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

<table>
<thead>
<tr>
<th>Příjmení:</th>
<th>Žáková</th>
<th>Jméno:</th>
<th>Kristýna</th>
<th>Osobní číslo:</th>
<th>456918</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fakulta/ústav:</td>
<td>Fakulta elektrotechnická</td>
<td>Zadávající katedra/ústav:</td>
<td>Katedra radioelektroniky</td>
<td>Studijní program:</td>
<td>Komunikace, multimédia a elektronika</td>
</tr>
<tr>
<td>Studijní obor:</td>
<td>Multimediální technika</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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:
Podívejte se na aktuální technologie pro kompresi a streamingu videa. Analyzujte současné metody pro subjektivní a objektivní hodnocení kvality videa pro různé aplikace. Realizujte model přenosového systému s možností simulace typických artefaktů vznikajících při přenosu. Soustředěte se zejména na použití standardu MPEG-DASH.

Seznam doporučené literatury:

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

prof. Ing. Pavel Ripka, CSc. 
podpis dozorčí(hy) práce

III. PŘEVZETÍ ZADÁNÍ

Studentku bere na vědomí, že je povinen 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 seznámuje s nejvyužívanějšími kompresními standardy a kodeky. Další části přiblížují 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 audio 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 dřív 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 multimedialního obsahu, jelikož v dnešní době dosahuje větší obliby mezi populaci než sledování pořadů v televizí 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 může využívat pasáží napsaných k výstupní dokumentaci tohoto předmětu.

V první kapitole je členěná seznámení s fenoménem streamingu, jeho fungováním a kompresí streamovaného obsahu. Kapitola druhá uvádí způsoby doručování streamovaného obsahu - seznámí 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 s 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 přijímac.

Závěrem jsou nejen zrekapitolovány 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 audio a videa - audiovizuálního obsahu - nicméně je možné přenášet jakákoli 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.) vyžadá jeho přehrání po provedení dekompresie 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](image1)

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 jimí pouze dále jednosměrně šíří k přijímačům (za možnosti obousměrné 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 přijemce má možnost ovládat přehrávání a sledovaný obsah pozastavovat, posouvat, přehrávat znovu, přeskočit na určitý čas záznamu aj. Tyto možnosti v online streamingu nemá, není schopen přenosi 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 vybírání obsahu, jež chtě přehrávat, volí také kvalitu, ve které chtě 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 její sledovat plynule bez přerušení.

**Obsah** lze dátli jako amatérský a volně širitelný (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 velký 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

Běžně 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 pojmů „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 \( \text{kodek} = \text{kodér} + \text{dekodér} \). Při volbě použitého kodeku je třeba zohlednit nejen účinnost komprese, ale také jejich vypočetní 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 nejpoužívanějšíí 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 všeobecněmu využití.


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

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 nástín a aplikace standardů H.26x je popsán například v [8] [9]. Standard **H.261** je využíván zejména v oblasti videokonferencí a videotelefonie v síťích 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 = <1,30>). 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]


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 shromáždí 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 seznamení - 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)
- (Ogg) Theora - opensource kodek, zpočátku srovnávaný s MPEG-4 a dříve využívaný společností Wikipedia.com, v současné době již není aktivně využíván. [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.

<table>
<thead>
<tr>
<th>Vydání</th>
<th>Metoda komprese</th>
<th>Patentováno</th>
<th>Srovnání účinnosti kompresy se věnuje například [15].</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.264</td>
<td>2003</td>
<td>Ztrátová/bezeztrátová</td>
<td>Ano</td>
</tr>
<tr>
<td>H.265</td>
<td>2013</td>
<td>Ztrátová/bezeztrátová</td>
<td>Ano</td>
</tr>
<tr>
<td>VP8</td>
<td>2008</td>
<td>Ztrátová</td>
<td>Ano</td>
</tr>
<tr>
<td>Theora</td>
<td>2002</td>
<td>Ztrátová</td>
<td>Ano, volně licencováno</td>
</tr>
</tbody>
</table>

*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 nejpopulárnější 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 standartně 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 - bezeztrátový kodek, využívaný zejména pokud je nahrána/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).


