A Comparative Study of DASH Representation Sets Using Real User Characteristics

Christian Kreuzberger, Benjamin Rainer, Hermann Hellwagner
Institute of Information Technology (ITEC)
Alpen-Adria-Universität Klagenfurt, Austria
firstname.lastname@itec.aau.at

Laura Toni, Pascal Frossard
Signal Processing Laboratory LTS4
EPFL, Switzerland
firstname.lastname@epfl.ch

ABSTRACT

Adaptive streaming strategies over HTTP allow to serve heterogeneous video users with varying demands. By providing different encoded versions (representations) of each video sequence on the server, clients have the freedom to select a representation that best fits their needs. While the topic of selecting a representation based on a pre-defined set is covered very well in the literature, the problem of how to properly select the representation set stored at the main server is usually an overlooked challenge. In this work, we provide an analysis on how the choice of representations on the server impacts the clients' quality. This is achieved by conducting *NS-3* based simulations with a total of 10k users and up to 300 *concurrent* DASH clients for several recommended sets (e.g., Netflix, YouTube, and Apple), and measuring the experienced quality over a timespan of 24 hours. The results show that under heavy load (at peak hours) there is still room for improvement.

CCS Concepts

• **Information systems** → **Multimedia streaming; Multimedia content creation;** • **Social and professional topics** → *System management; Network operations;*

Keywords

Dynamic Adaptive Streaming over HTTP; Representations; Encoding

1. INTRODUCTION

While multimedia streaming is becoming more and more popular, the large amount of data caused by multimedia streaming needs to be efficiently delivered to highly heterogeneous clients over resource-limited networks. To reach this goal, streaming technologies need to be *i*) highly adaptive to both users and network conditions, and *ii*) highly scalable with the size of the user population. HTTP-based adaptive streaming technologies are able to cover both key aspects, as demonstrated by the recently standardized MPEG-DASH (Dynamic Adaptive Streaming over HTTP) [14], henceforth denoted as DASH.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

NOSSDAV '16 May 13, 2016, Klagenfurt, Austria

© 2016 ACM. ISBN 123-4567-24-567/08/06…$15.00

DOI: http://dx.doi.org/10.1145/2910642.2910647

In a typical DASH system (Figure 1), each video is encoded at different bit rates, quality levels, and resolutions. Each encoded version, also called *representation*, is stored at the server, and split into temporally successive and equal size *segments* (e.g., two seconds). For each video, available representations are described in a so called *Media Presentation Description* (MPD) file. The segments can be decoded independently (self-contained), enabling clients to dynamically switch between different representations at segment boundaries. The decision of selecting an appropriate representation is carried out by each client independently and implemented in the so called *adaptation logic*.

Serving many users on the Internet with a good quality of experience (i.e., no playback stalls, high video quality) is still an open issue. For example, Conviva [4] reported that in 2014, 28.8% of streaming sessions were affected by video playback stalls (*buffering*) and 58.4% received a bad video quality. During the last years, tremendous effort has been devoted to analyzing and solving these issues on the client side. However, only few works consider using more than a single recommended representation set. Often a very small number of concurrent clients (usually 1 to 5) is considered, if at all, resulting in a very static workload for the network/server. In this work we take the content provider's perspective and study the impact of different recommended representation sets on a dynamic workload with up to 300 concurrent clients with various demands and device characteristics. The objective of this paper is to investigate how representation sets used by YouTube, Netflix, and Apple perform against an optimized choice of representations [16] in a multimedia streaming scenario with a large user base and a heterogeneous set of devices (e.g., smartphones, tablets, HD TVs, Full HD TVs).

To achieve this, we implemented an HTTP server and DASH clients in *NS-3* [7], a discrete-event network simulator. For our user base, we generated a 24-hour streaming scenario based on YouTube traces [12] and device statistics for Hulu and Netflix [10]. As metric we selected *user satisfaction* [16], modeled by Structural Similarity (SSIM) [18] based on the devices' spatial screen resolutions.

The contributions of this paper are as follows:

- we empirically assess the bit rates and resolutions of videos hosted by YouTube;
- we investigate the behavior of different recommended and optimized representation sets of Netflix, YouTube, Apple, and those obtained by solving the ILP from [16]

under a scenario with real traffic traces;
- we further provide the source code used to conduct the simulations.

