QoE-Centric Network-Assisted Delivery of Adaptive Video Streaming Services

Stefano Petrangeli
Ghent University – imec, Adobe Research
petrange@adobe.com

Tim Wauters
Ghent University – imec
tim.wauters@ugent.be

Filip De Turck
Ghent University – imec
filip.deturck@ugent.be

***Abstract*—Video streaming traffic has gone through an impressive growth over the last ten years. This growth can be rooted back to two different causes, namely widespread content accessibility and improved delivery techniques, such as HTTP Adaptive Streaming (HAS) and Web Real-Time Communication (WebRTC). HAS techniques are widely used in Video-on-Demand and live streaming scenarios, while WebRTC is mostly used in interactive streaming applications. Both HAS and WebRTC are unfortunately still affected by several inefficiencies. In HAS, the video client is equipped with a heuristic to dynamically adapt the video quality to the available network resources. This client-based adaptation cannot always guarantee the best experience to the end users, the so-called QoE. On the other hand, WebRTC is by design peer-to-peer, which makes this technique not scalable when the number of users in the system is high. To solve these problems, this PhD thesis proposes an advanced architecture where additional intelligent components are placed in the network to support the delivery of the video. Moreover, the designed network components aim at optimizing specific QoE parameters that directly impact the users' viewing experience, rather than QoS parameters, which represent low-level network performance. This architecture has been applied, for example, to reduce playout interruptions in the video playback of HAS clients up to 45% or to increase the scalability of WebRTC systems without affecting the delivered video quality.**

***Index Terms*—Quality of Experience, Software-Defined Networking, HTTP Adaptive Streaming, WebRTC**

I. Introduction

Video streaming applications currently dominate Internet traffic. According to Cisco reports, the amount of traffic generated by video applications has increased from 2900 Petabytes per month in 2009 to 58000 Petabytes per month in 2017, an impressive growth of almost twenty times in less than ten years. This streaming revolution has also occurred thanks to relevant improvements in video delivery techniques. Indeed, the best-effort Internet was not originally designed to support such bandwidth-intensive traffic. In Video-on-Demand (VoD) and live streaming events, traditional streaming techniques based on RTP/RTCP have been replaced by solutions based on the HTTP Adaptive Streaming (HAS) principle. At the cost of an increased end-to-end latency compared to RTP, HAS presents an increased scalability, better NAT and firewall traversal and the possibility to reuse the existing HTTP infrastructure, in terms of caches and web servers. Moreover, this technique allows to easily accommodate the streamed video quality to the varying bandwidth conditions of the video clients. Nevertheless, RTP-based techniques did not disappear, but they are actually widely used in interactive streaming scenarios, as in the Web Real-Time Communication (WebRTC) initiative. Indeed, HTTP-based techniques do not guarantee the low-latency and interactivity requirements of these applications.

Independently from the streaming protocol, an effective video delivery depends on the quality as perceived by the end user, the so-called Quality of Experience (QoE), rather than on classical Quality of Service (QoS) network-level parameters. As the relationship between QoS parameters and QoE is far from linear, a classical QoS-centric delivery is not able to fully optimize the quality as it is perceived by the end users. For this reason, this thesis [31] focuses on optimizing those general QoE factors, as video quality and freezes for instance, that are clearly quantifiable and have a direct impact on user experience in video streaming. Moreover, the work presented in this PhD thesis makes use of network-assisted elements, with the aim of supporting the delivery of adaptive video streams and providing a better QoE to the end users. This shift is needed as several QoE factors in video streaming cannot be optimized using a purely client-server architecture, but would benefit from the assistance of intelligent network components. The network elements presented in this thesis are therefore developed to assist and support the delivery of the video, and help the clients obtaining good streaming performance. This PhD research resulted in several journal papers ([4]–[6], [9], [14], [16], [20], [28], [29]), conference papers ([1]–[3], [7], [8], [10]–[13], [15], [17]–[19], [21]–[27]) and one patent ([30]) and it has been conducted in partnership with several leading industry partners, as Adobe, Nokia and Barco, and within the FLAMINGO European ICT-FP7 Network of Excellence.

