/DP/files/*.txt
... }
```

Jako druhý přijímač byl vytvořen **přijímací skript pro UDP provoz**. Tento skript je téměř totožný jako předchozí varianta. Příkaz *nc* obsahuje parametr *-u*, který určuje typ přijímaného proudu. Říká nám, že přenos bude uskutečněn přes

UDP protokol. Dále je také nastaven parametr pro timeout *-w1*. Parametr timeout určuje počet sekund stanovujících výchozí čas, po němž dojde k ukončení příkazu *nc*.

Výpis 7.18: Příkaz Netcat u přijímacího UDP skriptu

```
nc -u -vvlp $port -w1
```

7.5 Měřicí nástroje

Měření může probíhat v režimu záznamu v podobě ukládání do souboru s příponou csv nebo v reálném čase, v němž se uživateli zobrazuje jeho průběh přímo v terminálovém okně. V rámci měření byly využity tyto nástroje: *ifconfig*, *tshark*, *smag* a *youplot*. Nástroj *smag* je spuštěn v případě real-time vykreslování, aby uživatel mohl ihned sledovat průběh přenosu. Uložení změřených dat poskytují nástroje *tshark* a kombinace linuxových příkazů, jejichž základ tvoří příkaz *ifconfig*. Pokud uživatel již disponuje souborem s daty, je možné ho graficky znázornit příkazem *youplot*.

7.5.1 Tshark

Tato funkcionalita je rozdělena celkem do pěti funkcí uvedených ve výpisu 7.19. Zavoláním první funkce (*tshark_realtime()*) je spuštěno ukládání dat. Nejprve se však musí určit rozhraní, na kterém bude měřeno, a také název finálního pcap souboru. Parametr *-i* určuje název rozhraní, na němž probíhá zachytávání, parametr *-w* určuje, kam souboru bude uložen. Pro uložení je nutné vytvořit složku s administrátorskými právy. Druhá funkce, která se nazývá *tshark_conversion()*, zajišťuje konverzi z typu pcap do csv. V této funkci je zajištěna specifikace oddělovače dat pro nově vzniklý soubor. Pro detailnější analýzu lze využít různých přepínačů a linuxových příkazů. K tomuto účelu byla vytvořena funkce *tshark_byt()* sloužící k zobrazení bajtové rychlosti. Dále byla vytvořena čtvrtá funkce *tshark_rtp()*. Po zadaní povinného údaje (názvu souboru) je provedeno čtení zadaného pcap souboru. Jsou zobrazeny statistiky RTP streamu (jitteru, zpoždění a ztrátovosti paketů). Uživateli jsou zobrazena maxima či minima těchto parametrů. Poslední funkce (*tshark_bit()*) se týká převodu bajtové rychlosti na bitovou. Jelikož data budou později vykreslena, je nutné je převést do vhodného formátu. Zde se nachází hojně používaný příkaz *grep*, který slouží k vyhledávání řetězce znaků. Parametr *-P* tohoto příkazu umožňuje párovat regulární výrazy. *d+* testuje shodu na jedné nebo více číslicích, *s+<>s+* určuje jednu nebo více mezer následovaných znakem '<>'. V zobrazeném měření není uvedena mezera mezi jednotlivými položkami, ale data odděluje znak "|". Dále dochází k porovnání řetězců. Příkaz *awk -F* vytiskne sloupec s bajtovou

rychlostí. Jelikož je požadována bitová rychlost, tak tato data jsou vynásobena číslem osm. Následně jsou data uložena do csv souboru. Oddělovač tohoto souboru je čárka.

Výpis 7.19: Analýza dat pomocí programu Tshark

```
tshark_realtime(){
read -p "Insert name of capturing port: " port
read -p "Insert name of pcap file: " pcap
sudo tshark -i $port -w /home/capture/$pcap.pcap }
tshark_conversion(){
read -p "Insert name of pcap file: " pcap
read -p "Insert name of csv file  " nfile
sudo tshark -r /home/capture/$pcap.pcap -E separator=, \
>> /home/$dir/DP/measurement/$nfile.csv }
tshark_byt(){
read -p "Insert name of pcap file: " pcap
read -p "Insert source IP address: " ip
read -p "Insert name of csv file  " nfile
sudo tshark -r /home/capture/$pcap.pcap -q -z io,stat, \
1,"BYTES()ip.src == $ip" >> \
/home/$dir/DP/measurement/$nfile.csv }
tshark_rtp(){
read -p "Insert name of pcap file: " pcap
sudo tshark -r /home/capture/$pcap.pcap -q -z rtp,streams}
tshark_bit(){
read -p "Insert name of pcap file: " pcap
read -p "Insert name of csv file  " nfile
sudo tshark -nr /home/capture/$pcap.pcap -q -z io,stat, \
1,BYTES | grep -P "\d+\s+<>\s+\d+\s*\|\s+\d+" |
awk -F '[ |]+' '{print $2","($5*8)}' >> \
/home/$dir/DP/measurement/$nfile.csv }
```

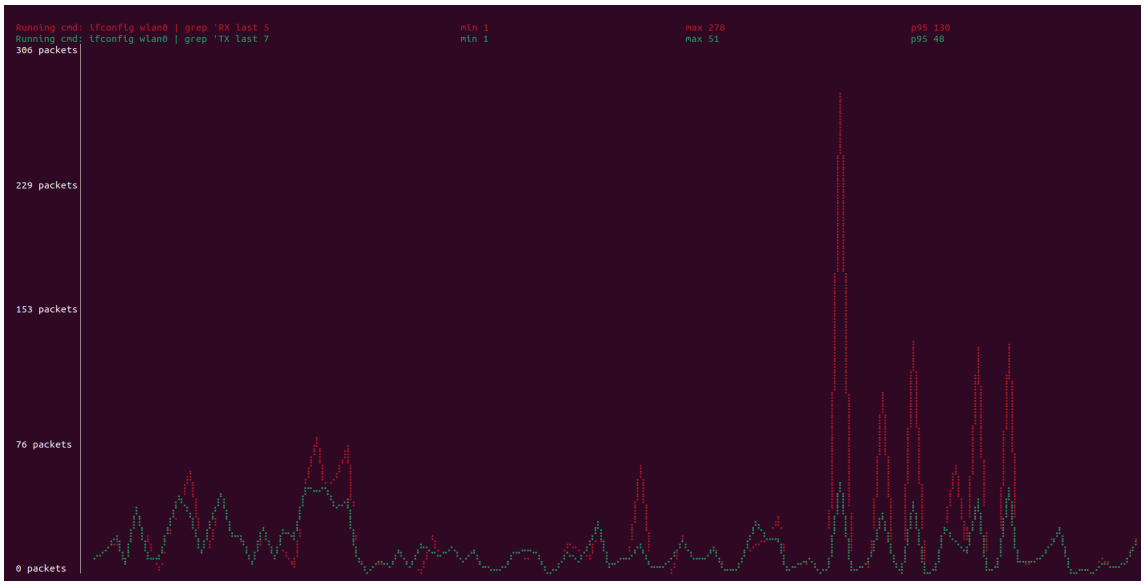
7.5.2 Nástroj smag

Pomocí příkazu ***smag*** lze zobrazit průběh komunikace v reálném čase. Tento příkaz nabízí grafický výstup při zřetězení různých typů příkazů, jejichž výstup udává nějaké číslo, které je následně převedeno do grafické podoby. Vykresleny jsou následující průběhy: přenosy paketů, bajtů, bitů a také je znázorněno měření uskutečněné příkazem *ping*. Tento příkaz odesílá ICMP zprávy a následně měří RTT. Základ funkcí uvedených ve výpisu 7.20 je tvořen tímto nástrojem. Než je spuštěno měření,

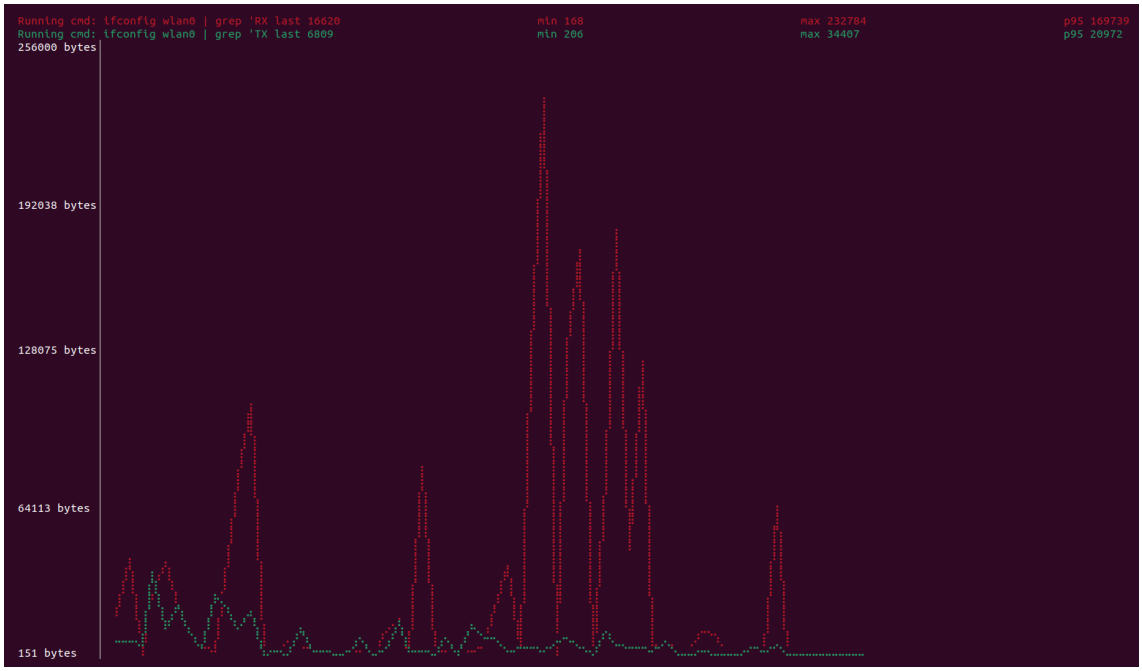
je uživatel povinen zadat název rozhraní, na kterém mají být data zachytávána. Samotný příkaz *smag* se spouští napsáním *./smag*. Parametr *-n* udává interval pro vykreslování dat, parametr *-y* nastaví název jednotky osy y. Parametr *-d* udává, že je požadováno vykreslení rozdílu mezi dvěma příslušnými výstupy (uvedené příkazy). Příkaz *ifconfig* slouží k zobrazení parametrů pro síťová rozhraní. Dochází zde k zřetězení příkazů. Příkaz *grep* vybere položku *RX packets/Tx packets*. Příkazem *awk* je vytisknuta číselná hodnota (počet paketů, počet bajtů a počet bitů) v daném sloupci. Dostupnost druhé strany je otestována příkazem ping. Jsou zde definovány různé přepínače. Přepínač *-c* udává, že chceme odeslat jeden ICMP paket, přepínač *-s* udává velikost v bajtech. Následuje příkaz *grep*, který vybere položku *time*, a příkazem *awk* je vypsána hodnota sedmého sloupce (obousměrné zpoždění v ms). Příkaz *sed* odstraní pět znaků tak, aby na výstupu zůstalo pouze číslo.

