Maximizing Minimum Throughput Guarantees: The Small Violation Probability Region

M. Majid Butt, *Member, IEEE*, Dževdan Kapetanović, *Member, IEEE*, and Björn Ottersten, *Fellow, IEEE*

***Abstract*—Providing minimum throughput guarantees is one of the goals for radio resource allocation schemes. It is difficult to provide these guarantees without defining violation probability due to limited power budget and rapidly changing conditions of the wireless channel. For every practical scheduling scheme, there is a feasibility region defined by the minimum guaranteed throughput and the corresponding probability that the users fail to get the guaranteed throughput (violation probability). In this work, we focus on minimizing the violation probability specifically in the small probability region. We compare our results with major schedulers available in literature and show that our scheme outperforms them in the small violation probability region.**

***Index Terms*—Multiuser diversity, opportunistic scheduling, throughput guarantees, violation probability.**

I. Introduction

THE objective of radio resource allocation schemes is to maximize/minimize a utility function depending on the problem definition. Some of the schemes aim to maximize the average throughput of a user. One representative of this class of schedulers is the maximum throughput scheduler (MTS), where a user with the best instantaneous channel is scheduled for transmission [1]. However, this scheme does not provide any kind of short term fairness or performance guarantees to the users. A natural solution to the fairness problem is the well-known round robin scheduler (RRS). It is a fully deterministic scheduler where every user is scheduled in a fixed pre-allocated time slot. This scheduler completely ignores the channel and therefore results in a small average throughput. Another scheduler which exploits channel conditions and still provides fairness is termed as proportional fairness scheduler (PFS) [2]. In PFS, a user is scheduled if he maximizes the ratio between his current rate and average throughput. The work in [3] provides a theoretical framework to compare the performance of these schedulers in terms of average sum rate and average fairness.

It is quite common to evaluate the performance of the schemes with the assumption of infinite length windows, i.e., the average throughput of a user is the sample mean of all his previous rates. Although this assumption simplifies the analysis, it gives little insight about the short term behavior of the scheme. Namely, even though the long term average throughput for a user might be large, it could be the case that he did not get channel access for some time. For real time applications like multimedia, this causes jitter in the received quality of service (QoS). It is thus important to evaluate the performance of a scheduler for the finite window case.

Our utility function is the violation probability for a user, defined as the probability that the average throughput (measured over a finite time window) of a user falls below a guaranteed throughput. Reference [4] provides approximate expressions for the violation probability for some of the existing schedulers under finite windows. The work in [5] uses a similar approach to compute the violation probability and proposes a heuristic scheme to provide throughput guarantees in finite windows. This work keeps the users out of contention whose throughput is above the target throughput. In our opinion, this step is highly suboptimal as confirmed through numerical results in Section IV. The authors in [6] discuss a similar problem where the objective is to minimize the transmit power and the number of scheduled users, assuming that the violation probability is below a certain value.

In this work, the power for each user is kept constant and the only degree of freedom is choosing which user to schedule at different time slots. We propose a new scheme which performs better than the state of the art scheduling schemes in terms of guaranteed throughput violation behaviour. We show that bringing some intelligence about the short-term scheduling history of the users can be helpful in the scheduling process. Our scheduling scheme predicts the violation events in the time domain and utilizes this information to schedule the users to minimize the occurrence of such events.

The rest of this paper is organized as follows. Section II describes the system model used in the work. We propose a scheme in Section III to improve the performance in the small violation probability regions. Our scheme is compared numerically with other schemes in Section IV and we conclude with the main contributions of the work in Section V.

II. System Model and Preliminaries

We assume a time-division multiple-access (TDMA) channel with K users distributed uniformly in a cell. The channel h_k of a user k is characterized by a fast fading environment, where the channel outcomes remain constant during the time span of a time slot and are independent and identically distributed (i.i.d.) across the time slots. This model is called a *block fading model*. The channels are also i.i.d. across the users. Thus, all the users are statistically symmetrical with respect to the channel distribution.

Manuscript received December 25, 2012. The associate editor coordinating the review of this letter and approving it for publication was L. Le.

The authors are with the Interdisciplinary Center for Security, Trust and Reliability (SnT), University of Luxembourg (e-mail: {majid.butt, dzevdan.kapetanovic, bjorn.ottersten}@uni.lu).

This work was carried out during an ERCIM "Alain Bensoussan" Fellowship tenure. This Programme is supported by the Marie Curie Co-funding of Regional, National and International Programmes (COFUND) of the European Commission.

