Class and Channel Condition Based Scheduler for EDGE/GPRS

Rajeev Agrawal, Anand Bedekar, Richard J. La,
Rajesh Pazhyannur, and Vijay Subramanian
Mathematics of Communication Networks
Motorola Inc., Arlington Heights, IL

Abstract

The efficient management of the radio resource of a 3-G system is important from an operator's perspective. This, however, cannot be the only concern when quality of service (QoS) negotiations have been made for various users and the operator has to uphold these. This leads to a fairness objective that the operator has to keep in mind. In this paper we outline a scheme to perform packet-level scheduling and resource allocation at the wireless node that takes into account the notions of both efficiency and fairness and presents a means to explore the trade-off between these two notions. As a part of this scheme we see the scheduling problem as deciding not just the packet transmission schedule but also the power allocation, the modulation and coding scheme allocation and the spreading code determination since the latter three directly influence the radio resources consumed. Using a utility maximization formulation based on the data-rates that the mobiles can transmit at, we decide on the weights for a weighted proportionally fair allocation based scheduling algorithm. We also show how one can adapt the weights and the algorithm for a time-varying channel. We conclude with a simulation based performance analysis for infinitely-backlogged sources and TCP sources on an EDGE system.

1 Introduction

The explosion of multimedia services on the Internet is leading to a demand of the same services in the non-tethered wireless space. There are, however, many peculiarities that a wireless channel possesses which makes supporting such services much tougher than on wireline networks. One of the key elements in this is the scheduler used at various nodes. Scheduling in traditional wireline networks consists mainly of deciding the order in which users access the channel. This is because it is quite easy to use these channels very close to their (information theoretic) capacity at any given power (used on the channel). Thus, it is best to operate at the maximum capacity by using the maximum power all the time. In addition, the channels, and thus the data rates, are not time-varying either. On wireless channels, however, there are many considerations which do not allow for such a mode of operation. The bandwidth available for transmission on a wireless channel and the power levels allowed (both regulated) put a hard limit on the capacity[1]. Another important element is the mobility of the end-user equipment which results in time-varying multipath and fading. Further, the size, battery power, and processing power of the end devices place additional constraints on system performance. Limited battery capacity also makes it necessary to use transmission schemes that would prolong battery life as much as possible. Finally, the multiuser nature of a wireless channel makes it interference-limited. Thus, one user transmitting at maximum power could severely impair the transmissions of other users. Thus, using a traditional

[1] The propagation characteristics of the atmosphere, and other media, are also deciding factors for the band of operation and hence, the bandwidth.

wireline scheduler is not a good approach on a wireless channel.

Some recent developments in wireless link scheduling include the work by Holtzman [7], Jalali *et al.* [9], Tse [16], Shakkottai and Stolyar [15], Chawla *et al.* [14], Leelahakriengkrai [12], and Berry and Gallager [6]. All of these works show that substantial benefits are achieved when the higher layers are aware of the radio conditions and can adapt the powers, the modulation schemes, the coding schemes, and spreading gains (and hence the data rates) based upon this knowledge. The upshot of this is that scheduling policies should be devised using the knowledge of channel conditions. In a cellular context there is an additional benefit to the network layer control of transmission strategies. In such a situation it is possible to trade capacity among cells (by changing the power levels, for instance) to alleviate periods of congestion or high demand. From the discussion above it is clear that a scheduler that jointly performs packet-level scheduling and radio-resource allocation is the solution for the wireless-link scheduling problem. To be able to implement such scheduling policies it is necessary to have a system that has controls in place to allow for changing the transmission parameters easily. The third-generation (3G) technologies are a first step in this direction. Nanda *et al.* [13] provide a fairly comprehensive overview of the 3G technologies and how they have been designed with multimedia-type services in mind. In all proposals it is possible for connections to not only choose from a variety of data rates but also change the data rate in a flexible and quick fashion. There is also added feedback, in terms of more frequent error and measurement reports, which in conjunction with flexible data rate allocation, can in turn be used for a better monitoring of QoS guarantees and provisioning of resources.

Since the radio resource is quite expensive the efficient management of this resource is critical. This, however, cannot be the only concern when QoS parameters have been agreed to for various users and when the operator is obliged to uphold them. Thus, it is imperative to have some fairness in the arbitration of resources amongst the various users. In this paper we outline a scheme to perform packet-level scheduling and resource allocation at the wireless (access) node that takes into account the notions of both efficiency and fairness and presents a means to explore the trade-off between these two notions. This trade-off between efficiency and fairness was not a concern of earlier cellular systems because voice was the major application and thus, only coverage (which is in reality just another terminology for fairness) was critical. It is only with the emergence of the 3G technologies and packetized data services that such a trade-off makes any sense. In wireline networking there are two broad philosophies when QoS provisioning is considered. One follows the IntServ approach and attempts to provide strict QoS guarantees [2]. Another approach uses the ideas in DiffServ to provide a class-based differentiation of services [3]. Differentiated services are supported through various per-hop-behaviors (PHBs) in DiffServ capable networks. For instance, expedited forwarding (EF) is aimed at supporting real-time applications such as video conferencing. Other PHBs, such as assured forwarding (AF) [4] and best effort (BE), support non-real-time applications that do not require strict delay guarantees. AF can further have different services, for instance, Gold, Silver, and Bronze services.

