Dynamic HTTP Download Scheduling with Respect to Energy Consumption

Andreas Blenk
and Wolfgang Kellerer
Institute for Communication Networks
Technische Universität München, Germany
Email: [andreas.blenk,wolfgang.kellerer]@tum.de

Florian Wamser, Thomas Zinner
and Phuoc Tran-Gia
Institute of Computer Science
University of Würzburg, Germany
Email: [wamser,zinner,trangia]@informatik.uni-wuerzburg.de

***Abstract*—Mobile devices, in particular smartphones, are battery powered. As their small size limits their energy capacity, an intelligent and energy-saving use of resources is crucial for the mobile user experience. Especially energy saving while sending and receiving data becomes important due to the always-on nature of smartphones. One solution for a sophisticated resource usage in wireless access networks is to tailor resource management to applications. The advantage of such mechanisms is to be able to address both the perceived quality of the users and the energy consumption of mobile devices at the same time. We propose several traffic scheduling algorithms for HTTP file download in wireless networks based on this application-aware paradigm. The core idea is to use download scheduling that avoids parallel transmissions and, in contrast, favors sequential transmission. We have implemented and measured the algorithms in a wireless mesh network. The evaluation results quantify the gain of an application-aware resource management with respect to quality of experience and energy consumption.**

***Index Terms*—Quality of Experience, Access Networks, Application-aware Resource Management, Traffic Prioritization**

I. Introduction

Huge efforts are undertaken to save energy at the terminal in today's mobile networks. Cell phones and smartphones are getting smaller while offering an increasing number of features. Quad-core CPUs and high resolution displays are typical features, so that one can go online with the smartphone as with a desktop computer and can retrieve news and messages at any time. These technical innovations tacitly assume that the battery has enough capacity to power the smartphones several days without recharging.

One indispensable rule for manufacturers of new mobile phones is today: the smartphone's central or graphic processing unit (CPU or GPU) must have decreasing power consumption and increasing performance per watt [1]. Taking the energy consumption profile of smartphones into account, it can be observed, however, that most energy is consumed while sending and receiving data over the network. It is shown that WiFi approximately uses 7 times more energy than CPU and RAM, and GSM about 6 times more [2]. Consequently, several new approaches attempting to save energy in the network by exploiting intelligent mechanisms are proposed [3]–[6] (see Section II).

The smartphones' energy consumption may also benefit from new resource management approaches that are deployed in the network. Emerging mechanisms such as discontinuous reception and transmission (DRX/DTX) of HSPA+ in 3GPP [6], [7] enable the network to influence the smartphones' energy consumption. The network determines time intervals where no data is send to a smartphone and, thus, the smartphone can enter a power saving state. The combination of both network resource management and emerging mechanisms such as DRX/DTX opens new possibilities to save energy at the terminal.

We investigate mechanisms to reduce the download time of multiple mobile users in wireless environments in this paper. Despite file downloads from applications like Dropbox or Picasa, this also includes software updates which may be automatically triggered by installed applications. The idea is to avoid long parallel downloads, and to allow consecutive short downloads with high data rate. This concept is related to approaches from process scheduling in operating systems. It addresses the trade-off in terms of waiting time and overall processing time in case of scheduling processes either parallel or sequentially. Accordingly, we try to avoid the parallel usage of network resources by different devices, where all devices consume energy. This is done by throttling all downloads despite a single one that consequently downloads faster. In our study we answer the research question whether such a mechanism is able to reduce the overall energy consumption. In addition, we investigate how much energy savings may be achieved by making a smartphone sleep and waking it up when it is scheduled to download.

This paper presents three different download scheduling algorithms and shows by means of practical measurements how they affect the user. It first introduces the basic idea of all algorithms and outlines the algorithms more in detail in Section III. In Section IV, the evaluation setup and the performance metrics are introduced. In Section V, we present the results of the performance evaluation of the proposed algorithms in terms of download times, discuss their impact on the user perceived quality and elaborate their influence on the energy consumption. An implementation of the algorithms

This work has been supported by the German Research Foundation (DFG) in context of the project *FunkOFDMA*.

in a cellular network is introduced in Section VI and, finally, we draw conclusion in Section VII.