Výpis 7.20: Vykreslení přenášených dat v reálném čase

```
bit_measurement(){
read -p "Insert name of capturing port: " port
./smag -n 1.0 -y bits -d "ifconfig $port | \
grep 'RX packets' | awk '{print \$5*8}'" "ifconfig \
$port | grep 'TX packets' | awk '{print \$5*8}'" }
byt_measurement(){
read -p "Insert name of capturing port: " port
./smag -n 1.0 -y bytes -d "ifconfig $port | \
grep 'RX packets' |awk '{print \$5}'" "ifconfig \
$port | grep 'TX packets' | awk '{print \$5}'" }
packets_measurement(){
read -p "Insert name of capturing port: " port
./smag -n 0.5 -y packets -d "ifconfig $port | \
grep 'RX packets' | awk '{print \$3}'" "ifconfig \
$port | grep 'TX packets' | awk '{print \$3}'" }
ping_measurement(){
read -p "Ping to xxx.xxx.xxx.xxx: " ipaddr
./smag -y ms -n 1.0 "ping -c 1 -s 100 -R $ipaddr | \
grep 'time=' | awk '{print \$7}' | sed -r 's/.{5}//'" }
```

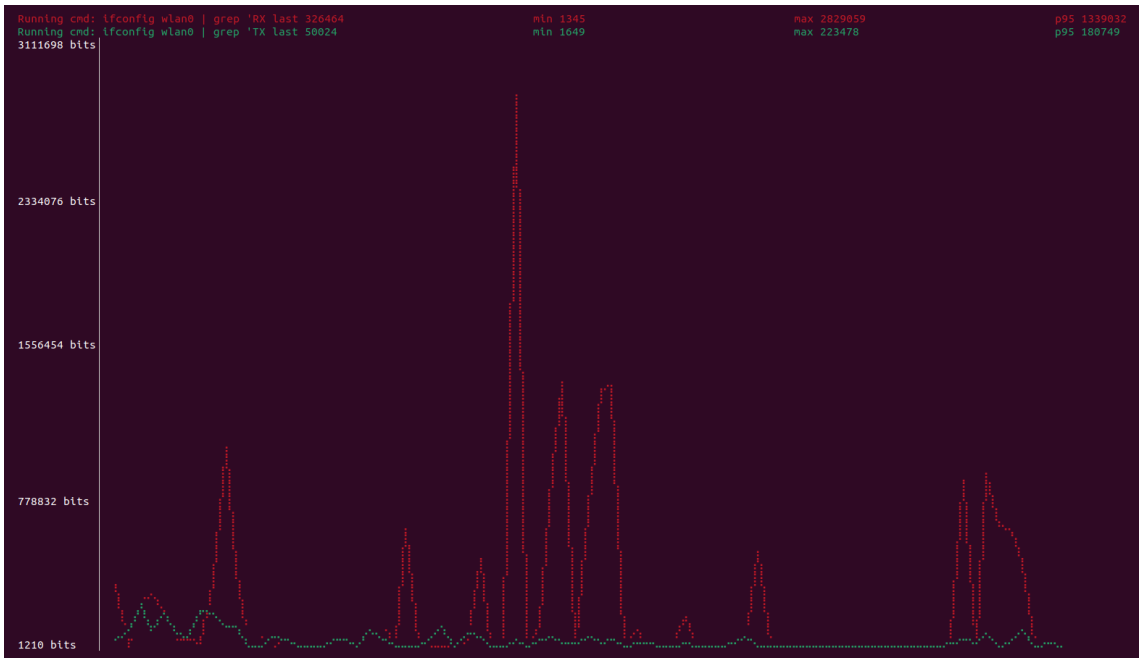


Obr. 7.4: Přenos paketů v čase

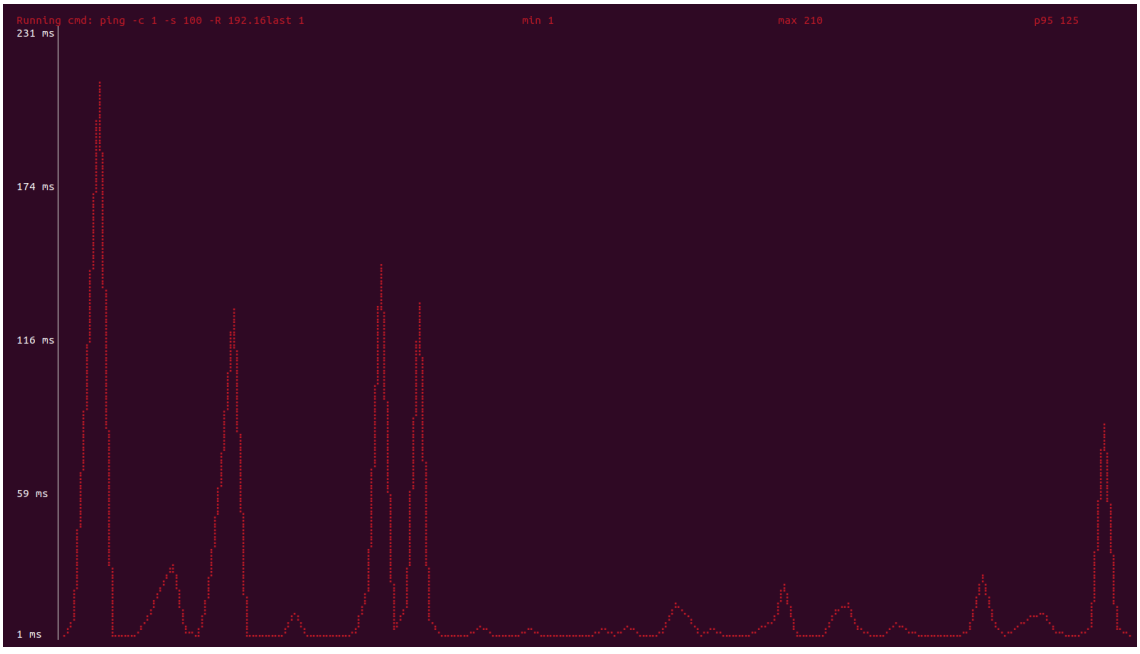


Obr. 7.5: Bajtová rychlost v čase

Na obrázcích 7.4, 7.5, 7.6 a 7.7 jsou zobrazeny ukázky grafických výstupů pro jednotlivé přenosové parametry.



Obr. 7.6: Bitová rychlost v čase



Obr. 7.7: Měření obousměrného zpoždění

7.5.3 Linuxový nástroj ifconfig

Linuxový nástroj *ifconfig* slouží k zobrazení parametrů síťových rozhraní. Vhodnou kombinací s dalšími nástroji lze dosáhnout požadovaného výstupu. Tyto vlast-

nosti byly uplatněny při programování funkcí, které umožňují měření a ukládání dat. Příkazy pro zobrazení základní přenosových parametrů měly stejnou podobu jako v případě real-time měření nástrojem *smag*, uvedeném v podkapitole výše. Celkem byly vytvořeny 4 funkce, jejichž základní ukázka je uvedena ve výpisech níže. Kompletní podoba funkcí je uvedena v diplomovém skriptu *dp.sh*. Funkce se nazývají *bit_measurement_csv*, *byt_measurement_csv*, *packets_measurement_csv* a *ping_measurement_csv*.

Zavoláním funkce *bit_measurement_csv* (výpis 7.23) je uživatel požádán o definování těchto údajů:

- rozhraní, na němž má být měřeno,
- název souboru k uložení dat,
- kolik hodnot vzorků má být zaznamenáno,
- počet sekund pro pauzu mezi záznamy.

