Investigation of Different Approaches for QoE-Oriented Scheduling in OFDMA Networks*

Florian Wamser, Sebastian Deschner, Thomas Zinner, and Phuoc Tran-Gia

University of Würzburg, Am Hubland, Germany
{wamser,deschner,zinner,trangia}@informatik.uni-wuerzburg.de

Abstract. QoE- and application-aware scheduling is a new paradigm for mobile communication networks. It aims at making better use of network resources with respect to the perceived quality of the users. To achieve this, it specifies an interaction between application and networking layers. Previous work has shown that such a resource management is possible by the weighting of applications and the definition of key quality indicators. However, quantification of the benefits and the impact on the application itself is hardly studied, since it requires precise modeling of both the data transmission in the mobile network as well as the application itself. In this paper the influence of different cross-layer scheduling heuristics on the application is examined for the air interface of LTE mobile networks. For this, not only the physical data transmission but also the application behavior is simulated in detail for Skype, YouTube, web browsing, and downloads. For each application quality indicators are defined that provide information on the current performance of the application. The investigated scheduling approaches take into account detailed application information of different levels like the application type, the current status of the application, or the ability of an application to adapt to the network situation.

1 Introduction

Nowadays, high data rates in mobile networks are possible. They allow high quality streaming or similar services that were usually reserved only for fixed network users. The requirements for these applications are diverse and typically vary over time due to video encoding, download patterns, or user behavior. Today's networks support them based on Quality of Service (QoS) parameters. However, compliance with the QoS policies on the network level does not necessarily guarantee a good quality of applications. From the user's perspective, a good application quality is defined by parameters related to the application such as video quality, waiting times or responsiveness of the application. A major challenge for network operators is therefore the consideration of these requirements within the network to meet the user expectations.

* This work has been funded by the German Research Foundation under grant TR 257/28-2 (FunkOFDMA).

D. Pesch et al. (Eds.): MONAMI 2013, LNICST 125, pp. 172–187, 2013.

Quality of Experience (QoE) and application-aware scheduling is a new paradigm for mobile communication networks based on the interaction between application and network layer. It enhances traditional scheduling approaches that rely on objective measures such as packet loss rate, channel quality, transfer delay or delay variation, by additionally taking application information into account. This requires an identification of the key influence parameters on the application quality as well as means to monitor these parameters. Depending on the type of application, these parameters may be measured at the end-device or in the network using techniques like deep packet inspection. Today's packet schedulers in cellular networks are based on hard QoS parameters, i.e., they either can provide the requested QoS or not. This differs for QoE-based schedulers which may provide several thresholds for a certain, as well as, for a varying application quality. Accordingly, this flexibility can be used to exploit the temporal variability of the transmission channel to better support applications or compensate local load peaks by using fewer resources.

Previous work has shown that QoE scheduling for applications is possible through the utilization of certain information and the definition of key quality indicators. Kahn et al. [1] define cross-layer scheduling with a so-called utility function. The aim is to maximize the utility in order to optimize the QoE. In [2], QoE influence factors are determined and measured for web traffic. According to them, a QoE-oriented scheduling is defined. Wamser et al. [3] utilize the buffered YouTube playtime for QoE-aware scheduling in OFDMA networks. All solutions consider QoE or application information. Nevertheless, quantification of the benefits and the impact on the application itself are hardly studied, since it requires precise modeling of both the data transmission in the mobile network as well as the application itself.

In this paper, a simulative investigation is performed on how application information of varying degrees of detail may provide a more effective packet scheduling in the air interface of LTE mobile networks. All simulation runs are performed within an LTE system level simulator which involves a detailed application-layer model of YouTube, Skype, and TCP, as well as the LTE protocol stack and wireless channel models. The scheduling approaches that are proposed and investigated are of increasing complexity. First, fixed service flows for applications are arranged with respect to the recognized application type. Then, approaches are proposed that require more information about the application such as application status. Finally, a direct signaling of QoE-related application parameters is assumed. Issues such as scalability are not addressed in this paper. Furthermore, only one single cell is simulated.

The remainder of the paper is structured as follows. In Section 2 the technical background for scheduling is summarized as well as an overview of various scheduling approaches is given. Section 3 describes the investigated scheduling algorithms. In Section 4 the application models are introduced that are used for the evaluation of the schedulers. After that follows the description of the simulation in Section 5. In Section 6 a scheduling taking into account the adaptation behavior of Skype is presented. Finally, a comprehensive statistical evaluation of different scheduling approaches is done in Section 7. At the end, conclusions are drawn in Section 8.

2 Background and Related Work

Unlike random access-based networks, the transmission of data packets in mobile networks is controlled by the base station. It performs a scheduling of the pending packets that determines when data is transmitted to a user and when a user is allowed to send data to the base station. The basic idea of the scheduling is to dynamically assign resources for the transmission according to their required data rate and QoS conditions.

