Comparison of Opportunistic Scheduling Algorithms for HSDPA Networks

Frank Feller and Marc C. Necker
Institute of Communication Networks and Computer Engineering,
University of Stuttgart, Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Email: frank.feller@gmx.de, necker@ikr.uni-stuttgart.de

Abstract— **The conditions for scheduling multiple users on a shared wireless channel strongly differ from those in wireline systems. The time-variant nature of the radio link in combination with adaptive modulation and coding schemes used in modern wireless systems gives way to a variety of resource assignment strategies. In particular, channel-aware scheduling algorithms increase the system performance by serving users when it is most favorable with respect to their present channel quality. In this paper, we compare several different channel-aware schedulers with a set of basic channel-independent reference algorithms at the example of a state-of-the-art HSDPA system. We discuss their characteristics and evaluate their performance in a single-service and a multi-service scenario.**

I. Introduction

In the past years, WCDMA networks based on the UMTS standard have widely been deployed. Traditionally, these systems provide voice and data services via circuit-switched channels. With the evolution of 3G systems, UMTS has been extended by High Speed Downlink Packet Access (HSDPA). HSDPA provides a packet-switched downlink channel with increased data rates of up to 14 MBit/s. It introduces an additional functional layer in the protocol stack, namely the MAC-hs layer. The MAC-hs functionality is implemented in the Node B, which is located directly at the air interface and therefore allows a fast reaction on errors and variations of the channel quality. This includes adaptive modulation and coding schemes as well as a powerful Hybrid Automatic Repeat Request (HARQ) mechanism [1]. Additionally, fast scheduling algorithms can be implemented in the Node B, which can significantly increase the system capacity by exploiting multi-user diversity.

Shared channels, such as the broadband downlink channel of an HSDPA system, require a scheduler which assigns transmission resources to the individual users. In time-invariant wireline systems, it is generally sufficient to base the scheduling decision on the assignment of just one kind of resource, e. g. the transmission time. This assignment then automatically results in proportional transmission rates. State-of-the-art wireless systems like HSDPA, in contrast, employ adaptive modulation and coding, leading to a variable data rate over time to each user. A scheduler fairly distributing the transmission time will therefore not assure equal transmission rates. Moreover, the transmission power is adapted to the current channel quality, turning it into an assignable resource. Hence, a scheduling algorithm may base its scheduling decision on a set of three different transmission resources, giving way to disciplines trying to provide an equal amount of transmission time, transmission power, or equal transmission rates, respectively, to all terminals.

Opportunistic schedulers, also known as channel-aware schedulers, exploit the time-variant nature of the radio channel in order to increase the system capacity. They include frameworks based on a good-bad channel model that avoid wasting resources by preventing a user experiencing an error-prone channel from being served. Such frameworks, discussed in [2], for instance, apply wireline schedulers to the users with a good channel. Other disciplines directly conceived for wireless links assume a continuous channel quality metric. By serving terminals with the currently best channel conditions, they realize significant scheduling gains by efficient exploitation of multi-user diversity.

The comparison of channel-aware algorithms with channel-independent or opportunistic reference schedulers is often done in simplified or specific scenarios highlighting the advantages of the respective algorithm. Moreover, several versions of the well-known *Proportional Fair* algorithm are used. Hence, the results of these evaluations are not comparable. Therefore, a performance evaluation of existing opportunistic disciplines in a unified scenario focusing on the basic differences of certain scheduler classes is of great interest.

In this paper, we address this issue and systematically compare a number of native channel-aware schedulers and two channel-independent reference algorithms at the example of a state-of-the-art HSDPA system and two different traffic scenarios. The first, homogeneous scenario allows for a comparison of the absolute performance gains – in terms of throughput and delay – realized by the opportunistic schedulers. The second, more realistic scenario includes three different traffic classes with particular quality of service requirements. For each of them, a representative metric is considered in order to show both the appropriateness of the algorithms for particular traffic classes and their potential preference of certain traffic characteristics.

This paper is structured as follows. After the introduction of the considered system and traffic scenario in section II, we give an overview of basic scheduling algorithms for wireless systems in section III. We evaluate their performance in section IV and conclude our paper in section V.

II. HSDPA System Model

A. System Overview and Simulation Model

Our scenario is shown in Fig. 1. We consider a single-cell environment, where several User Equipments (UEs) connect to the Node B via a High Speed Downlink Shared Channel (HS-DSCH) in the downlink and a dedicated channel (DCH) in the uplink. The Node B is connected to the Radio Network Controller (RNC), which itself is connected to the Internet via the 3G-SGSN and 3G-GGSN of the cellular system's core network. The UEs establish a data connection with a host in the Internet. The Internet and core network were assumed to introduce a constant delay $T_{\mathrm{INet}} = 20$ ms in each direction and not lose any packets.

All simulations were performed using an event-driven simulation tool, which was implemented using the IKR SimLib [3]. The HSDPA network model is shown in Fig. 2. It comprises all relevant RLC, MAC-d and MAC-hs protocols. The physical layer was modeled including the HARQ, based on BLER-curves obtained from physical layer simulations. Transport formats (TF) on the MAC-hs layer were selected based on the channel quality such that the BLER is 10%. The physical channels towards all mobiles were modeled with a variable path loss, including slow fading and fast fading. We assumed ideal conditions for the reporting of Channel Quality Indicators (CQI) from the UEs to the Node B, i.e. zero delay and error-free feedback, in order to isolate the performance influence of the scheduling mechanisms. Alike, the Iub flow control between the RNC and the Node B was assumed to operate with no dead time and short update periods, since non-optimal values can cause significant and unpredictable variations in performance metrics [4].

