Technical Report on DFG Project SDN-App

SDN-enabled Application-aware Network Control Architectures and their Performance Assessment

Andreas Blenk[1], Wolfgang Kellerer[1], Tobias Hoßfeld[2]

July 7, 2020

[1] Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik
Theresienstr. 30, 80333 München, Deutschland

[2] Julius-Maximilians-Universität Würzburg
Fakultät für Mathematik und Informatik
Lehrstuhl 3 für Kommunikationsnetze
Am Hubland, 97074 Würzburg, Deutschland

Contents

1 Introduction: DFG SDN-App

The DFG project "SDN-enabled Application-aware Network Control Architectures and their Performance Assessment" *(DFG SDN-App)* focused in phase 1 (Jan 2017 – Dec 2019) on software defined networking (SDN). Being a fundamental paradigm shift, SDN enables a remote control of networking devices made by different vendors from a logically centralized controller. In principle, this enables a more dynamic and flexible management of network resources compared to the traditional legacy networks.

Phase 1 focused on multimedia applications and their users' Quality of Experience (QoE). Two approaches were identified to improve the overall network performance, hence, the perceived user QoE. While taking network information into account, the *first approach* focuses on the application control plane, e.g., taking global quality decisions for video streams. The *second approach* investigates the network control plane while considering application information. Based on this, mechanisms and approaches for a joint optimization of application control and SDN network control were investigated and analyzed. In particular, mechanisms to ensure QoE of Internet applications by utilizing SDN-enabled application-aware network control architectures were developed. This also included the consideration of the overall QoE for multi-application scenarios. The performance evaluation and QoE assessment of those mechanisms and joint application and network control plane showed an improvement of QoE. Due to the coordination between application layer and network control plane, this QoE improvement was achieved in Phase 1 of DFG SDN-App.

Phase 2 of DFG SDN-App consequently extends the considered scenario: beside taking care of end user-oriented multimedia applications and QoE (Phase 1), it now integrates time-critical (industrial) services and their requirements — cases where end users do not always play a dominant role. In particular, we focus on *soft and hard real-time requirements* of those services. In the context of industrial networks (e.g. smart manufacturing) or automotive use cases (e.g. communication network within a self-driving car), the real-time requirements manifest, e.g., in a hard upper bound of the end-to-end delay, which must be ensured by the communication network. Beside industrial networks (e.g., ring topology with actuators/sensors as end hosts), data center networks (e.g., tree topology with virtual machines on servers as end hosts) or wide-area networks (e.g., base stations and data centers as end hosts) with time-critical or even real-time requirements will be considered in phase 2. The *overall goal* is an architecture that can allow both application types, i.e., industrial with soft and hard real-time requirements and best-effort, to coexist on the same infrastructure.

2 Results from Phase 1

(Jan 2017 – Dec 2019)

This documents reports the achievements of the first phase (Jan 2017 – Dec 2019) of the research project "SDN-enabled Application-aware Network Control Architectures and their Performance Assessment" (DFG SDN-App), which is jointly carried out by the Technical University of Munich, Technical University of Berlin, and University of Würzburg. The project started at the institutions in Munich and Würzburg in January 2017 and lasted until December 2019.

In Phase 1, the project targeted the development of fundamental control mechanisms for network-aware application control and application-aware network control in Software Defined Networks (SDN) so to enhance the user perceived quality (QoE). The idea is to leverage the QoE from multiple applications as control input parameter for application- and network control mechanisms. These mechanisms are implemented by an Application Control Plane (ACP) and a Network Control Plane (NCP). In order to obtain a global view of the current system state, applications and network parameters are monitored and communicated to the respective control plane interface. Network and application information and their demands are exchanged between the control planes so to derive appropriate control actions. To this end, a methodology is developed to assess the application performance and in particular the QoE. This requires an appropriate QoE modeling of the applications considered in the project as well as metrics like QoE fairness to be utilized within QoE management.

