Institute of Communication Networks and Computer Engineering
University of Stuttgart
Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Phone: ++49-711-685-68026, Fax: ++49-711-685-67983
email: mail@ikr.uni-stuttgart.de, http://www.ikr.uni-stuttgart.de

Improving the Quality of Experience with Size-Based and Opportunistic Scheduling

Magnus Proebster
Universität Stuttgart, Institute of Communication Networks & Computer Engineering, Stuttgart, Germany
Email: magnus.proebster@ikr.uni-stuttgart.de

***Abstract*—In cellular networks, the Shortest Remaining Processing Time first (SRPT) principle, which is known to be optimal for sharing jobs on a single-server system, can deliver significant advantages over conventional scheduling algorithms. Transmissions are finished earlier, which improves user experience and enables operators to allow more users in their networks. However, this comes at the cost of reduced cell throughput and a penalty for large objects in peak traffic situations. We therefore propose an extension to SRPT that combines it with opportunistic scheduling and allows to trade off the advantages of both sides. We evaluate and compare the performance of SRPT and conventional schedulers by simulation with a realistic traffic scenario. An important metric is the transmission duration of application layer objects as it is central for the users' Quality of Experience (QoE). The results show that the proposed scheduler provides the superior QoE of SRPT for short, interactive transmissions while keeping the rate reduction for larger traffic objects at an acceptable level.**

I. Introduction

Modern smartphones offer a rich set of Internet services to their users. The rapid growth of the traffic demand of mobile users brings cellular networks to their limits. Often, peak load situations occur, where many User Equipments (UEs) compete for the radio resources of a base station. While modern Orthogonal Frequency-Division Multiple Access (OFDMA) systems like 3GPP Long Term Evolution (LTE) offer a high peak throughput and low latency, a user may still suffer long loading times when many active UEs are in the cell. As base stations (enhanced NodeBs (eNBs) in LTE) are often connected to the Internet via fast optical links, the shared wireless access is the bottleneck in most cases. The loading time is one of the most important metrics for the service quality because the user directly notices it when surfing the Internet, watching videos or using the social web.

In this paper, we evaluate the performance of different scheduling algorithms with respect to the user satisfaction, i.e. QoE, under realistic peak traffic situations by simulation. Emphasis is laid on the SRPT principle, which is known to lead to optimal finish durations in single-server systems with a constant service rate [1]. Also in cellular networks, SRPT can improve the QoE. We propose a new algorithm combining an SRPT algorithm with the conventional *Max C/I* scheduler by introducing a parameter that trades the benefits of both schedulers off. For the evaluation of the users' QoE we will rely on the concept of *transactions* and duration-dependent *utility functions*, which were introduced in [2]. We show that with the new algorithm, a higher load than with the conventional *Proportional Fair (PF)* scheduler can be accepted in a cell at a high QoE level.

Previous work investigated the applicability and advantages of SRPT-scheduling. The basics for SRPT as a queuing model have been laid in [3]. Concerns about SRPT's fairness that are often brought up are addressed in [4]. More recently, the application of the SRPT-principle to cellular networks has been proposed. In [5], the authors propose the *Foreground-Background* scheduler which implicitly prefers short transmissions. [6] proposes different variants for a more explicit implementation of the principle, assuming that the base station knows about the size of the objects to be transmitted. An interesting paper establishing a basis for the trade-off between opportunistic and size-based scheduling is [7]. It defines capacity regions for flow-level scheduling and investigates scheduling algorithms with respect to this trade-off. In [8], the concept of time-scale separation and the optimal trade-off between size-based and opportunistic scheduling are generalized and mathematically analyzed. The authors also derive and evaluate actual schedulers for time-slotted systems from these findings.

However, many of the approaches do not consider or do not focus on the effects of continuous, realistic traffic behavior and of radio channel properties, which are dependent on the user positions and have a correlation in time. Furthermore, it is often difficult to implement the solutions found with flow level abstractions in generic capacity regions for actual slot-level scheduling decisions. It is crucial to apply scheduling decisions, which are robust in the sense that they are still valid when new transmissions arrive in the system and when channel conditions fluctuate. Transient models, where no further traffic arrives over time, do not occur in practical situations. Optimal solutions found with such models not always meet the expectations in a practical system. Therefore, we investigate in this paper the performance of existing SRPT schedulers and our new algorithm and compare them to conventional schedulers in a peak-hour traffic situation. For performance evaluation, we use simulations and focus on the users' QoE.

