

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra radioelektroniky



Technologie streamingu a hodnocení kvality videa
Streaming Technologies and Video Quality Assessment

Bakalářská práce

Kristýna Žáková

Studijní program: Komunikace, multimédia a elektronika

Studijní obor: Multimediální technika

Vedoucí práce: Ing. Karel Fliegel, PhD.

Praha 2018

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 22. 5. 2018

.....

Kristýna Žáková

Poděkování

Ráda bych touto cestou poděkovala Ing. Karlu Fliegelovi, PhD. za odborné vedení práce, věcné připomínky a vstřícnost při konzultacích.

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Žáková** Jméno: **Kristýna** Osobní číslo: **456918**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra radioelektroniky**
Studijní program: **Komunikace, multimédia a elektronika**
Studijní obor: **Multimediální technika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Technologie streamingu a hodnocení kvality videa

Název bakalářské práce anglicky:

Streaming Technologies and Video Quality Assessment

Pokyny pro vypracování:

Podějte přehled aktuálních technologií pro kompresi a streaming videa. Analyzujte související metody pro subjektivní a objektivní hodnocení kvality videa pro tuto aplikaci. Realizujte model přenosového systému s možností simulace typických artefaktů vznikajících při přenosu. Soustřeďte se zejména na použití standardu MPEG-DASH.

Seznam doporučené literatury:

- [1] Seufert, M., et al.: A Survey on Quality of Experience of HTTP Adaptive Streaming, IEEE Communications Surveys & Tutorials, vol. 17, no. 1, 2015.
- [2] Wu, H. R., Rao, K. R.: Digital Video Image Quality and Perceptual Coding, Signal Processing and Communications, CRC Press, New York, 2006.
- [3] Winkler, S.: Digital Video Quality: Vision Models and Metrics, Wiley, 2005.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Karel Fliegel, Ph.D., katedra radioelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **09.02.2018**

Termín odevzdání bakalářské práce: **25.05.2018**

Platnost zadání bakalářské práce: **30.09.2019**

Ing. Karel Fliegel, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Studentka bere na vědomí, že je povinna vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studentky

Abstrakt

Práce pojednává o technologiích a přístupech ke streamingu audiovizuálního obsahu. Popisuje rozdíly mezi online a offline streamingem, adaptivním a neadaptivním streamingem a seznamuje s nejvyužívanějšími kompresními standardy a kodeky. Další části přibližují problematiku kvality streamu, způsoby jejího hodnocení. Práce také uvádí nejznámější streamingové služby a popisuje realizaci celého přenosového řetězce.

Klíčová slova

Audio, video, streaming, komprese, streamovací služby, MPEG DASH, hodnocení kvality

Abstract

The thesis deals with technologies and approaches to streaming of audiovisual content. It describes the differences between online and offline streaming, adaptive and non-adaptive streaming and introduces the most used compression standards and codecs. Other sections describe the issue of the quality of the stream and the ways of its assessment. The most popular streaming services are stated as well as the description of the transmission realization.

Key words

Audio, video, streaming, compression, streaming services, MPEG DASH, Quality Assessment

Obsah

| | |
|--|----|
| Úvod | 1 |
| 1 Streaming audiovizuálního obsahu | 3 |
| 1.1 Rozdělení streamingu | 3 |
| 1.2 Komprese multimediálního obsahu v oblasti streamingu..... | 5 |
| 1.2.1 Kompresní standardy MPEG | 6 |
| 1.2.2 Kompresní standardy H.26x | 7 |
| 1.2.3 Nejpoužívanější kompresní audio a video formáty | 8 |
| 2 Způsoby doručování obsahu | 11 |
| 2.1 Progresivní download | 11 |
| 2.2 Tradiční streaming | 12 |
| 2.3 Adaptivní streaming..... | 14 |
| 2.4 MPEG DASH..... | 15 |
| 2.4.1 Princip MPEG DASH..... | 16 |
| 2.4.2 Media Presentation Description | 17 |
| 2.4.3 Dodatečné funkce..... | 18 |
| 2.5 Srovnání řešení pro adaptivní streaming..... | 19 |
| 3 Kvalita streamu | 20 |
| 3.1 Uživatelské faktory ovlivňující QoE..... | 20 |
| 3.1.1 Uživatel a prostředí | 20 |
| 3.1.2 Obsah..... | 21 |
| 3.2 Faktory související s vnímáním a technickými vlivy..... | 21 |
| 3.2.1 Počáteční zpoždění..... | 21 |
| 3.2.2 Pozastavení..... | 22 |
| 3.2.3 Zpoždění a jitter, ztrátovost paketů a propustnost | 22 |
| 3.2.4 Přizpůsobení kvality | 23 |
| 3.2.5 Technologické parametry adaptivního streamingu..... | 23 |

| | |
|--|----|
| 3.2.6 Server..... | 24 |
| 3.2.7 Klient..... | 24 |
| 3.3 Artefakty..... | 25 |
| 4 Hodnocení kvality..... | 26 |
| 4.1 Subjektivní metody hodnocení kvality videa | 26 |
| 4.1.1 Doporučení ITU-R BT.500-13 | 26 |
| 4.1.2 Doporučení ITU-T P.910..... | 28 |
| 4.2 Objektivní metody hodnocení kvality videa | 28 |
| 4.2.1 MSE..... | 29 |
| 4.2.2 PSNR..... | 30 |
| 4.2.3 VQM..... | 31 |
| 4.2.4 SSIM..... | 31 |
| 5 Streamingové služby | 32 |
| 5.1 Netflix | 32 |
| 5.2 Spotify..... | 33 |
| 5.3 Twitch.tv | 33 |
| 5.4 Hulu | 33 |
| 5.5 Srovnání..... | 34 |
| 6 Streamingový proces..... | 36 |
| 6.1 Příprava souboru k přenosu DASH | 37 |
| 6.1.1 Použitý software..... | 37 |
| 6.1.2 Práce s videosouborem | 38 |
| 6.1.3 Automatizovaná práce s videosouborem | 41 |
| 6.1.4 Kvalita výstupních souborů | 42 |
| 6.2 Alternativní možnost získání dat | 44 |
| 6.3 Server | 44 |
| 6.3.1 Porty..... | 47 |
| 6.4 Přehrávač..... | 48 |

| | |
|--|-----------|
| 6.5 Změna stavu připojení..... | 49 |
| 7 Závěr..... | 52 |
| 8 Použité zkratky..... | 54 |
| 9 Seznam použité literatury..... | 56 |
| 10 Seznam obrázků a tabulek..... | 67 |
| 11.1 Seznam příloh..... | 68 |
| Příloha 1: Ukázka programového toku, kódování videa pro MPEG DASH..... | 69 |
| Příloha 2: Ukázka programového toku, kódování audia pro MPEG DASH..... | 71 |
| Příloha 3: Soubor zakovkri-obsah_serveru.rar..... | 71 |
| Příloha 4: Soubor zakovkri-matlab.rar..... | 71 |
| Příloha 5: Soubor zakovkri-stream-zk.mp4..... | 71 |

Úvod

Pohyb a fungování člověka na internetu je dnes samozřejmostí. Velké množství aktivit, které byly dříve prováděny na fyzických médiích a v běžném životě, se staly nedílnou součástí naší aktivity na internetu. Čteme a posíláme emaily, sledujeme zprávy, posloucháme hudbu - tato pro nás pohodlná řešení téměř plně nahrazují dříve oblíbené posílání dopisů, čtení novin, hledání nejlepších pořadů televizního vysílání, přehrávání hudby z walkmanů a přehrávačů.

Přenosu jakéhokoliv audiovizuálního obsahu po internetu je v dnešní době využíváno nejen v rámci televizního a rozhlasového vysílání internetových stanic, ale také v korporacích, na školách, mezi rodinnými příslušníky atd. Je také velmi atraktivní pro poskytovatele placeného multimediálního obsahu, jelikož v dnešní době dosahuje větší obliby mezi populací než sledování pořadů v televizi či poslech radiových stanic na domácích přijímačích.

Právě touto problematikou se tato práce zabývá. Jejím cílem je seznámit s technologiemi, které jsou využívány pro kompresi a streaming videa, metodami pro hodnocení kvality tohoto přenosu a vlastní realizace streamu od přípravy souborů k přenosu DASH (Dynamic Adaptive Streaming over HTTP) až po jejich samotné přehrání na straně příjemce. Práce volně navazuje na předchozí individuální projekt a místy využívá pasáží napsaných k výstupní dokumentaci tohoto předmětu.

V první kapitole je čtenář seznámen s fenoménem streamingu, jeho fungováním a kompresí streamovaného obsahu. Kapitola druhá uvádí způsoby doručování streamovaného obsahu - seznamuje s jednotlivými typy streamování a zaměřuje se na MPEG DASH. Další kapitola, třetí, analyzuje faktory ovlivňující kvalitu přenosu a uvádí následky jejího negativního ovlivnění. Kapitola čtvrtá shrnuje nejužívanější metody pro objektivní a subjektivní hodnocení kvality videa a odkazuje na publikace s nimi související. V páté kapitole nechybí ani uvedení několika pravděpodobně nejznámějších streamingových služeb, které jsou v současnosti hojně využívány širokou veřejností spolu se základními informacemi o nich.

Kapitola šestá je pomyslnou druhou částí práce. Zaměřuje se na praktickou realizaci přenosového řetězce. Navrhuje a realizuje zpracování vstupních dat, uvádí ukázkové objektivní zhodnocení kvality, popisuje přenos dat a jejich přehrání (a případné další zpracování) na straně příjemce.

Závěrem jsou nejen zrekapitulovány a vyhodnoceny výsledky práce, ale také prezentovány náměty pro další možné kroky, které by mohly zpracovanou problematiku více rozvinout.

1 Streaming audiovizuálního obsahu

Jako streaming označujeme způsob přenosu libovolného obsahu do počítačových a mobilních zařízení připojených k internetu. V současné době se jedná zejména o přenos audia a videa - audiovizuálního obsahu - nicméně je možné přenášet jakákoliv data - a to ve formě kontinuálního datového toku (z angl. „stream“), který proudí k cílovému klientovi a umožňuje mu přehrávat přenášený obsah téměř okamžitě [1]. Jedná se tedy o formu distribuce dat pomocí technologie cloudu [2], přenášení informací datovým proudem ze serverů se vzdáleným přístupem, které jsou dostupné na internetu [3].

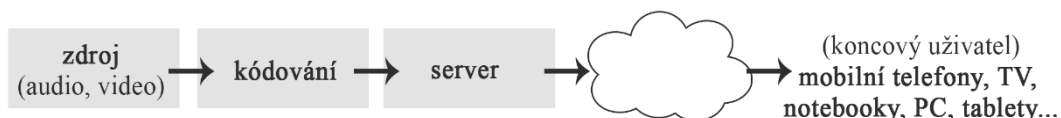
Tato práce se zabývá přenosem kompletního audiovizuálního signálu, ačkoliv je možné jeho jednotlivé složky přenášet odděleně.

Na začátku každého přenosu signálu musí dojít k samotnému pořízení záznamu na libovolném zařízení, k jeho zpracování a přípravě pro přenos. Tato data jsou dále přenesena na streamingový server (někdy označovaný také jako Media Server), kde se tok dat ukládá, kóduje, komprimuje a po částech posílá dále k příjemci. Obsah se přes webový server přenáší klientovi (příjemci), který si na svém koncovém multimediálním zařízení (může se jednat o tablet, počítač, mobilní telefon aj.) vyžádá jeho přehrávání po provedení dekomprese v reálném čase. Velmi zjednodušeně je tento řetězec vizualizován na Obrázku 1.

1.1 Rozdělení streamingu

Dle pohledu na streaming obecně lze uvést nespočet způsobů jak jej dělit. Čtenář si jistě doplní spoustu dalších dle vlastního uvážení, práce přibližuje dělení dle:

- času
- proměnlivosti bitového toku
- orientace na koncového uživatele



Obrázek 1 Základní proces streamování

Z hlediska **časového** je možné streaming rozdělit do dvou skupin podle toho, kdy přesně může příjemce obsah prohlížet. První skupinu označujeme jako

webcasting (resp. online streaming) a druhou jako VoD (Video on Demand, resp. offline streaming).

Online streaming je živé internetové vysílání. Zdroj celého vysílání, například standartní videokamera, je připojen ke kodéru, jehož mechanismus je zodpovědný za zpracování natáčené informace, její kompresi a přenos těchto dat na server. Ten šíří komprimovanou informaci ke všem klientům, kteří jsou připojeni k vysílání, přímo či skrze dodatečné servery (tzv. „splitters“) [4]. Tyto dodatečné servery jsou připojeny k vysílání a živě streamované video se jimi pouze dále jednosměrně šíří k příjemcům (za možnosti oboustranné kontroly, která bude zmiňována v kapitolách dále).

Jako příklad webcastu je možno uvést přenos sportovních utkání, audiovizuální události (živé koncerty, internetová rádia), internetové televize, přenos konferencí atd.

Při přehrávání obsahu VoD se nejedná o přehrávání v reálném čase, nýbrž - zjednodušeně řečeno - ze záznamu, většinou dlouhodobě dostupného, který si divák k přehrávání vybere. Na rozdíl od online streamingu není zdrojové zařízení přímo propojeno se serverem - natočený a zpracovaný klip je nutno nejdříve nahrát na disk, z něhož je „přebírán“ serverem k další distribuci přes internet (či lokální síť). Velkou výhodou je, že příjemce má možnost ovládat přehrávání a sledovaný obsah pozastavovat, posouvat, přehrát znovu, přeskočit na určitý čas záznamu aj. Tyto možnosti v online streamingu nemá, není schopen přenos takto volně ovládat, jelikož vysílání plynule probíhá.

Příkladem mohou být například uložené přenosy webcastingových pořadů, videa, filmy, seriály uložené na internetových serverech.

Z hlediska **proměnlivého bitového toku** se rozlišuje neadaptivní a adaptivní streaming.

Streaming neadaptivní se v dnešní době již příliš nevyužívá. Divák si při výběru obsahu, jež chce přehrát, volí také kvalitu, ve které chce obsah sledovat - a to vyjádřenou rozlišením či datovým tokem. Někdy přehrávač volí kvalitu obsahu sám po automatickém testování internetového připojení klienta. Čím vyšší rozlišení (datový tok) je zvoleno, tím je zpravidla přenos náročnější na propustnost připojení. Jakmile však streamování začne, v jeho průběhu se kvalita již nezmění.

Adaptivní streaming však, oproti neadaptivnímu, pracuje s aktuální kvalitou připojení a sám na jejím základě mění kvalitu obsahu v průběhu přehrávání tak,

aby uživatel nezaznamenal větší obtíže s jeho přehráváním a byl schopen jej sledovat plynule bez přerušení.

Obsah lze dělit jako amatérský a volně šiřitelný (od veřejnosti k veřejnosti, příkladem například server YouTube¹); obsah tvořen profesionály k propagaci jejich služeb (reklamy, propagační videospoty aj.) a obsah přímo distribuovaný na internetu za účelem zisku (přístup k elektronickým datům je zpoplatněn). Mezi poslední zmiňované patří služby jako Netflix² a Apple TV³, které v poslední době zaznamenávají veliký rozvoj a jejich vliv na trhu se neustále zvyšuje.

Streamování videí je v dnešní době převládající metodou pro šíření obsahu online na klientská zařízení. Dle [5] je celosvětový měsíční přenos videoobsahu na internetu odhadován na 42 PB (73% internetového provozu). Je očekáván nárůst tohoto čísla - a to až na 159 PB v roce 2021, což by odpovídalo přibližně 82% celého internetového provozu. Content Delivery Networks (CDN) - tedy sítě poskytující velký multimediální obsah, jako je například server YouTube, v roce 2016 představovaly 52% celosvětového internetového provozu (na rok 2021 je předpovídáno celých 71%).

1.2 Komprese multimediálního obsahu v oblasti streamingu

Běžné nekomprimované digitální video 60 minutové délky zabírá stovky GB prostoru surových dat. Takto objemné soubory by bylo velmi obtížné (respektive prakticky nemožné) streamovat po dnešní síti, ukládat a dále s nimi pracovat. Je tedy třeba před začátkem přenosu zredukovat jejich velikost v co největší míře tak, aby byla zachována uspokojující kvalita obsahu při jeho sledování.

V souvislosti s touto problematikou je používáno pojmu „kodek“ - což je v podstatě set instrukcí, které identifikují metodu, jíž jsou data komprimována, a zároveň dělají pravý opak v případě, že je obsah přehráván. [6] [7] Tedy *kodek = kodér + dekodér*. Při volbě použitého kodeku je třeba zohlednit nejen účinnost komprese, ale také jejich výpočetní náročnost.

¹ YouTube <https://www.youtube.com/>

² Netflix <https://www.netflix.com/cz/>

³ Apple TV <https://www.apple.com/cz/tv/>

V dnešní době nejpoužívanější kompresní standardy pro video sekvence jsou vyvíjeny organizacemi ISO/IEC Moving Pictures Experts Group (MPEG)⁴ a ITU-T Video Coding Experts Group (VCEG)⁵.

1.2.1 Kompresní standardy MPEG

Skupina MPEG (celým názvem Moving Pictures Experts Group), pracující v rámci společné technické komise vytvořené organizacemi ISO (International Organization for Standardization) a IEC (International Electrotechnical Commission), vyvíjí řadu nejpoužívanějších standardů pro kompresi audiovizuální informace.

Všechny standardy MPEG využívají převážně ztrátové komprese na principu transformačních kodeků, kdy jsou vzorky děleny do segmentů a po transformaci do oblasti prostorových kmitočtů kvantovány a kódovány. Proces kódování a dekódování využívaný pro standardy MPEG je popsán například v [8].

Začínající MPEG standardy byly specializovány zejména na ukládání audiovizuálních dat na datové nosiče využívané v externích zařízeních (DVD, CD). Postupem času se však přistoupilo k jejich všeobecnému využití.

MPEG-1 (1993, ISO/IEC 11172) je standard původně navržen hlavně pro ukládání dat na digitální datové nosiče CD. Kóduje video při přenosové rychlosti 1-1,5 Mb/s a 128-256 kb/s pro přidružené audio. Obsahuje také zvukový kompresní formát MPEG-1 Layer 3, známý jako MP3.

MPEG-2 (1995, ISO/IEC 13818) je široce využívaný standard pro distribuci signálů digitální televize (DVB-T, Digital Video Broadcasting - Terrestrial) a pro ukládání na DVD. Přenosová rychlost se pohybuje přibližně mezi 1,5-15Mb/s. Obsahuje mechanismy zabezpečení proti chybovosti a dokáže se tedy vypořádat se ztrátovým prostředím.

Se standardem MPEG-2 byl sloučen standard MPEG-3, který byl původně určen pro kódování digitální televize s vysokým rozlišením (HDTV), a tedy podporuje rozlišení 1920×1080px. Je však schopen video data generovat na několika kvalitativních úrovních (tzv. „škálování“). Standard **MPEG-4** (1999, ISO/IEC 14496) je založen na MPEG-2 a zdokonaluje jej. Přidává možnost práce

⁴ ISO/IEC Moving Pictures Experts Group <https://mpeg.chiariglione.org/>

⁵ ITU-T Video Coding Experts Group <https://www.itu.int/en/ITU-T/studygroups/2013-2016/16/Pages/video/vceg.aspx>

s trojrozměrnými objekty a různé druhy interaktivity. Mnoho z vlastností tohoto standardu je považováno za volitelné a nemusí být při jeho implementování použity.

Využití tohoto formátu je například při ukládání dat na CD a DVD ve vysokém rozlišení, videokonference, vysílání mobilní TV a televizní HD vysílání, internetová videa.

1.2.2 Kompresní standardy H.26x

Vývoj standardů H.26x probíhal pod skupinou VCEG (Video Coding Experts Group) organizace ITU (International Telecommunication Union). Původně se zaměřovaly zejména na video komunikaci v reálném čase.

Technický nástin a aplikace standardů H.26x je popsán například v [8] [9].

Standard **H.261** je využíván zejména v oblasti videokonferencí a videotelefonie v sítích ISDN (Integrated Services Digital Network). Jeho přenosová rychlost se pohybuje mezi 64kb/s a 1,92Mb/s (známý také jako Px64 - video kóduje rychlostí p-krát 64kb/s, kde $p = \langle 1,30 \rangle$). Tento standard podporuje pouze dvě rozlišení v poměru stran 4:3.