Previous approaches often aim at maximizing the potential total throughput in a cell while providing some degree of fairness. This includes common scheduling approaches that take advantage of time-varying channel conditions such as proportional fair scheduling or variants of opportunistic scheduling. Current approaches, however, consider more and more the application or the user as well as his demands on the network.

For example, there is utility-based scheduling. The idea of utility-based scheduling [1,4,5] is based on the fact that different QoS parameters have a different effect on the QoE of an application. Thus, the QoS parameters are weighted and a utility function is defined based on them. The aim is to maximize the utility in order to optimize the QoE. Furthermore, there are cross-layer approaches that consider directly the QoE that the end user experiences. In [6,7], the continuously monitored QoE of voice connections is considered for scheduling and admission control decisions for IEEE 802.16 networks. In [3], the buffered playtime of YouTube video streaming is utilized for a dynamic prioritization of YouTube traffic to increase the QoE. In [2], a mapping for web traffic is presented to allow for a direct consideration of web browsing within the scheduling.

In contrast to specific scheduling approaches, there is also a 3GPP working group to study general mechanisms and approaches to avoid user plane congestion in LTE networks [8]. One use-case of this working group is the specification of congestion-based policy rules for the radio access network for crowded cells.

3 Description of the Studied Scheduling Algorithms

The objective of the paper is to investigate and simulate different scheduling concepts or heuristics and have a look at the benefits and the impact on the applications.

For comparison, we first specify a simple proportional fair scheduler as a reference scheduler. Afterwards, we start with arranging fixed service flows for Skype. This is based on the concept of QoS provisioning for applications in mobile networks where QoS profiles are assigned to LTE bearers or IEEE 802.16 service flows. The idea however is that the classes are selected in such a way that the throughput is restricted for application that are able to adapt to low network resources such as Skype. Hence, there are resources that are available for other applications. Then, we move on to signaling approaches that provide direct feedback on the requirements of the applications to the scheduler.

3.1 Proportional Fair Scheduling

The proportional fair scheduler addresses both fairness and throughput in the network. This is achieved by assigning each user i at transmission frame f a priority M which is based on the present achievable transmission rate r and the previously achieved overall throughput R.

$$M_i^R(f) = \frac{{r_i}^{\alpha}}{{R_i}^{\beta}}. \tag{1}$$

It should be noted that the available total bandwidth B depends on the scheduling since different assignments lead to different user throughputs due to the adaptive modulation and coding of the users.

3.2 Fixed Service Class for Skype

Service classes define QoS parameters such as maximum throughput and minimum latency for all users of the class. The service class schedulers used in this work are specifically designed for Skype video calls to take advantage of the adaptation capabilities of Skype. The idea is to restrict a Skype user to a specified bandwidth in order to gain resources for other users. The constraint is selected in such a way that only a small QoE degradation occurs since Skype adapts to the given conditions. More precisely, the goal is to avoid that Skype changes the resolution but adjusts the frame rate and image quality.

Application-Based Service Class. The application based service class is a service class with a guaranteed bandwidth for all Skype users in the cell. This requires the base station to know whether the packets belong to Skype or not. All packets belonging to Skype users are assigned to this class. It is the simplest approach to set a restriction for Skype. Within the class a proportional fair scheduling is done. The scheduling is depicted in Algorithm 1.

Flow-Based Service Class. Flow-based service classes guarantee QoS parameters for single flows instead of application groups. In this case, we guarantee 30 % of the maximum transmission rate of Skype to each Skype user since this is slightly higher than the throughput at which Skype changes the resolution of the video encoding according to our model.

On Demand Flow-Based Service Class. The on demand flow-based service class is similar to the previous one but instead of throttling the throughput of the Skype users all the time, this is only done if there is another application that requires bandwidth. Hence, Skype is able to get more throughput if the situation allows it. In this paper, the limitation in throughput is only triggered by the YouTube users. If there is a YouTube flow in its initial best effort phase, or with a low buffer level (less than 3 s), the service class is activated and all Skype users get the guaranteed but limited throughput as long as enough resources are available.

Algorithm 1. Application-based Service Class Scheduling

```
$bandwidth\_skype\_class = 0$

while $bandwidth\_skype\_class < max.\ bandwidth\ for\ the\ class$ do
    $packet\ list \leftarrow ProportionalFair(skype\ packets)$
    $bandwidth\_skype\_class\ += packet.size$
end while

while $bandwidth\ available$ do
    $packet\ list \leftarrow ProportionalFair(\neg skype\ packets)$
end while
```

3.3 QoE Scheduling Using Application Parameters

The QoE scheduler is similar to the proportional fairness scheduler, but instead of using a priority proportional to the possible throughput, a throughput inversely proportional to the current estimated QoE is used. Hence, if the application is currently in a very good condition, it is assigned a very low priority. However, if the application is in a poor condition, it gets a higher priority. While this scheduler achieves good results since every application only is assigned as much bandwidth as it really needs, it makes signaling between the end-users and the base station necessary. In our case, we assume that a logical feedback channel exists between the client application and the scheduling entity in the base station [3]. The application condition is mapped onto a mean opinion score (MOS) scale by the metrics presented in Table 1. The MOS value is the arithmetic mean of individual subjective ratings of test users for the quality of a service. It ranges from 1 (poor) to 5 (excellent). According to the QoE model in [9], stalling (interruption in the video playback) is by far the main influence factor of YouTube QoE. Therefore, the current buffer level of the YouTube player is reported to the base station. For web browsing, the loading time of a web page is monitored and for downloads the current throughput is measured, c.f. [10]. In the following, we call this scheduler QoE feedback scheduler.