This paper is structured as follows. Sec. II presents the

Parts of this work were supported by Bell Labs, Alcatel-Lucent within the Project Context-Aware Resource Allocation 2 (CARA 2).

978-1-4799-5863-4/14/$31.00 ©2014 IEEE

reference schedulers and the proposed algorithm. In Sec. III, we introduce the scenario, traffic and channel models, and the evaluation metrics. Sec. IV discusses simulation results showing the performance of the individual scheduling strategies. Finally, the paper is concluded in Sec. V.

II. Scheduling Algorithms

We compare the performance of reference schedulers common in today's cellular networks and based on the SRPT-principle with the proposed enhanced SRPT-scheduler.

As it will be explained in Sec. III, we model traffic as application-layer objects called *transactions*. We assume that the eNB knows the size of these transactions. This cross-layer information is a prerequisite for size-based scheduling. In practice, it could be obtained for example by inspecting the HTTP content-length field or explicit signaling from the application. Furthermore, we assume an ideal channel knowledge for all schedulers.

A. Reference Schedulers

The *Max C/I* scheduler always assigns resources to an arbitrary transaction of the UE with the best channel quality. This strategy maximizes the instantaneously achievable throughput in the cell, as the user with the best Signal to Interference-plus-Noise Ratio (SINR) (also known as Channel/Interference=C/I) has the best spectral efficiency.

PF combines this idea of opportunistic scheduling with fairness. In time-average, it assigns the same amount of resources to each transaction. To do so, *PF* assigns a resource block to the transaction with the largest weight w_{PF}, with

$$w_{PF} = \frac{R(t)}{\overline{R}(t)} \tag{1}$$

where $R(t)$ is the instantaneously possible data rate of the transaction and $\overline{R}(t)$ is the exponential moving average of the data rate. It is updated as follows:

$$\overline{R}(t+1) = \begin{cases} \beta \cdot R(t) + (1-\beta) \cdot \overline{R}(t) & \text{if scheduled} \\ (1-\beta) \cdot \overline{R}(t) & \text{else} \end{cases} \tag{2}$$

Here, β is the so-called forgetting factor controlling the decay rate of earlier values.

Schedulers applying the SRPT-principle in cellular networks were presented in [6] under the term Traffic Aided Opportunistic Scheduling (TAOS). The most straight-forward implementation of SRPT, called *Shortest First (SF)* in the following, schedules the transaction with the smallest cost c_{SF}, with

$$c_{SF} = \frac{X(t)}{R(t)} \tag{3}$$

where $X(t)$ is the remaining size of the transaction. The cost c_{SF} represents the remaining transmission duration estimated with the current data rate, when the transaction would be scheduled all the time. In [6], this scheduler is called *TAOS1b*. Another variant is *TAOS2*, which aims to find the locally optimal solution with respect to the sum completion time of the currently active transactions [6]. The algorithm *TAOS2* works as follows:

Table I
System model parameters

Property	Value
Cellular layout	Hexagonal, 7 sites, wrap-around at the borders
UEs per cell	50
Inter BS distance	1 km
BS/UE height	32 m / 1.5 m
Carrier frequency	2 GHz
System bandwidth	10 MHz
BS TX power	46 dBm
Antenna model	Isotropic
Path loss	$128.1 + 37.6 \log_{10}(d)$, distance d in km [9]
Shadowing	8 dB log-normal, correlation distance 50 m
Multipath propagation	Rayleigh fading with Jakes-like temporal correlation [10], Vehicular A channel taps [11]
UE velocity	10 km/h
Mobility Model	Random walk; mean walk duration 30 s
Slot duration	1 ms
Frequency granularity	180 kHz (50 resource blocks)
Link adaptation	Ideal (Shannon-Hartley, SINR clipped at 25 dB)

1) Enumerate active transactions in ascending order of $\frac{X(t)}{\overline{R^*}(t)}$ with rank i. In contrast to [6], we use the moving average $\overline{R^*}(t)$ of the channel quality instead of the expectation. It is determined by always inserting the current channel in (2) (the upper case). Thus, we use the same information basis for both schedulers.
2) Compute the costs

$$c_{TAOS2} = (i-1) - (M(t) - i + 1)\left(\frac{R(t)}{\overline{R^*}(t)} - 1\right) \tag{4}$$

where $M(t)$ denotes the number of currently active transactions. Basically, the cost increases with the rank of the transaction and when the channel is below average. This means that a short transaction with a channel above average has a small cost.