H.263 byl vyvinut po H.261 se zaměřením na zlepšení kvality při nižších přenosových rychlostech. Kóduje video v pěti rozlišeních. [9]

Jeden z nejběžněji používaných standardů současnosti, využívaných pro streaming, je standard **H.264 AVC** (v ITU/T)/MPEG-4 Part 10 (norma ISO/IEC 14496 Part 10 [10]). Nabízí vysokou kvalitu videa při zatím nejefektivnější kompresi. Je možné jej využít pro videotelefonování, VoD i streaming. Používá se i pro ukládání obsahu na disky Blu-ray. [11]

Nejnovějším standardem zmiňované skupiny je standard **H.265 HEVC** (High Efficiency Video Coding). Strukturou je velmi podobný předcházejícím standardům (MPEG-2, H.264 AVC). S rostoucí oblibou videí v HD kvalitě a formátů HD+ bylo třeba zvýšení efektivity kódování, odolnosti vůči ztrátě dat a dalších vylepšení, umožňujících mimo jiné také paralelní zpracování - to HEVC řeší. Velmi podrobně shrnuje parametry a funkčnost standardu například [12]. Stejně jako je typické i pro předchozí generace, i v HEVC dochází k takovému vylepšení, že stejnou kvalitou videa dojde k omezení datového toku až na polovinu. [11]

1.2.3 Nejpoužívanější kompresní audio a video formáty

Následující výčet několika video a audio kodeků je uveden pro základní seznámení - a to zejména s těmi, které bývají v současné době nejčastěji uváděny v odborných publikacích a aplikacích.

Video kodeky

- MPEG-2 (viz 1.2.1.)
- MPEG-4 AVC/H.264 (viz 1.2.1. a 1.2.2.) - V porovnání se standardem MPEG-2 poskytuje lepší kvalitu obrazu při stejném datovém toku (resp. nižší datový tok při stejné kvalitě obrazu). [11]
- HEVC/H.265 (viz 1.2.2)
- Windows Media Video - formát, který je v dnešní době na ústupu, jedná se však o soubor kodeků vyvinutých společností Microsoft. Původní WMV kodek byl navržen přímo pro internetové streamingové aplikace. Byl používán společně s audio formátem Windows Media Audio. Roku 2003 byl vylepšen do známější verze VC-1, která se stala velmi využívanou v oblasti Blu-ray disků a byla standardizována. [13]
- (Ogg) Theora - opensource kodek, zpočátku srovnávaný s MPEG-4 a dříve využívaný společností Wikipedia.com, v současné době již není aktivně vyvíjen. [14]
- VP8 - otevřený a zdarma použitelný kodek společnosti Google, mající podobné vlastnosti jako H.264. Pro formát WebM VP8 je využíváno audio kodeku Vorbis (viz dále).

Následující tabulka (Tabulka 1) srovnává základní parametry několika zmíněných ztrátových videokodeků.

| | Vydání | Metoda komprese | Patentováno | |
|---------------|--------|-----------------------|------------------------|---|
| H.264 | 2003 | Ztrátová/bezeztrátová | Ano | Srovnání účinnosti komprese se věnuje například [15]. |
| H.265 | 2013 | Ztrátová/bezeztrátová | Ano | |
| VP8 | 2008 | Ztrátová | Ano | |
| Theora | 2002 | Ztrátová | Ano, volně licencováno | |

Tabulka 1 Parametry základních ztrátových video kodeků

Audio kodeky

- MPEG-1 Layer 3 MP3 (viz 1.2.1) - tento standard umožňuje snížit datový tok pro hudební skladbu cca na 1/10. S mluveným slovem dosahuje horších výsledků. Je většinou využíván pro hudební soubory určené k poslechu z vlastního úložiště, avšak je možné jej používat i ve streamingových službách. Jedná se však o nejpoužívanější audio kodek vůbec. [16]
- MPEG-2 AAC - tento ztrátový kodek byl vyvinut jako „následovník“ formátu MP3. Obecně dosahuje lepší kvality zvuku za nižšího datového toku, je schopný zpracovat větší rozsah frekvencí, obsahuje větší škálu filtrů a je obecně mnohem flexibilnější v použití. Byl standardizován jako část MPEG-2 a MPEG-4. Jedná se o výchozí (nebo standardně podporovaný) audio kodek pro obrovskou škálu služeb, produktů a webových stránek (Youtube, iPhone, PlayStation 3). [16] [17]
- Vorbis - jedná se o opensource ztrátový kodek, vytvářený stejnou společností jako Ogg Theora (Xiph). [18]
- PCM - bezztrátový kodek, využívaný zejména pokud je nahraná/přenášená informace určená k pozdějšímu zpracování [16] a kvalita internetového připojení příjemce je více než dostatečná (není třeba extrémně brát v potaz možnost zhoršení jeho kvality).

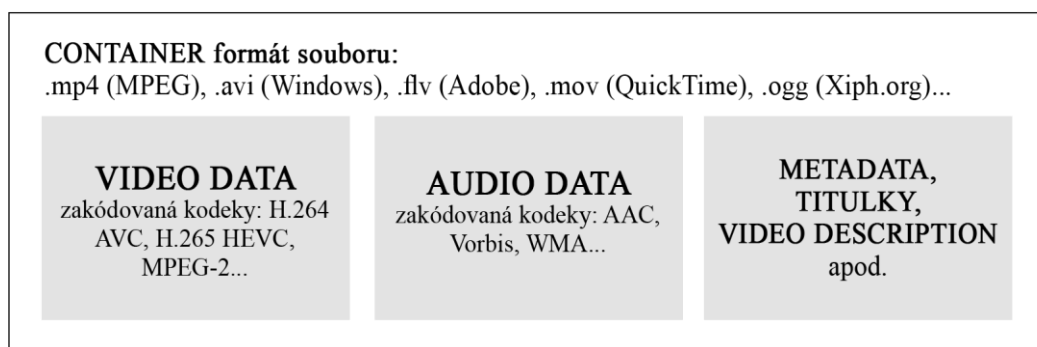
Podrobnější popis a přehled komprese standardu MPEG je dobře zpracován například v článku [17] či [19]. Tabulka 2 shrnuje základní parametry výše zmíněných audio kodeků.

| | Vzorkovací frekvence | Bitrate | Konstantní bitrate | Proměnný bitrate | Stereo |
|------------------------|----------------------|------------|--------------------|------------------|--------|
| MPEG 1 - L3 MP3 | 8-48 kHz | 8-32 kb/s | ano | ano | ano |
| MPEG 2 - AAC | 8-192 kHz | 8-529 kb/s | ano | ano | ano |
| Vorbis | 1 Hz - 200 kHz | proměnný | ano | ano | ano |

Tabulka 2 Shrnutí technických parametrů ztrátových audio kodeků

Všechna přenášená data jsou, jak bylo řečeno výše, kódována kodekem, který je zvolen vhodně s ohledem na konkrétní službu, situaci, podmínky a požadavky. Subjektivní srovnání několika užívaných videokodeků bylo publikováno v odborné literatuře, např. [20].

Pokud jsou všechna zpracovaná data definována kodekem sdružena do jednoho souboru, mluvíme o tzv. „containeru“ (v českých zdrojích také jako „kontejner“), který hraje roli při přenosu a prezentaci obsahu. Container je také označován jako formát a je reprezentován příponou souboru. Jeho strukturu nastiňuje Obrázek 2.



Obrázek 2 Struktura sdružených dat

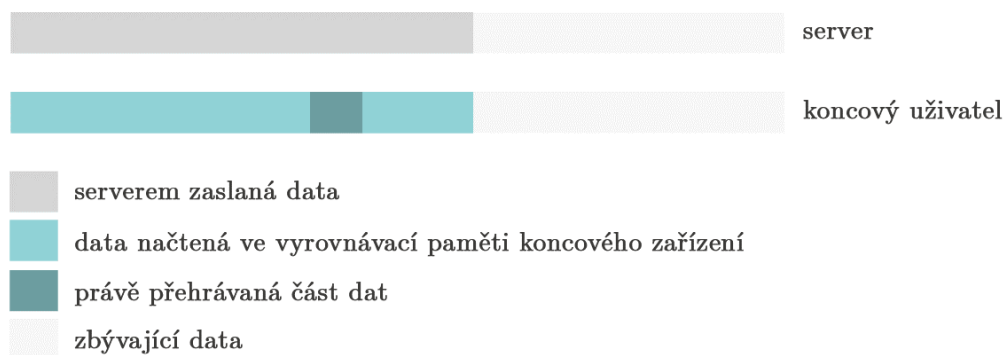
2 Způsoby doručování obsahu

Na základě způsobu doručování obsahu k příjemci je možné od sebe odlišit tři různé možnosti, jak takový stream přenášet: [21]

- progresivní download
- tradiční streaming
- adaptivní streaming

Které z nich se využívají kdy, závisí na tom, jak má být obsah přehráván a co je při jeho přenosu cílem.

2.1 Progresivní download



Obrázek 3 Progresivní download

Progresivní download (někdy označován také jako HTTP Streaming), je jednoduše stažení souboru přes webový HTTP server.

Video je stahováno do počítače a dočasně ukládáno na pevný disk. Jakmile je stažena dostatečná část videa, je možné jej začít přehrávat - ke zbylému stažení dojde v pozadí při sledování. [22]

Tento způsob downloadu je podporován širokou škálou přehrávačů a platforem. Z pohledu cílového diváka je sympatická možnost sledovat obsah před jeho kompletním stažením, avšak už není umožněno přeskokovat moc dopředu po časové ose (jelikož data ještě nebyla stažena) [21]. Klienti se špatným internetovým připojením musí čekat, než se video stáhne dostatečně na to, aby bylo plynule přehráno - může se jim tedy stát, že dojde k zaseknutí obrazu, načítání a celkovému diskomfortu při sledování.

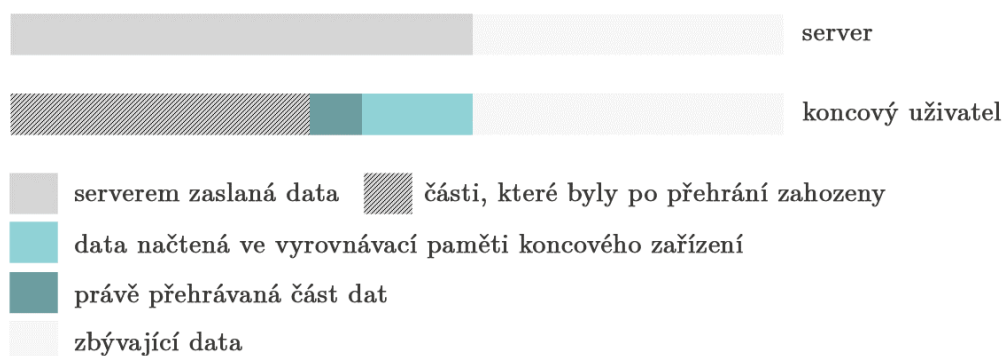
Data se stahují do doby, než dojde ke stažení celého souboru, což znamená, že pokud se divák rozhodne sledování ukončit (a například nemá o zbylý obsah zájem), může dojít ke zbytečnému využívání přenosového pásma. [23]

Pohledem autora (poskytovatele obsahu) může být nevýhodou fakt, že je soubor stahován na příjemcův disk - a je tedy mnohem snadnější jej dále šířit.

Nespornou výhodou je však fakt, že se jedná o běžné sdílení webového obsahu přes HTTP, není třeba speciálního softwaru.

Uvedený popis problematiky progresivního downloadu jednoduše nastiňuje Obrázek 3.

2.2 Tradiční streaming



Obrázek 4 Tradiční streaming

Speciálně pro streaming existují protokoly, které fungují na základě mechanismu rozdílného od progresivního downloadu - RTMP (Real Time Messaging Protocol), RTSP (Real Time Streaming Protocol). Této problematice se následující text věnuje. Doplnuje jej Obrázek 4, který názorně a jednoduše tradiční streaming představuje.

Streaming však zapojuje protokoly na několika různých vrstvách Referenčního ISO/OSI modelu. Fyzické, datové a síťové úrovně jsou brány jako samozřejmé, nicméně dále je nutné zmínit [24]:

- transportní vrstvu, zodpovědnou za přenos dat z jednoho konce na druhý,
- relační vrstvu, která organizuje streamingový přenos do probíhajících celků (takovým celkem může být film, přednáška, vysílání aj.),
- prezentační vrstvu, která provádí překlad informací přenášených po síti na ty, které jsou zobrazeny aplikací příjemci,

a

- aplikační vrstvu, která je zodpovědná za komunikaci mezi sítí a samotnou aplikací.

Většina komunikace probíhá pomocí TCP transportního protokolu, který je sestaven tak, aby poskytl spolehlivý přenos souboru - kontroluje, zda byly přijaty všechny datové pakety a v případě, že ne, snaží se ty ztracené dohledat [25]. Streaming, o kterém se však všeobecně mluví, klade větší důraz na jakousi časovou spolehlivost, a tak je často využíváno síťového protokolu UDP (User Datagram Protocol), který v případě ztráty paketu posílá další a nezdržuje přenos jakýmkoliv dohledáváním či přeposíláním.

Transportní protokol RTP (Real Time Protocol) je pro přenosy v reálném čase také využíván a ačkoliv to není běžné, je možné jej využívat společně s TCP. [24]

Protokol RTP je úzce spojen s protokolem RTCP (Real Time Control Protocol), který operuje na relační vrstvě OSI. Poskytuje informaci o kvalitě datového přenosu a umožňuje akce přizpůsobení datového toku. [24]

Spolu s RTP jsou využívány také jiné protokoly - jako například RTSP (Real Time Streaming Protocol), protokol prezentační vrstvy, umožňující vzdálenou kontrolu obsahu na síti - přenáší požadavky na činnosti, jako je přehrání, pozastavení, nahrávání.

Skupina protokolů RTP, RTSP a RTCP je občas označována pod souhrnným „RTSP“.

RTP relace může obsahovat několik streamů - například audio a video můžou být přenášeny na rozdílných kanálech a sloučeny na konci, na přijímači. ⁶

RTMP je primárně využíván v aplikaci Flash (ale i v jiných softwarových vybaveních) a je schopen pracovat se všemi multimediálními formáty. [24]

Multimediální obsah, přenášený pomocí RTMP, je na rozdíl od progresivního downloadu streamován po částech a vyžaduje přítomnost streamingového serveru v komunikačním řetězci. Ten totiž posílá soubor přes internet až k cílovému uživateli. Pokud bylo u progresivního downloadu nevýhodou stažení celého souboru na disk, zde je viditelný obrovský rozdíl. V tomto způsobu přenosu obsah nikdy

⁶ URL adresy UDP nejsou prohlížeči podporovány a využívá se plug-inů pro streamování RTP/UDP na PC - nejčastěji používaný je pravděpodobně Flash. Ani mobilní telefony nejsou kompatibilní s RTP protokoly, avšak je možná instalace aplikací třetích stran, které toto umožňují. [24]

není celý skladován na disku uživatele. [23] Obsah je po částech zaslán tak, jak si jej přehrávač vyžádá, a jakmile je část přehrána do konce, je zahozena.

Je také možné přeskakovat na libovolnou část obsahu - přehrávač jednoduše vyžádá zaslání dané části.

Streamované video je také distribuováno pouze v jednom datovém toku. Pokud přehrávače nabízejí možnost výběru kvality videa, předpokládá se, že server může soubor s daným datovým tokem poskytnout.

2.3 Adaptivní streaming

Další variantou přenosu obsahu je logicky kombinace obojího již zmíněného. Takovou metodou je tzv. adaptivní streaming. Obsah je sledován průběžně, ale postupně (po částech) stahován (progresivní download) na bázi protokolu HTTP. V praxi je tak možné přenášený multimediální obsah přizpůsobit aktuální kvalitě připojení příjemce a zatížení CPU zařízení, které zrovna pro sledování využívá. [23]

Video je na zdrojové straně po segmentech nahráváno na HTTP server a kódováno pro širokou škálu rozlišení (např. od HD videa v případě kvalitního a silného připojení, až po velmi nekvalitní a nečitelné video v případě slabého signálu na přijímači). Klientův videopřehrávač monitoruje stav kvality připojení a výpočetní možnosti pro určení takového (nejlepšího) datového toku, který je jeho přístroj schopen přijmout a bez problémů přehrát. Poté zašle serveru požadavek na stažení části ve správné kvalitě. Způsob a četnost takového požadavku se liší v závislosti na přehrávači, protokolu, stavu sítě atd. ale obecně můžeme říci, že k obnovení požadavku dochází každých cca 5 vteřin. [23]

Jelikož jsou jednotlivé segmenty kódovány nezávisle na sobě, bez mezer či překryvů, pro uživatele je výsledný sledovaný obraz k nerozeznání od jednoho celku a je bez přestávek a nepříjemného narušování přehráván v nejvyšší kvalitě, která je zrovna možná.

Tento způsob streamování se stává standardem pro většinu platforem a společností, jako jsou Microsoft (SS - Smooth Streaming), Apple (HLS - HTTP Live Streaming), nebo Adobe (HDS - HTTP Dynamic Streaming).

Benefitů pro adaptivní streaming se dá uvést několik:

- na rozdíl od přenosu na protokolu RTP nemá HTTP přenos problémy při překračování bran firewallu,

- reakce na zvýšení a snížení propustnosti sítě dynamickým přepínáním kvality obsahu,
- ke spuštění obsahu dochází relativně rychle, neboť zpočátku je možné přehrávat za nižší kvality (a následně ji rapidně zlepšit),
- do vyrovnávací paměti zařízení se nestahují žádná data,
- přenosová rychlost se přepíná dle vytíženosti CPU koncového zařízení a aktuální propustnosti sítě,
- tedy veškerá adaptace přenosu je na straně příjemce.

2.4 MPEG DASH

Každá komerční platforma, využívající streamování obsahu online, je jakýmsi uzavřeným systémem, využívající své formáty, protokoly atd. - jinými slovy - všechny výše zmiňované metody adaptivního streamingu (HLS, SS, HDS) jsou si sice velice podobné, ale postrádají úplnou kompatibilitu mezi sebou. Pro každý z nich je třeba odlišného přehrávače a nejednotný způsob distribuce na různá klientská zařízení je vzhledem k současnému dosahu streamování neatraktivní.

Úsilí organizace MPEG o vytvoření standardu pro doručování multimediálního obsahu (adaptivního streamingu či VoD), který by řešil problémy s nekompatibilitou, započalo v dubnu roku 2009. V následujících dvou letech MPEG spolupracovala zejména se standardizační telekomunikační skupinou 3GPP (Third Generation Partnership Project) a mimo jiné také s komerčními společnostmi (Adobe, Microsoft, Netflix) a dalšími průmyslovými organizacemi (jako například OIPF, W3C apod.) a výsledný standard, MPEG DASH, celým názvem MPEG Dynamic Adaptive Streaming over HTTP, byl ratifikován v listopadu 2011.

V dubnu následujícího roku byl publikován pod normou ISO/IEC 23009-1 [26] a byl označen za průlomový díky potenciálu pracovat se stávajícími technologiemi, nahradit je a spustit je na libovolném zařízení [27]. Přináší také obrovskou výhodu - je navržen tak, aby 100% využíval možností adaptivního streamingu, avšak byl schopen pracovat s audio/video obsahem jakéhokoliv kódování [28].

2.4.1 Princip MPEG DASH

Obrázek 5 nastiňuje jednoduchý scénář streamingu mezi serverem a příjemcem.

Doručování obsahu probíhá velmi obdobně jako v dříve zmiňovaných protokolech adaptivního streamingu - po malých částech, které jsou na straně příjemce „složeny“ a sledovány postupně jako celek. Jedná se o novou „odnož“ adaptivního streamování, tedy zachovává také přizpůsobování kvality obsahu příjemci.