The work in [15, 12] is more in the IntServ context and they schedule users based upon their current backlog to satisfy statistical delay guarantees. With such scheduling mechanisms one needs a good admission control policy and policing mechanism in place. The work in this paper is based on the DiffServ philosophy where we provide a differentiation based upon both class and channel-state. Further, we restrict our attention to non-real-time (rate-adaptive) services. The reason we choose to provide a service differentiation based upon channel-state is because this determines how much network resource is utilized by the application. This is very much in keeping with similar differentiation in wireline with protocols like TCP using quantities like round-trip time estimates and hop-count as surrogates for measuring how much of the network resource is utilized by various applications. The work in [7, 9, 16, 14] also adheres to the DiffServ philosophy. In the proposed algorithm in this paper we propose a flexible way of trading off efficiency for fairness as well as a flexible way of exploiting temporary fluctuations in channel conditions.

In Section 2 we give a detailed introduction to the wireless link scheduling problem. Thereafter, we introduce a related resource allocation problem in Section 3 and present our scheduling algorithm in Section 4. A performance analysis of the algorithm with and without TCP sources is discussed in Section 5. Finally, we conclude in Section 6.

2 The Wireless Link Scheduling Problem

Consider a cellular system. In a given cell b let J_b be the set of users on the downlink. Time is slotted into radio blocks (in GPRS and EDGE) or frames in (cdma2000 and UMTS) of fixed duration (20, 10, or 3.33 ms depending on the technology). We shall refer to these time slots as frames hereafter. The wireless link scheduling problem is one of deciding which of these users transmit in each frame. In a TDMA system like GPRS and EDGE only one user is allowed to transmit in a frame, whereas in a CDMA system like cdma2000 or UMTS multiple users may transmit in a frame. When a user transmits, we also need to decide what power level, modulation and coding scheme, time slot (in case of TDMA) and spreading factor (in case of CDMA) it will use.

Due to different base site to user distances, shadow fading and multipath, the channel conditions of the different users vary with time. This fluctuation in channel conditions results in a variation of the *effective data rate (per channel per unit power)* $\hat{R}_j(t)$ available to the different users $j \in J_b$ in different frames $t = 0, 1, \ldots$. This effective data rate per unit resource may be calculated in a variety of ways. We give two simple ones below:

Consider the signal to interference plus noise ration (SINR) of user j

$$\mathrm{SINR}_j(t) = \frac{P_j(t)G_{jj}(t)}{\sum_{i \neq j} P_i(t)G_{ij}(t) + \sigma^2}, \tag{1}$$

where σ^2 is the receiver noise variance and $G_{ij}(t)$ is the energy gain from base station of mobile station i to mobile station j. Once $\mathrm{SINR}_j(t)$'s are available for the users (either using the above formula or direct measurements), one can compute the data rates and frame error rates (FER) corresponding to different choices of modulation and coding schemes (MCSs) and/or spreading factors (SFs). Hence, one can find the choice of MCS and/or SF that maximizes $R_j(1 - \mathrm{FER})$, where R_j and FER are the data rate and frame error rate of user j corresponding to the MCS and/or SF, respectively.

In case of a TDMA system where we do not share the transmit power of the base station across multiple users in the same frame, we may consider the transmit power fixed. In this case the resulting optimum above may be considered to be the effective data rate per unit resource $\hat{R}_j(t)$.

In a CDMA system we may consider the E_b/N_0 given by:

$$\left(\frac{E_b}{N_0}\right)_j(t) = \frac{W}{R_j(t)}\mathrm{SINR}_j(t) = \frac{W}{R_j(t)} \frac{P_j(t)G_{jj}(t)}{\sum_{i \neq j} P_i(t)G_{ij}(t) + \sigma^2} \tag{2}$$

where W is the channel bandwidth and W/R_j is user j's spreading factor. As a surrogate to controlling the FER we may attempt to satisfy an E_b/N_0 target Γ_j. Let the smallest bit rate allowed (corresponding to the largest SF) be $R_{\min}$. We may think of this as a CDMA channel. Then we can define the effective data rate per channel per unit power as

$$\hat{R}_j(t) = \frac{R_j(t)/R_{\min}}{P_j(t)} = \frac{W}{R_{\min}\Gamma_j} \frac{G_{jj}(t)}{\sum_{i\neq j} P_i(t)G_{ij}(t) + \sigma^2} \tag{3}$$

What follows does not depend on precisely how $\hat{R}_j(t)$ is defined and measured. The key idea is that it will be monotonically increasing in the users' own channel gain and decreasing in interference plus noise.

Let M_b channels be available and let P_b be the power available per channel at base site b. In each frame t we have to decide what fraction $\rho_j(t) > 0$ of the resources (channels and powers) will be allocated to the different users $j \in J_b$; $\sum_{j\in J_b} \rho_j(t) \leq 1$. In which case it gets a throughput $r_j(t) = \hat{R}_j(t)\rho_j(t)P_bM_b$ (if all of the channel and power resources are given to user j, then it would get a throughput $r_j = \hat{R}_j(t)P_bM_b$). Typically there will be additional constraints on $\rho_j(t)$ depending on the technology as described below.