Digital Object Identifier 10.1109/WCL.2013.13.130077

We assume a fixed transmit power P allocated to a scheduled user k^*, and an additive white Gaussian noise power of N_0. Assuming an optimal Gaussian codebook, the resulting rate $R_{k^*}(t)$ at a time slot $t \in \mathbb{Z}$ is given by

$$R_{k^*}(t) = \log_2\big(1 + |h_{k^*}^2|\frac{P}{N_0}\big). \tag{1}$$

We assume that every user has enough buffered data to optimally use the rate provided by the channel. For the non-scheduled users, their rates at time slot t are 0. The average throughput $\bar{T}_k(t)$ of a user k up to time slot t is defined as

$$\bar{T}_k(t) \triangleq \frac{\sum_{j=1}^{L_W} R_k(t-j)}{L_W}, \tag{2}$$

where $R_k(t-1), \ldots, R_k(t-L_W)$ are the rates allocated to user k during the last L_W time slots. Note that some of these rates can be zero, which occurs in the time slots where user k was not scheduled. Hence, the average throughput for a user is calculated across a window of L_W time slots.

The violation probability for a user k is defined as

$$\delta_k(T_G) \triangleq \lim_{t\to\infty} \frac{\sum_{j=1}^{t} I(\bar{T}_k(j) < T_G)}{t}. \tag{3}$$

where $I()$ denotes a standard indicator function which is one if the argument is true, zero otherwise. It is clear from the definition of δ_k that $0 \leq \delta_k \leq 1$. The constant T_G is the guaranteed throughput, i.e., the least rate guaranteed to each user. Hence, δ_k is the probability that the average throughput of the user k falls below the guaranteed throughput. From a network operator and user point of view, it is of interest to keep the violation probability of each user as small as possible for a given T_G. Equivalently, we can express the problem as

$$\text{Max} \qquad T_G \tag{4}$$

$$\text{s.t.} \qquad \delta_k = c, \quad 0 \leq c \leq 1 \tag{5}$$

where c denotes a constant. However, the most significant region is when c (and violation probability) is small as the users tolerate only a small violation in guaranteed throughput.

There is a clear trade off between a large throughput guarantee and the violation probability – larger T_G's give rise to larger violation probabilities. The focus in this work is on the QoS guarantee, and therefore the goal is to provide as large throughput guarantee as possible up to a certain violation probability. We do not provide an explicit mathematical solution of the optimization problem defined in (4) and (5) as the problem is hard to solve due to involvement of a lot of finite parameters like number of users, channel distribution, window size, T_G, etc. Without claiming optimality, we develop a heuristic scheduling scheme that significantly outperforms the well-known schedulers in the finite window case.

III. Proposed Scheme

RRS tries to avoid outage in a deterministic manner while MTS makes no such attempt at all. Although scheduling based on the ratio $R_k(t)/\bar{T}_k(t)$ in PFS gives preference to the users with low average throughput, it does nothing to prevent a violation event from taking place at first hand. An intelligent algorithm should be able to early detect and prevent the violation events from happening. This is especially important for small window sizes L_W, since after L_W time slots, the rates once scheduled to a user disappear forever.

We propose a scheme which keeps track of the potential violation event in the time domain and prioritizes the scheduling accordingly. To do so, we define the term *throughput deadline* $D_k(t)$ for a user k at time slot t as the maximum number of time slots available until his average throughput falls below T_G if he is *not scheduled* continuously. To compute $D_k(t)$, we define a variable $j^*(t)$ at time t as the smallest integer where the sum of allocated rates (normalized by window size) in the most recent j^* time slots is greater than T_G:

$$j^*(t) \quad \triangleq \quad \arg\min_j \sum_{i=1}^{j} R_k(t-i), \qquad 1 \leq j \leq L_W \tag{6}$$

$$\text{s.t} \qquad I\Big(\frac{1}{L_W}\sum_{i=1}^{j} R_k(t-i) > T_G\Big) = 1 \tag{7}$$

where the summation in (6) is evaluated only if $I(.) = 1$. Equation (6) is evaluated for $j = 1$ at start and condition in (7) is checked. If $I(.) = 1$, $j^*(t) = 1$. Otherwise, j is incremented and the process is repeated until $I(.) = 1$. If $I(.) = 0, \ \forall j$, this implies that the user is already violating throughput guarantee and $D_k(t) = 0$. For $j^*(t) \geq 1$, $D_k(t)$ is computed by

$$D_k(t) = L_W - j^*(t) + 1. \tag{8}$$

Deadline $D_k(t)$ is calculated for every user in each time slot.