3) Schedule the transaction with the smallest cost c_{TAOS2}.

B. Combination of SRPT with Max C/I

We propose an enhancement of the *SF* scheduler to control the influence of the channel state on the scheduling decision. This is achieved by adding a parameter to (3):

$$c_{SF} = \frac{X(t)^{\alpha}}{R(t)} \tag{5}$$

where α is the so called *length exponent*. By choosing $\alpha \in [0, 1]$, we can gradually vary the behavior between that of SF ($\alpha = 1$) and that of *Max C/I* ($\alpha = 0$).

With this, it is possible to choose how much emphasis should be laid on cell throughput versus trying to achieve short transmission durations. For $\alpha > 0$ the scheduler is able to consider boundaries of application layer objects and to use this information for a reduction of transmission durations. In Sec. IV we show how this new parameter influences the scheduler performance.

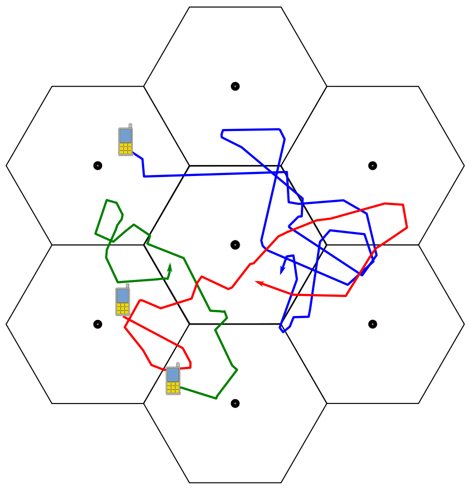

Figure 1. Cellular network layout; 7 sites with moving users and wrap-around.

III. Simulation Model

We use a simple cellular scenario with 7 hexagonally arranged eNBs with isotropic antennas as shown in Fig. 1. We apply wrap-around to avoid border effects. The parameter set models the downlink of a base-line LTE system with 10 MHz bandwidth; the parameters are summarized in Tab. I. For a simulation run, 20 independent replications are performed with 600 s of start-up phase and 1200 s of evaluated simulation time. For each replication, we simulate channel traces of 50 UEs that are placed randomly and travel randomly throughout the system (illustrated for 3 UEs in Fig. 1). A UE always connects to the eNB with the strongest signal, considering path loss and shadowing. (Fast fading changes too fast for being considered for handovers.)

We then use these channel traces, capturing the correlation in time for traveling users, to simulate scheduling in a single cell. Thus, we keep the number of distinguishable receiver locations (i.e. UEs) constant over the simulation time. The time granularity of the schedulers refers to LTE slots and the frequency granularity to LTE resource blocks.

All traffic flows from the eNBs to the UEs (downlink direction). We adjust the traffic volume by parameterizing the Inter-Arrival Time (IAT) of traffic objects to represent any load situation. Traffic consists of so-called *transactions* representing application-layer flow objects. All traffic that leads to an observable result for the user, e.g. loading a web page and its embedded objects, belongs to a single transaction. IATs follow a negative exponential distribution with varying mean to adapt the offered traffic. Object sizes represent a mixture of HTTP and FTP traffic derived from [12]. We assume that all objects arrive as a whole at the base station. For HTTP transactions, this means that the main object and embedded objects arrive at the same point in time. According to [12], the main object and embedded object sizes follow log-normal distributions and the number of embedded objects is Pareto-distributed. In total, we get an average object size of ≈50 kBytes for HTTP transactions. FTP transaction sizes follow a truncated log-normal distribution with an average object size of ≈2 MBytes and a maximum of 5 Mbytes. The parameters are summarized in Tab. II.