II. Current State of the Art

Research on energy consumption on mobile devices can be divided into two distinct areas. There are approaches to reduce the energy consumption of the hardware [1], [5], [8] and to control energy-intensive operations at the smartphone such as transmitting data [6], [7], [9]–[11].

Multi-core CPUs are highlighted in order to save energy on mobile phones in [1]. The authors of [5] propose to save energy within the hardware by using an adjustable power management with a hierarchical cascaded power gating at the integrated circuits.

In contrast, approaches such as DRX/DTX try to control energy-intensive operations. DRX/DTX [6], [7] addresses the problem that applications do not just download data and then go offline, but stay online for a longer period of time. In mobile networks, this means that the mobile phone stays connected which consumes a lot of energy. With DRX, the mobile can go into power saving states regularly. Thus, it is able to remain online at relatively low power consumption for a longer time.

Another approach proposed in the literature is to save energy by switching the current transmission technology to a more energy efficient one. Mobile networks are more widespread than WiFi networks, but require more energy for data transmission [12]. Therefore, delaying transmission until WiFi is available provides opportunities to save energy. The authors of [11] examine such impact of data offloading to WiFi networks with respect to energy consumption.

Further, there are approaches addressing the scheduling at the base station. A study about minimal transmission power for uplink systems of LTE is done in [9] . In contrast, an efficient scheduling is proposed to reduce the energy consumption of the base station in [10]. Both approaches make use of a special scheduling to save energy, which is similar in the implementation to our approach. However, they focus only on intelligent adjustment of transmission power without considering application information.

III. Scenario and Algorithm Definition

We define the scenario for which we have designed our new application-aware resource management algorithms first. Secondly, we outline the new algorithms in more detail.

A. Scenario and Basic Idea

In the investigated scenario, n users, $n \in \mathbb{N}$, perform k downloads, $k \in \mathbb{N}$, in a wireless environment. All users share one limited network resource, e.g. the data rate available in the wireless environment. According to a resource management algorithm, the available network resource is shared among the users, i.e., among the network flows f_k with $k \in \mathbb{N}$, where k belongs to a download. In particular, a resource management algorithm decides which data that belongs to flow f_k of all users n is transmitted. In our approach, the network resource allocation of the flows is done according to an algorithm dependent ordering metric O. Traditionally, wireless resource management mechanism deploy proportional fair scheduling mechanisms.

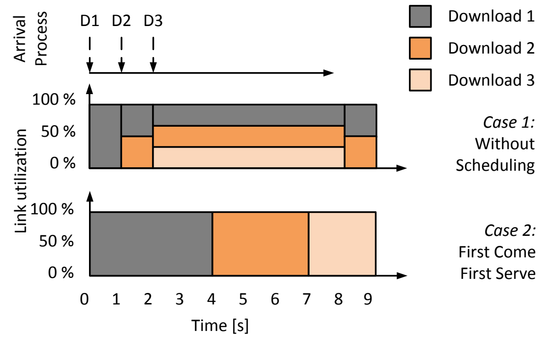


Fig. 1. Download process with and without a scheduling strategy.

Before describing our proposed algorithms more in detail, we explain their idea. Figure 1 illustrates two approaches to allocate a network resource to multiple users performing a download. Three downloads are arriving sequentially at time 0, time 1, and time 2. The first case illustrates the traditional approach with fairness on flow level. Due to the limited knowledge about the connections, e.g. the type of data that has to be transmitted, it allocates the available capacity equally between all network users. As we can see in the first time-slot, Download 1 receives the full capacity, in the second time-slot, the capacity is divided between Download 1 and 2, and in the third time-slot, all downloads share the capacity fairly. In terms of overall waiting and processing time, however, a task may benefit from a sequential allocation of the resource, as typically done in process scheduling. As illustrated in the second case, the downloads are scheduled sequentially according to the *first come first serve* principle. This decreases the average download time for the presented example from 7.7 seconds for the traditional case to 5.7 seconds for the sequential case. In addition, preferring small files over big files may additionally reduce the average download time, i.e. the sum of waiting time and loading time, of all download users. However, as this is only a special case, the general performance is evaluated in Section V.

B. Algorithms

In the following, we introduce three download scheduling algorithms and discuss their trade-offs.