Depending on the QoS class, the RLC layer is configured in RLC Acknowledged Mode (AM) or RLC Unacknowledged Mode (UM). In both cases, the MAC-hs HARQ was activated with a maximum number of MAC-hs retransmissions of 4. In RLC AM, the maximum number of RLC-retransmissions was 10, and the maximum RLC window size was assumed to be unlimited in order to avoid side effects in the results. In all cases, we neglect the convergence layer, as it only introduces a very small overhead in a single-cell environment.

B. Scenario

We consider two different traffic scenarios, namely a homogeneous and a heterogeneous traffic scenario. Both scenarios comprise several UEs with one active data flow each, moving at a velocity of $v = 30$ km/h, which corresponds to the well-known 3GPP-scenario Vehicular 30.

In the homogeneous traffic scenario, six UEs performing an FTP download are considered. The FTP traffic is modeled based on a greedy traffic source in combination with a TCP NewReno sender and a TCP timer granularity of 500 ms. TCP window scaling was activated to not limit the TCP sender's data rate by the transmission window size. For all UEs, the RLC layer is configured in AM.

The heterogeneous traffic scenario is equal to the scenario from [5]. It comprises ten UEs which break down into five UEs running a gaming application, two UEs running a streaming application, and three UEs performing an FTP download. The RLC layer is configured in AM for the gaming and FTP flows, and in UM for the streaming traffic. A timer mechanism deletes all packets in the streaming flows' RLC input queues with a waiting time larger than 2 s.

Voice calls are handled by the circuit switched domain using DCHs. This brings up the problem of resource management between the circuit and packet switched domains, both in terms of interference and transmission power. This issue was studied for example in [6]. In the following, we assume the circuit switched domain to consume a fixed amount of transmission power and produce a fixed amount of interference.

C. Resource Assignment

In each Transmission Time Interval (TTI, corresponding to a MAC frame and a scheduling round), the scheduler in the base station may use up to 8 Watts of transmission power. Per default, 4 Watts will be assigned to each scheduled terminal, and the Transport Format will be selected accordingly. If a terminal is in bad channel conditions, up to 8 Watts may be spent on this terminal. If the terminal experiences a particularly good channel, less than 4 Watts may be used. In each TTI, terminals are scheduled until the complete transmission power is used up.

We separate the overall scheduling process into two phases:

1) Assignment of scheduling tags (i.e., scheduling priorities)

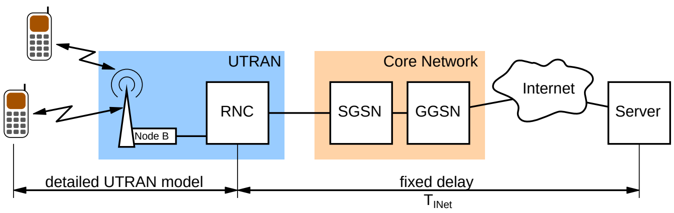

Fig. 1. Architecture of the considered 3G network

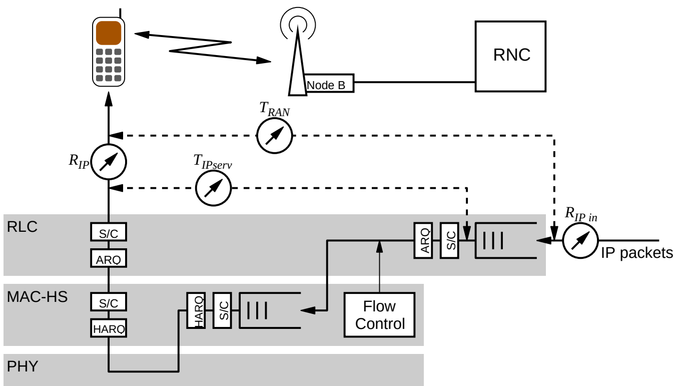

Fig. 2. Illustration of the considered metrics

2) Assignment of air interfaces resources (i.e., codes and transmission power)

During the first phase, the actual scheduling algorithms (e.g., a Round Robin or a Proportional Fair algorithm) assigns scheduling tags to each HS priority queue. These scheduling tags are denoted with tag_k for the kth priority queue. They directly correspond to the scheduling priority of the kth user for the current scheduling round.

In the second phase, a so-called packer uses these scheduling tags in order to assign air interface resources to the different users. In a typical scheduling round, two users will be assigned resources with 4 Watts each.

This procedure has the advantage that the first phase is technology-independent, allowing for an easy substitution and reuse of scheduling algorithms. Only the second phase is technology dependent, taking into account the constraints of limited channelization codes and transmission power.

III. Scheduling in Wireless Systems

A. Introduction

Channel-aware schedulers, also referred to as opportunistic algorithms, take into account the channel quality in order to increase the performance of a wireless system. This may be done in several ways.

Opportunistic schedulers can be characterized by how they abstract the wireless channel. A binary good-bad channel is used by several approaches which extend wireline schedulers to prevent transmission to users in bad channel conditions. Such transmissions are likely to provoke transmission errors, which translate into a waste of resources due to the necessary retransmissions. Terminals omitted for being in bad channel conditions may be compensated later. Such approaches with only a limited channel-dependent contribution and requiring the definition of a threshold are not subject to evaluation in this paper.

