MODELLING THE IMPACT OF CACHING AND POPULARITY ON CONCURRENT ADAPTIVE MULTIMEDIA STREAMS IN INFORMATION-CENTRIC NETWORKS

Christian Kreuzberger, Benjamin Rainer and Hermann Hellwagner

Institute of Information Technology (ITEC)
Alpen-Adria Universität (AAU) Klagenfurt, Austria
firstname.lastname@itec.aau.at

ABSTRACT

The Internet is nowadays mainly used for streaming of multimedia content, something it was not built for originally. To guarantee user satisfaction, one of the key concepts of the Internet as we know it is bandwidth sharing. While this concept is necessary to provide stability in the network, several issues can arise with adaptive multimedia streaming, e.g., efficiency and stability. Considering Information-Centric Networking (ICN) and its network-inherent caching, those issues tend to become worse. Many researchers have proposed to use traffic shaping on the server to enable fair bandwidth sharing and stabilize clients. However, existing research does not consider content popularity and in-network caching. The contribution of this paper is two-fold. First, we propose a cache-aware traffic shaping policy, in order to guarantee seamless playback of videos. Second, based on content popularity, we calculate an average video quality achieved by this traffic shaping policy for various cache sizes, to show the impact of popularity and caching for multimedia streaming in ICN.

Index Terms— Content Popularity, Adaptive Multimedia Streaming, Information-Centric Networking, Traffic Shaping

1. INTRODUCTION

In today's Internet, real-time entertainment platforms such as YouTube and Netflix cause over 60% of traffic [1], and according to Cisco [2], the relative and absolute amount of multimedia traffic will increase significantly over the next years. While many platforms already make use of Content Distribution Networks (CDNs), many users are displeased with the performance of video streaming, and in 2013 about 27% of users are experiencing video playback stalls [3]. Hoßfeld et al. [4] studied the effects of such stalls and found out that user satisfaction drops exponentially with two or more stalling events per clip.

Video playback stalls are usually caused by network congestion. In adaptive video streaming deployments, clients dynamically adapt the video quality based on their local knowledge of the network, e.g., the estimated throughput. If a client over-estimates the available bandwidth share and requests the video at a higher bitrate, it will eventually experience a buffer underflow. [5] showed that this over-estimation usually happens due to adaptive streaming clients competing for bandwidth over a bottlenecked/congested link.

The first thing that comes into mind for bandwidth sharing is TCP's congestion control. TCP is able to share the bandwidth fairly between multiple competing players. Using *adaptive video streaming*, clients can measure the throughput and select a representation with a lower or higher bitrate accordingly, therefore maximizing the Quality of Experience (QoE). However, the problem of competing clients usually occurs after clients fill up their local video buffer (e.g., 30 seconds). Once the buffer is filled, the client stops downloading until there is enough room (e.g., 2 seconds) to download the next part of the video. This *on-off behaviour* causes competing clients to misinterpret the measured throughput [5].

A promising approach to counter this problem is to minimize the idle time of clients by using traffic shaping on the server [6]. The effective throughput is throttled to a rate where clients will no longer over-estimate their bandwidth share and not switch to a higher quality, but also not deplete their buffer. Consider the example depicted in Figure 1. Applying traffic shaping with 1 Mbit/s per client will be beneficial for both clients.

However, when traffic shaping is applied on the server, we need to consider ICN's network-inherent caching [7]. Popular

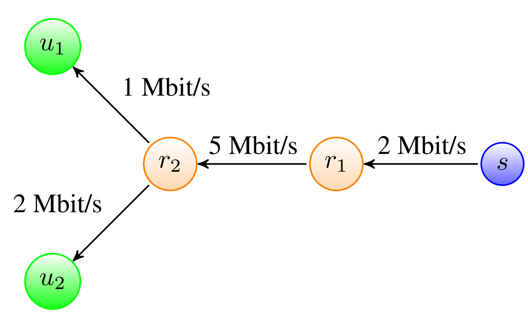


Fig. 1: A network with two clients (u_1, u_2), one server (s) and two routers (r_1, r_2). The link between r_1 and s is a bottleneck, and clients will compete for bandwidth on this link.

videos are cached on-path, and when multiple users request the same video (over the same path), only one request will arrive at the server. The remaining requests are served from the cache. The more users are served from the cache, the more bandwidth is available to serve the remaining users. If we consider the previous example with the addition of a cache on r_2, then u_1 could be served from the cache (without knowing it), while u_2 utilizes the full bandwidth towards s.