In a TDMA system, like GPRS or EDGE, M_b denotes the number of *time slot* channels available ($M_b = 1, \ldots, 8$). One and only one user can transmit in a time slot in a frame, so $\rho_j(t)M_b$ must be an integer. Additional restrictions on allocation of time slots to users may further constrain $\rho_j(t)$.

In a CDMA system like cdma2000 or UMTS, M_b may be the number of spreading codes (a code representing the smallest data rate allocation R_{min}) available and P_b be the power per code available, in which case M_bP_b is the total power budget at base site b. Let $f_j(t)$ be the fraction of spreading codes given to user j and let $p_j(t)$ be the fraction of per code power to be used by user j. Then $f_j(t)M_b$ will be the number of codes given to user j and must be an integer (it may even have to be a power of 2). $p_j(t)$ may be an arbitrary real number; $p_j(t)P_b$ will be the power per code and $f_j(t)M_bp_j(t)P_b$ the total power given to user j. We may require that the power per code given to user j be such that it satisfies a certain E_b/N_0 target (2). Of course we will need that $\sum_{j\in J_b} f_j(t)M_bp_j(t)P_b \leq M_bP_b$ or that $\sum_{j\in J_b} \rho_j(t) \leq 1$, where $\rho_j(t) = f_j(t)p_j(t)$.

In summary, the wireless link scheduling problem requires that in each frame t we decide what fraction $\rho_j(t) \geq 0$ of the resources (channels and powers) will be allocated to the different users $j \in J_b$; $\sum_{j\in J_b} \rho_j(t) \leq 1$, in which case it gets a throughput $r_j(t) = \hat{R}_j(t)\rho_j(t)P_bM_b$. Since $\hat{R}_j(t)$ varies with both the user and time we would like to do the scheduling in such a way that we capitalize on these variations to get high system throughput while providing some level of QoS differentiation.

3 Basic Algorithm

In the previous section we have considered the resource allocation problem on a frame-by-frame basis. This requires that the number of resources allocated to the users need be an integer. However, in the following sections as a first step of designing the scheduling algorithm we relax this constraint and consider the framework where we are interested in finding the fraction of resources to be allocated to the users over a sufficiently large period.

Given the effective data rate per unit resource $\hat{R}_j$ of the users as described in section 2 we compute the fraction of the resources that will be allocated to each user j for transmission by solving the following optimization problem:

$$\max_{\rho_j} \quad \sum_{j\in J_b} U_j(\rho_j\hat{R}_jP_bM_b) \tag{4}$$

$$\text{subject to} \qquad \sum_{j \in J_b} \rho_j \leq 1, \quad \rho_j \geq 0,$$

where $U_j(\cdot)$ is the utility function of user j as a function of the throughput it receives. The optimization problem in (4) computes the solution that maximizes the aggregate utility of the users given the resource and non-negativity constraints.

We first characterize the solution of (4) with the most commonly used utility functions of

$$U_i(r_i) = f_\alpha(r_i) = \begin{cases} \mathrm{sgn}(\alpha) \cdot r_i^\alpha, & \text{if } \alpha \neq 0,]\alpha < 1 \\ \log(r_i), & \text{if } \alpha = 0. \end{cases} \tag{5}$$

With the utility functions of $f_\alpha(\cdot)$, one can show that the solution of the optimization problem in (4) is given by

$$\rho_j = \frac{(\hat{R}_j)^{\beta-1}}{\sum_{k \in J_b} (\hat{R}_k)^{\beta-1}} \propto (\hat{R}_j)^{\beta-1}, \tag{6}$$

where $\beta = \frac{1}{1-\alpha}$. Note that if α is greater than zero, the allocation favors users with higher $\hat{R}_j$, and if α is less than zero, the allocation favors users with lower $\hat{R}_j$. The value of α equal to one leads to efficiency only solution in that all slots are allocated to the users with the highest $\hat{R}_j$, while a value of α close to $-\infty$ yields a fairness only solution in that every user receives approximately the same rate. In this sense the parameter α controls the extent to which this bias is enforced and hence how efficiency, i.e., throughput, is traded off in favor of fairness.

After computing the solution ρ^* to (4) we compute the credits C_j for the users, where the credit of user j is $C_j = \rho_j^* \cdot \hat{R}_j$. Note that the credit, C_j, of user j would be the throughput of the user normalized by $P_b M_b$ if it indeed received ρ_j^* of the resources. However, due to various system constraints a user' throughput may differ from its credit. For instance, in EDGE users are placed on one or more time slots, depending on whether they are single or multiple slot capable. A user's actual rate depends both on its ρ_j^* and time slot configuration. In a CDMA system only the users that satisfy the power budget are allowed to transmit. Incorporating these system constraints into the optimization problem leads to a weighted proportionally fair[2] (WPF) with weights $\frac{C_j}{\hat{R}_j}$ as proved in the following proposition.

Proposition 3.1 *The weighted proportionally fair rates with the weights $\frac{C_j}{\hat{R}_j}$s are also the optimal solution to the problem in (4) with the addition of the constraints in Section 2 where the rate of user j is given by $\rho_j \hat{R}_j P_b M_b$.*

Proof: See [5].

Further note that, as in weighted-fair queueing (WFQ), we only need unnormalized version of ρ_j's, which can be easily computed from (6). This is because the rates of the users depend on the ratio of their credits, but not on their exact values. This makes it easy to compute users' credits.