In summary, the application-network interaction can improve the QoE for multi-application scenarios. This is ensured by utilizing information from the application layer, which are mapped by appropriate QoS-QoE models to QoE within a network control plane. On the other hand, network information is monitored and communicated to the application control plane. Network and application information and their demands are exchanged between the control planes so to derive appropriate control actions.

The achievements in Phase 1 are presented in the following, where we first take the application perspective (application performance assessment, QoE modeling & QoE management) in Section 3 and then the network perspective (network performance assessment) in Section 4. Finally, this leads to application-aware control architectures and appropriate mechanisms.

3 Application and QoE Perspective

3.1 Application Performance Assessment

The developed application control plane considers the state of multiple end user applications and the current network state. The application control plane can choose from a set of mechanisms, such as direct application adaptation mechanisms or limited network adaptation mechanisms, while optimizing the current application performance. Due to the share of video traffic in the Internet, video streaming is of major interest. The work of [2] investigates several network-assisted streaming approaches that rely on active cooperation between video streaming applications and the network. A video control plane is established that enforces video quality fairness among concurrent video flows by solving a max-min fairness optimization problem at runtime. Two approaches are implemented in an SDN network: the first one allocates network bandwidth slices to video flows, and the second one guides video players in the video bitrate selection. The performance is assessed through several QoE-related metrics, video quality and switching frequency.

Schwarzmann et al. [16] presents an evaluation methodology to systematically assess application-network interaction mechanisms. Adaptive streaming is used as an example and the framework captures the QoE-relevant metrics as well as the amount of exchanged messages for application monitoring, application control, network monitoring, network control. The considered QoE metrics are the stalling pattern, the video quality based on the Structural Similarity Metric (SSIM), the number of quality switches, the initial delay before the video playout, as well as fairness based on Hossfeld's QoE fairness index [6]. The evaluation methodology [16] is also applied to investigate the benefits of variable video segments [17] instead of fixed duration video segments, e.g., of 2 s as implemented by some video platforms in the Internet. The results show that variable segment sizes, aligned to the video characteristics, outperform the fixed approach in 86 % of the evaluated cases with respect to video stalls.

A generic approach is provided in [1] that models the video buffer as a GI/GI/1 queue with pq-policy using discrete-time analysis. If the buffer level exceeds a threshold q upon arrival of a video segment, the segment requests are paused by the video player until the buffer level drops below a lower threshold p. This allows to model streaming behavior of common Internet video platforms. The model allows to accurately evaluate the impact of network and video bitrate dynamics on the video playback quality based on the buffer policy. The discrete time model is extended in [15] to compute QoE-relevant metrics for adaptive streaming. For evaluating QoE in practice, [25] provides a framework to derive QoE of adaptive video streaming by capturing the relevant QoS parameters and mapping them to QoE with different models from literature.

3.2 QoE Fairness and Optimal QoE

In general, the goal of QoE management is to maximize QoE, while ensuring fairness among users. However, the notions of fairness commonly applied in the QoS domain do not translate well to the QoE domain. In [7, 6], a definition for a QoE fairness index is introduced. The need for this definition of QoE fairness is demonstrated, since it does not, due to the nature of QoS to QoE mappings for most services, necessarily follow from QoS fairness. Hossfeld et al. [6] show that commonly used QoS fairness metrics such as Jain's fairness index are not suitable for quantifying QoE fairness, despite being used for that purpose in the literature. Contrary, the proposed QoE fairness index fulfills a number of desirable qualities, and it is intuitively simple to understand. This metric serves as basis for several works in Phase 1, e.g., [13, 2, 22, 23].

Moldovan et al. [13] analyze optimal QoE and QoE fairness for video streaming with multiple users. An optimization problem is formulated that can benchmark any solution with the theoretical optimum in terms of adaptation strategies and corresponding segment downloads across multiple users under given bandwidth constraints. Different mechanisms to achieve fairness in practical implementations are analyzed concerning the quality and QoE fairness. The performance evaluation is conducted with a discrete-event simulator for adaptive streaming of several users in a network.