Table II
Parameters for traffic model distributions [12]

Object type	μ	σ	minimum	maximum
FTP	14.45	0.35	0	5 MBytes
HTTP main obj.	8.37	1.37	100 Bytes	2 MBytes
HTTP embedded obj.	6.17	2.36	50 Bytes	2 MBytes

The QoE, a transaction has for the user, depends on the transmission duration of this transaction. In the example of web browsing, users are annoyed if it takes too long until a new page appears on the screen after they clicked on a link (see also [13]). Therefore, we apply *utility functions* to assess the satisfaction of users with their connections. We only consider the delay introduced by the MAC-layer, as we want to investigate the influence of the scheduler. Latency introduced by the upper or lower layers is not considered. The utility functions are S-shaped (logistic functions) and strictly monotonically decreasing over the duration the transaction requires to transmit. They are normalized to the interval [0, 1], where 1 is the best utility (instantaneous transmission) and 0 the worst (aborted transmission). We assume that users expect a certain duration t_{exp} for the transmission of a transaction, modeled as an expected bandwidth b_{exp}. With a minimum duration $t_{\text{exp,min}}$ recognizable by a user, we get for a transaction of size L: $t_{\text{exp}} = \max(L/b_{\text{exp}}, t_{\text{exp,min}})$. The expected duration depends on the application type. E.g., users accept longer durations for background tasks. Furthermore, we assume that users eventually abort their transmissions when they take too long. This threshold t_{drop} depends on the expected bandwidth and has the minimum $t_{\text{drop,min}}$, which gives us $t_{\text{drop}} = \max(10 \cdot t_{\text{exp}}, t_{\text{drop,min}})$. A completed transaction is accounted for with a minimum utility u_{min}, whereas a dropped transaction has a utility of 0. We get for the utility with duration t

$$U(t) = \begin{cases} u_{\text{min}} + \frac{1-u_{\text{min}}}{1+e^{s(t-x\cdot t_{\text{exp}})}} & \text{for } t < t_{\text{drop}} \\ 0 & \text{for } t \geq t_{\text{drop}} \end{cases} \tag{6}$$

with $s = \frac{1}{t_{\text{exp}}(1-x)} \ln\left(\frac{1-u_{\text{min}}}{u_{\text{exp}}-u_{\text{min}}} - 1\right)$, and the parameters in Tab. III (additional information on the parameterization in [2]).

This choice of utility parameters is an example to compare the QoE performance of different schedulers. It is derived from findings in [13] and [14]. However, our findings do not depend on the exact choice of these parameters. Instead, the relative difference in the behavior of the schedulers is of importance. Furthermore, the metric of transaction durations is independent of the choice of the utility functions.

IV. Performance Evaluation

We investigate the performance of the schedulers by simulation. The metrics that are of interest are utility, throughput and the durations of the transactions. Furthermore, we investigate

Table III
PARAMETERS OF EXEMPLARY UTILITY FUNCTIONS, ADAPTED FROM [2]

Parameter	Description	HTTP	FTP
b_{exp}	Expected bandwidth	3 Mbit/s	1.5 Mbit/s
u_{exp}	Utility for exp. bandwidth	0.955	0.955
u_{min}	Min utility of finished trans.	0.1	0.1
x	Inflection point factor	5.4462	5.7799
$t_{\text{exp,min}}$	Min exp. duration	0.1 s	0.1 s
$t_{\text{drop,min}}$	Min drop duration	10 s	10 s

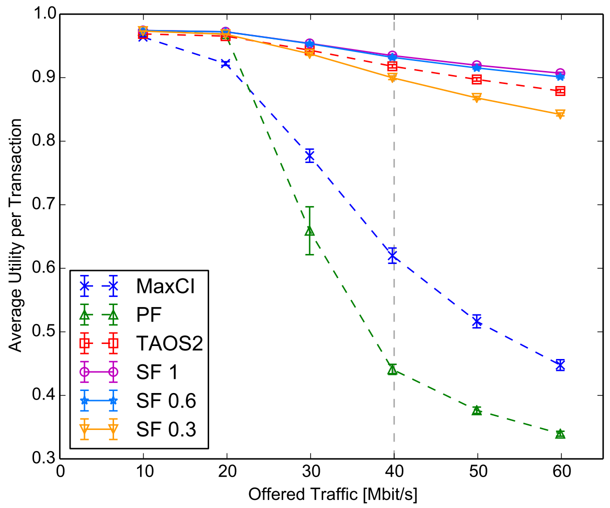

Figure 2. Comparison of the average utility over the offered traffic.

how many transactions are completed by the schedulers and the influence of the object size.