Particularly, the following *challenges*, and related *technical contributions*, are presented in this thesis:

- *Challenge #1: provide a video streaming service that is fair to the end user from the QoE point of view.* In a typical HAS setting, multiple clients can access the same content from the same server. Often, clients have to share a single medium and issues concerning fairness among them appear, meaning that the presence of a client has a negative impact on the performance of others [32]. Even though fairness is a system-wide characteristic rather than a user perceived QoE factor, it is often a desired property of the system, because it can finally improve user experience and should therefore be maximized. Therefore, in this thesis, an in-network system of network components is deployed to support the HAS clients and allow them to obtain both a high QoE and a fair behavior. The in-network system estimates the fair bandwidth share of the competing clients. This information is used by the clients to request the video quality maximizing their own QoE, while being fair to the other users.

- *Challenge #2: avoid playout interruptions, also called freezes, during the video playback.* Even though HAS solutions have been developed with bandwidth adaptation capabilities in mind, they can still fail in avoiding video freezes. As an example, the Conviva report reveals that almost 25% of the analyzed HAS sessions exhibit at least one video freeze [33]. This problem is mainly due to the unmanaged nature of current HAS technologies, as the clients are not aware of the real network conditions nor are they assisted in improving the delivered QoE. To reduce this problem, this thesis proposes a network framework, based on the Software-Defined Networking (SDN) principle, that can temporarily prioritize the delivery of HAS segments possibly leading to a video freeze. This decision is based on both measurements collected from the network nodes and an estimation of the client status. The proposed framework does not require any communication between clients and network, and can optimize the delivery of any client implementation.
- *Challenge #3: increase the adaptability of the network elements to support unseen video streaming scenarios, reducing human hand-tuning.* Even though network elements can help the video delivery, they are usually designed and developed with certain network and streaming characteristics in mind. This aspect entails that these systems might be unable to adapt and provide satisfactory performance under highly different streaming scenarios. It is therefore required to make the network elements adaptive and self-learning, so that they can learn the best action to take based on the network and streaming conditions. The benefits of this approach are showcased in the prioritization framework presented above. Particularly, a machine learning-based prioritization algorithm is designed to perform the prioritization task. Prioritization is based on the Random Undersampling Boosting (RUSBoost) algorithm and fuzzy logic [40], which can autonomously learn when an HAS segment should be prioritized.
- *Challenge #4: reduce costs and increase scalability and performance for interactive streaming applications.* As mentioned at the beginning of this introduction, HAS techniques are very effective for VoD and live streaming scenarios, but they do not guarantee the low latency required for remote collaborations. In this case, protocols based on RTP, as WebRTC, are used instead. The WebRTC framework has been developed with a peer-to-peer architecture in mind, where a small group of clients can directly communicate with each other. This approach can suffer from scalability issues when multiple participants are present at the same time, since the WebRTC senders would need to encode a separate stream for each of the WebRTC receivers. This aspect entails that each receiver is associated with an independent and dedicated encoder at sender-side, which is expensive and does not scale well. To improve the scalability of this peer-to-peer architecture, a framework is proposed in the thesis where each peer only needs to encode a limited number of streams, much smaller than the number of participants. These streams are sent to a centralized node, called conference controller, which dynamically forwards them to the receiving peers, based on their bandwidth conditions. This way, each encoder at the sender transmits to a multitude of receivers at the same time, improving scalability. Moreover, the controller periodically recomputes the set of encoding bitrates at the sending peers, in order to follow the long-term bandwidth variations of the receivers and increase the received video quality.

The remainder of this paper is structured as follows. Section 2 presents related works on QoE-centric solutions for adaptive streaming, in the form of an extensive survey paper [20]. Section 3 tackles the problem of fairness in adaptive streaming (*Challenge #1*), while Section 4 presents a SDN framework to avoid video freezes in HAS (*Challenge #2*). Section 5 redesign the network logic employed in Section 4 using a Machine Learning (ML) solution that can autonomously learn how to prevent freezes (*Challenged #3*). Section 6 addresses the problem of scalability and quality for scalable remote video collaboration applications based on WebRTC (*Challenge #4*). Finally, Section 7 concludes the paper and highlights some directions for future research that can leverage the PhD work presented in this paper.

