



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY**

**A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV TELEKOMUNIKACÍ**

DEPARTMENT OF TELECOMMUNICATIONS

## WEBOVÁ APLIKACE PRO SUBJEKTIVNÍ HODNOCENÍ STREAMOVÁNÉHO VIDEOA

WEB APPLICATION FOR SUBJECTIVE EVALUATION OF STREAMED VIDEO

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Matúš Lipa**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Dominik Kováč**

**BRNO 2016**



# Bakalářská práce

bakalářský studijní obor **Teleinformatika**  
Ústav telekomunikací

**Student:** Matúš Lipa

**ID:** 167861

**Ročník:** 3

**Akademický rok:** 2015/16

## NÁZEV TÉMATU:

### Webová aplikace pro subjektivní hodnocení streamovaného videa

#### POKyny PRO VYPRACOVÁNÍ:

V poslední době výrazně stoupá popularita streamování. Služby jako YouTube nebo Netflix mají dominantní podíl v mobilním provozu, a to hlavně z důvodu, že streamování videa vyžaduje vyšší přenosovou propustnost než jiný typ obsahu na Internetu. V současnosti se používají různé techniky adaptivního streamování při kterých dochází ke snížení kvality streamovaného videa. Bakalářská práce proto bude zaměřena na techniky adaptivního streamování přes HTTP provoz. Cílem práce bude vytvořit webovou aplikaci, která bude umožňovat subjektivní hodnocení kvality zážitku přehrávaného videa, tak aby bylo možné přednastavit průběh streamování videa (změna kvality videa, délka počátečního načítání, přerušování atd.).

#### DOPORUČENÁ LITERATURA:

[1] ISO/IEC 23009-1. Information technology — Dynamic adaptive streaming over HTTP (DASH): Media presentation description and segment formats. Second edition. Published in Switzerland: ISO copyright office, 2014.

[2] KEITH, Jeremy. HTML5 for web designers. New York: A Book Apart, 2010. ISBN 978-0-9844425-0-8.

[3] STOK, Peter van der. Dynamic and robust streaming in and between connected consumer-electronic devices. New York: Springer, c2005, ix, 239 p. Philips Research, v.3. ISBN 9781402034541.

**Termín zadání:** 1.2.2016

**Termín odevzdání:** 1.6.2016

**Vedoucí práce:** Ing. Dominik Kováč

**Konzultant bakalářské práce:**

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

#### UPOZORNĚNÍ:

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

## ABSTRAKT

Táto práca sa zaoberá vytvorením webovej aplikácie pre subjektívne hodnotenie streamovaného videa a častí k tomu potrebných. Popísané sú techniky streamovania od jednoduchých klasických techník až po adaptívne techniky. Zameriava sa na adaptívne streamovanie pomocou techniky MPEG - DASH a úpravu zdroja videa pomocou nástroja FFmpeg. Pomocou JavaScriptu je vytvorený prehrávač, ktorý umožňuje simulovať zmeny v sieti ako zmena dostupnej šírky pásma. Je možné vytvoriť scenáre, podľa ktorých prehrávač zvolí kvalitu prehrávaného videa. Prehrávač je umiestnený do komplexného nástroja pre hodnotenie kvality zážitku.

## KLÚČOVÉ SLOVÁ

Streamovanie, Progresívny download, Adaptívne streamovanie, MPEG-DASH, Dash prehrávač, FFmpeg, QOE, QOS, priepustnosť, kvalita, streamovanie, hodnotenie

## ABSTRACT

This work deals with creating web application for subjective evaluation of streamed video and parts which are necessary. Streaming techniques are described from simple classical technique to adaptive bitrate streaming techniques. This work focus on adaptive bitrate streaming technology using MPEG - DASH and editing video source with tool FFmpeg. With using JavaScript is create a player who's able to simulate changes in the network as a change in available bandwidth. It's possible to create scenarios under which the player selects the video playback quality. The player is placed into a complex tool for evaluation of quality of experience.

## KEYWORDS

Streaming, Adaptive bitrate streaming, MPEG - DASH, Dash player, Progressive download, FFmpeg, QOE, QOS, throughput, quality, streaming, evaluation

LIPA, Matúš *Webová aplikace pro subjektivní hodnocení streamovaného videa*: bakalárska práca. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 55 s. Vedúci práce bol Ing. Dominik Kováč

## VYHLÁSENIE

Vyhlasujem, že som svoju bakalársku prácu na tému „Webová aplikácie pro subjektivní hodnocení streamovaného videa“ vypracoval(a) samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor(ka) uvedenej bakalárskej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil(a) autorské práva tretích osôb, najmä som nezasiahol(-la) nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý(-á) následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka Českej republiky č. 40/2009 Sb.

Brno .....

.....

podpis autora(-ky)

## POĎAKOVANIE

Rád by som poďakoval vedúcemu bakalárskej práce pánovi Ing. Dominikovi Kováčovi za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Brno .....

.....

podpis autora(-ky)



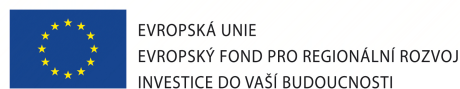
Faculty of Electrical Engineering  
and Communication  
Brno University of Technology  
Purkynova 118, CZ-61200 Brno  
Czech Republic  
<http://www.six.feec.vutbr.cz>

## POĎAKOVANIE

Výzkum popsaný v tejto bakalárskej práci bol realizovaný v laboratóriách podporených projektom SIX; registračné číslo CZ.1.05/2.1.00/03.0072, operačný program Výzkum a vývoj pro inovace.

Brno .....

.....  
podpis autora(-ky)



# OBSAH

Úvod	11
<b>1 Streamovanie multimedialneho obsahu</b>	<b>12</b>
1.1 Metódy Streamovania . . . . .	12
1.1.1 Tradičný streaming . . . . .	13
1.1.2 Progresívny download . . . . .	14
1.1.3 Adaptívne streamovanie založene na HTTP . . . . .	15
<b>2 Prehrávače DASH</b>	<b>21</b>
2.1 JW player . . . . .	21
2.2 Video.js . . . . .	21
2.3 MediaElement.js . . . . .	22
2.4 jPlayer . . . . .	23
2.5 Dash.js . . . . .	23
<b>3 Úprava videa</b>	<b>24</b>
3.1 Sony Vegas Pro . . . . .	24
3.2 Avidemux . . . . .	24
3.3 Nástroj FFmpeg . . . . .	25
<b>4 Meranie kvality zo zážitku</b>	<b>27</b>
4.1 Faktory vplývajúce na meranie QoE . . . . .	27
4.2 Metody merania kvality zo zážitku . . . . .	29
4.2.1 Objektívne hodnotenie QoE . . . . .	29
4.2.2 Subjektívne hodnotenie QoE . . . . .	31
4.3 Meranie QoE pri streamovaní videa . . . . .	31
<b>5 Vytvorenie webovej stránky s prehrávačom MPEG – DASH</b>	<b>32</b>
5.1 Príprava streamovaného videa . . . . .	32
5.2 Implementácia vytvoreného prehrávača na webovú stránku . . . . .	34
<b>6 Vytvorenie prehrávača na meranie QoE</b>	<b>37</b>
6.1 Príprava potrebných dát . . . . .	37
6.2 Inštalácia potrebných balíčkov pre vytvorenie prehrávača . . . . .	37
6.3 Vytvorenie prehrávača a súborov k tomu potrebných . . . . .	38

<b>Záver</b>	<b>44</b>
<b>Literatúra</b>	<b>45</b>
<b>Zoznam symbolov, veličín a skratiek</b>	<b>48</b>
<b>Zoznam príloh</b>	<b>49</b>
<b>A Jednotlivé časti vytvoreného prehrávača</b>	<b>50</b>
<b>B Obsah priloženého CD</b>	<b>55</b>



# ZOZNAM OBRÁZKOV

1.1	Tradičná metóda streamovania . . . . .	13
1.2	Progresívne sťahovanie . . . . .	14
1.3	Metóda MPEG – DASH . . . . .	17
1.4	Metóda adaptívneho streamovania od Apple . . . . .	18
1.5	Metóda streamovania Microsoft Smooth . . . . .	19
1.6	Metoda od spoločnosti Adobe . . . . .	20
2.1	Prehrávač JW Player . . . . .	21
2.2	Prehrávač Video.js . . . . .	22
2.3	Prehrávač Mediaelement.js . . . . .	22
2.4	Prehrávač jPlayer . . . . .	23
2.5	Prehrávač Dash.js . . . . .	23
3.1	Program Sony Vegas Pro . . . . .	25
3.2	Program Avidemux . . . . .	25
3.3	Nástroj FFmpeg . . . . .	26
4.1	Faktory vplývajúce na QoE . . . . .	29
5.1	Obraz prehrávaného videa v malom rozlíšení . . . . .	36
5.2	Obraz prehrávaného videa vo veľkom rozlíšení . . . . .	36
6.1	Bloková schéma prehrávača . . . . .	39

## ZOZNAM VÝPISOV

A.1	Nastavenie cesty k potrebným skriptom . . . . .	50
A.2	Súbor style.css . . . . .	50
A.3	Loading Spinner . . . . .	50
A.4	Načítanie JSON súboru . . . . .	52
A.5	Definovanie prehrávača . . . . .	53
A.6	Nastavenie zdrojov videa . . . . .	53
A.7	Funkcia na prepínanie kvality . . . . .	54

# ÚVOD

V posledných rokoch sa stáva čoraz viac populárnym streamovanie. Vzhľadom k narastajúcim požiadavkam na streamovanie sa kladú väčšie nároky na priepustnosť siete, kvôli tomu vznikli techniky adaptívneho streamovania ktorým sa venuje aj táto práca. Táto práca bude zameraná na techniky adaptívneho streamovania cez HTTP (Hypertext Transfer Protokol), konkrétne techniku MPEG – DASH. Cieľom je vytvoriť webovú aplikáciu na streamovanie pomocou zmienenej techniky MPEG – DASH.

V prvej časti práce sú popísané a vysvetlené techniky streamovania obsahu od klasického streamingu postupne až po jednotlivé techniky adaptívneho streamovania cez HTTP.

Druhá časť práce sa zaoberá prehrávačmi adaptívneho streamovania a výhodami jednotlivých prehrávačov.

Tretia časť práce sa venuje úprave videa a nástrojom na úpravu videa. Je tu popísané čo všetko je možné rozumieť pod úpravou videa, čo je potrebné upraviť pre adaptívne streamovanie, a nástroj FFmpeg ktorým je možné upravovať video.

Ďalšia časť sa zaoberá meraním kvality zo zážitku. V tejto časti je vysvetlené prečo sa používa meranie kvality zážitku, ako súvisí s kvalitou služieb, a aké faktory vplyvajú na meranie. Podrobnejšie sú popísané metódy merania kvality zážitku.

Predposledná časť práce sa zaoberá praktickou prípravou videa na adaptívne streamovanie pomocou nástroja FFmpeg, vytvorením prehrávača a webovej stránky na ktorej bude možné sledovať obsah pomocou adaptívneho streamovania s technikou MPEG –DASH.

Posledná časť práce sa zaoberá vytvorením prehrávača pre nástroj na meranie kvality zážitku. Je tu popísané vytvorenie prehrávača ktorý bude umiestnený do nástroja na meranie kvality zážitku a použitie už pripravených zdrojov tak, aby mohol podľa pred-pripravených scenárov prehrávať video.

# 1 STREAMOVANIE MULTIMEDIÁLNEHO OBSAHU

Streamovanie pochádza s anglického slova „Stream“ čo v preklade znamená prúd alebo prúdenie. Môžeme si to teda predstaviť ako dátový tok, ktorým k užívateľovi prúdia rôzne záznamy. Za streaming sa považuje prenos audiovizuálneho obsahu. Zdrojom streamu môže byť počítač, alebo iný zdroj videa so sieťovým pripojením. Na strane klienta je dôležité zobrazovacie zariadenie (počítač, mobil, tablet), ktorým je možno sledovať streamovaný obsah [5].