Traffic is composed of 90% HTTP transactions and 10% FTP transactions, which makes up for 20% and 80% of the traffic volume, respectively. This reflects the observation that most traffic objects in the Internet are very small but most volume is represented in relatively few large objects. Different combinations of the traffic mix have been evaluated but do not change the presented findings qualitatively. Therefore, they are omitted for clarity. We choose the forgetting factor $\beta = 0.01$ for *PF* and *TAOS2*.

A. Dependence on the Cell Load

First, we look at the scheduling behavior with varying load. Fig. 2 shows the average utility per transaction and Fig. 3 the sum throughput of the cell over the rate of offered traffic. The performance of the reference schedulers is compared against *SF*. For *SF*, the figures show results for length exponents of $\alpha = 1$ (the original *SF*), 0.6, and 0.3. Confidence intervals have a 95% confidence level.

For low offered load, all schedulers provide a good utility because the resources are not fully utilized and all transactions are transmitted as fast as possible, no matter which scheduling algorithm is in place. With increasing load, the average utility of *PF* drops sharply. Due to its fairness constraint, *PF* interleaves the transmission of different transactions. This increases the durations of all transactions and decreases their utilities. Also *Max C/I* has a bad utility at high load. It does not care

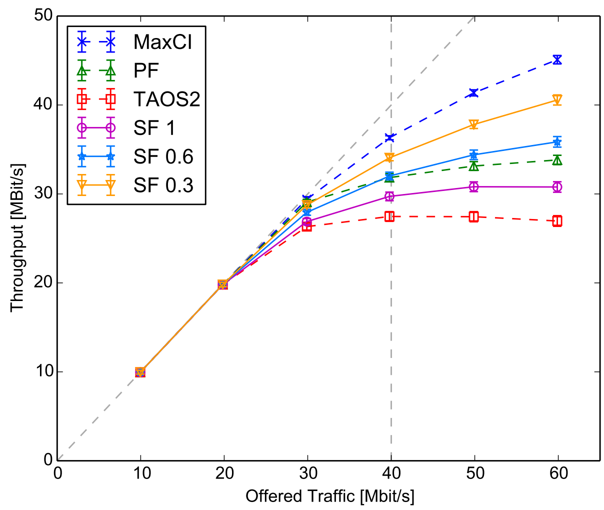

Figure 3. Comparison of the cell throughput over the offered traffic.

for finish durations, but its superior throughput performance keeps durations, especially for large transactions, smaller in average than with *PF*.

By focusing on short durations, all SRPT-disciplines offer a high and only slowly decreasing utility with higher load. In Fig. 2, this main advantage of SRPT is visible. The average utility remains high although the system bandwidth is overloaded. That is because small objects, which make up most transactions, finish as fast as they would in the low load case. Only large objects suffer from the overload situation, which we demonstrate in the following sections. In conclusion, an operator can serve more satisfied users (with high QoE) without extending network resources compared to conventional schedulers like *PF* and *Max C/I*.

Among the SRPT-disciplines, *SF* offers the best average utility. *TAOS2*, which, according to the authors in [6], finds the locally optimal solution, has drawbacks when new traffic arrives continuously. Here, with continuous traffic arrivals, *SF* performs better. With increasing length exponent, the utility of *SF* slightly decreases. However, for $\alpha = 0.6$, the average utility is almost the same as for the original *SF*. When going further to $\alpha = 0.3$, utility drops below *TAOS2* but is still far better than for *PF* and *Max C/I*.

Fig. 3 shows the cell throughput in dependence of the offered traffic. The bisecting line (offered traffic = cell throughput) is the upper bound, which means that all transactions could be completed. For increasing offered traffic, the cell throughput deviates from this line, because radio resources are fully occupied and not all transactions can be served. Still, cell throughput grows with increasing offered traffic as it gets more likely that a UE with good channel conditions has something to transmit. Eventually, cell throughput will saturate at the full buffer performance of the respective scheduler.