In contrast, a continuous channel quality indicator is used by most algorithms conceived explicitly for wireless links. One example is the *Maximum C/I* scheduler, which serves the terminal with the absolutely best channel. It maximizes the system throughput, but suffers from an inherent unfairness. All other schedulers detailed below relate the current channel quality to some average value for the respective terminal. This average value may be the average of the realized data rate in the case of the *Proportional Fair* (PF) algorithm, or the rank among previous channel qualities used by the *Score Based* scheduler [7].

The scheduling priorities computed by opportunistic algorithms may arithmetically be combined with those of other schedulers. This enables the combination of other properties, such as QoS awareness, with the exploitation of multi-user diversity. For instance, such a combination of the PF and the *Earliest Due Date* schedulers provides certain quality of service guarantees [8]. Except for the *TCP – Proportional Fair* algorithm, which is likewise an arithmetic extension of the PF scheduler, combined approaches will not be discussed here.

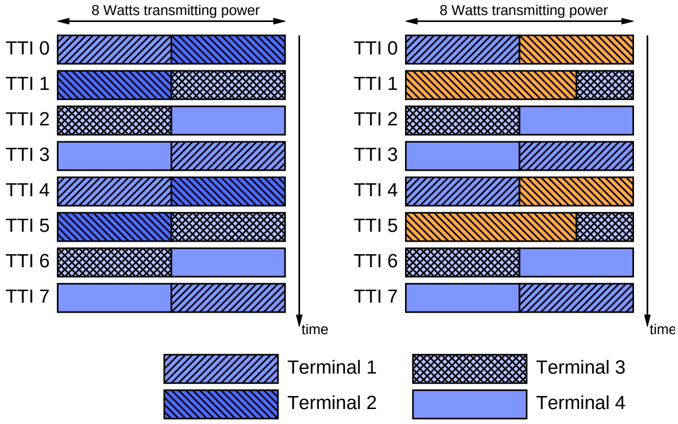

Fig. 3. Illustration of power assignment to different terminals during two scheduling rounds. Left: equal channel conditions for all terminals, Right: Bad channel condition for terminal 2.

B. Channel-Independent Scheduling Algorithms

In this section, we investigate the well-known RR scheduler, which distributes the transmission time equally among all terminals. We additionally present a variant we deduced that provides an equal amount of transmission power to all terminals. An approach equalizing the transmission rates will not be discussed in this paper, since it will inherently reduce the overall system performance compared to the traditional fair distribution of transmission times.

1) Round Robin (RR): The Round Robin scheduler periodically serves all terminals in a cyclic manner. It is the prototype of an algorithm equally distributing the transmission times among all terminals. In a homogeneous setup with constant data rate demands, it is expected to provide the best timing performance in terms of a relatively low and constant delay. This behavior results from the deterministic and constant inter-scheduling times (ISTs) of the basic algorithm.

On the downside, the RR scheduler has an inherent fairness problem in the investigated scenario, which directly results from the time-variant channel behavior. Figure 3 illustrates two possible cases of the resource assignment process during eight scheduling rounds (TTI 0 through TTI 7) with four terminals. Shown is the distribution of the basestation's transmission power to the different terminals for each scheduling round.

As already described in section II-C, two terminals will be scheduled with 4 Watts each in a typical scheduling round. In the left half of Fig. 3, the case of identical channel conditions for all terminals is sketched. In TTI 0, terminals 0 and 1 will be scheduled with equal power. According to the RR principle, terminals 1 and 2 will be scheduled in TTI 1, and so on. In this example, all terminals get a similar share of the total available transmission power, and consequently a fair share of the available bandwidth due to the identical channel conditions.

In the right half of Fig. 3, it is assumed that terminal 2 experiences a worse channel than the other terminals. This possibly leads to a larger assigned power, as seen in TTIs

1 and 5, and thus transmission power is taken away from terminal 3. Eventually, it leads to an unfair behavior towards those terminals which come directly after a terminal with a bad channel in the scheduling order.

Several possibilities are conceivable in order to overcome this unfairness. One possibility is to randomly schedule all terminals while maintaining the time share of all terminals in the RR case. In the following section, we will present a modification of the RR scheduler, which takes into account the transmission power allocated to each terminal.

2) Power Based Round Robin (PBRR): The Power Based Round Robin scheduler is an adaption of the RR principle aiming at a fair distribution of the transmission power. For this purpose, it sums up the transmission power attributed to each terminal and serves the one that has currently received the smallest share of power. The power counters are reset to 0 after n TTIs, where n is the number of active terminals. This measure both allows for a simple reintegration of temporarily idle terminals and assures a RR-like behavior under symmetrical conditions: Terminals having received the same amount of transmission power are ordered deterministically in the way a RR scheduler would do it. At the beginning of a scheduling cycle, when the counters were reset, all terminals are therefore served once in a fixed sequence, and since they receive equal power shares under ideally symmetrical conditions, this cycle will repeat, resulting in a behavior even closer to the theoretical RR algorithm than achieved by the RR scheduler described above.

Due to the compensating effect of the preferential service to terminals that received the smallest share of power, the transmission power is approximately distributed equally. An increase of the overall system throughput is to be expected, since a terminal in bad channel conditions will use up its power share within one TTI, whereas a terminal in favorable conditions realizing a high data rate with little transmission power may be served several times within one scheduling cycle. On the downside, the timing performance will slightly worsen since the ISTs become random. However, they are still deterministically bound by the duration of two scheduling cycles.