Furthermore, Moldovan et al. [12] investigates video adaptation strategies as implemented by commercial video platforms. The trade-off between the average video quality and switches in the quality during playout is quantified by solving an optimization problem for the best adaptation strategy maximizing QoE. The results show that the video quality can be significantly increased by allowing few additional switches. This novel discovery is important for QoE-management in practice.

Hossfeld et al. [9] investigate the impact of the system's delay on QoE for (1) video streaming, (2) authentication in social networks, (3) wireless 3G Internet connection setup. Existing QoE models are used to map the response time in the system, corresponding to the waiting time for users until the service is setup, to Mean Opinion Scores (MOS) as a measure of QoE. The system is then evaluated in terms of overall QoE and QoE fairness for the three services considered, under different load scenarios. The results show how differently users of different services perceive such systems and response times. The model further allows the dimensioning of the system with respect to QoE.

3.3 Phase 1: QoE Management

Finally, different QoE management mechanisms are provided in Phase 1 [12, 20, 10, 2, 17, 16, 26, 22, 23], which may take care of QoE fairness and are evaluated with the frameworks mentioned above [2, 17, 16, 26, 22, 23] or benchmarked with the QoE optimal solution [13, 12, 20, 10, 22, 23].

The work [10] extends video adaption mechanisms by including context awareness in order to overcome the variability, instability and unpredictability of network conditions disturbing QoE. The context awareness, e.g., locations of users and the spatial map of radio network quality in LTE networks, is combined with the adaptive streaming logic in the application control plane resulting into a proactive client-based video streaming strategy. The results show that such a context-aware strategy manages to successfully mitigate stalling in light of network connectivity problems, such as an outage. The performance of this strategy is compared to the optimal case, as well as by considering situations where the awareness of the context lacks reliability.

In [22, 23] the potential gain of QoE-aware control of several applications in a system is investigated. The problem is divided in first, how to define application- and QoE-aware control and second, how to distribute the shares of network resources based on the needs of the applications. The problem of determining the potential gain is formulated as a mixed-integer linear program (MILP) that calculates the fair shares of the available capacity per application using utility functions, a given network topology and a fairness criteria. The utility functions provide the relationship between network resources, e.g., throughput and delay, and the resulting QoE for the user of the application. Two experiment set-ups are required and developed as part of [22, 23]. First, the behavior of applications under network resource constrains is investigated in a testbed set-up in Würzburg using the framework in [16]. The second set-up developed at TUM replicates an enterprise environment consisting of a network topology and server and client applications. The server and client applications were distributed on multiple physical machines and have to share a bandwidth constrained link. Network control is performed by the developed application-aware architecture (see below). Software agents are deployed on the end-hosts that report active applications on the hosts to a logically centralized network controller. The network controller uses the empirical utility functions and the problem formulation to determine a max-min fair resource allocation and per-application forwarding graph. The resource allocation decisions are then pushed to the forwarding devices of the data-plane and the network resource constrains are pushed to the developed Kernel module at the end-hosts. The results show that the QoE for the different applications is improved with the application-aware control architecture. Furthermore, the available resources are distributed optimally with respect to the specified fairness criteria. Hossfeld et al. [4] analyzes QoE management in general by considering QoE fairness, user diversity, and different QoE metrics like MOS or Poor-or-Worse (PoW) ratio [5]. It is shown how the choice of QoE measures, the importance of QoE fairness, and the variations between users affects the optimal QoE management choices for service providers. Thus, the definition of utility functions requires a careful consideration of QoE metrics and relevance of QoE fairness. [8] provides an overview of state-of-the-art findings in QoE modeling and discusses emerging concepts and the challenges they raise with respect to managing QoE for networked media services.