As expected, *Max C/I* offers the best throughput. It only focuses on channel conditions and maximizes spectral efficiency in each time instant. As known from literature, *PF* sacrifices

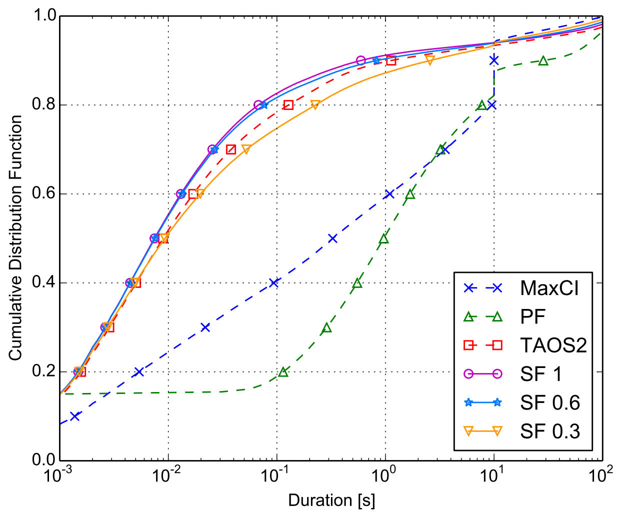

Figure 4. Cumulative distribution function of transaction durations (40 Mbit/s offered traffic).

throughput performance in favor of fairness. The original SRPT-schedulers (*TAOS2* and *SF 1*) have a limited throughput saturating at around 30 Mbit/s. Although the channel state is contained in their duration estimation, the object size has a larger influence and often overrules the channel state.

The newly introduced α-parameter significantly increases the throughput performance of *SF*. For $\alpha = 0.6$, *SF* outperforms *PF* at high load and approaches the performance of *Max C/I* for $\alpha = 0.3$.

In the following, to investigate the reasons for the utility performance, we pick a fixed offered load of 40 Mbit/s (dashed vertical line in Fig. 2 and Fig. 3). At this operation point, the SRPT-schedulers still offer a reasonable average QoE and all radio resources are occupied; i.e., we have an overload situation.

B. Transaction Durations

In Sec. III, we derived the utility metric from transaction durations. As already mentioned, we only consider MAC delay, i.e. queuing and transmission times. Fig. 4 shows the Cumulative Distribution Function (CDF) of the transaction durations for the different schedulers on a logarithmic scale (40 Mbits/s offered traffic; includes finished and dropped transactions). Except for *Max C/I*, the curves start at 15%. This is the fraction of transactions finishing within a single time slot. With SRPT-schedulers, transactions that are able to finish get scheduled immediately. For *PF*, new transactions get a "jump start", because the moving average is initialized with zero. The steps in the CDFs of *PF* and *Max C/I* at 10 s are short transactions dropped after the minimum dropping duration.

Fig. 4 illustrates the reason for the different utility performance of the schedulers. The CDF of *PF* lies completely below those of the SRPT-disciplines. That means that transactions of all sizes last much longer with *PF*. *PF* is unaware of object boundaries and interleaves allocations between UEs and transactions. With *Max C/I*, the top 5% of the long-lasting transmissions finish faster than for the SRPT-disciplines due to the higher cell throughput.

Comparing between the SRPT-schedulers shows that *TAOS2* has slightly longer finish durations than *SF 1* over the whole range. With decreasing α, the short object durations up to the 90%-ile increase. At the same time, the longer lasting objects profit, like in the case of *Max C/I*, from the improved cell throughput and finish faster.

C. Fairness between Transactions of Different Sizes

It is often stated that SRPT improves small objects' durations at the cost of larger objects. While this is adressed analytically in [4], we want to demonstrate the influence of transaction size on the scheduling decision in this practical situation. Especially the influence of the newly proposed length exponent α shall be investigated.

Fig. 4 shows that SRPT-disciplines improve transmission durations for *all* objects, compared to *PF*. However, with *Max C/I*, long-lasting (i.e. large) objects would finish faster. We investigate the influence of the transaction size with the histogram in Fig. 5. Bins are chosen equidistantly on a logarithmic scale and normalized to a sum of 1. The gray bars in the background represent the total number of transactions. The two peaks from the log-normal distributions of HTTP and FTP transaction sizes are clearly visible. The curves for the different schedulers show how many transactions were completed within the respective bin. The difference between the curves and the bars is the number of dropped transactions. Please note that the histogram does not represent the traffic volume but instead the transaction count for exponentially growing bin sizes. As stated earlier, about 80% of the traffic volume are contained in the right peak of the FTP traffic.