1) First-Come First-Serve (FCFS): Let A_k be the arrival timestamp of flow f_k, then, all network resources are allocated according to

$$O_{FCFS} = A_k \ .$$

2) Shortest File Size First (SFF): Let Q_k be the amount of download data that is requested in the HTTP header, then, the available network resources are scheduled according to metric

$$O_{SFF} = Q_k \ .$$

3) Shortest Remaining File Size First (SRFF): Let t be the update time of all downloads. At each time t the network resources are ordered according to $Q_k - F_k$ that indicates how much data has already been downloaded:

$$O_{SRFF} = Q_k - F_k \ .$$

F_k is the amount of already loaded bytes and is updated each t by setting $F_k = f_{kt}$ with f_{kt} equal to the already downloaded data of download flow f_k. It has to be noted that the update time can occur periodically or can be triggered by the downloads in a reactive manner. More information about the implementation issues of the update time t is given in Section IV.

The FCFS is a non-preemptive download scheduler that has the lowest implementation complexity. Here, the scheduling depends only on the time at which the downloads are requested. As soon as a download finishes, FCFS schedules the download with the oldest request time next. However, the FCFS may prefer one big file over multiple small files, which is not optimal. A resource management algorithm should schedule all downloads sequentially based on their file size or even their remaining file size, which has already been proofed to be optimal for processes [13]. The information about the file sizes have to be detected in case of the non-preemptive SFF download scheduler. When a download finishes, SFF schedules the download with the smallest file size next. The SRFF is a preemptive download scheduler that has the highest complexity. It always needs the information about the current state of all downloads. To provide this information, the already downloaded bytes have to be extracted from the downloads steadily. Based on this information, the SRFF interrupts the currently scheduled download and always allocates the network resources to the download with smallest remaining file size.

In order to avoid the starvation of downloads, we further recommend to reserve a minimum bandwidth for each download.

IV. Evaluation Setup and Performance Metrics

In this section, we provide further details about the evaluation setup and the performance metrics we use to evaluate the three algorithms.

A. Evaluation Setup

The proposed algorithms depend on information about the downloads that are active in the network. Application detection, i.e., the detection of download parameters, could be done by inspecting packet headers. If an HTTP header has been detected, further parameters are extracted by pattern matching according to the chosen algorithm [14]. An alternative approach could be direct signaling of the relevant parameters by the application. A user terminal program directly reads out the required parameters and signals them to a network proxy, which avoids the packet inspection in the network. In our experiments, we rely on the fact that we have access to the terminals. Instead of packet inspection, we install a program called *download monitor* on the terminals that directly provides the network with the necessary information. The download monitor sends these information about the downloads to the network to provide the resource management mechanisms with the download information in a reactive manner. Detailed information about the frequency of sending updates are implementation specific and out of scope of this work. However, it might affect the performance and needs further investigation in future work.

The study is carried out in a Wireless Mesh Network (WMN) as an experimental evaluation in an LTE network is practically not feasible. The WMN consists of one wireless access point granting access for the mobile user devices and one mesh node connecting the access point to the WMN's Internet gateway. The mobile users are requesting files with different sizes from a server located in the Internet. The download monitor on the mobile devices is extracting and sending information about the users' file downloads to a resource manager running on the access point. Based on the information about all current file downloads, a download algorithm specifies the data rate allocation for the network flows, i.e., it may delay flows of specific users.

B. Performance Metrics: QoE Model and Energy Consumption

This section specifies both, a model for user satisfaction as well as a model for energy consumption.

1) QoE of HTTP Downloads, Waiting Times: According to [15], the download time has a big impact on the QoE of a user. The following function describes the relationship between download time and QoE for downloads of a file size of 10 MB.

$$U_{QoE} = -1.68 \ ln(d_L) + 9.61 \tag{1}$$

U_{QoE} is the estimated mean opinion score (MOS) for one user and d_L is the perceived download time, i.e. the waiting time plus the loading time.

2) Energy Consumption: One model for each technology, i.e., LTE, 3G and WiFi, is provided in [12] to calculate the energy consumption respectively. Let t_u be the uplink throughput in [Mbps] and t_d the downlink throughput in [Mbps], then power P is:

$$P = \alpha_u t_u + \alpha_d t_d + \beta \tag{2}$$

Here, α_u, α_d and β are set according to the technology, i.e. LTE, 3G and WiFi.