We will show that the recommended representation sets differ significantly in their performance with respect to user satisfaction. We further show that the optimized representations obtained by solving the ILP given in [16] do provide an improvement in comparison to the recommended representation sets, but there is still room for improvement.

The remainder of this paper is organized as follows. A brief overview of related work is provided in Section 2. Section 3 explains the basics of our evaluation setup, including NS-3 simulations, the user population, as well as the recommended representations used. We discuss results of the evaluations in Section 4 and conclude the paper in Section 5.

2. RELATED WORK

Adaptive streaming is an active research area, especially when considering DASH [14]. However, efforts for improving users' multimedia streaming experience are focused on the content distribution and the client side (e.g., adaptation logics). Despite the growing interest in studying the provider's side of the problem [15, 2, 21, 5], only few works [15, 16] consider the impact of different representation sets used. Remaining works usually consider one pre-encoded recommended representation set for their evaluations.

In [15], the authors study adaptive streaming systems for live video streaming and claim that different representation sets may affect the behaviors of adaptation logics. How to efficiently create the representation set has been investigated in [2, 21]. However, [2] investigates the efficiency of transcoding operations in the cloud for live video streaming applications, while [21] optimizes the subset of representations that should be cached in the network. Finally, DASH from a provider's perspective is analyzed also in [5]. The authors do not focus on the representation set design; rather, they study the provider's gain in re-shaping users' requests.

Recently, Toni et al. [16] introduced the problem of determining *optimal representation sets* based on client and network settings. The authors formulated an Integer Linear Program (ILP) for finding an optimal representation set with respect to a given satisfaction function. They compared the resulting representation set with recommended ones (YouTube, Netflix and Apple) and showed the gain of the optimal set in terms of both, users' satisfaction and storage constraints. However, the analysis of the system performance is mainly static and theoretical. The dynamics of users joining and leaving and the impact of concurrent users on the CDN are not considered. Furthermore, an experimental evaluation of the performance of different representation sets is still missing in nowadays' literature.

3. EVALUATION SETUP

In this section, we provide the details of the evaluation setup, ensuring that our results can be re-produced by other researchers. First, we discuss the technical parts of the NS-3 based simulations, followed by how we generated the user base and the 24-hour streaming scenario. Then we list recommended representations by YouTube, Netflix, and Apple, as well as representations determined by solving an optimization problem. We further describe how the metric used to measure user satisfaction has been derived.

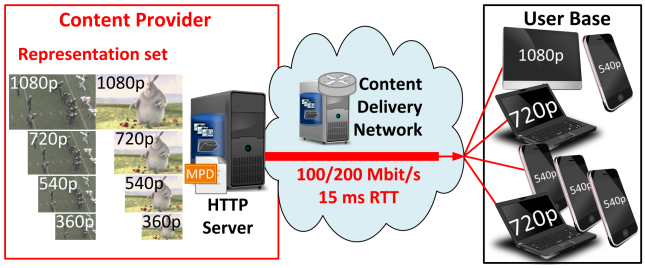

Figure 1: The considered DASH scenario, serving multiple encoded representations to users over a common bottleneck.

3.1 NS-3 and DASH

Since our goal was to create an adaptive streaming behavior as close to reality as possible, we decided to extend NS-3 with a client and server application capable of persistent HTTP connections. Based on libdash [8], we implemented a multimedia player with a video playback buffer, capable of simulating video playback. For the client-based adaptation, we implemented a simple rate-based adaptation logic, which selects the next segment's representation based on the last experienced goodput. We chose a rate-based adaptation logic, since the optimization model of [16] focuses on rate-adaptive control as well. A simplified network model as shown in Figure 1 was used, since our main concern was to investigate a single bottleneck link with many concurrent users.

To ensure efficient and realistic TCP behavior, we set up the NS-3 scenario to use TCP New Reno with a TCP segment size (MSS) of 1430 and an MTU of 1500 bytes. All routers were configured to use a RED (Random Early Detection) queue. The bottleneck link was set to a capacity of $C \in \{100, 200\}$ Mbit/s. All client connections were configured to their respective link capacities (see Section 3.2). The total RTT between each client and the server was 15 ms. In order to provide reproducibility of our results, we provide the source code of our client implementations as well as the used datasets at http://concert.itec.aau.at/NOSSDAV_2016/.