All the opportunistic schemes like MTS and PFS use maximization of rate R_k along with some function defining the characteristics of the scheduler. Following this approach, we develop a heuristic function based on the factors involved in violation event prevention. In our scheme, a user k^* is scheduled in a time slot t if he maximizes

$$k^* = \arg\max_k g_k(t) R_k(t) \tag{9}$$

where the priority function $g_k(t)$ is given by

$$g_k(t) = \begin{cases} \frac{1}{\big(D_k(t)\big)^{\alpha}}\big(\frac{\hat{D}_k(t+1)+1}{D_k(t)}\big)^{\beta} & \bar{T}_k(t) > T_G \\ 1 & \bar{T}_k(t) \leq T_G. \end{cases} \tag{10}$$

with $\hat{D}_k(t+1)$ denoting the *estimated* deadline of the user assuming that he is scheduled at time t. This is computed using (6) and (8) with the assumption of allocating $R_k(t)$ to the user. The constants α and β are optimized for every ratio K/L_W and are fixed for different values of K and L_W but a fixed ratio K/L_W. Based on the throughput deadline concept, we call our scheme Throughput Deadline Scheduling (TDS).

The priority function g_k is based on the phenomena of violation event prevention and violation period minimization. Let us discuss the case for $\bar{T}_k(t) > T_G$ first. The first term in (10) measures the priority level of the user if $\bar{T}_k(t) > T_G$, i.e., the user's throughput is larger than the target throughput and his priority weighting function aims at avoiding the violation event. In term $1/\big(D_k(t)\big)^{\alpha}$, the priority increases depending on factor α as $D_k(t)$ decreases. The term $\big(\frac{\hat{D}_k(t+1)+1}{D_k(t)}\big)^{\beta}$ measures the relative increase in deadline if the user is scheduled as

compared to the current deadline $D_k(t)$ and depends on rate $R_k(t)$ and constant β. We can further distinguish two cases.

1) $\hat{D}_k(t+1) = D_k(t) - 1$: This implies that the scheduling of a user k in time slot t does not help to increase his deadline at time $t+1$. Thus, the term $\left(\frac{\hat{D}_k(t+1)+1}{D_k(t)}\right)^{\beta}$ in (10) becomes one and the priority function depends solely on the term $1/(D_k(t))^{\alpha}$ for the user. Note that, the constant one in the numerator is necessary to keep $\frac{\hat{D}_k(t+1)+1}{D_k(t)}$ equal to one for this case. In the absence of the additive constant one in the numerator, $\frac{\hat{D}_k(t+1)}{D_k(t)} < 1$ and the bad channel kills the priority coming from the term $1/(D_k(t))^{\alpha}$ as well, which is undesirable.
2) $\hat{D}_k(t+1) > D_k(t)$: In this case, the channel dependent term $\left(\frac{\hat{D}_k(t+1)+1}{D_k(t)}\right)^{\beta} > 1$ enhances the priority of the user (because of good channel) in conjunction with the deadline dependent term $1/\big(D_k(t)\big)^{\alpha}$; hence the cumulative priority improves at a faster rate.

When $\bar{T}_k(t) \leq T_G$ and the user is already violating the throughput guarantee, his priority function helps to recover from this situation. The priority is set to one and the scheduler behaves like MTS for the users already violating the throughput guarantee. We observe that enhancing the priority further for the users violating the throughput guarantee reduces the effect of multiuser diversity and the performance suffers.

As compared to PFS and MTS, we propose throughput deadline as a new metric for giving priority to the user. This metric not only takes current average throughput into account but also gives weight to the location of the scheduled rates in the window. In a sliding window, a user may get a very good rate in a time slot which improves $\bar{T}_k(t)$ for some time. If the scheduler is based on the measure $\bar{T}_k(t)$ only (as in PFS), the user may not get a high priority as $\bar{T}_k$ contains no information about the time slot when a good rate is going to be lost due to sliding window. When the good rate becomes obsolete and moves out of the window, the average throughput drops *instantaneously* and causes throughput violation events. Our scheme predicts this event in the time domain and the scheduler uses this information to assign the priority correspondingly.

A. Recovery Time

QoS for a user often depends on the violation probability which gives the quality of experience (QoE) for a user over the short time period. Additionally, QoE also depends on the ability of the user to come out of this situation quickly and is measured in terms of recovery time. For example, if two users have the same T_G, the user with the smaller recovery time is expected to have better QoE. The large recovery time may deteriorate the performance for the real time applications where error concealment techniques sometimes depend on the temporal correlation of the received data and the larger delays are intolerable. Thus, recovery time is another key performance indicator (KPI) and we evaluate the performance of our scheme in terms of recovery time in Section IV.