Figure 2 shows the power needed in [mW] depending on the data rate in [Mbps] of the uplink and the downlink for all technologies. In case of a data rate of 0 Mbps, a mobile device already consumes energy. It is the amount of power consumed in idle mode. In case of WiFi, the amount of minimum consumed power is 190 mW. It can be observed that the gap between the download and the uplink power consumption is much smaller than in the case of LTE. The minimum consumed power for 3G is 810 mW. In general, LTE provides the highest data rate which is necessary for the

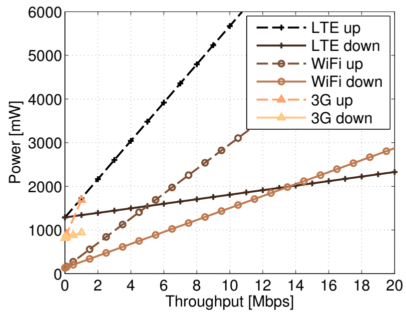


Fig. 2. Power Model for 3G, LTE and WiFi [12].

emerging high quality applications. However, it also consumes the most power compared to 3G and WiFi.

V. Evaluation

This section evaluates the performance of the application-aware scheduling for file downloads in terms of download time and energy consumption in a WMN. The overall goal is to decrease a file's download time to meet quality of experience expectations. We then show how the scheduling can also be used to reduce the energy consumption. Finally, we show a calculation based estimation of how this may reduce the energy consumption for different wireless technologies. In all scenarios, the results of the algorithms are compared to a setup without any resource management (NoRM). The total available data rate of the WMN is limited to 3 Mbps.

A. Influence of file size

The file size may have an impact on the effectiveness of the download algorithms. The influence of signaling and determination of the additional download information can diminish the benefit of the algorithms for small files. In other words, the file download has already been completed before scheduling becomes effective.

The study is performed 200 times for 5 users. The average download time increases linearly for larger file sizes as illustrated by the dashed lines in Figure 3. The gradient in case of NoRM is larger than the gradient of the FCFS. In case of NoRM, parallel running downloads are increasing the average download time. For a file size of 1 MB, however, the mean of the download time converges. The additional effort to determine the download parameters diminish the benefit of the algorithms. As a consequence, we use file sizes between 3 MB and 6 MB to show the benefit of our algorithms in more detail.

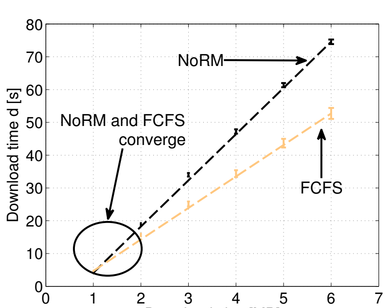


Fig. 3. Convergence for download file sizes.

B. Impact on the Download Time

As a proof of concept for all algorithms, we let five users download the same 5 MB file. The inter-arrival time between the file requests is fixed for 100 runs. Figure 4(a) shows the CDF of the download time d for all users of all runs. All download users need less download time than in the NoRM case. Further, as the number of users and the inter-arrival time is fixed, the CDFs proceed in steps. As there is no difference in the file size and also in the remaining file size, FCFS, SFF and SRFF have the same effect. More precisely, by improving the download time for a part of the downloads the average download time can be decreased.

In another setup, we investigate the behavior of the algorithms for different file sizes. The users are now downloading 3 MB and 6 MB files. The inter-arrival time is again fixed. Figure 4(b) shows that 75% of the measured download times range from $44 - 80$ seconds in case of NoRM. The FCFS decreases the download time of all downloads compared to the default case. Furthermore, the SRFF and the SFF perform similar during all runs due to the fixed inter-arrival time. We can conclude that the SFF and the SRFF improve the download time in case of different file sizes compared to the FCFS.