C. Channel-Aware Scheduling Algorithms

In the following paragraphs, we outline a number of basic opportunistic scheduling schemes. In particular, we will consider the generic approaches Maximum C/I, Proportional Fair (including variants hereof) and Score Based, which will be followed by the TCP – Proportional Fair discipline designed for TCP traffic.

1) Maximum C/I: The Maximum C/I scheduler, also known as SNR-based scheduler, is the most simple channel-aware scheduler. It bases its scheduling decision on the absolute instantaneous channel quality reported by each UE in each scheduling round. The main disadvantage of this approach is the inherent unfairness on small time-scales. Although it maximizes the cumulative system throughput, it will therefore not be subject to further consideration.

2) Proportional Fair (PF): The PF scheduler overcomes this problem by basing its scheduling decision on the ratio of the currently achievable data rate and the averaged data rate of the recent past. Without explicit differentiation, several versions of the algorithm have been described in literature. The basic one [9], referred to as the *standard* variant in this paper, calculates the scheduling tag as:

$$tag_k = \frac{R_k(t)}{\bar{R}_k(t)} \quad , \tag{1}$$

where $R_k(t)$ is the estimate of the data rate achievable for flow k at time instant t. $\bar{R}_k(t)$ is the average data rate updated after the scheduling decision as:

$$\bar{R}_k(t) = \frac{1}{\tau} R'_k(t) + \left(1 - \frac{1}{\tau}\right) \bar{R}_k(t - T_{\mathrm{TTI}}) \quad . \tag{2}$$

$R'_k(t)$ is the actually realized data rate to user k at time t, i. e. $R'_k(t) = 0$ for a user currently not served. T_{TTI} is the length of a scheduling round, i. e., the length of a TTI. τ is a time constant.

This algorithm has a number of drawbacks. The first one concerns the behavior of the average data rate in case of empty transmission buffers. This issue is illustrated in Fig. 4, which depicts the normalized instantaneously possible data rate and the average data rate of one terminal in a scenario of two users. At first, packets are assumed to be waiting in the buffer, and due to regular transmissions the realized average data rate follows the channel quality. At $t = 0.75\,\mathrm{s}$, however, the buffer runs empty, so no further transmissions are possible and the average data rate decreases steadily. As a consequence, the terminal is likely to be served regardless of the channel quality for several TTIs upon the arrival of new packets, which results in a waste of transmission capacity. Due to the same effect, terminals demanding small data rates are also served preferentially.

A waste of resources also characterizes the second drawback: The standard PF algorithm is ignorant to the amount of data in the transmission buffer. Therefore, a terminal scheduled despite a low buffer level may only use up a fraction of the assigned transmission resources when transmitting the entirety of the queued packets. This may partially be compensated by the packer.

The algorithm presented in [10] and referred to as *Kolding's* variant in this paper addresses these issues. It adapts the tag calculation as follows:

$$tag_k = \frac{\min\left\{R_k(t), \frac{B_k(t)}{T_{\mathrm{TTI}}}\right\}}{\bar{R}_k(t)} \quad , \tag{3}$$

where $B_k(t)$ is the transmission buffer level of flow k at time t. The min function is deactivated after a certain waiting time to avoid excessive delays when no further data arrives in the buffer. The update of the average data rate now follows:

$$\bar{R}_k(t) = \frac{1}{\tau} R'_k(t) + \left(1 - \frac{b}{\tau}\right) \bar{R}_k(t - T_{\mathrm{TTI}}) \quad . \tag{4}$$

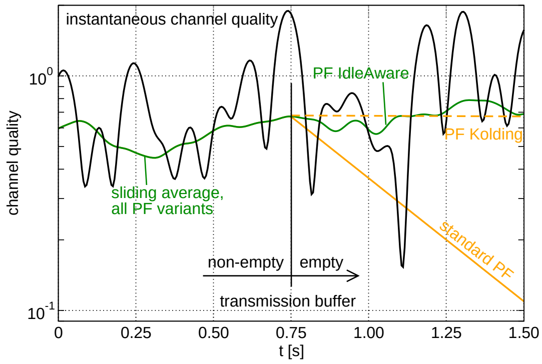

Fig. 4. Illustration of the average data rate for different PF variants

$b = 0$ if the buffer of flow k is empty at time t, otherwise $b = 1$, i.e. the update is skipped in case of an empty buffer avoiding artificially low average values for small data rates or temporary transmission interruptions.

On the downside, the channel information hidden in the average rate will outdate for an idle terminal. As illustrated in Fig. 4, it will stay constant, whereas the instantaneously possible data rate continues to vary, implying an adaption of the mean rate in the case of non-empty buffers. This prevents the scheduler from correctly determining the relatively best channel. In order to overcome this issue, we propose a modification of the update of the average data rate aiming at an imitation of the non-idle curve during idle periods:

$$\begin{aligned} \bar{R}_k(t) \;&=\; \frac{1}{\tau} \cdot \left(bR'_k(t) + (1-b)\alpha \frac{R_k(t)}{n} \right) \\ &\quad + \left(1 - \frac{1}{\tau}\right) \bar{R}_k(t - T_{\mathrm{TTI}}) \;. \end{aligned} \tag{5}$$

n is the number of active terminals. α is a correction factor corresponding to the expected scheduling gain, which usually lies between 1.0 and 1.5. We will refer to the algorithm based on this equation and Eq. (3) as the *IdleAware* variant.