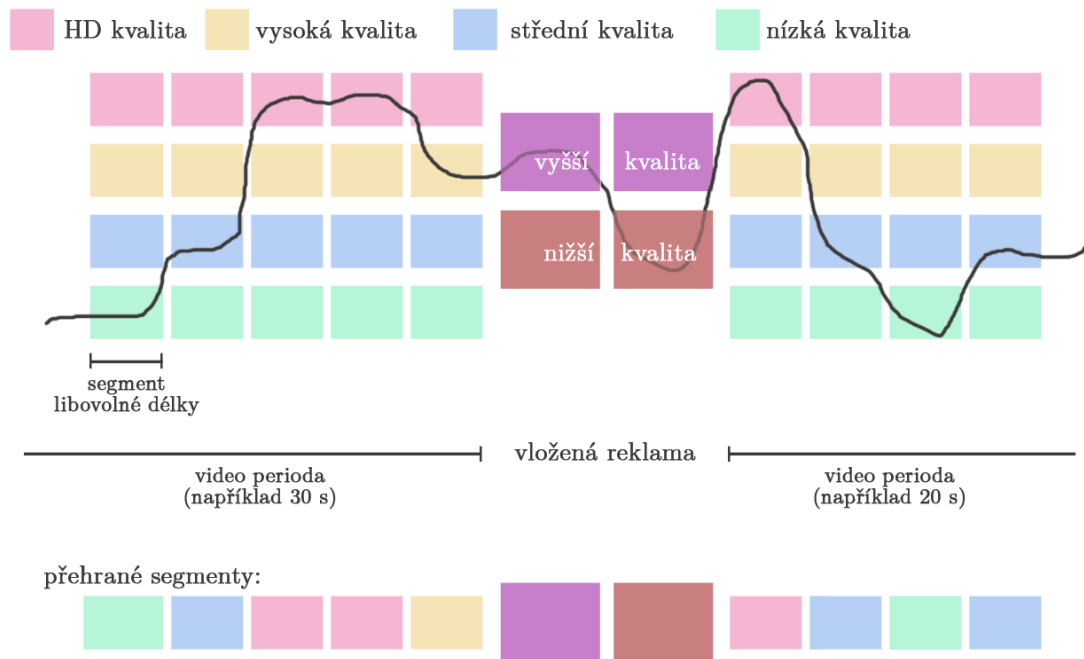
MPEG DASH využívá pro přenos informací stávající infrastrukturu HTTP, která se používá k doručování prakticky veškerého webového obsahu [29].

Obsah existuje na serveru v podobě dvou částí - Media Presentation Description (MPD, 2.4.2), který popisuje dostupný obsah a charakteristiky přenášeného média; a segmentů, které nesou konkrétní multimediální obsah ve formě jednoho nebo několika souborů.

Pro přehrání obsahu DASH klient nejprve obdrží MPD soubor. Jeho analýzou získá informace o časové charakteristice, typech médií, rozlišení, minimální a maximální šířce pásma, existenci kvalitativních alternativ obsahu, lokalizuje jednotlivé obsahové segmenty na serveru aj. Pro živé vysílání uvádí také čas začátku a konce pro přenos segmentů a začátku a konce samotného multimediálního obsahu, stejně jako zda se počítá s fixní či proměnnou délkou trvání jednoho segmentu [30].

Pomocí těchto informací DASH klient zvolí nejvhodnější kvalitativní alternativu a spustí streamování obsahu pomocí HTTP příkazů GET. Po nahrání dostatečného množství dat se spustí přehrávání a klient v průběhu přehrávání na pozadí načítá následující segmenty a monitoruje kvalitu připojení. Dle výkyvů v jeho kvalitě rozhoduje o tom, zda následně načítaný segment dat zůstane ve stejné kvalitě, nebo bude změněn - a to tak, aby přehrání pokračovalo plynule dál, byť došlo ke změně kvality obsahu.

MPEG DASH jako takový se však zabývá pouze samotným MPD a formátováním obsahu - stahování vlastního obsahu po přečtení MPD, rozhodování o dalším fungování přehrávače a následující úkony jsou zcela na klientské straně.



Obrázek 5 Adaptivní streamování - MPEG DASH

2.4.2 Media Presentation Description

Jak bylo řečeno v 2.4.1., pro dynamické streamování pomocí HTTP je nutné mít na serveru k dispozici alternativy obsahu v různé kvalitě (s různým bitratem). Multimediální obsah může být také složen z několika částí (například video, audio, text - každá z těchto částí je zdrojově kódována zvlášť) majících různé vlastnosti. Všechny tyto vlastnosti jsou ve standardu MPEG DASH popsány v souboru MPD. MPD je dokumentem typu XML. Je hierarchickým modelem přenášeného obsahu. Obsahuje jednu či několik period, charakterizovaných počátečním časem a délkou trvání. V jedné periodě je jeden či více „adaptačních částí“ (dále jen AS - „adaptation set“ z angl.). AS poskytuje informace o jednotlivých částech multimediálního obsahu a jeho kódovaných alternativách. Příkladem se dá uvést následující: jeden AS nese několik kvalitativních verzí videa, další několik kvalitativních verzí audio části - v každém AS je několik reprezentací totožného obsahu.

Každá reprezentace obsahu se od druhé liší - jedná se o kódované alternativy, liší se v bitratu, rozlišení, použitém kodeku, počtu kanálů, v případě audia třeba také v jazyce apod.

Jednotlivé reprezentace v AS se poté skládají z časově posloupných segmentů (aby bylo možné přeskakovat mezi reprezentacemi během přehrávání), majících vlastní adresu na serveru a mohou tak být stahovány pomocí HTTP GET. Každý segment má svou dobu trvání, čas počátku, unikátní URL a identifikační index.⁷

2.4.3 Dodatečné funkce

Standard MPEG DASH je velmi obsáhlý a nabízí tedy nové funkce, jako například [30]:

- vložení reklam - mezi segmenty s vlastním obsahem je možné vložit segmenty reklamní;
- jednotlivé segmenty můžou trvat různě dlouho;
- různé základní URL adresy - totožný obsah může být dostupný na několika URL adresách (serverech) - a klient tedy může streamovat z jakéhokoli z nich, aby maximálně využil přenosového pásma;
- MPD může být rozdělen do částí a stahován zvlášť, může odkazovat na externí části;
- klient může serveru poskytnout zpětnou vazbu ohledně kvality přenosu - standard přesně nastavuje hodnotící parametry;
- přepínání mezi jednotlivými streamy - MPD umožňuje klientovi volit/přepínat mezi jednotlivými datovými proudy - tedy například vybrat jednu audio stopu z několika jazykových mutací, jednu videostopu (například lišící se v úhlu kamery při snímání), nebo vybrat textovou stopu - komentáře, titulky, a podobně - a dynamicky mezi nimi přepínat dle kvality.

⁷ Standard MPEG DASH využívá segmentových formátů využívaných v HDS, SS (ISO BMFF) i v HLS (MPEG-2 TS), a tak umožňuje nepřiliš komplikovaný přechod na MPEG DASH - segmenty jako takové mohou zůstat ve většině případů stejné, jen je nutné je převést pod nový soubor MPD.

2.5 Srovnání řešení pro adaptivní streaming

V následující tabulce (Tabulka 3) je možné sledovat jednoduché srovnání různých řešení pro adaptivní streaming, již zmíněných výše.

| | SS (2.3) | HLS (2.3) | HDS (2.3) | MPEG DASH (2.4) |
|------------------------|-------------|---------------|------------|--|
| vlastník | Microsoft | Apple | Adobe | žádný, standard |
| video kodek (1.2.3) | H.264, VC-1 | H.264 | H.264, VP6 | libovolný |
| audio kodek (1.2.3.) | AAC, WMA | AAC, MP3, AC3 | MP3, AAC | libovolný |
| typická délka segmentu | 2 sekundy | 10 sekund | 2-5 sekund | 2-10 sekund (nejčastěji 1, 2, 4, 6 10, 15) |

Tabulka 3 Srovnání řešení pro adaptivní streaming

3 Kvalita streamu

Nejen v případě komerčního streamování je kladen důraz na kvalitu výsledného videa a tedy na kvalitu zážitku příjemce. V případě sledování VoD jsou data stahována a přehrávána následně - až po stažení a nahrání dostatečného množství dat. V ideálním případě tak recipient sleduje naprosto nerušenou kopii videa. V reálném scénáři však dochází k narušení komunikačního řetězce externími vlivy, fluktuaci kvality komunikačního kanálu.

V telekomunikačních sítích je QoS (Quality of Service, „kvalita služeb“) vyjadřována objektivně síťovými parametry jako je zpoždění, jitter, propustnost sítě, ztrátovost paketů aj. Monitoring a případné ovlivňování parametrů QoS jsou důležité pro kvalitu videa jako takového, kvalitu finálního zážitku však nijak nezaručují. Mnohem důležitější je sledovat a vyhodnotit kvalitu zážitku z pohledu koncového uživatele - QoE (Quality of Experience). QoE je tedy subjektivní vnímání kvality celkové služby, nejedná se o hodnocení na základě předem daných parametrů. Zákazník, jeho spokojenost, naplnění očekávání a jeho dojmy jsou rozhodujícím měřítkem - výsledky hodnocení se tedy mohou pokaždé lišit, neb každý z uživatelů je jiný a je ovlivňován rozdílnými faktory - technickými i netechnickými.

3.1 Uživatelské faktory ovlivňující QoE

Nejen technické okolnosti mají vliv na výsledné vnímání přijímané služby. Právě lidský faktor je, můžeme říci, největší proměnnou celého hodnotícího řetězce.

3.1.1 Uživatel a prostředí

Ústředním zmíněným lidským faktorem je vždy uživatel. Výsledné hodnocení je ovlivněno jen jím - jeho očekáváním, dojmy, zkušenosti se službou, ale také tím, jak se zrovna po psychické a fyzické stránce cítí (a mnoho dalšího). Záleží také na tom, v jakém prostředí se nachází. Případná hlučnost, nedostatečné osvětlení, nepříjemná teplota - vše může ovlivnit jeho vnímání.

Technické vybavení uživatele je také důležitým faktorem - ať už se jedná o faktickou kvalitu jeho zařízení (stáří, výkonnost komponent, atp.) či požadavky

a očekávání, které sám bere v potaz - rozlišení displeje u mobilního telefonu, velikost zařízení, pestrost, barevný rozsah a jiné.

3.1.2 Obsah

Obsah, který je koncovým uživatelem považován za atraktivní (a to ať si jej vyhledal sám, či je na něj cílen a vybrán) bude velmi pravděpodobně hodnocen pozitivněji, než v případě, kdy hodnocený obsah za zajímavý nepovažuje (a nemá tedy explicitní důvod na něj pohlížet v pozitivním světle).

3.2 Faktory související s vnímáním a technickými vlivy

V případě streamingu multimediálního obsahu (např. videa) je nejdůležitějším očekáváním to, že by divák měl video sledovat bez přerušení (tzv. „rebufferingu“ - zastavení přehrávání pro načtení následující sekvence) a bez degradace kvality způsobené procesem doručování (jako může být ztráta snímků, problémy při rekonstrukci dekodovaného obrazu aj.).

Zejména v případě adaptivního streamování je však mnohem více parametrů, které mohou přehrávaný výstup ovlivňovat.

3.2.1 Počáteční zpoždění

Počáteční zpoždění (Initial Delay) je ve službách multimediálního streamingu přítomno vždy a představuje dobu čekání před začátkem přehrávání videa. Dochází k němu, protože před začátkem samotného dekodování a přehrávání je nutné přenést určité množství dat. Reálná hodnota počátečního zpoždění je zohledňována vzhledem k rychlosti datového připojení a nastavení přehrávače, běžně však bývá počátek přehrávání pozdržen o něco déle, než je technicky nutné a do vyrovnávací paměti⁸ je načtena větší část videa. Jedná se o efektivní způsob jak se vyrovnat s výkyvy kvality připojení během přehrávání. [31]

⁸ Pro plynulé přehrávání mají systémy pro streaming určitou vyrovnávací paměť, tzv. „buffer“ (přesahující 20 ms), která - dalo by se říci - tlumí nežádoucí efekty, způsobené nekvalitním či nekonzistentním přenosem dat. Přizpůsobuje se změnám ve zpoždění či chybách datových paketů a umožňuje plynulé přehrávání.

Je však nutné toto počáteční zpoždění volit rozumně, čím více dat do vyrovnávací paměti načteme, tím delší zpoždění bude a mohlo by mít negativní vliv na QoE. Z [32] je pak patrné, že počáteční zpoždění několik sekund ovlivní vnímání uživatele jen do malé míry.

Obecně se dá říci, že krátké počáteční zpoždění bude výhodnější pro uživatele, kteří mají zájem pouze o přehrání malé části videa, avšak v případě, že uživatel má zájem o zhlédnutí celého videa, bude tolerovat delší (třeba i několikasekundové) zdržení.

3.2.2 Pozastavení

K pozastavení videa během přehrávání (v cizojazyčné literatuře označováno jako „stalling“ či „rebuffering“) dochází v případě, kdy je propustnost aplikace pro streaming videa nižší než bitrate videa. V průběhu přehrávání dojde k vyčerpání dat nahraných ve vyrovnávací paměti a je nutné čekat, dokud není opět načten dostatečný objem informací.

Celková délka čekání během přehrávání videa je ovlivněna nejen délkou rebufferingu, ale také frekvencí, s jakou se pozastavování během přehrávání objevuje.

Výsledky testů prezentované v [33] jasně poukazují na to, že uživatelská spokojenost s kvalitou se snižuje s počtem nastalých pozastavení videa a zároveň je jimi upřednostňováno jedno delší pozastavení než několik kratších.

Tradiční streaming není schopen na změny v síti reagovat pohotově, u adaptivního streamingu, který se aktuálnímu stavu připojení stále přizpůsobuje, je proto zmírnění pozastavování viditelné.

3.2.3 Zpoždění a jitter, ztrátovost paketů a propustnost

Zpoždění (či jinými slovy také latence) je doba, po kterou se informace dostává od odesílatele k příjemci. V malých lokálních sítích (LAN) bývá toto zpoždění zpravidla velmi malé, ale v případě internetu může být zpoždění tak dlouhé, že není vyloučeno znatelné zpoždění při sledování obsahu. To je znatelné zejména při živém vysílání. Delší zpoždění může způsobit také zhoršení kvality obrazu/zvuku [34].

Množství dat za sekundu, představujících multimediální signál, se mění v každém okamžiku v závislosti na konkrétním obsahu a použitém kodeku. Přenášená data jsou posílána na server a potom dále (přes další servery, nebo přímo) k uživateli

a během tohoto přenosu mohou datové pakety dorazit neúplně, v nesprávném pořadí, nebo se ztratit úplně. **Jitter** je označení pro měřítko variability zpoždění dat - jak konzistentní toto zpoždění je (tedy možno označit za „kolísání zpoždění“). Pro hlasový a video streaming je nutný „low jitter“ (tedy případ, kdy mají všechny datové pakety obdobné zpoždění), jinak dochází k „roztrhanému“ audiovizuálnímu zážitku.

Ztrátovost paketů (Packet loss) je průměrný počet ztracených paketů při přenosu za určité období vzhledem k celkovému počtu přenesených paketů, vyjádřený v procentech.

Výchozím parametrem pro nabídku poskytovaných služeb je **propustnost** (bandwidth), tedy množství informace, které je kanál schopen přenést za jednotku času.

3.2.4 Přizpůsobení kvality

Faktor přizpůsobení kvality zvýrazňuje výhodu adaptivního streamování, který přizpůsobí kvalitu přehrávání aktuálnímu stavu sítě a dle [35] může snížit pozastavení videa až o 80%.

Ze závěru studie [36] lze poznamenat, že je přípustnější ovlivnit vědomě kvalitu přehrávaného videa, než se potýkat s nevědomými degradacemi kvality přenosu. Příjemce tedy raději přehraje video v horší kvalitě (v nižším rozlišení), než aby se potýkal se ztrátou paketů, pozastavením, rebufferingem (3.2.2.), což lze ovlivnit zvolením nižšího rozlišení přehrávání i při tradičním streamingu.

3.2.5 Technologické parametry adaptivního streamingu

Fungování adaptivního streamingu a jednotlivých odlišností bylo již v této práci vysvětleno, základní srovnání je k naleznutí v Tabulce 3. Ač se způsoby jak popsat přenášená data při přenosu liší (MPD soubor u MPEG DASH se liší od souborů užívaných při MSS, HLS i HDS), parametry mající vysokou důležitost při ovlivnění QoE jsou pro ně společné. [31]

- Délka segmentu - v adaptivním streamingu představuje délka segmentu nejkratší možnou dobu, za kterou může dojít k přizpůsobení kvality (bitratu) videa. Některé systémy pracují s fixní délkou segmentu, jiné

s proměnlivou. Volba parametrů segmentů bude stručně analyzována dále v této práci.

- Kodek - MPEG DASH umožňuje práci s jakýmkoliv kodekem, jak již bylo zmíněno, jiné způsoby řešení adaptivního streamingu tuto možnost nemají. Existují však systémy, které podporují pouze specifická kódování - v současné době dominuje video kodekům H.264, volba audio kodeku nebývá v systémech tak přísně definována. [31]
- Formát - v dnešní době je nejčastěji používaným formátem pro zapouzdření MP4 (využíván v MPEG DASH), který různé typy dat přenáší odděleně a umožňuje flexibilně s nimi manipulovat. Jeho kompletní představení a uvedení vlastností je možné nalézt v doporučení [37].

Je využíváno také M2TS (MPEG-2), který přenáší data v paketech pevné délky (184 bytů délka + 4 byty záhlaví) a každý paket obsahuje pouze jeden druh obsahu. [31]

3.2.6 Server

Na straně serveru, který multimediální obsah poskytuje, je důležitá správná příprava obsahu, tedy správný výběr kvalitativních reprezentací, použitého kodeku a systémových parametrů, které budou při přípravě videa pro přenos použity. Mezi ty se řadí například délka jednoho segmentu videa. Ten musí být dostatečně krátký na to, aby bylo při přizpůsobování možné rychle reagovat na změnu kvality připojení, ale také dostatečně dlouhý na to, aby využil účinnost zdrojového kodéru a udržel požadované režijní náklady na skladování dat na serveru.

3.2.7 Klient

Na klientské straně dochází k rozhodování kdy začít se stahováním dat (a jaké segmenty zvolit) a jakým způsobem spravovat vyrovnávací paměť přehrávače. Rozhodovací stroj by měl využívat takového adaptačního algoritmu, který volí reprezentace k maximálnímu QoE. Pravděpodobně nejběžnějším přístupem pro rozhodování je využití odhadu aktuální šířky pásma jako rozhodovacího kritéria. Strategii pro rozhodování je však několik a jsou nastíněny například v těchto publikacích: [38] [39].

3.3 Artefakty

Při sledování multimediálního obsahu je možné pozorovat nechtěné prvky v obraze, artefakty, které mohly vzniknout při zpracování a úpravě surových dat (například při kompresi), nebo při přenosu. Při streamování multimediálního obsahu jsou artefakty vzniklé v obrazové složce mnohem dominantnější než ty ve složce zvukové. Neznamena to však, že audio artefakty nejsou slyšitelné - kromě časové prodlevy a pozastavování z důvodu načítání (což se týká i obrazové složky a bylo zmíněno v 3.2) se objevují také kompresní artefakty (jmenovitě například pre-echo, post-echo, zrnitost, ztráta nevýrazných zvuků a mnoho dalších [40] [41]).

Z vlastních zkušeností je však většina diváků náchylnější na problematické prvky v obraze. Při digitalizaci obrazového signálu může dojít k následujícím artefaktům:

- aliasing, překrývání frekvenčních spekter - nedodržení Nyquistovy vzorkovací podmínky [42]
 - časový aliasing (blikání na obvodu pohybujících se objektů, interference cyklického pohybu se snímkovou frekvencí - velmi známým a uváděným příkladem ze starších záběrů jsou kola dostavníků, točící se nesprávným směrem či zdánlivě stojící)
 - prostorový aliasing (zubaté zobrazení šikmých linií, interference obrazu s pixelovým rastrem)
- kvantizační šum a saturace (bledost obrazu, obtékání černou barvou), související s počtem kvantizačních úrovní [43]

Při přenosu datových paketů (viz 3.2.3) může dojít ke ztrátě části obrazu - například chybějící I snímek způsobí „splýnutí“ dvou scén do jednoho obrazu (velmi názorně viditelné například ve videu [44]).