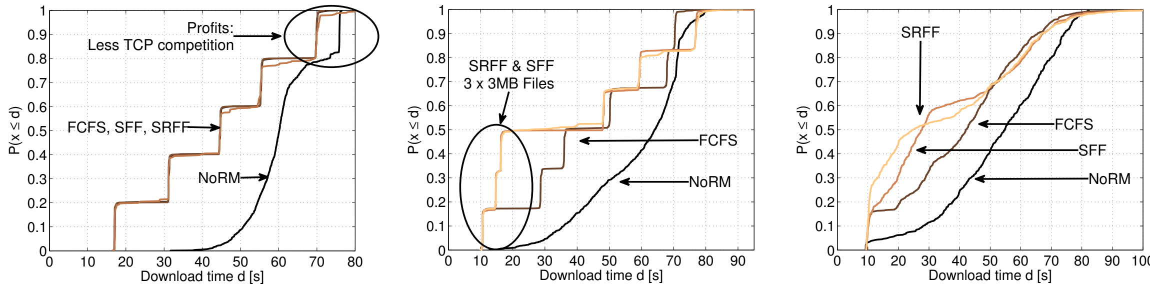
In the third setup, we examine the impact of the algorithms for more occasional events. The inter-arrival time for all download users is exponentially distributed with a mean of 5 seconds. The investigated file sizes are 3 MB and 6 MB. Figure 4(c) illustrates again that active scheduling improves the overall network performance. The SRFF shows the best performance in terms of download time. Based on the knowledge about the currently smallest remaining file size, it schedules the download that promises the fastest delivery. We conclude that users benefit most from the SRFF and the SFF, whereas the performance of the FCFS may degrade for a big variance of the downloads' file sizes.

C. Impact on the Power Consumption

As the relation between the energy consumption and the download time is linear according to Equation 2 given in Section IV-B, the energy consumption can be derived from the corresponding download time. Particularly the WiFi energy consumption model is investigated in the following. Figure 5 shows the boxplots for all previous setups. The energy consumption is given in Joule [J].

As we can see in Figure 5(a), no value is smaller than 9 J and 50% of all values lie between 13 J and 14 J in the NoRM case. For the proposed algorithms, 50% of all values lie between 10 J and 13 J. The spread of the algorithms ranges from 7 J to 18 J. This shows that the energy consumption could not be reduced for all file downloads. More precisely, the algorithms only improve the consumption of a part of all devices and therefore they decrease the average energy consumption.

As we can see in Figure 5(b), the schedulers decrease the average energy consumption compared to the NoRM case. The SFF and the SRFF moreover decrease the average energy consumption compared to the FCFS. The NoRM case shows an energy consumption ranging from 9 J to 16 J for 50% of the

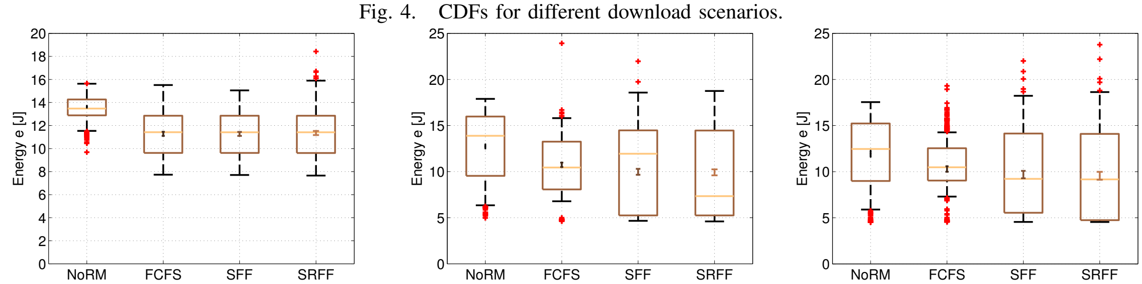


(a) Equal file sizes.

(b) Different file sizes

(c) Different file sizes with random inter-arrival time.

Fig. 4. CDFs for different download scenarios.



(a) Equal file sizes.

(b) Different file sizes.

(c) Different file sizes with random inter-arrival times.

Fig. 5. Boxplots for different download scenarios.

devices. For the FCFS, the energy consumption ranges from 9 J to 14 J for 50% of the devices. This is furthermore improved by the SFF and the SRFF, where 50% of all observed values lie between 5 J and 15 J. To sum up, the additional knowledge about the download parameters decreases the energy consumption of the devices on average.