3.4 Survey on QoE Management Architectures

Finally, [24] provides an extensive survey of emerging concepts and challenges for QoE management of multimedia services. SDN and NFV form the key infrastructure for nearly any upcoming QoE management approaches. However, standardization issues and the complex architectures involved may limit the applicability of QoE management mechanisms in the wild. Virtualization, both in the network (SDN and NFV) and in the service domain (cloud) will be the dominant deployment approach in the years to come. This presents interesting opportunities for QoE management, since QoE components (e.g, QoE optimization, co-located with an orchestrator as in an NFV MANO-like architecture) could have a much more comprehensive view of both the application and network layers, and more importantly, the ability to actuate on both of them. Such a generic architecture is described in [24], wherein most of the current (and likely upcoming) QoE management approaches can be inscribed. In particular, application-network interaction as in DFG SDN-App is a core element. Three main layers are identified: virtualized networks and infrastructure, the virtualized network functions, and a service layer, which relies on a northbound API to deal with the underlying virtualized environments. QoE management is done via a feedback loop (in the Service Assurance block), which gathers monitoring data, and informs the service layer (which in turn can pass the information to, e.g., the NFV MANO or the SDN controller, for concrete action). The top layer includes, besides the traditional OSS/BSS10 functions, the notion of the "telco cloud'", whereby service developers could deploy their whole services on top of a telco's own infrastructure. The architecture therefore includes an SDN layer, and NFV layer, the northbound API and Service Innovation layer, and a Service Assurance component, which would take care of the QoE data aggregation, analytics, etc. Schwarzmann et al. [14] surveys QoE management for real-time multimedia services supported by SDN, as well as big data analytics and methods that are used for QoE management. The benefits of incorporating big data analytics in QoE management are evaluated for video streaming services. A high-level view of an SDN-based architecture for QoE management enriched with big data analytics' functional blocks is provided and the corresponding challenges are summarized.

4 Network Performance Assessment

4.1 Hardware-based forwarding devices

Another objective of DFG SDN-App is the analysis up to which extent the performance of current networks can support fine-grained application-aware control. Sieber et al. [19] determines the theoretical performance bounds of the information exchange of application- and control-plane with the data plane in terms of change requests.

To do so, we first have to understand the characteristics of the forwarding devices in the network regarding their ability to process and implement the desired changes on the data plane. A measurement study of the devices in our testbed reveals a variation of an upper bound for the reconfiguration frequency of 1.5 requests/s to 1000 requests/s. Then, the analytical bound is derived using an M/D/1 queueing model for the device, which takes into account the topology of the network and employs the centrality of a device to determine the load of the device. An analytical upper bound for the information exchange frequency based on the topology of the network and the deployed forwarding devices is derived. The results show that even a low number of slow devices can severely restrict the maximum possible information exchange frequency. The analytical model combined with the measurements is not only applicable to this project, but also casts doubts on many works in the domain of networking where changes to the data plane are assumed instantaneous and the change frequency is assumed to be unconstrained.

P4 (Programming Protocol-Independent Packet Processors) is the next step in the evolution of SDN and allows the programmability of switches to a certain extent. In [11] the performance impact of modifying packets with P4-enabled hardware is analyzed. One of P4s promoted features is the possibility to modify packets at line rate. One proposed application by the P4 community is to add information, e.g., the processing latency of the packet itself, to packet headers during the processing in a switch. When continuously monitoring this information for the whole network, it is possible to identify processing bottlenecks without any additional controller involvement. The contribution of package is to analyze the performance impact of packet header modifications by adding and removing VLAN headers to a TCP packet stream. The analysis focuses on the processing latency within the switch and the impact on the resulting network bandwidth. As a result, we observe that thoughtful development and deployment of P4 programs is required as well as comprehensive performance evaluations. Nevertheless, for Phase 2 in DFG SDN-App, P4 may be promising to detect, e.g., congestion in switches at different levels (buffers and different processing pipelines) and to inform senders, e.g., in a data center scenario, with more explicit information. By monitoring the load of the switches, advanced counter-actions could be taken. If the network control is able to detect congestion, or even better to predict congestion in advance, precaution actions might be taken. A switch experiencing high load could generate packets and transmit them to the senders causing the congestion. With P4, this scenario is realizable as more informative packet generation on side of the switches is possible.

