Noname manuscript No.
(will be inserted by the editor)

Scheduling and Rate Assignment Algorithms For Policy Driven QoS Support in High Speed Cellular Networks

Joseph S. Gomes, JungKyo Sohn, Mira Yun, Jae-Hoon Kim, Hyeong-Ah Choi, and Hyeong In Choi

the date of receipt and acceptance should be inserted later

Abstract Due to the growing popularity of high data-rate multimedia services such as streaming, gaming, and video conferencing using mobile devices, providing strict QoS support in cellular networks is becoming crucial. Moreover, differentiated service requirements and service plans promote service providers to enforce their own policies aligned with their goal of maximizing revenue in meeting these QoS requirements. While network service providers may desire to impose operational policies on the packet schedulers, existing schedulers are not capable to deal with such challenges due to many factors including time varying channel conditions, constraints on the total transmit power allocated to forward links (i.e., base station to mobile stations), and lack of policy abiding fast scheduling algorithms. In this paper, we propose a QoS-aware packet scheduling algorithm that takes into account policy rules that govern the relationships between different QoS classes while jointly achieving throughput maximization and fairness. Our scheduler uses marginal utility functions, representing rates of changes in utility, defined to embody the given rules in the form of guaranteed rates specified by user applications. The objective of the scheduler at any given time is to maximize the total utility over all users. Our scheduling algorithm is implemented in the HSDPA module that we have developed in OPNET. Simulation results show that unlike other schedulers, our algorithm comply with the policy constraints if allowed by radio conditions and cell capacity. We also show that the total utility under our scheduling algorithm converges to the maximum utility value.

1 Introduction

Radio resource management (RRM) algorithms such as packet scheduling, rate assignment, admission control, and mobility management for wireless mobile networking environments have recently been in the forefront of wireless network research. RRM in such environments is a big issue because of the capacity-constrained and highly dynamic nature of wireless networks. If the recent proliferation of mobile applications with high bandwidth requirement is any indication, RRM algorithms would have to play a vital role to satisfy the communication demands of the users in future. The job of a RRM scheme or protocol is to define how a common resource of high demand, such as the radio channels, are distributed among competing users to provide user satisfaction as well as enhance the overall performance of the system.

There are three main aspects to RRM, namely admission control, packet scheduling and rate assignment. In this paper, we are mainly concerned with the latter two. RRM for CDMA based networks with dedicated

J. Gomes is with the Department of Computer Science, Bowie State University, Bowie, MD 20715, USA. e-mail: joe.sgomes@gmail.com.

J. Sohn and H. I. Choi are with the Department of Mathematics, Seoul National University, Seoul, Korea. H. I. Choi also holds a joint appointment with Research Institute of Mathematics, Seoul National University. e-mail: {jgsohn,hichoi}@snu.ac.kr

M. Yun and H. A. Choi are with the Department of Computer Science, George Washington University, Washington, DC, USA. e-mail: {mirayun,hchoi}@gwu.edu

J. Kim is with the Department of Industrial & Information Systems Engineering, Ajou University, Suwon, Korea. e-mail: jayhoon@ajou.ac.kr

This work was in part supported by the Access Network & Mobile Terminal R&D Center, SK Telecom, Seoul, Korea.

Address(es) of author(s) should be given

channels such as UMTS and CDMA1x has been well investigated, primarily with respect to adaptive rate control with the help of cross layer communication. References [1], [2] and [3] consider the effects of the network characteristics from physical layer to network layer on RRM. Xu et al. proposed a rate-assignment algorithm in [4] using both the traffic information in the link layer and the adaptivity of CDMA physical layer. In [5], the traffic fluctuations and the variations of the system interference are considered for adaptive scheduling in UMTS. An adaptive rate scheduler, based on the predictive and feedback-enhanced reactive adaptation control, was presented in [6]. However, for CDMA based networks with time-slotted shared downlink channels such as HSDPA and EvDO, an additional dimension is added to the problem, namely the time dimension. This implies that for such networks the RRM schemes must be able to efficiently manage resources during each time-slot using packet scheduling.

Most of the proposed scheduling algorithms address mainly three design goals: throughput maximization, fairness, and quality of service. Maximizing throughput is critical to support high data rate applications, e.g., web browsing, video and audio streaming, video conferencing and remote transfer of large files. In multiuser cellular systems, opportunistic scheduling can be used to maximize system throughput by utilizing multiuser diversity and favoring the users with peaks in their channel conditions [7, 8]. This can be further enhanced by using MIMO techniques to exploit multiplexing gains to dramatically increase the data rate without increasing required bandwidth [9–11]. However, opportunistic scheduling does not prevent certain users from being starved. On the other hand, the fairness goal ensures that all users get a fair share of the resource and nobody is starved. Fairness issues in scheduling have been studied in depth in [7, 12, 13]. *Proportional Fair* (PF), the most popular packet scheduling algorithm considering fairness, has been thoroughly investigated in [14–16].