II. Survey on QoE-Centric Management Solutions

Several works have been proposed in literature to optimize the video delivery for adaptive streaming services that go beyond purely client-based approaches that, despite effective, can still fail providing the end users with a high QoE. An elaborated survey of the current research efforts in HAS is first presented in the thesis to clearly highlight the different approaches that can be adopted in this domain. Particularly, the different approaches can be clustered based on where the optimization takes place: (i) server- and network-assisted solutions, (ii) application level optimizations and (iii) transport level modifications. Server- and network-assisted solutions place additional intelligence in the network to support the delivery of the video. Traffic rerouting, bandwidth shaping techniques and caching represent the most typical examples in this space. In approaches exploiting traffic rerouting, the performance of the paths connecting server and clients is continuously monitored. When certain conditions are detected (e.g., increased packet loss or congestion) a path recalculation occurs to reroute the video traffic and guarantee good

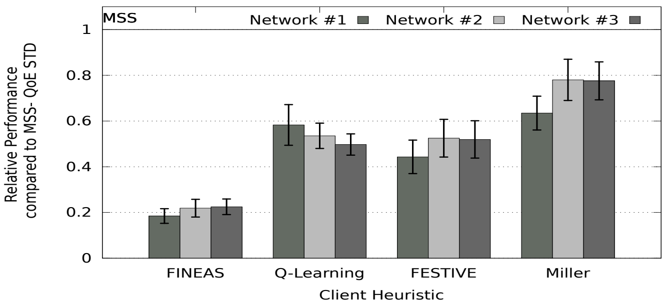

Fig. 1. The proposed network framework of coordination proxies can help reducing unfariness up to 80% compared to benchmarking heuristics. Fairness is measured as the standard deviation of clients' QoE values.

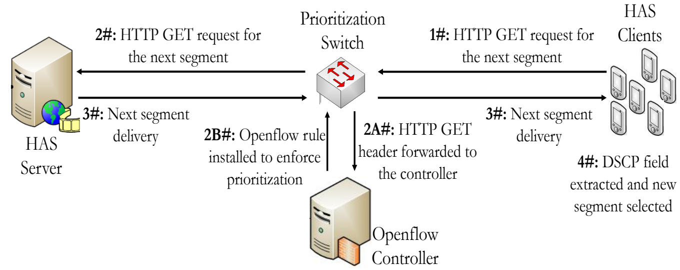

Fig. 2. The OpenFlow controller intercepts clients' requests and decides whether the requested segment should be prioritized or not.

streaming performance [34], [35]. Bandwidth shaping/bitrate guidance techniques are usually deployed on a bottleneck link located in the access or edge network, and are designed to select and enforce a specific quality for the streaming clients [4], [36]. This selection aims to optimize the revenue of the network operator, by maximizing the QoE of particular users, or provide fairness among competing clients. Application level solutions can optimize the behavior of any adaptation heuristic by exploiting, for instance, the new features of the HTTP/2 protocol or prefetching techniques. Particularly, HTTP/2 push-based solutions can consistently reduce latency in live streaming, increase bandwidth utilization and video quality, at the cost of reduced bandwidth adaptability that can result in more freezes [6], [37]. Transport level approaches modify the congestion control algorithms or retransmission policy of TCP, to better support video traffic. Enhanced TCP solutions for adaptive streaming generally follow two main trends. First, additional application-level information can be shared with the transport layer to improve the scheduling of TCP packets, in order to primarily avoid freezes [39]. Second, Multi-path TCP can be employed to improve the aggregate throughput of the streaming client [38].

III. QoE-Driven Heuristic for Fair HAS

This section tackles one of the problems still affecting HAS systems, namely fairness. Concretely, this means that different HAS clients negatively influence each other as they compete for shared network resources. To solve this problem, we present a fair HAS client able to achieve smooth video playback, while coordinating with other clients in order to improve the fairness of the entire system. The proposed fair HAS client performs the quality level selection based on three inputs: the local perceived bandwidth, the video player buffer status and the so-called *fairness signal*. The fairness signal is an additional measure introduced to achieve fairness, obtained when the client downloads a segment. The fairness signal is computed by a system of network nodes, called *coordination proxies*, and represents an estimate of the fair bandwidth share of all the clients streaming video. In order to maintain scalability, the computation of the fairness signal is performed periodically and in a hierarchical way by the coordination proxies. A generic coordination proxy *P* receives an estimate of the fairness signal from its parent node and computes a new estimate of the fairness signal for each of its child proxies. This estimate is computed by monitoring the available bandwidth for HAS traffic on the links connecting proxy *P* to its child nodes. In order to limit overhead, the calculated fairness signal can be added as an HTTP header field and returned to the clients when delivering the next segment to play. Particularly, the clients translate the fairness signal into a *reference* quality level, representing the theoretical quality level to request in order to obtain perfect fairness among the clients. This reference gives an indication on the best quality level to achieve fairness, rather than determining the actual quality to be requested. The reason for this behavior is twofold. First, directly requesting the reference quality level would be optimal from the fairness point of view but not from the QoE point of view, because of the frequent quality switches that would occur. Second, directly requesting the reference quality level would alter the classical HAS principle, as the decision on the quality level to download would no longer be carried out by the clients. The fair in-network system has been evaluated through simulations, under highly variable bandwidth conditions and in several multi-client scenarios. The proposed approach can improve fairness up to 80% compared to state-of-the-art HAS heuristics in a scenario with three networks, each containing 30 clients streaming video at the same time, as reported in Figure 1.