<table>
<thead>
<tr>
<th>Vzorkovací frekvence</th>
<th>Bitrate</th>
<th>Konstantní bitrate</th>
<th>Proměnný bitrate</th>
<th>Stereo</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPEG 1 - L3 MP3</td>
<td>8-48 kHz</td>
<td>8-32 kb/s</td>
<td>ano</td>
<td>ano</td>
</tr>
<tr>
<td>MPEG 2 - AAC</td>
<td>8-192 kHz</td>
<td>8-529 kb/s</td>
<td>ano</td>
<td>ano</td>
</tr>
<tr>
<td>Vorbis</td>
<td>1 Hz - 200 kHz</td>
<td>proměnný</td>
<td>ano</td>
<td>ano</td>
</tr>
</tbody>
</table>

*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].


**CONTAINER formát souboru:**
- .mp4 (MPEG), .avi (Windows), .flv (Adobe), .mov (QuickTime), .ogg (Xiph.org)...

**VIDEO DATA**
zakódovaná kodeky: H.264, AVC, H.265 HEVC, MPEG-2...

**AUDIO DATA**
zakódovaná kodeky: AAC, Vorbis, WMA...

**METADATA, TITULKY, VIDEO DESCRIPTION**
apod.

*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

![Diagram progresivního downloadu]

**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 platform.

Z pohledu cílového diváka je sympatická možnost sledovat obsah před jeho kompletním stažením, avšak už není umožněno přeskakovat 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 plnule přehráno - může se jim tedy stát, že dojde k zaseknutí obrazu, načítání a celkovému diskomfortu při sledování.

11
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. Doplňuje 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 úrovni 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 jakouži 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 aplikací Flash (ale i v jiných softwarových vybaveních) a je schopen pracovat se všemi multimedialními formáty. [24]

Multimedialní 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 zasílán 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 přehrávač 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 přijímaci 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 silněho 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 přehrávat. Poté zaše 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 kódované nezávisle na sobě, bez mezer či překryvů, pro uživatele je výsledný sledovaný obraz kerozeznání 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 platform a společností, jako jsou Microsoft (SS - Smooth Streaming), Apple (HLS - HTTP Live Streaming), nebo Adobe (HDS - HTTP Dynamic Streaming).

Benefity pro adaptivní streaming se dá uvést několik:

- na rozdíl od přenosu na protokolu RTP nemívá 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 rychlě 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 zmínované 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 neatráktivní.

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.

2.4.1 Princip MPEG DASH

Obrázek 5 nastiňuje jednoduchý scénář streamingu mezi serverem a přijímačem.
Doručování obsahu probíhá velmi obdobně jako v dříve zmiňovaných protokolech adaptivního streamingu - po malých částech, které jsou na straně přijímací „složeny“ a sledovány postupně jako celek. Jedná se o novou „odnož“ adaptivního streamování, tedy zachovává také přizpůsobování kvality obsahu přijímací.

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 přeh-rání 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 příkazu 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 clientské straně.
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 délku 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, lišící se v bitratu, rozlišení, použitím kodeku, počtu kanálů, v případě audio 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 jakéhokoliv z nich, aby maximálně využil přenosového pásma;
- MPD může být rozdělen do částí a stahován zvlášť, může odkazovat na externí část;
- klient může serveru poskytnout zpětnou vazbu ohledně kvality přenosu - standard přesně nastavuje hodnotící parametry;
- přepínavání mezi jednotlivými streamy - MPD umožňuje klientovi volit/přepíнат 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 nepříliš 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.

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

*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 zázitku přijemce. 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, fluktuací kvality komunikačního kanálu.

V telekomunikačních sítích je QoS (Quality of Service, „kvalita služeb“) vyjadřována objektivně sítovými 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 zázitku však nijak nezaručují. Mnohem důležitější je sledovat a vyhodnotit kvalitu zázitku 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. Základním, 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 hodnoticího 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 (štáří, 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. video) 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 dekódovaného 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 dekódování 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 paměti 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 - tlamí 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 několik sekundové) 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 délku 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 snížuje 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 latency) 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 znatelně 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 používaném 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 kvalitou přehrávání aktuálnímu stavu sítě a dle [35] může snížit pozastavení videa až o 80%.

Ze závěru studie [36] lze poznamenat, že je příspěvky 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. Ačkoli způsob 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 kodem, jak již bylo zmíněno, jiné způsoby řešení adaptivního streamingu tuto možnost nemají. Existují však systémy, které podporují pouze specifická kódování - v současné době domínuje 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 multimedialní 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 kodeuru 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].