IV. Numerical Performance Evaluation

We perform comparisons of our scheme with some of the well-known scheduling schemes using Monte Carlo simulations. The users are uniformly distributed in the cell. The constant transmit power for each user is fixed to 10 dbW. Our channel model employs Rayleigh fading with mean one and the allocated rate for the scheduled user is computed using (1). For a given value of T_G, scheduling operations in $30,000$ time slots are simulated to evaluate the violation probability. Note that the empirical violation probability for a user converges to the system violation probability when the algorithm is run for a long time because the channel distributions for the users are completely symmetrical with respect to each other. The system violation probability can be interpreted as the proportion of the users violating the throughput guarantee in a given time slot.

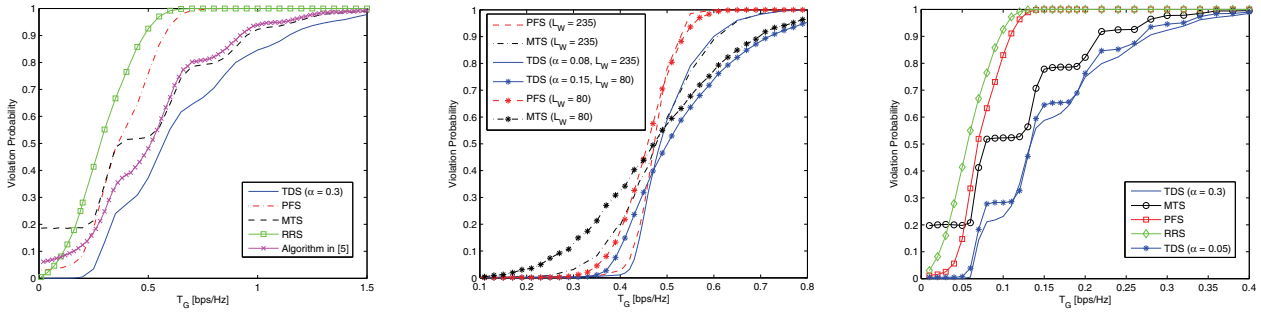
TABLE I
Optimized Constant α for different K/L_W

K/L_W	$\alpha = \beta$
10/16,50/80	0.3
10/80	0.15
10/150	0.1
10/235	0.08

Our numerical investigation shows that the parameters α and β depend on the ratio K/L_W and the channel distribution; and the best solution is achieved for $\alpha = \beta$ for Rayleigh fading distribution. We still parameterize $g_k(t)$ in (10) with independent α and β because it may well be the case that the best performance is achieved for non-equal constants for a different fading distribution. The values of the constants for different K/L_W ratios are shown shown in Table I.

Fig. 1 compares the guaranteed throughput violation performance of our scheme with other well-known schemes. The window size and the number of users are fixed to 16 and 10, respectively in Fig. 1(a). The results reveal that the deterministic schedulers like RRS which do not exploit good channels are severely sub optimal in the violation probability sense and same is the case for the purely opportunistic scheduler MTS. PFS handles the guaranteed throughput violation event in a better way. However, it is evident that our scheme outperforms all the schemes including the scheme proposed in [5]. Specifically, the gain in small violation probability region is substantial. In Fig. 1(b) we observe the behaviour of the schemes when we decrease the ratio K/L_W at fixed $K = 10$. We compare the schemes for the window size 80 and 235 (as suggested by the European Winner I project for future mobile systems [5]). We observe that increase in window size closes the gap between the schemes and all the schemes converge to a long term average behaviour. For a very large window size, the guaranteed throughput violation probability for all the schemes would converge eventually to a sharp step function which implies that temporal windowing effect just vanishes and the short-term average throughput of the users converges to the mean value. In Fig. 1(c), we have $K = 50$ and $L_W = 80$; which is the same ratio as for the case $K = 10$ and $L_W = 16$ in Fig. 1(a). When the ratio K/L_W is the same, we observe very similar behavior of the curve for various (but different) T_G values. Thus, we conclude that ratio K/L_W is one of the influential parameters that determines the guaranteed throughput violation behaviour of the scheduling schemes for the finite window cases.

We compare the long term average throughput for all the schemes in Fig. 2. Both MTS and PFS are independent of T_G values and show the same average throughput throughout.



(a) Comparison for $K = 10$ and $L_W = 16$ (Mobile Wimax standard [4]).

(b) Comparison for $K = 10$ and large L_W.

(c) Comparison for $K = 50$ and $L_W = 80$ but with the same K/L_W as in Fig. 1(a).

Fig. 1. Guaranteed throughput violation behaviour of our scheme in comparison to well-known scheduling schemes.

