INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt 0000; **00**:1–22
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nem

Modeling and Performance Analysis of Application-Aware Resource Management

Florian Wamser[1]*, Andreas Blenk[2], Michael Seufert[1], Thomas Zinner[1], Wolfgang Kellerer[2], and Phuoc Tran-Gia[1]

[1]*Institute of Computer Science, University of Würzburg, Germany*
[2]*Chair of Communication Networks, Technische Universität München, Germany*

SUMMARY

Application-aware resource management is the approach to tailor access networks to have characteristics beneficial for the running applications and services. This is achieved through the monitoring and integration of key performance indicators from the application layer within the network resource management. The aim is to increase user-perceived quality and network resource efficiency by traffic engineering with the help of these indicators. Using analytic and simulative approaches, this paper provides analysis methods for network operators to quantify the performance gains of alternative resource allocation algorithms that implement the application-aware concept. Network operators can use the proposed methods to evaluate possible performance gain trade-offs between investing in a pure capacity increase (over-provisioning) and the realization of an application-aware resource allocation. For this purpose, we model and analyze the application quality trade-offs of four algorithms for application-aware resource management at a single link in varying traffic situations. The algorithms are chosen with respect to different complexity and implementation level in order to cover the design space in a systematic way. The study of the algorithms focuses on the application-layer performance for the most used applications today, namely web browsing and video streaming with constant bit-rate as well as HTTP progressive streaming with variable bit-rate. Application quality trade-offs are analyzed in particular for a high resource utilization at a bottleneck link. The results confirm that application-aware resource management outperforms best-effort resource management in terms of QoE. Moreover, our study provides guidelines for the selection and configuration of the evaluated algorithms.

Received . . .

KEY WORDS: Application-Aware Resource Management, Quality of Experience, Access Networks, QoS Management

*Correspondence to: Institute of Computer Science, University of Würzburg, Germany. E-mail: florian.wamser@informatik.uni-wuerzburg.de

1. INTRODUCTION

The way how customers use today's communication networks has changed dramatically in recent decades. In the past, users ran elastic and less interactive applications, such as file transmissions or emails. Today, they run a huge set of high quality applications, which are dominating today's network traffic, e.g., video streaming, online games and social networking. In particular for high quality applications, users do not require high data rates, they require a good application Quality of Experience (QoE). Among other subjective and objective factors that influence QoE (user expectations, hardware device capabilities, etc.), the applications have different and varying demands on the network in terms of QoE. This includes, for example, low latency for interaction, high data rate for data retrieval, or low packet loss for live content streaming. Consequently, just optimizing network parameters does not necessarily lead to QoE improvement without considering applications needs.

One solution in order to meet the application requirements is to employ over-provisioning in the network. As pure over-provisioning of network resources typically results in high costs and thus is not efficient from the economical point of view, new concepts have to be considered in order to improve networks' resource efficiency and application QoE simultaneously. As network operators do not have control over the server or customer premises, they have to introduce new concepts inside their networks. Application-aware resource management is such a concept that has emerged as a first step towards improving network resource efficiency in terms of QoE [7, 10, 16–18, 27, 34–36]. The idea of application-aware resource management is to orchestrate networks according to user demands that are based on application quality indicators with high impact on users' QoE. Network resource management and orchestration actions are, for example, changing the routing of application flows among the network or configuring the resource allocation of particular network links, e.g, by using different resource scheduling such as priority queuing or weighted fair queuing. In this way, the utility per allocated network resources in terms of QoE can directly be influenced. The quality indicators can be fetched from the network based on in-network measurements [21, 37]. A typical implementation of the concept consists of three entities: (1) the network and application monitoring deriving the key performance indicators and network characteristics [30,36], (2) a decision unit [36], and (3) a resource management action that changes the traffic handling within the network (e.g., traffic scheduling with queuing disciplines, routing, dynamic path selection) or the traffic produced by the applications (e.g. lowering video resolution/quality, server selection) [18, 36, 38]. Although network resources may be shared unequally, such a traffic handling even leads to an improved overall performance for all users in terms of QoE [10].

In existing work [10, 12, 31], the benefit of application-aware resource management mechanisms in terms of network resource efficiency and user satisfaction was shown for certain applications. However, the trade-offs between different application-aware resource management mechanisms for network links in a multi-application scenario in terms of complexity and implementation level have not been investigated. In detail, most research has been use-case driven and evaluated solutions for only tiny scenarios. The existing concepts have not provided a holistic evaluation approach that tries to understand underlying concepts of how to technically realize resource allocation algorithms for a multi-application scenario under varying traffic conditions. An algorithm designed for a particular setup may not be efficient under all circumstances. Consequently, a systematic analysis

DOI: 10.1002/nem

of application-aware resource management is important to quantify the real gain of the concept, in particular, when comparing to alternative solutions, such as network over-provisioning.

This paper provides methods that allow network operators quantifying the benefit of application-aware resource allocation algorithms for network links in three dimensions; for web browsing and progressive video streaming applications for constant bit rate and variable bit rate streams; for varying traffic situations and mixes; for four algorithms covering the space of application-aware resource allocation. This can be carried out by the proposed analysis algorithms and the simulation, by inserting the own network settings and traffic situations. Although an application-aware resource management should consider different network operations, such as changing the routing or server selection, we focus on resource allocation for network links, as there is already a high potential for improvement. In addition, improvements can be simply implemented by the operator, since only one link is touched, which is under the responsibility of the operator. The implementation complexity varies in terms of actions during run-time and the needed information about the applications. As we are interested in improving the performance for one network link, we abstract the link by a queuing model. By the model, application parameters can be derived which allow to draw conclusions about the resulting application QoE. The prerequisite is that this link represents a bottleneck in the network. In particular, only problems on the bottleneck link are addressed in this paper. Other influences outside the bottleneck link can not be solved and are not in focus of the algorithms considered in this paper. To conduct the evaluations, analytic models are used that are based on approaches using a Markov M/M/1 queue with processor sharing policies (M/M/1-PS) [6, 22]. In case of algorithms designed for managing dynamic applications like progressive video streaming with variable bit rate, a discrete event simulator implemented in Java is used. Using both, analytic and simulative approaches, a study of all algorithms for different traffic scenarios is performed in order to give selection and configuration guidelines for the algorithms.

In particular, the paper highlights the following contributions:

1. Modeling the application-aware resource management problem
2. Modeling four application-aware resource allocation algorithms with different complexity
3. Analytical and simulative analysis of the algorithms
4. Guidelines for the selection and configuration of the algorithms

The structure of the paper is as follows. First, we outline related work and discuss resource management options for operators in access networks in Section 2. We introduce the model for the considered scenario and formulate the resource allocation problem in Section 3. In Section 4, we introduce four resource allocation algorithms based on the application-aware concept. In Section 5, we explain the analytic and the simulative methodologies for the evaluation of the algorithms. In Section 6, we show the results of performance evaluation for the algorithms in terms of network load, traffic mix, and the aforementioned quality indicators. Finally, we draw conclusions in 7.

2. BACKGROUND AND RELATED WORK

Today, the dimensioning of the capacity of access networks is a crucial task for operators. On one hand, the cost must be kept as low as possible (*cost-efficiency*). On the other hand, sufficient capacity

Prepared using nemauth.cls
DOI: 10.1002/nem

must be available for any expected application or service to satisfy the users (*QoE improvement*). Although it was widely believed that network traffic may not be predictable, currently published measurements in literature about access networks show a predictable network traffic behavior [9,23]. Based on such a predictable behavior, an operator can analyze the trade-offs before upgrading the network. According to the traffic mix and patterns in the network, the best setup can be chosen in order to improve the overall network performance in terms of user-perceived service quality. One possibility to achieve this, is the methodology presented in this work for *application-aware resource management*.

In this section, we summarize related work which is subject to application-aware resource management. The current state of the art consists of numerous approaches and concepts, which use information from different networking layers and of different degrees of complexity in terms of their functionality, the information flow, and the number of involved entities.

Interfaces and APIs for Application-Aware Resource Management The currently prevalent idea in resource management for improving the quality for end-users is to differentiate traffic flows into quality of service levels. In [26], an application network interface is proposed for OpenFlow-based networks to specify different service levels in the network. The interface allows application service providers to define application specific requirements that are implemented by the network. To achieve a network-wide enforcement, Application Label Switching (APLS) is introduced that adds additional protocol headers (*labels*) to IP flows that can be utilized by OpenFlow SDN switches in order to differentiate traffic at bottleneck links. The advantage of this approach is the usage of the standardized OpenFlow protocol that defines existing bandwidth provisioning algorithms such as weighted deficit round-robin or weighted fair queuing for resource management. However, a comprehensive quantification of the benefits with respect to the different algorithms is lacking. Similar to this approach, an application-network interface via an additional protocol layer is proposed in [25]. This Service Access Layer (SAL) facilitates a split between service level control and the application data plane, thus enabling service aware network routing. For the realization the proposed protocol must be implemented on each network device. In [24], the concept for traffic prioritization is presented for small networks. The authors specify a prioritization of applications for active users and a de-prioritization for applications in the background. Finally, there are specific QoS- or QoE-based management frameworks that define network-wide elements to coordinate the traffic forwarding. For residential networks, [20] proposes to setup a QoS system that uses hints from individual hosts to make decisions about traffic prioritization.