Finally, Figure 5(c) confirms that the average consumed energy decreases with more knowledge about the download files. We conclude again that users benefit most from the SRFF and the SFF, whereas the performance of the FCFS may degrade for a variance of the downloads' file sizes.

D. Power Consumption for Different Wireless Technologies

After investigating the WiFi power consumption, we compare the results of the occasional scenario for different wireless technologies in the following. Figure 6 shows the power consumption for NoRM, FCFS, SFF and SRFF in case of random inter-arrival times and different file sizes. In general, the gain is the largest in LTE because LTE uses the most energy in absolute terms. Also for 3G, all algorithms decrease the consumed energy compared to the NoRM case. However, in case of WiFi, the performance improvement and the energy consumption is smaller since WiFi overall consumes less energy. Thus, we conclude that LTE and 3G users will benefit most from an active application-specific resource management. This emphasizes that cellular networks may benefit from performing our approaches for example at the base stations.

E. Enable/Disable Idle Option - A Possible Improvement

As described in the introduction, a convenient approach for saving energy is the mechanism of smartphones to be able to temporarily suspend their transmission. Therefore, we consider theoretically how high the savings in energy would be if the mobile phone is able to go idle. As we only want to highlight the possible benefits of algorithms that avoid parallel data transmissions, we only investigate the results for NoRM and FCFS. Figure 7 shows two CDFs for the energy consumption for the NoRM, for the FCFS without the idle option, and for the FCFS with an enabled idle option. The FCFS without the idle option reduces the overall power consumption compared to the NoRM in the equal file size and the different file size scenario. However, the FCFS with an enabled idle option outperforms both the FCFS and the NoRM mechanism. Here, the mobile devices are only activated as long as they are downloading a file. In the lower figure, we can also see the effect of different file sizes. The step behavior is the result of energy consumed by the mobile devices for the 3 MB and the 6 MB files.

VI. Implementation Within a Cellular Network

Packet scheduling in a 4$^{\text{th}}$ generation OFDMA-based mobile communication network is concerned with the resource allocation such that a system utility function is optimized [16]. Such a utility function could be enhanced to take application

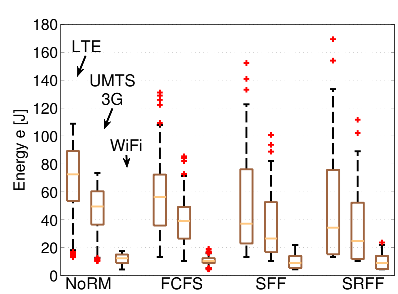


Fig. 6. Download times for LTE, 3G and WiFi, random inter-arrival time and different file sizes.

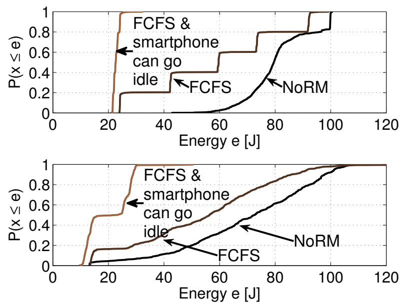


Fig. 7. Impact if smartphone is able to go idle in transmit pauses in conjunction with FCFS scheduling. Top: equal file sizes, bottom: random arrival and different sizes.

information into account. Figure 8 shows the system model of the OFDMA scheduling architecture.

The download scheduling should run in addition to a "traditional" packet scheduling mechanism that relies on typical radio performance indicators. Thus, the scheduling does not interfere with other mechanisms as long as no HTTP download is done. The application information determination could be done by inspecting packet headers. Accordingly, the appropriate scheduling mechansims can be utilized with respect to the available information. Similar scheduler extensions aiming at improving the QoE are discussed within the 3GPP specification on user plane congestion management (UPCON) [17].

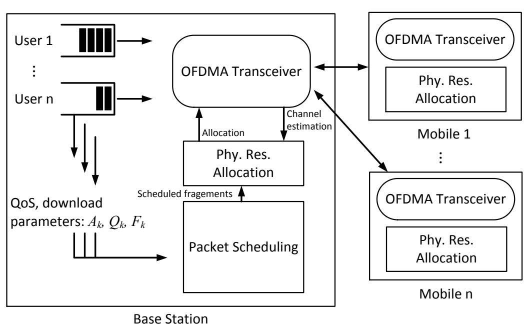


Fig. 8. System model of OFDMA scheduling architecture.