3.2 User Population

We implemented an adaptive streaming scenario consisting of heterogeneous users that dynamically leave and join over a timespan of 24 hours. To achieve (close to) realistic evaluations, we combined publicly available traces and datasets on users, user demands, and device statistics [10, 12, 16].

First, we generated a heterogeneous user base (consisting of 10,808 users) by randomly assigning clients to one out of four device categories as described in Table 1. We set the probability p, which denotes the probability of a category to be chosen, based on a survey of Netflix and Hulu users gathered by Nielsen in 2013 [10]. Each category is characterized by the display's spatial resolution and the network connection. The link capacity of each device is drawn uniformly from the interval $[c_{min}, c_{max}]$, which denote the minimum and maximum link capacities for each device, as in [16].

To model the population with varying user demands, we considered YouTube traces provided by [12]. The dataset provides the number of users watching a specific video at a certain point in time, with one measure every 5 minutes. This measure is available for different videos and for several time instants. From this dataset, we selected four video

Device Type (Connection)	Screen Res.	c_{min}	c_{max}	p
Smartphone (3G, WiFi)	360p, 540p	0.4	4	21.4%
Tablet (3G, WiFi)	540p, 720p	0.4	4	14.8%
Laptop (ADSL)	720p, 1080p	0.7	10	32.1%
HDTV (FTTH, Cable)	720p, 1080p	1.5	25	31.7%

Table 1: Devices with available screen resolutions and min/max link capacities (c_{min}/c_{max}) expressed in Mbit/s. p denotes the distribution of those devices [10].

Id	Video Name	Category	Length	#Users
1	Touchdown Pass	Sports	60 min	579
2	Snow Mnt	News	10 min	8,209
3	Big Buck Bunny	Cartoon	20 min	1,823
4	Aspen	Movie	90 min	197

Table 2: Test sequences from Xiph [19] for our evaluation and the number of users that requested it in our scenario. Videos have been chosen as representatives for their categories and are used to determine the user satisfaction metric.

categories: *Sports, News, Cartoon, Movie*. These video categories were popular enough such that they had a large and fluctuating number of users during the day.

In order to implement the users' requests over an entire day based on these data and keep the simulations computationally feasible, we needed to scale down the number of concurrent streams by a factor of 150. This resulted in a video streaming scenario with at least 50 and at most 300 concurrent users at any point in time. (Still, the simulations take several weeks.) The resulting cumulative number of concurrent users is depicted in Figure 2 for 2014-1-6.

The requests for each category depicted in Figure 2 are elastic over time and differ between the four categories. For instance, the *Sports* category is requested rarely for most of the day, and highly requested over a small portion of the day; the *News* category is popular the whole day, while *Movie* is more intensely requested between 15 and 20 hours. Finally, we considered the four video categories to consist of video sequences with a pre-defined duration, as listed in Table 2.

Based on the number of concurrent users at any point in time in the dataset, we start or stop, respectively, video streaming clients. In addition, clients stop streaming once the video has finished (according to the video's length). Shorter sequences, such as news, are requested more often than longer sequences, such as movie or sports. This is also expressed by the number of requests per day in Table 2.

3.3 Recommended Representation Sets

In this section, we investigate different recommended representation sets for hosting video content, as provided by YouTube, Netflix, and Apple. The representations of YouTube were experimentally derived, whereas Apple [3] and Netflix [9] explicitly provide the encoding parameters. A summary of those three sets is provided in Table 3.

YouTube. To the best of our knowledge, there are no publicly available recommendations for the representation sets for videos hosted on YouTube's servers. The only recommendation provided by YouTube itself consists of bit rates for streaming live videos with their platform [20]. To bridge this gap, we carried out an experimental analysis of representations used at YouTube by parsing metadata of 51,288 YouTube videos. We deem this as necessary for this evaluation because YouTube is one of the most important