[2] A vector of rates r^* is said to be weighted proportionally fair with p if and only if it is feasible and for any other feasible rate vector r it satisfies $\sum_j p_j \frac{r_j - r_j^*}{r_j^*} \leq 0$.

4 The Class and Channel Condition based Weighted Proportional Fair (C^3WPF) Scheduler

In our algorithm that is described in this section we use user's credits to allocate the available bandwidth. The idea behind the algorithm is to mimic the behavior of weighted fair queueing (WFQ) without explicitly computing the virtual times for the arriving protocol data units (PDUs). The credits C_j are similar to the weights ϕ_j in WFQ. We show that our algorithm leads to weighted proportionally fair (WPF) rate allocation in the sense that the users that have the same set of bottlenecks or system constraints receive rates that are proportional to their credits.

We first consider the simple case where the channel conditions and thus the effective data rates per unit resource of the users are time-invariant so as to explain the key idea behind our algorithm. Then, we describe the actual algorithm that uses the values of current and average effective data rates per unit resource of the users.

4.1 A Simple Algorithm for Time-invariant Channels

Each user has a traffic class associated with its connection. For instance, in EDGE there are six traffic classes: conversational, streaming, interactive best-effort (I1, I2, and I3), and background best-effort. Associated with each traffic class is a weight w. This weight may reflect the price charged to the traffic class per unit time of usage [10, 11] and is used in the computation of credits in order to provide differentiated services among the traffic classes.

The scheduling algorithm described below attempts to deliver throughtputs proportional to credits $C_j = w_j \hat{R}_j^\beta$, which reflects both users' traffic classes and channel conditions. However due to additional constraints on slots, powers, codes, etc., this precise proportion may not be obtainable. We would therefore like to obtain the weighted proportional fair (WPF) solution with weights equal to the credits. In order to achieve this we use the following algorithm described below:

Let $W_j(t)$ be the total throughput of user j up to time t. Let $\bar{W}_j(t) := W_j(t)/C_j$ be the throughput normalized by credits. At time $t+1$ we sort users in increasing order of their $\bar{W}_j(t)$. In case of ties we give preference to users with higher credit values. Any further ties are resolved by ordering the users in the tie in a random manner. The scheduler then picks the user at the front of this list and schedules it for transmission in frame $t+1$. At the same time it determines the channel and power resources needed for this user. Should resources remain, it goes down the sorted list, in order, to select additional users for transmission in that frame. Users selected for transmission should obviously have data to send in that frame. Note that by favoring users with low $\bar{W}(t)$ for transmission, this algorithm tries to equalize the normalized throughputs $\bar{W}_j(t)$ over all users $j \in J_b$ as time $t \to \infty$ so as to get throughputs proportional to their credits C_j. However as mentioned earlier this may not be feasible due to additional constraints. The best achievable in that case would be the WPF throughput allocation in the sense that the users with the same system constraints would receive rates proportional to their credits. We show below that this algorithm does deliver the WPF throughputs asymptotically.

Proposition 4.1 *The average throughputs of the users, i.e., $\frac{W_j(t)}{t}$, converge asymptotically to the weighted proportionally fair rates with the weights $\frac{C_j}{\hat{R}_j}$'s as $t \to \infty$.*

Proof: See [5].

From this we have the following corollary.

Corollary 4.1 *The average throughputs of the users, i.e.,* $\frac{W_j(t)}{t}$*, converge asymptotically to the optimal rates as* $t \to \infty$.

4.2 The Actual Algorithm for Time-varying Channels

In the algorithm described above we have assumed that the channel conditions, as captured in the effective data rate per unit resource $\hat{R}_j$, do not vary with time. We now describe the actual algorithm that can take advantage of the time-varying channel conditions of the users to improve the system throughput. The key change is in the values of $\hat{R}_j$ used above and in the update equation for $\bar{W}_j$. Let $\hat{R}_j(t)$ be the current effective data rate per unit resource based on current channel conditions. Let $\hat{R}_j^{av}(t)$ be corresponding average obtained using geometric IIR filtering, i.e.,

$$\hat{R}_j^{av}(t+1) = \psi \cdot \hat{R}_j^{av}(t) + (1-\psi)\hat{R}_j(t) \quad . \tag{7}$$

The credits calculated use both $\hat{R}_j(t)$ and $\hat{R}_j^{av}(t)$ as

$$C_j(t) = w_j(\hat{R}_j^{av}(t))^{\beta} \left(\frac{\hat{R}_j(t)}{\hat{R}_j^{av}(t)}\right)^{\gamma} = w_j(\hat{R}_j^{av}(t))^{\beta-\gamma}(\hat{R}_j(t))^{\gamma} = C_j^1(t)C_j^2(t) \quad , \tag{8}$$

where $0 \leq \gamma \leq \beta$. The value of γ should depend on how accurate or reliable $\hat{R}_j(t)$'s are. We use different factorizations of $C_j(t)$ into $C_j^1(t)$ and $C_j^2(t)$ to construct different scheduling algorithms.

Let $D_j(t)$ be the amount of data transmitted in frame t for user j. $\bar{W}$ is updated as per the following algorithm.

$$\bar{W}_j(t+1) = \phi \cdot \bar{W}_j(t) + (1-\phi)\frac{D_j(t)}{C_j^1(t)} \tag{9}$$

In keeping with the algorithm described earlier, at the beginning of time frame $t+1$, we sort users in increasing order of $\bar{W}_j(t+1)/C_j^2(t)$ and select users for transmission based on available resources.