Streaming môžeme rozdeliť do jednotlivých základných kategórií [5] [15]:

- Podľa obsahu – môžeme prenášať audio, to je využívané najmä v šírení rádiového vysielania. Alebo je šírený audiovizuálny obsah.
- Podľa dostupnosti – sa stream rozdeľuje na dve skupiny, podľa toho či klienti môžu vidieť len počas vysielania „offline“ streaming alebo kedykoľvek „online“ streaming.
- Podľa premenlivosti bitového toku – rozlišuje sa neadaptívny a adaptívny streaming. Neadaptívny streaming, ktorý sa v dnešnej dobe využíva najmä pri streamovaní audia. Neadaptívny streaming znamená, že klient si pri vyberaní obsahu zvolí kvalitu v ktorej bude sledovať obsah. Niekedy si samotný prehrávač otestuje priepustnosť linky a kvalitu automaticky zvolí. Kvalita je väčšinou identifikovaná bitovým tokom alebo rozlíšením. Väčšie rozlíšenie znamená okrem vyššej kvality aj vyššiu náročnosť na priepustnosť dátovej linky. Podstatou neadaptívneho streamingu je to, že akonáhle začne streamovanie jeho kvalita sa v priebehu nezmení. Adaptívny streaming oproti neadaptívnemu využíva aktuálnu priepustnosť siete a vyťaženosť procesora a sám v prípade zníženia dátovej priepustnosti zníži aj kvalitu videa tak, aby sa stream nezastavil. To znamená, že ak klient sleduje stream vo vysokej kvalite a niekto v sieti obmedzí priepustnosť, tak prehrávač tuto situáciu vyhodnotí a sám prepne kvalitu tak, aby klient mohol sledovať obsah bez výpadku. Tento typ streamovania využívajú aj služby ako YouTube alebo Netflix.

## 1.1 Metódy Streamovania

Metódy streamingu sa môžu líšiť na základe použitej technológie a prehrávača, ktorý používame. V dnešnej dobe sa používajú tri hlavné metódy:

- Tradičný streaming.
- Progresívny download.
- Adaptívny streaming.

### 1.1.1 Tradičný streaming

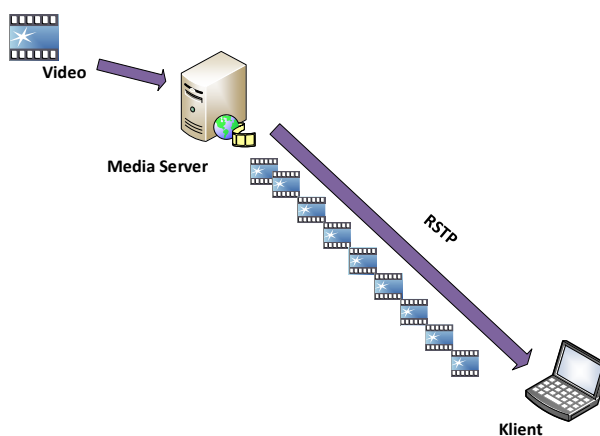
Vhodnými príkladmi tradičných streamingových protokolov sú napríklad RTSP (Real Time Streaming Protocol), RTMP (Real Time Messaging Protocol) a MMS (Microsoft Media Server), ktoré sú definované ako stavové protokoly, čo znamená, že klient je k streamingovému serveru pripojený od začiatku do konca prenosu.

Tieto protokoly využívajú dva druhy spojenia – jeden pre doručovanie video obsahu, druhý pre kontrolné spojenie medzi prehrávačom a serverom. Klient môže vďaka kontrolnému pripojeniu využiť funkcie PLAY, PAUSE alebo ukončiť spojenie.

Potom, čo sa vytvorí spojenie medzi klientom a serverom, začne server posilať fragmenty videa v paketoch, typicky o veľkosti 1452 bajtov. V prípade, že stream má dátový tok 1 Mbps, jednotlivé pakety sú prenášané približne po 11 milisekundách. Tieto pakety môžu byť prenášané pomocou sieťového protokolu UDP (User Datagram Protocol) aj TCP (Transmission Control Protocol), ktorý je uprednostňovaný v prípade blokovania UDP spojenia. Metóda klasického streamingu je zobrazená na obr. 1.1.

TCP protokol je však zložitejší a tak má TCP spojenie typicky vyššie oneskorenie ako prenos pomocou bezstavového UDP.

HTTP (Hypertext Transfer Protocol) je oproti predchádzajúcemu RTSP známy ako bezstavový protokol. V prípade kedy HTTP klient požaduje nejaké dáta, server ich pošle, pričom si nemusí pamätať klienta a nemusí s ním udržiavať stále spojenie. Každá HTTP požiadavka je odbavená ako samostatná [5].



Obr. 1.1: Tradičná metóda streamovania

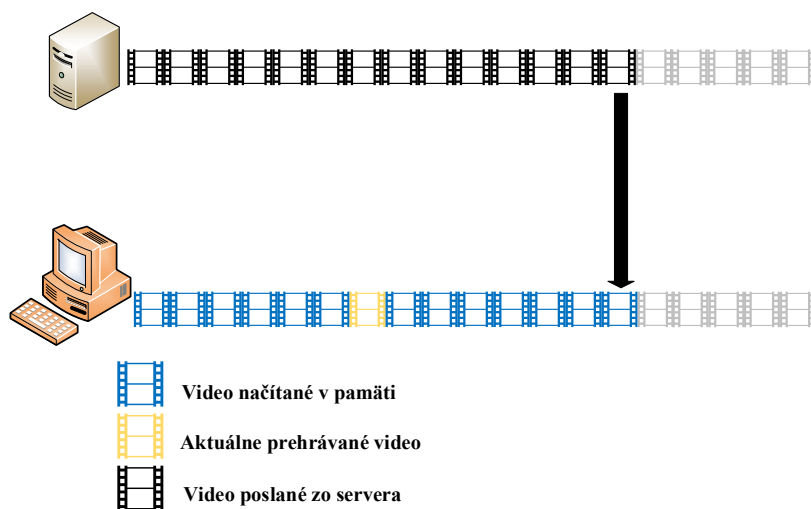
### 1.1.2 Progresívny download

Ďalšou bežne používanou formou doručovania obsahu je progresívny download, ktorý nie je ničím iným než obyčajným stiahnutím súboru z HTTP servera.

Progresívny download je podporovaný veľkou časťou prehrávačov i platforiem vrátane Adobe Flash, Silverlight a Windows Media Player. Slovo „progressive“ vyplýva z toho, že väčšina prehrávačov je schopná prehrávať multimedialný súbor, zatiaľ čo jeho sťahovanie prebieha, čo je predtým než je kompletný súbor zapísaný na disk. Prehrávače podporujúce HTTP 1.1 môžu dokonca v súbore seekovať a užívateľ tak môže preskočiť aj na pozíciu, ktorá ešte nebola stiahnutá. Týmto zažiada HTTP server o zaslanie požadovaného rozmedzia dát.

Progresívny download je v dnešnej dobe využívaný webovými portálmi ako napríklad MySpace, MSN. Na rozdiel od klasického streamingu, kedy sa dáta málokedy posielajú s viac ako 10 sekundovým predstihom, pri progresívnom downloade sa dáta sťahujú do tej doby, kým nie je stiahnutý celý súbor. Progresívna metóda je zobrazená na obr. 1.2.

Nevýhodou je, že akonáhle sa užívateľ rozhodne video v polovici zastaviť a ďalej nesledovať, bol už pravdepodobne celý súbor stiahnutý, pričom bola zbytočne využitá kapacita prenosového pásma na stiahnutie celého súboru [5].



Obr. 1.2: Progresívne sťahovanie

### 1.1.3 Adaptívne streamovanie založene na HTTP

Adaptívny streaming je založený na postupnom sťahovaní HTTP (progresívnom downloade). Táto moderná koncepcia však nevyužíva nový protokol, ale pracuje s protokolom HTTP, pričom využíva delenie dlhého súboru na mnoho veľmi malých segmentov.

Pri adaptívnej streamingovej implementácii sa video / audio rozdelí na mnoho malých segmentov, a tie sú kódované do daného formátu. Tieto malé časti sú dlhé typicky od 2 do 4 sekúnd. Na úrovni kodeku to znamená, že sa každá časť rozdelí tak, aby mal kľúčový snímok, bez závislosti na snímkach minulých alebo budúcich. Umožňuje to následné dekódovanie každej časti nezávisle od ostatných.

Jednotlivé chunky sú hostované na HTTP servery a postupne požadované klientom. Kúsky videa sa následne sťahujú za sebou ako by to boli jednotlivé súbory, pričom sa použije metóda progresívneho downloadu. Vzhľadom k tomu, že chunky sú zakódované bez medzier a bez prekryvania, tak je výsledný obraz pre koncového užívateľa nerozoznateľný od jedného video súboru.

Adaptívna časť riešenia prichádza v čase, keď sa video kóduje v niekoľkých rôznych kvalitách. Prehrávač si teda môže voľiť medzi rôznymi kvalitami. HTTP servery totiž najčastejšie fungujú na princípe čo najrýchlejšieho doručenia dát k užívateľovi na základe dostupnej priepustnosti linky. Prehrávač môže vyhodnotiť túto priepustnosť a rozhodnúť, v akej kvalite bude sťahovať. Veľkosť prehrávacej / sťahovacej pamäte je plne voliteľná. Adaptívny streaming, rovnako ako ostatné formy HTTP doručovania, ponúka oproti tradičnému streamovaniu nasledujúce výhody [5] [15]:

- Lacnejšie nasadenie – adaptívny streaming využíva HTTP cache / proxy a nevyžaduje špecializované servery v každom uzle siete.
- Lepšia dostupnosť – znížením problému v koncových častiach siete, ktorú nemôžeme ovplyvniť. Kvalita videa sa môže v priebehu dynamicky prepínať z horšej na lepšiu a naopak, a tým reagovať na zníženie / zvýšenie priepustnosti siete.
- Rýchle spustenie a skrátenie času pre posúvanie, pretože pri týchto operáciách môže byť video spustené na najnižšej prenosovej rýchlosti s prechodom na vyššiu prenosovú rýchlosť neskôr, akonáhle sa začne prehrávať v normálnom režime.
- Plynulé prepínanie prenosovej rýchlosti založené na aktuálnej priepustnosti siete a vyťažnosti procesoru.
- Všeobecne konzistentnejšie a plynulejšie prehrávanie poskytuje lepší zážitok z obsahu [5] [15]. Jednotlivé techniky adaptívneho streamovania sú priblížené v nasledujúcu

## MPEG–DASH

MPEG – DASH (Dynamic Adaptive Streaming over HTTP) je najnovšia doručovacia metóda video obsahu. Používa ju už napríklad YouTube, vzhľadom k jej prednostiam však určite čoskoro nahradí staršie technológie.

Najznámejšie metódy adaptívneho streamingu používané v dnešnej dobe (HLS od Apple, Smooth Streaming od Microsoftu a HDS od Adobe) sú z väčšej časti podobné, bohužiaľ však nie sú úplne kompatibilné. Princíp technológie DASH môžeme vidieť na obr 1.3. Pre každú technológiu je nutné mať špeciálny prehrávač, ktorý tento formát dokáže spracovať. Roztrieštenosť v metódach doručovaní viedla k tomu, že v apríli 2009 pracovná skupina MPEG požiadala o návrhy nového štandardu. V priebehu 3 mesiacov dostala celkom 15 návrhov, na základe ktorých potom začala spolu s organizáciou 3GPP vyvíjať nový štandard. Výsledkom bol dynamický adaptívny HTTP streaming MPEG – DASH.

MPEG – DASH nie je systém ani protokol, ale nový štandard pre podporu efektívneho a vysoko kvalitného streamingu. MPEG – DASH bol štandardizovaný na začiatku roka 2012 pod normou ISO / IEC 23009-1. Dizajn MPEG – DASH bol navrhnutý tak, aby mohol využiť [7]:

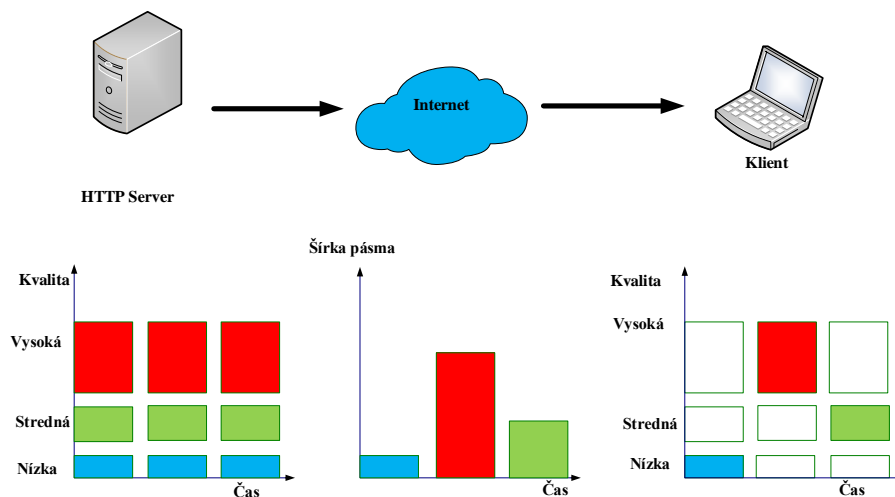
- Existujúce technológie (video kontajnery, kodeky, DRM a pod.).
- Existujúcu sieťovú infraštruktúru zameranú na HTTP.
- Uživatelské pohodlie v podobe rýchleho štartu a čo najmenšieho čakania.
- Hladkého prepínania medzi kvalitami.
- Streaming ako live, tak VoD obsahu, prípadne time-shiftového obsahu.
- Možnosti presunúť riadenie a rozhodovanie zo strany siete na stranu klienta.

