Interaction of Control Loops on Application, Transport, and Network Layer

Thomas Zinner*, Giuseppe Cofano§, Florian Wamser*, Luca De Cicco§, Saverio Mascolo§, Phuoc Tran-Gia*
*Institute of Computer Science University of Würzburg, Germany.
Email: {zinner|florian.wamser|trangia}@informatik.uni-wuerzburg.de
§ Dipartimento di Ingegneria Elettrica e dell'Informazione (DEI), Politecnico di Bari, Italy
Email: {giuseppe.cofano|luca.decicco|saverio.mascolo}@poliba.it

I. Introduction

A common occurrence in some network scenarios, such as home or access network, is the simultaneous presence of different applications competing for the available network resources. To provide a high user-perceived application quality (QoE), these applications pose diverse requirements to the network which have to be fulfilled. Data packets in the network, however, are still mostly forwarded in a best-effort manner without taking relevant information like application-type or application-state into account. Resource distribution between different competing flows is typically realized by the TCP-congestion algorithm which enables a fair share of the network resources on an end-to-end basis. This QoS fair share may result in an unequal QoE distribution among concurrent applications, as for the example of a parallel video streaming application and a file download. In this scenario, the shared bandwidth may not be enough to fulfill the bandwidth requirements of the video clip resulting in video stalls and an intense reduction of the QoE [1]. The QoE for the file download may be less sensitive to the amount of available network resources resulting in a higher QoE. In order to reduce the impact of uncontrollable resource fluctuations and bottlenecks on the application quality, applications may adjust their demands to the available network resources [2]. This adaptation is typically achieved by implementing an application-specific control loop which adjusts the application behavior like the bandwidth demand based on monitoring parameters like the current application quality or an estimation of the available network resources.

For instance, in the case of HTTP-based adaptive video streaming (HAS), a control algorithm at the client is employed to avoid stalling events while selecting the best video quality for the current network conditions [3]. However, these techniques can have only a limited impact in providing the desired QoE, since they cannot act at the network level, as discussed in the next section.

II. Resource Distribution between Download and a HAS

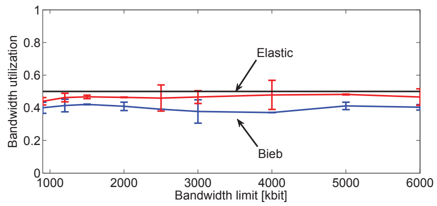
In this Section we discuss the interaction between TCP congestion control and the video-specific adaptation algorithm employed to maximize the video QoE. The adaptation algorithm adapts the video quality on-the-fly to the current network conditions in order to avoid video playback interruptions. Ideally, the video quality should match the fair bandwidth share (QoS fair share). Adaptation algorithms rely on a playout buffer at the client to pre-fetch video content in order to avoid stalling events due to bandwidth fluctuations. Based on quality indicators like the estimated current bandwidth and the current playout buffer level they select the video quality. The design of the adaptation algorithm has to carefully take into account several factors, such as the specific use case (VoD or Live Streaming) or codec dependencies (single-layer or multi-layer codec).

In this paper we consider two algorithms, Elastic [4] and BIEB [5]. Elastic has been proposed for single-layer flows and makes use of a feedback control technique known as feedback linearization. BIEB on the other hand is optimized for multi-layer codecs taking dependencies between the layers into account. Both of them have been designed to overcome the limitations of some algorithms proposed in the literature (cf. [3]), that suffer from severe fairness and poor utilization issues when competing with TCP flows. In our testbed two flows, a large file download and an HTTP-based adaptive video streaming application, share the same linkwhose bandwidth is shaped by means of the tool Netem. Our testbed is composed of three computers placed in linear topology: the client pc, both executing the file download and playing the video, the shaper pc, running the shaper tool, and the HTTP server Apache serving both the file and the video content. The link bandwidth has been shaped by means of the tool Netem. For the Elastic algorithm the video "Big Buck Bunny" has been employed, encoded with 5 qualities. For the BIEB algorithm the video "Tears of Steel" has been employed, encoded with 3 qualities. The results for the bandwidth distribution in case of concurrent applications, as well as the normalized playback quality are illustrated in Figure 1. Figure 1a highlights the bandwidth utilization obtained by the two algorithms, which in this scenario corresponds to the QoS fair share. For both algorithms the bandwidth utilization is not sensitive to the bandwidth input and higher than 40 %. Further, Elastic outperforms BIEB and is almost able to obtain the ideal QoS fair share, depicted by the black line. In Figure 1b the normalized average video quality is shown in order to confirm that the adaptation logic is working. For both algorithms it is shown that the average video quality increases as the bandwidth input is increased. Please note that both algorithms can not be compared due to different video content, utilized coded, as well as encoding configuration.