Application-Aware Resource Management with Central Entity Besides [20], the authors introduce in [40] a central entity that has knowledge about the network and application situation. It performs a dynamic resource allocation decision that utilizes SDN in order to enforce application demands in the network. An implementation featuring a video on demand application competing with a file download at a bottleneck link shows that monitoring the VoD application's buffer state can help improving the end users' QoE. The work on participatory networking presented in [7], proposes a communication mechanism between applications and the network which is initiated by the applications. Key concepts include conflict resolution between the needs of different applications or users and decomposition of network control, i.e. limiting an application's authority. The authors of [10] introduce an OpenFlow based framework that aims at achieving fairness with respect to QoE

DOI: 10.1002/nem

among all users in an adaptive video streaming context. The expected QoE of a user is estimated on the quality of information during the video streaming. This approach combines a network-aware service with a network based control entity. Based on monitoring results gathered via intercepting Media Presentation Description (MPD) files used in MPEG-DASH, the framework dictates video quality levels for each user in order to maximize the resulting QoE for every participant. Despite the achieved fairness, open issues here are how universal this approach is and how it can be applied to other applications.

A system design featuring functional blocks that represent functions for packet handling in the network is presented in [36, 41]. Feedback loops including a central decision unit allow for combining network aware applications with an application aware network. A central idea consists of translating user demands into application demands which in turn can be translated into network requirements. The system's capabilities are demonstrated and verified for various use cases including live video and video on demand scenarios.

Application-Aware Resource Management for Video Streaming In [16], the authors propose to consider the buffered playtime of a video when assigning network resources. They propose algorithms optimizing the delivered video quality. In [10], the authors propose to maximize the QoE of the end-users running adaptive video streaming applications. In [35], the authors propose to adapt the video rate based on client-side feedback of the buffered video time.

Multi-Application Optimization [15] considers video streaming, VoIP, and data services when optimizing radio resource management for mobile WiMAX systems. However, they consider only the requested bandwidth per application and do not differentiate between different application quality levels. In [29], the authors focus on cross-layer optimization between application layer and medium access control layer. The application parameters are given by a generic mathematical model for the MOS as a metric for the perceived quality. [28] considers video streaming, audio streaming, and data services in a wireless mesh network. The authors' proposed multidimensional optimization tries to minimize the distortion across flows on wireless links under given fairness constraints. In [32], the author propose resource allocation for LTE based on a particle swarm optimization while considering multiple applications.

ALTO: Application Layer Traffic Optimization A protocol realizing the exchange of information between network and application is standardized by the IETF Application Layer Traffic Optimization (ALTO) working group [1]. It aims at providing guidance to content delivery applications such as P2P or CDNs which have to select one or several hosts or endpoints out of a set of possible candidates. Such appropriate candidates can be selected and the performance with respect to user-centric, network-centric and application-centric metrics can be improved. Currently, several extensions of the ALTO protocol including data centers and cloud applications are being discussed [2].

Data Center Applications More work on application-aware resource management has been conducted in the context of data center applications. Research on data center architectures already shows that combined solutions, e.g. for traditionally separated mechanisms such as routing and service migration, may increase data center efficiency. However, data center applications are not end-user applications as in access networks, but run in isolated and manageable server farms. In

DOI: 10.1002/nem

general, the constrained environment of data centers enables the combination of these traditionally separated mechanisms. Data center application-aware solutions are also provided in [11, 19, 39].

In summary, there are many approaches in the literature that propose user- or application-aware resource management. All related works however do not evaluate the impact of their algorithms in a global fashion with different network loads or different application traffic mixes, i.e. a variable proportion of different applications on the network use. The reason for this is simple: modeling of the problem and a theoretical abstraction are necessary, besides the practical use-case driven evaluations, to be able to perform more extensive studies.

3. MODEL

In this section, we begin by describing the system of interest - a bottleneck link shared by multiple users running either browsing or video streaming applications. Following the model, we define a resource allocation objective for the network scenario. The symbols, which are used throughout this work, are summarized in Table I.

Symbol	Definition
$\mathcal{A}$	The set of active application flows
$a \in \{v, w\}$	An active video or web flow
$S \subset \mathcal{A}$	A subset of active flows
S_v, S_w, S_p	The set of active video/web/prioritized flows
C	Link capacity
q	Allocated bandwidth
q_a, q_S, q_p	Bandwidth allocated to flow/set of flows/set of prioritized flows
$\bar{b}$	Average video bit rate
σ_b	Standard deviation of video bit rate
s	Flow size
λ	Arrival rate
μ	Service rate
ρ	Utilization

Table I. Summary of used symbols for resource allocation problem formulation, resource allocation algorithms, and analytic performance evaluation model.

3.1. Network and Applications

A bottleneck link l has a limited capacity C. While sharing the link, users are running either web browsing applications, which we identify via w, or video streaming applications, which we identify via v. Each application requests content of a certain size. The size of each application request is given by s_w and s_v respectively. Furthermore, a video also has a specific video length t_v. All active web flows and all active video flows build the set of active application flows $\mathcal{A} = \{a_1, ..., a_N\}$. Each flow may occupy a certain amount of the link's capacity. The request times of the users for web w and for video streaming v are modeled as Poisson processes, i.e., they arrive with rate λ_w and λ_v.

To assess the performance of the applications in the given network, we evaluate specific key performance indicators for each application [5, 13]. In case of web, the page load time of a web site has the most dominant impact on perceived QoE [5]. Therefore, we consider the download time of a web page as the key performance indicator of web browsing. For video streaming, there are several well-known metrics and evaluation criteria: SSIM, PSNR, number of stallings during

Prepared using nemauth.cls
DOI: 10.1002/nem

playback, length of stalling periods. We investigate HTTP progressive video streaming in this paper. Objective metrics such as SSIM and PSNR consider only the quality of each single video frame. This is valid and useful for live streaming (RTP streaming etc.) without buffering periods where image distortion can occur. For HTTP progressive video streaming the image quality itself is never impaired due to the use of the TCP transmission protocol. If the bandwidth is not sufficient, the video playback simply stops (stalling) and the picture freezes until the buffer is refilled. According to [4, 13], for HTTP progressive video streaming, stalling (i.e., playback interruptions) is the most dominant factor of QoE, clearly exceeding the significance of video resolution as a second impact factor. Thus, we investigate the buffering ratio, i.e., the total stalling time t_s divided by the total video length t_v, as a performance indicator of video flows.

3.2. Resource Allocation

TCP is the dominant transport protocol for both application types. Because of TCP, all active application flows $\mathcal{A} = \{a_1, ..., a_N\}$ may share the link capacity C equally. We assume that the share ratio can additionally be influenced by a network algorithm, i.e., a resource allocation policy, for each flow $a_i \in \mathcal{A}$ or for each disjoint set of flows $S_j \subset \mathcal{A}$. This means, there is the possibility of assigning each flow a_i a bandwidth q_{a_i} with $\sum_{i=1}^{N} q_{a_i} \leq C$. Similarly, each set of flows S_j can be assigned a bandwidth q_{S_j}, such that $\sum_{S_j \subset \mathcal{A}} q_{S_j} \leq C$. The bandwidth of a set of flows can be again shared equally such that each flow $a_j \in S_j$ has bandwidth $q_{a_j} = \frac{q_{S_j}}{|S_j|}$.

Different application characteristics, usage behaviors, and QoE factors have to be considered when assigning bandwidths to the active flows. Thus, the problem on which we focus can be formulated as follows: **Find a mapping $\mathcal{A} \to \mathbb{R}, a_i \mapsto q_{a_i}$, such that $\sum_{i=1}^{N} q_{a_i} \leq C$ holds, and the average QoE of all active flows $a_i,\ i \in \{1, 2, \ldots, N\}$ is maximized and fair in terms of the objective given by the application type.**

As described in Section 3.1, the key performance indicators of each application will be evaluated that have a high correlation to the QoE. This means, a well performing resource allocation algorithm is supposed to achieve short download times for web flows and low buffering ratios for video streams.

4. RESOURCE ALLOCATION ALGORITHMS

This section introduces four algorithms aiming at an improved resource allocation in terms of perceived QoE of the end user. In the following, we take only two applications, i.e., video streaming and web browsing, into account.