Table 1. QoE metric for the different applications

QoE	file download	web browsing	YouTube	Skype		
	throughput	page loading	buffer	frame rate	image quality I	resolution
[MOS]	[Mbps]	time [s]	level [s]	F [fps]		
1	< 0.25	> 5	< 2	-	-	160x120
2	0.25 - 0.5	3 - 5	2 - 4	-	-	320x240
3	0.5 - 1	2 - 3	4 - 8	$3 + \frac{F}{35fps} + (2I - 1)$		640x480
4	1 - 2	1.5 - 2	8 - 16			
5	> 2	< 1.5	> 16			

4 Modeling of the Investigated Applications

For application-aware scheduling, it is important to model the application accurately. This section describes the modeling of the four investigated applications for the evaluation, namely file download, web browsing, YouTube, and Skype.

4.1 File Download

The file download is the most simple application in the simulation. It represents the download of a big data file. Therefore, a best effort transmission over TCP is simulated. The HTTP protocol is not simulated. The size of the downloaded data can be specified by the user. Hence, the download only depends on the simulated physical link and the behavior of the TCP congestion avoidance algorithm.

4.2 Web Browsing

Web browsing of a user is modeled as follows. A web session consists of the download of a web page followed by an exponentially distributed reading time of a mean of 3 s. The web page itself consists of a main object and several embedded objects. Embedded objects are images, JavaScript code or CSS style sheet instructions. The number of embedded objects, the size of these objects and the size of the main object follow random variables whose distributions are listed in Table 2. TCP is used as transport protocol. The web server takes care about the TCP connection handling. The keep-alive timeout for HTTP/1.1 connections is set to 5s based on the values of the default configuration of the Apache web server. Furthermore, no speed or connection limit is set.

Table 2. Web session simulation parameters [3]

reading time	neg. exponential: Exp(3s)
volume main object	log-normal: $\ln\mathcal{N}$(10 kbytes, 25 kbytes) $\in$ [100 bytes, 2 Mbytes]
number of embedded objects	truncated Pareto(scale, shape, max): Pr(1.1,2,55)
volume embedded object	log-normal: $\ln\mathcal{N}$(8 kbytes, 126 kbytes) $\in$ [50 bytes, 2 Mbytes]

4.3 YouTube

The YouTube Flash Player and a YouTube download server is simulated for YouTube users. The player processes HTTP data to display the YouTube video. In particular, it calculates the current buffered video playtime in seconds. The player may stall if the playtime buffer is empty. The play-out delay after stalling is set to 3 s buffered playtime which is the current value of the YouTube video player. The YouTube download server behavior follows [11] with refinements according to own measurements. The download speed is controlled by the server in two phases. The size S_{ip} of the initial best effort phase depends on the mean data rate x of the Adobe Flash video. It corresponds to a buffered playtime of 40 s, hence it is calculated as

$$S_{ip} = 40s \cdot x. \tag{2}$$

The periodic phase sends data in blocks of 64 kB with a fixed inter-arrival time. The inter-arrival time $\varDelta T_{arr}$ depends on the target transmission rate which is 125 % of the mean data rate x of the Flash video, but has a maximum of 2.096 s. Therefore it is calculated as

$$\varDelta T_{arr} = min(2.096s, \frac{64kbytes}{1.25x}). \tag{3}$$

4.4 Skype-Like Video Conferencing

The objective is to model a Skype-like application that dynamically adjusts the video parameters depending on the network quality. For this purpose, measurements of Skype from February 2012 serve as a basis for modeling. This section is separated into two different parts, describing the server model for the sending behavior on the one hand, and the self-adapting client behavior on the other hand.