Po vyplnění údajů je uživatel znovu dotazován, zda chce restartovat počítadlo bitové rychlosti. Pokud zvolí odpověď *y* (yes), tak se doposud zaznamenané statistiky vynulují. Hodnoty přijatých/odeslaných bitů jsou počítány ihned po spuštění operačního systému stanice. Pokud by nebyla implementována tato možnost, mohlo by docházet ke zkreslení měřených dat. Vynulování počítadla probíhá následujícím způsobem. Proměnná *eth* obsahuje příkaz *ethtool*, který v kombinaci s nástrojem *grep* zajistí vypsání názvu ovladače pro dané síťové rozhraní. Tento název (*$eth*) je poté předán do příkazu *modprobe* s parametrem *-r*, jenž značí reset neboli odebrání modulu z jádra operačního systému. Tento modul je následně ihned znovu zaveden do jádra operačního systému příkazem *modprobe $eth*, ale tentokrát bez jakéhokoliv parametru. Odebráním modulu z jádra (byť na malý moment) dochází k vynulování dosud zaznamenaných hodnot. Následuje cyklus *for* probíhající od hodnoty jedna až po hodnotu *$end*, jejíž hodnota byla zadána uživatelem. S průchodem cyklu je inkrementována hodnota času, která je ukládána do prvního sloupce csv souboru. Tělo cyklu obsahuje kombinaci příkazů pro zobrazení přijatých a odeslaných bitů, jež jsou ukládány do druhého a třetího sloupce. Jakmile měření doběhne, je spočítána hodnota bitové chybovosti, která je definována jako podíl chybně přijatých bitů ku celkovému počtu přijatých bitů.

Funkce *byt_measurement_csv* a *packets_measurement_csv* se liší od předchozí funkce pouze příkazy (*ifconfig*) pro zobrazení přenosových parametrů. Poslední funkce (*ping_measurement_csv*) se týká měření a ukládání hodnot obousměrného zpoždění. Před měřením jsou zadány parametry: název csv souboru, počet požadovaných záznamů, pauza mezi vzorky a IP adresu, na níž má být příkaz *ping* proveden. Následuje naprosto stejně definovaný cyklus jako v předchozích třech případech. Příkaz *ping* má definované tyto parametry: *-c* udává, že je požadováno odeslání jednoho ICMP paketu, přepínač *-s* udává velikost v bajtech (B). Dále jsou připojeny pří-

kazy: *grep* (vybere položku *time*) a *awk* (výpis hodnoty sedmého sloupce) a *sed -r 's/.5//'* (odstraní pět znaků, zůstane pouze číslo). V závěru funkce je naprogramováno počítání maximální, minimální a průměrné hodnoty obousměrného zpoždění (v ms).

Výpis 7.21: Funkce pro záznam měření obousměrného zpoždění

```
ping_measurement_csv(){
...
for (( i=1; i<=$end; i++ )) do
printf '"%s"\n' "$i" >> date4.csv
ping -c 1 -s 100 -R $ipaddr | grep 'time=' | \
awk '{print $7}' | sed -r 's/.{5}//' >> ping.csv
...
echo "Calculate max, min and avg values"
awk -F "," 'NR==2{max=$2} $2 > max {max = $2} END \
{print "Max:",max,"ms"}' measurement/$pcname.csv
awk -F "," 'NR == 1 || $2 < min {line = $0; min = $2} \
END {print "Min:",min,"ms"}' measurement/$pcname.csv
awk -F "," '{total += $2; count++} END \
{print "Avg:",total/count,"ms"}' measurement/$pcname.csv}
```

Výpis 7.22: Ukázka funkcí pro měření počtů bajtů a paketů

```
byt_measurement_csv(){
...
for (( i=1; i<=$end; i++ )) do
printf '"%s"\n' "$i" >> date2.csv
ifconfig $port | grep 'RX packets' |awk '{print $5}' \
>> rxp2.csv
ifconfig $port | grep 'TX packets' |awk '{print $5}' \
>> txp2.csv  ...}
packets_measurement_csv(){
...
for (( i=1; i<=$end; i++ )) do
printf '"%s"\n' "$i" >> date3.csv
ifconfig $port | grep 'RX packets' |awk '{print $3}' \
>> rxp3.csv
ifconfig $port | grep 'TX packets' |awk '{print $3}' \
>> txp3.csv ...}
```

Výpis 7.23: Ukázka funkce pro ukládání bitové rychlosti v čase

```
bit_measurement_csv(){
...
while true do
read -p "Reset ifconfig counter? (y/n):" ex
case $ex in
y)
eth=$(ethtool -i $port | grep driver | awk '{print $2}')
sudo modprobe -r $eth
sudo modprobe $eth
sleep 5
break ;;
n)
break ;;
*)
echo "Repeat the option.." ;;
esac
done
for (( i=1; i<=$end; i++ )) do
printf '"%s"\n' "$i"  >> date1.csv
ifconfig $port | grep 'RX packets' | awk '{print $5*8}' \
>> rxp1.csv
ifconfig $port | grep 'TX packets' | awk '{print $5*8}' \
>> txp1.csv
sleep $pause
done
echo "t[s]","RxBits","TxBits" > measurement/$cname.csv
paste -d "," date1.csv rxp1.csv txp1.csv >> \
measurement/$cname.csv
rm date1.csv; rm rxp1.csv; rm txp1.csv
bits_in_error=$(ifconfig $port | grep 'RX errors' | \
awk '{print $5*8}')
total_bits_received=$(ifconfig $port | grep 'RX \
packets' | awk '{print $5*8}')
ber=$(($bits_in_error/$total_bits_received))
berpr=$(($ber*100))
echo "BER = bits_in_error / total_bits_received = \
"$ber" = "$berpr "%"
echo "Completed!" }
```

7.5.4 Nástroj youplot

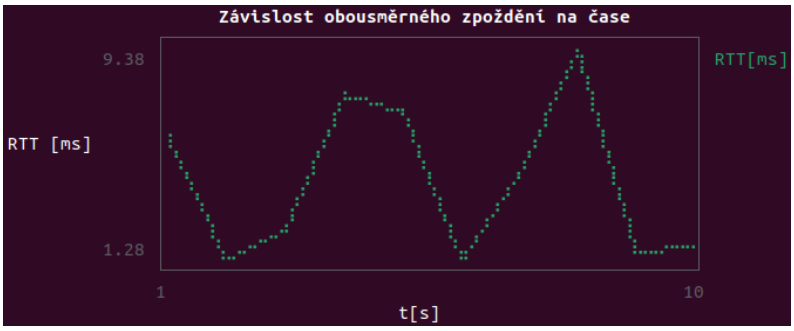
Youplot je nástrojem příkazového řádku, který umí vykreslit grafy přímo do terminálu. Ve výpisu 7.24 je uvedena hlavní část funkce obsahující tuto funkcionalitu. Po zavolání této funkce je uživatel vyzván k definici:

- názvu souboru, z něhož se má číst,
- názvu grafu,
- názvu osy y,
- zadání šířky grafu,
- výšky grafu.

Proměnná *xvar* obsahuje příkaz *awk*, který přečte data z prvního sloupce zadaného souboru. Tento sloupec obsahuje hodnoty času uvedené v sekundách. Tyto hodnoty jsou vykresleny na ose x. Příkaz *cat*, který vytiskne obsah zvoleného souboru, je zřetězen pomocí *pipe* s *youplot*, jenž poté vykreslí data. Příkaz *youplot* má následující parametry: *lineplots* nastaví spojnicový typ grafu, *-d* značí použitý oddělovač dat, *-H* data obsahují popis záhlaví, *-t* nastaví nadpis grafu, *xlabel* název parametru vykreslovaného na ose x, *ylabel* název parametru vykreslovaného na ose y, *-w* nastavení šířky a *-h* nastavení šířky grafu. Uvozovky nad proměnnýma *$title* a *$yvar* umožňují zadání víceslovného řetězce. Jako poslední je uveden příkaz *sleep* zajišťující, že vykreslený obraz zůstane v okně terminálu po dobu dvou minut.

Výpis 7.24: Vykreslení změřených dat pomocí příkazu youplot

```
youplotcsv(){
...
xvar=$(awk -F "," 'NR==1{print $1}'measurement/$file.csv)
echo "Plotting data"
cat measurement/$file.csv | youplot lineplots -d , -H
-t "$title" --xlabel $xvar --ylabel "$yvar" -w $w -h $h
sleep 120 }
```



Obr. 7.8: Ukázka grafického výstupu příkazu youplot

8 Porovnání účinnosti kompresních formátů

Pro testování byly vybrány dva kompresní formáty: **AVC** a **HEVC**. Byly zvoleny z důvodu, že jsou v praxi hojně využívány pro multimediální přenosy v reálném čase a splňují požadavky pro poměrně rychlé zakódování videa vysoké kvality. Kompresní formáty AV1 a VP9 nebyly testovány z důvodu, že se spíše používají pro webové a VoD služby. Videosoubory byly připravovány nástrojem FFmpeg, jehož funkcionalita je součástí testovací sady a je dostupná přes třetí volbu uživatelského rozhraní. Základní videosoubor měl rozlišení 4K (4000x2250) se snímkovou frekvencí o velikosti 60. Tento soubor byl upraven na segment trvající 180 sekund. Při generování AVC videí docházelo ke snižování rozlišení ze 4K60FPS na 2K60FPS, 1080p60FPS, 720p60FPS, 480p60FPS a 360p60FPS. Následně pro každé rozlišení byla snížena snímková frekvence ze 60 na 48, 30 a 24. HEVC videosoubory byly generovány přes volbu 3a (HEVC kódování videa). U těchto videí nedocházelo ke snížení rozlišení a snímkové frekvence, ale pouze byly překódovány z AVC na HEVC formát. Ve finále bylo tedy vygenerováno 48 progresivních souborů. Celkový počet snímků pro videoobsah se snímkovou frekvencí je 60, 48, 30 a 24 byl 4320, 5400, 8640 a 10800. Na jednodeskovém počítači Raspberry Pi 4B je možné najednou překódovat až šest videí s rozlišením 4K a 2K, než dojde k úplnému vyčerpání systémových prostředků. Videa s nižším rozlišení lze překódovat současně v dvojnásobném počtu oproti předchozímu uvedenému případu.