MPEG – DASH, využíva k doručeniu obsahu fragmenty, ktoré sa doručujú pomocou protokolu HTTP až ku prehrávaču na strane klienta, ktorý ich poskladá v jedno súvislé video [7].

## Apple HTTP Live Streaming

HTTP Live Streaming umožňuje posielanie zvuku a videa cez HTTP z bežného webového servera pre prehrávanie na iOS zariadeniach založených na iPhone, iPad, iPod touch a Apple TV a stolových počítačoch (Mac OS X). HTTP Live Streaming podporuje ako živé vysielanie aj nahrané video na požiadanie, podporuje tiež rôzne prúdy rôznych prenosových rýchlostí, a klientsky softvér môže prepínať prúdy inteligentne podľa zmeny šírky pásma siete. HTTP Live Streaming poskytuje tiež šifrovanie obsahu a overovanie používateľov cez protokol HTTPS, ktorý umožňuje vydavateľom chrániť ich prácu. HTTP Live Streaming vysiela audio a video ako sériu malých súborov, typicky 10 sekúnd nazývané media segment files. Index file alebo playlist dáva klientom URL adresy mediálnych segmentov súborov. Zoznam skladieb





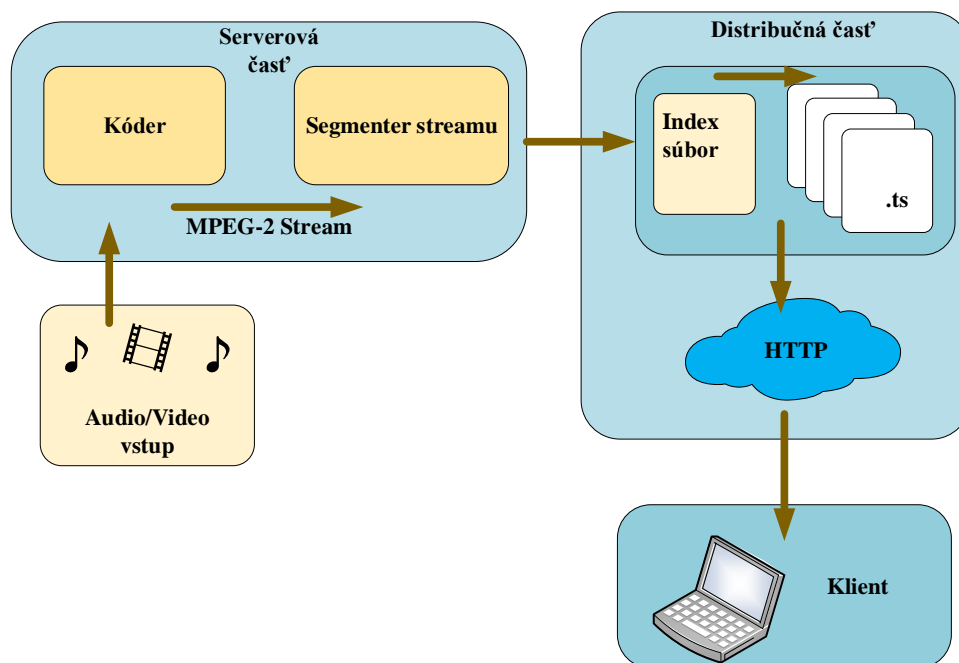
Obr. 1.3: Metóda MPEG – DASH

môže byť pravidelne aktualizovaný pre live vysielanie kde sa mediálne súbory segmentov neustále produkujú. Môžeme vložiť odkaz na playlist na webovej stránke, alebo ho odoslať do aplikácie. Konceptne je HTTP Live Streaming zobrazený na obr. 1.4 a skladá z troch častí [10]:

- Serverová časť – je zodpovedná za prijatie vstupných dát, digitálne kódovanie a zapúzdrenie vo vhodnom formáte na distribúciu, teda prípravu dát na distribúciu.
- Distribučná časť – pozostáva zo štandardných webových serverov, ktoré sú zodpovedné za prijatie požiadaviek klientov a poskytovanie pripravených médií.
- Klientsky softvér – zodpovedá za určenie príslušného média na vyžiadanie. Sťahovanie a skompletizovanie týchto zdrojov prebieha tak, aby média boli prezentované užívateľovi nepretržitom prúde. Klientsky softvér je obsiahnutý od iOS 3.0 a v Safari od verzie.

### Microsoft smooth streaming

IIS Media Services je prvým produktom Microsoftu, ktorý podporuje adaptívne streamovanie. Prvý krát bol použitý na prenos letných olympijských hier v roku 2008 pre portál NBCOlympics.com. V Smooth stream klienti využívajú jednoduchý koncept doručenia malých fragmentov, typicky 2 sekundové časti. Fragmenty sú



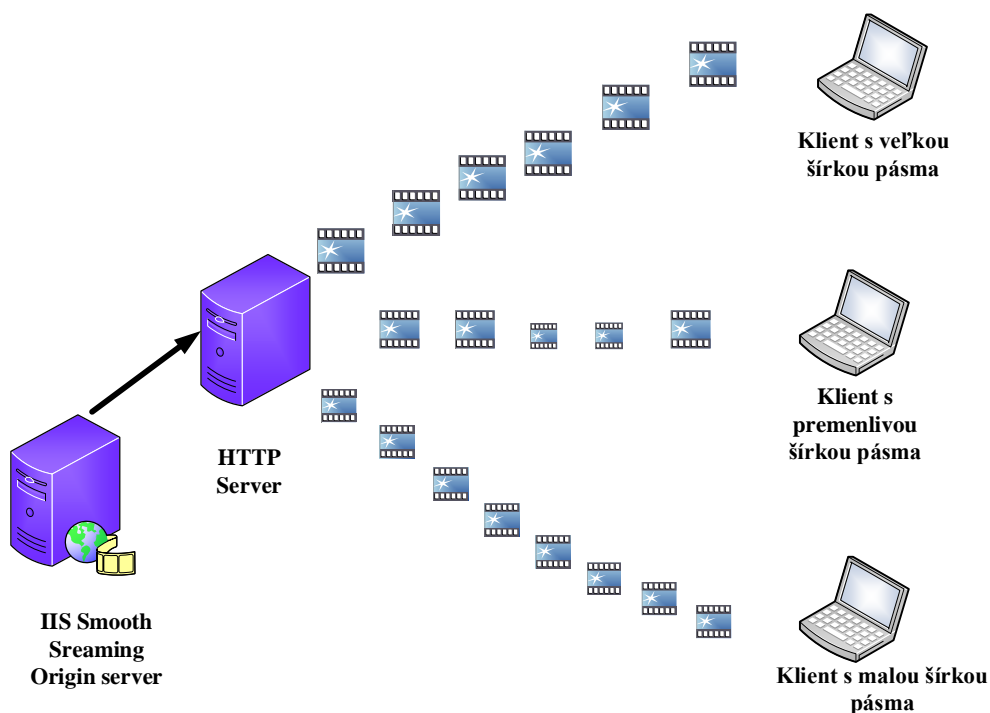
Obr. 1.4: Metóda adaptívneho streamovania od Apple

kontrolované, či dorazili včas a môžu byť prehrávané v požadovanej kvalite. Ak niektorý fragment nesplní tieto požiadavky, ďalší fragment bude odoslaný v nižšej kvalite, a ak to priepustnosť siete dovoľí, tak sa kvalita videa prepne na vyššiu. Na obr. 1.5 sú zobrazení klienti z rôznymi veľkosťami šírky pripojenia. IIS Media services nieje založená na klasických streamovacích protokoloch ale na HTTP.

Microsoft pre Smooth Streaming použil nový kontajner MP4 namiesto ASF. Najmä pre menšiu réžiu a ľahšie analyzovanie a širokej podpory tretích strán. MP4 je vyladený na podporu H.264 ktorý sa používa v širokom spektre Microsoft produktov ako Silverlight, Windows 7, Xbox 360 a Mediaroom. Streamingové formáty Smooth sú dva: sieťový formát (wire) a disk file formát. Pri enkódovaní sa video nahráva do jedného dlhého súboru pre každú kvalitu na disk, ale ku klientovi je obsah doručovaný ako séria malých chunkov. Wire formát definuje štruktúru súvislého súboru na disku, čo zlepšuje správu súboru. MP4 špecifikácia dovoľuje vnútornú organizáciu série fragmentov, z toho teda vyplýva že Smooth Streaming Wire formát je podmnožinou disk file formátu [7].

### Adobe HTTP Dynamic Streaming

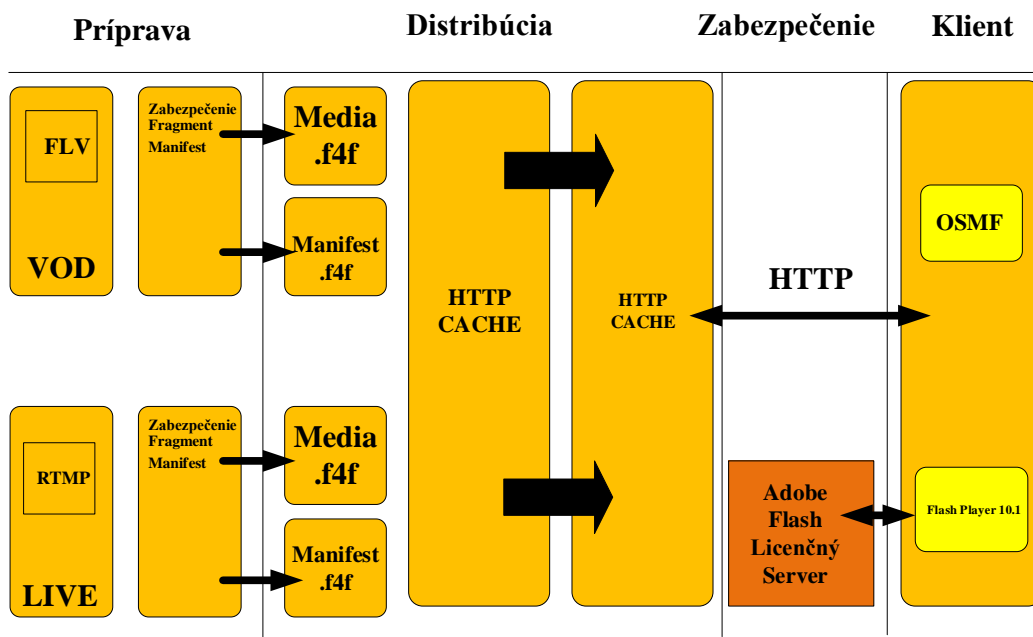
Dynamické streamovanie je proces efektívneho doručovania streamovaného videa pre používateľa tým, že dynamicky prepína medzi rôznymi prúdmi rôznej kvality a



Obr. 1.5: Metóda streamovania Microsoft Smooth

veľkosti pri prehrávaní. To umožňuje užívateľom najlepší možný zážitok zo sledovania podľa ich šírky pásma. Ďalším významným cieľom dynamického streamovania je, aby bol tento proces plynulý a hladký pre užívateľov tak, že v prípade zmeny šírky pásma sa kvalita prispôsobí, a zmena je takmer nerozpoznatelná od kontinuálneho prehrávania, princíp Adobe HDS môžeme vidieť na obr. 1.6.

Najnovšia verzia prehrávača Flash Player a servera Flash Media Server podporuje adaptívnu prenosovú rýchlosť streamingu cez tradičné RTMP, rovnako ako HTTP, podobné riešenie založené na HTTP spoločnosti Apple a Microsoft, HTTP Dynamické streamovanie podporované v prehrávači Flash Player 10.1 a novších. HTTP na báze streamingu má tú výhodu, že nevyžaduje žiadne porty firewallu pri otvorení mimo normálnych portov používaných webovým prehliadačom [1].



Obr. 1.6: Metoda od spoločnosti Adobe

## 2 PREHRÁVAČE DASH

Táto kapitola sa venuje prehrávačom adaptívneho streamovania. Popísané sú výhody a technológie ktoré využívajú. V kapitole 2.4 je popísaný prehrávač Dash.js ktorý je použitý v praktickej časti.

### 2.1 JW player

JW prehrávač umožňuje doručovanie live prenosu, ako aj prenosu na vyžiadanie pomocou dvoch technológií Adobe RTMP a Apple HLS. Obidva formáty automaticky vyberú kvalitu videa na dostupnej šírke pásma. Rozdiel je, že Adobe RTMP ide len vo Flash desktop a vyžaduje špecializované servery pričom Apple HLS funguje naprieč všetkými zariadeniami v sieti.