4.1. Fixed bandwidth allocation for all video flows (FBV)

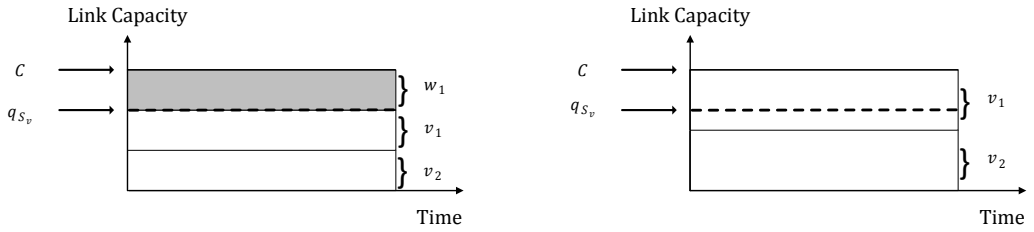
The FBV algorithm reserves a fixed bandwidth for video traffic as it is the more demanding application. The reserved capacity q_{S_v} is a parameter which can be adjusted to reflect the actual traffic share of video traffic.

Algorithm: All currently active flows are split into two sets, one set S_v containing all video flows and one set S_w containing all web browsing flows. S_v is allocated a certain bandwidth q_{S_v} which

Prepared using nemauth.cls
DOI: 10.1002/nem

is shared equally among all active video flows. This means, each video flow gets a bandwidth of $q_v = \frac{q_{S_v}}{|S_v|}$. Consequently, S_w is assigned $q_{S_w} = C - q_{S_v}$ which is also shared equally, such that each web browsing flow receives $q_w = \frac{q_{S_w}}{|S_w|}$. In order to avoid starvation of web browsing, q_{S_v} has to be smaller than C. If one of the two sets S_v or S_w is empty, the other set can share the whole link capacity C among its flows. This means, $q_v = \frac{C}{|S_v|}$ if S_w is empty, and $q_w = \frac{C}{|S_w|}$ if S_v is empty.

Figure 1(a) illustrates the resource allocation of FBV for two video flows (v_1 and v_2) and one web flow w_1. The video flows are sharing q_{S_v} equally and the web flow receives the remaining capacity $C - q_{S_v}$. In case one set S_v or S_w is empty, the applications share C equally, as illustrated in Figure 1(b).



(a) FBV: Video flows share reserved data rate

(b) FBV: Video flows share link capacity

Figure 1. Behavior of FBV for two different cases in terms of number of application flows and types of application flows.

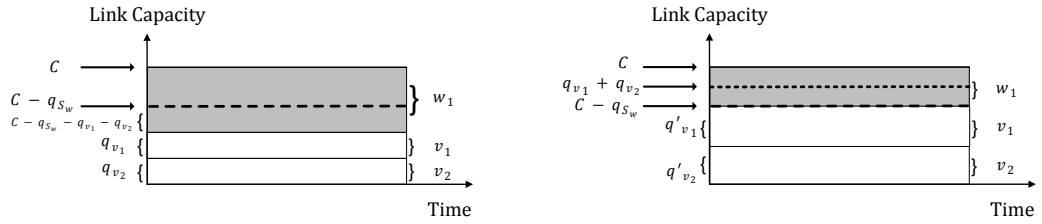
4.2. Fixed bandwidth allocation for each video flow (FBF)

A fixed bandwidth can not only be reserved for the entire set of video flows (FBV), but also for each individual flow. Thus, specific characteristics of each flow can be taken into account. In the performance evaluation in Section 6, the reserved capacity q_{v_i} for a video flow a_i will depend on video attributes such as its video bit rate b_i. For example, for videos having a constant bit rate (CBR), the average video bit rate ($q_{v_i} = \bar{b_i}$) is allocated. In case of videos which show a variable bit rate (VBR) behavior, the average video bit rate plus its standard deviation ($q_{v_i} = \bar{b_i} + \sigma_{b_i}$) is assigned to additionally account for bit rate variations of VBR content. However, different characteristics and bandwidth allocations could also be used in this algorithm. Technically, the FBF algorithm also has to prevent starvation of web browsing by reserving a minimum bandwidth q_{S_w} (parameter) and adjust the allocated bandwidths in case more bandwidth than available is requested.

Algorithm: All currently active flows are split into two sets, one set S_v containing all video flows and one set S_w containing all web browsing flows. Each active video flow $a_i \in S_v$ can request a certain bandwidth q_{v_i}. If the remaining available capacity $q_r = C - q_{S_w} - \sum_{a_j \in S_v} q_{v_j} \geq 0$ (i.e., all requests can be fulfilled), the remaining capacity q_r is added to q_{S_w}. Each video flow a_i is assigned its requested bandwidth q_{v_i} and all web browsing flows equally share $q_{S_w} + q_r$. Thus, each web browsing flow is assigned $q_w = \frac{q_{S_w} + q_r}{|S_w|}$. In case $q_r < 0$, more bandwidth is requested than available. Therefore, all web flows equally share only their guaranteed minimum bandwidth ($q_w = \frac{q_{S_w}}{|S_w|}$) and the capacity $C - q_{S_w}$ is assigned to the video flows relatively to the requested bandwidths. Thus, each video flow a_i is allocated a bandwidth of $q'_{v_i} = (C - q_{S_w}) \cdot \frac{q_{v_i}}{\sum_{a_j \in S_v} q_{v_j}}$.

DOI: 10.1002/nem

Figure 2(a) shows the bandwidth allocation in case that the requested video rates $q_{v_1} + q_{v_2} \leq q_{S_v}$. The video flows use their requested data rate q_{v_1}, q_{v_2} and the web flow w_1 uses the remaining capacity q_r. Figure 2(b) illustrates the case in which $q_{v_1} + q_{v_2} > q_{S_v}$. The video flows share the allocated video capacity q_{S_v} according to their requested video rates.



(a) FBF: Requested video rate does not exceed overall reserved video rate

(b) FBF: Requested video rate exceeds overall requested video rate

Figure 2. Behavior of FBF for two different cases in terms of varying requested data rate by the video flows.

4.3. Weighted bandwidth allocation for all video flows (WBV)

Another approach utilizes the service ratio between the two classes. The idea is that videos, in general, require a multiple of the bandwidth needed for web browsing. Thus, there is a weight parameter $w_V > 1$, such that each video flow is assigned a w_V times larger bandwidth than a web browsing flow. In contrast to FBV, there is no fixed separation between the resources of the two classes. Instead, the resource distribution between the web class and the video class scales according to the number of active flows.

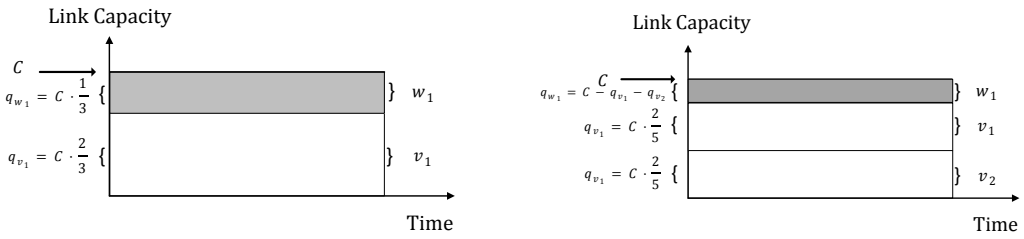
Algorithm: All active flows are split into two sets, one set S_v containing all video flows and one set S_w containing all web browsing flows. Each active flow contributes to the calculation either weighted (video) or normal (web). Thus, a video flow is assigned a bandwidth of $q_v = C \cdot \frac{w_V}{w_V \cdot |S_v| + |S_w|}$, whereas a web browsing flow will be allocated $q_w = C \cdot \frac{1}{w_V \cdot |S_v| + |S_w|}$. This in turn results in a total link share with $q_{S_v} = |S_v| \cdot q_v$ and $q_{S_w} = |S_w| \cdot q_w$ as $C = q_{S_v} + q_{S_w}$.

Figure 3(a) shows the behavior of the WBV algorithm with weight $w_V = 2$. As there are two flows sharing the link, the sharing ratio between v_1 and w_1 is $2:1$. Figure 3(b) illustrates the case in which an additional video flow is on the link. Here, each video flow gets $\frac{2}{5}$ of the link capacity C, whereas w_1 gets $\frac{1}{5}$ of the link capacity C.