IV. SDN-Based Prioritization to Avoid Video Freezes in HAS

While the previous section focuses on a system-wide property like fairness, the goal of this section is to reduce the occurrence of video freezes in HAS, which are the main factor influencing users' QoE. To this aim, a network framework is presented, based on the SDN principle and OpenFlow. The main component of this framework is an OpenFlow controller deciding which HAS segments should be prioritized in order to avoid interruptions in the video playout of the clients. Based on the current network conditions and a network-based prediction of the HAS clients' status, the controller can decide to prioritize the delivery of particular segments in order to avoid video freezes at the clients. Prioritization is enforced in the network by using an OpenFlow-enabled switch, the so-called *prioritization switch*, which is equipped with a best-effort and a prioritized queue. Based on controller decisions, the prioritization switch enqueues the clients' segments in one of these queues. An illustrative sequence diagram of the proposed framework is shown in Figure 2. The OpenFlow controller intervenes each time a client requests a new segment from the HAS server. When an HTTP GET is received, the controller decides whether the delivery of the analyzed segment should be prioritized or not. This decision is based on network measurements collected from the prioritization switch and client related measurements, estimated at the controller side. Network measurements are obtained using the OpenFlow protocol, which provides well-defined APIs to collect data from the OpenFlow switches. More specifically, the controller periodically polls the prioritization switch to compute the available bandwidth for the HAS clients in the best-effort queue (not shown in Figure 2). Client related measurements include the segment bitrate and the video player buffer filling level. The former is obtained by analyzing the GET message intercepted by the prioritization switch. The latter is estimated through buffer reconstruction methods. This means that no explicit clients-to-controller communication is required in our framework. A prioritization logic running at the controller decides which segment to prioritize, by computing an estimate of the segment download time in the best-effort queue. If a best-effort delivery does not guarantee a timely download of the segment, i.e., if the download time is larger than the estimated client buffer filling level, the segment is prioritized. Next, the controller installs a new OpenFlow rule on the prioritization switch to guarantee a proper delivery of the analyzed segment, i.e., best-effort or prioritized delivery. This way, the controller only supports the delivery of particular video segments, rather than determining the actual quality to be requested by the clients. This approach is robust toward controller and switch failures, as the clients can still operate even if prioritization cannot be enforced into the network. Moreover, the proposed OpenFlow framework can be deployed to optimize the delivery of any existing heuristics. The OpenFlow framework has been evaluated through emulation in several multi-client scenarios in presence of realistic Internet cross-traffic. Results show that the proposed approach can reduce freeze time due to network congestion by more than 50% when compared to a baseline client-based heuristic not supported by prioritization, without impacting the performance of cross-traffic applications.