Artefakty způsobené kompresí se v obrazové složce vyskytují samozřejmě také:

- tzv. „čtverečkování“, blokový efekt - v kodeku H.264 snaha o jeho potlačení deblokovacím filtrem
- „haló“, nebo též „ringing“ jsou zkreslení (rozmazání, okolní opar) nebo malé tečky kolem pohybujících se objektů a hran s rychlým přechodem na pozadí (vysoký kontrast) - tomuto typu obrazový typ se říká Gibbsův efekt, moskyti nebo moskytí křídla (z angl. „mosquito wings“) [43] [45]

Zmiňované artefakty a obrazové chyby zdaleka nejsou kompletním výčtem. Více z nich uvádí a přibližuje například [45].

4 Hodnocení kvality

Kvalita videa, jakožto charakteristika videa přenášeného a zpracovávaného systémem, je definována jako formální nebo neformální míra vnímané degradace obrazu (obvykle ve srovnání s videem původním). [46] [47]

Metody pro vyhodnocení kvality obrazu je v zásadě možné rozdělit do dvou skupin (které jsou dále v textu vizualizovány také na Obrázku 6):

- Subjektivní metody, ve kterých pozorovatelé hodnotí systém v subjektivním měřítku. Tyto metody jsou relevantnější, neboť kvalitu hodnotí skutečný pozorovatel. Jsou však náročnější na čas a zdroje, což je činí relativně nevhodnými pro praktické aplikace.
- Objektivní metody, které probíhají automaticky bez účasti pozorovatelů.

Vztah mezi objektivním a subjektivním hodnocením je velmi složitý a zatím neexistuje objektivní parametr, který by ve velké míře koreloval se subjektivním hodnocením.

4.1 Subjektivní metody hodnocení kvality videa

Metody subjektivního hodnocení jsou statistické metody založené na porovnání původního a přijatého videosignálu skupinou diváků, kteří podle předdefinované stupnice hodnotí kvalitu a srozumitelnost informace. Jak vykonávat různé typy subjektivních testů specifikují mezinárodní doporučení ITU-R BT.500-13 [48] (z roku 2012) či ITU-T P.910 [49] (z roku 2008, popisující neinteraktivní subjektivní metody pro vyhodnocení video kvality).

4.1.1 Doporučení ITU-R BT.500-13

Podle tohoto doporučení existují dvě třídy subjektivního hodnocení kvality, a to:

- Hodnocení kvality - hodnocení, které charakterizují systém za optimálních podmínek;
- Hodnocení poruch - hodnocení schopnosti systémů zachovat kvalitu při neoptimálních podmínkách, souvisejících s přenosem (vysíláním).

Zároveň podrobně popisuje dvě hlavní doporučené metody - DSCQS a DSIS, stejně jako alternativní metody hodnocení.

V dokumentu je také uvedeno přesné nastavení laboratorního (měřicího) prostředí a podmínky pro průběh subjektivního testování (například maximální pozorovací úhel, špičkový jas, okolní osvětlení, a další).

DSIS

Double Stimulus Impairment Scale (DSIS) je metoda užitečná pro posouzení vlivu zkresleného obrazu. Zadavatel disponuje originální nezkreslenou verzí v plné kvalitě. Předpokládá se tedy, že hodnocená videosekvence bude v horší či stejné kvalitě než známá reference. Pozorovateli je vzhled referenčního obrazu znám a na pětistupňové hodnotící škále je dotazován na zkreslený obraz (stupnice v rozmezí od „nepostřehnutelná“ až „nepříjemná“ chyba). Detailnější popis je dostupný v [48].

DSCQS

Double Stimulus Continuous Quality Scale (DSCQS) pracuje na podobném principu jako DSIS, ale je relativní. Je ideální pro použití v případě, kdy nejsme schopni pozorovateli poskytnout původní sekvenci v nezkreslené kvalitě. Pozorovatel nezná referenční sekvenci, a tak hodnotí kvalitu obrazů na spojitě stupnici. Více detailů v [48].

SS

Single Stimulus (SS) není založen na porovnávání obrazu s referencí. Více simuluje domácí podmínky, avšak cyklus testování je obdobný jako u DSIS. Metoda má dvě varianty lišící se v počtu opakování sekvencí (a tedy časovou náročností).

SC

V metodě Stimulus Comparison (SC) pozorovatel porovnává dvě videosekvence, kdy ani jedna není původní. Jsou porovnávány všechny videosekvence se všemi, je tedy časově velmi náročná. Ve třech druzích hodnocení pozorovatel zařadí sekvenci do kvalitativní třídy, zhodnotí rozdíl kvality obrazů spojitě a poté také kvalitu každého videa zvlášť.

SCQE

Metoda Single Stimulus Continuous Quality Evaluation (SCQE) je opět hodnocením bez reference. Kvalita sekvence se kontinuálně mění dle průběžné úpravy sledovaných parametrů. Pozorovatel hodnotí kvalitu průběžně.

SDSCE

S malým rozdílem byla z metody SCQE vyvinuta Simultaneous Double Stimulus Continuous Evaluation (SDSCE). Je navrhnutá na porovnání věrnosti obrazu mezi dvěma sekvencemi v reálném čase, zmiňovaný rozdíl je tedy ve způsobu prezentování obrazu divákovi (a stupnici hodnocení).

4.1.2 Doporučení ITU-T P.910

Neinteraktivní subjektivní metody popsané v tomto doporučení mohou být použity pro různé účely - například výběr algoritmů, ohodnocení účinnosti komprese systému, hodnocení úrovně kvality spojení. Popisuje také vlastnosti použitých zdrojových sekvencí, jako je typ obsahu, délka a počet sekvencí atp.

Metody ACR (Absolute Category Rating), DCR (Degradation Category Rating) a PC (Pair Comparison) považují za velmi podobné výše zmíněným, jsou podrobně popsány v [49] a [50] - první zmíněná publikace také jednotlivé metody srovnává a určuje jejich vhodné aplikace.

ACR je metoda jednou z nejvíce používaných metod pro posuzování kvality telekomunikačních služeb. DCR je také známa jako DSIS a je tedy vhodná na hodnocení systémů, kde je degradace posuzovaná s vyšší citlivostí. Sekvence jsou prezentované v párech a porovnává se míra znehodnocení videa v porovnání s referenčním. PC již podle názvu jasně udává, že sekvence se testují v párech - umožňuje porovnání mezi videi a tedy je to metoda schopná rozeznat jemné kvalitativní rozdíly.

4.2 Objektívni metody hodnocení kvality videa

Automatické objektívni metody hodnocení kvality videa probíhají bez účasti pozorovatele, jedná se o algoritmické určení kvality obrazu (videa). Publikace [46] shrnuje objektívni metody pro hodnocení kvality pod pojem metriky (ve smyslu „metody, míra“) a definuje je jako algoritmy, použité na definování a modelování

obrazu, jejichž výsledkem je číselná hodnota, na jejímž základě je možné chápat míru poškození obrazu videa. Jejich snahou je pro příslušnou aplikaci najít metodu s největší korelací s výsledky subjektivního hodnocení. [51]

Pro objektivní hodnocení kvality se uvádějí tři dostupné přístupy, metody, klasifikované podle dostupnosti originálního videa:

- Metody úplné reference (FR - Full-Reference) přistupují k původní verzi (nezkreslené například přenosem či kompresí) a nejčastěji fungují tak, že porovnávají jednotlivé pixely původního a zkresleného obrazu.
- Metody snížené reference (RR - Reduced-Reference), porovnávající a využívající vlastností obou verzí videa. Používají se v případě, kdy není část informací původního videa k dispozici.
- Metody bez reference (NR - No-Reference), mající přístup pouze ke zkreslenému videu a snaží se tedy hodnotit jeho kvalitu bez jakéhokoliv odkazu na jeho referenční obraz. Obvykle se používají v případech, ve kterých je známa metoda kódování videa.

Stejně jako pro subjektivní metody hodnocení, jsou i pro objektivní metody vydány mezinárodní doporučení, které je popisují a blíže specifikují. Podle charakteru obrazu (videa) a aktuální potřeby jsou neustále vyvíjené a vylepšované.

V doporučeních je možné nalézt mj. metriky MSE, PSNR, VQM, SSIM [52], které budou stručně vysvětleny dále.

| Hodnocení kvality videa | | | | |
|-------------------------|-------------------|--|--|---------------------------------|
| subjektivní | | objektivní | | |
| hodnocení kvality | hodnocení poruch | matematické metriky (např. PSNR, MSE) | metriky založené na vlastnostech lidského vizuálního systému | ostatní metriky (např. SSIM) |
| „Double Stimulus” | „Single Stimulus” | | | |

Obrázek 6 Rozdělení hodnocení kvality videa

4.2.1 MSE

Rozdílové metriky, porovnávající jednotlivé pixely originálního a testovaného videa jsou rychlé na vyhodnocení a co do výpočtu jednoduché. Jejich nevýhodou je fakt, že nepřibližují dostatečně přesně vnímání zkreslení a kvality lidským vizuálním

systemem. Výpočet MSE je jednou z nejoblíbenějších metod tohoto typu. Jeho využití však není striktně pouze jako základ pro metriky kvality videa. [53] [54] [55] Originální nezkraslený obraz je možné vnímat jako součet zkrasleného obrazu a signálové chyby. Předpokládá se, že ztracená kvalita přímo souvisí s viditelností signálové chyby. Velikost signálové chyby objektivně kvantifikuje střední kvadratická chyba, MSE (Mean Square Error). Vyjadřuje průměrný kvadratický rozdíl úrovní jasových hodnot pixelů mezi dvěma sekvencemi (obrazy).

Dva zkraslené obrazy se stejným MSE však mohou mít odlišné chyby - méně či více viditelné. [56]

Dle [57] se vypočítá následovně:

$$\text{MSE} = \frac{1}{T \times M \times N} \sum_t \sum_m \sum_n [X_0(m,n,t) - X_r(m,n,t)]^2 \quad [-] \quad (1)$$

Kde X_0 je vstupní testovaná sekvence a X_r referenční sekvence o T snímcích s rozměry $M \times N$.

Odmocninou z MSE je pak definována také efektivní hodnota střední kvadratické chyby RMSE, známá také jako směrodatná odchylka. Ta je však ve svém použití značně limitována, neb je založena na předpokladu, že je člověk náchylný pouze na vnímání odchylek mezi sekvencemi, ne však na barevné odchylky a další aspekty (jako je například časová a prostorová frekvence). [56]

4.2.2 PSNR

Další z metrik porovnávající jednotlivé pixely, je PSNR (Peak Signal-to-Noise Ratio), špičkový poměr signálu k šumu, který se nejčastěji využívá pro měření kvality rekonstrukce komprese ztrátových kodeků. V tomto případě jsou původní data signálem a chyba komprese šumem.

Dle [58] se PSNR vypočítá z MSE z rovnice:

$$\text{PSNR} = 10 \times \log_{10} \frac{m^2}{\text{MSE}} \quad [\text{dB}] \quad (2)$$

m je maximální hodnota, kterou může nabývat obrazový signál (např. 255 pro 8 bitový obraz).

4.2.3 VQM

Model VQM (Video Quality Measurement) je metrikou kvality videa založenou na diskrétní kosinové transformaci. Měří vliv několika poruch (mimo jiné rozmazání obrazu, trhavý pohyb, barevného zkreslení) a kombinuje je do jedné metriky. Výsledky testů ukazují, že VQM vcelku vysoce koreluje se subjektivními hodnoceními. [59]

K jeho výpočtu dochází ve čtyřech krocích, které jsou popsány v [56].

4.2.4 SSIM

Structural Similarity Index Method je metoda, měřící strukturální zkreslení obrazu k odhadu vnímání vizuálního zkreslení. Vychází z předpokladu, že systém lidského vidění je specializovaný na získávání strukturální informace ze sledovaného obrazu a nespécializuje se na extrakci konkrétních chyb. Měření tohoto strukturálního zkreslení by tedy mělo lépe korelovat se subjektivním hodnocením. [59]

V [60] a [61] je navržen jednoduchý a účinný algoritmus výpočtu.

Tento výčet objektivních metod pro zhodnocení kvality videa není zdaleka úplný. Výše uvedené byly vybrány pro podrobnější popis zejména z důvodu jejich častého využívání v odborných člancích - zdá se, jsou jimi ve velké míře považovány za základní (výchozí) pro pochopení problematiky. Z těch, které nejsou v této práci podrobněji uvedeny, je možné dále zmínit například VIF (Visual Information Fidelity) [62], M-SVD (Mean-Singular Value Decomposition) [63] a další [64] [65].

5 Streamingové služby

V dnešní době velmi pravděpodobně není nutné spekulovat o popularitě online obsahu mezi uživateli. Zatímco obsah vysílaný na TV uživatele spojují spíše s kontextem (tedy sledování „tradičního vysílání“ s rodinou či přáteli), streaming je asociován spíše s obsahem, který cíleně vyhledávají a jehož aspekty jsou pro ně při trávení času atraktivní. Je však pravdou, že streaming je preferovanou platformou zejména pro mileniály (skupinu lidí, narozených mezi lety 1984-1997) a starší ročníky stále upřednostňují tradiční sledování televizního přenosu. [66] [67]

Těchto informací jsou si vědomi i marketingoví specialisté - a to nejen z prováděných průzkumů. Jejich uvědomění je viditelné na reklamním cílení na odlišných multimediálních platformách - reklamy jsou často výrazným (případně jediným) příjmem webů a aplikací, které streamování svým uživatelům nabízí.

Firmy si své „know-how“ pro přenos obsahu uživatelům velice bedlivě střeží a není proto možné se s ním detailně seznámit. Pro vytvoření představy je však nutné uvést alespoň několik nejznámějších (nejvyužívanějších) streamingových služeb.

5.1 Netflix

Netflix je americká společnost, založená 29. srpna 1997 dvojicí Reed Hastings a Mark Randolph v Kalifornii. Zpočátku se zabývala pouze pronájemem nosičů s multimediálním obsahem, avšak roku 2007 vstoupila na trh streamování multimediálního obsahu na internetu a od roku 2013 také vlastní obsah produkuje. Služba Netflix je zpoplatněna měsíčně a na několika cenových úrovních - uživatel dle předplatného volí takovou variantu, která jeho využívání vyhovuje nejlépe.

Mezi zařízení, která jsou s touto službou kompatibilní, je možné zmínit mobilní telefony, Blu-ray přehrávače, HDTV přijímače, set-top boxy, videoherní konzole, tablety, systémy domácích kin, počítače a laptopy, chytré televize. [68]

Kodeky využívané Netflixem se liší v závislosti na cílovém zařízení, na kterém příjemce obsah sleduje. Jsou uváděny zejména kodeky H.264 AVC, VC-1 a H.265 HEVC pro video a AAC či Ogg Vorbis pro audio. [69]

Jelikož sledování Netflix obsahu využívá adaptivního streamingu, přenosová rychlost je vždy volena jako nejvyšší možná (pro HD video se streamovací bitrate

pohybuje kolem 6000 kbit/s [70]). Bitová hloubka je uváděna 16/24 bit v závislosti na užitém kodeku a vzorkovací frekvence je pak shodná - 48 kHz.

Obsah je možné sledovat až v Ultra HD rozlišení, avšak v případě nejdražších verzí předplatného.

5.2 Spotify

V říjnu 2008 ve Švédsku vznikla platforma pro streamování a podcast hudby, nazvaná Spotify. V současné době je služba lokalizovaná do 20 jazyků (včetně češtiny) a dostupná v Oceánii, většině Severní a Jižní Ameriky a v západní Evropě. Je možné ji využívat nejen na chytrých telefonech, ale také na PC a laptotech, herních konzolích.

V současné době využívá ztrátového kodeku Ogg Vorbis a AAC. Datový tok přehrávaných audio souborů se liší na základě verze, která je využívána. Klienti, kteří za službu platí, poslouchají záznamy při 320 kb/s, vlastníci verze, která je poskytována zdarma, pak při 160 kb/s.

5.3 Twitch.tv

Twitch.tv je živá video platforma, založená roku 2011 a od roku 2014 vlastněna společností Amazon. Vysílání této platformy se zaměřuje na videohry a akce s herním účelem, v menší míře také na hudbu a umění. Vysílání je možné sledovat živě, či ze záznamu. Registrace na serveru pro sledování obsahu není vyžadována, je tedy možné jej využívat bez jakýchkoliv poplatků, avšak při úhradě měsíčního předplatného je divákovi umožněno vstupovat do diskuzí, finančně přispívat autorům sledovaných videí, streamovat.

Platforma využívá kodeků H.264 a AAC, doporučený bitrate je uváděn mezi 3-6 Mb/s pro video a 96 kb/s pro audio. Živé vysílání je omezeno maximální délkou - a to 48 hodin. [71]

5.4 Hulu

Hulu je služba, poskytující zejména VoD obsah svým předplatitelům - a to pouze na území Spojených států amerických a Japonska. Primárně se orientuje na televizní seriály a živé vysílání několika stanic, spadajících do stejného

vlastnictví (společnosti 21st Century Fox, Comcast, The Walt Disney Company a Time Warner v USA a Nippon TV v Japonsku).

Vysílání platformy Hulu podporuje zařízení, jako jsou mobilní telefony a tablety, televize, herní konzole, PC a laptopy. [72]

Platforma využívá dvou video kodeků - VP6 (bitrate 480 a 700 kb/s pro nižší rozlišení) a H.264 (bitrate 1000 kb/s a 2500 kb/s pro vyšší rozlišení) a pro audio pak PCM či AAC (s minimálním bitratem 192 kb/s). Bitová hloubka je uváděna 16 či 24 bitů, vzorkovací frekvence 48 kHz.

5.5 Srovnání

V rámci ujasnění a doplnění některých informací je v následujících tabulkách shrnuto několik hlavních informací o jednotlivých službách, případně jejich srovnání (Tabulka 4 a 5).

| Streamovací služba | Dostupné v ČR | Nejnižší poplatky | Počet uživatelů | Výnos za rok 2017 |
|--------------------|---------------|--------------------------|-----------------------|------------------------|
| Netflix | ano | od 7.99 EUR/měsíc | 125 milionů [73] | 11.69 miliard USD [74] |
| Spotify | ano | zdarma či 5.99 EUR/měsíc | 70 milionů [75] | 4.09 miliard EUR [76] |
| Twitch.tv | ano | zdarma | přes 100 milionů [77] | --- ⁹ |
| Hulu | ne | 5.99 USD/měsíc | přes 17 milionů [78] | 1 miliarda USD [78] |

Tabulka 4 Základní uživatelské srovnání streamovacích služeb

⁹ Nenalezeno ve veřejně dostupných zdrojích.

Jak je pravděpodobně čtenářem očekáváno, výběr používaných kodeků v uvedených službách se příliš neliší, jejich výčet je uveden v Tabulce 5.

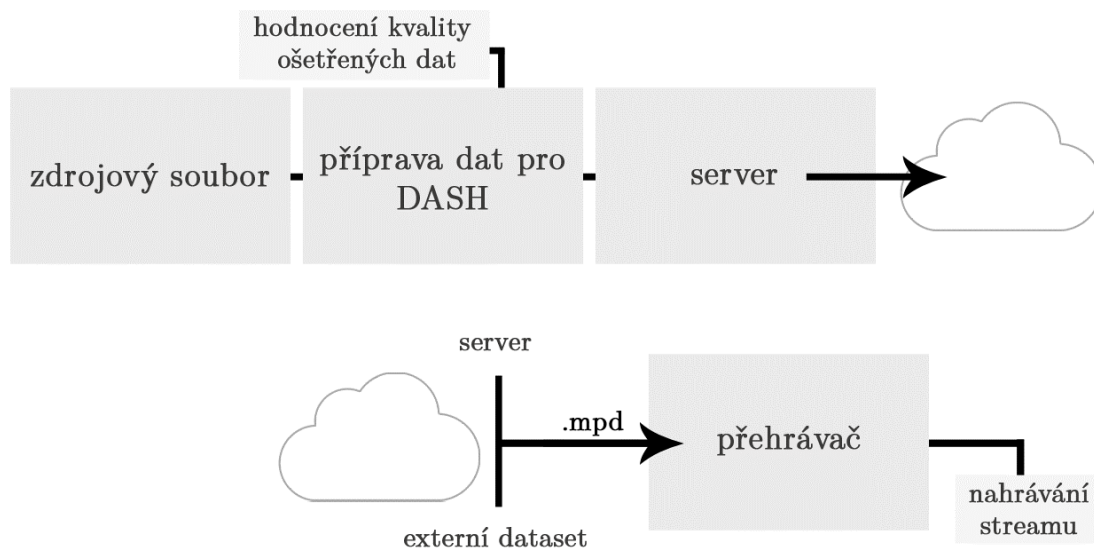
| Streamovací služba | Využívaný video kodek | Využívaný audio kodek |
|---------------------|-----------------------|-----------------------|
| Netflix [69] | H.264, H.265, VC-1 | Ogg Vorbis, AAC |
| Spotify [79] | --- | Ogg Vorbis, AAC |
| Twitch.tv [71] [80] | H.264 | AAC |
| Hulu [81] | VP6, H.264 | AAC, PCM |

Tabulka 5 Kodeky streamingových služeb

6 Streamingový proces

Na základě znalostí nabytých v předchozích kapitolách je možné nastítnit a realizovat celý přenosový řetězec pro streaming od zdrojového videosouboru až po jeho vlastní sledování na straně recipienta (a případné další zpracování), viz schéma na Obrázku 7. Tato kapitola se zaměřuje na praktickou realizaci zadání bakalářské práce. Jejím cílem je popsat a realizovat celý přenosový řetězec tak, aby bylo možné získat výstupní data k případné realizaci subjektivních či objektivních testů. V následujících krocích je proces zaměřen na MPEG DASH, se kterým práce seznamovala v 2.4 a jehož využití bylo součástí zadání.