Obr. 2.1: Prehrávač JW Player

### 2.2 Video.js

Video.js je moderný prehrávač používaný na prehrávanie videa na web stránkach. Je založený na Open Source knižnici takže je možné si ho volne prispôbiť. A v tom spočíva jeho veľká výhoda, ponuka množstvo rozšírení a prispôsobení podľa potrieb užívateľa.



Obr. 2.2: Prehrávač Video.js

## 2.3 MediaElement.js

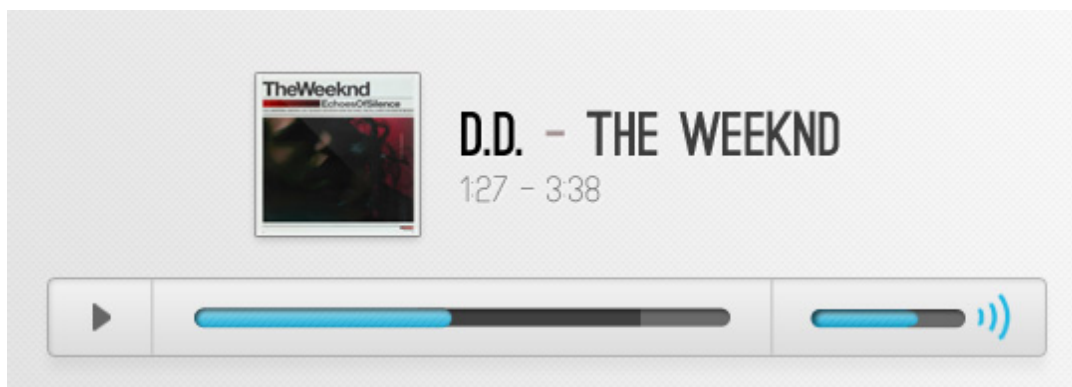
Mediaelement.js moderný prehrávač ktorý podporuje širokú škálu formátov. Ponúka HTML5 prehrávač pre novšie prehliadače a samostatné Flash a Silverlight pluginy, ktoré napodobňujú HTML5 MediaElement API pre staršie prehliadače. Je to audio/ video prehrávač s možnosťami veľkého prispôsobenia [6].



Obr. 2.3: Prehrávač Mediaelement.js

## 2.4 jPlayer

jPlayer je open source riešenie, knižnica je napísaná v JavaScripte. Umožňuje rýchle zostavenie audio a video streamu z webovej stránky. Podporuje takmer všetky webové prehliadače. JPlayer je ľahko nasaditeľný, má rozšíriteľnú architektúru a konzistentné API pre HTML5 a Adobe Flash [12].



Obr. 2.4: Prehrávač jPlayer

## 2.5 Dash.js

Dash.js je open source MPEG – DASH prehrávač napísaný v JavaScripte. Jeho cieľom je poskytovať robustný multiplatformový prehrávač, je možné ho voľne používať v aplikáciách na prehrávanie videa. Poskytuje širokú škálu funkcií, prehrávanie MPEG – DASH v ľubovoľnom prehliadači ktorý podporuje MSE (Media Source Extensions) čo je napríklad Google Chrome a Internet Explorer 11 [9].



Obr. 2.5: Prehrávač Dash.js

## 3 ÚPRAVA VIDEA

Pri použití adaptívneho streamingu je potrebné upraviť zdroj videa v rôznych variantách. Úpravou môžeme rozumieť napríklad zníženie bitového toku, prekódovanie do správneho kodeku, zmenu rozlíšenia, pridanie loga či nejakej grafiky, alebo prípravu zdroja pre adaptívne streamovanie. Napríklad YouTube používa hlavné dva kodeky H.264 a VP9. Vo väčšine streamovacích softvérov máme tieto jednotlivé možnosti v ponuke, teda stačí iba správne nastaviť parametre a na výstupe už budeme mať hotový stream v nami požadovanom dátovom toku a rozlíšení. Ak však nechceme využívať streamovací softvér, máme možnosť predpripraviť si vstupný zdroj pre nejakú aplikáciu, napríklad pomocou programu ffmpeg, ktorý bol použitý v tejto práci [16] [17].

### 3.1 Sony Vegas Pro

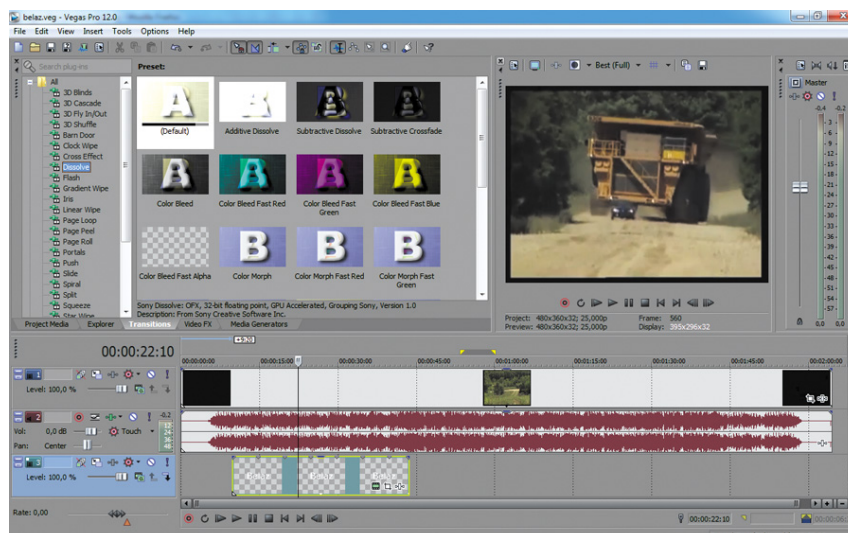
História video editoru Sony Vegas sa viaže k známej spoločnosti Sonic Foundry, ktorej vlastníkom je teraz skupina Sony Creative Software, ktorá produkuje a vyvíja softvérové produkty pre spracovanie videa a zvuku. Sonic Foundry Vegas Video bol pôvodne v podstate rozšírený editor zvukových nahrávok so schopnosťou pracovať aj s obrazovými stopami. Prostredie programu Sony Vegas Pro je na obr. 3.1. Dnes je z Vegas Video celá produktová rada video editorov pre spracovanie a produkciu domáceho i profesionálneho videa. Mimo iných novinek prináša predovšetkým GPU akceleráciu rôznych pracovných akcií, efektov a kódovanie / dekodovanie videa.

Balík Sony Vegas Pro 11 je vlastne sada dvoch hlavných aplikácií pre prácu s videom a zvukom, ktoré možno v editore importovať, zostrihať, obrazovo a efektovo upraviť a následne publikovať v celej rade formátov. Mimo samotného video editoru rovnakého názvu obsahuje Vegas Pro ešte aplikáciu DVD Architect Pro, ktorá slúži k príprave a zostavovaniu obsahu pre DVD a Blu-ray médiá a vie s Vegas Pro priamo spolupracovať na úrovni výmeny projektov a ich dát. Architect Pro má necelých štyridsať nových grafických šablón pre tvorbu navigačných menu v HD rozlíšení, podporuje grafické dokumenty Adobe Photoshopu a integruje v sebe jednoduchú časovú os pre skladanie a manipuláciu s obsahom DVD a Blu-ray kompilácií [14].

### 3.2 Avidemux

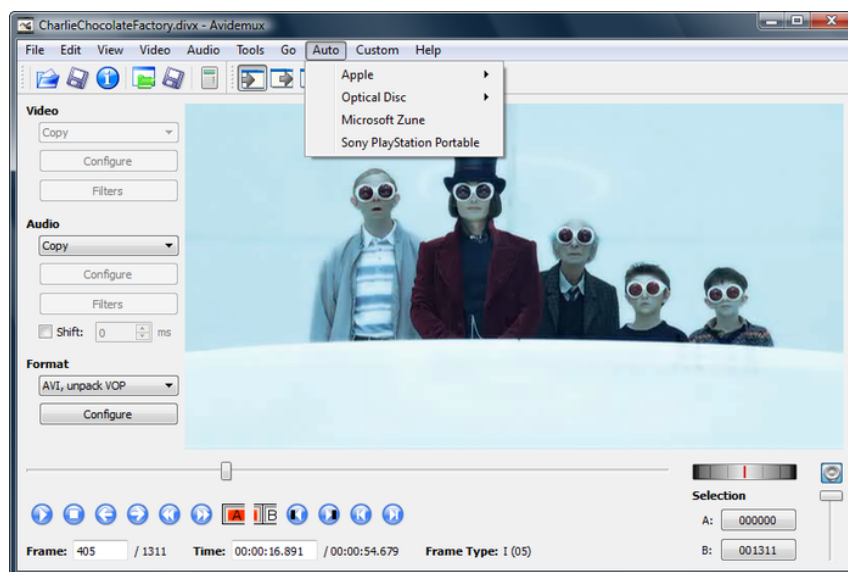
Avidemux je voľný editor videa určený pre jednoduché strihanie, úpravy s filtrami a kódovanie videa. Podporuje mnoho typov súborov, vrátane AVI, MP4 a ASF za použitia rôznych kodekov. Úlohy je možné automatizovať pomocou projektov a





Obr. 3.1: Program Sony Vegas Pro

výkonných skriptov. Avidemux je dostupný pre operačné systémy: Linux, Mac OS X, a Microsoft Windows pod licenciou GNU GPL [2].



Obr. 3.2: Program Avidemux

### 3.3 Nástroj FFmpeg

FFmpeg vznikol pôvodne pod GNU / Linuxom, ale teraz je podporovaný nielen pod operačným systémom Linux, ale tiež pod operačnými systémami Windows a

Mac OS X. Tento softvér je snáď najuniverzálnejším dekodovacím, kódovacím, transkodovacím a prehrávacím nástrojom, pretože si dokáže poradiť s väčšinou video-kompresných formátov. Možnosti úprav videa ktoré FFmpeg ponúka, boli aspekty k tomu aby bol vybraný na úpravu obsahu pre adaptívne streamovanie pomocou techniky MPEG-DASH v tejto práci Podrobnejší návod a použité príkazy sú uvedené v popise praktickej časti v kapitole 4.

FFmpeg sa snaží poskytovať najlepšie technicky možné riešenie pre vývojárov aplikácií a koncových užívateľov, nie je spoplatnený. Tento program sa ovláda pomocou príkazového riadku, kde si ktokoľvek môže naprogramovať svoje grafické rozhranie, prípadne použiť nejaké z voľne dostupných, ale nemusí mať zaručenú presnosť nastavenia a bezpečnosť. Podrobný manuál programu, je dostupný na oficiálnych stránkach FFmpeg, kde sú podrobné vysvetlené jednotlivé príkazy a prepínače. Nástroj FFmpeg [8].



```
c:\WINDOWS\System32\cmd.exe
E:\vdvd>ffmpeg
ffmpeg version 0.4.8, Copyright (c) 2000-2003 Fabrice Bellard
usage: ffmpeg [options] -i input_file... [options] outfile...
Hyper fast Audio and Video encoder

Main options:
-l          show license
-h          show help
-formats   show available formats, codecs, protocols, ...
-f fmt     force format
-img img_fmt force image format
-i filename input file name
-y         overwrite output files
-t duration set the recording time
-title string set the title
-author string set the author
-copyright string set the copyright
-comment string set the comment
-v verbose control amount of logging
-hq       activate high quality settings

Video options:
-b bitrate set video bitrate (in kbit/s)
-r rate    set frame rate (Hz value, fraction or abbreviation)
-s size    set frame size (WxH or abbreviation)
-aspect aspect set aspect ratio (4:3, 16:9 or 1.3333, 1.7777)
-croptop size set top crop band size (in pixels)
-cropbottom size set bottom crop band size (in pixels)
-cropleft size set left crop band size (in pixels)
-cropright size set right crop band size (in pixels)
-vn        disable video
-bt tolerance set video bitrate tolerance (in kbit/s)
-maxrate bitrate set max video bitrate tolerance (in kbit/s)
-minrate bitrate set min video bitrate tolerance (in kbit/s)
-buffersize size set ratecontrol buffere size (in kbit)
-vcodect codec force video codec ('copy' to copy stream)
-sameq     use same video quality as source (implies VBR)
-pass n    select the pass number (1 or 2)
-passlogfile file select two pass log file name

Advanced Video options:
-pix_fmt format set pixel format
-g gop_size set the group of picture size
-intra     use only intra frames
-qscale q  use fixed video quantiser scale (VBR)
-qmin q   min video quantiser scale (VBR)
-qmax q   max video quantiser scale (VBR)
-mbqmin q min macroblock quantiser scale (VBR)
-mbqmax q max macroblock quantiser scale (VBR)
```

Obr. 3.3: Nástroj FFmpeg

## 4 MERANIE KVALITY ZO ZÁŽITKU

V poslednej dobe výrazne stúpa popularita streamovania videa. Cisco Visual Networking Index poukazuje na skutočnosť, že streamovanie videa bude mať dominantný podiel na raste mobilnej prevádzky [4]. A to hlavne z dôvodu, že streamovanie videa vyžaduje vyššiu prenosovú priepustnosť ako iný typ obsahu na Internete. V dnešnej dobe najpoužívanejšie a najobľúbenejšie služby pre streamovanie videa sú YouTube a Netflix.