Currently we have three different scheduling algorithms:

- **Variant 1**: For this we use $C_j^1(t) = C_j(t)$ and $C_j^2(t) = 1$. Note that this is based upon the algorithm analyzed earlier.
- **Variant 2**: Here we use $C_j^1(t) = w_j(\hat{R}_j^{av}(t))^{\beta}$ and $C_j^2(t) = (\frac{\hat{R}_j(t)}{\hat{R}_j^{av}(t)})^{\gamma}$.
- **Variant 3**: Here we use $C_j^2(t) = C_j(t)$ and $C_j^1(t) = 1$.

Note that choosing the credits as described above does two things. Since the effective data rate per unit resource $\hat{R}_j(t)$ is time varying, it uses an estimate $\hat{R}_j^{av}(t)$ for its *average*. The estimate is based on an IIR filter; other estimates may easily be substituted. If we set $\gamma = 0$ above, then we will simply be biasing the throughput in proportion to this average (raised to the power β). However, by dividing by non-unity values of the factor $C^2(j(t)$ (in variants 2 and 3), we tend to favor those users that have better channel conditions relative to their own average conditions. This second idea is also exploited in the scheduling algorithms proposed by Holtzman [7], and Jalali *et al.* [9]. In fact the algorithm proposed is similar to the Variant 3 proposed above with $\beta = \gamma = 1$. One major difference is that they use a time-average computation of $\bar{W}$ instead of the IIR filter estimator in (9). The time constant in the IIR filter estimator for the average must be chosen depending on how frequently channel condition measurements are available and the time constants involved in the channel fluctuations (distance based path loss, shadow fading, and fast fading).

Among the three variants we expect Variants 2 and 3 to out-perform Variant 1. A close inspection of Variant 1 reveals that the current channel conditions are only reflected for users who get scheduled whereas the in the other two Variants the relative gains of all the users come into play (with a non-unity $C_j^2(t)$ factor).

5 Performance Analysis

In this section we describe our experiments with the proposed scheduling algorithm. In addition to evaluating the performance of the algorithm in a simplistic setting, we are also interested in comparing the performance of a scheduling algorithm that is aware of the channel conditions with a scheduling approach that does packet-level scheduling separate from radio-resource allocation. This would help us document the gains we get with the scheduler being aware of the radio conditions. Henceforth in this document we refer to the class of schedulers that are not aware of channel conditions as split-schedulers. The specific split-scheduler that we consider is one that tries to equalize the data rates that all the users get. Note that this can be easily achieved in the framework of our algorithm by setting $\beta = 0$. This is equivalent to a WFQ scheduler with equal weights. Throughout the performance analysis section we concentrate on an EDGE/GPRS system.

5.1 Simulation Set-up

In this sub-section we describe the set-up for our performance analysis. First we describe the physical constraints, and thereafter the network-level characteristics like traffic sources, fragmentation and ARQ-mechanisms.

We consider 7 cells, which use the same carrier and the same sector and 9 mobiles in each cell. The positions of the mobiles in each cell are chosen at random, spatially uniform within each cell. The distances between the cells can be calculated from the reuse pattern. The reuse pattern considered is a $4/12$ reuse pattern which means we have 3 sectors per base-site and a reuse factor of 4. From this we can find the matrix of distances between each of the base-sites and mobiles. Once this is done one can calculate the distance-based propagation loss term and from the earlier number the propagation loss between the base stations can also be determined. In our experiments the propagation loss model is such that at distance of one meter we get a gain of 1. For each mobile we then determine a log-normal shadow-fading term which has two terms, one, depending on the transmitting base-site, and the other independent across mobiles. We also introduce a time-correlation in the second term by modulating it using a Markov process. Whenever the Markov process changes states we choose an independent gain value, which is then held fixed till the next transition. The transition probability between the states is a function of the speed of the mobile. The speed also fixes the Doppler frequency for each mobile, which is then used to generate the fast-fading gain coefficients for each radio block [8]. This helps us fix the propagation model. The noise term includes receiver noise, interference from cells outside the first ring and adjacent channel interference. The interference from other cells is calculated accounting for the effects of sectorization assuming a 20dB loss for the back-lobe. The power from adjacent channels is assumed to suffer a 20dB loss as well. These quantities allow us to determine the SINR at each of the receivers. The modulation and coding scheme is chosen based on link curves (SINR to FER). We use a link adaptation mechanism that assigns to each user the MCS with the best effective data-rate such that the FER is below 0.1. The same link curves are used to determine the accuracy of transmission. We use a $\bar{W}$-update algorithm in (9) based only upon the average credit value and use exponential averaging for the update with $\phi = 0.02$. The average effective data is not updated using an averaging mechanism but we use just the shadow fading terms and the propagation loss terms to calculate this at regular points in the simulation. The interval between two successive recalculation points is assumed to be 480 frames or 1.2 seconds. During this time we hold the average rate fixed at the number calculated at the beginning of this interval. The parameters used in the simulation are summarized in the Table 1.