Within this paper, we are proposing a model for fair bandwidth sharing, based on content popularity and caching. By using video quality as our utility function, we provide an upper bound for the achievable quality over a bottlenecked link, with and without caching.

The remainder of this paper is structured as follows. Section 2 provides an overview of the related work on this topic. In Section 3 we describe adaptive video streaming and the problem of multiple users competing for bandwidth. We provide our model in Section 4, explaining the optimization goal and restrictions, and show initial results in Section 5. The paper is concluded in Section 6, indicating future work based on our model.

2. RELATED WORK

Plenty of related work exists about traffic shaping. Houdaille et al. [8] studied two concurrent adaptive video streams over a bottlenecked gateway, and showed that two concurrent streams can already negatively influence each other. By using simple traffic shaping on a residential broadband router, bitrate and quality fluctuations were reduced significantly.

Akhshabi et al. [5] identified the root cause of oscillations and unfairness: after HTTP-adaptive streaming clients fill up their buffer, they stay in a *steady state*. Periods of activity are followed by periods of inactivity (*on-off behaviour*), which can cause the client to misinterpret the measured throughput. [5] further introduced an instability metric based on the bandwidth and determined a fair share for each client. [6] implemented traffic shaping on the server-side to stabilize the client based on metrics defined in [5]. The authors were able to significantly improve the switching behaviour in the case of multiple competing clients. [9] investigated the problem of fairness even further and described a stateful bitrate selection scheme to ensure convergence of bitrates. [10] proposed a QoE-aware traffic shaping method by using Structural Similarity (SSIM) [11]. A linear program was used to calculate a shaping policy based on SSIM and profit per user.

In addition to traffic shaping, some related work exists about *proxy effects*. Mueller et al. [12] described negative effects that can occur when multiple DASH clients compete over a bottlenecked proxy server. They performed experiments which show that a proxy server can have a negative impact on the throughput estimation, potentially causing a high number of quality switches and even playback stalls. [13] implemented an intelligent shaping algorithm for in-network caches and reduced oscillations caused by in-network caching.

None of the mentioned works considered content popularity for their caching or shaping decisions. [10] used grades and profit of users to determine throughput per client, but did not consider content popularity. In terms of modelling the problem, Toni et al. [14] formulated a linear program, determining the number of representations and the bitrates for them. In addition they studied a satisfaction metric to determine the user's satisfaction based on bitrate and spatial resolution. However, they also did not consider content popularity, but assumed a uniform distribution.

Our model distinguishes itself from existing research since we introduce content popularity and in-network caching to the idea of traffic shaping. Our model also uses SSIM as the primary optimization goal, and by using traffic shaping we are trying to prevent quality fluctuations and stalls – the model is therefore QoE-aware.

3. CONCURRENT ADAPTIVE STREAMING

In this section we describe problems occurring when multiple users stream varying videos at the same time, especially in the case of Information-Centric Networking. When using a pull-based approach, such as DASH, the client adaptation logic is responsible for selecting a certain representation of the requested video. Each video has several representations, characterized by the spatial resolution, frame-rate and the bitrate. Therefore, clients can use local information, such as the estimated network throughput and current video buffer state, as well as the display resolution, to determine which representation should be requested.

The main goal of adaptive streaming is to provide excellent QoE by selecting an optimal representation. As network conditions might change over time, the client's adaptation logic has the possibility to change to a higher or lower representation at specified borders. For instance, in the beginning the client usually starts requesting a representation with a low bitrate to fill up the buffer to a certain threshold (e.g., 10 seconds). Once this threshold is met, the adaptation logic could decide to request a higher representation (e.g., based on the measured throughput), while keeping the buffer above or near this threshold. However, the client's buffer is often subject to limitations (e.g., 30 seconds), and once it is full, the client enters the so called *steady state*.