3) Score Based (SB): The Score Based scheduler [7] has been conceived in order to overcome two issues of the basic PF algorithm: The preferential treatment of flows with small data rates, and the bias against variable radio channels in asymmetric conditions observed in [11]. The scheduling decision is based on the rank of the currently achievable data rate among the values in a recent time window of size $N \cdot T_{\mathrm{TTI}}$:

$$tag_k = 1 + \sum_{l=1}^{N-1} \begin{cases} 1 & \text{if } R_k(t) > R_k(t - l \cdot T_{\mathrm{TTI}}) \\ X_l & \text{if } R_k(t) = R_k(t - l \cdot T_{\mathrm{TTI}}) \\ 0 & \text{otherwise} \end{cases} \;, \tag{6}$$

where X_l are i.i.d. random variables on $\{0, 1\}$ with $\mathrm{P}\{X_l = 0\} = 1/2$. The tags are therefore independent of the history of scheduling decisions.

4) TCP – Proportional Fair (TCP-PF): The TCP – Proportional Fair scheduler [12] is an adaption of the PF algorithm suitable to TCP traffic on wireless links. Generic channel-aware schedulers provide no bounds on ISTs, possibly resulting in strongly varying packet delays. Excessive delays may erroneously provoke TCP timeouts eventually triggering the protocol's slow start mechanism, and therefore reduce the achievable throughput. To avoid such spurious timeouts, the TCP-PF scheduler introduces a correction factor limiting inter-scheduling gaps. The scheme aims at improving the performance on the TCP layer by estimating transport layer metrics from measurements available on the link layer. Applying the PF update function (Eq. (2)), the scheduling tag is calculated as:

$$tag_k = \underbrace{\frac{R_k(t)}{\bar{R}_k(t)}}_{\text{PF tag}} \left(\frac{1}{2} \frac{\hat{\Sigma}_k(t)}{\hat{\Phi}_k(t)} + \frac{\bar{B}_k(t)}{\bar{R}_k(t)} \right) \;, \tag{7}$$

where $\bar{B}_k(t)$ is the sliding average of the transmission buffer level of flow k at time t. $\hat{\Phi}_k(t)$ is the estimation of the average of the IST in a sliding time window, $\hat{\Sigma}_k(t)$ its second moment. Details are given in [12].

IV. Performance Evaluation

All scheduling schemes introduced in the previous chapter are evaluated both in the homogeneous and the heterogeneous scenario. The metrics under consideration are illustrated in Fig. 2. In the homogeneous scenario, the indicated throughput is the data rate R_{IP} of the IP traffic arriving at the UE, which excludes retransmissions in the RAN and therefore represents the actually usable data rate. The timing performance is given by the IP packet service time T_{IPserv}, which comprises all delays induced by the RAN except for the RLC layer input queue. Since the delay in this queue strongly depends on the offered traffic, T_{IPserv} is more suitable to evaluate the scheduling algorithms than T_{RAN}.

In the heterogeneous scenario, one performance metric most suitable to express the respective QoS requirements is given for each traffic class. For gaming traffic, the total IP packet delay in the RAN T_{RAN} is indicated. Since Timing constraints of the streaming traffic directly translate into packet losses due to the deployed timer mechanism, the loss probability $P_{\mathrm{loss}} = 1 - R_{\mathrm{IP}}/R_{\mathrm{IPin}}$ is most significant for this traffic class. For the best effort FTP traffic aiming at a maximal throughput, R_{IP} is considered like in the homogeneous scenario.

In this section, we will first illustrate the differences between an equal distribution of the transmission time and an equal distribution of the transmission power on the basis of two channel-independent disciplines. Then, we will highlight the effects of a channel-aware algorithm by comparing the RR scheduler and the variants of the PF discipline. Finally, we will address the differences between several opportunistic concepts by evaluating the SB and the TCP-PF schedulers.

A. Analysis of Round Robin Schemes

1) Homogeneous scenario: Figure 5 plots the average datarate in the homogeneous scenario. Plotted are the achieved data rates for the different channel-aware schemes depending on the observation window τ, as well as for the channel-independent schemes. As expected, the PBRR scheduler achieves a slight gain over the RR scheduler. Additionally,

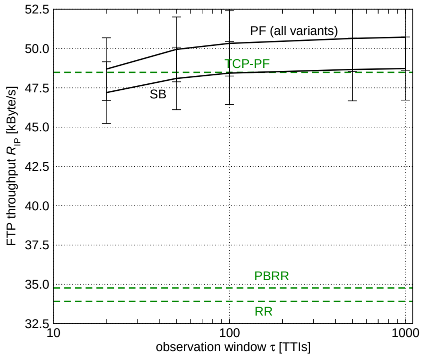

Fig. 5. FTP throughput for the homogeneous scenario

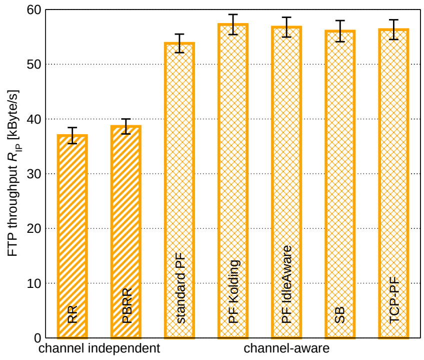

Fig. 6. Throughput of FTP traffic in the heterogeneous scenario

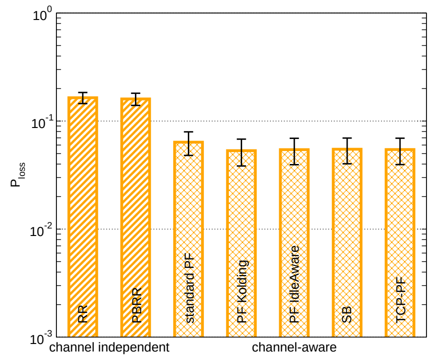

Fig. 7. P_{loss} of the streaming traffic (heterogeneous scenario)