We use *ns-2* [1] to generate the traffic for our simulation experiments. This gives us the flexibility of trying various traffic sources like TCP-based traffic or a UDP flow as well as allows us to monitor the performance at all layers of the protocol stack. At each radio network controller[3] the packets from *ns-2* are stored in separate queues; one for each mobile attached. We assume that we get instantaneous feedback for the transmissions on the wireless channel and the erroneous frame is immediately put back at the head-of-the-line for transmission. Thus, we are assuming that our ARQ mechanism transmits packets till they are correctly received. The ARQ mechanism also reacts instantaneously to MCS changes recommended by the link-adaptation algorithm fragmenting in a way such that it transmits the right number of bits in each frame. The protocol stacks at the implemented components are shown in Figure 1.

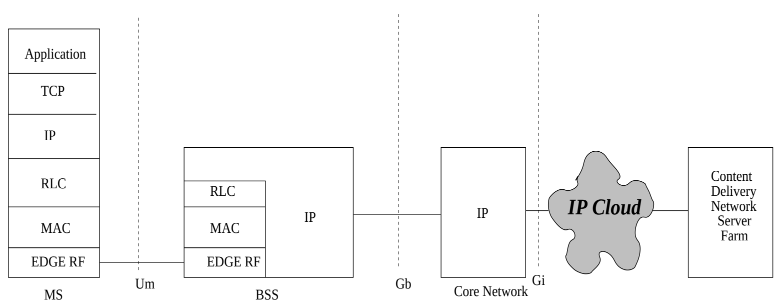


Figure 1: Implemented protocol stacks

5.2 Experiment 1

In our first set of experiments we restrict our attention to infinitely-backlogged sources transferring data to the mobiles. This would help us document the performance of the scheduler without any effects that higher level protocols or the on-off nature of traffic might introduce. In this set-up we try $\beta = 0$, 1, and 2, which would allow us to compare the split-scheduler with a scheduler that uses $U(x) = \log(x)$ ($\beta = 1$) as the utility function and another that uses $U(x) = \sqrt{x}$ as the utility function ($\beta = 2$). We keep the number of mobiles fixed at 9 per cell making it a total of 63 users. We assume that only 4 time-slots out of the 8 possible are available to the EDGE component of the system. The slot numbers are assumed to be the same across all

[3]The radio network controller is equivalent of MSU in cellular telephony systems.

Parameters	Values
Reuse Pattern	$4/12$
Cell Radius	1 Km
Back-lobe loss due to sectorization	20dB
Adjacent channel power loss	20dB
Propagation Loss Model	$36 \log 10(d)$dB with d in meters
Log-normal shadowing variance	8dB
Fast-fading model	Jakes' simulator with 30 interferers - depends on speed
Transmit power - Mobile	10 Watts
Transmit power - Base-site	100 Watts
Speed	Uniform in $[0, 60]$ mph
Doppler Frequency	In $[0 - 100]$ Hz depending on speed
Receiver noise level - Mobile	-116dBm
Receiver noise level - Base-site	-120dBm

Table 1: System parameters used for simulations.

	$\beta = 0.0$	$\beta = 1.0$	$\beta = 2.0$
W/O slot constraints no TCP	874 Kbps	1000 Kbps	1125 Kbps
W/O slot constraints with TCP	827 Kbps	1003 Kbps	1134 Kbps
W slot constraints with TCP	909 Kbps	1064 Kbps	1189 Kbps

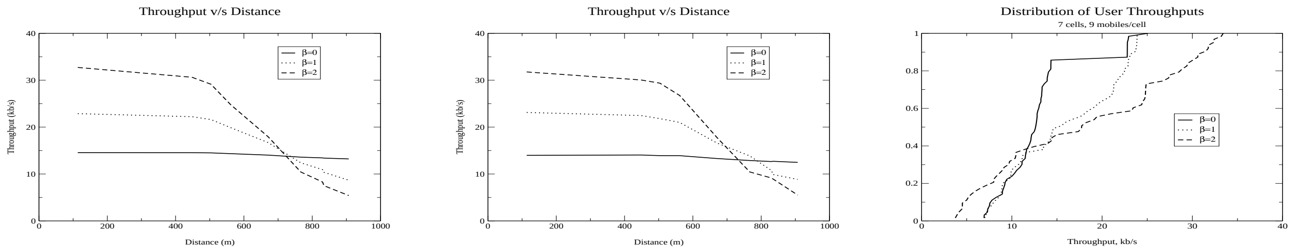
Table 2: Comparison of total system throughput with and without TCP traffic.

the cells. The rest are assumed to be used by voice. We consider a system where all the 9 users can transmit on all the available slots. Each frame is 2.5 milliseconds long and we run the simulations for 100 to 150 seconds which translates to $20,000$ and $30,000$ frames available to EDGE respectively. The answers we seek are the rate-distance profiles in each cell as well as sum throughput in each cell.

As described earlier we assume that all mobiles can use all the available slots. This would be a system with no extra constraints and we except the results to follow the analysis in Section 3 except for effects of the time-varying channel. The total system throughput tells us how many bits are served. The larger this is the more efficient the system. On the other hand a throughput-distance profile gives an indication of the fairness; the less skewed the profile the more fair the system is. Comparing the total system throughputs in Table 2 we have that the system with $\beta = 2$ is the most efficient and the system with $\beta = 0$ is the least efficient. From Figure 2(a) we can see that $\beta = 0$ is most fair and $\beta = 2$ gives much higher throughput to users close to the base-site. The important point to notice is that the sacrifice in throughput for the users at the edge is much smaller than the gains for the users close to the center.