Celý pracovní proces probíhal na běžném domácím notebooku s operačním systémem Windows Pro 10. Jakožto zdrojový soubor pro modelování takového přenosu byl zvolen krátký animovaný snímek Big Buck Bunny od institutu Blender (část nadace Blender), vydaný pod licencí Creative Commons¹⁰. Tento snímek byl vybrán nejen z důvodu krátké stopáže, ale také díky četnosti jeho využívání v testech, měřeních, přenosech a další práci s videosoubory, celosvětově vykonávaných a publikovaných.



Obrázek 7 Schéma realizovaného řetězce

¹⁰ Big Buck Buuny <https://peach.blender.org/>

6.1 Příprava souboru k přenosu DASH

Po výběru souboru je třeba jej před přenosem ošetřit kodekem a předzpracovat tak, aby byl k dispozici funkční MPD soubor¹¹ (viz 2.4.2).

6.1.1 Použitý software

Pro přípravu a kódování pro DASH bylo zvoleno rozšířených opensource programů FFmpeg a GPAC. Oba jsou ovládány přes příkazovou řádku Windows.

FFmpeg

Kolekce FFmpeg je softwarem pro nahrávání, konverzi a streaming digitálního obsahu. Je primárně vyvíjen pro systém Linux, ale je možné jej zkompileovat také pod Windows a většinu jiných operačních systémů.

FFmpeg obsahuje databázi knihoven – mimo jiné například využití *libavcodec* (audio/video kodéry a dekodéry), *libavformat* (multiplexery a demultiplexery pro kontejnerové audio/video formáty), *libswscale* (změna rozlišení a barevného modelu videa), *libavutil* (pro pomocné operace), *libswresample* a *libavresample* (převzorkování a změna formátu audio/video) a jiné.

Instalační soubor je veřejně dostupný například na oficiálních stránkách FFmpeg¹² nebo FFmpeg Builds¹³.

GPAC

GPAC je open source program pro práci se soubory standardu MPEG-4. Jeho základní využívanou knihovnou příkazů je *libgpac*, umožňující práci se servery a multiplexory (pro streaming), multimediálním přehrávačem a příkazovým řádkem a *MP4box* pro práci se soubory.

Instalační soubor ke stažení například na oficiálních stránkách GPAC - sekce Downloads¹⁴.

¹¹ Často se pro MPD soubor využívá také pojmu „manifest“. [91]

¹² FFmpeg <https://www.ffmpeg.org/>

¹³ Builds - Zerano FFmpeg <https://ffmpeg.zerano.com/builds/>

¹⁴ GPAC Multimedia Open Source Project - Downloads <https://gpac.wp.imt.fr/downloads/gpac-nightly-builds/>

6.1.2 Práce s videosouborem

Po instalaci obou programů a nastavení cest v systémovém nastavení PC je nutné otestovat správnost implementace softwaru (v případě nefunkčnosti těchto částí není možné se zpracováváním dále pokročit). Tento a všechny další kroky se provádějí v příkazové řádce systému Windows.

Správně nainstalované a implementované programy poskytnou následující odezvu (Obrázek 8, Obrázek 9¹⁵ - může se mírně lišit v závislosti na instalované verzi):

```
ffmpeg version N-89936-gf61e6c06d0 Copyright (c) 2000-2018 the FFmpeg developers
  built with gcc 7.2.0 (GCC)
  configuration: --enable-gpl --enable-version3 --enable-sdl2 --enable-bzlib --enable-fontconfig --enable-
gnutls --enable-iconv --enable-libass --enable-libbluray --enable-libfreetype --enable-libmp3lame --enable-
libopencore-amrnb --enable-libopencore-amrwb --enable-libopenjpeg --enable-libopus --enable-libshine --en
able-libsnappy --enable-libsoxr --enable-libtheora --enable-libtwolame --enable-libvpx --enable-libwavpack
--enable-libwebp --enable-libx264 --enable-libx265 --enable-libxml2 --enable-libzimg --enable-lzma --enab
le-zlib --enable-gmp --enable-libvidstab --enable-libvorbis --enable-libvo-amrwbenc --enable-libmysofa --e
nable-libspeex --enable-libxvid --enable-libbmf --enable-amf --enable-cuda --enable-cuvid --enable-d3d11va
--enable-nvenc --enable-dxva2 --enable-avisynth
  libavutil      56. 7.100 / 56. 7.100
  libavcodec     58. 9.100 / 58. 9.100
  libavformat    58. 7.100 / 58. 7.100
  libavdevice    58. 0.101 / 58. 0.101
  libavfilter     7. 11.101 / 7. 11.101
  libswscale     5. 0.101 / 5. 0.101
  libswresample  3. 0.101 / 3. 0.101
  libpostproc   55. 0.100 / 55. 0.100
Hyper fast Audio and Video encoder
usage: ffmpeg [options] [[infile options] -i infile]... {[outfile options] outfile}...
```

Obrázek 8 Výsledek úspěšného testování FFmpeg

```
MP4Box [option] input [option]
-h general          general options help
-h hint            hinting options help
-h dash           DASH segmenter help
-h import         import options help
-h encode         encode options help
-h meta          meta handling options help
-h extract       extraction options help
-h dump          dump options help
-h swf           Flash (SWF) options help
-h crypt         ISMA E&A options help
-h format        supported formats help
-h rtp           file streamer help
-h live         BIFS streamer help
-h all          all options are printed

-nodes           lists supported MPEG4 nodes
-node NodeName  gets MPEG4 node syntax and QP info
-xnodes        lists supported X3D nodes
-xnode NodeName gets X3D node syntax
-snodes        lists supported SVG nodes
-languages     lists supported ISO 639 languages
-boxes         lists all supported ISOBMF boxes and their syntax

-quiet          quiet mode
-noprogram     disables progress
-v            verbose mode
-logs          set log tools and levels, formatted as a ':'-separated list of toolX[:toolZ]@levelX
-log-file FILE sets output log file. Also works with -lf FILE
-log-clock or -lc logs time in micro sec since start time of GPAC before each log line.
-log-utc or -lu logs UTC time in ms before each log line.
-version       gets build version
-- INPUT      escape option if INPUT starts with - character
```

Obrázek 9 Výsledek úspěšného testování MP4Box

¹⁵ Pro lepší čitelnost byly barvy v Obrázku 8 a 9 invertovány.

Následuje samotné zpracování snímku Big Buck Bunny do několika výstupních sekvencí tak, aby cílový MPEG DASH přehrávač mohl přepínat mezi několika kvalitativními úrovněmi dle aktuální potřeby.

Parametry těchto sekvencí je možné volit dle vlastního uvážení, avšak existuje několik doporučení pro kvality streamu a je možné některé z nich uvažovat (jejich dohledání není problematické, pro ukázkou [82] - využito v této práci, Tabulka 6).

| Kvalita | Rozlišení [px] | Video bitrate [kbps] | Audio bitrate [kbps] |
|----------|----------------|----------------------|----------------------|
| Mobilní | 480×270 | 150 | 48 |
| Normální | 512×288 | 350 | 96 |
| Střední | 768×432 | 550 | 128 |
| Vysoká | 848×480 | 1500 | 128 |
| HD | 1280×720 | 2000 | 128 |
| HD+ | 1920×1080 | 5000 | 192 |

Tabulka 6 Zvolené parametry výstupních sekvencí

Obrazová i zvuková část je kódována zvlášť, vstupní příkaz pro kódování videa zní pro příklad následovně:

```
ffmpeg -i input.avi -s 480x270 -c:v libx264 -b:v 150k -g 90 -an
input_video_480x270_150k.mp4
```

Příčemž jednotlivé argumenty znamenají:

- **-i input.avi** - označuje původní video soubor (= vstupní soubor)
- **-s 480x270** - rozlišení, do kterého vstupní soubor kódujeme
- **-c:v libx264** - specifikuje jaký video kodek používáme (v tomto případě H.264)
- **-b:v 150k** - bitrate kódovaného videa
- **-g 90** - délka GOP (Group of Pictures), kterou při MPEG kódování používáme, tzv. *keyframe interval*
- **-an** - nekódujeme audio současně s videem (to je pro DASH přenos kódováno zvlášť)
- **input_video_480x270_150k.mp4** - označuje název výstupního souboru

Ukázka samotného programového toku je ke zhlédnutí viz Příloha 1.

Kódování audia pak probíhá obdobně - s příkazem:

```
ffmpeg -i input.avi -c:a aac -b:a 48k -vn input_audio_48k.mp4
```

- **-i input.avi** - označuje vstupní soubor

- **-c:a aac** - využívaný audio kodek (AAC v tomto případě)
- **-b:a 48k** - audio bitrate
- **-vn input_audio_48k.mp4** - výstupní formát

Ukázka datového toku pak viz Příloha 2.

Doba, za kterou jsou jednotlivé sekvence zpracovány, se liší zejména v závislosti na technických možnostech používaného stroje, jedná se však řádově o jednotky až desítky minut (v případě osobního notebooku).

Po přípravě několika kódovaných verzí souboru včetně audio částí se vytvářejí DASH kompatibilní soubory - MPEG-4 inicializační soubory a MPD manifest, který přehrávač informuje o charakteru potřebných souborů.

Příkaz

```
mp4box -dash 5000 -rap -profile dashavc264:onDemand -mpd-title BBB -out manifest.mpd -frag 2000 input_audio_48k.mp4 ...
```

má pak tyto parametry:

- **-dash 5000** - rozděluje vstupní soubory na segmenty trvající 5 s [83]¹⁶
- **-rap** - umožňuje vyhledávání jednotlivých segmentů (respektive tedy start s náhodným přístupem)
- **-profile dashavc264:onDemand** - využití VOD
- **-mpd-title** - nastaví název manifestu, v tomto případě BBB (zkratka pro „Big Buck Bunny“)
- **-out manifest.mpd** - název výstupního MPD souboru
- **-frag 2000** - nastavení délky fragmentu na 2 s (musí být vždy nižší než -dash)
- na konci příkazu se píší názvy všech souborů, které chceme do MPD zahrnout (tedy všechny vytvořené audio a video soubory)

Příkaz se následovně provede. Jeho průběh je velmi krátký a nesleduje žádné zajímavé změny, jeho programový tok proto není přiložen.

Výstupem předchozích kroků jsou tedy audio, video soubory a manifest MPD - pro realizaci vlastního streamu jinak soubor není třeba zpracovávat. Všechny tyto vytvořené soubory je třeba pouze publikovat (nahrát na server, 6.3) - a to kromě vstupního souboru. Příloha 3 dává tyto soubory k nahlédnutí.

¹⁶ Zhodnocení závislosti délky segmentu na výsledné kvalitě sledovaného streamu se věnuje například [90].

6.1.2.1 Kontrola MPD

Vytvořený MPD soubor není nutné nijak následně editovat, jak již bylo řečeno výše, je však možné ověřit jeho správnost a úplnost. DASH Industry Forum pro tento účel vyvinula a volně zpřístupnila tzv. „DASH Validator“¹⁷, fungující podle oficiální normy ISO/IEC 23009-1.

Soubor manifest.mpd vytvořený v této práci byl tímto validátorem úspěšně ověřen, jak demonstruje Obrázek 10.



Obrázek 10 DASH Validator

6.1.3 Automatizovaná práce s videosouborem

Vzhledem k tomu, že v případech, které jsou popsány v 6.1.2, se jedná o tvorbu dat pomocí příkazové řádky, byl v prostředí MATLAB vytvořen skript, který po uvedení požadavků na kvalitu výstupních sekvencí (dle parametrů, které uvádí Tabulka 6) celý proces tvorby zpracovávání vstupní videosekvence následně provede automaticky - bez nutné přítomnosti a dalšího zásahu uživatele. Nutnost předchozí instalace doporučeného softwaru (6.1.1) a jejich přidání do Path systémové/uživatelské proměnné použitého zařízení (Windows, Vlastnosti systému-Úpřesnit, Proměnné prostředí).

Je možné volit vstupní videosekvenci z libovolného umístění na disku, výsledné soubory se ukládají do složky, ze které jsou skripty spouštěny. Pro spuštění tohoto skriptu je určen soubor *main.m* v Příloze 4 (zakovkri-matlab.rar).

¹⁷ DASH Industry Forum: DASH Validator je dostupný na <http://dashif.org/conformance.html>

6.1.4 Kvalita výstupních souborů

Pokud pouze realizujeme streamovací řetězec, tuto část je možné zcela zanedbat. V rámci hodnocení kvality však je možné pro zajímavost uvést několik provedených měření pro získání představy zda a jakým způsobem se výstupní soubory liší od naprosto nezpracovaných.

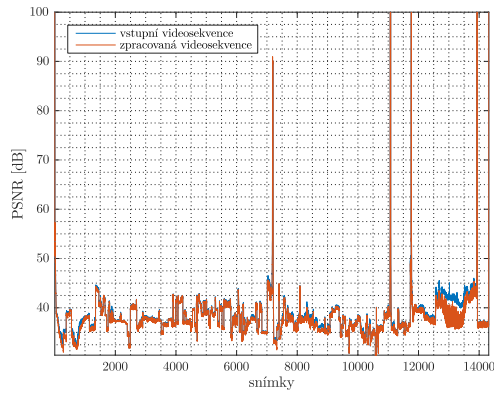
Nezpracovanými daty je myšlen původní (referenční) soubor, vzniklý při nahrávání (vytváření) audiovizuální informace (tedy výstup z animačního programu, videokamery aj), pro objektivní srovnání využívanými metrikami (4.2) je nutné mít tento soubor k dispozici. V případě užití volně šiřitelných snímků je jejich původní formát dostupný většinou na oficiálních stránkách projektu, pro nejčastěji užívané testovací sekvence je kvalitním zdrojem kolekce organizace Xiph.org¹⁸.

Tyto „raw“ nezpracované soubory však mívají velikost několika desítek (až stovek) GB - pro verze, se kterými se pracuje v této práci, byl jako referenční snímek použit videosoubor Big Buck Bunny ve formátu .avi, a to z důvodu omezených možností domácího počítače.

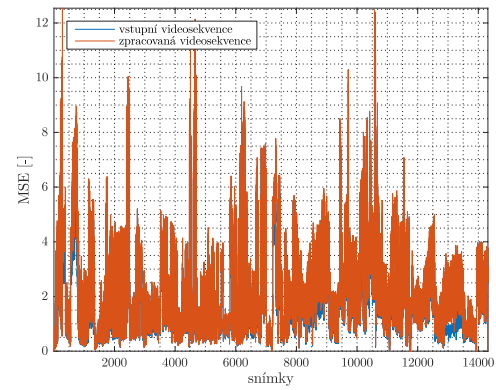
Pro objektivní hodnocení kvality souborů (a zejména pro posuzování efektivity ošetření různými kodeky) byl využit program MSU Video Quality Measurement Tool 64-bit ve verzi PRO Demo¹⁹, grafy vykresleny v prostředí MATLAB (Obrázek 11). Referenčním souborem je nezpracovaná verze snímku, porovnává se jak s vstupním souborem, používaným ke zpracování v 6.1.2., tak s výstupním, který je využíván v manifestu. Rozdíl mezi výsledky, který je možné na grafech pozorovat, je pochopitelný - zpracovávaný vstupní soubor má oproti „raw“ souboru menší velikost - kolem 1 GB (a je také ve ztrátovém .avi formátu) a došlo tak ke ztrátě určitého množství informace. Původní a nezpracovaný „raw“ Big Buck Bunny snímek má cca 42 GB - a jak již bylo řečeno, s takovým objemem nebylo možné, vzhledem k možnostem použitého PC zařízení, pracovat.

¹⁸ Xiph.org Test Media <https://media.xiph.org/>, konkrétněji například Video Test Media [derf's collection] <https://media.xiph.org/video/derf/>

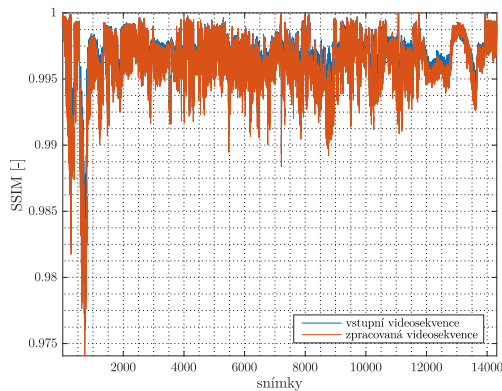
¹⁹ Volně dostupné na oficiálních stránkách MSU Video Quality Measurement Tool http://www.compression.ru/video/quality_measure/video_measurement_tool.html. Jelikož se zdálo vhodné porovnat nejlepší kvalitativní verzi, bylo nutné zvolit verzi Demo Pro, která jako jediná zdarma dostupná umí pracovat se soubory s vyšším rozlišením.



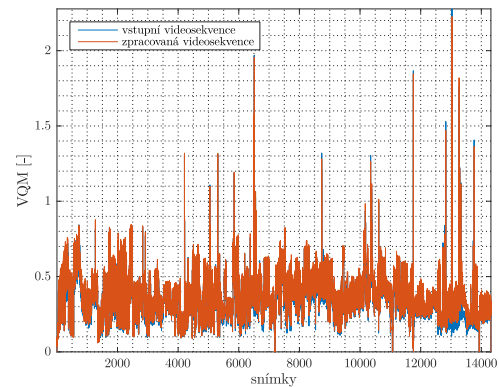
(a) PSNR



(b) MSE



(c) SSIM



(d) VQM

Obrázek 11 Grafy - objektivní hodnocení používaných videosekvencí

Průměrné naměřené PSNR pro vstupní videosekvenci je 37.26 dB, pro zpracovanou videosekvenci pak 36.84 dB. Mírně horší výsledek pro druhý případ je očekávatelný. Vezme-li se v úvahu fakt, že za velmi dobrý výsledek se obecně dá považovat rozmezí 30 až 50 dB a 25 dB je hranicí akceptovatelnosti [84], dosáhlo měření uspokojivých výsledků.

S hodnotou PSNR přímo souvisí také MSE, jak je patrné z rovnice (2) kapitoly 4.2.2 - pro vstupní videosekvenci nabývá průměrně 1.903, pro zpracovanou 2.16. Pro měření metrikou SSIM jsou akceptovatelnými hodnotami 0 „nejmenší shoda“ (největší zkreslení) až 1 „nejlepší shoda“ (nejmenší zkreslení - žádný rozdíl vůči referenci). Z grafu je patrné, že i nepatrně horší zpracovaná videosekvence je té vstupní tak podobná, že jejich rozdíl se projeví až na čtvrté desetinné pozici - a obě tedy po zaokrouhlení nabývají průměrné hodnoty 0.99.

Hodnocení VQM pracuje s výsledky od 0 (která vyjadřuje nejlepší kvalitu - žádný rozdíl oproti referenci), přičemž čím vyšší hodnoty nabývá, tím horší kvalitu vyjadřuje [85]. Průměrné hodnoty VQM pro vstupní videosekvenci jsou po zaokrouhlení 0.35, pro zpracovanou 0.37 - a můžeme tedy vzhledem ke všem

výsledkům hodnocení říci, že námi využívané videosekvence jsou před přenosem zpracovány dobře a věrně k originálu, nedochází k markantnímu zhoršení kvality ještě před samotným přenosem informace.