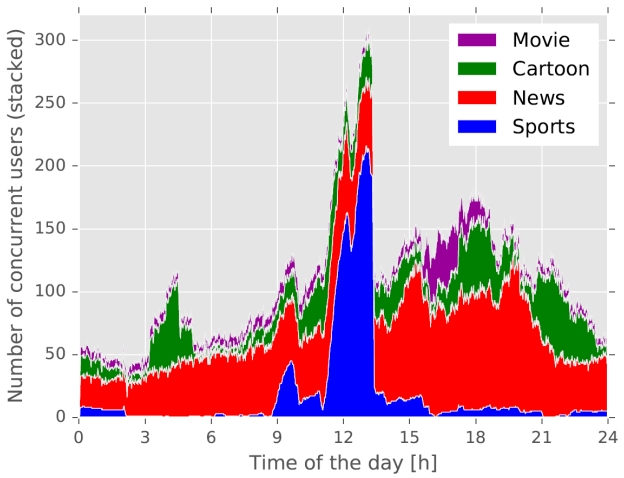

Figure 2: Number of concurrent users per category for 2014-1-6 in a *stacked* plot (users for categories are added on top of each other).

Name	Resolution	Bit Rates [kbit/s]
YouTube	1080p (1920x1080)	4,072
	720p (1280x720)	2,168
	540p (960x540)	1,109
	360p (640x360)	110 247 606
Netflix	1080p (1920x1080)	4,300 5,800
	720p (1280x720)	2,350 3,000
	540p (960x540)	1,050 1,750
	360p (640x360)	235 375 560 750
Apple	1080p (1920x1080)	11,000 24,000 39,000
	720p (1280x720)	2,500 4,500
	540p (960x540)	1,800
	360p (640x360)	110 200 400 600 1,200

Table 3: Summary of recommended representation sets from YouTube (experiment), Netflix [9], and Apple [3].

platforms for user-generated videos, with more than a billion videos played every day[1]. As YouTube provides MPD files for most of their videos, we decided to evaluate roughly $51k$ random videos from YouTube by parsing the MPD files. The resulting video bit rates of our evaluations conducted in October 2015 cover only MPEG-4/AVC and are provided in Table 4. Extrapolating the mean bit rate value, we built the representation set for YouTube provided in Table 3. For simplicity, the results for the original resolutions 144p, 240p, and 360p have been associated to 360p, and 480p to 540p.

Optimized Set. Toni et al. [16] proposed an optimization problem for the selection of the representation set that maximizes the average satisfaction of users. We applied this model to our adaptive streaming scenario and generated *optimized representations* as exemplified in Table 5, where C denotes the bottleneck capacity in Mbit/s and K the total number of representations for all videos.

Max–Min Model. In order to obtain an upper performance bound, we assume that every client will try to get as much goodput as possible, regardless of any adaptive streaming mechanism (adaptation logic, video playback buffer, ...). Let n be the number of clients. The throughput

[1]https://www.youtube.com/yt/press/en/statistics.html

Res./FPS	#Videos	Avg. BR [kbit/s]	(2.5, 50, 97.5%) percentiles [kbit/s]
2160p/24-30	29	21,580	[12,416 – 22,167 – 30,083]
1440p/24-30	102	8,008	[2,305 – 7,715 – 17,346]
1080p/48-60	222	5,391	[3,549 – 5,530 – 5,635]
1080p/24-30	**13,891**	4,072	[1,871 – 4,129 – 4,366]
720p/48-60	828	3,136	[2,032 – 3,316 – 3,401]
720p/24-30	**28,722**	2,168	[1,025 – 2,204 – 2,291]
720p/12-15	157	1,424	[379 – 1,251 – 2,308]
480p/24-30	**40,726**	1,109	[496 – 1,105 – 1,149]
480p/12-15	348	864	[221 – 692 – 1,164]
360p/24-30	**45,035**	606	[236 – 603 – 626]
360p/10-15	530	385	[111 – 344 – 632]
240p/24-30	**49,127**	247	[246 – 250 – 294]
144p/12-15	**51,288**	110	[108 – 112 – 129]

Table 4: Empirical YouTube bit rates (BR) [kbit/s] for video streaming. Values in brackets display the 2.5%, 50%, and 97.5% percentiles. Important representations are marked bold.