4.4. Dynamic bandwidth allocation for each video flow (DBF)

In contrast to the above described resource allocations, which are based on flow types or flow characteristics, also dynamic bandwidth allocation based on current application information is possible. In particular, the buffer fill of video flows is used in this work. Thereby, the short term flexibility introduced by the buffer can be utilized and struggling flows can be helped to avoid stalling, which is the worst QoE degradation for video flows. The strategy relies on an equal sharing of the capacity among all flows. Video flows that have a low buffer level (below a threshold t_l) will be added to a special set S_p of flows, which can utilize a reserved bandwidth fraction q_p. q_p is a parameter, which has to be configured by the network operator in order to adjust the

Prepared using nemauth.cls
DOI: 10.1002/nem



(a) WBV: Link share in case of two flows (one web, one video)

(b) WBV: Link share in case of three flows (one web, two videos)

Figure 3. Behavior of WBV for two different cases in terms of number of video flows. Weight $w_V = 2$ for both figures.

extent of prioritization. This prioritization shall help to fill the buffer. If the buffer usage exceeds a threshold t_h, the video flow is removed from S_p. Note that multiple video flows may be prioritized simultaneously. to avoid stalling, which is the worst QoE degradation for video flows. The strategy relies on an equal sharing of the capacity among all flows. Video flows that have a low buffer level (below a threshold t_l) will be added to a special set S_p of flows, which can utilize a reserved bandwidth fraction q_p. q_p is a parameter, which has to be configured by the network operator in order to adjust the extent of prioritization. This prioritization shall help to fill the buffer. If the buffer usage exceeds a threshold t_h, the video flow is removed from S_p. Note that multiple video flows may be prioritized simultaneously.

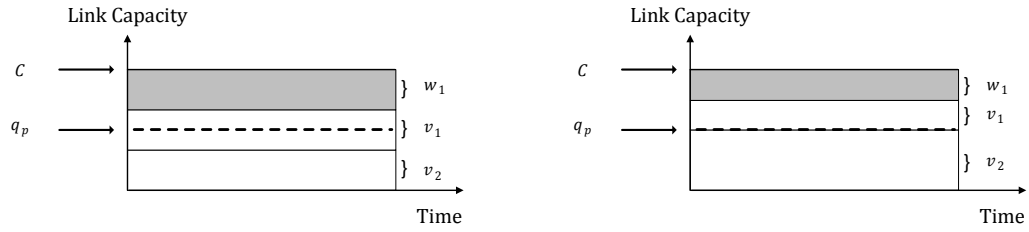
Algorithm: All active flows are put into a set S_u in the beginning. Additionally, an empty set S_p is defined for prioritized flows and a bandwidth q_p is reserved for that set. For each video flow the playout buffer is monitored. If the playback buffer falls below the lower threshold t_l, the flow is moved from S_u to S_p. If the playback buffer of a flow in S_p is above the high fill threshold t_h, it is moved back to S_u. All prioritized video flows share q_p equally, i.e., $q_{v_p} = \frac{q_p}{|S_p|}$. All other flows (i.e., web browsing flows and non-prioritized video flows) equally share the remaining bandwidth. This means, the bandwidth allocated to non-prioritized video flows is $q_{v_u} = \frac{C-q_p}{|S_u|}$, or $q_{v_u} = \frac{C}{|S_u|}$ if S_p is empty. Web browsing flows are never prioritized and thus always receive $q_w = \frac{C-q_p}{|S_u|}$, or $q_w = \frac{C}{|S_u|}$ if S_p is empty.

Figure 4(a) illustrates the case in which the playback buffer of all videos is greater then t_l. All flows share the available capacity C equally. Figure 4(b) shows the case in which the video flow v_2 is added to the set S_p. As in this case only the buffer of v_2 is below the threshold t_l, v_2 receives all the allocated capacity q_p. All other flows, i.e., w_1 and v_1, share the remaining capacity $C - q_p$ equally.

5. EVALUATION METHODOLOGIES AND SCENARIO

In Section 5.1, we introduce the analytic approach that can be used to evaluate the performance of FBV, FBF, and WBV. In order to analyze DBF, we provide a simulation that is described in Section 5.2. In Section 5.3, the investigated scenario is outlined; details about the video encoding, average bit rates etc. are given.

DOI: 10.1002/nem



(a) DBF: All video flows have a buffer level higher than t_l (b) DBF: One video flow gains a higher priority

Figure 4. Behavior of DBF for two different cases in terms of current state of the video applications.

5.1. Analytic Approach

We model the access link as a birth-death process with a single server and model its utilization variations as a stationary process of singular and independent arrivals of traffic. This means, flows enter the system, they are served with a certain download bandwidth, and leave the system after the download is finished. For mathematical tractability, we consider Poisson arrivals with rate λ, and a negative exponentially distributed job size (i.e., video or web page size) with mean s, which gives the service rate $\mu = \frac{C}{s}$. Thus, the model for normal TCP behavior without any resource allocation policy, which shares the capacity equally among all flows, is described by a M/M/1 queue with processor sharing policy (M/M/1-PS). To account for different application usage, we use different arrival rates λ_v, λ_w and mean job sizes s_v, s_w for video streaming and web browsing.

This two class M/M/1-PS model is a special case of the more general discriminatory processor sharing (DPS) model described in [22]. Following the work of [6], it is possible to find the conditional average response times $V(\tau)$ for each class depending on the job size τ. Thus, for any given web page size or video size, the average download time can be given. As $V_v(\tau)$ consists of the video length (depending on video size τ and video bit rate) and the stalling time, the average stalling time should be definable. However, as we are not interested in a single user request, but in the situation in the whole network, we analyze the unconditional average response times V, for which we have:

$$\begin{aligned} V_v &= \frac{1}{\mu_v(1-\rho)} \\ V_w &= \frac{1}{\mu_w(1-\rho)} \end{aligned} \tag{1}$$

where $\rho = \rho_v + \rho_w = \frac{\lambda_v}{\mu_v} + \frac{\lambda_w}{\mu_w}$. Note that V_v does not allow for a computation of average stalling times, as video bit rate distributions have to be taken into account. Thus, here (and for all resource allocation strategies) we follow a simulation approach as described below in Section 5.2 to obtain more detailed results.

The WBV strategy corresponds to the above described more general DPS model, in which each class is assigned a weight which controls the division of processor capacity. Thus, using the weight w for allocating w times larger bandwidth to videos than to web pages, we can use the generalized

DOI: 10.1002/nem

results from [6]:

$$V_v = \frac{1}{\mu_v(1-\rho)}(1+\frac{\mu_v\rho_w(1-w)}{w\mu_v(1-\rho_v)+\mu_w(1-\rho_w)})$$
$$V_w = \frac{1}{\mu_w(1-\rho)}(1+\frac{\mu_w\rho_v(w-1)}{w\mu_v(1-\rho_v)+\mu_w(1-\rho_w)}) \quad (2)$$

FBV and FBF are best described by the generalized processor sharing (GPS) model in which each request or class of requests receives an arbitrary service rate. Thus, FBV can be described by a model with two classes, one for video and one for web pages. From the work of [3], stationary joint distributions can be obtained but no closed form for the conditional response times is given. However, numerical results can be obtained by solving the steady state equations of the FBV model. FBF can be modeled by a single class for each video and one class for all web pages, which enlarges and complicates the FBV model such that we did not conduct an analytical performance evaluation. DBF is a dynamic strategy prioritizing flows depending on current playback buffers, so no analytical model can be provided. Therefore, our analysis of FBF and DBF will be based on results of the simulation runs only.

5.2. Simulation

To confirm the analytic findings on the average download times, to obtain more detailed results, and to evaluate the strategies for which no analytic approach is present, we follow a simulative approach based on the analytic model. The streaming of videos and the download of web pages over a single link is simulated with a Java discrete-event simulation. Again, a single link has a fixed capacity C, which is shared among all application flows of the users in the system. The streaming is implemented such that different flow control (i.e., bandwidth allocation) strategies can be applied. Each run simulates the link utilization slightly more than one day. The arrivals of both video and web page requests follow Poisson processes with rate λ_v and λ_w respectively. The size of videos s_v and web pages s_w is exponentially distributed. With video streaming, the stalling of each video depends on the download bandwidth, video bit rate, and buffering strategy. The average video bit rate $\bar{b}$ is exponentially distributed. For the simulation of variable bit rate videos, the video is split in chunks of 2 s playtime. The bit rate of each chunk is different, but constant bit rate was assumed within a chunk. For the varying chunk bit rate, we use a simple auto-regressive model, which can accurately model videos without scene changes [33]. Note that, in practice, higher variations may occur if the video content is composed of several scenes, which can negatively effect the presented results. For each video, a playout buffer is maintained and a simple buffering strategy is adopted: stalling of a video begins when the playout buffer is empty, and playback resumes after $5s$ of playtime are in the buffer. Thus, not only the resulting video and web page download times, but also stalling times can be obtained. In the performance evaluation, the buffering ratio, i.e., the ratio of stalling time and playback time, will be used. An initial delay of $2s$ was chosen as it provides some time to pre-buffer video data without significantly deteriorating the perceived quality [14].