When a client is in *steady state*, the next couple of seconds of the video are only requested once there is more room in the buffer, which leads to an *on-off behaviour* [5]. While in the case of a single client this behaviour is expected and working as intended, [5] depicted several cases for concurrent multimedia streams where this on-off behaviour causes instabilities and oscillations at the client side.

When multiple users stream videos from the same server at the same time, they need to share the link's capacity some-how. In the case of TCP, congestion control protocols try to equally distribute the available bandwidth to all consumers. This behaviour is well studied and fairness is guaranteed. However, with the on-off behaviour described above, [5] shows that while bandwidth might be shared in a fair way, clients experience different throughput and choose different representations. To stabilize clients and guarantee fair sharing in terms of video bitrate, [6] proposed traffic shaping on the server. With fair traffic shaping, the authors were able to control the on-off behaviour and reduce idle times of clients, which led to more accurate throughput measurements and similar qualities among all clients.

Last but not least, the problem of concurrent adaptive streaming and fair bandwidth sharing is an important topic within the ICN community. In ICN, intermediate nodes have the ability to serve cached versions of the content. However, clients do not know whether they receive content from a cache or from the original source. Therefore, when a client requests content which is stored on an intermediate node, it experiences a higher throughput. While this is intended, it creates issues with adaptive streaming, since the adaptation logic might then decide to switch to a higher representation, which might not be cached. [12] showed that clients experience oscillations when consuming cached content. [13] proposed to apply traffic shaping at the cache to prevent clients from over-estimating the available bandwidth. In addition, [15] showed that popularity of content plays an important role when multimedia clients compete for bandwidth in ICN.

4. MODELLING CONCURRENT STREAMS

In this section we are going to detail the process of modelling concurrent adaptive multimedia streams. First, a simple approach is used for modelling multiple clients competing for bandwidth over a bottlenecked link, leading to a fair traffic shaping policy (based on video quality), but without considering caching. This approach will be used as the reference model we want to improve. We then introduce a novel model, based on an optimization problem which considers the server's network capacity, content popularity and in-network caching. By calculating the average video quality for both approaches, we determine the impact of caching and popularity.

4.1. Videos, Users and Traffic Shaping

The content catalogue, a set of videos available for streaming at the server, is denoted as $\mathcal{V}$. We denote the set of (equidistant) bitrates as $\mathcal{BR}$, and each video $v \in \mathcal{V}$ is available at all bitrates $b \in \mathcal{BR}$. Furthermore, let $b_{min} = \min \mathcal{BR}$ be the smallest bitrate representation available, therefore also the minimum required bandwidth to stream a video without experiencing playback stalls.

Let $\mathcal{U}$ be the set of users. For the purpose of this paper, each user $u \in \mathcal{U}$ represents a unique user requesting exactly one video $v \in \mathcal{V}$. Furthermore, we are assuming an idealized network model as displayed in Figure 2, where all users are requesting content from a single content server s, and the network acts as a large cache. The bottleneck capacity of the link between the last router and server s is denoted as $bw(s)$. We now define a traffic shaping policy π as follows:

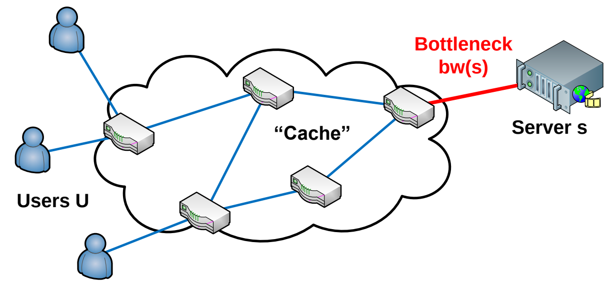


Fig. 2: Idealized network model - the network consists of nodes with a cache, we assume that the bottleneck is on the ingress/egress link towards the content server.

$$\forall u \in \mathcal{U} : \pi(u) \in \mathcal{BR} \cup \{0\}. \tag{1}$$

4.2. Naive Approach

The naive approach for calculating a fair traffic shaping policy π would be to use the available bandwidth $bw(s)$ and distribute it equally across $|\mathcal{U}|$ users (like it would happen in TCP), therefore $\pi(u) = b_{avg} = bw(s)/|\mathcal{U}|$ ($\forall u \in \mathcal{U}$). To analyze this approach for adaptive multimedia streaming, the following two cases need to be distinguished for now:
(i) $b_{avg} \leq b_{min}$ and (ii) $b_{avg} > b_{min}$.