Netflix si drží prvú priečku hlavne v Severnej Amerike, celosvetovo však drží prvú pozíciu v online streamovaní videa YouTube, ako pre pevné tak aj pre mobilné pripojenia (15 – 30% celkového dátového prenosu). Neustále narastajúci multimediálny prenos v reálnom čase predstavuje výzvu hlavne pre mobilných operátorov, ako zvládnuť veľké množstvo dát a vysoký počet užívateľov čo najefektívnejšie. Vďaka rýchlemu vývoju informačných a komunikačných technológií sú už na trhu bežne dostupné mobilné zariadenia ponúkajúce displej s vysokým rozlíšením a na druhej strane už tiež existujú služby pre streamovanie videa, ktoré ponúkajú zdroj multimediálneho obsahu vo vysokom rozlíšení (720p, 1080p, 4K). Video v rozlíšení 1080p má bitový tok okolo 8 Mb/s a video s rozlíšením 4K (3840x2160) cca 35 až 45 Mb/s. Služba s takým dátovým tokom predstavuje vysoké komunikačné nároky na strane operátora.

Kľúčovou snahou mobilných operátorov je zaistenie vysokej kvality ponúkaných služieb a udržiavanie spokojnosti ich zákazníkov na čo najvyššej úrovni. S narastajúcim objemom prenášaných dát a požiadaviek na vysokú sieťovú priepustnosť je tento cieľ však stále ťažší. V dnešnej dobe sa kladie veľmi veľký dôraz práve na samotných zákazníkov a koncových užívateľov, rieši sa takzvaná kvalita zážitku zákazníka. Táto problematika je závislá na kvalite sieťovej prevádzky, ale tiež i na mnoho ďalších faktoroch ako je napríklad samotný užívateľ a jeho očakávania. Z tohto dôvodu nie je riešenie tejto problematiky triviálne a je dôležité sa ňou zaoberať.

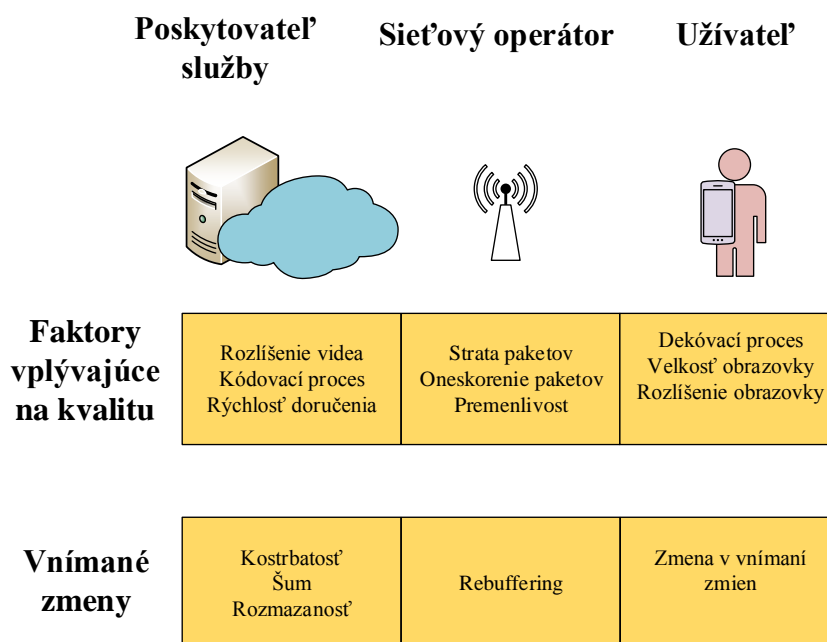
### 4.1 Faktory vplyvajúce na meranie QoE

Pri zisťovaní a nastavovaní optimálnych parametrov na prenos dát sa vychádza z dvoch hľadísk ktoré spolu úzko súvisia. Jedným je kvalita služby známa ako QoS (Quality of Services) a druhým je kvalita zážitku QoE (Quality of Experience). Kvalita služby bola definovaná v ITU štandarde X.902 [11] ako súbor požiadaviek na fungovanie jedného alebo viacerých objektov. Kvalita služby zahŕňa požiadavky na všetky aspekty spojenia, ako je čas odpovede služby, strata, pomer signál šum, prerušenia, frekvenčná citlivosť a miera hlasitosti. Pojem Kvalita zážitku je o niečo

komplikovanejši. Dá sa však zovšeobecniť ako spokojnosť s poskytovanými službami, potreby zákazníkov pri využívaní týchto služieb a v neposlednom rade celková spokojnosť zákazníka s poskytnutou službou. Kvalita zážitku je teda miera koncovej výkonnosti pripojenia z pohľadu používateľa a indikátor, ako dokáže splniť toto pripojenie používateľské požiadavky [21]. QoE zahŕňa kompletný end-to-end systém (klient, terminál, sieť, služby infraštruktúry), kde celková prijateľnosť môže byť ovplyvnená očakávaním používateľa a kontextu [20]. Výsledné hodnotenie kvality môže byť ovplyvňované celým radom faktorov. Tieto faktory môžu mať technický základ, ale môžu byť tiež iného pôvodu, ako napríklad pôsobenie konkrétneho obsahu na psychiku užívateľa. Faktory možno rozdeliť do 4 hlavných skupín [19]:

- Prostredie – predstavuje skutočnú situáciu v ktorej sa daná mobilná dátová služba využíva. Prostredie, v ktorom dochádza k sledovaniu príslušnej služby má už samo o sebe veľký vplyv na výsledné hodnotenie subjektívneho vnímania kvality. Medzi tieto vplyvy prostredia možno napríklad zaradiť osvetlenie, alebo pozorovacia vzdialenosť od koncového zariadenia. Ďalej sociálne a kultúrne vplyvy prostredia ovplyvňujú vnímanie služby, kde záleží na type a účele prehliadania obsahu.
- Používateľ – vzhľadom k tomu, že QoE je o vnímaní aplikácie alebo služby tak, ako je doručená koncovému užívateľovi, tak výsledné hodnotenie môže byť ovplyvnené vnútorným, či už psychickým alebo fyzickým, stavom užívateľa, ako osoby, tak aj jeho očakávaním, ktoré bolo viazané na konkrétny typ služby. Očakávanie o určitej službe značne ovplyvňuje výsledné hodnotenie a tvorí vnútornú referenciu užívateľa na vnímanie doručenej služby. Prípadná skúsenosť z minulosti s podobným typom služieb má tiež vplyv na výsledné subjektívne hodnotenie.
- Systém – predstavuje technické vybavenie, ktorým je doručovaná a využívaná daná služba. Vlastnosti tohto systému ako celku ovplyvňujú výsledné hodnotenia subjektívneho vnímania. S pohľadu mobilných komunikácií je braný systém ako celok skladajúci sa z koncového zariadenia a siete na prenos dát. Vlastnosti koncového zariadenia aj siete majú vplyv na výsledné vnímanie služby. Medzi tieto vlastnosti z pohľadu koncového zariadenia patrí napríklad rozlíšenie displeja, veľkosť zariadenia, rýchlosť odozvy na jednotlivé požiadavky. Ďalej možno zahrnúť parametre dátovej siete určené pre doručenie požadovaného obsahu, ktoré sú definované v rámci QoS. Medzi tieto parametre je možné napríklad zaradiť dostupnú šírku pásma pre konkrétnu službu a oneskorenie v sieti. Tieto parametre majú vplyv na doručenie obsahu smerom k užívateľovi.
- Obsah – doručený obsah v rámci služby je jeden z ďalších faktorov. Na účely získania výsledkov z hodnotenia možno tento faktor čiastočne eliminovať výberom vhodného obsahu pre konkrétny test. Ak dochádza k hodnoteniu na

obsahu, ktorý z nejakého dôvodu nie je koncovému užívateľovi príjemný, tak sa táto skutočnosť negatívne prejaví na získaných výsledkoch. To platí aj v opačnom prípade, kedy obsah služby môže výsledky ovplyvniť pozitívne a používateľ sa v takom prípade skôr zameria na obsah než na ostatné parametre služby. Dôležité je zvoliť čo najviac neutrálny obsah služby.



Obr. 4.1: Faktory vplyvajúce na QoE

## 4.2 Metody merania kvality zo zážitku

Metódy pre meranie QoE môžu byť rozdelené do dvoch hlavných kategórií: objektívne a subjektívne. Subjektívna metóda je najčastejšie postavená na priamom dopytovaní skupiny ľudí na kvalitu danej služby. Na druhú stranu objektívna metóda pre streamovanie videa poskytuje predikciu QoE iba na základe charakteristík a parametrov streamovaného videa.

### 4.2.1 Objektívne hodnotenie QoE

Objektívna metóda hodnotenia QoE sa snaží modelovať výsledky subjektívnej metódy so zameraním len na parametre siete a streamovaného videa ako bitový

tok videa, rozlíšenie, počiatočné načítanie do vyrovnávacej pamäte, počet prerušení kvôli ukladaniu do pamäte. Modely pre objektívne hodnotenie QoE zobrazovaných médií možno rozdeliť do troch typov na základe informácií použitých v modeloch.

### **Nereferenčný model**

Ne-referenčný model hodnotí kvalitu obrazu na základe spracovaných rámcov. Model nevyžaduje žiadne informácie o zdrojovom videu, čo môže byť použité v mnohých prostrediach. Vzhľadom k tomu že model nepoužíva informácie zo zdrojového videa, je tento model menej presný ako čiastočný a úplný referenčný model [20].

### **Čiastočne referenčný model**

Čiastočne referenčný model objektívne hodnotí kvalitu obrazu na základe informácií spracovaného videa a malého množstva informácií zo zdrojového videa. Model využíva informácie o zdrojovom videu a informácie spracovaného videa tento model je veľmi presný, no nie až tak, ako úplný referenčný model. Vzhľadom k tomu že model používa malé množstvo dát zo zdrojového videa na vykonanie hodnotenia je model možné použiť na kontrolu kvality v sieťových uzloch [20].

### **Úplný referenčný model**

Úplný referenčný model objektívne hodnotí kvalitu obrazu spracovaného videa porovnaním informácie o zdrojovom videu. Vzhľadom k tomu, model porovnáva zdrojové video so spracovaným videom, poskytuje vysoko presné objektívne posúdenie kvality. Hlavnou nevýhodou je, že to vyžaduje veľmi veľké množstvo dát z pôvodného videozáznamu a zahŕňa teda prostredie, ktoré zvládne značné náklady na získanie zdrojového videa a nie je ľahko implementovateľné u koncových užívateľov. Model sa najlepšie hodí pre kvalitné videá na konci hlavných obrazových sietí [20].

Objektívne metódy pre hodnotenie kvality zážitku je možné rozdeliť do piatich nasledujúcich typov [3]:

- End-to-end QoS – model sa snaží nájsť koreláciu medzi QoE a QoS, ktorá potom vedie k prvému významnému kroku k optimalizácii niektorých ovplyvňujúcich faktorov.
- Parametrický model sieťovej vrstvy – odhaduje QoE z informácií obsiahnutých v hlavičke paketu. Tento model využíva kvalitatívne parametre siete a terminálu ako vstupné hodnoty.

- Model médií – odhaduje QoE analýzou signálu médií cez HVS (Human Visual System) model, ktorý sa snaží vyrovnat s biologickými a psychologickými procesmi človeka, ktoré nie sú celkom pochopiteľné, tento model sa z komplexného systému urobí jednoduchší.
- Model bitového streamu – je z časti parametrický model sieťovej vrstvy a z časti ako model médií. Odhad kvality je zaistený analýzou obsahu vlastností z kódovaného bitového toku.
- Hybridný model – je kombináciou niektorých alebo všetkých vyššie uvedených modelov. Je to efektívny model pokiaľ sa využije pre predikciu QoE čo najviac dostupných informácií, jeho nevýhodou je ale náročnosť na zhromaždenie potrebných informácií a výpočetná náročnosť na zariadenia a samotná implementácia modelu.

#### 4.2.2 Subjektívne hodnotenie QoE

Subjektívne hodnotenie kvality zahŕňa vizuálne psychologické testy, kde sú hodnotitelia vystavení videu a posudzujú jeho kvalitu na základe vlastného subjektívneho úsudku. Použitím subjektívneho hodnotenia získavame najpresnejší popis vnímanej kvality ponúkanej služby, keďže nie je lepší indikátor kvality ako ten, ktorý je daný človekom krotý danú službu používa. Výsledok sa vyhodnotí ako priemerná známka názoru MOS (Mean Opinion Score) [21]. Vyjadruje sa pomocou stupnice od 1(najhoršia) po 5(najlepšia). Vyjadruje hodnotenie danej služby z pohľadu koncového užívateľa. Aj napriek svojej presnosti popisu je pre tento prístup potrebné veľké množstvo pozorovateľov a taktiež je nepoužiteľný v prípade automatických meraní alebo monitorovacích nástrojov v reálnom čase. Pre poskytovateľov služieb a sietí je vhodnejšie mať nástroje, ktoré objektívne odrážajú s rozumnou presnosťou. Problematika riešenia v tejto oblasti je veľmi široká a existuje celá rada metód pre rôzne typy obsahu [21].