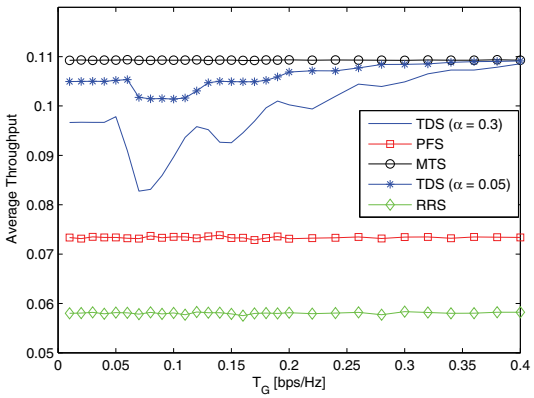


Fig. 2. Long term average throughput comparison for system parameters $K = 50$ and $L_W = 80$ when throughput is averaged over long time periods.

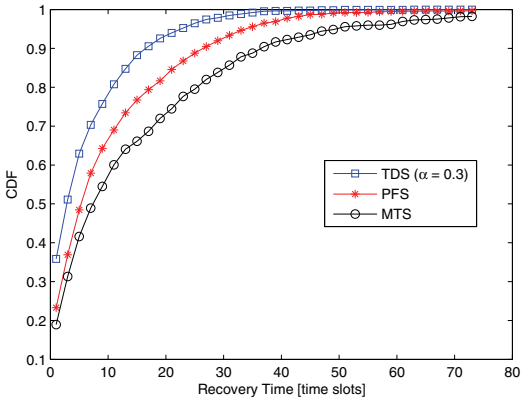


Fig. 3. Cumulative Distribution Function (CDF) for the recovery time of different scheduling schemes for system parameters $K = 10$ and $L_W = 16$.

As expected, MTS provides the best average throughput for this case. On the contrary, TDS depends on T_G via function $g_k(t)$. For the violation probability optimal $\alpha = 0.3$, TDS outperforms PFS for small T_G's, then the gap gets closer with PFS; and finally converges to MTS for large T_G's. We show that if our KPI is average throughput instead of violation probability, a different $\alpha = 0.05$ improves the long term average throughput performance; but the guaranteed throughput violation behaviour deteriorates as shown in Fig. 1(c). A joint comparison of MTS and TDS in Fig. 2 and Fig. 1(c) gives us an interesting insight. Up to $T_G \simeq 0.05$ bps/Hz, average throughput for TDS is constant as $\delta_k = 0$ regardless of T_G. After this point, violation probability and average throughput curves for MTS and TDS converge and diverge from each other correspondingly. Finally, both MTS and TDS converge in average throughput and violation probability at $T_G = 0.4$ bps/Hz while average throughput performance of our scheme is still better than PFS.

We compare the recovery time as another KPI for all the schemes in Fig. 3. We observe that probability of a user coming out of outage in a short time for TDS is much higher as compared to other schemes. Specially, the difference is significant for the small number of time slots.

V. Conclusion

Our objective is to minimize the violation probability for a given target throughput and finite window sizes. Based on the time domain analysis of the short-term behavior of the users' throughput, we propose a heuristic scheduling scheme which outperforms the well-known and state of the art schemes. The results demonstrate that the short window size impact on all the classical algorithms which ignore the window size is much worse as compared to our scheme. The proposed scheme keeps track of the allocated resources in the time span of the window and optimizes the resource allocation accordingly. As window size increases, the short-term throughput converges to the long-term average throughput.

References

[1] R. Knopp and P. Humblet, "Information capacity and power control in single cell multiuser communications," in *Proc. 1995 IEEE Int. Conference on Communications*.

[2] P. Viswanath, D. N. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *IEEE Trans. Inf. Theory*, vol. 46, no. 6, pp. 1277–1294, June 2002.

[3] E. A. Jorsweick, A. Sezgin, and X. Zhang, "Framework for analysis of opportunistic schedulers: average sum rate vs. average fairness," in *Proc. 2008 International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*.

[4] V. Hassel, G. E. Øien, and D. Gesbert, "Throughput guarantees for wireless networks with opportunistic scheduling: a comparative study," *IEEE Trans. Wireless Commun.*, vol. 6, no. 12, pp. 4215–4220, 2007.

[5] J. Rasool, V. Hassel, S. de la Kethulle de Ryhove, and G. Oien, "Opportunistic scheduling policies for improved throughput guarantees in wireless networks," *EURASIP J. Wireless Commun. and Networking*, vol. 2011, no. 1, p. 43, 2011.

[6] N. Chen and S. Jordan, "Downlink scheduling with guarantees on the probability of short-term throughput," *IEEE Trans. Wireless. Commun.*, vol. 8, no. 2, pp. 593–598, 2009.