V. A Machine Learning-Based Framework for Preventing Video Freezes in HAS

The previous section presented a SDN framework that can successfully avoid video freezes at the clients. Despite its good performance, this approach requires the network controller to know the client configuration, in terms of buffer behavior and streamed video. Moreover, the proposed algorithms are hardcoded and require a certain level of hand-tuning to reach the best performance. For this reason, this section presents a complete redesign of the prioritization logic, based on ML algorithms. This way, the controller can autonomously learn when an HAS segment needs to be prioritized. The ML-based prioritization logic is composed of two modules, the *freeze predictor* and the *congestion detection* modules. When a segment is requested by a client, the freeze predictor has to decide whether the download of the segment is going to be affected by a freeze or not. This problem can be modelled as a classification problem with two classes: freeze and non-freeze. The classification is based on measurements obtained by the controller without any a priori assumption on the video and clients' configuration, in terms of video bitrates, segment duration, maximum buffer size and start-up buffering time. This aspect clearly complicates the classification task but allows a more general applicability of the proposed approach in real environments. The prediction is performed by analyzing the inter-arrival time of GET requests issued by the client, the requested quality level and the available bandwidth for HAS traffic in the best-effort queue. The predictor is trained off-line using the RUSBoost classification algorithm, which is particularly suited for classification problems affected by class imbalance, as in the freeze prediction case [40]. In fact, only a small percentage of a video is usually affected by freezes, since in normal conditions the HAS principle is able to achieve a continuous playout. The congestion detection module is used instead to avoid congesting the prioritization queue and to allow a fair share among the different clients. This detection is performed by a fuzzy engine based on inputs as the bandwidth in the prioritization queue, the requested quality level and the number of consecutive prioritizations experienced by a client. The freeze prediction and the congestion detection modules return the probability for the current segment to freeze and the probability for the current segment to congest the prioritization queue, respectively. By combining these two factors, the controller obtains the actual probability of prioritizing the segment. A client is prioritized with a higher probability when the risk of a freeze is high, as indicated by the freeze predictor, and prioritizing the segment does not congest the prioritized queue nor is unfair to the other clients, as indicated by the congestion detection module. Extensive experimentation under diverse network conditions, client configurations and video sequences show that the proposed approach can reduce video freezes and freeze time with about 65% and 45% respectively, when compared to client-based benchmarking algorithms (Figure 3).

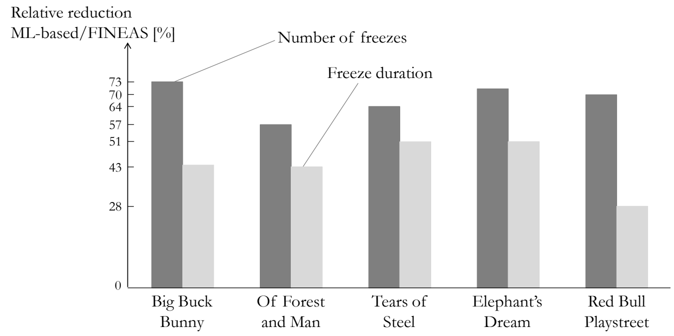

Fig. 3. The ML-based prioritization framework can reduce the number of freezes and average freeze duration by 65% and 45% across all tested videos, when compared to the purely client-based FINEAS algorithm.

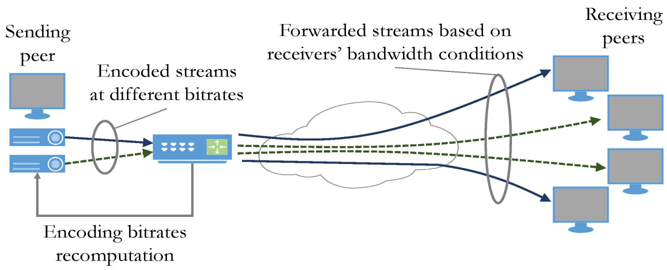

Fig. 4. The conference controller behaves as the terminal endpoint for both the sender and the receivers, and performs the dynamic stream forwarding and dynamic bitrate recomputation tasks.