### 4.3 Meranie QoE pri streamovaní videa

Pri meraní QoE streamovaného videa u koncového užívateľa sú vizuálne najdôležitejšie faktory pre kvalitu obrazu sú nasledovné: sledovacia vzdialenosť, veľkosť displeja, rozlíšenie, jas, kontrast, ostrosť, farebnosť, a prirodzenosť. K ďalším zdrojom chýb dochádza počas prenosu. Tri najdôležitejšie prenosové artefakty sú strata paketov, oneskorenie a kolísanie oneskorenia. Vizuálny efekt stratenej informácie je závislý na kodeku a type informácií, ktoré sú stratené. Niektoré chyby môžu byť ukryté za použitia správnej maskovacej stratégie, zatiaľ čo iné majú vážne následky [13].

## 5 VYTVORENIE WEBOVEJ STRÁNKY S PREHRÁVAČOM MPEG – DASH

Táto časť práce sa zaoberá prípravou zdroja, teda videa pre adaptívne streamovanie v niekoľkých verziách rozlíšenia a k ním príslušnej bitovej rýchlosti pomocou nástroja FFmpeg a vytvoreniu prehrávača adaptívneho streamu pomocou techniky MPEG – DASH a jeho implementácií na vytvorenú webovú stránku. Na porovnanie sú priložené dva obrázky, prvý pri prehrávaní v malom rozlíšení obr. 5.1 a druhý pri prehrávaní vo veľkom rozlíšení na obr. 5.2.

### 5.1 Príprava streamovaného videa

Pre adaptívne streamovanie videa je potrebné zdroj upraviť tak bolo možné zaisťiť prepínanie medzi niekoľkými verziami videa pri zmene v sieti. Pre túto úlohu bol vybraný nástroj FFmpeg, ktorý je univerzálny nástroj na úpravy videa ako napríklad prekódovanie, zmena rozlíšenia a mnoho ďalších možností pri úprave videa. Ako prvé je potrebné vytvoriť rôzne verzie videa, teda s rôznym rozlíšením a rôznou rýchlosťou. V ukážke pod textom sú zobrazené použité príkazy v príkazovom riadku systému Linux Ubuntu. Ďalej je potrebné vytvoriť zvukovú stopu k už vytvoreným verziám videa. V prvom kroku ukážky máme zdroj videa s názvom video.mp4 a zdroj zvuku s názvom audio.wav, vytvoríme päť video streamov a jeden audio stream každý s rôznym rozlíšením a prenosovou rýchlosťou. Vstupné súbory môžu mať akýkoľvek formát ktorý podporuje FFmpeg.

```
VP9_DASH_PARAMS="-tile-columns_4_-frame-parallel_1"
```

```
ffmpeg -i video.mp4 -c:v libvpx-vp9 -s 160x90 -b:v 250k  
-keyint_min 150 -g 150 ${VP9_DASH_PARAMS} -an -f webm  
-dash 1 video_160x90_250k.webm
```

parameter -i: určenie vstup

parameter -c:v libvpx-vp9 : určenie kodeku VP9

parameter -s 160x90 určenie rozlíšenia

parameter -b:v 250k rychlosť videa 250 kbit/s

parameter -keyint\_min 150 -g 150 určuje dĺžku skupiny snímkov

parameter -an zakáže nahrávanie zvuku

parameter -f webm -dash 1 formát výstupného súboru

video\_160x90\_250k.webm je názov výstupného súboru



```
ffmpeg -i video.mp4 -c:v libvpx-vp9 -s 320x180 -b:v 500k
-keyint_min 150 -g 150 ${VP9_DASH_PARAMS} -an -f webm
-dash 1 video_320x180_500k.webm
```

```
ffmpeg -i video.mp4 -c:v libvpx-vp9 -s 640x360 -b:v 750k
-keyint_min 150 -g 150 ${VP9_DASH_PARAMS} -an -f webm
-dash 1 video_640x360_750k.webm
```

```
ffmpeg -i video.mp4 -c:v libvpx-vp9 -s 640x360 -b:v 1000k
-keyint_min 150 -g 150 ${VP9_DASH_PARAMS} -an -f webm
-dash 1 video_640x360_1000k.webm
```

```
ffmpeg -i video.mp4 -c:v libvpx-vp9 -s 1280x720 -b:v 1500k
-keyint_min 150 -g 150 ${VP9_DASH_PARAMS} -an -f webm
-dash 1 video_1280x720_500k.webm
```

```
ffmpeg -i audio.wav -c:a libvorbis -b:a 128k -vn -f webm
-dash 1 audio_128k.webm
```

Ďalšou dôležitou časťou je vytvorenie manifest file ktorý používajú všetky technológie adaptívneho streamingu, používajú kombináciu kódovaných súborov a manifest súborov, ktoré identifikujú alternatívne streamy a ich príslušne URL (Uniform Resource Locator) adresy. WebM DASH manifest súbor je XML súbor ktorý môžeme pomocou nástroja ffmpeg vytvoriť nasledovne:

```
ffmpeg \
-f webm_dash_manifest -i video_160x90_250k.webm \
-f webm_dash_manifest -i video_320x180_500k.webm \
-f webm_dash_manifest -i video_640x360_750k.webm \
-f webm_dash_manifest -i video_640x360_1000k.webm \
-f webm_dash_manifest -i video_1280x720_500k.webm \
-f webm_dash_manifest -i audio_128k.webm \
-c copy -map 0 -map 1 -map 2 -map 3 -map 4 -map 5 \
-f webm_dash_manifest \
-adaptation_sets "id=0,streams=0,1,2,3,4,id=1,streams=5" \
manifest.mpd
```

Čím vytvoríme súbor ktorý obsahuje informácie o jednotlivých videách a priradí ich k hodnotám šírky pásma tak, aby vytvorený prehrávač podľa aktuálnej šírky pásma mohol vybrať video vo vhodnej kvalite.

Takto máme vytvorené potrebné súbory na adaptívne streamovanie máme teda päť rôznych verzií videa jednu audio verziu a manifest súbor. V ďalšej časti vytvoríme webovú stránku s prehrávačom.

## 5.2 Implementácia vytvoreného prehrávača na webovú stránku

V tejto časti vytvoríme jednoduchú webovú stránku na ktorej bude implementovaný adaptívny prehrávač MPEG – DASH s použitím dash.webm.min JavaScriptu. Výsledný prehrávač s prehrávaním videa v malom rozlíšení je zobrazený na obr. 5.1, a prehrávanie videa vo veľkom rozlíšení je zobrazený na obr. 5.2.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Semetralny projekt Matus Lipa </title>
    <script src="dash.webm.min.js"></script>
  </head>
```

Vytvorenie hlavičky stránky a definovanie zdroju použitého JavaScriptu.

```
<body>
  <style>
    body {
      font-family:'Times';
      background-color:#A9F6BC;}
  </style>
```

Nastavenie použitého písma a farby pozadia webovej stránky.

```
<script>
function startVideo(mpd)
{
  var video, context, player;
  video = document.querySelector('video');
  context = new Webm.di.WebmContext();
  player = new MediaPlayer(context);

  player.startup();
```

```

        player.attachView(video);
        player.attachSource(mpd);
        player.setAutoPlay(false);
    }
</script>

```

V tejto časti sa vytvorí samotný prehrávač, načíta sa mpd súbor, nastaví sa možnosť automatického spustenia videa pri načítaní stránky.

```

<body onload="startVideo('manifest.mpd');">
  <video id="video"
    width="640" height="480"
    class="dash-video-player" preload controls> </video>
</body>
</html>

```

Nakoniec sa načíta vytvorený prehrávač, definuje sa veľkosť video elementu a nastaví sa funkcie na načítanie videa pri načítaní stránky a pridanie kontrolných tlačidiel.

Vytvorený prehrávač pre adaptívne streamovanie je schopný sám podľa aktuálnej dostupnej šírky pásma automaticky reagovať na zmenu a sám vie v prípade zmeny v sieti reagovať zmenou kvality videa na lepšiu alebo horšiu kvalitu. V praxi sa táto technika streamovania používa hlavne väčších a rozšírenejších službách. Nakoľko tento prehrávač reaguje na zmeny v sieti hneď nieje možné ho nijakým spôsobom ovládať.

Preto sa v ďalšia časť zaoberá vytvorením prehrávača ktorý je možné ovládať podľa vytvorených scenárov a tým simulovať prehrávač MPEG – DASH čo bude mať prínos pre meranie subjektívneho hodnotenia streamovaného videa. Takto vytvorený prehrávač bude možné umiestniť priamo do komplexného nástroja ktorý umožní meranie kvality zážitku zo streamovaného videa.



Obr. 5.1: Obraz prehrávaného videa v malom rozlíšení



Obr. 5.2: Obraz prehrávaného videa vo veľkom rozlíšení

## 6 VYTVORENIE PREHRÁVAČA NA MERA- NIE QOE

V predchádzajúcej kapitole bol vytvorený prehrávač adaptívneho streamu MPEG – DASH ktorý automaticky reaguje na náhle zmeny v sieti prepnutím kvality no nie je možné ho ovládať. V tejto časti bakalárskej práce je popísaný postup vytvorenia prehrávača videa ktorý umožňuje dynamické prepínanie kvality streamovaného videa počas sledovania a nastavenie času načítavania medzi prepnutím jednotlivých kvalít. Vytvorený prehrávač sa riadi podľa vytvorených scenárov ktorých vytvorenie je popísané nižšie. Prehrávač vytvorený v tejto kapitole s pomocou vytvorených scenárov umožňuje simulovať zmeny v sieti a podľa scenárov prepínať kvalitu v danom čase na lepšiu alebo horšiu a zároveň nastavenie času načítavania pri prepnutí kvality. Prehrávač neumožňuje užívateľovi presúvať sa vo videu ani prepínať kvalitu videa kôli vytvoreniu prostredia vhodného na hodnotenie je skrytý tzv. progres bar ktorý umožňuje posúvať sa vo videu. Prehrávač bude umiestnený v komplexnom nástroji na subjektívne hodnotenie kvality zážitku zo streamovaného videa.

### 6.1 Príprava potrebných dát

Vzhľadom k tomu že prehrávač bude dynamicky prepínať kvalitu videa ako prvé je potrebné vytvoriť obsah, v našom prípade video vo viacerých kvalitách. Použitý bol nástroj FFmpeg ktorý je jeden s najlepších a najuniverzálnejších voľných nástrojov a jeho detailnejší popis je v kapitole 3.3. Celý popis vytvorenia zdroju dát vo viacerých kvalitách je podrobne popísaný pri vytvorení prehrávača v Semestrálnej práci v kapitole 5.1. Ak už sú dáta pripravené je možné prejsť na ďalší krok ktorým je vytvorenie prehrávača ktorý je podrobne popísaný v ďalšej kapitole. Na obrázku 6.1 je zobrazená bloková schéma behu prehrávača.

### 6.2 Inštalácia potrebných balíčkov pre vytvorenie prehrávača

Pre vytvorenie prehrávača bol vybraný VIDEOJS je to univerzálny prehrávač ktorý umožňuje pridanie mnoho rozšírení. Existuje niekoľko možných riešení ako začať používať Videojs prehrávač. Pre pokročilé riešenie a lepšiu škálovateľnosť som si vybral riešenie pomocou balíčkovacieho systému NPM (Node Package Manager). Pred začatím práce je nutné si nainštalovať NODE.js a Grunt. Pre inštaláciu boli použité nasledovné príkazy v príkazovom riadku systému Windows, pri zadávaní

príkazov je potrebné sa premiestniť do aktuálneho priečinku kde prehrávač zostavujeme načo slúži príkaz `cd`. Ak sme v správnom priečinku môžeme použiť nasledujúce príkazy:

```
npm install video.js
```

Nainštalovanie všetkých potrebných súvislostí pre použitie Videojs.

```
npm install -g grunt-cli
```

Nainštalovanie spúšťaču úloh.

```
npm i videojs-resolution-switcher
```

Inštalácia pluginu ktorý slúži na prepínanie kvalít videa.