24
3.3 Artefakty

Při sledování multimedialního 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 multimedialního obsahu jsou artefakty vzniklé v obrazové složce mnohem dominantnější než ty ve složce zvukové. Neznamená 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, překryvání 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 linii, 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 „splynutí“ 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 (rozmažání, 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, moskytí nebo moskytí křídla (z angl. „mosquito wings“) [43] [45]

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

25
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


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 neoptimalních 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 (měřícího) prostředí a podmínky pro průběh subjektivního testování (například maximální pozorovací úhel, špičkový jas, okolní osvětlení, a další).

**DSIS**

**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 spojité 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**
SCQE

SDSCE
S malým rozdílem byla z metody SCQE vyvinuta Simultaneous Double Stimulus Continuous Evaluation (SDSCE). Je navrhována na porovnání věrnosti obrazu mezi dvěma sekvencemi v reálném čase, zmínovaný 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 úrovne 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 prezentovány 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 rozoznat jemné kvalitativní rozdíly.

4.2 Objektivní metody hodnocení kvality videa

Automatické objektivní metody hodnocení kvality videa probíhají bez účasti pozorovatele, jedná se o algoritrické určení kvality obrazu (videa). Publikace [46] shrnuje objektivní 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 zkršleného 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 zkršlenému 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.

<table>
<thead>
<tr>
<th></th>
<th>subjektivní</th>
<th>objektivní</th>
</tr>
</thead>
<tbody>
<tr>
<td>hodnocení kvality</td>
<td>„Double Stimulus“</td>
<td>matematické metriky</td>
</tr>
<tr>
<td>hodnocení poruch</td>
<td>„Single Stimulus“</td>
<td>(např. PSNR, MSE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>metriky založené na vlastnostech lidského vizuálního systému</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ostatní metriky (např. SSIM)</td>
</tr>
</tbody>
</table>

Obrázek 6 Rozdílení 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 přesné vnímání zkršlení a kvality lidským vizuálním
systémem. 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 nezkreslený obraz je možné vnímat jako součet zkresleného 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 zkreslené obrazy se stejným MSE však mohou mít odlišné chyby - méně či více viditelné. [56]

Dle [57] se vypočítá následovně:

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

(1)

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

Odmočninou 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, nebo 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 kompresní ztrátových kodeků. V tomto případě jsou původní data signálem a chyba komprese šumem.

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

\[
PSNR = 10 \times \log_{10} \frac{m^2}{MSE}
\]  

[dB]  

(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 nespecializuje 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 článcích - zdá se, jsou jimi ve velké míře považovány za základní (výchozi) 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 uživatelé spojují spíše s kontextem (tedy sledování „tradičního vysílání“ s rodinou či přáteli), streaming je asociován spíše s obsahem, který cíleně využívá a jehož aspekty jsou pro ně při trávení času atraktivní. Je však pravdou, že streaming je preferovanou platformou zejména pro milenály (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 mediálních 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 pronájmem nosičů s mediálním obsahem, avšak roku 2007 vstoupila na trh streamování mediálního 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 laptopesty, chytré televíze. [68]

Kodeky využívané Netflixem se liší v závislosti na cílovém zařízení, na kterém přijímací 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 kódu 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 laptopench, herních konzolích.
V současné době využívá ztrátového kóduku 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 jakýchkoli poplatků, avšak při úhradě měsíčního předplatného je divákovi umožněno vstupovat do diskuzí, finančně přispívat autorům sledovaných videí, streamovat.
Platforma využívá kódeků H.264 a AAC, doporučovaný 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 laptopsy. [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 doplňení některých informací je v následujících tabulkách shrnuto několik hlavních informací o jednotlivých službách, případně jejich srovnání (Tabulka 4 a 5).

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

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.

<table>
<thead>
<tr>
<th>Streamovací služba</th>
<th>Využívaný video kodek</th>
<th>Využívaný audio kodek</th>
</tr>
</thead>
<tbody>
<tr>
<td>Netflix [69]</td>
<td>H.264, H.265, VC-1</td>
<td>Ogg Vorbis, AAC</td>
</tr>
<tr>
<td>Spotify [79]</td>
<td>---</td>
<td>Ogg Vorbis, AAC</td>
</tr>
<tr>
<td>Twitch.tv [71] [80]</td>
<td>H.264</td>
<td>AAC</td>
</tr>
<tr>
<td>Hulu [81]</td>
<td>VP6, H.264</td>
<td>AAC, PCM</td>
</tr>
</tbody>
</table>

*Tabulka 5 Kodeky streamingových služeb*
6 Streamingový proces

Na základě znalostí nabytých v předchozích kapitolách je možné nastínit 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 se zaměřovala 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 Commons\(^\text{10}\). Tento snímek byl vybrán nejen z důvodu krátké stopáže, ale také díky četnosti jeho využívání v testech, měřeních, přenesech a další práce s videosoubory, celosvětově vykonávaných a publikovaných.