VII. Conclusion

Reducing the power consumption becomes a new important topic next to the goal of increasing the user perceived quality in today's wireless access networks. In this paper, we proposed new algorithms for scheduling file downloads of mobile users in wireless communication networks. Taking application-specific information into account, the algorithms schedule downloads sequentially in order to avoid the parallel and competitive resource usage of multiple users. Thereby, the scheduling decreases the energy consumption and the average downloading times of the mobile devices. Although the overall download times do not increase significantly, waiting times are introduced. The algorithms were implemented and evaluated in a wireless mesh network. The measurement results show how the scheduling affects the download time and the energy consumption for different scenarios. As the algorithms do not increase the total available capacity of the access network, they have the potential to improve the situation for a subset of the users and, thereby, the average perceived quality and energy consumption of all users. Future work will apply models from queueing theory for sequential and parallel processing. This would enable a deep evaluation of different scenarios and allow to explore the impact of varying input parameters like file size distributions, inter arrival times and loads on the presented system.

Acknowledgments

We thank the anonymous reviewers for their feedback that helped improve the final version of the paper.

References

[1] Nvidia White Paper, "The benefits of quad core CPUs in mobile devices," Tech. Rep., 2011.

[2] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proceedings of the 2010 USENIX conference on USENIX annual technical conference*, 2010, pp. 21–21.

[3] C. Han, T. Harrold, S. Armour, I. Krikidis, S. Videv, P. M. Grant, H. Haas, J. S. Thompson, I. Ku, C.-X. Wang *et al.*, "Green radio: radio techniques to enable energy-efficient wireless networks," *Communications Magazine, IEEE*, vol. 49, no. 6, pp. 46–54, 2011.

[4] J.-H. Yeh, J.-C. Chen, and C.-C. Lee, "Comparative analysis of energy-saving techniques in 3GPP and 3GPP2 systems," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 1, pp. 432–448, 2009.

[5] G. S. Kim, Y. H. Je, and S. Kim, "An adjustable power management for optimal power saving in LTE terminal baseband modem," *Consumer Electronics, IEEE Transactions on*, vol. 55, no. 4, pp. 1847–1853, 2009.

[6] C. Bontu and E. Illidge, "DRX mechanism for power saving in LTE," *Communications Magazine, IEEE*, vol. 47, no. 6, pp. 48–55, 2009.

[7] Nokia, "Active mode DRX," R2-062752, 3GPP, Oct. 2006.

[8] D. Ferling, T. Bitzer, T. Bohn, D. Wiegner, and A. Pascht, "Power efficient transceivers to enable energy-efficient mobile radio systems," *Bell Labs Technical Journal*, vol. 15, no. 2, pp. 59–76, 2010.

[9] Z. Li, C. Yin, and G. Yue, "Delay-bounded power-efficient packet scheduling for uplink systems of LTE," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on*. IEEE, 2009, pp. 1–4.

[10] C. Han, K. C. Beh, M. Nicolaou, S. Armour, and A. Doufexi, "Power efficient dynamic resource scheduling algorithms for LTE," in *VTC 2010-Fall, 2010 IEEE 72nd*. IEEE, 2010, pp. 1–5.

[11] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: how much can WiFi deliver?" in *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4. ACM, 2010, pp. 425–426.

[12] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4G LTE networks," in *conf. on Mobile systems, and services*. ACM, 2012.

[13] L. Schrage, "A proof of the optimality of the shortest remaining processing time discipline," *Operations Research*, vol. 16, no. 3, pp. 687–690, 1968.

[14] Ericsson White Paper, "Traffic inspection for visibility, control and new business opportunities," Tech. Rep., Sep. 2010. [Online]. Available: http://www.ericsson.com

[15] 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, Jun. 2012.

[16] S.-B. Lee, I. Pefkianakis, A. Meyerson, S. Xu, and S. Lu, "Proportional fair frequency-domain packet scheduling for 3GPP LTE uplink," in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 2611–2615.

[17] 3GPP, "System Enhancements for User Plane Congestion Management," *3GPP, Tech. Rep. TS 23.705 Version 0.6.0*, 2013.