5.3 Experiment 2

So far we have only considered infinitely backlogged sources and have not incorporated any protocol or traffic dynamics in the performance analysis. In this series of experiments we assume that the traffic is generated by open-ended *ftp* sources with TCP-reno as the transport layer protocol. Since TCP requires acknowledgments to be delivered back to the source we implement the same scheduler and ARQ mechanism on the uplink as well. The rest of the set-up is exactly the same as in Section 5.2.



(a) Throughput-distance profile for one cell with different utility functions and UDP traffic.

(b) Throughput-distance profile of TCP traffic for one cell with different utility functions.

(c) Distribution of TCP throughput across all users in system with different utility functions.

Figure 2: Performance of algorithm without and with TCP traffic.

For the first simulation run we assume that all mobiles can use all the available slots. We except the results to follow the analysis in Section 5.2 except for effects of the time-varying channel. Comparing the total system throughputs in Table 2 we again have that the system with $\beta = 2$ is most efficient and the system with $\beta = 0$ is the least efficient. As expected throughput with TCP is very close to the infinitely-backlogged case. From Figures 2(a) and 2(c) we can see that $\beta = 0$ is most fair and $\beta = 2$ gives much higher throughput to users close to the base-site.

In our second simulation run we consider the system with slot constraints described. The first three users are 2-slot capable and the rest are only single slot capable. These experiments will help us validate the WPF algorithm on a time-varying channel as well as help us explore the impact of slot-restrictions.Table 2 also displays the total system throughput values for this case. The answers we seek are the $\bar{W}$ values for the mobiles in one cell and corresponding throughput values.

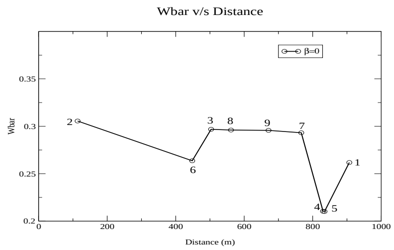
The $\bar{W}$ values for the different β values are shown in Figures 3(a), 3(b), and 3(c). From the problem formulation we expect the users to split-up into partitions such that each partitions shares a unique set of slots disjoint from the slots of the other partitions. Within the slots associated with each partition the users in each partition obey their slot restrictions as well. From a close inspection of the $\bar{W}$ values we find that the users fall into the following partitions:

- $\beta = 0$: $\{2, 3, 7, 8, 9\}$, $\{1, 6\}$, and $\{4, 5\}$;
- $\beta = 1$: $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$; and
- $\beta = 2$: $\{1, 4, 5\}$, $\{2, 6\}$, and $\{3, 7, 8, 9\}$.

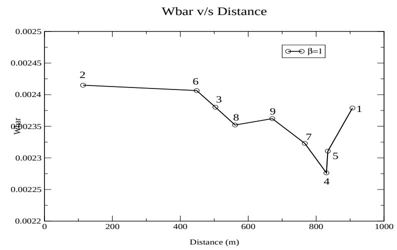
The numerical calculation of the weighted proportionally fair allocation assuming a static channel, with the rates $\hat{R}$ given by the averages obtained from simulation above, yields the same partitions. The throughputs obtained from the simulations are also in agreement with the numbers computed using analysis. A typical rate-distance profile is shown in Figure 4. This experiment reinforces our belief that this algorithm will be well-behaved even with time-varying channels.

Another interesting means to see the WPF solution is to look at the usage of the slots by the various users. In Table 3 we list the slot usage for the different users for the case of $\beta = 0$. Different partitions are indicated by different fonts. It is clear from this that after an initial transience period the users settle into the partitions described earlier.

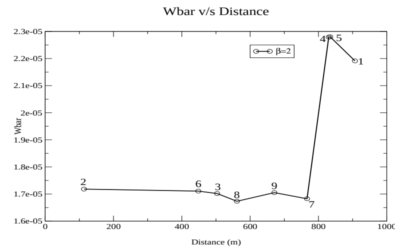
Figures 5(a), and 5(b) display the evolution of the TCP congestion window and the TCP round-trip time estimate for the



(a) $\bar{W}$ values for different users in a cell with $\beta = 0$.



(b) $\bar{W}$ values for different users in a cell with $\beta = 1$.



(c) $\bar{W}$ values for different users in a cell with $\beta = 2$.

Figure 3: $\bar{W}$ values for different utility functions with slot constraints.

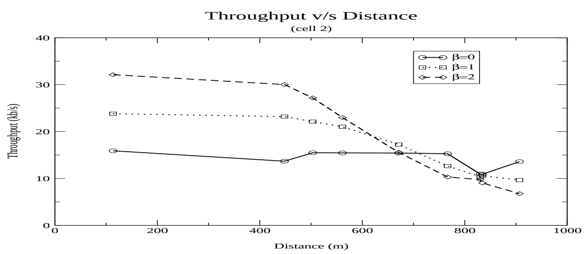
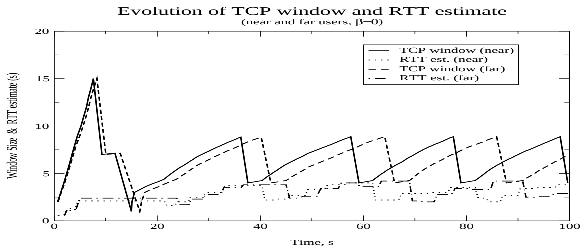


Figure 4: Comparison of throughput for various users with slot constraints for different utility functions.