VI. A Scalable WebRTC Framework for Remote Video Collaboration Applications

The previous sections have focused on the optimization of HAS solutions, which are the dominant technology in live streaming and VoD scenarios. The goal of this section is to provide a different perspective, by considering those streaming cases that require low-latency and interactivity, as remote conferencing, telehealth and remote teaching applications. In this case, real-time communication solutions based on RTP are generally used instead of HTTP-based solutions. Particularly, the WebRTC protocol is of extreme interest in this case, as it allows any two or more remote peers equipped with a browser to communicate among each other, without the need of external plugins. The WebRTC framework has been developed with a peer-to-peer architecture in mind, meaning the peers in communication, or *senders*, would need to encode a separate stream for each receiving peer, the *receivers*. This aspect entails that each receiver is associated with an independent and dedicated encoder at sender-side, which is expensive and does not scale well when the number of participants is large (e.g., in a remote classroom scenario). To solve these problems, this chapter proposes a WebRTC-compliant framework where only a limited number of encoders are used by the sending peers, much smaller than the number of receiving peers. The high-level architecture of the proposed framework is presented in Figure 4 (for the sake of simplicity, the figure only depicts one sender). At sender-side, the remote event is captured and encoded in different streams at different bitrates, which are sent to the conference controller. The controller is then responsible for the forwarding of the highest sustainable stream to each receiver in the session, based on their estimated bandwidth. This approach allows to relieve the peer-to-peer architecture of classical WebRTC and improve the scalability of the system. Indeed, each encoder at sender-side is now associated with multiple receivers, with the number of receivers usually much larger than the number of encoders. Moreover, the video bitrate of the encoders at sender-side is not static, but is periodically recomputed by the controller to better follow the bandwidth conditions of the receivers. In the proposed framework, each encoder at sender-side transmits to multiple receivers at the same time. In order to maximize the video rate received by the receivers, the encoded rate should be as close as possible to the actual bandwidth of the receivers. Static, fixed encoding bitrates are suboptimal, as the receivers' conditions can change over time, especially in wireless environments, where the available bandwidth can highly fluctuate. By allowing the encoding bitrates to dynamically vary, it is possible to follow the long-term bandwidth variations of the receivers and, therefore, maximize the delivered video quality. We formulate the dynamic bitrate recomputation problem as an ILP formulation, which is executed every T_{opt} seconds by the conference controller. Moreover, we also propose an alternative formulation based on the K-means clustering algorithm to solve the bitrate recomputation problem in an approximate but highly scalable way. The gains brought by the proposed framework have been confirmed in both simulation and emulation, through a testbed implemented using state-of-the-art WebRTC software. In an emulated scenario where a single sending peer equipped with three encoders transmits to 28 receivers, the proposed framework improves the average received video bitrate up to 15%, compared to a static solution where the encoding bitrates do not change over time.

VII. Conclusions

In this dissertation, several optimizations have been proposed for the efficient delivery of adaptive video streaming services. These solutions make extensive use of network components, developed and deployed to support the video clients and improve their final QoE. Several future opportunities and challenges can also be identified both in the domain of network-assisted solutions and adaptive streaming in general, which can profit from the work presented in this dissertation. New streaming applications will appear in the near future, as immersive video streaming, which are extremely bandwidth intensive and require careful delivery in terms of quality and latency. Also, ultra-low latency and high-mobility applications will become more and more important. These scenarios concern not only real-time applications, but also machine-to-machine communication, as in car-to-car entertainment systems or drones surveillance. New network paradigms as 5G and softwarized networks, which can support both traditional forms of communication and more unstructured ones, will be essential to provide the best service to the final users.

References

[1] S. Petrangeli, M. Claeys, S. Latré, J. Famaey, and F. De Turck, "A multi-agent q-learning-based framework for achieving fairness in http adaptive streaming," in *Network Operations and Management Symposium (NOMS), 2014 IEEE*, May 2014, pp. 1–9.

[2] S. Petrangeli, N. Bouten, E. Dejonghe, J. Famaey, P. Leroux, and F. De Turck, "Design and evaluation of a dash-compliant second screen video player for live events in mobile scenarios," in *International Symposium on Integrated Network Management (IM), 2015 IEEE/IFIP*, May 2015.

[3] S. Petrangeli, T. Wauters, R. Huysegems, T. Bostoen, and F. De Turck, "Network-based dynamic prioritization of http adaptive streams to avoid video freezes," in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, May 2015, pp. 1242–1248.

[4] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. De Turck, "Qoe-driven rate adaptation heuristic for fair adaptive video streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 2, pp. 28:1–28:24, Oct. 2015.

[5] S. Petrangeli, T. Wauters, R. Huysegems, T. Bostoen, and F. De Turck, "Software-defined network-based prioritization to avoid video freezes in http adaptive streaming," *International Journal of Network Management*, vol. 26, no. 4, pp. 248–268, 2016.

[6] J. van der Hooft, S. Petrangeli, T. Wauters, R. Huysegems, T. Bostoen, and F. De Turck, "An http/2 push-based approach for low-latency live streaming with super-short segments," *Journal of Network and Systems Management*, pp. 1–28, 2017.

[7] R. Huysegems, J. van der Hooft, T. Bostoen, P. Rondao Alface, S. Petrangeli, T. Wauters, and F. De Turck, "Http/2-based methods to improve the live experience of adaptive streaming," in *Proceedings of the 23rd ACM International Conference on Multimedia*, ser. MM '15. New York, NY, USA: ACM, 2015, pp. 541–550.