Shrnutí výstupních hodnot hodnocení je v následující tabulce:

| | | PSNR [dB] | MSE [-] | SSIM [-] | VQM [-] |
|-------------------------------------|--------|--------------|---------|----------|---------|
| Vstupní videosekvence | min | 30.4168 | 0.0 | 0.9766 | 0.0 |
| | max | 100 | 12.4767 | 1 | 2.2773 |
| | průměr | 37.25941 | 1.93086 | 0.99679 | 0.35281 |
| Zpracovaná videosekvence | min | 30.3844 | 0.0 | 0.9741 | 0.0 |
| | max | 100 | 12.5441 | 1 | 2.2260 |
| | průměr | 36.84127 | 2.16086 | 0.99618 | 0.37477 |

Tabulka 7 Hodnocení kvality souborů, shrnutí hodnot

6.2 Alternativní možnost získání dat

V případě, že není žádoucí k přenosu zpracovávat vlastní videosoubor, je možné využívat volně dostupných datasetů, které obsahují již předpřipravené soubory pro MPEG DASH přenos včetně manifestů a jejich zdrojové soubory (kterými jsou videosekvence, které se široce využívají - mimo jiné také Big Buck Bunny) je možné stáhnout přímo v nich, či na výše zmiňovaných stránkách Xipg.org.

- dataset z roku 2014, autor: Florian Bacher²⁰
- dataset z roku 2018, pracující s několika kodeky²¹ [86]
- SVC dataset vytvořený ve spolupráci s ITEC²² [87]

6.3 Server

Používají-li se soubory, dostupné v datasetech, uvedených v 6.2, je možné tuto část přeskočit. Tyto soubory již dostupné na internetu jsou a je možné je tedy číst (respektive stahovat, přehrávat) bez jakékoliv další činnosti. Pokud ale pracujeme

²⁰ dostupný na <http://www-itec.uni-klu.ac.at/ftp/datasets/DASHDataset2014/>

²¹ dostupný na <http://www.itec.aau.at/ftp/datasets/mmsys18/>

²² dostupný na <http://concert.itec.aau.at/SVCDataset/dataset/>

s vlastními soubory, je nutné je veřejně zpřístupnit tak, aby bylo možné k nim externě přistupovat.

Řešením je vytvořit vlastní server, na který soubory umístíme a zpřístupníme dle požadovaného příjemce.

Možností je více, jistě se nabízí varianta webhostingu - uvážíme-li však objem dat, která přenášíme (v případě této práce se jedná například o cca 2.4 GB), zjistíme, že toto řešení by mohlo být ekonomicky náročnější - webservery s takovým volným prostorem jsou vždy placené (či určeny pro výzkumné účely a je třeba dokumentace k jejich schválení).

Proto bylo v rámci této práce využito volně dostupného softwaru Apache HTTP Server²³ k vytvoření lokálního serveru, na který byly umístěny všechny přenášené (vytvořené) soubory.

Proces instalace a zprovoznění takového serveru není obsahem této práce a je jistě možné je dohledat z volně dostupných zdrojů (jako příklad možné uvést [88] [89]²⁴). Funkční Apache Server je však po instalaci a zprovoznění ve výchozím stavu dostupný pouze pro adresaci na vlastní počítač (localhost). Je nutné provést změny v konfiguračním souboru *httpd.conf*, a to:

```
DocumentRoot "celá adresa kořenové složky"  
<Directory />  
    AllowOverride All  
    Require all granted  
</Directory>
```

a také:

```
<Directory "celá adresa kořenové složky">  
Options Indexes FollowSymLinks  
Header set Access-Control-Allow-Origin "*"   
Require all granted  
AllowOverride all  
Require all granted  
</Directory>
```

²³ Apache HTTP Server <https://httpd.apache.org/>

²⁴ Mimo články a publikace je možné často velmi užitečné rady a poznatky nalézt na internetových diskuzích - např. známý Stack Overflow: <https://stackoverflow.com/>.

Případně také v souboru *httpd-vhosts.conf*, velmi obdobně:

```
DocumentRoot celá adresa kořenové složky
<Directory "celá adresa kořenové složky">
Options +Indexes +FollowSymLinks +MultiViews
AllowOverride All
Require all granted
Header always set Access-Control-Allow-Origin "*"
</Directory>
```

Kromě povolení přístupu na server také uživatelům z externích sítí (jiné než z lokální LAN - mobilní sítě atd) je v části *Access-Control-Allow-Origin* "*" povolen tzv. CORS (Cross-Origin Resource Sharing), důležitý pro fungování streamu v přehrávačích. Pro plné fungování této úpravy musí být povolen *headers_module*.

Aby bylo možné číst pro streamování velmi důležitý *manifest.mpd*, je nutné také přidat formát *.mpd* do serverem rozpoznatelných MIME typů. To je možné buď opětovnou úpravou *http.conf* souboru:

```
<IfModule mime_module>
...
AddType application/dash+xml .mpd
...
</IfModule>
```

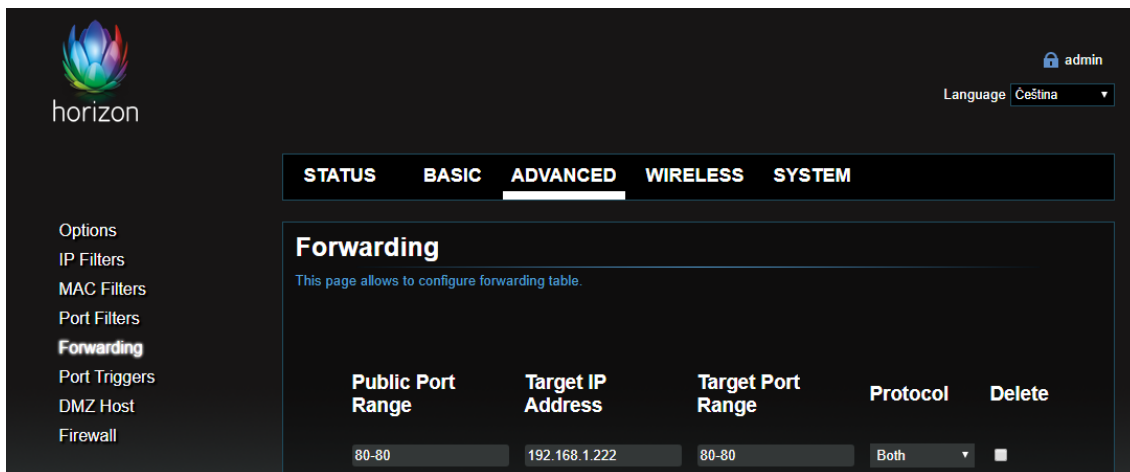
nebo přímým připsáním do souboru *mime.types*:

```
...
application/dash+xml          mpd
...
```

Je samozřejmě nutné také zkontrolovat, že všechny využívané formáty souborů jsou zde podporovány, nicméně většina běžně využívaných formátů bývá.

6.3.1 Porty

Pro veřejnou přístupnost Apache Serveru je velmi důležité otevření portu 80 (který protokol HTTP implicitně používá).



Obrázek 12 UPC internetové připojení, Port Forwarding

To je třeba provést individuálně (Obrázek 12), možnost otestování jednotlivých portů nabízí například stránka YouGetSignal pod názvem Port Forwarding Tester (Open Port Check Tool)²⁵, viz Obrázek 13.

Port Forwarding Tester


your external address

78.102.192.160

open port finder

Remote Address Port Number

[Use Current IP](#)

 Port 80 is open on 78.102.192.160.

Obrázek 13 Otevřený port 80

A jak je tedy patrné, jednotlivé soubory adresujeme pomocí externí IP adresy počítače, na kterém Apache Server běží.²⁶

²⁵ Open Port Check Tool <https://www.yougetsignal.com/tools/open-ports/>

²⁶ Pokud by toto řešení nebylo žádoucí, existuje služba NO-IP, která monitoruje adresu dynamické externí IP adresy počítače a přiřazuje jí vybranou adresu (tzv. hostname). Služba vyžaduje registraci, ale nabízí variantu bez poplatků. Dostupné na: <https://www.noip.com/>

```
===== Tested by attempting to open a socket on port 80 =====  
Your port 80 is actually used by :  
Server: Apache/2.4.17 (Win32) PHP/5.6.15
```

Obrázek 14 Aplikace používající port 80

Velkým problémem pro funkčnost serveru mohou být také nainstalované aplikace, které běží na stejném portu (v tomto případě zpravidla aplikace typu Skype, Team Viewer aj), které je třeba v případě nefunkčnosti serveru ukončit a spustit až po obnovení jeho fungování. Obrázek 14 poukazuje na fakt, že v realizovaném řetězci port 80 jiná aplikace neblokuje.

6.4 Přehrávač

Klientů přehrávajících streamovaný obsah, je mnoho. Často se využívá vlastní implementace dash.js či přehrávače vyvinutého společností Bitmovin²⁷. Pro čisté sledování změn, parametrů a artefaktů streamu existuje DASH IF Reference Client²⁸, ve kterém je možné sledovat parametry testovacích streamů (bitrate, ztracené snímky, velikost vyrovnávací paměti aj). Tento přehrávač však nepodporuje některé adresy, a je proto pro rozšířené využití (měření, sledování změn, zpracovávání informací) mírně nevhodný.

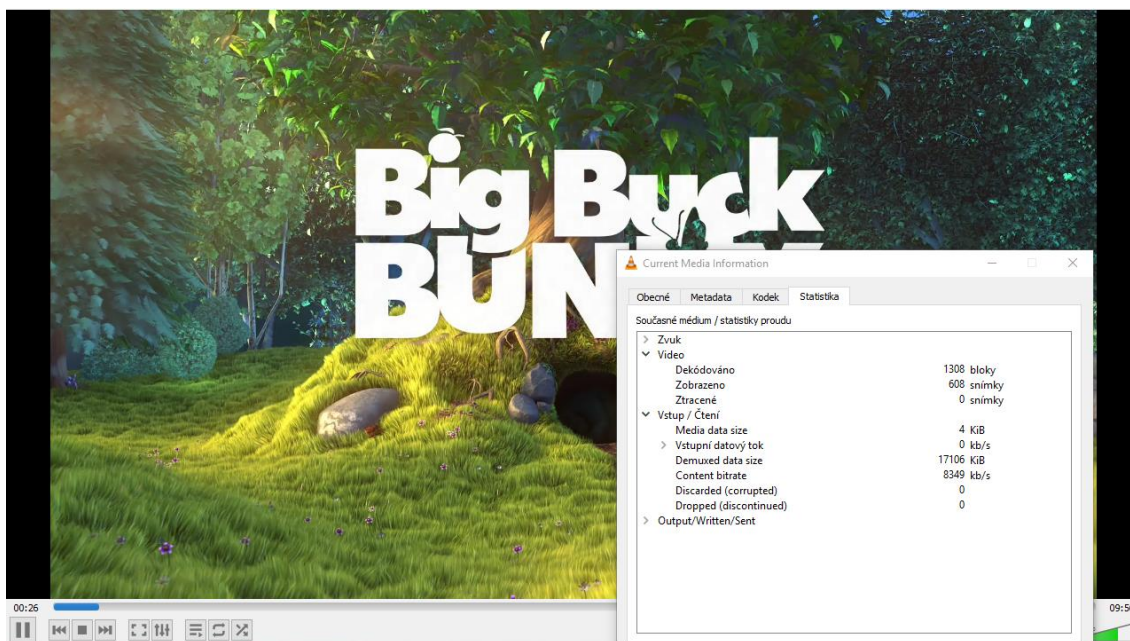
V této práci bylo pro sledování streamu využito velmi rozšířeného freeware multimediálního přehrávače VLC Media Player, který podporuje MPEG DASH a díky svému GUI (a možnosti ovládání přes příkazovou řádku Windows) je jeho využití jednoduché a intuitivní.

Po otevření programu VLC stačí stisknout CTRL+N, nebo zvolit Média → Otevřít síťový proud, následně zadat URL adresu souboru MPD a zvolit možnost Přehrát. Stream se začne načítat a přehrávat v kvalitě, která odpovídá aktuálnímu stavu sítě. V běžném provozu na domácí WiFi síti to znamená načtení HD rozlišení během několika sekund.

Po celou dobu přehrávání je také možno sledovat statistiku streamu (CTRL+I nebo Nástroje → Informace o médiu → Statistiky), kde je monitorováno několik parametrů, které lze vidět na následujícím obrázku.

²⁷ Bitmovin Player for Adaptive Streaming <https://bitmovin.com/video-player/>

²⁸ dash.js JavaScript Reference Client <http://reference.dashif.org/dash.js/>



Obrázek 15 VLC, aktuální informace o médiu

Ne vždy je však vhodné (a možné) sledovat změny a vlastnosti přímo - navíc v případě, že chceme přenesený stream nějakým způsobem dále zpracovávat (hodnotit, prezentovat, sledovat opakovaně totéž atd). Z tohoto důvodu je žádané mít možnost zachytit přijímaný stream a uložit jej. K tomu je možné využít příkazové řádky a příkazu:

```
vlc adresa mpd souboru --sout="#std{access=file,mux=ts,dst=cílová adresa souboru}" vlc://quit
```

který nahrává přijímaný stream a po jeho konci se sám ukončí - není tedy třeba přítomnosti u počítače během celé doby přehrávání videa (v případě Big Buck Bunny například čas kolem 10 minut) a umožňuje jej uchovat například na disku a dále s ním pracovat. Ukládá do většiny formátů videa (z přímo vyzkoušených například .avi, .mpg, .mp4).

6.5 Změna stavu připojení

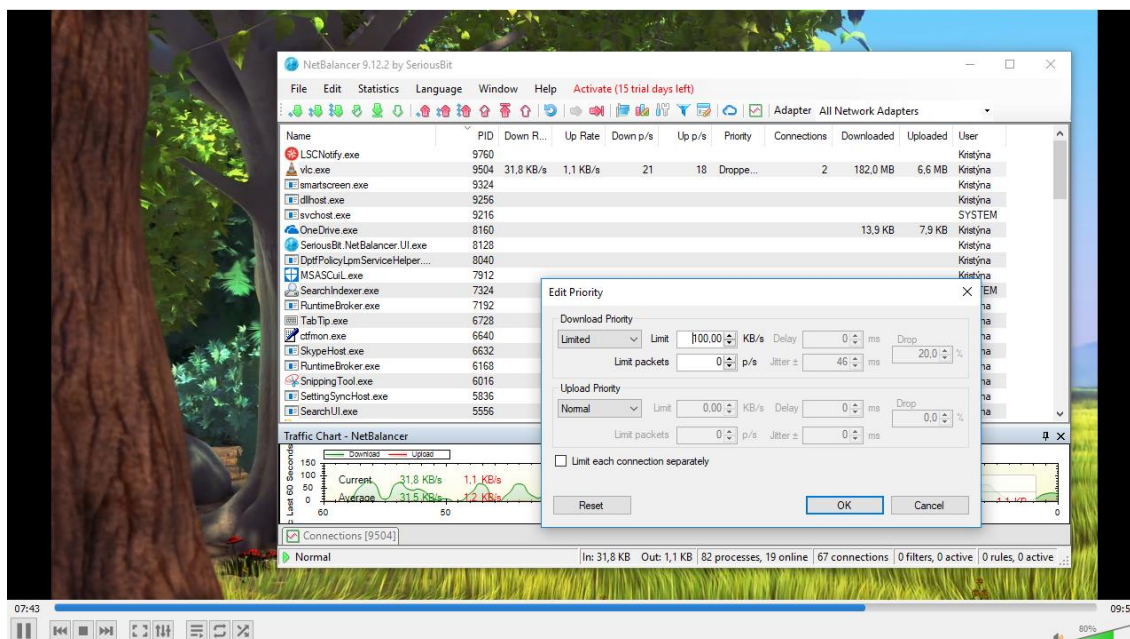
Pro sledování různých artefaktů a vlivů připojení na kvalitu streamu je možné omezovat internetové připojení a sledovat změny, které s takovými omezeními přicházejí.

V nejjednodušším případě je možné nastavení úrovně (kvality) internetového připojení přímo v aplikaci (či programu) poskytovatele internetu - podobně jako bylo v 6.3.1 pracováno s porty.

Existují však aplikace, které monitorují a omezují kvalitu připojení a případně umožňují nastavit hodnoty pro jitter a jiné chybové parametry přenosu. Z volně dostupných byly vyzkoušeny programy NetLimiter 4.0.35²⁹ a NetBalancer 9.12.2³⁰. Oba je možné bez poplatku využívat jen po omezenou dobu, avšak umožňují měnit parametry pouze pro jednotlivé aplikace - tedy separátně jen pro VLC přehrávač a to v reálném čase.

V obou případech ovlivňování spočívá v přidělování pravidel pro omezení připojení. V případě NetLimiteru je tato možnost nepříliš atraktivní - je možné pouze omezovat limit stahování, což pro streamování není zcela žádoucí.

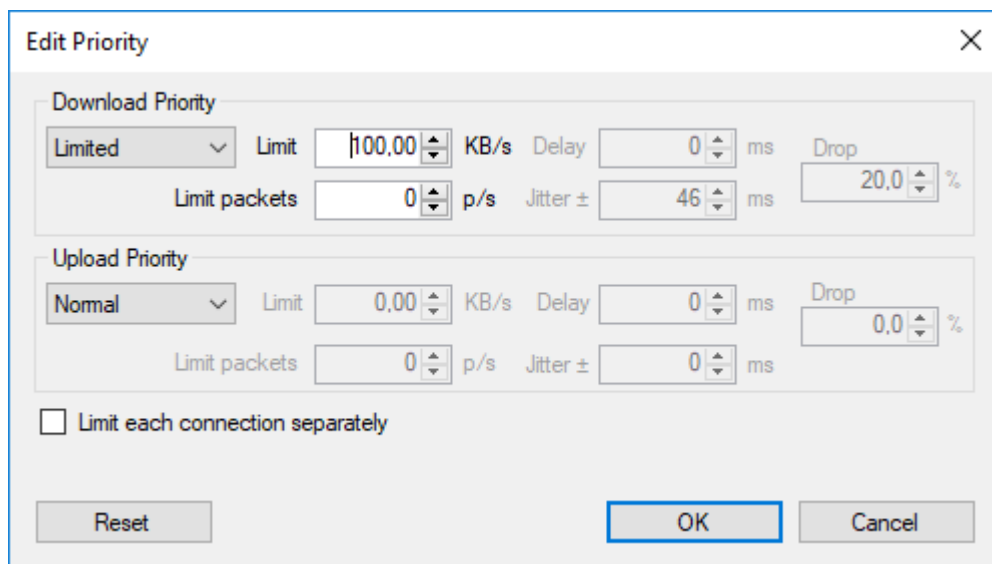
NetBalancer však nabízí možnost ztráty paketů, limitování přenosové rychlosti, nastavení zpoždění - byl proto shledán pro použití ve spojení s VLC efektivnější (Obrázek 16 a 17).



Obrázek 16 NetBalancer

²⁹ ke stažení na <https://www.slunecnice.cz/sw/netlimiter/>

³⁰ ke stažení na <https://www.slunecnice.cz/sw/netbalancer/>



Obrázek 17 NetBalancer - detail, možné priority

Po několika prováděných změnách a sledování jejich vlivu na přehrávání musím jakožto divák zkonstatovat, že změna kvality byla viditelná pouze místy (zejména v obrazových přechodech), zato rebuffering a sekání je při sledování ztelnější. Jeden z přenosů byl dle 6.4 přes příkazovou řádku uložen a je možné na něm sledovat nejen počáteční nabíhání kvality, ale také mírné přenosové artefakty. Tento přenos je v přehratelném formátu přiložen k odevzdané práci, Příloha 5.

7 Závěr

Očekávání kladená na přenos a přehrání audiovizuálního obsahu se v průběhu let stále zvyšují. Streaming se stává mocným nástrojem na komerční i nekomerční scéně - oslovuje diváky širokého spektra a je tedy významným nosičem a šířitelem informace.