DOI: 10.1002/nem

5.3. Evaluation Scenario

The aim is to investigate a recently new emerging scenario with progressive video streaming. We use the observations of [8] for our evaluations and additionally investigated the sizes of 50 randomly downloaded videos and web pages, respectively. In detail, we focus on 360p Flash videos, which is the current default quality for YouTube videos on smartphones. According to [8], the average encoding bit rate of 360p YouTube videos is $\bar{b} = 0.5$ Mbps. Furthermore, we measured a mean video clip length of 110 s. Therefore, we define $s_v = \bar{b} \cdot$ video clip length $= 6.87$ MB. To align the simulation and the analytical approach, we model the video size as a random variable following an exponential distribution with the reciprocal of s_v as rate parameter. Each video is divided into chunks with random sizes, which again follow an exponential distribution with the reciprocal of $2s \cdot \bar{b}$ as rate parameter. Only the last chunk is cropped to reach the desired video size. The size of a web page is exponentially distributed with a mean of $s_w = 1.3$ MB. For the performance evaluation, we simulate one link with capacities of 5,10, and 20 Mbps and a maximum of 50 flows.

6. PERFORMANCE EVALUATION AND INSIGHTS

This section presents the results of the performance evaluation of all algorithms of this paper. In particular, we start with investigating for which scenarios an active resource scheduling for a link is needed, i.e., for which level of network use a resource scheduling may improve the quality of the applications. Based on this observation, the algorithms are evaluated analytically and with simulation for different traffic mixes, i.e., for different sharing ratios between web and video flows for a particular network use. Finally, we provide an overview of all algorithms and differentiate their trade offs between the quality of web and video traffic. Note that the following plots contain confidence intervals for 95 % confidence level.

6.1. Situation in Current Networks Without Resource Management

To motivate why an application-aware resource management is needed, the performance of all network flows in terms of QoS parameters is presented and then linked to the actually perceived quality of each application. The results allow us stating that pure network parameters may not show any performance issues, whereas looking at the application parameters reveals dramatic performance degradation.

6.1.1. Best Effort Networks As described in Section 3, it is assumed that with TCP all flows share the available link capacity equally. This means, every flow gets the same bandwidth share. Figure 5 shows the normalized download time of 1 MB for different capacities and network loads. Both the results of the analytical model and the simulation of Section 5 are presented. First, it can be seen that the analytical model closely resembles the simulation as the analytical model results fall within the 95 % confidence intervals of the simulation. Second, for a given network load, a higher link capacity leads to lower average download times. Third, confirming the analytical model in Equation 1, when the network load increases towards 1, the time to download data grows hyperbolically. This leads to unacceptable download times that are commonly avoided in most networks by over-provisioning.

Prepared using nemauth.cls
DOI: 10.1002/nem

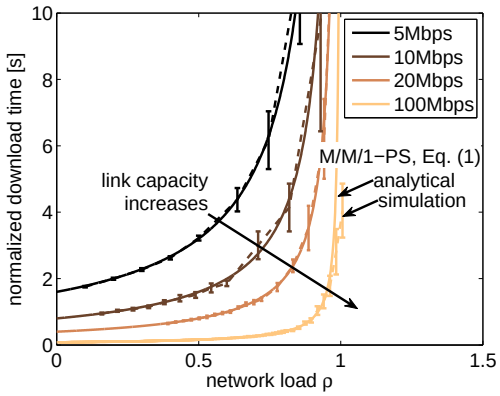


Figure 5. Normalized download time of 1 MB in best effort networks.

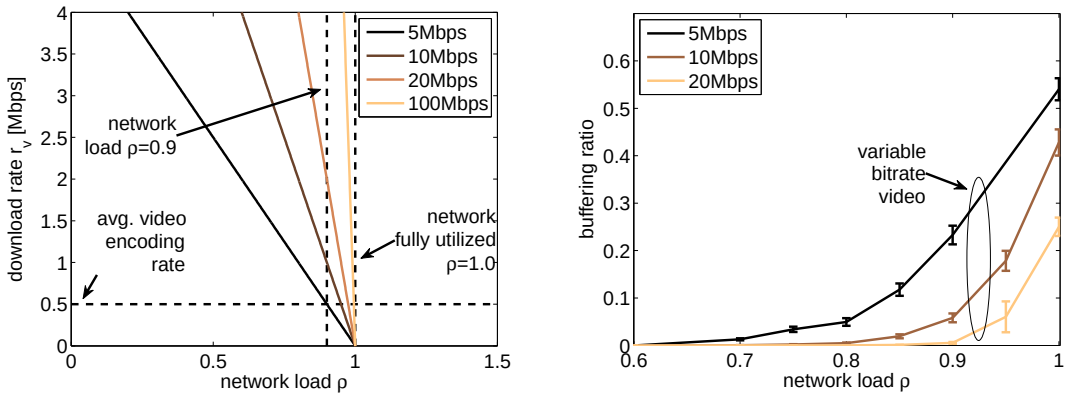
6.1.2. Application Performance Although the general results of Section 6.1.1 hold, different applications have different requirements to the network. Thus, also the changing of network parameters has a different impact on the quality of the various applications. This holds especially for the two considered applications in this work, i.e., video streaming and web browsing. Therefore, we analyze application inherent parameters that have a high correlation with the user-perceived quality of the application as described in Section 3.1. These key performance indicators are download time of a web page for web browsing and buffering ratio for video streaming.

Figure 6(a) compares the download rate r at different link capacities and loads for videos with an average encoding rate of 0.5 Mbps. It shows that the download rates decrease linearly when network load increases. The download rate has to be at least as high as the average video rate to guarantee a video playback without interruptions. It can be seen that a 5 Mbps link can provide such download rates up to a load of 0.9. Higher link capacities can even guarantee smooth playback for higher loads. However, this is only true for CBR videos where the download rate does not fluctuate. Currently deployed streaming applications apply pre-buffering of video content and use video encoding with VBR, which may result in different video performance. For example, with VBR the network requirements of the video can vary over time, but the use of the buffer can overcome certain load peaks. Therefore, this behavior was emulated in the above presented simulation as described in Section 5.2. Figure 6(b) shows the simulation results for the average buffering ratio of VBR videos for different capacities and network loads. It can be seen that, in contrast to CBR, stalling of videos with slightly varying bit rate can occur already with loads around 0.7 and above. Consequently, the benefits of resource allocation strategies in high load scenarios will be investigated in the following.

6.2. Performance Evaluation of Resource Management Algorithms

This sections evaluates the resource management algorithms, which were presented in Section 4. In this section, if not stated otherwise, results for a link capacity of 10 Mbps and a network load of 0.9 are shown. Note that this is the load of a best effort system (M/M/1-PS) with given arrival rates λ_v, λ_w, job sizes s_v, s_w, and capacity C (cf. Section 5) which is also used for the systems that apply the different resource allocation algorithms. Nevertheless, this general approach can also evaluate situations with different network parameters.

Prepared using nemauth.cls
DOI: 10.1002/nem



(a) Network load vs. download rate video.

(b) Buffering ratio.

Figure 6. Video application performance in best effort networks.

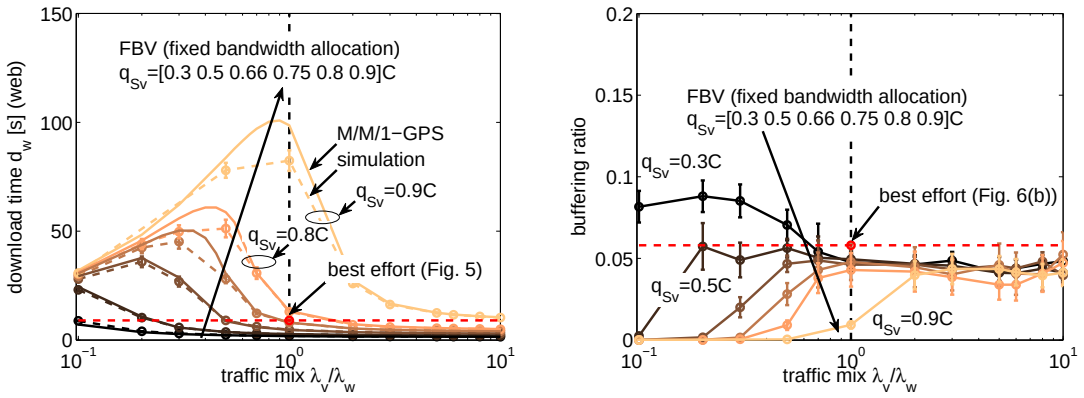
This analysis investigates the key QoE influence factors of the two considered applications, web browsing and video streaming. As the performance of a given resource management algorithm depends not only on application characteristics but also their popularity in the network, the traffic mix ($\frac{\lambda_v}{\lambda_w}$) is introduced. For example, a traffic mix of $\frac{\lambda_v}{\lambda_w} = 2$ means that on average there are two times more video downloads than web page downloads over the link.