[8] J. van der Hooft, S. Petrangeli, M. Claeys, J. Famaey, and F. De Turck, "A learning-based algorithm for improved bandwidth-awareness of adaptive streaming clients," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2015, pp. 131–138.

[9] J. van der Hooft, S. Petrangeli, T. Wauters, R. Huysegems, P. Alface, T. Bostoen, and F. De Turck, "Http/2-based adaptive streaming of hevc video over 4g/lte networks," *IEEE Communications Letters*, vol. 20, no. 11, pp. 2177–2180, Nov 2016.

[10] S. Petrangeli, V. Swaminathan, M. Hosseini, and F. De Turck, "An http/2-based adaptive streaming framework for 360-degree virtual reality videos," in *Proceedings of the 2017 ACM on Multimedia Conference*, ser. MM '17. New York, NY, USA: ACM, 2017, pp. 306–314.

[11] S. Petrangeli, N. Bouten, M. Claeys, and F. D. Turck, "Towards svc-based adaptive streaming in information centric networks," in *2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, June 2015, pp. 1–6.

[12] S. Petrangeli, V. Swaminathan, M. Hosseini, and F. De Turck, "Improving virtual reality streaming using http/2," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, ser. MMSys'17. New York, NY, USA: ACM, 2017, pp. 225–228.

[13] J. van der Hooft, S. Petrangeli, N. Bouten, T. Wauters, R. Huysegems, T. Bostoen, and F. D. Turck, "An http/2 push-based approach for svc adaptive streaming," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, April 2016, pp. 104–111.

[14] S. Petrangeli, T. Wu, T. Wauters, R. Huysegems, T. Bostoen, and F. D. Turck, "A machine learning-based framework for preventing video freezes in http adaptive streaming," *Journal of Network and Computer Applications*, vol. 94, pp. 78 – 92, 2017.

[15] S. Petrangeli, P. V. Staey, M. Claeys, T. Wauters, and F. D. Turck, "Energy-aware quality adaptation for mobile video streaming," in *2016 12th International Conference on Network and Service Management (CNSM)*, Oct 2016, pp. 253–257.

[16] T. Wu, S. Petrangeli, R. Huysegems, T. Bostoen, and F. De Turck, "Network-based video freeze detection and prediction in http adaptive streaming," *Computer Communications*, vol. 99, pp. 37–47, 2017.

[17] S. Petrangeli, J. van der Hooft, T. Wauters, R. Huysegems, P. R. Alface, T. Bostoen, and F. De Turck, "Live streaming of 4k ultra-high definition video over the internet," in *Proceedings of the 7th International Conference on Multimedia Systems*, ser. MMSys '16. New York, NY, USA: ACM, 2016, pp. 27:1–27:4.

[18] S. Petrangeli, N. Bouten, E. Dejonghe, J. Famaey, P. Leroux, and F. D. Turck, "Design and evaluation of a dash-compliant second screen video player for live events in mobile scenarios," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2015, pp. 894–897.

[19] R. I. T. da Costa Filho, M. C. Luizelli, M. T. Vega, J. van der Hooft, S. Petrangeli, T. Wauters, F. De Turck, and L. P. Gaspary, "Predicting the performance of virtual reality video streaming in mobile networks," in *Proceedings of the 9th ACM Multimedia Systems Conference*, ser. MMSys '18. New York, NY, USA: ACM, 2018, pp. 270–283.

[20] S. Petrangeli, J. V. D. Hooft, T. Wauters, and F. D. Turck, "Quality of experience-centric management of adaptive video streaming services: Status and challenges," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 14, no. 2s, pp. 31:1–31:29, May 2018.

[21] J. van der Hooft, C. D. Boom, S. Petrangeli, T. Wauters, and F. D. Turck, "An http/2 push-based framework for low-latency adaptive streaming through user profiling," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, April 2018, pp. 1–5.

[22] J. van der Hooft, S. Petrangeli, T. Wauters, R. Rahman, N. Verzijp, R. Huysegems, T. Bostoen, and F. D. Turck, "Analysis of a large multimedia-rich web portal for the validation of personal delivery networks," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, May 2017, pp. 714–719.