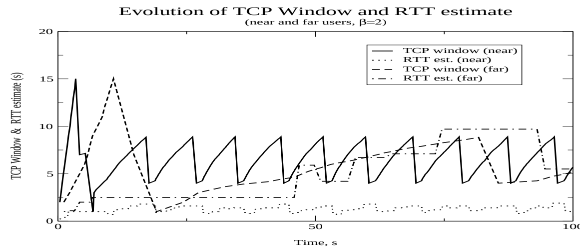
nearest and farthest users in one particular cell for different values of β. The first thing to note is that owing to the lossless ARQ mechanism there is one timeout after which the congestion window evolves as in steady state. This is also a feature of the ARQ mechanism in the EDGE standard. Thus, a similar behaviour can be expected unless the file size is too small or the buffer size at the RNC is too small. The main thing to note is that the period of the window size evolution depends on the throughput and hence is very similar for near and far users in the $\beta = 0$ case. In contrast for $\beta = 2$ the period of far user is much larger than that of near user. Note that as expected the maximum congestion window values are determined by the receiver window size.

Mobile	Slot 1	Slot 2	Slot 3	Slot 4	$\bar{W}$
2	0	207	1136	0	0.304611
6	*0*	*1306*	*0*	*0*	*0.248013*
3	0	0	591	1023	0.311836
8	0	0	0	1658	0.313796
9	0	0	0	2157	0.310027
7	0	0	3110	0	0.300173
5	**2282**	**0**	**0**	**0**	**0.208691**
4	**2469**	**0**	**0**	**0**	**0.207404**
1	*195*	*3432*	*0*	*0*	*0.247868*

Table 3: Slot usage numbers for the different users for $\beta = 0$.



(a) Congestion window and round-trip time estimate evolution for the nearest and farthest user in one cell with $\beta = 0.0$.



(b) Congestion window and round-trip time estimate evolution for the nearest and farthest user in one cell with $\beta = 2.0$.

Figure 5: Congestion window and round-trip time estimate evolution for different utility functions.

6 Conclusion

We have proposed a scheduling algorithm that provides a flexible means of trading off efficiency for fairness as well as a flexible way of exploiting temporary fluctuations in channel conditions. Fairness trade-off is based on the utility optimization with appropriate choices of utility functions and thus β parameters. The exploitation of the variation in channel conditions is based on biasing the algorithm in favor of users with better relative current channel condition.

The analysis shows that with an appropriate choice of utility function a substantial gain in system throughput can be achieved while maintaining reasonable fairness amongst the users. This improvement in system throughput is obtained with both TCP and UDP connections.

Acknowledgement

We thank Juan Alvarez, and Yiannis Argyropolous for their help in developing the software used to obtain the simulation results.

References

[1] NS network simulator. Available at http://www.isi.edu/nsnam/ns/.

[2] RFC 1633. Integrated Services in the Internet Architecture: an Overview. June 1994.

[3] RFC 2475. An Architecture for Differentiated Service. June 1999.

[4] RFC 2597. Assured Forwarding PHB Group. June 1999.

[5] R. Agrawal, A. Bedekar, R. La, and V. Subramanian. Analysis of a class and channel condition based weighted proportional fair scheduler. $Document\ in\ preparation$.

[6] Randall Berry. *Power and Delay Trade-offs in Fading Channels*. PhD thesis, MIT, Cambridge,MA, 2000.

[7] J. M. Holtzman. CDMA Forward Link Waterfilling Power Control. Preprint, May 2000.

[8] W. C. Jakes. *Microwave Mobile Communications*. Wiley-Interscience, 1974.

[9] A. Jalali, R. Padovani, and P. Rankaj. Data Throughput of CDMA-HDR a High Efficiency - High data Rate Personal Communication Wireless System. In *Proceedings of IEEE VTC 2000*.

[10] Frank Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[11] Richard J. La and Venkat Anantharam. Charge-sensitive TCP and rate allocation in the Internet. In *Proceedings of IEEE INFOCOM 2000*.

[12] Rangsan Leelahakriengkrai. *Scheduling in Multimedia CDMA Wireless Networks*. PhD thesis, University of Wisconsin, Madison,WI, 2001.

[13] S. Nanda, K. Balachandran, and S. Kumar. Adaptation Techniques in Wireless Packet Data Services. *IEEE Communications Magazine*, Janunary 2000.

[14] X. Qui, K. Chawla, L. Chuang, N. Sollenberger, and J. Whitehead. RLC/MAC Design Alternatives for Supporting Integrated Services over EGPRS. *IEEE Personal Communications Magazine*, 7(2):20–33, April 2000.

[15] S. Shakkottai and A. Stolyar. Scheduling for Multiple Flows Sharing a Time-Varying Channel: the Exponential Rule. Preprint, January 2001.

[16] D. Tse. Forward-Link Multiuser Diversity Scheduling. submitted for publication to IEEE JSAC.