Sending Behavior. We consider only video calls. A call can be started and finished. Hence, it cannot be degraded to a voice call or instant messaging. Additionally, no connection process and buddy list updates take place in background. Only the downlink direction is taken into account. Due to this, the application in the simulation only performs unidirectional transmitting of data directly from a so-called Skype server to a client with UDP transport protocol. Consequently, server and client must be started two times for a realistic video call, as in both directions usually video is transmitted. It is assumed that Skype can adjust three parameters in order to adapt to the current network performance. According to our measurements, the frame rate p_{mfr}, the resolution p_{res}, and the image quality p_{qua} can be adjusted during video calls. The maximum frame rate is set to 35 fps. The image quality is modeled by a factor between 0 and 1. An image quality of 1 corresponds to the best quality. Values below 1 indicate that some kind of lossy compression is used. There are three resolutions available. They are '640x480', '320x240', and '160x120' which result in a data rate ratio of 100 %, 25 %, and 6.25 %. The Skype server periodically sends data blocks to the client. The block inter-arrival time can be described by

$$\varDelta T_{ar} = 1/p_{mfr}. \tag{4}$$

Next to p_{res} and p_{qua}, the blocksize additionally depends on a total data rate. It is set to $p_{tdr} = 1.2\,Mbps$. The maximum frame rate is set to $p_{mfr} = 35$. Consequently, the blocksize for our Skype-like application is described by

$$B_{size} = p_{tdr} \cdot p_{qua} \cdot p_{res}/p_{mfr}. \tag{5}$$

Self-adapting Behavior. In order to instantly react to poor network conditions, the application in our model measures packet delay. A high packet delay is assumed to be caused by high network load. Therefore, the client signals the server that it should enter a "poor network" routine in order to decrease the quality of the video call. This is exactly done if the measured mean packet delay during the last second exceeds the threshold of 75 ms. In the next step, the application resets the parameters for the frame rate and the image quality. Afterwards, periodic requests of the current packet delay serve to estimate the current performance of the connection. In case the packet delay stays below the threshold of 75 ms, the frame rate is increased in our model up to 17 fps. In case the packet delay exceeds the threshold, the image quality is decreased in steps of 0.1 to a minimum of 0.5. Afterwards, the resolution is decreased, while the image quality is set to 1. The algorithm stops if either the minimum resolution and image quality or the targeted frame rate is reached.

In addition, the application in our model also performs a second routine in order to increase the video quality, if possible. This routine runs periodically every 10 s during the complete Skype video call. It is only activated if the previous described routine is not active. If the packet delay is lower than a threshold of 35 ms, the algorithm starts to increase the video encoding until the packet delay exceeds the threshold. The encoding is increased every 500 ms. Our model first tries to increase the resolution, afterwards the image quality is increased and finally the frame rate is increased. If the packet delay exceeds the threshold during this procedure, the encoding is set back to the last working encoding and the routine stops.

5 Description of the Simulation

One mobile cell is simulated with a time-discrete event-based simulator for LTE mobile networks. The physical data transmission is performed on the basis of precalculated link-level curves for packet error and goodput from separate simulations with the LTE Downlink Link Level Simulator of the Technical University of Vienna[1]. The simulator implements a complete signal processing chain for the traffic channel. PHY and MAC functions are implemented according to LTE release 8 [12] as specified in [13,14]. A carrier frequency of 2.5 GHz, a bandwidth of 5 MHz, and a cell diameter of 250 m have been chosen. The signaling and control channels are simulated as error-free. Based on this physical simulation a complete system model is implemented with TCP transport protocol and application layer. TCP Cubic with congestion control, error detection and flow control is simulated for each user to obtain realistic scenarios even in overload situations. The propagation model for the data transmission consists of path loss, shadow fading, and multipath fading. Path loss is calculated according to the Winner II urban macro-cell model [15]. Furthermore, the shadow fading decorrelation distance is set to 50 m. The users move around randomly within the cell with a speed of 1 m/s. For this purpose, 200 SNR channel traces has been precalculated

[1] http://www.nt.tuwien.ac.at/ltesimulator

since on the fly computation is very time consuming. One SNR channel trace is assigned to each user with a random time offset. The users are able to watch YouTube videos, conduct Skype video calls, download files, or surf the Internet. On the packet level, a queue based scheduling at the base station is done.

Only the downlink is considered in this work, since it is assumed that this constitutes the bottleneck of the access network. The transmission is controlled by a packet scheduler. Each user has a packet buffer which is limited in size. The packet scheduler chooses the packets from the user queues according to the scheduling algorithm and passes them to the resource allocator. The resource allocation then selects the appropriate modulation and encoding based on the link-level curves depending on the users channel and places it in the frame.

6 Scheduling Taking into Account the Adaptation Behavior of Skype

As an example for semi-static scheduling, we demonstrate in the following how the adaptive capabilities of Skype can be exploited to gain bandwidth for other users. In overload situations an application like Skype, which adaptively adjusts to different network situations, is restricted to a reasonable extent to obtain resources for other applications. Such a redistribution of resources does not necessarily mean that the network resources are shared fairly. A reasonable allocation must be achieved depending on the application type (video, browsing, gaming, e.g.) because a high definition video obviously needs more bandwidth than a user surfing the Internet. The overall goal is a consistent and equal QoE on application level for all users.