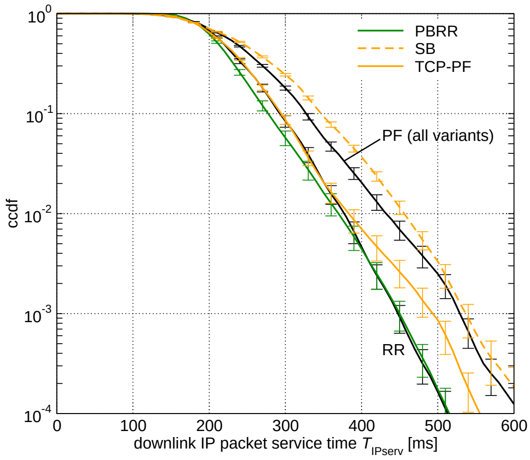

Fig. 8. IP packet service time for the homogeneous scenario and window sizes of $\tau = 100$ (where applicable)

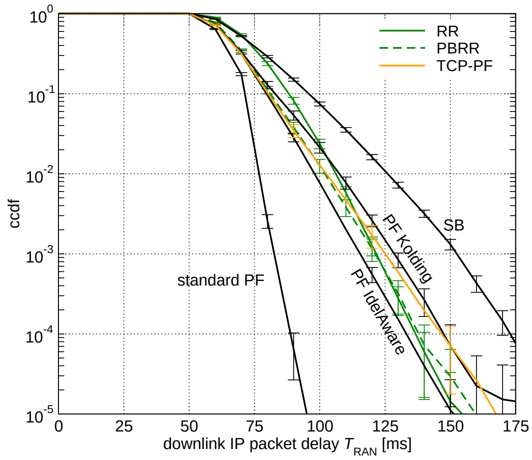

Fig. 9. Downlink IP packet delay of the gaming traffic (heterogeneous scenario)

Fig. 8 plots the cumulative complementary distribution function (ccdf) of the IP packet service time T_{IPserv}, which exhibits a slightly greater variance for the PBRR than for the cyclically operating RR discipline.

2) Heterogeneous scenario: Figure 9 plots the ccdf of T_{IPserv} for the gaming traffic in the heterogenous traffic scenario. Compared to the RR discipline, the PBRR scheduler achieves shorter, but more variant service times. Formerly idle terminals turning active during a PBRR scheduling cycle are directly integrated in the scheduling process and served in the same cycle. They are attributed an initial power count of 0 and therefore tend to be served rapidly when they enter towards the end of the scheduling cycle, which reduces the experienced delays. As the low data-rate gaming sources are often in the situation of empty queues and newly arriving packets, they are served preferentially. However, the dependence of the experienced delay on the instance at which a packet arrives results in a greater variance.

Figure 6 shows the average FTP througput in the heterogenous scenario. The slightly higher throughput achieved by the FTP terminals in combination with the PBRR algorithm goes well along with the observation in the homogeneous scenario and the theoretical considerations.

Figure 7 presents the loss probability of gaming packets in the heterogeneous scenario. Unlike for the FTP traffic, the PBRR discipline does not exhibit a better perfomance than the RR scheduler. More variant service times stated in the homogeneous scenario counteract a potential throughput increase, since they are disadvantageous for the required constant flow of the streaming data.

B. Comparison of PF-Variants

1) Homogeneous scenario: Since the PF scheduler potentially increases the system capacity, we expect a better performance compared to the RR scheduler. Fig. 5 shows an increase of the data rate of about 50 %, which goes well along with the findings in [10]. Decisive deviations between the PF variants are not observed, since the algorithmic differences only concern the treatment of low or empty buffers, which hardly occur during continuous FTP downloads.

The delay ccdfs for a PF observation window size of $\tau = 100$ in Fig. 8 certify the RR scheduler a better delay performance. While the RR scheduler serves all flows in regular intervals, the PF scheduler may wait for their corresponding radio channel to be in a favorable condition. This leads to larger inter-scheduling intervals, which is a well-known effect of the PF scheduler [12]. Again, differences between the PF variants are negligible.