4.2 Software-based forwarding devices

In [19] we only consider (mostly) hardware-based forwarding devices. For the hardware-based devices we observe mostly deterministic behavior and variance in speed was only observed between different devices. In [18], we take a closer look at forwarding devices deployed on commodity servers, also referred to as software-based forwarding devices. While hardware-based forwarding devices are designed from ground up for packet forwarding, software-based run on commodity servers, which are designed for general compute workloads and not optimized for network workloads. Specifically, we take a look at the influence of the memory architecture and the CPU cache sharing on the network performance. The results show that the limited CPU cache decreases the maximum packet forwarding and processing capacity of commodity hardware. Furthermore, the results highlight the problem of modern memory segregation designs, where multiple CPU sockets are used for one server. Here, the allocation of the forwarding processes to the architecture at hand has to be done with care to prevent costly data transfers between two CPU sockets. Based on these findings, we propose the first scheduler for CPU caches for network workload in [3]. The scheduler incrementally adapts the allocated cache to converge to a state of optimal cache allocation. With the scheduler, we are able to decrease the maxi- mum CPU utilization over all processes handling network workloads by up to 20 % in the investigated scenarios.

4.3 Control Plane

Finally, we shift our focus from the data plane to the control plane. In [21], we focus the evaluation on the performance of the control plane and proposed a machine learning pipeline do learn the information exchange performance of a network controller instance at runtime. An accurate estimation of the upper performance bound of a specific instance is required for system dimensioning to avoid underutilization of available resources and prevent overutilization, which leads to a delayed information exchange and reaction to events in the network. The pipeline consists at the heart of an orthogonal distance regression (ODR) with sample weighting and a Support Vector Machine (SVM). The SVM provides outlier detection in order to determine when the learned model differs from the actual model. The performance characteristics can change for example when its workload changes or the available resources on the host machine change. The ODR with sample weighting trains a performance model at runtime with the samples gathered from a monitoring component. The evaluation of the pipeline shows that it can provide accurate estimations of the maximum performance bound even in cases where only low utilization of the controller instance is observed. Even so the pipeline is only tested for the control plane, the pipeline can also be applied to the application plane. There the estimation can be the base line for dimensioning the maximum number of supported applications in the network and for the frequency of the information exchange.

4.4 Application-aware Control Architectures

The multi-application control within DFG SDN-App requires the categorization of the application space, the identification of key performance indicators of applications for QoE-aware control and the design of the application plane – which was described in the previous paragraphs on "Application Performance Assessment", "QoE Fairness and Optimal QoE" and "QoE Management". Applications in the network are labeled with a class, e.g., video streaming or web browsing, and a communication intent. The intent describes the purpose of the communication of the application, e.g., video on demand vs. live video streaming, and the resource demand, e.g., a minimum of 1 Mbps throughput and a maximum of 100 ms end-to-end delay. To this end, an end-host software agent is implemented for enterprise environments that allows accurate identification of application flows at the source and/or destination of the flows. Furthermore, novel protocols for the communication between application instances and the application control plane are developed within DFG SDN-App. The categorization provides the input for the optimization algorithm(s) that determine the optimal application-aware resource allocation. The evaluation of the application-network inter-play shows that the application categorization in combination with accurate network control capabilities (see previous paragraph "Network Performance Assessment") provides a useful abstraction of single application instances and helps to improve QoE while taking into account fairness.

The application control architecture is described in detail in [22, 23]. The architecture consists of a logically centralized application controller that can either be a part or extension of the network controller or deployed as a separate entity. Furthermore, there are software agents executed on each end-host in the system. The agents are connected via an open HTTP/REST-based protocol with the application controller. The agents re-port the class and intent of each active application on the end-hosts to the controller. Additional to monitoring and reporting of active applications, the agent receives the amount of allocated resources per application from the application controller. The agent allocates the resources to each application through the network stack of the host's operating system.