```
npm install
```

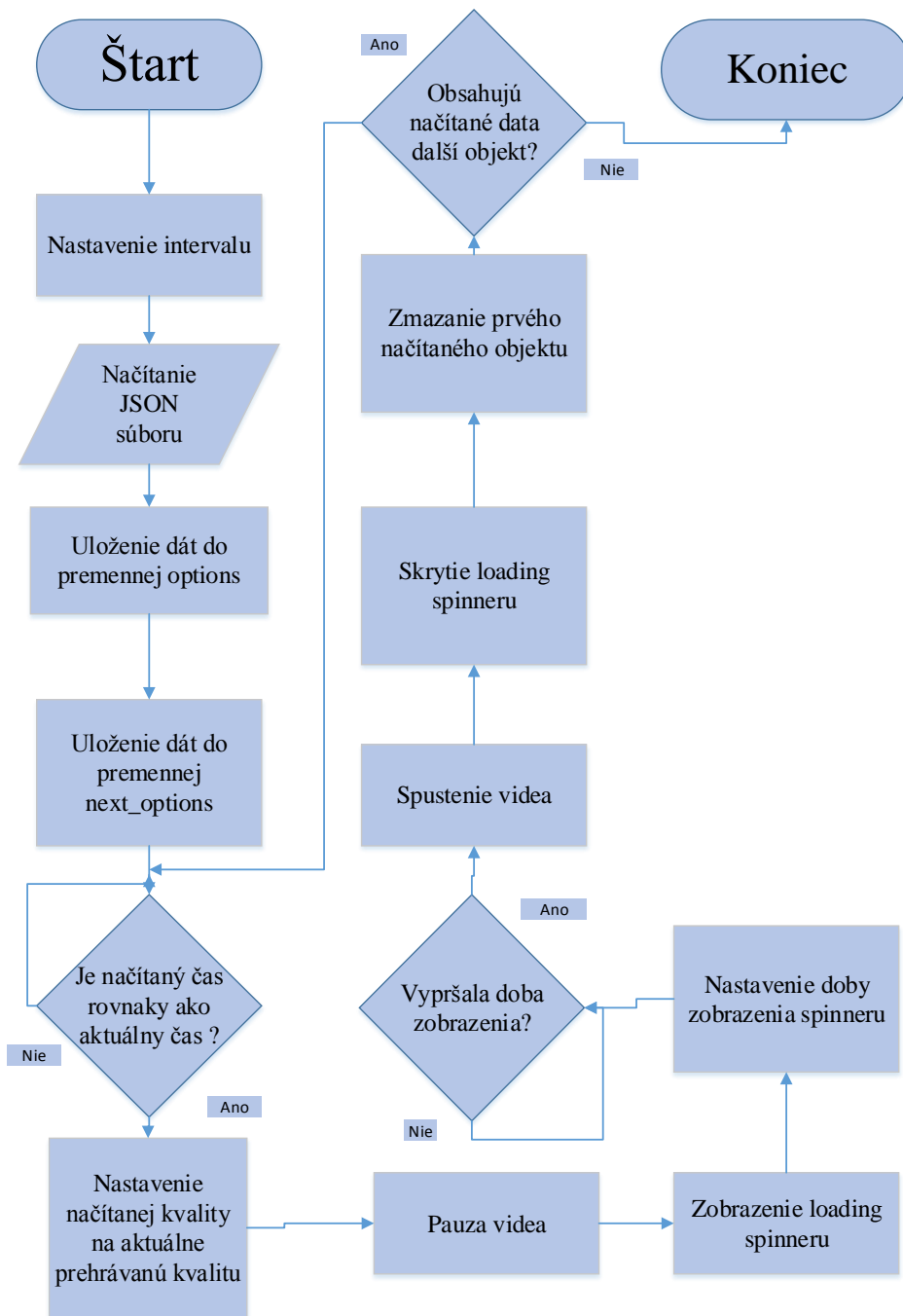
Nainštalovanie všetkých ostatných potrebných závislostí.

V tejto chvíli je nainštalované všetko potrebné na to aby bolo možné vytvoriť prehrávač.

## 6.3 Vytvorenie prehrávača a súborov k tomu potrebných

Ako prvé potrebné vytvoriť video element pre prehrávač ktorý sa bude ďalej rozvíjať. Ďalej je potrebné uviesť cestu k potrebným súborom ktoré budeme využívať ako napríklad Javacriptové súbory ktoré sa ktoré sa využívajú pri stavbe základného prehrávača alebo slúžia pri použití rozšírenia. Ďalšími potrebnými súbormi sú CSS súbory ktoré slúžia na štýlovanie vzhľadu prehrávača, na zobrazenie alebo skrytie rozšírení a celkovo na dizajn toho ako bude daný vytvorený prehrávač vyzeráť. Pri vytvorení prehrávača je nutné vytvoriť dva vlastné CSS súbory. Prvý s názvom `style.css` ktorý nastavuje výšku a šírku video elementu, skrýva nepotrebné časti ako zvyšný čas do konca videa a časovú os prehrávača s ktorou je možné pretáčať sledované video nakoľko to pri meraní kvality zážitku nieje vhodné, a zároveň tento súbor umožňuje zobrazenie častí ktoré v prehrávači chceme ako je aktuálny čas prehrávaného videa. Súbor `style.css` je uvedený v prílohe A.2.

Na obrázku 6.1 pod textom je zobrazená bloková schéma vytvoreného prehrávača. Podrobne je zobrazený beh programu ako prehrávač inicializuje potrebné súbory, aké sú momenty rozhodovania a vykonané príkazy.



Obr. 6.1: Bloková schéma prehrávača

## Vytvorenie loading spinneru

Znak načítavania videa v prehrávači VIDEOJS je nazvaný loading spinner. Na koľko originálny loading spinner má striktné dané situácie pri ktorých sa zobrazí a skryje, je veľmi obtiažne ho použiť. Preto v tejto práci pri vytvorení upraveného prehrávača na subjektívne hodnotenie kvality zážitku je potrebné vytvoriť vlastný loading spinner ktorý je možné použiť kedykoľvek. V tejto práci sa používa k simulovaniu načítavania videa, zobrazuje na nastavenú dobu podľa scenára pri prepnutí kvality a je ho možné zobraziť aj keď sa nezmení kvalita. Celý obsah vytvoreného spinneru je uvedený v prílohe A.3:

## Vytvorenie scenáru

V tejto práci sa na vytvorenie scenárov využíva JSON súbor nazvaný data.json, v ktorom sú pomocou poľa objektov zadané hodnoty ktorými sa bude prehrávač pri prepínaní kvality riadiť. Súbor v tejto práci definuje tri premenné, „t“ ktorá udáva čas v sekundách v ktorom sa vykoná prepnutie do inej kvality, hodnota „q“ udáva kvalitu ktorú v daný čas t začne prehrávač prehrávať, a hodnotu „l“ ktorá udáva koľko sekúnd po prepnutí kvality bude prehrávač zastavený a bude tváriť že načítava video čím sa simuluje pomalé pripojenie na Internet a video sa nestíha načítavať. Výsledný súbor môže vyzeráť takto:

```
[
  {"t": 13, "q": "90p", "l": 5},
  {"t": 18, "q": "360p", "l": 7},
  {"t": 25, "q": "1080p", "l": 4},
  {"t": 45, "q": "90p", "l": 0}
]
```

Na vloženie cesty k Javascriptu sa používa HTML tag „<script>“ v ktorom sa definuje cesta k danému súboru pomocou src. Na vloženie cesty k CSS súborom sa používa HTML tag link v ktorom sa definuje cesta k danému súboru. Nastavenie cesty k potrebným skriptom je v prílohe A.1.

Keď sú všetky cesty k potrebným súborom zadané a okno pre video prehrávač vytvorené je možné už pokračovať priamo k vytvoreniu funkcionality prehrávaču. Pre vytvorenie funkcionality sa používa HTML tag script. Ako prvá bola vytvorená premenná options do ktorej sa ďalej ukladajú hodnoty načítané z JSON súboru. Následne je zavolaná metóda getJSON ktorá načíta JSON súbor a hodnoty uloží do premennej options A.4.



## Inicializácia prehrávaču

Následne sa vytvorí prehrávač v ktorom sa definujú jeho základné vlastnosti, názov, zobrazenie tlačítok na ovládanie prehrávača, zakáže sa automatické spustenie prehrávača po načítaní stránky, a nastaví sa hodnota na možnosť načítania videa pred jeho spustením. Pri použití pluginov pre rozšírenie možností používania prehrávača sa definujú použité pluginy. V tejto práci bol použitý plugin videoJsResolutionSwitcher ktorého základná hodnota bola nastavená na 360 čo určuje, že kvalita ktorú prehrávač načíta ako prvú kvalitu je s rozlíšením 360 pixelov celé nastavenie je v prílohe A.5 .

```
var player = videojs('video',
{
    controls: true,
    autoplay: false,
    preload: "auto",
    plugins:
    {
        videoJsResolutionSwitcher:
        {
            default: 360,
            dynamicLabel: false
        }
    }
},
```

## Určenie dostupných zdrojov

Nasledujúcim krokom je vytvorenie funkcie na prepínanie aktuálne prehrávaného videa, pričom ako prvé je nutné zavolať metódu pridá jednotlivé zdroje videa ktoré je možné prehrávať, pričom je nutné definovať cestu k videu, typ, zväzok podľa ktorého sa bude prepínať kvalita a rozlíšenie. Postup je uvedený v prílohe A.6

```
function () {
    player.updateSrc([
        {
            src: 'videos/video_1280x720_500k.webm',
            type: 'video/webm',
            label: '720p',
            res: 720
        }
    ],
```

```

    {
        src: 'videos/video_640x360_750k.webm',
        type: 'video/webm',
        label: '360p',
        res: 360
    },
    {
        src: 'videos/video_160x90_250k.webm',
        type: 'video/webm',
        label: '90p',
        res: 90
    }
]);}

```

## Vytvorenie hlavných funkcií na prepínanie videa

Poslednou a zároveň najdôležitejšou časťou je vytvorenie funkcie ktorá zabezpečí najdôležitejšiu časť v prehrávači. Jedná sa o funkciu ktorá nastaví interval opakovania programu. Ďalej uloží prvý objekt z načítaného súboru ktorý sme uložili do premennej options. Následnej porovná načítanú hodnotu času s aktuálnou pozíciou času prehrávaného videa prepnutie do správnej kvality v správny čas. Ak je splnená podmienka času, zavolá sa metóda ktorá nastaví kvalitu podľa hodnoty ktorá je zadaná v premennej q. Následne sa zavolá metóda ktorá obsahuje funkciu ktorá pozastaví prehrávané video, zobrazí spinner teda koliesko ktoré značí načítavanie videa. Po uplynutí hodnoty l ktorú sme zadali v JSON súbore sa zavolá metóda na opätovné spustenie videa a odstráni sa koliesko načítavania. Ako posledné je nutné zmazať prvý načítaný objekt metódou shift aby bolo možné proces opakovať pri ďalšej zmene kvality kód je uvedený v prílohe A.7 .

```

setInterval(function ()
{
    var nextOptions = options[0];
    if (nextOptions && nextOptions.t < player.currentTime())
    {
        player.currentResolution(nextOptions.q);
        player.one('resolutionchange', function ()
        {
            player.pause();
            player.addClass('custom-seeking');
            setTimeout(function ()

```