Video Id	Resol.	C100M-K24 kbit/s	C100M-K44 kbit/s
1	1080p	586	387 669
	720p	-	344 606
	540p	709	709
	360p	297 375	297 375 558
2	1080p	619 745 1190	526 619 745 1,042 1,380
	720p	297 534 676 1,093	297 370 534 676 777 1,093 1,361
	540p	173 407 529 747	329 529 620 747 1,242
	360p	315 568	220 315 568
3	1080p	-	819
	720p	761	533 761
	540p	553	320 553 785
	360p	245	245 595
4	1080p	-	-
	720p	-	1448
	540p	669 1,081	570 669 798 1,081
	360p	289 561	289 360 561

Table 5: Optimized representation sets [16], exemplified for $C = 100$ Mbit/s, $K = 24$ and $K = 44$ representations.

x_i of client i ($1 \leq i \leq n$) is affected by two bottlenecks: a) the local link capacity c_i, and b) the shared bottleneck towards the server. For the shared bottleneck we implemented a *Max-Min fairness* model (a fairness measure applicable for TCP's congestion control) and solved it with an iterative approach [11].

Based on the elastic user population (Section 3.2) we calculated the expected throughput for all users at any point in time. As detailed in Section 3.1, our TCP packets have 1430 bytes payload with 1500 bytes packet size, meaning that we have a goodput of 95.33%. The goodput ($0.9533 \cdot$ *throughput*) values serve as input for our satisfaction model (Section 3.4), leading to satisfaction values at any point in time for all clients. Essentially, this model assumes that there is an infinite number of representations available at the main server. This assumption will not hold in practice, but it allows to investigate the gap between the proposed representation sets and the theoretical bounds.

3.4 User Satisfaction

Our metric to assess the users' satisfaction [16] is an objective video quality based on the users' spatial screen resolution. We assess video quality by evaluating the SSIM [18] of multiple videos with the recommended representations provided in Table 3. While [16] used a *1-VQM* metric, in this paper we prefer SSIM since it has been shown that SSIM models the characteristics of the human eye better than other metrics [17]. However, the general idea of the resulting satisfaction curve stays the same.

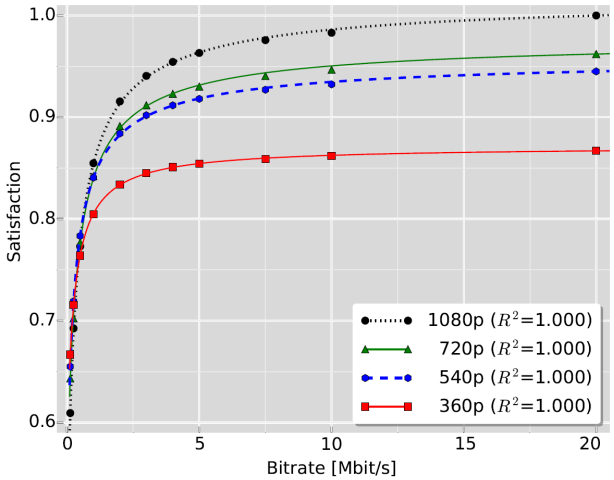

Figure 3: Satisfaction curves for Aspen (encoded at various spatial resolutions) measured for an 1080p screen.

The notion of satisfaction we use is adopted from [16, 6] and considers that a lower bit rate is required to satisfy a user with a 360p screen, than for a user with a 540p, 720p, or 1080p screen. In addition, a user with a 1080p screen will require the highest bit rate to be satisfied. To achieve this, we determined SSIM values of encoded videos by up- or down-scaling them, respectively, to a certain spatial screen resolution and comparing them to the source video with the same resolution. Due to space constraints we can only show one of the fitting curves in Figure 3.

First, we selected test sequences that are representative for their categories, as listed in Table 2. We encoded the sequences using *x264* (an MPEG-4/AVC encoder) with two-pass encoding, using the resolutions listed in Table 3, and various bit rates between 100 kbit/s and 20 Mbit/s. This allows us to interpolate/predict the SSIM value for arbitrary bit rates. The encoder was configured to produce DASH-compliant files with a segment length of two seconds containing 48 video frames. This segment length was chosen because it provides an acceptable trade-off [13] between encoding efficiency and the dynamic behavior introduced by the adaptation logic.

Second, to obtain SSIM values for all four spatial resolutions, we up- and down-scaled (using *ffmpeg* with bicubic scaling) all encoded videos to four spatial resolutions (360p, 540p, 720p, 1080p), and compared them with the respective source videos for each spatial resolution. The resulting SSIM value is then associated with the 4-dimensional tuple *video id, screen resolution, representation resolution, representation bit rate*.