We can see from Fig. 5 that most small objects finish with *PF*, but many larger transactions get dropped. *Max C/I*, on the other side, finishes most large transactions among all schedulers, but many small objects get dropped. The superior QoE of the SRPT-disciplines is also visible in Fig. 5. Practically none of the small transactions get dropped, while also the major part of the large transactions finish.

In Fig. 6, we take a closer look at transaction utilities. It shows the average utility in bins equal to those of the histogram in Fig. 5. *PF* offers a good utility to objects, which profit from the jump start mentioned earlier. However, for larger objects, utility drops sharply. It gets marginally better again for FTP objects at sizes around several MBytes, because they are not so time-critical. As seen before, *Max C/I* is a bad option for small transactions and offers the best utility for the large ones.

Fig. 6 shows the penalty for large transactions under the original SRPT-disciplines. While we learned from Fig. 5 that many of the largest objects finish, they require long transmission times and only get a poor average utility. Trading off the utility of small and large objects can be directly controlled with our proposed α-parameter. Choosing small values like $\alpha = 0.3$ greatly improves the average utility of large transactions at the cost of the medium sized ones.

Table IV
UTILITIES FOR DIFFERENT CHOICES OF THE LENGTH EXPONENT.

Length exponent	0.1	0.3	0.5	0.6	0.7	1
Total utility	0.739	0.900	0.928	0.932	0.934	0.935
FTP utility	0.754	0.709	0.670	0.656	0.644	0.619
HTTP utility	0.737	0.922	0.958	0.964	0.967	0.971

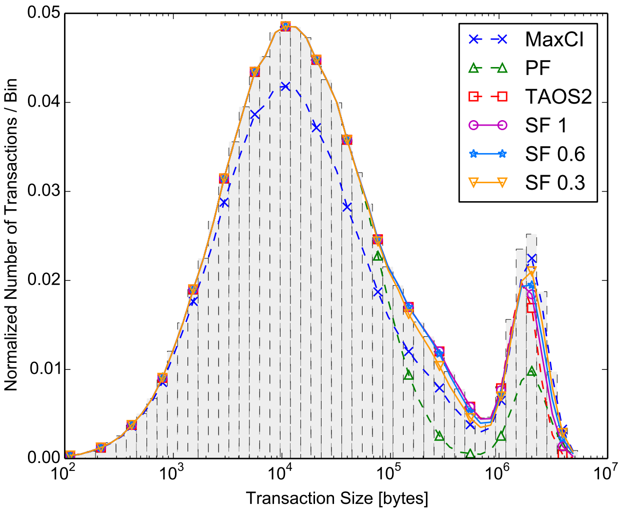

Figure 5. Histogram of finished transactions over size on a logarithmic scale (40 Mbit/s offered traffic).

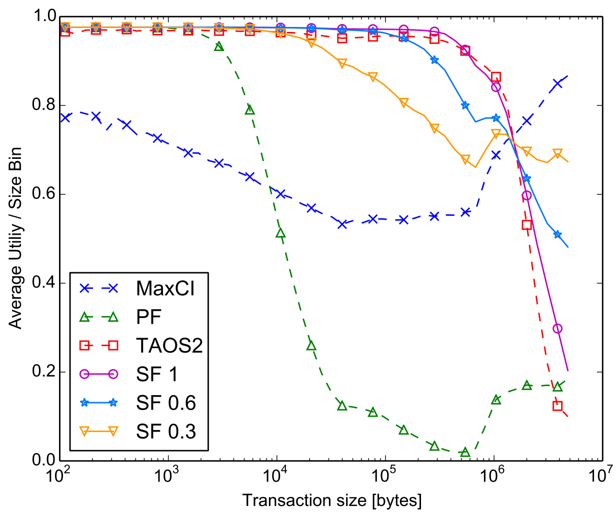

Figure 6. Average utility per bin of the logarithmic size histogram in Fig. 5 (40 Mbit/s offered traffic).