The simulated scenario includes 7 Skype and 7 YouTube users and is simulated for 80 s. One Skype user and two YouTube users are shown exemplarily in the figures. The explanation of the scheduling behavior is based on these users. The channel conditions of the Skype user and the two YouTube users during the simulation are depicted in Figure 1. The YouTube users have good channel conditions at the beginning and up to a simulation time of 50 s. During this time period they achieve the maximum possible bandwidth.

Figure 2 shows the resulting behavior of the applications. In this case the proportional fair scheduler is used. Figure 2(a) depicts the throughput of the YouTube users, their corresponding buffer levels during the simulation and the mean data rates of the videos. YouTube user A with a mean data rate of about 1.2 Mbps maintains full video playback buffers during the complete simulation. In contrast, the video playback of YouTube user B is interrupted for 5 times during the simulation since his throughput is below the mean data rate of his video. Figure 2(b) shows the throughput of the Skype user and resulting video quality during the simulation. The resolution is not shown here since it stays at the maximum during the whole simulation. However, the frame rate and the image quality is changed often after the YouTube users have started their transmission. This can be explained by the insufficient channel quality of the user and the congestion in the cell. As soon as the channel conditions for the Skype user are better, the video encoding returns to its maximum.

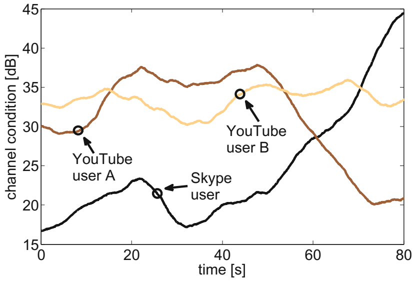
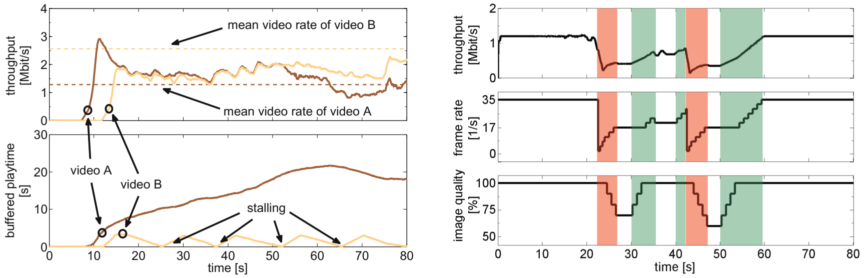


Fig. 1. Channel of users during the simulation



(a) Throughput and buffered playtime of YouTube users

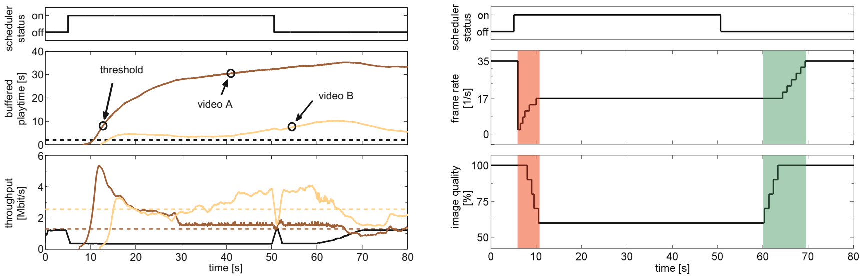
(b) Throughput and video properties of the Skype user

Fig. 2. Skype and YouTube scenario with proportional fair scheduler

After 50 s the channel quality of YouTube user A drops significantly which results in about half of the available bandwidth for this user. The channel for the Skype user is of low quality from 0 s to 50 s, but improves quickly after 50 s.

Figure 3 shows the simulation of the same scenario with the on demand flow-based service class described in Section 3.2. Figure 3(a) depicts the situation for the YouTube users. The restriction for Skype users is activated immediately after the YouTube users start the transmission. The throughput of both YouTube users is significantly higher now, compared to the scenario with the proportional fair scheduler. Between 20 s and 30 s the throughput of user B drops below the mean data rate of his video. Yet this does not lead to stalling (interruptions in the video playback) since YouTube user A enters his periodic phase after 30 s and so the throughput of YouTube user B exceeds the mean data rate of his video again. Both YouTube users experience no stalling during the whole simulation. Their QoE is not degraded. After 50 s the service class is deactivated since both YouTube users completed their initial phases and their buffer levels are higher than three seconds.

Figure 3(b) shows the video encoding of the Skype user for the scheduling with the on demand flow-based service class scheduler. As soon as the service class



(a) Throughput and buffered playtime of YouTube users for on demand flow-based service class scheduling

(b) Video properties of the Skype user for on demand flow-based service class scheduling

Fig. 3. Skype and YouTube scenario with on demand flow-based service class scheduler

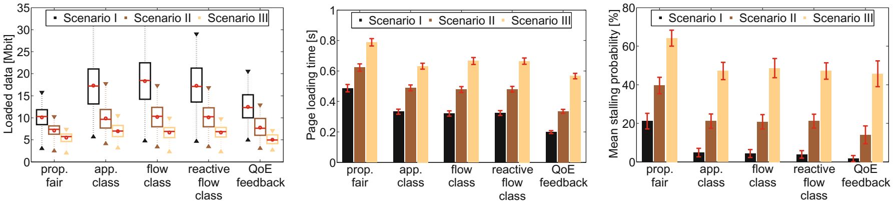
is activated, the encoding drops to an image quality of 60 % and a frame rate of 17 fps. The resolution nevertheless remains at its maximum. Shortly after the deactivation of the restriction for the Skype user, the encoding changes again. It returns to its maximum within 10 seconds.