Most of the required components and protocols did not exist prior to the project. The application controller and software agent was developed in Python as part of the project. We developed a Linux Kernel module in C to allocate network resources per application. The Kernel module provides a network queue per application with queue-based pacing. Pacing refers to the technique of inserting short pauses in-between consecutive packets to reach a certain target sending rate. The evaluation results show that pacing at end-hosts, in combination with the empirical utility functions, results in the targeted optimum QoE for the user. In summary, the proposed architecture is able to significantly improve the QoE of the users for multi-application classes. Furthermore, the available resources are distributed optimally with respect to the specified fairness criteria.

Bibliography

[1] Valentin Burger, Thomas Zinner, Lam Dinh-Xuan, Florian Wamser, and Phuoc Tran-Gia. “A generic approach to video buffer modeling using discrete-time analysis”. In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14.2s (2018), p. 33.

[2] Giuseppe Cofano, Luca De Cicco, Thomas Zinner, Anh Nguyen-Ngoc, Phuoc Tran-Gia, and Saverio Mascolo. “Design and performance evaluation of network-assisted control strategies for HTTP adaptive streaming”. In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 13.3s (2017), p. 42.

[3] Raphael Durner, Christian Sieber, and Wolfgang Kellerer. “Towards Reducing Last-Level-Cache Interference of Co-located Virtual Network Functions”. In: *Submitted to NetSoft 2019*. IEEE. 209, p. X.

[4] Tobias Hoßfeld, Poul E Heegaard, Lea Skorin-Kapov, and Martin Varela. “No silver bullet: QoE metrics, QoE fairness, and user diversity in the context of QoE management”. In: *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2017, pp. 1–6.

[5] Tobias Hoßfeld, Poul E Heegaard, Martín Varela, and Sebastian Möller. “QoE beyond the MOS: an in-depth look at QoE via better metrics and their relation to MOS”. In: *Quality and User Experience* 1.1 (2016), p. 2.

[6] Tobias Hoßfeld, Lea Skorin-Kapov, Poul E Heegaard, and Martín Varela. “A new QoE fairness index for QoE management”. In: *Quality and User Experience* 3.1 (2018), p. 4.

[7] Tobias Hoßfeld, Lea Skorin-Kapov, Poul E Heegaard, and Martin Varela. “Definition of QoE fairness in shared systems”. In: *IEEE Communications Letters* 21.1 (2016), pp. 184–187.

[8] Tobias Hoßfeld, Martin Varela, Poul E Heegaard, and Lea Skorin-Kapov. “Observations on emerging aspects in qoe modeling and their impact on qoe management”. In: *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2018, pp. 1–6.

[9] Tobias Hoßfeld, Martín Varela, Poul E Heegaard, and Lea Skorin-Kapov. “QoE Analysis of the Setup of Different Internet Services for FIFO Server Systems”. In: *International Conference on Measurement, Modelling and Evaluation of Computing Systems*. Springer. 2018, pp. 234–248.

[10] Eirini Liotou, Tobias Hoßfeld, Christian Moldovan, Florian Metzger, Dimitris Tsolkas, and Nikos Passas. "The Value of Context-Awareness in Bandwidth-Challenging HTTP Adaptive Streaming Scenarios". In: *Autonomous Control for a Reliable Internet of Services*. Springer, Cham, 2018, pp. 128–150.

[11] Christopher Metter. "Resilience, Availability, and Serviceability Evaluation in Software-defined Networks". PhD thesis. Universitiy of Würzburg, Jan. 2019.

[12] Christian Moldovan, Korbinian Hagn, Christian Sieber, Wolfgang Kellerer, and Tobias Hoßfeld. "Keep calm and don't switch: About the relationship between switches and quality in has". In: *2017 29th International Teletraffic Congress (ITC 29)*. Vol. 3. IEEE. 2017, pp. 1–6.