In case (i), all clients will experience video playback stalls, until some (possibly only a few) users decide to stop streaming and b_{avg} increases above b_{min}. This behaviour leads to a poor QoE [4] for all users, and could have negative long-term effects on the streaming service (e.g., loss of subscribers). While this situation should not occur in theory, in practice it can (and does), and our model needs to handle this case by dropping some users from the system (by setting $\pi(u) = 0$). However, it remains to be determined which users should be dropped. Eventually, after dropping enough users, this case will lead to case (ii).

Case (ii), which should occur more often, allows streaming without playback stalls for all clients (contrary to case (i)). The solution is fair, as every user receives the same share of bandwidth. However, in general, $b_{avg} \notin \mathcal{BR}$. If b_{avg} is used, clients might experience problems due to the on-off behaviour described in Section 3. Instead, we could use the smallest following bitrate:

$$b_{feasible} := \max\{b \in \mathcal{BR} : b \leq b_{avg}\}. \tag{2}$$

This approach is feasible, but does not fully utilize the link's capacity. The questions that remain are, which users to abandon in case (i), and how to properly select bitrates in case (ii), to fully utilize the bottlenecked link's capacity $bw(s)$. One solution could be to randomly decide or work on a first-come first-serve basis, though we want to go one step further and use content popularity as input.

4.3. Content Popularity

To improve the initial approach and to answer the two open questions, we are going to look at content popularity. Many types of ranked data studied in science can be explained by using the power law [16]. In the case of content popularity, studies have shown that Video on Demand content can be characterized by a Zipf distribution [17]. By varying the parameters of the distribution, several cases, such as viral videos, 80-20 principle or almost uniformly distributed content can be covered. However, we are aware that it has been shown by [18] that the power law does not necessarily apply to all cases of content distribution. Due to space constraints, we will stick to the Zipf distribution. Our model will however not be tied to the Zipf distribution, and any other distribution can be plugged in.

Given a ranked content catalogue $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$, where v_i is more popular than v_{i+1}, the popularity of each video is characterized by the Zipf distribution's density,

$$\mathrm{Pop}(v_i) = \frac{1}{i^\alpha} \Bigg/ \sum_{k=1}^{|\mathcal{V}|} \frac{1}{k^\alpha}, \tag{3}$$

where α denotes the popularity parameter and $i \in \{1, 2, \ldots, |\mathcal{V}|\}$ denotes the content rank. For $\alpha > 0$ the distribution is long tailed, and by increasing α, the popularity of content with low ranks will slightly increase. By decreasing α, the distribution will become more balanced and for $\alpha \to 0$ it approaches a uniform distribution.

We now modify the initial approach, by simply sorting the user base $\mathcal{U}$ according to their requests (with users requesting the most popular videos at the beginning). For case (i), we start by satisfying the first $\gamma = \lfloor bw(s)/b_{min} \rfloor$ users. We justify this decision with the following argument: users consuming content with high popularity are more important as those consuming unpopular content. Popular content could be what keeps customers interested in the service. As expected, this simple solution leads to a shaping policy, satisfying as many users as possible, and consuming as much bandwidth as possible on the bottlenecked link.