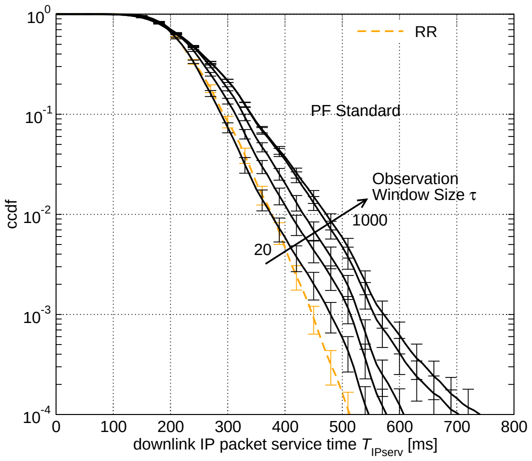

Fig. 10. IP packet service time for the standard PF scheduler and window sizes of $\tau = 20, 50, 100, 500, 1000$ in the homogeneous scenario

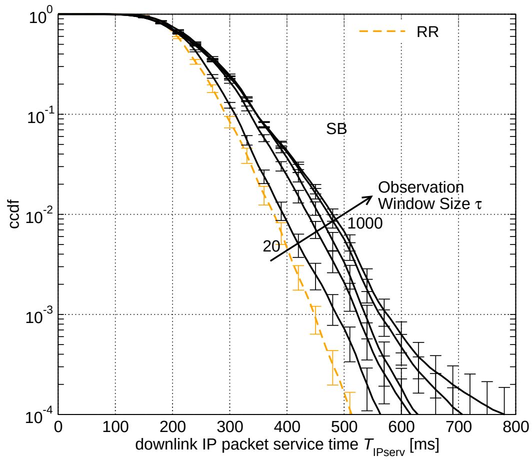

Fig. 11. IP packet service time for the SB scheduler and window sizes of $\tau = 20, 50, 100, 500, 1000$ in the homogeneous scenario

Figure 5 reveals that the FTP throughput increases with the observation window size. The observation window indicates how long the scheduler waits for the channel of a terminal to improve after its degradation before serving the terminal again. Hence, transmissions during bad channel conditions are more likely to occur for smaller window sizes. Therefore, larger windows allow for higher multiuser diversity gains. The delay ccdfs for the standard PF variant in Fig. 10 show a contrary behavior with respect to the packet delay. Small observation windows improve the timing performance of the PF scheduler by reducing the inter-scheduling intervals. For a window size of 20, the PF delay curve even partly falls below the one of the RR scheduler. Both phenomena are well-known for the PF algorithm [9]. As a good tradeoff, a window size of $\tau = 100$ is used for the PF component of the TCP-PF scheduler as well as for all evaluations in the heterogeneous scenario.

2) Heterogeneous scenario: According to the consideration motivating the other PF variants, the standard algorithm is expected to treat terminals demanding low data rates preferentially. Fig. 9 accordingly certifies this variant excellent delays of the gaming traffic. Even a static prioritization would not improve the timing performance [13]. For the two other PF variants, average delays close to those of the PBRR scheduler are observed, but with a greater variance. Longer delays compared to the standard PF algorithm are a direct consequence of the intended independence of the data rates; the greater variance is inherent to opportunistic, non-cyclical scheduling. Differences between Kolding's and the IdleAware variants are of minor importance. They depend on both the channel development in idle periods and the parameter α of the IdleAware variant, which is set to $\alpha = 1.5$ for this evaluation.

The loss probability of the streaming traffic in Fig. 7 and the FTP throughput in Fig. 6 show a contradictory behavior. Higher losses and a lower throughput certify the standard PF algorithm a worse performance than the other two variants, which do not differ significantly in their performance. Since the homogeneous scenario resulted in an identical performance of all PF variants when serving continuous data streams, the observed discrepancies can be traced back to the differences of the gaming packet delays. Similar to the homogeneous scenario, the PF variants achieve an increase in throughput over the RR variants. The decrease of the losses of streaming packets corresponds to a minor throughput increase. Since the constant data rates of the streaming sources prevent a full exploitation of a temporally excellent channel, the scheduling gains are limited.

C. Comparison of PF-Variants and SB / TCP-PF

1) Homogeneous scenario: The FTP throughput for the SB scheduler in Fig. 5 still shows a significant increase over the RR scheduler, but it is inferior to the data rate for the PF algorithm. The SB approach is based on the rank of the current channel quality and does not take into account the ratio of a good channel to its average. Consequently, a terminal with a nearly constant channel quality that slightly exceeds its average at a given time may be scheduled despite another terminal experiencing an extremely good channel (relative to its average). A PF algorithm, in contrast, would serve the latter one. Therefore, the SB discipline proves less efficient in exploiting multi-user diversity.

As Fig. 8 reveals, the delays observed for the SB scheduler not only exceed those of the RR, but even those of the PF scheme for equal window sizes. Though both the SB and the PF algorithm imply non-deterministic inter-scheduling intervals, the PF scheduler preferably serves previously neglected terminals by considering the ratio of the achievable data rate and the actually *realized* one in the recent past. Hereby, ISTs are limited. The SB discipline, in contrast, ranks the currently achievable throughput among the *achievable* data rates in the time window and hence lacks the compensating effect. This, in turn, results in longer delays.

The SB algorithm has been conceived to overcome some drawbacks of the PF scheduler. However, these drawbacks only materialize in asymmetrical channel conditions. Hence, this aspect cannot be evaluated in the homogeneous scenario.

The dependency of the throughput and delay of the SB scheduler on the observation window shown in Fig. 5 and 11, respectively, is similar to the behavior observed for the PF scheduler and allows for the same explanations.

The throughput gain of the TCP-PF scheduler over the PF discipline reported in [12] results from the avoidance of spurious TCP timeouts triggering the slow start mechanism and thus reducing the data rate. However, timeouts are not observed in statistically relevant numbers for the PF algorithm in our setting. Consequently, no throughput gain over the PF scheduler is observed. Figure 5 certifies the TCP-PF scheme a worse performance for a PF window of 100. The delay ccdfs in Fig. 8 show a contrary behavior, where the TCP-PF delays range around those of the RR for 99 % of all packets, well below the PF. This is the intended smoothing effect of the timeout-avoiding functionality. The interdependence between the throughput and the delay in case of a PF-style algorithm is highlighted by the fact that the TCP-PF scheduler and the standard PF discipline with a window size of 20 exhibit both the same data rate and a similar timing behavior.