Výsledné soubory byly poté analyzovány z hlediska objemu dat, obrazového toku a bitové rychlosti IP přenosu. Velikost obrazových dat je dána součinem průměrné bitové rychlosti obrazu s celkovou dobou trvání videa. Velikost zvukových dat je dána součinem vzorkovací frekvence s bitovou hloubkou, s počtem kanálu (tyto složky dohromady tvoří bitovou rychlost zvuku) a s dobou trvání videa. Z toho vyplývá, že celková velikost souboru je právě rovna součtu těchto dvou zmíněných složek (obrazových a zvukových dat). Poslední analýza se týkala měření parametrů v souvislosti se schopností dekódovat jednotlivá videa na jednodeskovém počítači Raspberry Pi model 4B. Pro tento účel byl napsán krátký skript v němž jsou využity linuxové příkazy v kombinaci s nástrojem *ffmpeg*. Nejprve je určen název složky v níž se nachází příslušná videa. Poté pomocí *for* cyklu je tato složka procházena a dochází k načítání názvu videa s předáním do příkazu *ffmpeg*. Příkaz *ffmpeg* byl definován přepínačem *-benchmark*. Tento přepínač spustil testování průběhu dekódování videa, jenž bylo předáno a načteno přepínačem *-i*. Po dokončení měření byl výstup upraven příkazem *tr*, ten zajistil odstranění nepotřebných znaků s následným uložením do textového průběžného souboru. Textový soubor byl dále upraven příkazem *sed 's/ +/,/g'*, který data ještě vhodněji uspořádal a oddělil pomocí čárky. Tento výstup byl uložen do finálního csv souboru, z něhož již byly vytvořeny grafické průběhy.

8.1 Dekódování a přenos AVC videa

První testování proběhlo v rámci **AVC** kompresního formátu. Bylo vytvořeno celkem 24 AVC souborů, jejichž velikosti jsou uvedeny v tabulce 8.1. Mezi velikostí a rozlišením souboru lze pozorovat přímou úměrnost. Se zvyšující se kvalitou rostl celkový objem dat. 4K videa zabírala nejvíce místa na disku, naopak videa 360p měla nejmenší velikost. Rozdíl mezi těmito velikostmi byl 93 %. Zvyšující se hodnota snímkové frekvence zvětšovala finální velikost souboru.

	4K	2K	1080p	720p	480p	360p
	[MB]	[MB]	[MB]	[MB]	[MB]	[MB]
60FPS	239,2	114,7	72,4	40,9	25,3	18,7
48FPS	225,2	107,5	68,4	38,2	23,7	17,6
30FPS	204,8	98,1	62,9	36,1	23,0	17,3
24FPS	193,6	92,5	61,7	35,1	22,1	16,9

Tab. 8.1: Celková velikost pro AVC videa

V tabulce 8.2 (viz níže) jsou zaznamenány průměrné hodnoty pro obrazový tok jednotlivých AVC videí. V rámci uživatelského menu testovací sady (*dp.sh*) byla průměrná bitová rychlost zobrazena pomocí volby 3h. Rychlejší scény videa značí vyšší hodnoty bitové rychlosti obrazu. Tyto parametry ovlivňují bitovou rychlost přenosu videa. Pro přenos náročnějších scén je zapotřebí vyšší bitové rychlosti provázané s nižším zpožděním v počítačové síti. Průměrná bitová rychlost byla nejvyšší pro videorozlišení 4K60FPS, naopak nejnižší hodnota byla změřena pro video 360p24FPS. Tento rozdíl byl téměř 10 Mb/s (procentuální rozdíl 96,1 %).

	4K	2K	1080p	720p	480p	360p
	[kb/s]	[kb/s]	[kb/s]	[kb/s]	[kb/s]	[kb/s]
60FPS	10224,075	4742,326	2861,787	1461,697	768,273	475,823
48FPS	9658,678	4425,085	2687,657	1345,546	700,653	432,843
30FPS	8751,843	4007,990	2443,620	1254,044	671,617	418,560
24FPS	8256,019	3763,842	2394,797	1212,612	635,824	401,668

Tab. 8.2: Průměrný obrazový tok pro AVC videa

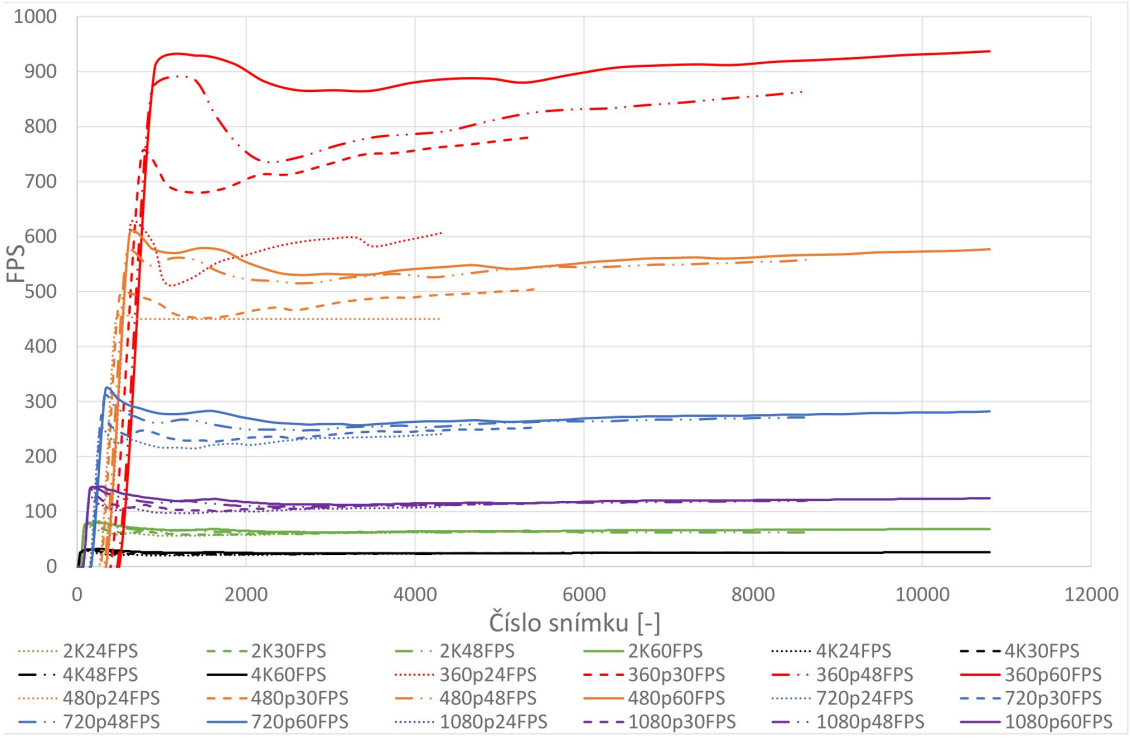
Dalším zaznamenaným typem parametru byla bitová rychlost IP přenosu. Videa byla přenášena přes síťovou topologii znázorněnou na obrázku 6.1. Doba přenosu odpovídala době trvání videa, tedy celým třem minutám. K měření byl zvolen program *tshark*, jenž je součástí testovací sady. Pomocí volby 9d byla bajtová rychlost převedena na rychlost bitovou s následným uložením těchto dat do csv souboru,

z něhož byla data následně graficky znázorněna. Průměrné hodnoty bitové rychlosti jsou uvedeny v tabulce 8.3 níže. Pro videa s vyšším rozlišením byly změřeny vyšší hodnoty bitové rychlosti IP přenosu. Pro tato videa je nutné zajistit přenos s dostatečnou šířkou pásma. Vytvořená síť poskytující propustnost 100 Mb/s byla dostačující a výsledná kvalita přenosu nebyla vůbec ovlivňována. Nejvyšší hodnota byla změřena při přenosu 4K60FPS s průměrnou bitovou rychlostí přibližně 11 Mb/s. Naopak nejnižší hodnota byla zaznamenána u videa nejnižší kvality 360p24FPS. Při streamování videa s rozlišením 480p došlo oproti videu 360p k nárůstu průměrné bitové rychlosti o 47 %. Rozdíl mezi HD (720p) a 480p byl téměř 79 %. Streamování videa Full HD (1080p) přineslo oproti předchozímu rozlišení další významný nárůst, a to o 76 %. Nejvyšší datové toky byly změřeny pro videa 4K a 2K.

	4K	2K	1080p	720p	480p	360p
	[kb/s]	[kb/s]	[kb/s]	[kb/s]	[kb/s]	[kb/s]
60FPS	11364,182	5526,331	3553,839	2017,169	1130,508	596,613
48FPS	11297,135	5233,171	3461,948	1874,556	975,059	489,536
30FPS	9744,278	4697,715	3016,309	1695,773	894,542	449,194
24FPS	9207,369	4432,415	2952,751	1623,461	817,413	426,370

Tab. 8.3: Průměrná hodnota bitové rychlosti IP přenosu pro AVC videa

Na obrázku 8.1 je zobrazen průběh dekódování pro jednotlivá AVC videa. Průměrné hodnoty snímkové frekvence změřené při dekódování se nachází v tabulce 8.4, přičemž byly zaznamenány průměrné hodnoty tohoto parametru. Pokud je změřená hodnota nižší než původní snímková frekvence, značí to zpomalenou rychlost přehrávání. Následně tím je přehrávaný videoobsah pro uživatele obtížně sledovatelný. Nejvyšší hodnota byla zaznamenána u videa 360p60FPS (828 snímků/sekunda). V rámci tohoto měření byly změřeny nejvyšší rychlosti přehrávání pro videorozlišení 360p. Video 480p vykazovalo druhou nejlepší hodnotu. Toto video bylo přehráváno s nejvyšší průměrnou rychlostí 529 snímků za sekundu. U videa 720p byly zaznamenány téměř o polovinu nižší hodnoty než u předchozího videa. Video s rozlišením 1080p se přibližovalo průměrné hodnotě okolo 100 snímků za sekundu. Pod 100 snímků za sekundu vykazovala videa 2K a 4K. Video 2K lze bez problému přehrát, protože jeho změřené hodnoty byly vyšší než původní snímková frekvence. To se naopak nedá říci o videu 4K, jehož hodnoty byly podlimitní. Nejvyšší průměrná hodnota snímkové frekvence činila 25 snímků/sekunda pro 4K60FPS.