[23] S. Petrangeli, D. Pauwels, J. van der Hooft, T. Wauters, F. De Turck, and J. Slowack, "Improving quality and scalability of webrtc video collaboration applications," in *Proceedings of the 9th ACM Multimedia Systems Conference*, ser. MMSys '18. New York, NY, USA: ACM, 2018, pp. 533–536.

[24] S. Petrangeli, D. Pauwels, J. van der Hooft, J. Slowack, T. Wauters, and F. D. Turck, "Dynamic video bitrate adaptation for webrtc-based remote teaching applications," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, April 2018, pp. 1–5.

[25] D. Pauwels, J. van der Hooft, S. Petrangeli, T. Wauters, D. D. Vleeschauwer, and F. D. Turck, "A web-based framework for fast synchronization of live video players," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, May 2017, pp. 524–530.

[26] S. Petrangeli and F. De Turck, "Qoe-centric management of advanced multimedia services," in *Intelligent Mechanisms for Network Configuration and Security*. Cham: Springer International Publishing, 2015, pp. 50–55.

[27] J. van der Hooft, D. Pauwels, C. De Boom, S. Petrangeli, T. Wauters, and F. De Turck, "Low-latency delivery of news-based video content," in *Proceedings of the 9th ACM Multimedia Systems Conference*, ser. MMSys '18. New York, NY, USA: ACM, 2018, pp. 537–540.

[28] J. van der Hooft, C. D. Boom, S. Petrangeli, T. Wauters, and F. D. Turck, "Performance characterization of low-latency adaptive streaming from video portals," *IEEE Access*, vol. 6, pp. 43 039–43 055, 2018.

[29] S. Petrangeli, D. Pauwels, J. van der Hooft, M. Žiak, J. Slowack, T. Wauters, and F. De Turck, "A scalable webrtc-based framework for remote video collaboration applications," *Multimedia Tools and Applications*, Aug 2018.

[30] S. Petrangeli, J. Famaey, and S. Latré, "Fair adaptive streaming," US patent US9826016B2, Granted on 21 November 2017. [Online]. Available: https://patents.google.com/patent/US9826016B2/en

[31] S. Petrangeli, "Qoe-centric network-assisted delivery of adaptive video streaming services," Doctoral Thesis, Ghent University, 2018. [Online]. Available: https://biblio.ugent.be/publication/8553995/file/8553998

[32] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, "What happens when http adaptive streaming players compete for bandwidth?" in *Proceedings of the 22Nd International Workshop on Network and Operating System Support for Digital Audio and Video*, ser. NOSSDAV '12. ACM, 2012, pp. 9–14.

[33] CONVIVA, "Mid-year 2015 update: Conviva viewer experience report," http://www.conviva.com/conviva-viewer-experience-report/midyear-vxr-2015/, 2015.

[34] H. E. Egilmez, S. Civanlar, and A. M. Tekalp, "An optimization framework for qos-enabled adaptive video streaming over openflow networks," *IEEE Transactions on Multimedia*, vol. 15, no. 3, pp. 710–715, April 2013.

[35] K. T. Bagci, K. E. Sahin, and A. M. Tekalp, "Queue-allocation optimization for adaptive video streaming over software defined networks with multiple service-levels," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sept 2016, pp. 1519–1523.

[36] G. Cofano, L. De Cicco, T. Zinner, A. Nguyen-Ngoc, P. Tran-Gia, and S. Mascolo, "Design and experimental evaluation of network-assisted strategies for http adaptive streaming," *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2017.

[37] S. Wei and V. Swaminathan, "Low latency live video streaming over http 2.0," in *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*, ser. NOSSDAV '14. New York, NY, USA: ACM, 2014, pp. 37:37–37:42.

[38] C. James, E. Halepovic, M. Wang, R. Jana, and N. K. Shankaranarayanan, "Is multipath tcp (mptcp) beneficial for video streaming over dash?" in *2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Sept 2016, pp. 331–336.

[39] Z. Lu, V. S. Somayazulu, and H. Moustafa, "Context-adaptive cross-layer tcp optimization for internet video streaming," in *2014 IEEE International Conference on Communications (ICC)*, June 2014, pp. 1723–1728.

[40] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Rusboost: A hybrid approach to alleviating class imbalance," *Trans. Sys. Man Cyber. Part A*, vol. 40, no. 1, pp. 185–197, Jan. 2010.