For case (ii), where every user $u \in \mathcal{U}$ receives a certain share of bandwidth $b_{avg} > b_{min}$, we can apply a similar logic as above. Algorithm 1 determines a bandwidth allocation which maximizes the bitrate of all users. The algorithm works as follows. Starting with the smallest feasible bitrate $b_{feasible}$, satisfy as many users – denoted as γ_b – as possible and decrease the remaining capacity c. Consecutively, switch to the next higher bitrate b and try to satisfy as many users as possible again, but take into account users who were already satisfied with the previous bitrate b_{last}. Stop when the remaining capacity c is less than bitrate b.

```
π(u_i) ← 0, ∀i ∈ {1, 2, ..., |U|}:                 ▷ Initialize policy
b_last ← 0 ; γ_last ← 0                             ▷ Helper variables
c ← bw(s)                                           ▷ Remaining capacity
BR' ← {b ∈ BR : b ≥ b_feasible}
for all b ∈ Sorted(BR') do                  ▷ Iterate over sorted bitrates
    if c < b then
        break                           ▷ Not enough capacity remaining
    end if
    γ_b = min{ |U|, ⌊ c / (b − b_last) ⌋ }                   ▷ Satisfy
    c ← c + γ_b · (b_last − b)           ▷ Update remaining capacity
    π(u_i) ← b, ∀i ∈ {1, 2, ..., γ_b}                   ▷ Update policy
    b_last ← b ; γ_last ← γ_b
end for
```

Algorithm 1: Determine a bandwidth allocation which prefers users consuming popular content, and maximizes the quality for all users.

4.4. Video Quality

While the preceding algorithm is easy to implement and provides a fair solution (in terms of bitrate), it does not consider the video quality. Especially when videos of different categories (e.g., sports, animation, movie) are viewed, the video quality differs significantly. In general, the primary goal when looking at adaptive streaming is to maximize the QoE for all users. We have already ensured that as many users as possible can stream the video without playback stalls. Traffic shaping will also help stabilize the clients in terms of quality switching.

The remaining task is to maximize the video quality for each user, e.g., by using an objective metric such as SSIM. SSIM can be calculated by encoding a video at a certain target bitrate $b \in \mathcal{BR}$, and then comparing it to the original source video. Therefore, SSIM is usually a function of the bitrate and the video, denoted as $\mathrm{SSIM}(v, b)$.

To analyze the results of Algorithm 1, we need to define the average video quality based on SSIM as follows. Let $v_u \in \mathcal{V}$ be the video requested by user u, then the average quality is given by:

$$\Omega = \frac{\sum_{i=1}^{|\mathcal{U}|} \mathrm{SSIM}(v_{u_i}, \pi(u_i))}{|\mathcal{U}|}. \tag{4}$$

4.5. Introducing Caching

The preceding algorithm is nice, but has three issues: (i) it does not scale well with a growing user-base, (ii) it can not consider caching properly and, (iii) it does not consider video quality. To overcome these problems, we modify our problem such that we do not consider a shaping policy π per user, but per video. Therefore we define $\pi(v) \in \mathcal{BR} \cup \{0\}$ as the bitrate available for each user requesting video $v \in \mathcal{V}$. In addition, we introduce the idea of caching k items/videos ($0 \leq k \leq |\mathcal{V}|$) on an intermediate node ($k = 0$ means no caching).

$$\Omega = \max \sum_{v \in \mathcal{V}} \mathrm{Pop}(v) \cdot \mathrm{SSIM}(v, \pi(v)) \tag{5}$$

$$\text{s.t.} \quad \sum_{j=1}^{k} \pi(v_j) + |\mathcal{U}| \cdot \sum_{j=k+1}^{|\mathcal{V}|} \mathrm{Pop}(v_j) \cdot \pi(v_j) \leq bw(s) \tag{6}$$

$$\forall v \in \mathcal{V} : \pi(v) \in \mathcal{BR} \cup \{0\} \tag{7}$$

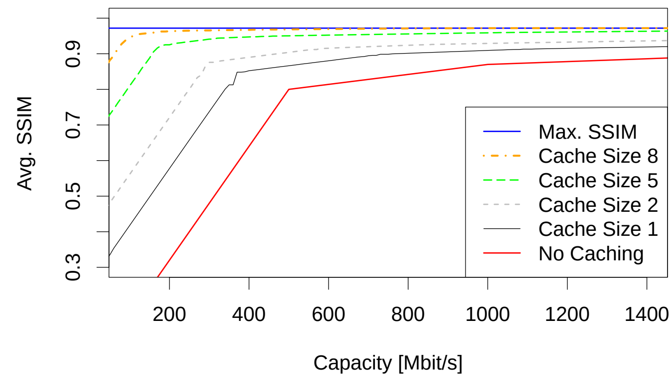
This model maximizes the average SSIM Ω of all users (Equation 5) based on popularity of videos, by assigning an appropriate bitrate $\pi(v) \in \mathcal{BR} \cup \{0\}$ for each video (Equation 7). In addition, Equation 6 guarantees that the available capacity $bw(s)$ is not exceeded. The first k videos are requested only once from the origin server, while the other videos are still requested as if there were no cache. For each video $v \in \mathcal{V}$ that is not cached, the required capacity is given by $|\mathcal{U}| \cdot \mathrm{Pop}(v) \cdot \pi(v)$.