From these results, it can be concluded that a scheduling mechanism, which is aware of the Skype-like adaptation capabilities, is able to use this knowledge in temporary overload situations to help other applications without these capabilities. This is based on the assumption that a stalling of YouTube is considered worse than a controlled short-term change in the video encoding parameters of Skype [16]. However, the benefit is dependent on a large number of active Skype users in the cell because each Skype-user only provides a small resource gain. For network operators, this means that this solution is only useful if enough Skype users are on the network.

7 Investigation and Comparison of Scheduling Approaches and Their Effect on Applications

In the following, the performance of the schedulers described in Section 3 is statistically evaluated with the aim to compare the benefit and the impact of the different approaches on the applications. The following paragraph describes three different scenarios that have been chosen for the evaluation. 100 runs are conducted per scheduler and scenario. Afterwards the impact of the schedulers is described for each application separately. The evaluation is based on carefully chosen application quality indicators which provide a high QoE correlation: For YouTube the buffered playtime is used, since stalling is the main factor for a QoE degradation [9]. For web browsing the download time of the content is chosen while for file downloads the amount of downloaded data is considered as key quality indicator, c.f. [10]. For Skype the image quality and the steadiness of the video encoding are used.

Scenario Description. Scenario I, II, and III represent different situations in a cell. In the first scenario the cell is frequented only moderately, in the second scenario highly. In the third scenario the system is overloaded. The Skype users, download users, and web browsing users start directly at the beginning of the simulation. The starting time of the YouTube users is calculated from a uniform distribution between second 5 and second 20. For each user one out of 10 videos is randomly chosen. The system is simulated for 100 s. Scenario I simulates 19 users. There are 7 Skype users, two download users, four web browsing users, and 6 YouTube users. Scenario II consists of 25 users. There are 8 Skype users, three download users, 6 web browsing users, and 8 YouTube users. Scenario III simulates 34 users. There are 10 Skype users, four download users, 10 web browsing users and 10 YouTube users.



(a) Average loaded data of the file download users

(b) Average loading time of the web browsing users

(c) Average stalling probability of the YouTube users

Fig. 4. Evaluation of the amount of downloaded data, the page load time of web browsing users, and the stalling probability of YouTube

File Download. Figure 4(a) shows the amount of downloaded data of the file download users for the five different schedulers during all three scenarios. The red line indicates the median of the loaded data, the red circle indicates the average loaded data. The box shows the 40 % quantile of the results. The complete range is indicated by the dotted line and the triangles. The results differ but a tendency can be observed. With a higher number of users in the system the amount of data that the users are able to load decreases. Accordingly, the 40 % quantiles and the entire range of the results become smaller indicating less variance. The three service class schedulers provide comparable results. On average, users are able to load about 17 Mbit in Scenario I, 10 Mbit in Scenario II, and 7 Mbit in Scenario III. Both, the proportional fair scheduler and the scheduler with QoE feedback offer less quality to the file download users. The scheduler with QoE feedback however performs slightly better in Scenario I and II (12 and 8 Mbit) than the proportional fair scheduler (10 and 7 Mbit). For the overloaded Scenario III, the proportional fair scheduler shows similar results as the QoE feedback scheduler.

While the scheduling with service classes performs by far the best regarding the file download users, it should be noted that a lower download throughput might be beneficial for the other users. File downloads are flexible applications and a certain delay or a low download throughput can be tolerated. From a QoE perspective, assigning a lot of bandwidth is not necessary or even a waste of resources. Therefore, a more moderate behavior like the one of the QoE feedback scheduler is desirable to gain resources for other applications as explained in the next section.

Web Browsing. Figure 4(b) shows the average page loading time of the web browsing users between second 30 and 80 of the simulation for the five different schedulers during all three scenarios. Only the transfer time within the mobile network is considered. Delays by the web server or the transmission over the Internet are not included here. The 95 % confidence intervals are indicated by the red lines on top of the bars. The results of all schedulers show the same tendencies. If the number of active users inside the system increases, the page loading time increases, too. However, the respective loading times of the scenarios differ with different schedulers. While there is no significant difference between the three service class schedulers, the proportional fair scheduler shows significantly worse performance by about 0.1 s. The QoE scheduler provides loading times of 0.2 s, 0.3 s and 0.6 s which are about 0.1 s better than the times of the service class schedulers.