These findings are backed by Tab. IV. It compares total, HTTP and FTP average utilities for different choices of α. Total utility increases with α and saturates at $\alpha \approx 0.5$. The utility of FTP traffic, representing the larger objects, decreases as expected with increasing α. On the other side, HTTP utility (contributing 90% to the total utility) increases with α and saturates at $\alpha \approx 0.6$. For the simulated traffic mix, we propose setting α to 0.6 as HTTP utility cannot gain much from increasing it further while FTP would loose utility.

V. CONCLUSION

We proposed a new scheduler that combines SRPT with classical opportunistic scheduling. By introducing a parameter to the *Shortest First* scheduler it is now possible to tune continuously towards *Max C/I*. We evaluated the proposed and reference schedulers in a cellular network by simulation, with a focus on QoE. The QoE is expressed in terms of utility functions depending on the transmission duration of application layer traffic objects (*transactions*). A user has a satisfactory service experience when transactions finish quickly and a bad experience when interactions result in long waiting times

The results show the large advantage of SRPT-disciplines with respect to QoE in overload situations. Where conventional schedulers like *Proportional Fair* or *Max C/I* offer bad service to all applications because they interleave transmissions, SRPT-disciplines finish especially small objects fast by scheduling one after the other. With this, the majority of transactions profits in a typical traffic scenario and more users can be allowed in the cell at a satisfactory service quality.

Our proposed length exponent α allows to improve the throughput of *Shortest First* by increasing the weight of the channel state in the scheduling decision. With this, an operator can trade the interactive service quality for cell throughput.

REFERENCES

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

[2] M. Proebster, M. Kaschub, T. Werthmann, and S. Valentin, "Context-aware resource allocation for cellular wireless networks," *EURASIP Journal on Wireless Communications and Networking*, no. 1, p. 216, 2012.

[3] L. E. Schrage and L. W. Miller, "The queue m/g/1 with the shortest remaining processing time discipline," *Operations Research*, vol. 14, no. 4, pp. 670–684, 1966.

[4] N. Bansal and M. Harchol-Balter, "Analysis of srpt scheduling: investigating unfairness," *SIGMETRICS Perform. Eval. Rev.*, vol. 29, no. 1, pp. 279–290, Jun. 2001.

[5] Z. Shao and U. Madhow, "A qos framework for heavy-tailed traffic over the wireless internet," in *MILCOM 2002. Proceedings*, vol. 2, Oct. 2002, pp. 1201 – 1205.

[6] M. Hu, J. Zhang, and J. Sadowsky, "Traffic aided opportunistic scheduling for wireless networks: algorithms and performance bounds," *Computer Networks*, vol. 46, no. 4, pp. 505 – 518, 2004.

[7] B. Sadiq and G. De Veciana, "Balancing srpt prioritization vs opportunistic gain in wireless systems with flow dynamics," in *Teletraffic Congress (ITC), 2010 22nd International*, Sep. 2010, pp. 1–8.

[8] S. Aalto, A. Penttinen, P. Lassila, and P. Osti, "Optimal size-based opportunistic scheduler for wireless systems," *Queueing Systems*, vol. 72, pp. 5–30, 2012.

[9] *Physical layer aspects for evaolved Universal Terrestrial Radio Access (UTRA)*, 3GPP Std. TR 25.814, Rev. V7.1.0, Sep. 2006.

[10] P. Dent, G. Bottomley, and T. Croft, "Jakes fading model revisited," *Electronics Letters*, vol. 29, no. 13, pp. 1162–1163, Jun. 1993.

[11] *Selection procedures for the choice of radio transmission technologies of the UMTS*, ETSI Std. TR101 112/UMTS30.03 V3.2, Apr. 1998.

[12] R. Irmer (ed.), "Radio access performance evaluation methodology," NGMN White Paper V1.3, Jan. 2008.

[13] J. Nielsen, "Website response times," http://www.nngroup.com/articles/website-response-times/, Jun. 2010, link verified on May 1st, 2014.

[14] S. Niida, S. Uemura, and H. Nakamura, "Mobile services," *IEEE Vehicular Technology Magazine*, vol. 5, no. 3, pp. 61–67, Sep. 2010.