Práce seznámila s pojmem „streaming multimediálního obsahu“ - zejména videa - a základními informacemi s ním souvisejícími - kompresí dat, způsobem jejich přenosu. Při seznámení se zaměřuje na aktuální standard adaptivního streamování MPEG DASH, implementující všechny předchozí způsoby streamování s přizpůsobováním kvality obsahu, uvádí funkce, které jej „vzdvihují“ mezi ostatními řešeními, seznamuje se změnami, které přináší.

Zmiňuje kvalitu uživatelské zkušenosti se sledovaným streamem, uvádí faktory, které ji ovlivňují a způsoby, jak je možné ji hodnotit. Zejména pro povrchovou orientaci v tom, které služby jsou právě populární, je uveden jejich přehled a srovnání - rozhodně však není možné říci, že se jedná o shrnutí směřodatné a kompletní - pro každého diváka je důležitý jiný obsah a trh se streamingovými službami v dnešní době zažívá veliký boom, je tedy možné, že čtenář nenajde svou preferovanou či využívanou platformu.

Dále byla zdokumentovány kroky realizace celého streamovacího procesu s počátkem ve výběru přenášeného videa a jeho přípravě a kódování. Zejména z důvodu vytvoření konkrétní představy o hodnotách, vycházejících ze subjektivního hodnocení kvality videa, bylo několik z metod pro objektivní hodnocení provedeno se sekvencemi s nejvyšší kvalitou, která byla zpracovávána a jejich výsledky jsou v práci uvedeny.

Dále popisuje realizaci (a nastavení) serveru, na němž je přenášený obsah uložen s volným přístupem z libovolné sítě. Nechybí ani přehrání videa a návrh k monitorování a ovlivnění jeho kvality, stejně jako uvedení možnosti (a přiložená ukázka) pro zachycení takto ovlivněného obsahu pro pozdější využití.

V návaznosti na to (a samotnou bakalářskou práci) je možné zmínit několik potenciálních pokračování v této problematice - prohloubení přehledu aktuálního trhu se streamingovými službami, vyhlídky do budoucna, či provedení hodnocení získaných a přehraných videí například formou subjektivního testování v laboratorním prostředí - ideálně pomocí testovacího rozhraní, které

by automatizovaným způsobem simulovalo vlastnosti přenosového média. Některé z těchto návrhů by mohly být také výchozí pro mou případnou diplomovou práci.

8 Použité zkratky

| | |
|-------|--|
| 3GPP | Third Generation Partnership Project |
| AAC | Advanced Audio Coding |
| ACR | Absolute Category Rating |
| AS | Adaptation Set |
| AVC | Advanced Video Coding |
| BMFF | Base Media File Format |
| CDN | Content Delivery Networks |
| CORS | Cross-Origin Resource Sharing |
| CPU | Central Processing Unit |
| DASH | Dynamic Adaptive Streaming over HTTP |
| DCR | Degradation Category Rating |
| DSCQS | Double Stimulus Continuous Quality Scale |
| DSIS | Double Stimulus Impairment Scale |
| DVB-T | Digital Video Broadcasting - Terrestrial |
| FR | Full-Reference |
| GOP | Group of Pictures |
| GUI | Graphic User Interface |
| HDS | HTTP Dynamic Streaming |
| HDTV | High-Definition Television |
| HEVC | High Efficiency Video Coding |
| HLS | HTTP Live Streaming |
| HTTP | Hypertext Transfer Protocol |
| IEC | International Electrotechnical Commission |
| ISDN | Integrated Services Digital Network |
| ISO | International Organization for Standardization |
| ITU | International Telecommunication Union |
| LAN | Local Area Network |
| M2TS | MPEG-2 Transport Stream |
| MIME | Multipurpose Internet Mail Extensions |
| MPD | Media Presentation Description |
| MPEG | Moving Pictures Experts Group |
| MSE | Mean Square Error |

| | |
|-------|--|
| MSU | Michigan State University |
| NR | No-Reference |
| OIPF | The Open IPTV Forum |
| PC | Pair Comparison |
| PSNR | Peak Signal-to-Noise Ratio |
| QoE | Quality of Experience |
| QoS | Quality of Service |
| RMSE | Root Mean Square Error |
| RR | Reduced-Reference |
| RTCP | Real Time Control Protocol |
| RTMP | Real Time Messaging Protocol |
| RTP | Real Time Protocol |
| RTSP | Real Time Streaming Protocol |
| RTSP | Real Time Streaming Protocol |
| SC | Stimulus Comparison |
| SDSCE | Simultaneous Double Stimulus Continuous Evaluation |
| SS | Single Stimulus |
| SS | Smooth Streaming |
| SSCQE | Single Stimulus Continuous Quality Evaluation |
| SSIM | Structural Similarity Index Method |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |
| URL | Uniform Resource Locator |
| URL | Uniform Resource Locator |
| VCEG | Video Coding Experts Group |
| VCEG | Video Coding Experts Group |
| VIF | Visual Information Fidelity |
| VoD | Video on Demand |
| VQM | Video Quality Measurement |
| W3C | World Wide Web Consortium |
| WMA | Windows Media Audio |
| WMV | Windows Media Video |
| XML | eXtensible Markup Language |

9 Seznam použité literatury

- [1] COSTELLO, Sam. Internet Streaming: What It Is and How It Works. In: *Lifewire* [online]. b.r. [cit. 2018-05-16]. Dostupné z: <https://www.lifewire.com/internet-streaming-how-it-works-1999513>
- [2] ZHENG, Yuanhuan, Di WU, Yihao KE, Can YANG, Min CHEN a Guoqing ZHANG. Online Cloud Transcoding and Distribution for Crowdsourced Live Game Video Streaming. *IEEE Transactions on Circuits and Systems for Video Technology*. 2017, **27**(8), 1777-1789. DOI: 10.1109/TCSVT.2016.2556584. ISSN 1051-8215. Dostupné také z: <http://ieeexplore.ieee.org/document/7457232/>
- [3] Co je to streamování videa?. *IBM* [online]. b.r. [cit. 2018-05-15]. Dostupné z: <https://www.ibm.com/cloud-computing/cz-cs/video-streaming/>
- [4] CONKLIN, Gregory, Gary GREENBAUM, Karl LILLEVOLD, Alan LIPPMAN a Yuriy REZNIK. Video Coding for Streaming Media Delivery on the Internet. *IEEE Transactions on Circuits and Systems for Video Technology*. IEEE, 2001, **11**(3), 269-281.
- [5] Cisco Visual Networking Index: Forecast and Methodology, 2016–2021. In: *Cisco* [online]. b.r. [cit. 2018-03-21]. Dostupné z: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf>
- [6] OZERA, Jan. What is a codec?. In: *StreamingMedia.com* [online]. 2000 [cit. 2018-12-12]. Dostupné z: <http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-a-Codec-74487.aspx>
- [7] ROBERTSON, Mark. VIDEO FILE CONTAINER FORMATS, COMPRESSION AND CODECS – OH MY!. In: *Tubular Insights* [online]. b.r. [cit. 2017-12-12]. Dostupné z: <http://tubularinsights.com/file-formats-containers-compression/>
- [8] HASKELL, B. a A. PURI. MPEG Video Compression Basics. *The MPEG digital representation of information*. New York, NY: Springer, 2012, s. 7-38. ISBN 9781441961839.

- [9] VETRIVEL, S., K. SUBA a G. ATHISHA. An Overview of H.26x Series and Its Applications. *International Journal of Engineering Science and Technology*. 2010, **2**(9), 4622-4631.
- [10] Information technology -- Coding of audio-visual objects: Part 10: Advanced Video Coding. *ISO - International Organization for Standardization* [online]. b.r. [cit. 2018-05-22]. Dostupné z: <https://www.iso.org/standard/66069.html>
- [11] An Overview of H.264 Advanced Video Coding. *Vcodex* [online]. b.r. [cit. 2017-12-12]. Dostupné z: <https://www.vcodex.com/an-overview-of-h264-advanced-video-coding/>
- [12] SULLIVAN, Gary, Jens-Rainer OHM, Woo-Jin HAN a Thomas WIEGAND. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology* [online]. 2012, **22**(12), 1649-1668 [cit. 2018-05-16]. DOI: 10.1109/TCSVT.2012.2221191. ISSN 1051-8215. Dostupné z: <http://ieeexplore.ieee.org/document/6316136/>
- [13] MOOK, Nate. Microsoft VC-1 Codec Now a Standard. In: *BetaNews* [online]. 1998 [cit. 2017-12-12]. Dostupné z: <https://betanews.com/2006/04/03/microsoft-vc-1-codec-now-a-standard/>
- [14] Theora FAQ. *Theora.org* [online]. b.r. [cit. 2017-12-12]. Dostupné z: <https://theora.org/faq/>
- [15] BIENIK, Juraj, Miroslav UHRINA, Michal KUBA a Martin VACULIK. Performance of H.264, H.265, VP8 and VP9 Compression Standards for High Resolutions. *2016 19th International Conference on Network-Based Information Systems (NBiS)*. IEEE, 2016, , 246-252. DOI: 10.1109/NBiS.2016.70. ISBN 978-1-5090-0979-4. Dostupné také z: <http://ieeexplore.ieee.org/document/7789765/>
- [16] CUSIAC, GM. Audio codec choices for great live streaming video and recording. In: *Epiphan Video* [online]. 2002 [cit. 2017-12-03]. Dostupné z: <https://www.epiphan.com/blog/audio-codec-choices/>
- [17] BRANDENBURG, Karlheinz. MP3 and AAC explained. *AES 17th International Conference on High Quality Audio Coding*. New York, NY, 1999, 1-12.

- [18] Vorbis FAQ. *Vorbis.com* [online]. b.r. [cit. 2017-12-12]. Dostupné z: <http://www.vorbis.com/faq/>
- [19] ARORA, Manas a Neha MAURYA. Audio Compression in MPEG Technology. *International Journal of Scientific and Research Publications*. 2013, **3**(12), 266-269. ISSN 2250-3153.
- [20] CIKA, P., D. KOVAC, V. SKORPIL a T. SRNEC. Subjective comparison of modern video codecs. *2017 Progress In Electromagnetics Research Symposium - Spring (PIERS)*. IEEE, 2017, , 776-779. DOI: 10.1109/PIERS.2017.8261847. ISBN 978-1-5090-6269-0. Dostupné také z: <http://ieeexplore.ieee.org/document/8261847/>
- [21] NARANG, Nitin. #6 Concept Series : What is the difference between Progressive Download, RTMP Streaming and Adaptive Streaming. In: *Media & Entertainment Industry Trends, Technology and Research* [online]. 2010 [cit. 2018-05-16]. Dostupné z: <http://www.mediaentertainmentinfo.com/2015/04/6-concept-series-what-is-the-difference-between-progressive-download-rtmp-streaming-and-adaptive-streaming.html/>
- [22] CHURCH, Cameron. Progressive Download versus Streaming. In: *Brightcove / The Leading Online Video Platform* [online]. b.r. [cit. 2018-05-16]. Dostupné z: <https://www.brightcove.com/en/blog/2009/03/progressive-download-versus-streaming>
- [23] PISHON, Crystle. Understanding the different types of On-Demand streaming. In: *Planet Stream* [online]. b.r. [cit. 2018-12-17]. Dostupné z: <http://www.planetstream.net/blog/streaming-on-demand-types.html>
- [24] MCGATH, Gary. Basics of streaming protocols. In: *Gary McGath: Professional Software Developer* [online]. 2013 [cit. 2017-12-21]. Dostupné z: <http://www.garymcgath.com/streamingprotocols.html>
- [25] TCP, Transmission Control Protocol. *Network Sorcery: RFC Sourcebook* [online]. b.r. [cit. 2018-05-16]. Dostupné z: <http://www.networksorcery.com/enp/protocol/tcp.htm>
- [26] Information technology -- Dynamic adaptive streaming over HTTP (DASH): Part 1: Media presentation description and segment formats. *ISO -*

- International Organization for Standardization* [online]. b.r. [cit. 2018-05-22]. Dostupné z: <https://www.iso.org/standard/65274.html>
- [27] PENNINGTON, Adrian. MPEG-DASH: Why it Matters, Where it Goes From Here. In: *Streaming Media Europe* [online]. 2013 [cit. 2018-04-25]. Dostupné z: <http://www.streamingmediaglobal.com/Articles/ReadArticle.aspx?ArticleID=92147&PageNum=1>
- [28] THANG, Truong, Quang-Dung HO, Jung KANG a Anh PHAM. Adaptive streaming of audiovisual content using MPEG DASH. *IEEE Transactions on Consumer Electronics*. 2012, **58**(1), 78-85. DOI: 10.1109/TCE.2012.6170058. ISSN 0098-3063. Dostupné také z: <http://ieeexplore.ieee.org/document/6170058/>
- [29] LEDERER, Stefan. Why YouTube & Netflix use MPEG-DASH in HTML5. In: *Bitmovin* [online]. b.r. [cit. 2017-12-21]. Dostupné z: <https://bitmovin.com/status-mpeg-dash-today-youtube-netflix-use-html5-beyond/>
- [30] SODAGAR, Iraj. The MPEG-DASH Standard for Multimedia Streaming Over the Internet. *IEEE Multimedia*. 2011, **18**(4), 62-67. DOI: 10.1109/MMUL.2011.71. ISSN 1070-986x. Dostupné také z: <http://ieeexplore.ieee.org/document/6077864/>
- [31] SEUFERT, Michael, Sebastian EGGER, Martin SLANINA, Thomas ZINNER, Tobias HOBFELD a Phuoc TRAN-GIA. *A Survey on Quality of Experience of HTTP Adaptive Streaming*. 2015, **17**(1), 469-492. DOI: 10.1109/COMST.2014.2360940. ISSN 1553-877X. Dostupné také z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6913491>
- [32] HOSSFELD, T., S. EGGER, R. SCHATZ, M. FIEDLER, K. MASUCH a C. LORENTZEN. Initial delay vs. interruptions: Between the devil and the deep blue sea. *2012 Fourth International Workshop on Quality of Multimedia Experience*. IEEE, 2012, , 1-6. DOI: 10.1109/QoMEX.2012.6263849. ISBN 978-1-4673-0726-0. Dostupné také z: <http://ieeexplore.ieee.org/document/6263849/>
- [33] QI, Yining a Mingyuan DAI. The Effect of Frame Freezing and Frame Skipping on Video Quality. *2006 International Conference on Intelligent Information Hiding and Multimedia*. IEEE, 2006, , 423-426. DOI: 10.1109/IIH-

- MSP.2006.265032. ISBN 0-7695-2745-0. Dostupné také z:
<http://ieeexplore.ieee.org/document/4041752/>
- [34] DAPENG WU, Dapeng, Y.T. HOU, Wenwu WENWU ZHU, Ya-Qin YA-QIN ZHANG a J.M. PEHA. Streaming video over the Internet: approaches and directions. *IEEE Transactions on Circuits and Systems for Video Technology*. 2001, **11**(3), 282-300. DOI: 10.1109/76.911156. ISSN 10518215. Dostupné také z: <http://ieeexplore.ieee.org/document/911156/>
- [35] YAO, Jun, Salil KANHERE, Imran HOSSAIN a Mahbub HASSAN. Empirical Evaluation of HTTP Adaptive Streaming under Vehicular Mobility. *NETWORKING 2011*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, , 92-105. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-20757-0_8. ISBN 978-3-642-20756-3. Dostupné také z: http://link.springer.com/10.1007/978-3-642-20757-0_8
- [36] ZINNER, Thomas, Tobias HOSSFELD a Markus FIEDLER. Controlled vs. Uncontrolled Degradations of QoE – The Provisioning-Delivery Hysteresis in Case of Video. *New Dimensions in the Assessment and Support of Quality of Experience (QoE) for Multimedia Applications* [online]. Tampere, Finland: Proc. 1st Workshop QoEMCS, 2010, , 1-3 [cit. 2018-04-25]. Dostupné z: <https://www.diva-portal.org/smash/get/diva2:835273/FULLTEXT01.pdf>
- [37] *ISO/IEC 14496-14: Information technology — Coding of audio-visual objects — Part 14: MP4 file format* [online]. b.r. [cit. 2018-05-16]. Dostupné z: http://jchblog.u.qiniudn.com/doc/ISO_IEC_14496-14_2003-11-15.pdf
- [38] ARSAN, Taner. Review of bandwidth estimation tools and application to bandwidth adaptive video streaming. *High Capacity Optical Networks and Emerging/Enabling Technologies*. IEEE, 2012, , 152-156. DOI: 10.1109/HONET.2012.6421453. ISBN 978-1-4673-2890-6. Dostupné také z: <http://ieeexplore.ieee.org/document/6421453/>
- [39] JARNIKOV, Dmitri a Tanir OZCELEBI. Client intelligence for adaptive streaming solutions. *2010 IEEE International Conference on Multimedia and Expo*. IEEE, 2010, , 1499-1504. DOI: 10.1109/ICME.2010.5583268. ISBN 978-1-4244-7491-2. Dostupné také z: <http://ieeexplore.ieee.org/document/5583268/>

- [40] What Data Compression Does To Your Music. In: *Sound On Sound* [online]. SOS Publications Group, b.r. [cit. 2018-05-16]. Dostupné z: <https://www.soundonsound.com/techniques/what-data-compression-does-your-music>
- [41] Compression artifact. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2018-05-16]. Dostupné z: https://en.wikipedia.org/wiki/Compression_artifact
- [42] PELIKÁN, Josef. CGG MFF UK PRAHA. *Anti-aliasing a vzorkovací metody*. b.r. Dostupné také z: <http://cgg.mff.cuni.cz/~pepca/lectures/pdf/pg2-13-sampling.pdf>
- [43] BASITH, Shanawaz a Stephen DONE. *Digital Video, MPEG and Associated Artifacts* [online]. b.r. [cit. 2018-05-16]. Dostupné z: https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/sab/report.html#compdigi
- [44] Video Compression as Fast as Possible. In: *Youtube* [online]. b.r. [cit. 2018-02-05]. Dostupné z: <https://youtu.be/qbGQBT2Vwvc>
- [45] TAYLOR, Jim, Mark JOHNSON a Charles CRAWFORD. *Velký průvodce DVD: jedinečný zdroj všech dostupných informací o DVD na profesionální úrovni*. 1. vyd. Praha: Grada, 2007. Profesionál. ISBN 978-80-247-1721-0.
- [46] MSU Video Quality Measurement Tool. *MSU Graphics & Media Lab (Video Group)* [online]. Moskva, 2008 [cit. 2018-04-25]. Dostupné z: http://www.compression.ru/video/quality_measure/video_measurement_tool.html
- [47] KADDAR, Lamia. Chapter 14: Incorporating Security and Energy Efficiency for Multimedia Communications in WANets: A Cross-Layer Design. *Technological advancements and applications in mobile ad-hoc networks: research trends*. Hershey, PA: Information Science Reference, 2012, s. 228-246. ISBN 9781466603219.
- [48] *ITU-R BT.500-13: Methodology for the subjective assessment of the quality of television pictures* [online]. b.r. [cit. 2018-04-25]. Dostupné z: https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.500-13-201201-I!!PDF-E.pdf

- [49] *ITU-T P.910: Subjective video quality assessment methods for multimedia applications* [online]. b.r. [cit. 2018-04-25]. Dostupné z: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-P.910-200804-I!!PDF-E&type=items
- [50] *ITU-T P.800.1: Mean opinion score (MOS) terminology* [online]. b.r. [cit. 2018-04-25]. Dostupné z: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-P.800.1-201607-I!!PDF-E&type=items
- [51] KAWAYOKE, Yoshikazu a Yuukou HORITA. NR objective continuous video quality assessment model based on frame quality measure. *2008 15th IEEE International Conference on Image Processing*. IEEE, 2008, , 385-388. DOI: 10.1109/ICIP.2008.4711772. ISBN 978-1-4244-1765-0. Dostupné také z: <http://ieeexplore.ieee.org/document/4711772/>
- [52] *ITU-R BT.1683-0: Objective perceptual video quality measurement techniques for standard definition digital broadcast television in the presence of a full reference* [online]. 2004 [cit. 2018-04-27]. Dostupné z: https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.1683-0-200406-I!!PDF-E.pdf
- [53] SOO-CHUL HAN, a C.I. PODILCHUK. Video compression with dense motion fields. *IEEE Transactions on Image Processing*. b.r., **10**(11), 1605-1612. DOI: 10.1109/83.967388. ISSN 10577149. Dostupné také z: <http://ieeexplore.ieee.org/document/967388/>
- [54] JUNGHYUN HAN, , YO-HAN KIM a JITAE SHIN. Video quality estimation for packet loss based on No-Reference method. *The 12th International Conference on Advanced Communication Technology ICT for Green Growth and Sustainable Development, Phoenix Park, Korea, Feb. 7-10, 2010, proceedings: ICACT 2010*. Piscataway, N.J: IEEE, 2010, s. 418-421. ISBN 9788955191462.
- [55] MENDI, E, M MILANOVA, Y ZHOU a J TALBURT. Objective Video Quality Assessment for Tracking Moving Objects from Video Sequences. *Recent advances in signal processing, robotics and automation: proceedings of the 9th WSEAS International Conference on Signal Processing, Robotics and Automation (ISPRA '10) ; University of Cambridge, UK, February 20 - 22, 2010*. S.l.: WSEAS Press, 2010, s. 121-126. ISBN 9789604741571.