We fitted the bit rate and the resulting satisfaction values using Equation 1, similar to [6], where x is the bit rate and $f_{v,s_u,s_r}(x)$ ($f : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{R}_{\geq 0} \rightarrow [0, 1]$) is the predicted satisfaction for video v, screen resolution s_u, and representation resolution s_r.

$$f_{v,s_u,s_r}(x) = c - \frac{a}{(x+d)^b} \tag{1}$$

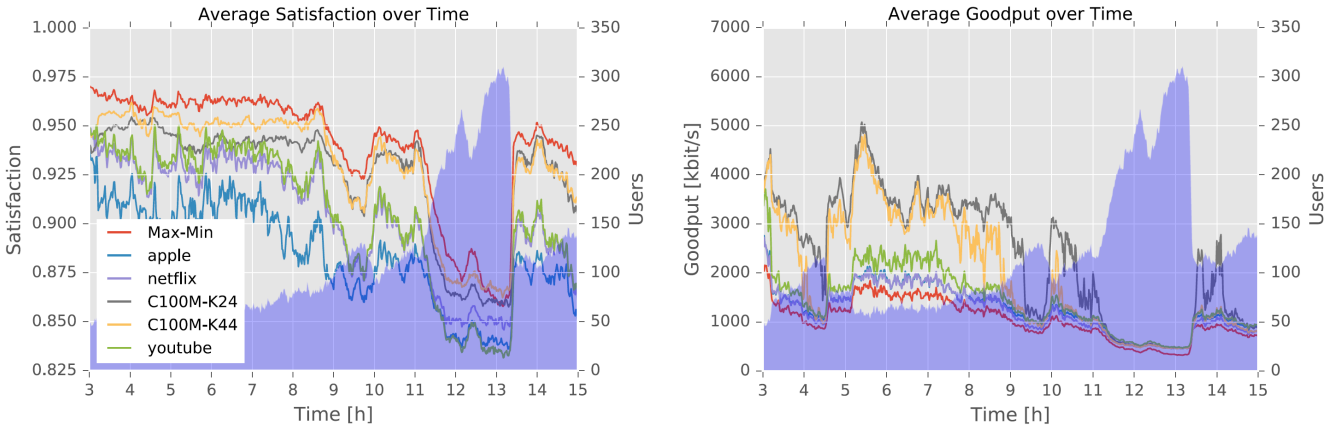

Figure 4: Average satisfaction and goodput values over time for a bottleneck bandwidth of 100 Mbit/s.

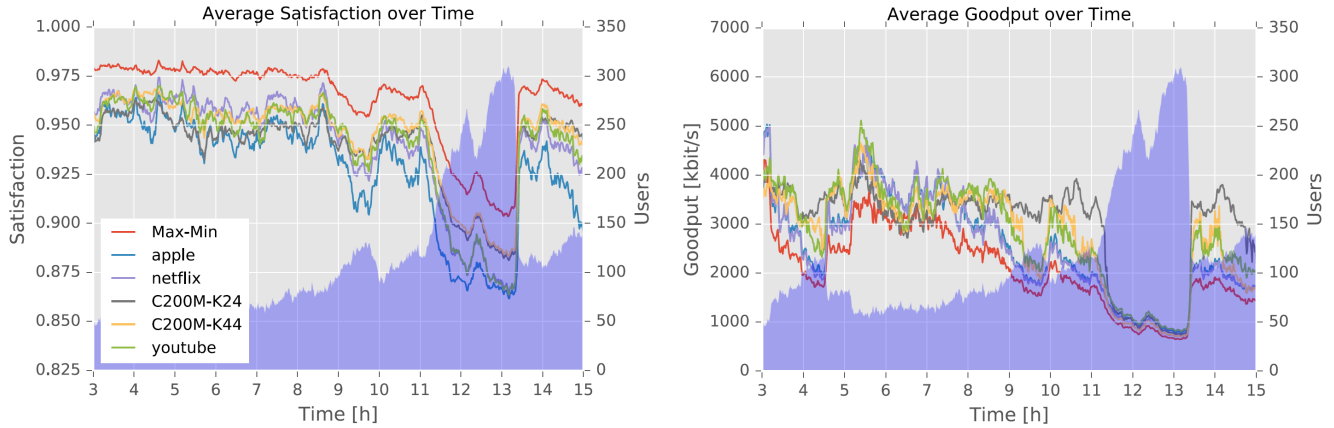

Figure 5: Average satisfaction and goodput values over time for a bottleneck bandwidth of 200 Mbit/s.