Obr. 8.1: Dekódování AVC videa

	4K	2K	1080p	720p	480p	360p
	[FPS]	[FPS]	[FPS]	[FPS]	[FPS]	[FPS]
60FPS	25	66	118	266	529	828
48FPS	24	63	114	254	507	738
30FPS	24	61	107	231	439	634
24FPS	22	59	101	217	405	507

Tab. 8.4: Průměrná hodnota FPS pro AVC videa

8.2 Dekódování a přenos HEVC videa

Druhé testování proběhlo s použitím **HEVC** kompresního formátu. Pro toto testování bylo vytvořeno také 24 souborů. Velikosti jednotlivých souborů jsou uvedeny v tabulce 8.5 níže. Soubory byly vytvořeny dekódováním z původních AVC videí. Došlo k výraznému snížení objemu dat. V rámci stejného rozlišení se vytvořená videa v objemu dat příliš nelišila. Největší rozdíl byl zaznamenán pro video 4K, a to mezi 60FPS a 24FPS.

	4K	2K	1080p	720p	480p	360p
	[MB]	[MB]	[MB]	[MB]	[MB]	[MB]
60FPS	85	45,2	31,8	21	15,4	12,8
48FPS	82,6	44	31,1	20,6	15	12,5
30FPS	80,6	43,3	30,4	20,2	14,8	12,3
24FPS	78,7	42,1	30,2	19,9	14,6	12,1

Tab. 8.5: Celková velikost pro HEVC videa

V tabulce 8.6 níže jsou zaznamenány průměrné hodnoty bitové rychlosti, které byly opět zobrazeny pomocí funkce definované v testovací sadě diplomového skriptu. Nejvyšší hodnota bitové rychlosti obrazu byla u videa 4K60FPS (3424,515 kb/s) a nejnižší opět u videa 360p24FPS, u něhož byla změřena hodnota 190,083 kb/s. Rozdíl mezi těmito hodnoty činil 94,6 %. Se snižujícím se rozlišením klesaly také hodnoty tohoto parametru, což značí přímou úměrnost mezi rozlišením a bitovou rychlostí videa.

	4K	2K	1080p	720p	480p	360p
	[kb/s]	[kb/s]	[kb/s]	[kb/s]	[kb/s]	[kb/s]
60FPS	3424,515	1651,961	1056,335	578,207	328,161	213,048
48FPS	3320,342	1606,430	1032,612	563,142	314,037	205,255
30FPS	3231,001	1575,074	1002,615	546,921	305,405	196,701
24FPS	3150,457	1525,685	993,055	538,939	299,455	190,083

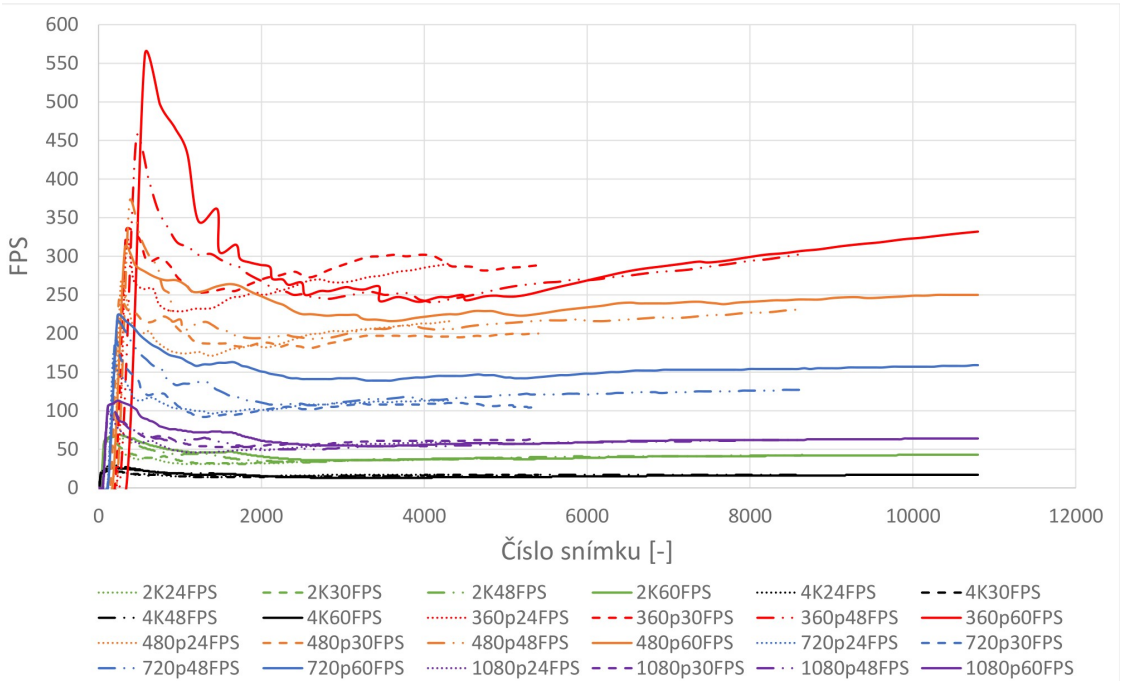
Tab. 8.6: Průměrný obrazový tok pro HEVC videa

Stejně jako v předchozím případě, IP přenos HEVC multimediálního obsahu proběhl přes vytvořenou topologii. Data byla zpracována a vykreslena stejným způsobem. Z důvodu vysoké účinnosti HEVC komprese došlo k významnému snížení bitové rychlosti. V tomto odstavci jsou dále porovnávány procentuální nárůsty bitové rychlosti oproti předchozímu videu. Video 360p bylo streamováno s maximální bitovou rychlostí 266 kb/s. U videa 480p došlo k nárůstu o 41 %. HD video bylo přenášeno rychlostí pod hranicí 1 Mb/s. Procentuální nárůst činil 105 %. Full HD video je nutné přenášet s propustností okolo hodnoty 1 Mb/s, což odpovídá procentuálnímu nárůstu 37 %. Tento nárůst byl nižší než v předchozím případě. 2K video se již dostalo na hodnotu 2 Mb/s. Nejvyšší bitová rychlost byla změřena pro 4K video. Průměrná přenosová rychlost dosáhla 4 Mb/s.

	4K	2K	1080p	720p	480p	360p
	[kb/s]	[kb/s]	[kb/s]	[kb/s]	[kb/s]	[kb/s]
60FPS	4077,158	2178,828	1145,746	771,351	375,423	266,443
48FPS	3839,869	1897,524	1058,147	726,026	365,063	253,833
30FPS	3487,228	1729,302	1032,458	657,741	324,628	215,141
24FPS	3392,966	1638,438	1006,348	616,341	311,715	201,895

Tab. 8.7: Průměrná hodnota bitové rychlosti IP přenosu pro HEVC videa

V případě použití kompresního formátu HEVC bylo možné přehrávat pouze videa 360p, 480p, 720p, 1080p, 2K24FPS a 2K30FPS. Pro videa 4K a 2K byly změřeny příliš nízké průměrné hodnoty snímků za sekundu. Z tohoto důvodu by nedocházelo k plynulému přehrávání videa koncovému uživateli. Podmínku pro plynulé přehrávání splňovala pouze videa 2K24FPS a 2K30FPS, jejichž změřené hodnoty se rovnaly 35, resp. 37 snímkům za sekundu. Průměrné hodnoty pro další videa těchto rozlišení byly nedostačující. Nejvyšší průměrné hodnoty byly změřeny pro video 360p. Následovalo video s rozlišením 480p, jehož průměrné hodnoty FPS byly menší. Pro videorozlišení 720p byly změřeny hodnoty, jež se nedostaly k hranici 100 snímků. Tyto hodnoty byly zaznamenány až u videa 1080p a pohybovaly okolo 60 snímků.