- [56] XIAO, Feng. *DCT-based Video Quality Evaluation: Final Project for EE392J* [online]. b.r., , 1-10 [cit. 2018-04-25]. Dostupné z: http://compression.ru/video/quality_measure/vqm.pdf
- [57] WU, H. a K. RAO. *Digital video image quality and perceptual coding*. Boca Raton, 2006. ISBN 08-247-2777-0.
- [58] WINKLER, Stefan. *Digital video quality: vision models and metrics*. Hoboken, NJ, 2005. ISBN 04-700-2404-6.
- [59] WANG, Yubing. *Survey of Objective Video Quality Measurements* [online]. EMC Corporation, 2006, , 1-7 [cit. 2018-04-25]. Dostupné z: <https://digitalcommons.wpi.edu/computerscience-pubs/42/>
- [60] ZHOU WANG, a A.C. BOVIK. A universal image quality index. *IEEE Signal Processing Letters*. 2002, **9**(3), 81-84. DOI: 10.1109/97.995823. ISSN 1070-9908. Dostupné také z: <http://ieeexplore.ieee.org/document/995823/>
- [61] WANG, Z., A.C. BOVIK, H.R. SHEIKH a E.P. SIMONCELLI. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*. 2004, **13**(4), 600-612. DOI: 10.1109/TIP.2003.819861. ISSN 1057-7149. Dostupné také z: <http://ieeexplore.ieee.org/document/1284395/>
- [62] CHIKKERUR, Shyamprasad, Vijay SUNDARAM, Martin REISSLEIN a Lina KARAM. Objective Video Quality Assessment Methods: A Classification, Review, and Performance Comparison. *IEEE Transactions on Broadcasting* [online]. 2011, **57**(2), 165-182 [cit. 2018-05-16]. DOI: 10.1109/TBC.2011.2104671. ISSN 0018-9316. Dostupné z: <http://ieeexplore.ieee.org/document/5710601/>
- [63] CIKA, Petr, Dominik KOVAC a Jan BILEK. Objective video quality assessment methods: Video encoders comparison. *2015 7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)* [online]. IEEE, 2015, , 335-338 [cit. 2018-05-16]. DOI: 10.1109/ICUMT.2015.7382453. ISBN 978-1-4673-9283-9. Dostupné z: <http://ieeexplore.ieee.org/document/7382453/>
- [64] ARUN KUMAR, P. a S. CHANDRAMATHI. Video Quality Assessment Methods: A Bird's-Eye View. *International Journal of Computer and Information Engineering* [online]. World Academy of Science, Engineering and

- Technology, 2014, **8**(5), 772-778 [cit. 2018-05-16]. Dostupné z: <https://waset.org/publication/-Video-Quality-Assessment-Methods:-A-Bird%E2%80%99s-Eye-View-/9998257>
- [65] MOHAMMADI, Pedram, Abbas EBRAHIMI-MOGHADAM a Shahram SHIRANI. Subjective and Objective Quality Assessment of Image: A Survey. *Majlesi Journal of Electrical Engineering* [online]. 2015, **9**(1), 55-83 [cit. 2018-05-16]. Dostupné z: <http://mjee.iaumajlesi.ac.ir/index/index.php/ee/article/view/1376>
- [66] My Show on My Schedule: How Streaming Has Changed the Game. In: *Facebook IQ* [online]. b.r. [cit. 2018-01-06]. Dostupné z: https://www.facebook.com/iq/articles/my-show-on-my-schedule-how-streaming-has-changed-the-game?ref=wpinsights_rd
- [67] SWENEY, Mark. Ofcom: young people watch a third less TV on sets as they move online. In: *The Guardian: International Edition* [online]. b.r. [cit. 2018-05-16]. Dostupné z: <https://www.theguardian.com/tv-and-radio/2017/jul/07/ofcom-young-people-watch-a-third-less-broadcast-tv-as-they-move-online>
- [68] *Netflix Supported Devices* [online]. 2017 [cit. 2017-12-10]. Dostupné z: <https://devices.netflix.com/en/>
- [69] *Netflix Licensed Content Delivery Specification and Operators Manual* [online]. 8.1. Netflix, Inc., 2017 [cit. 2018-01-06]. Dostupné z: <https://drive.google.com/file/d/0B9DJydDVOVKKLVdCdlF2cFVDVEE/view>
- [70] SUMMERSON, Cameron. How Much Data Does Netflix Use?. In: *How To Geek: We Explain Technology* [online]. b.r. [cit. 2018-05-16]. Dostupné z: <https://www.howtogeek.com/338983/how-much-data-does-netflix-use/>
- [71] Broadcast Requirements. In: *Twitch.tv* [online]. b.r. [cit. 2018-01-06]. Dostupné z: <https://help.twitch.tv/customer/portal/articles/1253460-broadcast-requirements>
- [72] Search for Answers by Device. *Hulu* [online]. b.r. [cit. 2018-01-07]. Dostupné z: https://help.hulu.com/s/all-devices?language=en_US

- [73] SOLSMAN, Joan. Netflix hits 125 million subscribers worldwide. In: *MSN: Money* [online]. Microsoft, b.r. [cit. 2018-05-16]. Dostupné z: <https://www.msn.com/en-gb/money/companies/netflix-hits-125-million-subscribers-worldwide/ar-AAvXq2M>
- [74] Netflix's annual revenue from 2002 to 2017 (in million U.S. dollars). *Statista: The Statistics Portal* [online]. b.r. [cit. 2018-05-16]. Dostupné z: <https://www.statista.com/statistics/272545/annual-revenue-of-netflix/>
- [75] PLAUGIC, Lizzie. Spotify now has 70 million subscribers. In: *The Verge* [online]. Vox Media, b.r. [cit. 2018-05-16]. Dostupné z: <https://www.theverge.com/2018/1/4/16850742/spotify-subscriber-count-70-million-users>
- [76] Spotify's revenue worldwide from 2013 to 2017 (in million euros). *Statista: The Statistics Portal* [online]. b.r. [cit. 2018-05-16]. Dostupné z: <https://www.statista.com/statistics/813713/spotify-revenue/>
- [77] *Twitch Advertising: Audience* [online]. Twitch Interactive, Inc., 2017 [cit. 2018-05-16]. Dostupné z: <http://twitchadvertising.tv/audience/>
- [78] HA, Anthony. Hulu reached more than 17M subscribers and \$1B in ad revenue last year. In: *TechCrunch* [online]. Oath Tech Network, b.r. [cit. 2018-05-16]. Dostupné z: <https://techcrunch.com/2018/01/09/hulu-17m-subscribers/>
- [79] Audio settings. *Spotify* [online]. b.r. [cit. 2018-01-07]. Dostupné z: <https://support.spotify.com/us/article/high-quality-streaming/>
- [80] Broadcasting Guidelines. *Twitch.tv* [online]. b.r. [cit. 2018-01-07]. Dostupné z: <https://stream.twitch.tv/encoding/>
- [81] Video Commercial. *Hulu* [online]. b.r. [cit. 2018-01-07]. Dostupné z: <https://www.hulu.com/advertising/ad-product/video-commercial/>
- [82] What are the Streaming Quality Options for the Broadcasters?. In: *Livestream Customer Care* [online]. b.r. [cit. 2018-05-08]. Dostupné z: <https://help.livestream.com/hc/en-us/articles/212227808-What-are-the-Streaming-Quality-Options-for-the-Broadcasters->
- [83] SIDERIS, A., E. MARKAKIS, N. ZOTOS, E. PALLIS a C. SKIANIS. MPEG-DASH users' QoE: The segment duration effect. *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)* [online]. IEEE,

- 2015, , 1-6 [cit. 2018-05-16]. DOI: 10.1109/QoMEX.2015.7148117. ISBN 978-1-4799-8958-4. Dostupné z: <http://ieeexplore.ieee.org/document/7148117/>
- [84] Image and Video quality assessment – Part One: MSE & PSNR. In: *Emanuele Colucci Personal Home* [online]. b.r. [cit. 2018-05-08]. Dostupné z: <http://emanuelecolucci.com/2011/04/image-and-video-quality-assessment-part-one-mse-psnr/>
- [85] JUNG, Agata. *Comparison of Video Quality Assessment Methods*. Švédsko, 2017. Blekinge Institute of Technology. Vedoucí práce Doc. Benny Löfvström, Dr. Muhammad Shahid.
- [86] ZABROVSKIY, Anatolij, Christian FELDMANN a Christian TIMMERER. Multi-Codec DASH Dataset. In: CESAR, Pablo, Michael ZINK a Niall MURRAY (EDS.). *Proceedings of ACM Multimedia Systems Conference 2018 (MMSys'18)*. Amstrdam, 2018.
- [87] DASH SVC Dataset. *ITEC - Institute of Information Technology* [online]. b.r. [cit. 2018-05-08]. Dostupné z: <http://concert.itec.aau.at/SVCDataset/>
- [88] Using Apache HTTP Server on Microsoft Windows. In: *Apache HTTP Server Project* [online]. The Apache Software Foundation, b.r. [cit. 2018-05-22]. Dostupné z: <https://httpd.apache.org/docs/2.4/platform/windows.html>
- [89] BUCKLER, Craig. How to Install Apache Web Server on Windows. In: *SitePoint: Learn HTML, CSS, JavaScript, PHP, Ruby & Responsive Design* [online]. b.r. [cit. 2018-05-22]. Dostupné z: <https://www.sitepoint.com/how-to-install-apache-on-windows/>
- [90] LEDERER, Stefan. Optimal Adaptive Streaming Formats MPEG-DASH & HLS Segment Length. In: *Bitmovin* [online]. b.r. [cit. 2018-05-16]. Dostupné z: <https://bitmovin.com/mpeg-dash-hls-segment-length/>
- [91] What is MPEG-DASH?. In: *Wolfcrow* [online]. Sareesh Sudhakaran, b.r. [cit. 2018-05-22]. Dostupné z: <https://wolfcrow.com/blog/what-is-mpeg-dash/>

10 Seznam obrázků a tabulek

| | |
|---|----|
| Obrázek 1 Základní proces streamování..... | 3 |
| Obrázek 2 Struktura sdružených dat..... | 10 |
| Obrázek 3 Progresivní download | 11 |
| Obrázek 4 Tradiční streaming | 12 |
| Obrázek 5 Adaptivní streamování - MPEG DASH..... | 17 |
| Obrázek 6 Rozdělení hodnocení kvality videa | 29 |
| Obrázek 7 Schéma realizovaného řetězce..... | 36 |
| Obrázek 8 Výsledek úspěšného testování FFmpeg..... | 38 |
| Obrázek 9 Výsledek úspěšného testování MP4Box | 38 |
| Obrázek 10 DASH Validator | 41 |
| Obrázek 11 Grafy - objektivní hodnocení používaných videosekvencí | 43 |
| Obrázek 12 UPC internetové připojení, Port Forwarding..... | 47 |
| Obrázek 13 Otevřený port 80 | 47 |
| Obrázek 14 Aplikace používající port 80 | 48 |
| Obrázek 15 VLC, aktuální informace o médiu | 49 |
| Obrázek 16 NetBalancer | 50 |
| Obrázek 17 NetBalancer - detail, možné priority | 51 |
| | |
| Tabulka 1 Parametry základních ztrátových video kodeků | 8 |
| Tabulka 2 Shrnutí technických parametrů ztrátových audio kodeků | 9 |
| Tabulka 3 Srovnání řešení pro adaptivní streaming..... | 19 |
| Tabulka 4 Základní uživatelské srovnání streamovacích služeb..... | 34 |
| Tabulka 5 Kodeky streamingových služeb | 35 |
| Tabulka 6 Zvolené parametry výstupních sekvencí..... | 39 |
| Tabulka 7 Hodnocení kvality souborů, shrnutí hodnot | 44 |

11 Přílohy

11.1 Seznam příloh

- Příloha 1: Ukázka programového toku, kódování videa pro MPEG DASH
- Příloha 2: Ukázka programového toku, kódování video pro MPEG DASH
- Příloha 3: Soubor zakovkri-obsah_serveru.rar
- Příloha 4: Soubor zakovkri-matlab.rar
- Příloha 5: Soubor zakovkri-stream-zk.mp4

Příloha 1: Ukázka programového toku, kódování videa pro MPEG DASH

```
Input #0, avi, from 'input.avi':
  Metadata:
    encoder           : MEncoder 2:1.0~rc2-0ubuntu13
  Duration: 00:09:56.46, start: 0.000000, bitrate: 9586 kb/s
    Stream #0:0: Video: msmpeg4v2 (MP42 / 0x3234504D), yuv420p,
    1920x1080, 9328 kb/s, 24 fps, 24 tbr, 24 tbn, 24 tbc
    Stream #0:1: Audio: mp3 (U[0][0][0] / 0x0055), 48000 Hz,
    stereo, s16p, 245 kb/s
  Stream mapping:
    Stream #0:0 -> #0:0 (msmpeg4v2 (native) -> h264 (libx264))
  Press [q] to stop, [?] for help
  [libx264 @ 000001ec0242cdc0] using cpu capabilities: MMX2 SSE2Fast
  SSSE3 SSE4.2 AVX
  [libx264 @ 000001ec0242cdc0] profile High, level 2.1
  [libx264 @ 000001ec0242cdc0] 264 - core 155 r2901 7d0ff22 -
  H.264/MPEG-4 AVC codec - Copyleft 2003-2018 -
  http://www.videolan.org/x264.html - options: cabac=1 ref=3
  deblock=1:0:0 analyse=0x3:0x113 me=hex subme=7 psy=1
  psy_rd=1.00:0.00 mixed_ref=1 me_range=16 chroma_me=1 trellis=1
  8x8dct=1 cqm=0 deadzone=21,11 fast_pskip=1 chroma_qp_offset=-2
  threads=6 lookahead_threads=1 sliced_threads=0 nr=0 decimate=1
  interlaced=0 bluray_compat=0 constrained_intra=0 bframes=3
  b_pyramid=2 b_adapt=1 b_bias=0 direct=1 weightb=1 open_gop=0
  weightp=2 keyint=90 keyint_min=9 scenecut=40 intra_refresh=0
  rc_lookahead=40 rc=abr mbtree=1 bitrate=150 ratetol=1.0
  qcomp=0.60 qpmin=0 qpmax=69 qpstep=4 ip_ratio=1.40 aq=1:1.00
  Output #0, mp4, to 'input_video_480x270_150k.mp4':
  Metadata:
    encoder           : Lavf58.7.100
    Stream #0:0: Video: h264 (libx264) (avc1 / 0x31637661),
    yuv420p, 480x270, q=-1--1, 150 kb/s, 24 fps, 12288 tbn, 24 tbc
  Metadata:
    encoder           : Lavc58.9.100 libx264
  Side data:
    cpb: bitrate max/min/avg: 0/0/150000 buffer size: 0
    vbv_delay: -1
  frame=14315 fps= 93 q=-1.0 Lsize= 10986kB time=00:09:56.33
  bitrate= 150.9kbits/s speed=3.87x
  video:10836kB audio:0kB subtitle:0kB other streams:0kB global
  headers:0kB muxing overhead: 1.386921%
```

```

[libx264 @ 000001ec0242cdc0] frame I:230   Avg QP:25.11   size:
13930
[libx264 @ 000001ec0242cdc0] frame P:5879   Avg QP:29.75   size:
919
[libx264 @ 000001ec0242cdc0] frame B:8206   Avg QP:34.35   size:
303
[libx264 @ 000001ec0242cdc0] consecutive B-frames: 20.1%   8.2%
6.9% 64.9%
[libx264 @ 000001ec0242cdc0] mb I   I16..4: 12.9% 63.5% 23.6%
[libx264 @ 000001ec0242cdc0] mb P   I16..4:   0.8%   2.7%   0.4%
P16..4: 18.9%  5.6%  3.2%  0.0%  0.0%   skip:68.4%
[libx264 @ 000001ec0242cdc0] mb B   I16..4:   0.1%   0.5%   0.1%
B16..8: 20.0%  1.9%   0.4%  direct: 0.5%  skip:76.5%  L0:41.9%
L1:55.2% BI: 3.0%
[libx264 @ 000001ec0242cdc0] final ratefactor: 29.05
[libx264 @ 000001ec0242cdc0]   8x8   transform   intra:66.3%
inter:70.5%
[libx264 @ 000001ec0242cdc0] coded y,uvDC,uvAC intra: 59.8% 68.5%
41.2% inter: 4.6% 3.9% 0.5%
[libx264 @ 000001ec0242cdc0] i16 v,h,dc,p: 25% 31% 10% 33%
[libx264 @ 000001ec0242cdc0] i8  v,h,dc,ddl,ddr,vr,hd,vl,hu: 17%
17% 22%  6%  7%  9%  7%  7%  8%
[libx264 @ 000001ec0242cdc0] i4  v,h,dc,ddl,ddr,vr,hd,vl,hu: 18%
18% 16%  7%  9%  9%  9%  7%  7%
[libx264 @ 000001ec0242cdc0] i8c dc,h,v,p: 56% 22% 15%  7%
[libx264 @ 000001ec0242cdc0] Weighted P-Frames: Y:3.9% UV:2.0%
[libx264 @ 000001ec0242cdc0] ref P L0: 62.1% 16.3% 12.9%  7.9%
0.7%
[libx264 @ 000001ec0242cdc0] ref B L0: 86.0% 10.4%  3.5%
[libx264 @ 000001ec0242cdc0] ref B L1: 94.3%  5.7%
[libx264 @ 000001ec0242cdc0] kb/s:148.81

```


Příloha 2: Ukázka programového toku, kódování audia pro MPEG DASH

```
Input #0, avi, from 'input.avi':
  Metadata:
    encoder          : MEncoder 2:1.0~rc2-0ubuntu13
  Duration: 00:09:56.46, start: 0.000000, bitrate: 9586 kb/s
    Stream #0:0: Video: msmpeg4v2 (MP42 / 0x3234504D), yuv420p,
1920x1080, 9328 kb/s, 24 fps, 24 tbr, 24 tbn, 24 tbc
    Stream #0:1: Audio: mp3 (U[0][0][0] / 0x0055), 48000 Hz,
stereo, s16p, 245 kb/s
Stream mapping:
  Stream #0:1 -> #0:0 (mp3 (native) -> aac (native))
Press [q] to stop, [?] for help
Output #0, mp4, to 'input_audio_48k.mp4':
  Metadata:
    encoder          : Lavf58.7.100
    Stream #0:0: Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz,
stereo, fltp, 48 kb/s
  Metadata:
    encoder          : Lavc58.9.100 aac
size= 3634kB time=00:09:56.43 bitrate= 49.9kbits/s speed=40.5x
video:0kB audio:3524kB subtitle:0kB other streams:0kB global
headers:0kB muxing overhead: 3.122402%
[aac @ 000001b66ae5e0c0] Qavg: 122.585
```

Příloha 3: Soubor zakovkri-obsah_serveru.rar

Soubor obsahující všechna vytvořená data, která jsou publikována na HTTP serveru.

Příloha 4: Soubor zakovkri-matlab.rar

Obsahuje soubory skriptu prostředí MATLAB, jak vysvětleno v 6.1.3.

Příloha 5: Soubor zakovkri-stream-zk.mp4

Videosoubor, nahrávka streamu přijatého VLC přehrávačem.