Equation 1 allows us to interpolate objective quality based on bit rates between 125 kbit/s and 20 Mbit/s. As shown by the example in Figure 3, the curve fits the data almost perfectly (R^2 values of other curves are also close to 1.0).

4. RESULTS

Figures 4 and 5 show the average user satisfaction and goodput over time for the discussed representation sets (cf. Section 3.3). Due to space constraints we show a 12 hour excerpt from 3 to 15 hours. The blue area in the background sketches the (total) number of concurrent users as illustrated in Figure 2, the lines show the average satisfaction and goodput on a per minute scale.

4.1 Satisfaction Analysis

The representation sets provided by Apple (cf. Table 3) do perform worst in both cases with respect to user satisfaction (bottleneck of 100 Mbit/s and 200 Mbit/s, cf. Figures 4 and 5). Apple recommends that there should be a 1200 kbit/s representation for a resolution of 360p, which leads to a low satisfaction for devices using a resolution of 540p or higher (see Figure 3). Representation sets recommended by Netflix and YouTube provide a more balanced set (with respect to bit rate) of representations for each resolution which explains the higher satisfaction, especially when there is a peak of users (cf. Figures 4 and 5, from 9 to 15 hours).

The *optimized* representation sets (according to [16]) indicated by C100M-K24, C100M-K44, C200M-K24 and C200M-K44, provide a higher satisfaction than the recommended representation sets by Netflix, YouTube, and Apple when the bottleneck link is congested (9 to 15 hours). In the case where the bottleneck link has a bandwidth of 200 Mbit/s, the representation sets do not differ significantly over time, except when there are peaks in the number of concurrent users.

The Max–Min model provides us with an upper bound for the satisfaction (assuming an infinite number of representations as explained in Section 3.3). It is evident from Figures 4 and 5 that a higher bottleneck bandwidth leads to a greater gap between the satisfaction obtained by the Max–Min model and the optimized and recommended representation sets. This provides room for improvement, especially during peak hours. For a bottleneck bandwidth of 100 Mbit/s (having heavy congestion during peak hours), the optimized representation sets are close to the results obtained by the Max–Min model.

4.2 Goodput Analysis

For every segment downloaded by each client we measure the time needed and the number of bits transferred, resulting in the goodput value. While a segment belongs to a certain representation with a pre-defined bit rate, the goodput can be lower or higher than this bit rate. Furthermore, the goodput can fluctuate heavily. Figures 4 and 5 show the average goodput (right side) of all clients active at a given time instant, thus the more clients are active, the lower the average goodput.

As already shown by [1], adaptive streaming clients fol-

low an on/off pattern when their video playback buffers are filled. This leads to fluctuations in the goodput which may have a negative impact on the client's adaptation logic. We noticed the same effect with the set of optimized representations (C100M-K24 and C100M-K44) in the 100 Mbit/s scenario, which achieve a higher goodput (e.g., between 5 and 8 hours in Figure 4) than the vendor representations.

This does not indicate that C100M-K24 and C100M-K44 perform better than the other representations. This behavior is caused by the bit rates of the representations. Due to C100M-K24 and C100M-K44 only having representations with a bit rate of up to 1400 kbit/s, there is plenty of capacity left (between 5 and 8 hours) for the active clients. Vendor recommendations provide representations with higher bit rates (up to at least 4 Mbit/s), which leads to a higher link utilization, and therefore a lower average goodput.

5. CONCLUSION

The results obtained show that choosing a different representation set clearly has an impact on the users' satisfaction. Moreover, we were able to partially confirm results from [16], where the authors already showed that recommended representations are not always the best choice.

Our results show that the recommended representation sets of Netflix, YouTube, Apple, and the optimized set of representations work well to a certain extent, but especially in peak hours where the bottleneck link(s) are fully utilized they do not provide enough flexibility for the heterogeneous (with respect to devices) clients. This motivates future work in optimizing representation sets. However, calculating optimized representation sets as proposed in [16] induces a certain amount of computational effort. Thus, future work shall focus on collecting statistics (e.g., user base over a certain time) and then optimizing representations based on data from the past towards providing an increased user satisfaction while maintaining a reasonable computational effort.