Obr. 8.2: Dekódování HEVC videa

	4K	2K	1080p	720p	480p	360p
	[FPS]	[FPS]	[FPS]	[FPS]	[FPS]	[FPS]
60FPS	16	41	63	151	235	286
48FPS	17	40	61	122	212	272
30FPS	16	37	60	107	190	270
24FPS	16	35	54	107	187	244

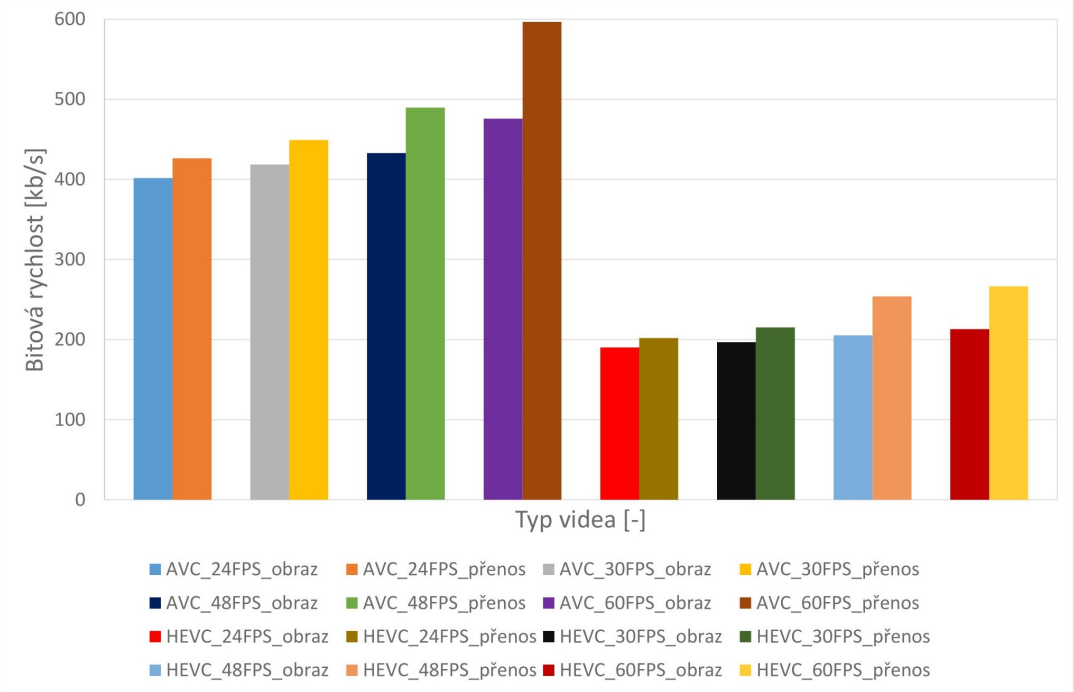
Tab. 8.8: Průměrná hodnota FPS pro HEVC videa

8.3 Porovnání bitové rychlosti

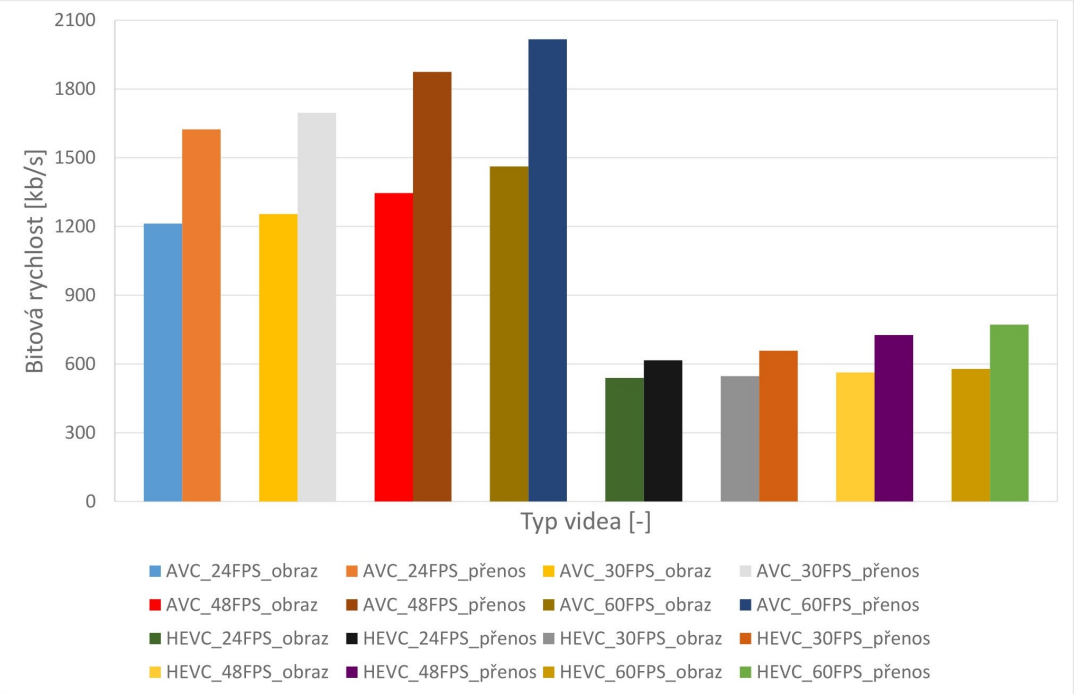
Bitová rychlost obrazu a přenosu jsou dva rozdílné parametry, avšak úzce spolu souvisí. Při IP přenosu dochází vlivem přidání řídících dat k nárůstu bitové rychlosti. Řídící data jsou potřebná ke správnému fungování samotného přenosu počítačovou sítí. Ve sloupcových grafech byl znázorněn nárůst bitové rychlosti obrazu na bitovou rychlost přenosu. Grafy jsou tvořeny dvojicí parametrů, parametrem obrazového toku a parametrem rychlosti IP přenosu. Srovnání bitové rychlosti bylo provedeno pro videa 360p, 720p, 1080p a 4K, protože tato rozlišení jsou nejvíce využívána ze strany poskytovatelů multimediálních služeb.

Bitová rychlost obrazu a přenosu pro video 360p je znázorněna na obrázku 8.3. Nejvyšší bitové rychlosti byly zjištěny pro video kódované AVC formátem s rychlostí 60 snímků za sekundu. To samé platí i v případě HEVC formátu. HEVC formát data výrazně komprimoval, a tak byla rychlost snížena o 55 %. Nárůst bitové rychlosti vzhledem k obrazovému toku nebyl příliš výrazný. AVC video 360p24FPS má o polovinu vyšší hodnoty bitové rychlosti než HEVC video 360p60FPS.

Další srovnání se týká HD videa (viz obrázek 8.4). Kompresní HEVC formát zajistil snížení bitové rychlosti o 42 %. V případě AVC videa činila nejvyšší bitová rychlost 2 Mb/s. I v tomto případě byl zaznamenán nejvyšší nárůst z obrazové na přenosovou rychlost. Menší rozdíly mezi obrazovou a bitovou rychlostí byly změřeny v případě HEVC, tyto rozdíly se naopak zvětšovaly pro jednotlivé typy AVC. HEVC 720p60fps dosahovalo menší bitové rychlosti přenosu než u AVC 720p24FPS. Rozdíl činil 852 kb/s (53 %). Je tedy možné použitím vhodného kompresního formátu přenést video s vyšší snímkovou frekvencí s menší přenosovou rychlostí.



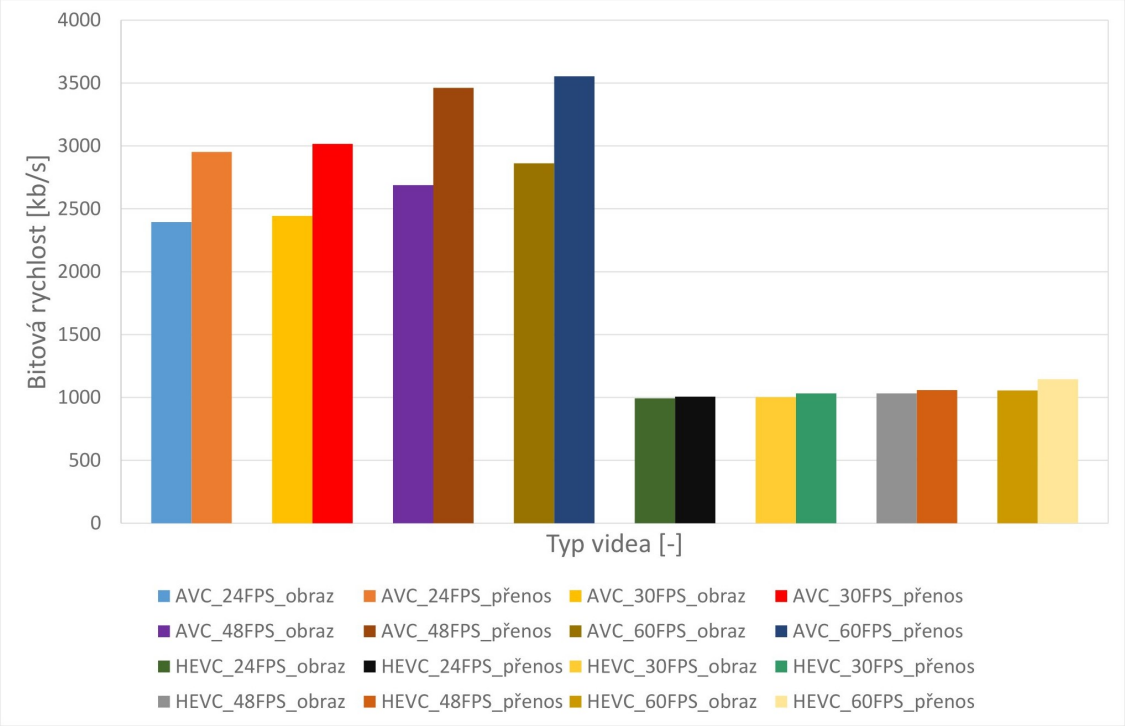
Obr. 8.3: Průměrná bitová rychlost obrazu a přenosu pro videa 360p



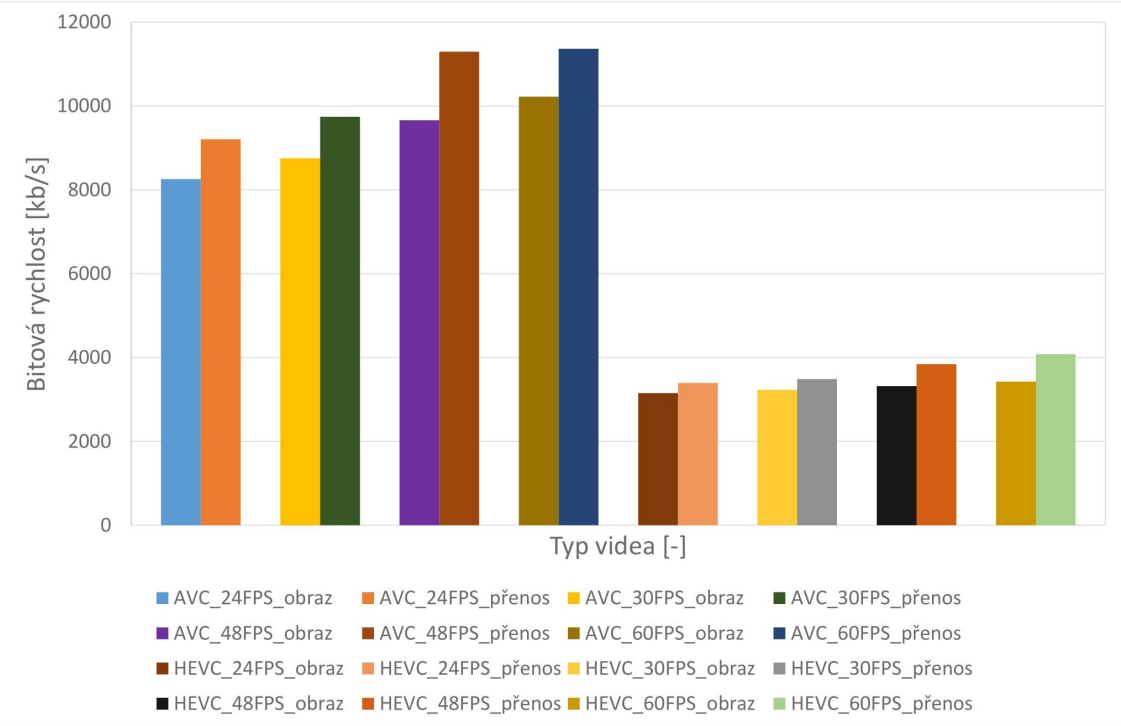
Obr. 8.4: Průměrná bitová rychlost obrazu a přenosu pro videa 720p

Na obrázku 8.5 jsou uvedeny hodnoty bitových rychlostí pro Full HD video. Maximální přenosová rychlost u AVC přesáhla 3,5 Mb/s, naopak nejnižší rychlost se blížila 3 Mb/s. Nejvyšší obrazový tok u AVC videa dosahoval 2,8 Mb/s. V porovnání s rychlostí obrazu tak řídící data přidala do přenosu průměrnou redundanci okolo 0,7 Mb/s. Pro HEVC se rychlosti pohybovaly okolo 1 Mb/s. Z toho vyplývá, že se podařilo snížit přenosové nároky přibližně o 33 %. V rámci přenosu HEVC obsahu došlo oproti hodnotám obrazového toku k nepatrnému nárůstu dat.

Jako poslední byly porovnávány hodnoty bitové rychlosti pro video 4K. Toto rozlišení charakterizuje nejvyšší obrazovou kvalitu. Vysoká kvalita obrazu zvyšuje nároky na přenos nárůstem bitové rychlosti. Ostřejší, lepší a propracovanější kvalita obrazu zvyšuje obrazový tok. Na moderních přenosových médiích je možné tento druh přenosu bez problému uskutečnit. Průměrné hodnoty bitových rychlostí byly graficky znázorněny na obrázku 8.6. Obrazový tok AVC 4K60FPS činil 10 Mb/s. U HEVC videa stejného formátu byl obrazový tok snížen o více než polovinu. Co se týče IP přenosu, režijní data přidala redundanci v průměru 1 Mb/s. Redundance řídících dat v případě HEVC přenosu byla nižší, hodnoty se pohybovaly v rozmezí 0,2-0,6 Mb/s. Přenos HEVC 4K60FPS byl zaznamenán okolo 4 Mb/s, což odpovídá snížení bitové rychlosti o 64 % (z 11 Mb/s na 4 Mb/s).



Obr. 8.5: Průměrná bitová rychlost obrazu a přenosu pro videa 1080p



Obr. 8.6: Průměrná bitová rychlost obrazu a přenosu pro videa 4K

8.4 Souhrnné porovnání