Even the service class schedulers which do not use detailed information about the web browsing users, show significant advantages in the average page loading times over the proportional fair scheduler. This benefit is gained by the restriction of the Skype users. The QoE scheduler performs also better. It demonstrates that basic knowledge about the application conditions can improve the scheduling process.

YouTube. Figure 4(c) shows the average stalling probability for YouTube users. Stalling is defined as at least one interruption between second 30 and 80 of the simulation. While the stalling probability increases with a larger number of users in the system, the use of the different application-aware schedulers has no significant impact on the results. All application-aware schedulers result in similar stalling probabilities for all three scenarios. These are about 4 % in Scenario I, 21 % in Scenario II, and 47 % in Scenario III. Especially in the overload Scenario III, stalling can not be completely prevented. The proportional fair scheduler achieves the worst results in all three scenarios with stalling probabilities of 21 %, 40 %, and 64 %.

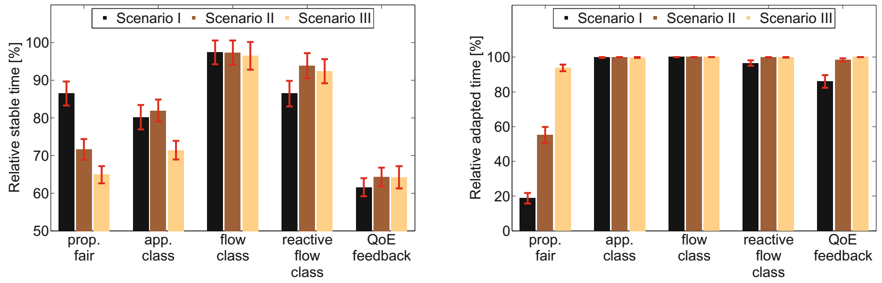
Skype. Figure 5(a) shows the average time period of a steady video quality relative to the usage time of Skype. The video encoding is considered steady if it does not change within 2 seconds. The results for the different schedulers vary. The video encoding is the most stable with the service class schedulers since they guarantee a certain data rate which is necessary for Skype users. The scheduler with QoE feedback achieves the worst results regarding the relative time period of steady video quality with about 62 % in all three scenarios. The proportional fair scheduler shows different results for the three scenarios. The relative time period of steady video quality is very high (about 86 % in

Scenario I), but much lower for Scenario II (72 %) and Scenario III (65 %). The application-based service class performs comparably in the first two scenarios with a relative steady time of about 80 % but decreases in Scenario III to about 70 % due to the overload. The two flow-based service class schedulers achieve by far the best results.

For this it is concluded that for some applications a smaller but more steady throughput is more efficient than a slightly higher but fluctuating one. The video encoding of Skype is very unstable for schedulers like the QoE feedback scheduler or the proportional fair scheduler. The reason for this is the constantly changing bandwidth assignments with these schedulers.

The downgrade algorithm which is essentially responsible for the adaptation to new network conditions always results in a frame rate of 17 fps if there is congestion in the network. The main changing feature is the image quality. Therefore, the crucial question is how often the video encoding is not at its maximum while Skype is in an adaptive mode. Figure 5(b) shows the average time period of Skype's video coding not being at maximum. During all three scenarios and with all five schedulers the results are similar. Skype is in an adaptive mode during 90 % of the time period for all schedulers, but for the proportional fair scheduler. The proportional fair scheduler tries to provide bandwidth fairness with respect to the throughput. Since Skype requires little bandwidth compared to applications like YouTube or file downloads, this is sufficient and it is much more often in a non-adapted state with the maximum video quality. The more users are in the system, the higher is the probability to be in an adapted state. In Scenario I the probability is 18 %, in Scenario II 55 % and in Scenario III it is 93 %.

For some applications certain compromises have to be made. The proportional fair scheduler is the only one able to provide a high and steady bandwidth so that the maximum video encoding can be used with Skype. However, as Figure 5(a) shows, this is achieved at the expense of a constant quality and to the disadvantage of other applications. Even the scheduler with QoE feedback which



(a) Average time period of steady video quality relative to the usage time of Skype

(b) Average time period of Skype's video coding not being at maximum

Fig. 5. Skype video properties

is the most unstable regarding the video quality does not achieve significantly higher probabilities for the maximum encoding. This is due to the load in the network of the other applications which are too greedy to allow this scheduler to assign high bandwidths for flexible applications such as Skype.

Generally considered, the evaluation of the application-aware schedulers in comparison to the proportional fair scheduler demonstrated that application awareness improves the overall situation of the applications in the network. However, there are different results for different applications. For example, the YouTube application has achieved useful results with a scheduler with QoE feedback because the latter can tolerate dynamic changes in the bandwidth due to the video buffer. The Skype application in contrast does not require such a scheduling. For Skype video the QoE scheduling results in a very unstable situation due to the automatic adaptation. In contrast, Skype can benefit from a static service class with guaranteed QoS properties like small delay and fixed throughput.