6.2.1. FBV The FBV strategy reserves a certain share of the capacity to video flows. The challenge, however, is the selection of the right prioritization settings. If too much bandwidth is allocated for video, the download time of web pages is considerably affected. If the reserved rate is too small, the video may stall and the prioritization is useless. Therefore, several fixed allocations ranging from $q_{S_v} = 0.3C$ up to $q_{S_v} = 0.9C$ are simulated. Figure 7(a) depicts the analytical and simulative results of the average download time of a web page for a fixed network load $\rho = 0.9$ and varying traffic mixes. It can be seen that the download times in a best effort network are 9s on average for each traffic mix. On the left side of the dashed vertical line, more web page downloads are on the link. The plot depicts that an increased bandwidth allocation generally increases the web page download times as videos are in the system, which leave little resources for the web page downloads. When the traffic mix increases towards $10^0 = 1$ and beyond, this effect is mitigated because less web pages are in the system. Thus, they are not penalized to such big extent as the bandwidth share becomes more aligned to the traffic share. In the rightmost part of the figure, only few web flows are in the system. They can then use the remaining (i.e., not reserved for videos) capacity almost exclusively, which stems from but also leads to short download times. For the analytical M/M/1-GPS model, the mean number of users in the system is obtained from the state probabilities. Little's Theorem is used to compute the mean delay in the system, which is equal to the mean service time in a processor sharing system. As can be seen in the plot, the simulation runs are approximated very well by the analytical model. Thus, also the analytical model can be used for the performance evaluation.

Figure 7(b) shows the exactly same situation from the perspective of video flows and depicts the average buffering ratio for different traffic mixes and different bandwidth allocations. Again the best effort performance (cf. Figure 6(b)) is shown as a base value. In the rightmost part of the

Prepared using nemauth.cls
DOI: 10.1002/nem

figure, it can be seen that the buffering ratio converges towards the best effort performance for any prioritization. This can be explained by the fact that web flows are seldom in the system and for short times only. This means that there are many videos which compete most of the time only among themselves, which leads to a best effort behavior. In the left part of the plot, video flows are more seldom in the system but have no problems because of the bandwidth allocation. In general, a higher bandwidth allocation leads to smaller buffering ratio with traffic mixes below 1. Only in the case of $q_{S_v} = 0.3C$, the allocation is not sufficient as the buffering ratio is higher than the best effort base line. This again shows the need for careful selection of the right allocation parameter. The trade-off between the optimization of the two applications will be further discussed in Section 6.3.



(a) Download time of web pages.

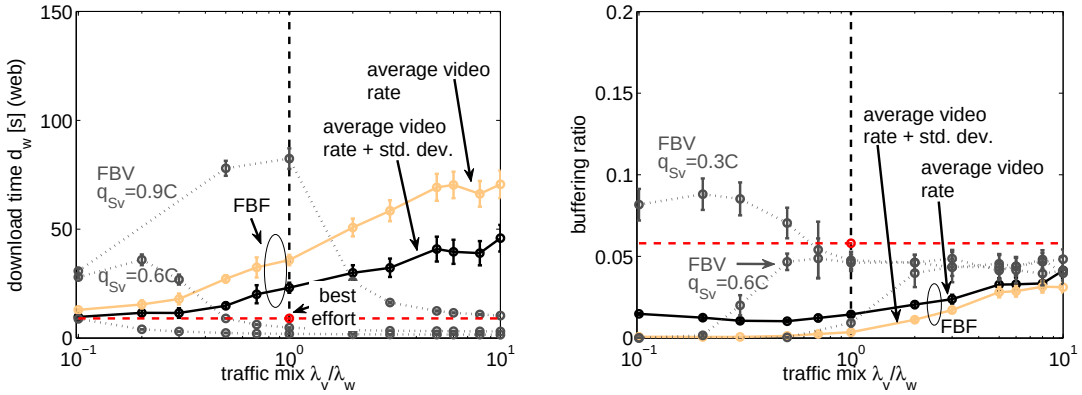
(b) Buffering ratio of video flows.

Figure 7. FBV: Fixed bandwidth reservation q_{S_v} for all video traffic at a load $\rho = 0.9$.

6.2.2. FBF The difference of FBF compared to FBV is that it allocates bandwidth to individual video flows. As described in Section 4, two allocations are investigated. First, each video is assigned its average bit rate, and second, each video is assigned its average bit rate plus standard deviation to account for bit rate variations (VBR content). Figure 8(a) shows the average download times of web pages for both allocations. For comparison selected FBV performances ($q_{S_v} = [0.3\ 0.6\ 0.9]C$) are depicted in gray. Obviously, it can be observed that the FBF download times increase if the traffic mix increases. This is due to the fact that more and more video flows are in the system which request bandwidth, hence, the reserved bandwidth for video, which is unavailable for web page downloads, adds up.

Figure 8(b) shows the buffering ratio of the video flows in the same scenario. It can be seen that the average video bit rate allocation is not sufficient as it cannot avoid stalling of VBR videos. However, the allocation of average video rate plus standard deviation performs quite well for traffic mixes below 1 where it almost completely avoids stalling. When more video flows are in the system, the reservation of the videos exceeds the available capacity. Thus, each video receives a share relative to its request which leads to increasing buffering ratios and consequently bad performances of the video streams. With increasing traffic mix, it performs better than all FBV algorithms, which use an equal share policy if too many videos in the system, as the relative sharing of bandwidth is better aligned to the demands of the video flows.

Prepared using nemauth.cls
DOI: 10.1002/nem



(a) Download time of web pages.

(b) Buffering ratio of video flows.

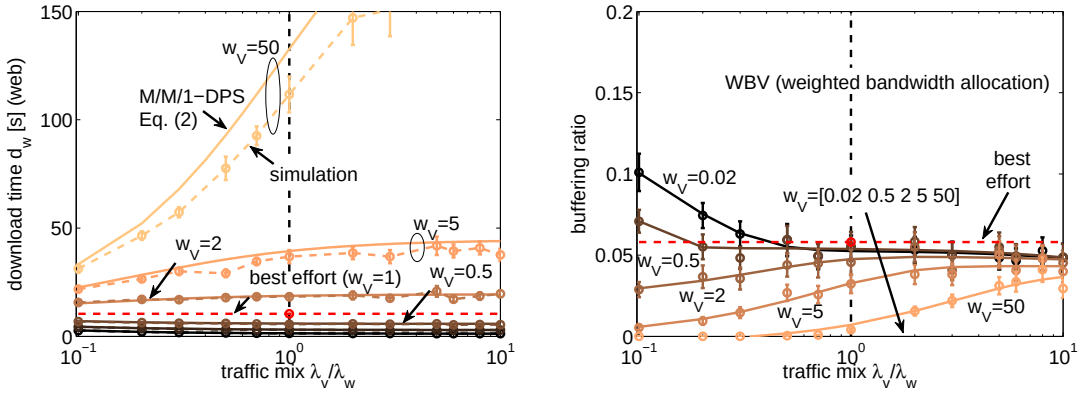
Figure 8. FBF: Fixed bandwidth reservation for each flow with either average bit rate or average bit rate plus standard deviation at link load $\rho = 0.9$.

6.2.3. WBV The WBV strategy assigns a weight w_V to each video flow such that it receives a w_V times larger bandwidth than a web flow. Figure 9(a) shows the results of the analytic DPS model (cf. Equation 2 in Section 5) and of the simulation runs. It can be seen that the analytic model and the simulation results are well aligned, especially for smaller values of w_V. Moreover, a larger w_V increases the download time, as web flows receive less bandwidth compared to video flows for any traffic mix. Only for large weights and large traffic mixes, the bandwidth given to web flows becomes almost arbitrarily small which leads to a fast increase of download times.

The buffering ratio in the same scenario can be seen in Figure 9(b). The video flows benefit from WBV with $w_V > 1$ in case of traffic mix below 1. If more video flows are in the system, the effect reduces and the buffering ratio again converges towards that of best effort networks (equal sharing). It can be seen that WBV is suitable to align bandwidth allocation to the flow sizes by choosing an appropriate weight w_V for the given traffic mix. As the resulting resource allocation also accounts for the number of flows, it can significantly reduce buffering ratios for video flows while only slightly increasing the download time of web pages. Again trade-offs between the two applications have to be taken into account and are further investigated in Section 6.3.

6.2.4. DBF The results presented so far showed the performance of resource allocation algorithms in which a parameter was set statically for each flow. This means, the bandwidth allocation changed only depending on the current number of flows and the respective parameters. In contrast, DBF is a dynamic resource allocation strategy, which prioritizes a video flow if its buffer fill is below a threshold t_l, and de-prioritizes it once its buffer level has passed a threshold t_u. For the performance evaluation of DBF, the prioritization bandwidth q_p, which can be utilized by suffering video flows, is set to $q_p = 0.9 \cdot C$ and the thresholds are set to $t_l = 10s,\ t_u = 20s$. Figure 10(a) shows the performance of DBF (dashed) and compares it to WBV (solid), which showed a decent performance with respect to both traffic classes. It can be seen that the download time of web pages is highest for an equal traffic mix. The more the traffic mix becomes imbalanced (either more videos or more web pages are in the system), the download time improves and comes close to the best effort situation.