[13] Christian Moldovan, Lea Skorin-Kapov, Poul E Heegaard, and Tobias Hoßfeld. "Optimal Fairness and Quality in Video Streaming with Multiple Users". In: *2018 30th International Teletraffic Congress (ITC 30)*. Vol. 1. IEEE. 2018, pp. 73–78.

[14] Susanna Schwarzmann, Andreas Blenk, Ognjen Dobrijevic, Michael Jarschel, Andreas Hotho, Thomas Zinner, and Florian Wamser. "Big-Data Helps SDN to Improve Application Specific Quality of Service". In: *Big Data and Software Defined Networks*. The Institution of Engineering and Technology (IET), 2018. ISBN: 978-1-78561-304-3.

[15] Susanna Schwarzmann, Paula Breitbach, and Thomas Zinner. "Computing QoE-Relevant Adaptive Video Streaming Metrics Using Discrete-Time Analysis". In: *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. IEEE. 2019, pp. 1–6.

[16] Susanna Schwarzmann, Thomas Zinner, and Ognjen Dobrijevic. "Quantitative comparison of application–network interaction: a case study of adaptive video streaming". In: *Quality and User Experience* 2.1 (2017), p. 7.

[17] Susanna Schwarzmann, Thomas Zinner, Stefan Geissler, and Christian Sieber. "Evaluation of the Benefits of Variable Segment Durations for Adaptive Streaming". In: *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2018, pp. 1–6.

[18] Christian Sieber, Raphael Durner, Maximilian Ehm, Wolfgang Kellerer, and Puneet Sharma. "Towards optimal adaptation of nfv packet processing to modern cpu memory architectures". In: *Proceedings of the 2nd Workshop on Cloud-Assisted Networking*. ACM. 2017, pp. 7–12.

[19] Christian Sieber, Raphael Durner, and Wolfgang Kellerer. "How fast can you reconfigure your partially deployed SDN network?" In: *2017 IFIP Networking Conference (IFIP Networking) and Workshops*. IEEE. 2017, pp. 1–9.

[20] Christian Sieber, Korbinian Hagn, Christian Moldovan, Tobias Hoßfeld, and Wolfgang Kellerer. "Towards Machine Learning-Based Optimal HAS". In: *arXiv preprint arXiv:1808.08065* (2018).

[21] Christian Sieber, Andreas Obermair, and Wolfgang Kellerer. "Online learning and adaptation of network hypervisor performance models". In: *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE. 2017, pp. 1204–1212.

[22] Christian Sieber, Susanna Schwarzmann, Andreas Blenk, Thomas Zinner, and Wolfgang Kellerer. "Scalable Application-and User-aware Resource Allocation in Enterprise Networks Using End-host Pacing". In: *arXiv preprint arXiv:1811.02367* (2018).

[23] Christian Sieber, Susanna Schwarzmann, Andreas Blenk, Thomas Zinner, and Wolfgang Kellerer. "Scalable Application-and User-aware Resource Allocation in Enterprise Networks Using End-host Pacing". In: *Submitted to ACM Transactions on Modeling and Performance Evaluation of Computing Systems (ToMPECS)* (2018).

[24] Lea Skorin-Kapov, Martín Varela, Tobias Hoßfeld, and Kuan-Ta Chen. "A survey of emerging concepts and challenges for qoe management of multimedia services". In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14.2s (2018), p. 29.

[25] Sebastian Surminski, Christian Moldovan, and Tobias Hoßfeld. "Practical QoE Evaluation of Adaptive Video Streaming". In: *International Conference on Measurement, Modelling and Evaluation of Computing Systems*. Springer. 2018, pp. 283–292.

[26] Sebastian Surminski, Christian Moldovan, and Tobias Hoßfeld. "Saving bandwidth by limiting the buffer size in HTTP adaptive streaming". In: *MMBnet 2017* (2017), p. 5.