V rámci této podkapitoly jsou porovnávány výsledky testů kompresních formátů AVC a HEVC. Z tabulek 8.1 a 8.5 je patrné, že kompresní HEVC formát dokáže významně snížit celkovou velikost videosouboru. Tato vlastnost je především výhodná při ukládání velkého množství souborů na paměťová média. Účinnost komprese se zvyšuje s rostoucím rozlišením videa. Nejúčinnější komprese z hlediska snížení celkové velikosti proběhla pro video 4K, kde v porovnání s AVC formátem dochází ke snížení objemu o více než 50 %. To samé platí i pro 2K video, i když zde byly hodnoty snížení nepatrně nižší. Rozdíl mezi 720p soubory byl již menší než 50 %. Pro videa 480p a 360p byl tento rozdíl ještě nižší. Komprese dat se také projevila na průměrné bitové rychlosti obrazu, která jde ruku v ruce s celkovou velikostí souborů. Rozdíly jsou znázorněny přehledně v tabulkách 8.2 a 8.6. U 4K60FPS byla snížena průměrná bitová rychlost z 10,2 Mb/s na 3,4 Mb/s, to odpovídá více než 50 % snížení velikosti oproti původnímu souboru. Opět zde platí, že čím větší rozlišení video mělo, o to víc byla průměrná bitová rychlost videa snížena. V rámci případného síťového přenosu se snížení bitové rychlosti projeví pozitivně tím, že bude ušetřena značná část šířky pásma přenosového média. Bitová rychlost IP přenosu AVC videa byla ve všech případech vyšší než pro HEVC videa. Při streamování videa o stejné

kvalitě obrazu dochází k významnému poklesu bitové rychlosti. Snížení celkového objemu, resp. obrazového toku, se příznivě projeví na výsledné přenosové rychlosti. Tyto parametry jsou úzce spjaty.

Průběhy dekódování videa jsou znázorněny na obrázcích 8.1 a 8.2. Při dekódování AVC videa byly změřeny vyšší hodnoty snímků v čase, a to pro všechna videa. Tento jev nastal z důvodu toho, že HEVC soubory byly vytvářeny z původních AVC souborů. AVC videa byla již z principu dostatečně komprimována. Zdrojové AVC soubory neměly příliš vysoké hodnoty bitové rychlosti obrazu, a proto došlo při překódování k občasné ztrátě kvality. Ztráta kvality videa se projeví v rámci snížených hodnot snímkové frekvence (viz obrázek 8.2). Překódování z AVC formátu na HEVC formát je vhodné pouze pro videa vyšší kvality, tedy videa s dostatečně vysokou bitovou rychlostí obrazu.

Závěr

Úvodní část diplomové práce pojednává o vlastnostech multimediálních služeb v IP sítích. První kapitola se týká teoretického rozboru základních typů multimediálních přenosů s následným porovnáním jejich vlastností. Následuje teoretická kapitola, v níž jsou rozebrány typy přenosů paketů běžně používaných v IP sítích. Detailní rozbor je věnován multicastovému vysílání a jeho příslušným náležitostem. V třetí kapitole jsou rozebrány protokoly, jež umožňují přenos v reálném čase, podávání zpětné vazby o kvalitě přenosu, odesílání oznámení o relaci a zajišťování správného průběhu přenosu. Předposlední kapitola obsahuje definici a rozbor jednotlivých přenosových parametrů, které značně ovlivňují kvalitu služeb. Také jsou popsány příčiny vzniku deformací zvuku a obrazu. Poslední kapitola pojednává o běžně používaných rozlišeních a různých kompresních formátech.

Cílem praktické části práce bylo vytvoření testovací sady pro uskutečnění různých typů síťových přenosů s celkovou analýzou klíčových síťových parametrů. Testovací sada byla vytvořena ve skriptovacím jazyce Bash. Finální soubor se nazývá *dp.sh*. Pro automatickou instalaci nástrojů, které jsou součástí sady, byl vytvořen skript *installation.sh*. Hlavní skript je rozdělen do dílčích funkcí různých vlastností. Součástí tohoto řešení je ucelené uživatelské konzolové rozhraní, jenž zobrazuje nabídku všech možností. Hlavními prvky testovací sady jsou dva typy generátorů, které je možné nasadit do libovolné síťové topologie. Funkce pro datový přenos umožňuje generování dat dle libovolné předlohy. Uživatel si může vybrat mezi dvěma transportními protokoly TCP a UDP. Před samotným multimediálním přenosem je možné provést úpravu dat pomocí funkcí definovaných nástrojem *ffmpeg*. Multimediální data mohou být překódována různými kompresními formáty. Testovací sada dále obsahuje funkce ke zkrácení délky času, změně snímkové frekvence a změně rozlišení. Multimediální funkce umožňuje spouštění unicastových či multicastových multimediálních přenosů ke koncovému zařízení. Pro příjem a zobrazení multimediálních dat byla vytvořena přijímací funkce. K výsledné analýze a grafickému zobrazení změřených dat byly sestaveny funkce, které je možné využívat v rámci textového prostředí, a to v reálném čase, nebo si lze data uložit pro pozdější hloubkovou analýzu.