2) Heterogeneous scenario: According to Fig. 9, the SB scheduler shows the worst timing performance for gaming users among the studied algorithms. The behavior can directly be traced back to the long delays observed in the homogeneous scenario, since the operation of the SB scheduler is independent of the realized data rate and the scheduling history.

Despite longer delays for gaming packets compared to the PF algorithm, the SB scheduler does not provide better performance to the other traffic classes. Fig. 7 and 6 show that the achieved streaming loss and FTP throughput, respectively, range between the values for the standard PF and the other two PF variants. This underlines the fact that the SB algorithm has problems to fully exploit multi-user diversity in the investigated scenario compared to the PF scheduler, as already observed in the homogeneous scenario.

The timing performance provided to the gaming traffic by the TCP-PF scheduler given in Fig. 9 ranges between those of Kolding's and the IdleAware PF variant. Since the TCP-PF algorithm is based on the standard PF scheme, the increased delays as well as their greater variance trace back to the factor for timeout avoidance.

Alike, the service by the TCP-PF algorithm to the other traffic classes is comparable to the advanced PF variants. While the FTP throughput in Fig. 6 slightly trails these PF variants, Fig. 7 certifies the TCP-PF the same performance as Kolding's PF with respect to streaming losses. The limited variance of packet delays this scheduler aims at is advantageous for the streaming traffic with a constant data rate.

V. Conclusion

In this paper, we evaluated several channel-aware scheduling algorithms in different traffic scenarios. We first verified the fact that channel-aware schedulers can significantly increase the experienced data rates at the expense of the packet delay performance. We then showed that the different variants of the classical Proportional Fair scheduler perform very well compared to more specialized algorithms, such as the Score Based or the TCP-PF algorithm, both in a homogeneous traffic scenario and a multi-service scenario. Depending on the parametrization of the PF algorithm, the throughput gain can be traded off against the packet delay penalty. Additionally, users with a low transmission volume may be prioritized using the standard PF variant. On the other hand, if an equal treatment of users with different traffic characteristics is desired, the Kolding or IdleAware variants showed to deliver good performance with respect to the overall system throughput.

Acknowledgment

This research was done in cooperation with Alcatel Research and Innovation Department, Stuttgart.[1]

References

[1] R. A. Comroe and D. J. Costello, Jr., "ARQ schemes for data transmission in mobile radio systems," *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 4, pp. 472–481, July 1984.

[2] H. Fattah and C. Leung, "An overview of scheduling algorithms in wireless multimedia networks," *IEEE Wireless Communications*, vol. 9, no. 5, pp. 76–83, October 2002.

[3] *IKR Simulation Library*. [Online]. Available: http://www.ikr.uni-stuttgart.de/Content/IKRSimLib/

[4] M. C. Necker and A. Weber, "Impact of Iub flow control on HSDPA system performance," in *Proc. 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2005)*, Berlin, Germany, September 2005.

[5] M. C. Necker, "A comparison of scheduling mechanisms for service class differentiation in HSDPA networks," *International Journal of Electronics and Communications*, vol. 60, no. 2, pp. 136–141, February 2006.

[6] K. I. Pedersen, T. F. Lootsma, M. Stottrup, F. Frederiksen, T. E. Kolding, and P. E. Mogensen, "Network performance of mixed traffic on high speed downlink packet access and dedicated channels in WCDMA," in *Proc. Vehicular Technology Conference (VTC 2004-Fall)*, Los Angeles, CA, September 2004.

[7] T. Bonald, "A score-based opportunistic scheduler for fading radio channels," in *Proc. European Wireless 2004*, Barcelona, Spain, February 2004.

[8] A. K. F. Khattab and K. M. F. Elsayed, "Channel-quality dependent earliest deadline due fair scheduling schemes for wireless multimedia networks," in *Proc. International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, Venice, Italy, October 2004, pp. 31–38.

[9] A. Jalali and R. P. R. Pankaj, "Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system," in *Proc. Vehicular Technology Conference (VTC 2000-Spring)*, vol. 3, Tokyo, Japan, May 2000, pp. 1854–1858.

[10] T. Kolding, "Link and system performance aspects of proportional fair scheduling in WCDMA/HSDPA," in *Proc. Vehicular Technology Conference 2003 (VTC 2003-Fall)*, vol. 3, October 2003, pp. 1717–1722.

[11] J. Holtzman, "Asymptotic analysis of proportional fair algorithm," in *Proc. 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2001)*, vol. 2, San Diego, CA, October 2001, pp. F–33 – F–37.

[12] T. E. Klein, K. K. Leung, and H. Zheng, "Enhanced scheduling algorithms for improved TCP performance in wireless IP networks," in *Proc. IEEE GlobeCom*, Dallas, TX, December 2004.

[13] M. C. Necker, M. Scharf, and A. Weber, "Performance of TCP and HTTP proxies in UMTS networks," in *Proc. European Wireless*, vol. 2, Nicosia, Cyprus, April 2005, pp. 504–510.

[1] Alcatel SEL AG, Research & Innovation, Lorenzstr. 10, 70435 Stuttgart, Germany. Contact: Andreas Weber (Andreas.Weber@alcatel.de).