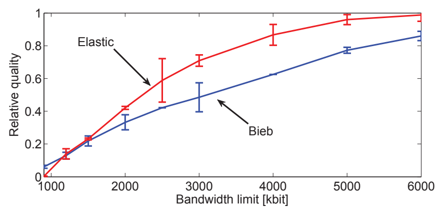
III. Application-aware Network Management

In this Section we aim at motivating the need for an application-aware network management. Let us consider again the above-mentioned experiment. Video quality adaptation al-

978-1-4799-0913-1/14/$31.00 © 2014 IEEE



(a) Bandwidth utilization



(b) Normalized video quality (relative to maximum quality of the transmitted video clip

Fig. 1: Bandwidth utilization in the presence of a concurrent file download and normalized video quality for Elastic and Bieb.

gorithms have been roughly able to obtain the QoS fair share. However, this may not be sufficient. If the QoS fair share drops below the bit rate of the minimum video quality, the video playback would be interrupted several times. This results in a bad QoE, even if the algorithm is able to reach its QoS fair share in the link. Since file download is less sensitive to a low bandwidth, cf. [6], it might be better to prioritize the video flow in such cases, ensuring a continuous playback at the minimum quality, and thus providing a better average QoE for all involved users. In other words, QoS fair share is different from QoE fair share.

Differentiating the service at the network, for instance by allocating more bandwidth to the video flow, is a possible approach to go towards QoE fairness. The chance is offered by the recent SDN paradigm, which allows agile and dynamic network management and programmability. SDN can be considered as a natural enabler for such applications, since it separates the network control plane from the forwarding data plane and provides a centralized view of the distributed state of the network. By introducing an external and programmable network control plane, SDN creates a flexible, adaptable, and open interface to the network data plane. Via the "Northbound-API" [7] applications may provide information to the network control plane, which then can control the data plane.

The idea to utilize SDN to provide a better application-network interaction is also addressed by other approaches, c.f. [8], [9], however a holistic assessment of influence factors in multi-application scenarios is still missing.

IV. Interacting Control Loops

Application-aware management (or In-Network Traffic Optimization) is a promising approach to optimize the perceived QoE of different applications, but comes at the price of additional complexity. With application-aware management, in fact, three different control loops [10] play a part in maximizing the QoE: the TCP congestion control algorithm, the HTTP adaptation algorithm and the network resource management algorithms. In this Section we summarize some issues arising with application-aware management for video streaming. In general, application-aware networking can be based either on an information exchange with the client or on flow detection techniques (packet inspection). With the information exchange approach, in principle, any kind of information can be exploited by the network to maximize QoE: 1) information on parameters related to the video content and its encoding, such as video quality bitrates; 2) parameters related to the client and its device, such as the device resolution; 3) parameters related to the quality adaptation algorithm during its execution, such as the current playout buffer level and the current video quality. A broad range of optimization possibilities is open, such as admission control, dedicated bandwidth allocation, or flow prioritization. However, control loops interaction and different timescales, scalability issues, and communication overhead require a careful investigation.

Future work will address these research questions and highlight the prospects of application-aware networking using SDN.

References

[1] T. Zinner, M. Jarschel, A. Blenk, F. Wamser, and W. Kellerer, "Dynamic Application-Aware Resource Management Using Software-Defined Networking: Implementation Prospects and Challenges," in *IFIP/IEEE International Workshop on Quality of Experience Centric Management (QCMan)*, (Krakow, Poland), May 2014.

[2] T. Hoßfeld, M. Fiedler, and T. Zinner, "The QoE Provisioning-Delivery-Hysteresis and Its Importance for Service Provisioning in the Future Internet ," in *Proceedings of the 7th Conference on Next Generation Internet Networks (NGI)*, (Kaiserslautern, Germany), June 2011.

[3] 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*, vol. 17, Mar. 2015.

[4] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "Elastic: a client-side controller for dynamic adaptive streaming over http (dash)," in *Proc. of Packet Video Workshop '13*, Dec. 2013.

[5] C. Sieber, T. Hofeld, T. Zinner, P. Tran-Gia, and C. Timmerer, "Implementation and user-centric comparison of a novel adaptation logic for dash with svc," in *IM'13*, pp. 1318–1323, 2013.

[6] S. Egger, T. Hoßfeld, R. Schatz, and M. Fiedler, "Waiting Times in Quality of Experience for Web Based Services," in *QoMEX 2012*, (Yarra Valley, Australia), July 2012.

[7] M. Jarschel, T. Zinner, T. Hoßfeld, P. Tran-Gia, , and W. Kellerer, "Interfaces, Attributes, and Use Cases: A Compass for SDN," *IEEE Communications Magazine*, vol. 52, June 2014.

[8] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards network-wide qoe fairness using openflow-assisted adaptive video streaming," in *Proceedings of the 2013 ACM SIGCOMM Workshop on Future Human-centric Multimedia Networking*, FhMN '13, (New York, NY, USA), pp. 15–20, ACM, 2013.

[9] A. D. Ferguson, A. Guha, C. Liang, R. Fonseca, and S. Krishnamurthi, "Participatory networking: An api for application control of sdns," *SIGCOMM Comput. Commun. Rev.*, vol. 43, pp. 327–338, Aug. 2013.

[10] G. Cofano, L. De Cicco, and S. Mascolo, "A control architecture for massive adaptive video streaming delivery," in *Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming*, VideoNext '14, (New York, NY, USA), pp. 7–12, ACM, 2014.