Prepared using nemauth.cls
DOI: 10.1002/nem

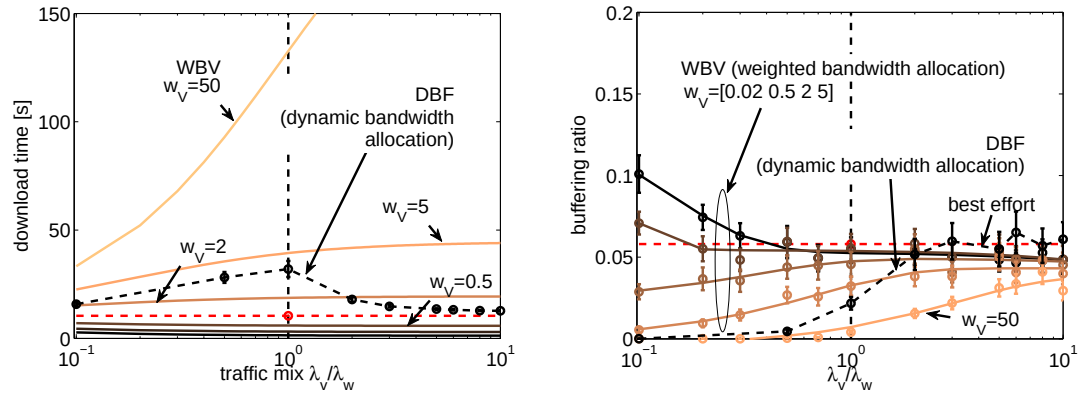


(a) Download time of web pages. (b) Buffering ratio of video flows (exponential fits).

Figure 9. WBV: Weighted bandwidth reservation for all video traffic with weight w_V at link load $\rho = 0.9$.

The video flows, on the other hand, face a significant performance improvement, which is depicted in Figure 10(b). If there are many video flows in the system, the buffering ratio is worst but still converges to the best effort situation. However, if there are less videos in the system, the buffering ratio is much smaller than without resource allocation strategy. It can be seen that DBF supports video flows even better than WBV with a high $w_V > 5$, but at the same time does not penalize web flows too much.

To sum up, DBF has been shown to effectively reduce stalling for traffic mixes $\frac{\lambda_v}{\lambda_w} < 2$. But although DBF is a video prioritization algorithm, it only slightly increases the download time of web pages. Thus, DBF ranges among the best investigated algorithms especially taking balanced and fair behavior into account. However, it has to be noted that the influence of the parameters $q_p, t_l,\ t_u$ was not investigated in this work and is thus still open. Moreover, the gain of dynamic allocation based on application state comes with the cost for monitoring and signaling application information to the resource management entity. The trade-offs for this additional cost will also be investigated in future work.



(a) Download time of web pages. (b) Buffering ratio of video flows.

Figure 10. DBF: Dynamic bandwidth allocation for each video flow based on buffer fill at link load $\rho = 0.9$.

DOI: 10.1002/nem

6.3. *Comparison of the Different Strategies*

After presenting the performance of each individual algorithm, we conduct a Pareto analysis to compare how the different strategies influence the QoE in our scenario. Therefore, the two QoE performance indicators, i.e., download time for web flows and buffering ratio for video streams, are used as axes for the Pareto plot in Figure 11. The best effort performance (mean download time 9.04 s, mean buffering ratio 0.058) is depicted by a red cross and serves as a reference. Similarly, the Pareto-optimal performances of the algorithms for a given traffic mix are marked in the plot. We distinguish five different traffic mixes by colored lines ranging from 0.1 (black), i.e., ten times more web flows than video flows, over 1 (light brown), i.e., the same amount of web and video flows, to 5 (yellow), i.e., five times more video flows than web flows. Note that only the marked points are Pareto-optimal algorithm performances, the lines connecting the points, which belong to the same traffic mix, however, are only for visualization purposes and do not indicate the location of other Pareto optima. The marker shape of the Pareto-optimal points represents the type of algorithm, which achieved the respective results.

The Pareto-optimal performances, i.e., performances, in which it is impossible to improve in one dimension without deteriorating the other dimension, are either below or left of the best effort performance. This means, the usage of any of the resource allocation algorithms does improve the performance at least in one dimension. It can be seen that especially WBV algorithms can be used to optimize the web page download time, whereas FBF and DBF achieve minimal buffering ratios. The most interesting part is the bottom left gray shaded box, which indicates an improvement for both web and video flows. The marker shapes indicate that WBV and FBV algorithms can achieve such performances when they use the right parameter for the given traffic mix. Note that the specific algorithm parameter for a Pareto optimum is not given in this plot but can be obtained from the respective plots in the previous sections.

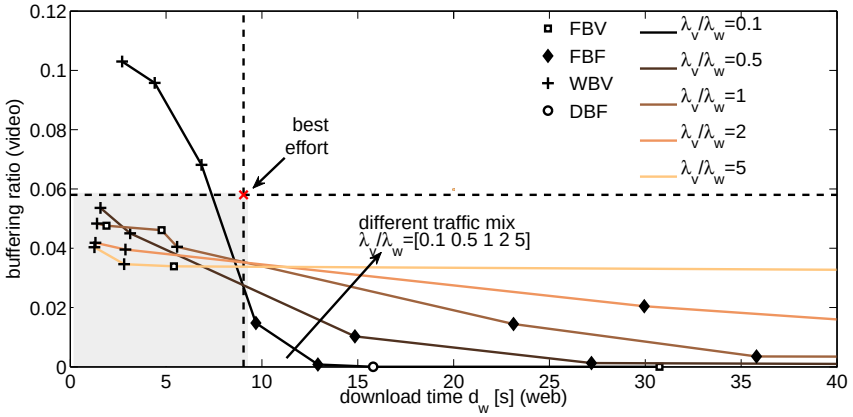


Figure 11. Overview of Pareto-optimal performances for different traffic mixes.

7. CONCLUSION

This work presented new methods that allow analyzing application-aware resource management analytically and with simulation. Application-aware resource management aims at improving not only network's resource efficiency but also users' satisfaction, i.e., the application Quality of

Prepared using nemauth.cls
DOI: 10.1002/nem

Experience (QoE) of users. To count for varying traffic situations, a joint optimization of two application types, namely web browsing and video streaming, is considered, which has not yet been investigated by the research community. With the proposed methods, network operators can quantify performance gain trade-offs between investing in pure capacity increase (over-provisioning) and the realization of application-aware resource management approaches.

This work proposed a model for the application-aware resource management problem and discussed four algorithms for resource management. The algorithms vary in complexity and implementation levels in order to cover the design space of application-aware resource management. In order to evaluate the performance of the algorithms, analytical models based on processor sharing queues (M/M/1-PS) were presented if possible. Moreover, a discrete event simulation was implemented in Java to assess all algorithms.

Furthermore, a first study of application-aware resource management approaches of this paper provides configuration guidelines for the selection and configuration of the considered algorithms. The results of the study show that it is possible to improve the quality of all applications compared to best-effort transmission. This means that for specific traffic mixes it is possible to get a significant gain in network resource use compared to simple best-effort transmission, while still having a low implementation complexity. In case of more complex, i.e., more dynamic algorithms, the application quality of videos were significantly improved. Based on all analysis, operators have the possibility to set up a resource management according to their objectives and traffic situations of their networks. Such an analysis in terms of quality of experience and varying traffic conditions has not yet been considered in the literature.

REFERENCES

1. IETF Working Group on Application-Layer Traffic Optimization (ALTO).
2. R. Alimi, Y. Yang, and R. Penno. Application-Layer Traffic Optimization (ALTO) Protocol, September 2014.
3. J. W. Cohen. The Multiple Phase Service Network with Generalized Processor Sharing. *Acta Informatica*, 12(3):245–284, 1979.
4. F. Dobrian, A. Awan, D. Joseph, A. Ganjam, J. Zhan, V. Sekar, I. Stoica, and H. Zhang. Understanding the impact of video quality on user engagement. In *ACM SIGCOMM conference*, pages 362–373. ACM, 2011.
5. S. Egger, P. Reichl, T. Hoßfeld, and R. Schatz. Time is Bandwidth? Narrowing the Gap between Subjective Time Perception and Quality of Experience. In *2012 IEEE International Conference on Communications (ICC 2012)*, Ottawa, Canada, June 2012.
6. G. Fayolle, I. Mitrani, and R. Iasnogorodski. Sharing a Processor Among Many Job Classes. *Journal of the ACM (JACM)*, 27(3):519–532, 1980.
7. A. D. Ferguson, A. Guha, C. Liang, R. Fonseca, and S. Krishnamurthi. Participatory networking. In *ACM SIGCOMM 2013 Conference - SIGCOMM '13*, page 327, New York, New York, USA, 2013. ACM Press.
8. A. Finamore, M. Mellia, Maurizio M. Munafò, R. Torres, and S. G. Rao. YouTube Everywhere: Impact of Device and Infrastructure Synergies on User Experience. In *ACM SIGCOMM 2012 Conference on Internet Measurement Conference - IMC '11*, page 345, New York, New York, USA, 2011. ACM Press.
9. J. L. Garcia-Dorado, A. Finamore, M. Mellia, M. Meo, and M. Munafo. Characterization of ISP Traffic: Trends, User Habits, and Access Technology Impact. *IEEE Transactions on Network and Service Management*, 9(2):142–155, June 2012.
10. P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race. Towards Network-wide QoE fairness Using OpenFlow-assisted Adaptive Video Streaming. In *ACM SIGCOMM 2013 Workshop on Future uman-centric Multimedia Networking - FhMN '13*, page 15, New York, New York, USA, 2013. ACM Press.
11. B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown. Elastictree: Saving energy in data center networks. In *NSDI*, volume 10, pages 249–264, 2010.