The model can easily be converted into an assignment problem with $|\mathcal{V}| \cdot |\mathcal{BR}|$ variables. As $0 \leq \mathrm{SSIM}(v, b) \leq 1$ holds for any video and bitrate, the problem is bounded ($0 \leq \Omega \leq 1$). A trivial feasible solution exists (either $\pi(v) = 0$ or $\pi(v) = b_{feasible}$), therefore an optimal solution exists and can be found using the Simplex method.

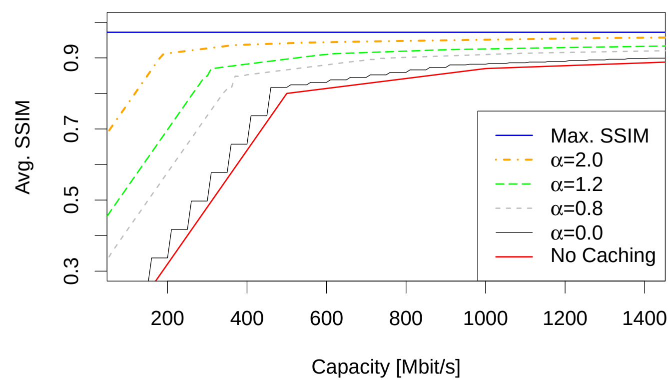
5. RESULTS

We evaluated our model for a fixed content catalogue with $|\mathcal{V}| = 10$, different values of α and different cache sizes $0 \leq k \leq 10$. Existing literature [19, 20] suggests different values for α. As we wanted to investigate only certain cases, we decided to follow [20] ($\alpha \in \{0.8, 1.2\}$) and also evaluate the cases of $\alpha = 0$ and $\alpha = 2$. The user base was set to 1000 users. We varied $bw(s)$ between 100 and 1500 Mbit/s in steps of 10 Mbit/s. We used fixed bitrates $\mathcal{BR} = \{500, 1000, 1500, \ldots, 9500, 10000\}$ (kbit/s) and SSIM values $\{0.75, 0.85, 0.88, \ldots, 0.988, 0.989\}$ for each video.

Figure 3 visualizes our results by plotting the capacity $bw(s)$ against the average video quality (SSIM). The thick red line shows the results from Algorithm 1, and therefore the theoretical maximum without caching, and therefore also a feasible solution of our model. The thick blue line on top shows the SSIM value of the highest-quality representation available, therefore the upper bound. The results show the potential impact of caching in ICN in terms of quality improvement and bandwidth savings. Figure 3a shows the impact of caching for $\alpha = 0.8$. Clearly, just by caching the most popular item, a significant improvement of video quality can be achieved. Figure 3b shows the impact of different α values for the Zipf distribution, therefore the impact of popularity, under a constant cache size $k = 1$. Even for $\alpha = 0.0$ (uniform distribution), the model suggests that there is a benefit of caching one item.



(a) Impact of caching for $\alpha = 0.8$ and various cache sizes. The lowest red line shows the case of caching disabled, using Algorithm 1.



(b) Impact of popularity for cache size $k = 1$ and various popularities. The lowest red line shows the case of caching disabled, using Algorithm 1.

Fig. 3: Impact of popularity and cache size

6. CONCLUSION

As the amount of traffic related to multimedia streaming is growing rapidly, traffic shaping is used as a solution. In this paper we proposed a novel model to determine shaping bandwidths per video, under varying popularities and cache sizes. The results show that even in the case of uniformly distributed content, a proper selection of shaping bitrates is able to improve the average video quality of all users. Future work will consist of evaluations of this model in test-beds and simulations, as well as comparisons with different approaches.