\(^{10}\) Big Buck Bunny https://peach.blender.org/
6.1 Příprava souboru k přenosu DASH
Po výběru souboru je třeba jej před přenosem ošetřit kodem a předzpracovat tak, aby byl k dispozici funkční MPD soubor\textsuperscript{11} (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 softwarom pro nahrávání, konverzi a streaming digitálního obsahu. Je primárně vyvíjen pro systém Linux, ale je možné jej zkompilovat také pod Windows a většinu jiných operačních systémů.
FFmpeg obsahuje databázi knihoven – mimo jiné například využitě *libavcodec* (audio/video kodéry a dekodéry), *libavformat* (multiplexery a demultiplexery pro kontejnerové audio/video formáty), *libsuscale* (změna rozlišení a barevného modelu videa), *libavutil* (pro pomocné operace), *libswresample* a *libavresample* (převzorkování a změna formátu audio/videa) a jiné.
Instalační soubor je veřejně dostupný například na oficiálních stránkách FFmpeg\textsuperscript{12} nebo FFmpeg Builds\textsuperscript{13}.

**GPAC**
GPAC je open source program pro práci se soubory standardu MPEG-4. Jeho základní využívánou knihovnou příkazů je *libgpac*, umožňující práci se servery a multiplexory (pro streaming), multimedijním přehrávačem a příkazovým řádkem a *MP4box* pro práci se soubory.
Instalační soubor ke stažení například na oficiálních stránkách GPAC - sekce Downloads\textsuperscript{14}.

\textsuperscript{11} Často se pro MPD soubor využívá také pojm „manifest“. [91]
\textsuperscript{12} FFmpeg https://www.ffmpeg.org/
\textsuperscript{13} Builds - Zeranoe FFmpeg https://ffmpeg.zeranoe.com/builds/
\textsuperscript{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 software (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 9\(^{15}\) - může se měrně lišit v závislosti na instalované verzi):

\(^{15}\) Pro lepší čitelnost byly barvy v Obrázcích 8 a 9 invertovány.
Následuje samotné zpracování snímku Big Buck Bunny do několika výstupních sekvencí tak, aby cílový MPEG DASH přehravač 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).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobilní</td>
<td>480×270</td>
<td>150</td>
<td>48</td>
</tr>
<tr>
<td>Normální</td>
<td>512×288</td>
<td>350</td>
<td>96</td>
</tr>
<tr>
<td>Střední</td>
<td>768×432</td>
<td>550</td>
<td>128</td>
</tr>
<tr>
<td>Vysoká</td>
<td>848×480</td>
<td>1500</td>
<td>128</td>
</tr>
<tr>
<td>HD</td>
<td>1280×720</td>
<td>2000</td>
<td>128</td>
</tr>
<tr>
<td>HD+</td>
<td>1920×1080</td>
<td>5000</td>
<td>192</td>
</tr>
</tbody>
</table>