Poslední část praktické části diplomové práce je zaměřena na porovnání kompresních formátů AVC a HEVC z hlediska objemu dat, průměrné bitové rychlosti obrazu a IP přenosu. Dále byl měřen průběh dekódování videa pro oba formáty. Tyto hodnoty byly průměrovány. Pokud byla výsledná hodnota snímků za sekundu nižší než původní snímková frekvence, značilo to zpomalenou rychlost přehrávání. Přehrávaný videoobsah byl tak pro uživatele těžce sledovatelný. Měření probíhalo na jednodeskovém počítači Raspberry Pi 4B. Hlavním úkolem kompresních formátů je snížení výsledného datového toku v průběhu síťového přenosu a zmenšení nároků

na paměťová média. Těchto vlastností se využívá především při přenosu videa sítí s nižší propustností. Díky kompresním formátům je možné využít stávající metalické infrastruktury, které poskytují přenosovou šířku pásma alespoň 100 Mb/s. Výsledky měření ukazují, že HEVC dokáže snížit celkovou velikost souboru o více než 50 %. Bitová rychlost obrazu byla redukována v závislosti na dané kvalitě multimediálního obsahu. Pro videa s vyšší kvalitou docházelo ke zvýšení účinnosti redukce. Nárůst bitové rychlosti IP přenosu vzhledem k obrazovému toku nebyl konstantní, tento nárůst se opět zvyšoval dle kvality videoobsahu. Nevýhoda kompresního formátu HEVC spočívá v tom, že pokud je při překódování již použit AVC multimediální soubor, jenž je z principu dostatečně komprimovaný, nemá dostatečně vysokou bitovou rychlost. V průběhu komprese dochází nejen ke zmenšení finální velikosti souboru, ale také ke ztrátě kvality. Pokud je tedy video již dostatečně komprimováno, je tento druh kódování vhodné použít pouze pro videa vysoké kvality. Tento kompresní formát je možné použít pro nekomprimovaný surový soubor dat. HEVC kodek vyžaduje větší výpočetní výkon a operace kódování/dekódování trvá delší čas. Volba mezi těmito dvěma kompresními formáty závisí na konkrétním využití a potřebách uživatele.

Literatura

[1] SIMPSON, Wes a Howard GREENFIELD. *IPTV and Internet Video: Expanding the Reach of Television Broadcasting*. 2nd edition. Oxford: Elsevier, c2009. ISBN 978-0-240-81245-8.

[2] O-DRISCOLL, GERARD. *NEXT GENERATION IPTV SERVICES AND TECHNOLOGIES*. New Jersey: Wiley, c2008. ISBN 978-0-470-16372-6.

[3] XIAO, Yang, Xiaojiang DU, Jingyuan ZHANG, Fei HU a Sghaier GUIZANI. Internet Protocol Television (IPTV): The Killer Application for the Next-Generation Internet. *IEEE Communications Magazine*. 2007, **45**(11), 126-134. ISSN 0163-6804. Dostupné z: doi:10.1109/MCOM.2007.4378332.

[4] BING, Benny. *NEXT-GENERATION VIDEO CODING AND STREAMING*. New Jersey: Wiley, c2015. ISBN 978-1-118-89130-8.

[5] SIMPSON, Wes. *Video Over IP: IPTV, Internet Video, H.264, P2P, Web TV, and Streaming: A Complete Guide to Understanding the Technology*. 2nd edition. Oxford: Focal Press, c2008. ISBN 9780240810843.

[6] HARTE, Lawrence. *IPTV Testing*. Fuquay-Varina: Althos Publishing, c2008. ISBN 1-932813-88-8.

[7] MINOLI, Daniel. *IP MULTICAST WITH APPLICATIONS TO IPTV AND MOBILE DVB-H*. New Jersey: Wiley, c2008. ISBN 978-0-470-25815-6.

[8] HARTE, Lawrence. *Introduction to Data Multicasting*. Fuquay-Varina: Althos Publishing, c2008. ISBN 1-932813-55-1.

[9] JOSEPH, Vinod a Srinivas MULUGU. *Deploying Next Generation Multicast-Enabled Applications: Label Switched Multicast for MPLS VPNs, VPLS, and Wholesale Ethernet*. Waltham: Elsevier, c2011. ISBN 978-0-12-384923-6.

[10] JEŘÁBEK, Jan. *Pokročilé komunikační techniky*. 2. dopl. vyd. Brno: Fakulta elektrotechniky a komunikačních technologií Ústav telekomunikací, 2022.

[11] LOVELESS, Josh, Ray BLAIR a Arvind DURAI. *IP Multicast, Volume 1: Cisco IP Multicast Networking*. Indianapolis: Cisco Press, c2017. ISBN 978-1-58714-459-2.

[12] DAVIES, Guy. *Designing and Developing Scalable IP Networks*. Hoboken, NJ: Wiley, c2004. ISBN 0-470-86739-6.

[13] KALMANEK, Charles R. a Y. Richard YANG. *Guide to Reliable Internet Services and Applications*. New York: Springer, c2010. ISBN 978-1-84882-827-8.

[14] HENS, Francisco J. a José Manuel CABALLERO ARTIGAS. *Triple play: building the converged network for IP, VoIP and IPTV*. Chichester: Wiley, c2008. ISBN 978-0-470-75367-5.

[15] W. BARZ, Hans a Gregory A. BASSETT. *MULTIMEDIA NETWORKS: PROTOCOLS, DESIGN, AND APPLICATIONS*. West Sussex: Wiley, c2016. ISBN 9781119090137.

[16] J. HANDLEY, Mark, Colin PERKINS a Edmund WHELAN. Session Announcement Protocol: RFC 2974. *IETF Datatracker* [online]. IETF, 2013 [cit. 2022-10-02]. Dostupné z: https://datatracker.ietf.org/doc/rfc2974/

[17] KREJČÍ, J. a T. ZEMAN. Hodnocení kvality IPTV. *Access Server* [online]. Praha: České vysoké učení technické, c2007 [cit. 2022-09-26]. Dostupné z: http://access.fel.cvut.cz/view.php?cisloclanku=2010050004

[18] PEFKIANAKIS, Ioannis, Henrik LUNDGREN, Augustin SOULE, et al. Characterizing home wireless performance: The gateway view. *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, 2015, 2713-2731. ISBN 978-1-4799-8381-0. Dostupné z: doi:10.1109/INFOCOM.2015.7218663.

[19] RICHARDSON, Iain E. G. *H.264 and MPEG-4 video compression: video coding for next-generation multimedia*. Chichester: Wiley, c2003. ISBN 04-708-4837-5.

[20] RAO, K. R., Do Nyeon KIM a Jae Jeong HWANG. *Video Coding Standards: AVS China, H.264/MPEG-4 PART 10, HEVC, VP6, DIRAC and VC-1*. London: Springer, c2014. ISBN 978-94-007-6741-6.

[21] SZE, Vivienne, Madhukar BUDAGAVI a Gary Joseph SULLIVAN. *High efficiency video coding (HEVC): algorithms and architectures*. Cham: Springer, [2014]. Integrated circuits and systems. ISBN 978-3-319-06894-7.

[22] WANG, Yunqing, Chi Yo TSAI, Jingning HAN a Yaowu XU. High Performant AV1 for VOD Applications. *2022 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. IEEE, 2022, 2022-7-18, 1-4. ISBN 978-1-6654-7218-0. Dostupné z: doi:10.1109/ICMEW56448.2022.9859523.

[23] GOEBEL, Jones, Lucas AGOSTINI, Luciano AGOSTINI, Bruno ZATT a Marcelo PORTO. High Throughput Multiplierless Architecture for VP9 Fractional Motion Estimation. *2018 31st Symposium on Integrated Circuits and Systems*

Design (SBCCI). IEEE, 2018, 2018, 1-6. ISBN 978-1-5386-7431-4. Dostupné z: doi:10.1109/SBCCI.2018.8533255.

Seznam symbolů a zkratek

ASM Any Source Multicast

AV1 AOMedia Video 1

AVC Advanced Video Coding

ES Elementary Stream

FPS Frames Per Second

HD High Definition

HDMI High-Definition Multimedia Interface

HEVC High Efficiency Video Coding

HTTP Hyper Text Transfer Procotol

IANA Internet Assigned Numbers Authority

IEEE Institute of Electrical and Electronics Engineers

IGMP Internet Group Management Protocol

IP Internet Protocol

IPTV Internet Protocol television

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

ISM Internet Standard Multicast

LAN Local Area Network

MAC Media Access Control

MPEG Moving Picture Experts Group

MPEG-PS MPEG Program Stream

MPEG-TS MPEG Transport Stream

MPTS Multiple Program Transport Stream

NAL Network Abstraction Layer

NIC Network Interface Controller

OSPF Open Shortest Path First

OTT Over-the-top

PAT Program Association Table

PDV Packet Delay Variation

PES Packetized elementary stream

PID Program Identifier

PIM Protocol Independent Multicast

PMT Program Map Table

QoE Quality of Experience

QoS Quality of Services

RAM Random Access Memory

RFC Request For Comments

RP Rendezvous Point

RPF Reverse Path Forwarding

RSSI Received Signal Strength Indication

RTCP Real-time Transport Control Protocol

RTP Real-time Transport Protocol

RTSP Real-time Streaming Protocol

RTT Round Trip Time

SAP Session Announcement Protocol

SD Standard Definition

SDP Session Description Protocol

SOHO Self-Office Home-Office

SPT Shortest Path Tree

SSM Source Specific Multicast

TCP Transmission Control Protocol

UDP User Datagram Protocol

UHD Ultra High Definition

UTP Unshielded Twisted Pair

VLC VideoLAN Client

VoD Video on Demand

VoIP Voice over Internet Protocol

VP9 Video Predictor version 9