7. 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*.

8. REFERENCES

[1] Sandvine, "Global Internet Phenomena Report," Tech. Rep. 1H, 2014.

[2] Cisco, "Visual networking index," Tech. Rep., 2014.

[3] Conviva, "Viewer experience report," Tech. Rep., 2014.

[4] T. Hoßfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz, "Quantification of YouTube QoE via crowdsourcing," in *IEEE International Symposium on Multimedia (ISM), 2011*, Dec 2011, pp. 494–499.

[5] Saamer Akhshabi, Lakshmi Anantakrishnan, Ali C. Begen, and Constantine Dovrolis, "What happens when HTTP adaptive streaming players compete for bandwidth?," in *Proceedings of the 22nd ACM Workshop on Network and Operating System Support for Digital Audio and Video*, New York, NY, USA, 2012, NOSSDAV '12, pp. 9–14, ACM.

[6] Saamer Akhshabi, Lakshmi Anantakrishnan, Constantine Dovrolis, and Ali C. Begen, "Server-based traffic shaping for stabilizing oscillating adaptive streaming players," in *Proceeding of the 23rd ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, New York, NY, USA, 2013, NOSSDAV '13.

[7] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, July 2012.

[8] Rémi Houdaille and Stéphane Gouache, "Shaping HTTP adaptive streams for a better user experience," in *Proceedings of the 3rd Multimedia Systems Conference*, New York, NY, USA, 2012, MMSys '12, pp. 1–9, ACM.

[9] Junchen Jiang, Vyas Sekar, and Hui Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, New York, NY, USA, 2012, CoNEXT '12, ACM.

[10] Xinying Liu Xinying Liu and Aidong Men Aidong Men, "QoE-aware traffic shaping for HTTP adaptive streaming," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 9, no. 2, 2014.

[11] Zhou Wang and AC. Bovik, "Mean squared error: Love it or leave it? A new look at signal fidelity measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, Jan 2009.

[12] C. Mueller, S. Lederer, and C. Timmerer, "A proxy effect analyis and fair adaptation algorithm for multiple competing dynamic adaptive streaming over HTTP clients," in *Proceedings of IEEE Visual Communications and Image Processing (VCIP), 2012*, Nov 2012.

[13] Danny H. Lee, Constantine Dovrolis, and Ali C. Begen, "Caching in HTTP adaptive streaming: Friend or foe?," in *Proceeding of the 24th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, New York, NY, USA, 2014, NOSSDAV '14, ACM.

[14] L. Toni, R. Aparicio-Pardo, G. Simon, A. Blanc, and P. Frossard, "Optimal set of video representations in adaptive streaming," in *Proceedings of the 5th ACM Multimedia Systems Conference*, New York, NY, USA, 2014, MMSys '14, ACM.

[15] D. Posch, C. Kreuzberger, B. Rainer, and H. Hellwagner, "Using in-network adaptation to tackle inefficiencies caused by dash in information-centric networks," in *Proceedings of the 10th International Conference on Emerging Networking Experiments and Technologies, VideoNext Workshop*, Dec 2014.

[16] J.R. Pierce, *An Introduction to Information Theory: Symbols, Signals & Noise*, Dover Books on Mathematics Series. 1980.

[17] Xu Cheng, C. Dale, and Jiangchuan Liu, "Statistics and social network of YouTube videos," in *16th International Workshop on Quality of Service, 2008.*, June 2008, pp. 229–238.

[18] György Dán and Niklas Carlsson, "Power-law revisited: Large scale measurement study of P2P content popularity," in *Proceedings of the 9th International Conference on Peer-to-Peer Systems*, Berkeley, CA, USA, 2010.

[19] L. Breslau, Pei Cao, Li Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: evidence and implications," in *Proceedings of the 18th IEEE Computer and Communications Societies Conference, INFOCOM '99*, Mar 1999, vol. 1, pp. 126–134 vol.1.

[20] Christine Fricker, Philippe Robert, James Roberts, and Nada Sbihi, "Impact of traffic mix on caching performance in a content-centric network," in *In Proceedings of Computer Communications Workshops (IEEE INFOCOM Workshop), 2012*. IEEE, 2012.