6. ACKNOWLEDGMENTS

This work was partly funded by the Austrian Science Fund (FWF) under the CHIST-ERA project CONCERT (A Context-Adaptive Content Ecosystem Under Uncertainty), project number *I1402*.

7. REFERENCES

[1] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis. What Happens when HTTP Adaptive Streaming Players Compete for Bandwidth? In *Proc. 22nd ACM Workshop on Network and Operating System Support for Digital Audio and Video*, 2012.

[2] R. Aparicio-Pardo, K. Pires, A. Blanc, and G. Simon. Transcoding Live Adaptive Video Streams at a Massive Scale in the Cloud. In *Proc. 6th ACM Multimedia Systems Conference*, 2015.

[3] Apple. Using HTTP Live Streaming. https://goo.gl/Kx5ZgS, 2014. Last accessed: Oct. 6, 2015.

[4] Conviva. Viewer Experience Report, 2015.

[5] A. El Essaili, D. Schroeder, E. Steinbach, D. Staehle, and M. Shehada. QoE-Based Traffic and Resource Management for Adaptive HTTP Video Delivery in LTE. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(6):988–1001, 2015.

[6] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race. Towards Network-wide QoE Fairness Using OpenFlow-assisted Adaptive Video Streaming. In *Proc. 2013 ACM SIGCOMM Workshop on Future Human-Centric Multimedia Networking*, 2013.

[7] T. R. Henderson, M. Lacage, and G. F. Riley. Network Simulations with the NS-3 Simulator. Demonstration. In *Proc. ACM SIGCOMM*, 2008.

[8] C. Müller, S. Lederer, J. Poecher, and C. Timmerer. libdash - An Open Source Software Library for the MPEG-DASH Standard. In *Proc. IEEE International Conference on Multimedia and Expo Workshops*, 2013.

[9] Netflix. Per-Title Encode Optimization. http://techblog.netflix.com/2015/12/per-title-encode-optimization.html, 2015. Last accessed: Feb. 9, 2016.

[10] Nielsen. "Binging" is the New Viewing for Over-the-top Streamers. http://www.nielsen.com/us/en/newswire/2013/binging-is-the-new-viewing-for-over-the-top-streamers.html, September 2013. Last accessed: Nov. 12, 2015.

[11] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Elsevier, 2004.

[12] K. Pires and G. Simon. YouTube Live and Twitch: A Tour of User-generated Live Streaming Systems. In *Proc. 6th ACM Multimedia Systems Conference*, 2015.

[13] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia. A Survey on Quality of Experience of HTTP Adaptive Streaming. *IEEE Communications Surveys & Tutorials*, 17(1):469–492, 2014.

[14] I. Sodagar. The MPEG-DASH Standard for Multimedia Streaming over the Internet. *IEEE Multimedia*, 18(4):62–67, 2011.

[15] T. C. Thang, H. Le, A. Pham, and Y. M. Ro. An Evaluation of Bitrate Adaptation Methods for HTTP Live Streaming. *IEEE Journal on Selected Areas in Communications*, 32(4):693–705, April 2014.

[16] L. Toni, R. Aparicio-Pardo, K. Pires, G. Simon, A. Blanc, and P. Frossard. Optimal Selection of Adaptive Streaming Representations. *ACM Transactions on Multimedia Computing Communications and Applications*, 11(2s):43:1–43:26, Feb. 2015.

[17] Z. Wang and A. Bovik. Mean Squared Error: Love it or Leave it? A New Look at Signal Fidelity Measures. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009.

[18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[19] Xiph.org. Xiph.org Video Test Media (derf's collection). https://media.xiph.org/video/derf/, 1994. Last accessed: Oct. 6, 2015.

[20] YouTube. Live Encoder Settings. https://support.google.com/youtube/answer/2853702?hl=en-GB, 2015. Last accessed: Oct. 5, 2015.

[21] W. Zhang, Y. Wen, Z. Chen, and A. Khisti. QoE-Driven Cache Management for HTTP Adaptive Bit Rate Streaming Over Wireless Networks. *IEEE Transactions on Multimedia*, 15(6):1431–1445, 2013.