8 Conclusion

In this paper, we investigated the impact of using application information for scheduling decisions on the downlink within cellular networks on the user perceived quality. In particular, we investigated different scheduling approaches together with varying degrees of available application information like application type, application status or application intelligence, and highlighted their impact on key QoE indicators for web browsing, file downloading, progressive video streaming and Skype video conferencing.

Our results reveal a strong improvement of the overall QoE in case of application-aware scheduling for the investigated use-cases. We further showed that each of the considered applications can benefit from an application-aware scheduling. However, different applications may prefer different types of scheduling. Furthermore, for some applications a large signaling overhead is required. For YouTube a very flexible scheduling can be carried out due to the buffering of the video content. This can be exploited to obtain a multi-user diversity gain. However, the signaling of the buffer level to the scheduling entity is required. Skype in contrast does not require a dynamic scheduling according to our results. It can sufficiently benefit from a static service class with guaranteed QoS properties like small delay and fixed throughput. The results quantify the trade-off between the degree of application information and the gain in terms of QoE. Thus, our evaluations allow to compare different candidate approaches with theoretical thresholds and to select the most appropriate ones.

Future work will deal with a precise evaluation of the signaling overhead, i.e., the costs, and the consideration of other applications for the concept of QoE-oriented and application aware scheduling. In addition combined scheduling approaches will be investigated to allow a scenario-independent optimization of the overall QoE.

References

1. Khan, S., Duhovnikov, S., Steinbach, E., Kellerer, W.: MOS-Based Multiuser Multiapplication Cross-Layer Optimization for Mobile Multimedia Communication. Advances in Multimedia 2007 (2007)
2. Ameigeiras, P., Ramos-Munoz, J.J., Navarro-Ortiz, J., Mogensen, P., Lopez-Soler, J.M.: QoE oriented cross-layer design of a resource allocation algorithm in beyond 3G systems. Computer Communications 33(5), 571–582 (2010)
3. Wamser, F., Staehle, D., Prokopec, J., Maeder, A., Tran-Gia, P.: Utilizing Buffered YouTube Playtime for QoE-oriented Scheduling in OFDMA Networks. In: International Teletraffic Congress (ITC), Krakw, Poland (September 2012)
4. Thakolsri, S., Khan, S., Steinbach, E., Kellerer, W.: QoE-Driven Cross-Layer Optimization for High Speed Downlink Packet Access. Journal of Communications 4(9), 669–680 (2009)
5. Song, G., Li, Y.G.: Utility-Based Resource Allocation and Scheduling in OFDM-Based Wireless Broadband Networks. IEEE Communications Magazine 43(12), 127–143 (2005)
6. Bohnert, T.M., Staehle, D., Kuo, G.-S., Koucheryavy, Y., Monteiro, E.: Speech Quality Aware Admission Control for fixed IEEE 802.16 Wireless MAN. In: IEEE ICC, Beijing, China (May 2008)
7. Bohnert, T.M., Staehle, D., Monteiro, E.: Speech Quality Aware Resource Control for Fixed and Mobile WiMAX. In: Marcos Katz, F.F. (ed.) WiMAX Evolution, p. 227. John Wiley and Sons (January 2009)
8. 3rd Generation Partnership Project., 3GPP TR 22.805 V12.1.0; Feasibility study on user plane congestion management (Release 12), v12.1.0 (December 2012)
9. Hoßfeld, T., Schatz, R., Seufert, M., Hirth, M., Zinner, T., Tran-Gia, P.: 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)
10. Egger, S., Hoßfeld, T., Schatz, R., Fiedler, M.: Waiting Times in Quality of Experience for Web Based Services. In: QoMEX 2012, Yarra Valley, Australia (July 2012)
11. Alcock, S., Nelson, R.: Application flow control in youtube video streams. ACM SIGCOMM Computer Communication Review 41(2), 24–30 (2011)
12. 3GPP Technical Specification Group RAN, E-UTRA; LTE physical layer – general description, 3GPP, Tech. Rep. TS 36.201 Version 8.3.0 (March 2009)
13. 3GPP Technical Specification Group RAN, E-UTRA; physical channels and modulation, 3GPP, Tech. Rep. TS 36.211 Version 8.7.0 (May 2009)
14. 3GPP Technical Specification Group RAN E-UTRA; multiplexing and channel coding, 3GPP, Tech. Rep. TS 36.212 (March 2009)
15. Winner II consortium, Channel Models Part II: Radio Channel Measurements and Analysis Results, Deliverable 1.1.2," IST-4-027756 WINNER II, Tech. Rep. (September 2007)
16. Zinner, T., Hoßfeld, T., Minash, T.N., Fiedler, M.: Controlled vs. Uncontrolled Degradations of QoE The Provisioning-Delivery Hysteresis in Case of Video. In: New Dimensions in the Assessment and Support of Quality of Experience (QoE) for Multimedia Applications (June 2010)