*Tabulka 6 Zvolené parametry výstupních sekvencí*

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

```bash
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 zvláště)
- `input_video_480x270_150k.mp4` - označuje název výstupního souboru

Ukázka samotného programového toku je ke zhlédnutí viz Příloha 1.

Kódování audio pak probíhá obdobně - s příkazem:

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

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

39
-ca 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 části 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]
- **-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 zajišťovací 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 (nahradit na server, 6.3) - a to kromě vstupního souboru. Příloha 3 dává tyto soubory k nahlédnutí.

---

[83] Zhodnocení závislosti délky segmentu na výsledné kvalitě sledovaného streamu se věnuje například [90].
6.1.2.1 Kontrola MPD


![DASH Validator](image)

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 předběžné instalace doporučeného softwaru (6.1.1) a jejich přidání do Path systémově/uživatelské proměnné použitého zařízení (Windows, Vlastnosti systému-Upřesnit, 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 (zakovkři-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 šířitelných 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.

---

Průměrné naměřené PSNR pro vstupní videosekvenci je 37.26 dB, pro zpracovanou videosekvenci pak 36.84 dB. Mírně horší výsledek pro druhý případ je očekávatevný. 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:

<table>
<thead>
<tr>
<th>Vstupní videosekvence</th>
<th>PSNR [dB]</th>
<th>MSE [-]</th>
<th>SSIM [-]</th>
<th>VQM [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>min</td>
<td>30.4168</td>
<td>0.0</td>
<td>0.9766</td>
<td>0.0</td>
</tr>
<tr>
<td>max</td>
<td>100</td>
<td>12.4767</td>
<td>1</td>
<td>2.2773</td>
</tr>
<tr>
<td>průměr</td>
<td>37.25941</td>
<td>1.93086</td>
<td>0.99679</td>
<td>0.35281</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Zpracovaná videosekvence</th>
<th>PSNR [dB]</th>
<th>MSE [-]</th>
<th>SSIM [-]</th>
<th>VQM [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>min</td>
<td>30.3844</td>
<td>0.0</td>
<td>0.9741</td>
<td>0.0</td>
</tr>
<tr>
<td>max</td>
<td>100</td>
<td>12.5441</td>
<td>1</td>
<td>2.2260</td>
</tr>
<tr>
<td>průměr</td>
<td>36.84127</td>
<td>2.16086</td>
<td>0.99618</td>
<td>0.37477</td>
</tr>
</tbody>
</table>

**Tabulka 7** Hodnocení kvality souborů, shrnutí hodnot

6.2 Alternativní možnost získání dat

V případě, že není žádoucí k přenosu zpracovávat vlastní videosequency, 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 zmíněných stránkách Xipg.org.

- dataset z roku 2014, autor: Florian Bacher\(^{20}\)
- dataset z roku 2018, pracující s několika kódeky\(^{21}\) [86]
- SVC dataset vytvořený ve spolupráci s ITEC\(^{22}\) [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 přístupovat.

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

Možností je více, jistě se nabízí varianty webhostingu - uvádíme-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 v provést změny v konfiguračním souboru httpd.conf, a to:

```
<Directory "celá adresa kořenové složky">
  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:

```xml
<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 `httpd.conf` souboru:

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

nebo přímým připsáním do souboru `mime.types`:

```conf
...
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 (ktéřý protokol HTTP implicitně používá).

![Obrázek 12 UPC internetové připojení, Port Forwarding](image)

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

![Port Forwarding Tester](image)

A jak je tedy patrné, jednotlivé soubory adresujeme pomocí externí IP adresy počítače, na kterém Apache Server běží.\(^\text{26}\)

---

\(^\text{25}\) Open Port Check Tool https://www.yougetsignal.com/tools/open-ports/

\(^\text{26}\) 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/
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 Bitmovin. Pro čisté sledování změn, parametrů a artefaktů streamu existuje DASH IF Reference Client, ve kterém je možné sledovat parametry testovacích streamů (bitrate, ztracené snímky, velikost vyrovnaní paměti aj). Tento přehrávač však nepodporuje některé adresy, a je proto pro rozšířené využití (měření, sledování změn, zpracovávání informací) méně 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 Přehrávat. 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 médii → Statistiky), kde je monitorováno několik parametrů, které lze vidět na následujícím obrázku.

---

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

ktorý 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 nejednodušším 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 jiter a jiné chybové parametry přenosu. Z volně dostupných byly vyzkoušeny programy NetLimiter 4.0.35\(^{20}\) a NetBalancer 9.12.2\(^{30}\). 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).

\[\text{Obrázek 16 NetBalancer}\]

\[\text{\textsuperscript{20} ke stažení na https://www.slunecnice.cz/sw/netlimiter/}\]

\[\text{\textsuperscript{30} ke stažení na https://www.slunecnice.cz/sw/netbalancer/}\]
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 znatelnější. 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.