```

        {
            player.play();
            player.removeClass('custom-seeking');
        }, nextOptions.l * 1000);
    }); options.shift();
}
}, 500);
});

```

V tejto časti bol vytvorený kompletný videoprehrávač určený pre komplexný nástroj na subjektívne hodnotenie kvality streamovaného videa. Prehrávač MPEG – DASH automaticky reaguje na zmeny v sieti zmenou kvality na horšiu alebo lepšiu. Prehrávač vytvorený v tejto práci spolu s vytvorenými scenármi dokáže simulovať správanie prehrávaču MPEG – DASH. Vytvorené scenáre umožňujú nastaviť dobu prehrávania videa v jednotlivých kvalitách, umožňujú prepnutie do ľubovolnej kvality a nastavenie doby načítavania pri prepnutí medzi jednotlivými kvalitami. S týmito vlastnosťami je možné simulovať sieť s veľkou aj malou šírkou pásma. Prehrávač je vytvorený pre umiestnenie do komplexného nástroja pre subjektívne hodnotenie kvality streamovaného videa.

## ZÁVER

V tejto práci sú popísané jednotlivé techniky streamovania videa od jednoduchých „tradičných“ postupne až po pokročilé techniky adaptívneho streamovania ktoré začínajú nahradzovať staršie techniky.

Podrobnejšie boli rozobraté jednotlivé techniky adaptívneho streamovania, ktoré sa v dnešnej dobe využívajú. K nim patria aj prehrávače určené na prehrávanie adaptívneho streamu, ktoré majú k dispozícii predpripravený zdroj streamu tak, aby klientovi mohli ponúknuť čo najlepšiu variantu.

V prvej praktickej časti boli vytvorené všetky časti potrebné k adaptívnemu streamovaniu. Bol pripravený zdroj videa v niekoľkých variantách s rôznym rozlíšením a bitovou rýchlosťou, súbor manifest potrebný pre prehrávač a jednoduchá webová stránka na ktorú bol umiestnený prehrávač adaptívneho streamu DASH – MPEG.

V druhej praktickej časti bol vytvorený prehrávač určený pre nástroj na meranie kvality zážitku, pričom bol použitý zdroj videa v niekoľkých variantách ktorý bol vytvorený v predchádzajúcej časti. Vytvorený videoprehrávač bol upravený tak aby bolo možné vytvoriť scenár ktorý umožňuje prehrávaču prikázať zmenu kvality v daný čas a simulovať načítavanie po dobu zadanú v scenári. Vytvorený prehrávač je teda schopný simulovať zmeny v sieti podľa vytvoreného scenára. Tento prehrávač je umiestnený v nástroji pre subjektívne hodnotenie kvality streamovaného videa.

# LITERATÚRA

- [1] *Adaptive bitrate Streaming*. Wikipedia. 2015. Online: [https://en.wikipedia.org/wiki/Adaptive\\_bitrate\\_streaming#Adobe\\_Dynamic\\_Streaming\\_for\\_Flash](https://en.wikipedia.org/wiki/Adaptive_bitrate_streaming#Adobe_Dynamic_Streaming_for_Flash)
- [2] Avidemux in Launchpad. DIAS BUGARIN, Tiago. *Avidemux is a free video editor designed for simple cutting, filtering and encoding tasks*. [online]. 2006 [cit. 2015-12-10]. Dostupné z: <https://launchpad.net/avidemux>
- [3] A. Takashashi *Applications of objective quality assessment models in IPTV*. tech.rep., NTT Service Integration Labs, 2007
- [4] CISCO. *Cisco visual networking index: Global mobile data traffic forecast update, 2014–2019*. [online]. tech.rep., 3.2.2015. Dostupné z: [http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white\\_paper\\_c11-520862.html](http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html).
- [5] Download IIS Smooth Streaming Technical Overview from Official Microsoft Download Center. ZAMBELLI, Alex. MICROSOFT CORPORATION. *IIS Smooth Streaming Technical Overview* [online]. 2009 [cit. 2015-11-18]. Dostupné z: <http://www.microsoft.com/en-us/download/details.aspx?id=17678>
- [6] DYER, John. MediaElement.js - HTML5 video player and audio player with Flash and Silverlight shims. *How It Works* [online]. [cit. 2015-12-10]. Dostupné z: <http://mediaelementjs.com/>
- [7] Dynamic Adaptive Streaming over HTTP (DASH). TIMMERER, Christian a Christopher MÜLLER. *Dynamic Adaptive Streaming over HTTP (DASH)* [online]. 03.05.2011 [cit. 2015-11-19]. Dostupné z: <http://www.slideshare.net/christian.timmerer/dynamic-adaptive-streaming-over-http-dash>
- [8] FFmpeg. *FFmpeg Documentation* [online]. [cit. 2015-11-15]. Dostupné z: <http://www.ffmpeg.org/documentation.html>
- [9] GARDLER, Ross. MPEG-DASH Tutorial: Embedding an adaptive streaming video within your HTML5 application. In: *What is MPEG-DASH and dash.js?* [online]. 2014 [cit. 2015-12-06]. Dostupné z: <http://blogs.msdn.com/b/interopability/archive/2014/01/03/mpeg-dash-tutorial-embedding-an-adaptive-streaming-video-within-your-html5-application.aspx>

- [10] *HTTP Live Streaming Overview*. Apple inc. 2014. Online: <https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/Introduction/Introduction.html>
- [11] ITU-T *ITU-T recommendation X.920: Information technology - Open Distributed Processing - Reference Model: Foundations*. tech.rep., ITU-T, 2009 Dostupné z: <https://www.itu.int/rec/T-REC-X.902/en>
- [12] jPlayer : HTML5 Audio and Video for jQuery. HAPPYWORM. [online]. 2015 [cit. 2015-12-10]. Dostupné z: <http://jplayer.org/>
- [13] KUIPERS, Fernando, Robert KOOIJ, Danny DE VLEESCHAUWER a Kjell BRUNNSTRÖM. *Techniques for Measuring Quality of Experience*. [online]. 2015 [cit. 2016-04-20]. Dostupné z: <https://www.nas.ewi.tudelft.nl/people/Fernando/papers/WWIC2010.pdf>
- [14] NĚMEC, Luboš. Recenze - Sony Vegas Pro 11. *Sony Vegas Pro 11* [online]. 2011 [cit. 2015-12-10]. Dostupné z: <http://www.video-zone.cz/recenze/sony-vegas-pro-11-t82.html>
- [15] OZER, Jan. What is Adaptive Streaming - Streaming Media Magazine. *What is Adaptive Streaming?* [online]. 2011 [cit. 2015-12-06]. Dostupné z: <http://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-is-Adaptive-Streaming-75195.aspx>
- [16] RICHARDSON, Iain. H.264 A Technical Introduction. In: *A Technical Introduction to H.264 / AVC* [online]. 2013 [cit. 2015-12-06]. Dostupné z: <http://www.vcodex.com/images/uploaded/342454811635986.pdf>
- [17] ROBERTSON, Steven. YouTube Engineering and Developers Blog: VP9: Faster, better, buffer-free YouTube videos. *VP9: Faster, better, buffer-free YouTube videos* [online]. 2015 [cit. 2015-12-06]. Dostupné z: <http://youtube-eng.blogspot.cz/2015/04/vp9-faster-better-buffer-free-youtube.html>
- [18] S. I. ULC. *Global internet phenomena report: 1H 2014*. [online .tech.rep., 15.5.2014. Dostupné z: <http://www.ntt.co.jp/qos/qoe/eng/technology/visual/index.html>
- [19] VAJSAR, Pavel. *Směrovací protokol pro síť MANET zohledňující požadavky na kvalitu služeb*. Brno: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií, 2014.

- [20] Video quality assessment methods. *Communications Service QoE Assessment Research Portal Site*. [online]. [cit. 2016-03-13]. Dostupné z: <http://www.ntt.co.jp/qos/qoe/eng/technology/visual/index.html>
- [21] ZBONČÁK, Jozef a Matej KAVACKÝ. *Kvalita služby (QoS) a Kvalita vnímania (QoE)* [online]. 2011 [cit. 2016-04-10]. Dostupné z: <http://www.posterus.sk/?p=11948>

## ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

RTSP	Real Time Streaming Protocol
RTMP	Real Time Messaging Protocol
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
MMS	Microsoft Media Server
HTTP	Hypertext Transfer Protokol
MPEG – DASH	Dynamic Adaptive Streaming over HTTP
URL	Uniform Resource Locator
MSE	Media Source Extensions
QoE	Quality of Experience
QoS	Quality of Services
MOS	Mean Opinion Score
HVS	Human Visual System
QoE	Quality of Experience
NPM	Node Package Manager



## ZOZNAM PRÍLOH

A	Jednotlivé časti vytvoreného prehrávača	50
B	Obsah priloženého CD	55

## A JEDNOTLIVÉ ČASTI VYTVORENÉHO PREHRÁVAČA

Výpis A.1: Nastavenie cesty k potrebným skriptom

```
<video id="video" class ="video-js vjs-default-skin
vjs-big-play-centered" poster="img/vlk.jpg"></video >
<link href="node_modules/video.js/dist/video-js.css"
rel="stylesheet">
<link href="node_modules/videojs-resolution-switcher/lib/
videojs-resolution-switcher.css" rel="stylesheet">
<script src="node_modules/video.js/dist/video.js"></script>
<script src="node_modules/videojs-resolution-switcher/lib/
videojs-resolution-switcher.js"></script>
<script type="text/javascript" src="node_modules/jquery/dist/
jquery.js"></script>
<script type="text/javascript" src="data.json"></script>
<link href="style.css" rel="stylesheet">
<link href="spinner.css" rel="stylesheet">
```

Výpis A.2: Súbor style.css

```
.video-js
{
  height: 400px;
  width: 600px;
}
.video-js .vjs-control-bar .vjs-resolution-button {
  display: none;
}
.vjs-control-bar .vjs-progress-control.vjs-control {
  display: none;
}
.video-js .vjs-current-time {
  display: block;
}
.video-js .vjs-remaining-time {
  display: none;
}
```

### Výpis A.3: Loading Spinner

```
.vjs-loading-spinner
{
    display: none;
    position: absolute;
    top: 50%;
    left: 50%;
    margin: -25px 0 0 -25px;
    opacity: 0.85;
    text-align: left;
    border: 6px solid rgba(43, 51, 63, 0.5);
    box-sizing: border-box;
    background-clip: padding-box;
    width: 50px;
    height: 50px;
    border-radius: 25px;
}
.custom-seeking .vjs-loading-spinner
{
    display: block;
}
.vjs-loading-spinner:before,
.vjs-loading-spinner:after
{
    content: "";
    position: absolute;
    margin: -6px;
    box-sizing: inherit;
    width: inherit;
    height: inherit;
    border-radius: inherit;
    opacity: 1;
    border: inherit;
    border-color: transparent;
    border-top-color: white;
}
.custom-seeking .vjs-loading-spinner:before,
.custom-seeking .vjs-loading-spinner:after
{
    -webkit-animation: vjs-spinner-spin 1.1s
```

```

    cubic-bezier(0.6, 0.2, 0, 0.8) infinite,
    vjs-spinner-fade 1.1s linear infinite;
    animation: vjs-spinner-spin 1.1s
    cubic-bezier(0.6, 0.2, 0, 0.8) infinite,
    vjs-spinner-fade 1.1s linear infinite;
}
.custom-seeking .vjs-loading-spinner:before
{
    border-top-color: rgb(255, 255, 255);
}
.custom-seeking .vjs-loading-spinner:after
{
    border-top-color: rgb(255, 255, 255);
    -webkit-animation-delay: 0.44s;
    animation-delay: 0.44s;
}
@keyframes vjs-spinner-spin {
    100% { transform: rotate(360deg); }
}
@-webkit-keyframes vjs-spinner-spin
{
    100% { -webkit-transform: rotate(360deg);}
}
@keyframes vjs-spinner-fade
{
    0% { border-top-color: ghostwhite; }
    20% { border-top-color: ghostwhite;}
    35% { border-top-color: white;}
    60% { border-top-color: ghostwhite;}
    100% {border-top-color: ghostwhite;}
}
@-webkit-keyframes vjs-spinner-fade
{
    0% { border-top-color: ghostwhite; }
    20% { border-top-color: ghostwhite; }
    35% { border-top-color: white;}
    60% {border-top-color: ghostwhite; }
    100% {border-top-color: ghostwhite; }
}

```

#### Výpis A.4: Načítanie JSON súboru

```
<script>
var options;
$.getJSON("data.json", function (json) {
    options = json;
});
```

#### Výpis A.5: Definovanie prehrávača

```
var player = videojs('video',
{
    controls: true,
    autoplay: false,
    preload: "auto",
    plugins:
    {
        videoJsResolutionSwitcher:
        {
            default: 360,
            dynamicLabel: false
        }
    }
},
```

#### Výpis A.6: Nastavenie zdrojov videa

```
function () {
    player.updateSrc([
        {
            src: 'videos/video_1280x720_500k.webm',
            type: 'video/webm',
            label: '720p',
            res: 720
        },
        {
            src: 'videos/video_640x360_750k.webm',
            type: 'video/webm',
            label: '360p',
            res: 360
        },
        {
            src: 'videos/video_160x90_250k.webm',
            type: 'video/webm',
```

```
        label: '90p',
        res: 90
    }
    ]);
</script>
```

#### Výpis A.7: Funkcia na prepínanie kvality

```
setInterval(function ()
{
    var nextOptions = options[0];
    if (nextOptions && nextOptions.t < player.currentTime())
    {
        player.currentResolution(nextOptions.q);
        player.one('resolutionchange', function ()
        {
            player.pause();
            player.addClass('custom-seeking');
            setTimeout(function ()
            {
                player.play();
                player.removeClass('custom-seeking');
            }, nextOptions.l * 1000);
        }); options.shift();
    }
}, 500);
});
```

## B OBSAH PRILOŽENÉHO CD

Na priloženom CD sú všetky potrebné súbory pre zostavenie plne funkčného prehrávača určeného pre subjektívne hodnotenie kvality zážitku zo streamovaného videa.

/ .....	korenový adresár priloženého CD
├── img .....	úvodné logo prehrávača
├── videos .....	Používané videá
├── node_modules .....	Potrebné javascripty
├── data.json .....	Vytvorený scenár
├── index.html .....	Webová stránka s prehrávačom
├── package.json .....	Konfiguračný súbor
├── spinner.css .....	CSS subor na vytvorenie spinneru
└── style.css .....	CSS súbor pre upravu prehrávača