Prepared using nemauth.cls
DOI: 10.1002/nem

12. D. Hock, F. Wamser, M. Seufert, R. Pries, and P. Tran-Gia. OC^{2}E^{2}AN: Optimized Control Center for Experience Enhancements in Access Networks. *PIK - Praxis der Informationsverarbeitung und Kommunikation*, 36, 2013.
13. T. Hoßfeld, R. Schatz, M. Seufert, M. Hirth, T. Zinner, and P. Tran-Gia. Quantification of YouTube QoE via Crowdsourcing. In *IEEE International Workshop on Multimedia Quality of Experience - Modeling, Evaluation, and Directions (MQoE 2011)*, Dana Point, CA, USA, December 2011.
14. Tobias Hoßfeld, Sebastian Egger, Raimund Schatz, Markus Fiedler, Kathrin Masuch, and Charlott Lorentzen. Initial Delay vs. Interruptions: Between the Devil and the Deep Blue Sea. In *QoMEX 2012*, Yarra Valley, Australia, July 2012.
15. C. Huang, H.H. Juan, M.S. Lin, and C.J. Chang. Radio Resource Management of Heterogeneous Services in Mobile WiMAX Systems [Radio Resource Management and Protocol Engineering for IEEE 802.16]. *IEEE Wireless Communications*, 14(1):20–26, 2007.
16. T. Huang, R. Johari, and N. McKeown. Downton Abbey Without the Hiccups. In *ACM SIGCOMM 2013 Workshop on Future human-centric Multimedia Networking - FhMN '13*, page 9, New York, New York, USA, 2013. ACM Press.
17. S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, et al. B4: Experience with a globally-deployed software defined wan. In *ACM SIGCOMM 2013 Conference - SIGCOMM '13*, pages 3–14. ACM, 2013.
18. M. Jarschel, F. Wamser, T. Höhn, T. Zinner, and P. Tran-Gia. SDN-based Application-Aware Networking on the Example of YouTube Video Streaming. In *2nd European Workshop on Software Defined Networks (EWSDN 2013)*, Berlin, Germany, October 2013.
19. J. W. Jiang, T. Lan, S. Ha, M. Chen, and M. Chiang. Joint vm placement and routing for data center traffic engineering. In *IEEE INFOCOM 2012*, pages 2876–2880. IEEE, 2012.
20. M.J. Katchabaw, H.L. Lutfiyya, and M.A. Bauer. Usage Based Service Differentiation for End-to-end Quality of Service Management. *Computer Communications*, 28(18):2146–2159, 2005.
21. S. Khan, Y. Peng, E. Steinbach, M. Sgroi, and W. Kellerer. Application-Driven Cross-Layer Optimization for Video Streaming over Wireless Networks. *IEEE Communications Magazine*, 44(1):122–130, January 2006.
22. L. Kleinrock. Time-shared Systems: A Theoretical Treatment. *Journal of the ACM (JACM)*, 14(2):242–261, 1967.
23. G. Maier, A. Feldmann, V. Paxson, and M. Allman. On Dominant Characteristics of Residential Broadband Internet Traffic. In *ACM SIGCOMM 2009 Conference on Internet Measurement - IMC '09*, page 90, New York, New York, USA, 2009. ACM Press.
24. J. Martin and N. Feamster. User-driven Dynamic Traffic Prioritization for Home Networks. In *ACM SIGCOMM 2012 Workshop on Measurements Up the Stack*, pages 19–24. ACM, 2012.
25. E. Nordström, D. Shue, P. Gopalan, R. Kiefer, M. Arye, S. Ko, J. Rexford, and M. J. Freedman. Serval: An end-host stack for service-centric networking. In *NSDI*, pages 85–98, 2012.
26. S. Paul and R. Jain. Openadn: Mobile apps on global clouds using openflow and software defined networking. In *Globecom Workshops (GC Wkshps), 2012 IEEE*, pages 719–723. IEEE, 2012.
27. Z. A. Qazi, J. Lee, T. Jin, G. Bellala, M. Arndt, and G. Noubir. Application-awareness in sdn. In *ACM SIGCOMM 2013 conference - SIGCOMM '13*, pages 487–488. ACM, 2013.
28. A.B. Reis, J. Chakareski, A. Kassler, and S. Sargento. Quality of Experience Optimized Scheduling in Multi-service Wireless Mesh Networks. In *IEEE Conference on Image Processing (ICIP)*, pages 3233–3236. IEEE, 2010.
29. A. Saul. Simple Optimization Algorithm for MOS-Based Resource Assignment. In *VTC Spring 2008 - IEEE Vehicular Technology Conference*, pages 1766–1770. IEEE, May 2008.
30. B. Staehle, M. Hirth, R. Pries, F. Wamser, and D. Staehle. YoMo: A YouTube Application Comfort Monitoring Tool. In *New Dimensions in the Assessment and Support of Quality of Experience for Multimedia Applications*, Tampere, Finland, June 2010.
31. L. Superiori, M. Wrulich, P. Svoboda, M. Rupp, J. Fabini, W. Karner, and M. Steinbauer. Content-aware Scheduling for Video Streaming over HSDPA Networks. In *Second International Workshop on Cross Layer Design, 2009. IWCLD'09.*, pages 1–5. IEEE, 2009.
32. P. Tang, P. Wang, N. Wang, and V. Nguyen Ngoc. QoE-Based Resource Allocation Algorithm for Multi-Applications in Downlink LTE Systems. In *Proceedings of the 2014 International Conference on Computer, Communications and Information Technology*, Paris, France, 2014. Atlantis Press.
33. S. Tanwir and H. Perros. A Survey of VBR Video Traffic Models. *IEEE Communications Surveys & Tutorials*, 15(4):1778–1802, January 2013.
34. S. Thakolsri, S. Khan, E. Steinbach, and W. Kellerer. QoE-Driven Cross-Layer Optimization for High Speed Downlink Packet Access. *Journal of Communications, Special Issue on Multimedia Communications, Networking and Applications*, Vol. 4, No. 9, pp. 669-680, 2009.

DOI: 10.1002/nem

35. G. Tian and Y. Liu. Towards Agile and Smooth Video Adaptation in Dynamic HTTP Streaming. In *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies - CoNEXT '12*, page 109, New York, New York, USA, 2012. ACM Press.
36. F. Wamser, D. Hock, M. Seufert, B. Staehle, R. Pries, and P. Tran-Gia. Using Buffered Playtime for QoE-Oriented Resource Management of YouTube Video Streaming. *Transactions on Emerging Telecommunications Technologies*, 24, 2013.
37. F. Wamser, T. Zinner, L. Iffländer, and P. Tran-Gia. Demonstrating the Prospects of Dynamic Application-Aware Networking in a Home Environment. In *ACM SIGCOMM 2014, Demonstrations*, Chicago, IL, USA, August 2014.
38. F. Wamser, T. Zinner, J. Zhu, and P. Tran-Gia. Dynamic Bandwidth Allocation for Multiple Network Connections: Improving User QoE and Network Usage of YouTube in Mobile Broadband. In *ACM SIGCOMM Capacity Sharing Workshop (CSWS 2014)*, Chicago, IL, USA, August 2014.
39. D. Xie, N. Ding, Y. C. Hu, and R. Kompella. The only constant is change: incorporating time-varying network reservations in data centers. *ACM SIGCOMM Computer Communication Review*, 42(4):199–210, 2012.
40. 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.
41. T. Zinner, T. Hoßfeld, M. Fiedler, F. Liers, T. Volkert, R. Khondoker, and R. Schatz. Requirement driven prospects for realizing user-centric network orchestration. *Multimedia Tools and Applications*, pages 1–25, 2014.

DOI: 10.1002/nem