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 směrodatné 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 metodí pro objektivní hodnocení provedeno se sevřenemi 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 - prohlubení přehledu aktuálního trhu se streamingovými službami, vyhlídky do budoucně, či provedení hodnocení získaných a přehráných 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

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3GPP</td>
<td>Third Generation Partnership Project</td>
</tr>
<tr>
<td>AAC</td>
<td>Advanced Audio Coding</td>
</tr>
<tr>
<td>ACR</td>
<td>Absolute Category Rating</td>
</tr>
<tr>
<td>AS</td>
<td>Adaptation Set</td>
</tr>
<tr>
<td>AVC</td>
<td>Advanced Video Coding</td>
</tr>
<tr>
<td>BMFF</td>
<td>Base Media File Format</td>
</tr>
<tr>
<td>CDN</td>
<td>Content Delivery Networks</td>
</tr>
<tr>
<td>CORS</td>
<td>Cross-Origin Resource Sharing</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>DASH</td>
<td>Dynamic Adaptive Streaming over HTTP</td>
</tr>
<tr>
<td>DCR</td>
<td>Degradation Category Rating</td>
</tr>
<tr>
<td>DSCQS</td>
<td>Double Stimulus Continuous Quality Scale</td>
</tr>
<tr>
<td>DSIS</td>
<td>Double Stimulus Impairment Scale</td>
</tr>
<tr>
<td>DVB-T</td>
<td>Digital Video Broadcasting - Terrestrial</td>
</tr>
<tr>
<td>FR</td>
<td>Full-Reference</td>
</tr>
<tr>
<td>GOP</td>
<td>Group of Pictures</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphic User Interface</td>
</tr>
<tr>
<td>HDS</td>
<td>HTTP Dynamic Streaming</td>
</tr>
<tr>
<td>HDTV</td>
<td>High-Definition Television</td>
</tr>
<tr>
<td>HEVC</td>
<td>High Efficiency Video Coding</td>
</tr>
<tr>
<td>HLS</td>
<td>HTTP Live Streaming</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>IEC</td>
<td>International Electrotechnical Commision</td>
</tr>
<tr>
<td>ISDN</td>
<td>Integrated Services Digital Network</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>ITU</td>
<td>International Telecommunication Union</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>M2TS</td>
<td>MPEG-2 Transport Stream</td>
</tr>
<tr>
<td>MIME</td>
<td>Multipurpose Internet Mail Extensions</td>
</tr>
<tr>
<td>MPD</td>
<td>Media Presentation Description</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Pictures Experts Group</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error</td>
</tr>
</tbody>
</table>
MSU Michigan State University
NR No-Reference
OIPF The Open IPTV Forum
PC Pair Comparision
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
SC Stimulus Comparision
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
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


[26] Information technology -- Dynamic adaptive streaming over HTTP (DASH): Part 1: Media presentation description and segment formats. ISO -


[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/IH-


[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


64


[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,


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 streamovacích 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 video 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

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:
    cbpb: 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 audio pro MPEG DASH

<table>
<thead>
<tr>
<th>Input #0, avi, from 'input.avi':</th>
<th>Metadata:</th>
</tr>
</thead>
<tbody>
<tr>
<td>encoder : MEncoder 2:1.0-rc2-0ubuntu13</td>
<td>Duration: 00:09:56.46, start: 0.000000, bitrate: 9586 kb/s</td>
</tr>
<tr>
<td>Stream #0:0: Video: msmpeg4v2 (MP42 / 0x3234504D), yuv420p, 1920x1080, 9328 kb/s, 24 fps, 24 tbr, 24 tbn, 24 tbc</td>
<td>Stream #0:0: Audio: mp3 (U[0][0][0] / 0x0055), 48000 Hz, stereo, sl6p, 245 kb/s</td>
</tr>
<tr>
<td>Stream mapping:</td>
<td>Stream mapping:</td>
</tr>
<tr>
<td>Stream #0:1 -&gt; #0:0 (mp3 (native) -&gt; aac (native))</td>
<td></td>
</tr>
<tr>
<td>Press [q] to stop, [?] for help</td>
<td></td>
</tr>
<tr>
<td>Output #0, mp4, to 'input_audio_48k.mp4':</td>
<td>Metadata:</td>
</tr>
<tr>
<td>encoder : Lavf58.7.100</td>
<td>Stream #0:0: Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, stereo, fltp, 48 kb/s</td>
</tr>
<tr>
<td>Metadata:</td>
<td>Metadata:</td>
</tr>
<tr>
<td>encoder : Lavc58.9.100 aac</td>
<td></td>
</tr>
<tr>
<td>size= 3634kB time=00:09:56.43 bitrate= 49.9kbits/s speed=40.5x</td>
<td>[aac @ 000001b66ae5e0c0] Qavg: 122.585</td>
</tr>
</tbody>
</table>

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.