A vast amount of research has been conducted on QoS-aware radio resource management in CDMA networks [4, 17–23]. In [19], Huang and Zhuang proposed a QoS based fair resource allocation scheme which combines packet scheduling and power assignment. A QoS-oriented packet scheduling algorithm was presented in [21] to maximize the number of served users under QoS constraints. In [22], the notion of QoS for a real-time user was proposed in the following way: the QoS requirement of user i is $Prob\ \ \{W_i > T_i\} \leq \delta_i$, where W_i is a packet delay for user i, and parameters T_i and δ_i are the delay threshold and the maximum probability of exceeding it, respectively. It proposes a *throughput optimal* scheduling algorithm MLWDF that supports QoS definitions of the above form. A different notion of QoS where the requirement is that the average throughput r_i provided to user i must not fall below a predefined value $r_{i_{thresh}}$, was also addressed in [22]. Exponential utility functions or barrier functions were used in [23, 24] for rate sensitive services.

1.1 Our Contributions

QoS enforcement schemes in the existing literature is based solely on experienced throughput and is oblivious to channel conditions. Moreover, very little work has been done on differentiated QoS support. Differentiated QoS support will become increasingly important as multi-rate applications with different rate requirements as well as various types of real-time and non real-time service classes with different traffic characteristics become more prevalent. In addition, the packet and rate schedulers must also adapt to the time-varying location and mobility dependent channel issues. In other words, an intelligent scheduling and rate assignment (SRA) algorithm should be designed with due consideration to all the realistic constraints such as throughput maximization, fairness, differentiated QoS, and channel adaptation. However, in a shared-medium wireless network the goals of fairness, throughput maximization and differentiated QoS are in conflict with each other and finding a balance among them can be particularly challenging. This is where our work comes in. In this paper, we provide scheduling algorithms that take advantage of the adaptive modulation and coding built in high speed cellular networks and jointly address all of the above issues such as QoS differentiation, throughput maximization, and fairness.

First, we provide a general scheduling and rate assignment framework that uses user-specific marginal utility functions to represent user preferences and system policies. The marginal utility values and the current achievable data rates for the users are used to determine the transmitting users and their data rates in each time slot. We then prove that utility achieved under our proposed algorithm converges to the maximum possible utility over a fixed resource domain. We introduce the notion of policy driven QoS-support where a priority value is used to differentiate the requirements imposed by the separate user services and a set of system policies is defined to balance QoS, fairness and throughput maximization. The goal of our policies is to provide differentiated service during times of overload and to fairly distribute surplus capacity when the total system load do not exceed the system capacity. We develop a marginal utility based algorithm, SPS, aligned with the aforementioned framework and in accordance with the system policies. This is followed by a more relaxed policy based algorithm, LPS, that tries to find a balance between throughput maximization and fairness.

For experimentation, we have developed an HSDPA system by extending the UMTS model in OPNET simulator. Our system includes all the key entities of the network such as MS, BS, RNC, and SGSN with most of the functionalities and aspects of HSDPA at all the protocol layers, e.g. shared data channels, adaptive modulation and coding, various MS categories, hybrid automatic repeat requests etc. We experimentally show that our SPS and LPS algorithms very effectively satisfy high priority users by guaranteeing their QoS requirements if it is at all possible, and perform better than other well-known algorithms such as Max C/I and PF, especially under stressful conditions.

2 CDMA Background and System Model

We consider a typical hierarchical cellular structure consisting of all the key entities of the network such as mobile stations (MS), base stations (BS), radio network controllers (RNC), and serving GPRS support nodes (SGSN). The uplink and downlink communication are separated from each other by being in different frequency bands. Here, we focus on downlink data transportation. Below we discuss, various aspects of our system model that were implemented in our OPNET simulator for HSDPA.

Downlink Communication using Shared Channels: Instead of dedicated channels, high speed cellular networks such as HSDPA and EvDO use shared wireless downlink channels for high speed data. A shared downlink transport channel carries data to the selected MSs during each time-slot of α milliseconds. The transported bits from the transport channel are mapped onto physical downlink shared channels, each using a separate orthogonal CDMA *code channel*. A code channel is created using a *chipping sequence* or *spreading sequence* (possibly a Pseudo-Noise sequence) which uniquely identifies the channel. A higher number of codes result in a higher data rate.

Adaptive Modulation and Coding: In the uplink direction, the MSs notify the BS of the channel condition using a Channel Quality Indicator (cqi) and a positive or negative acknowledgement pertaining to the received frame. cqi indicates the instantaneous channel quality experienced by the user, so that the BS can adjust its transmission parameters (modulation type, coding rate, number of codes) to cope with variations in channel conditions. The cqi reported by the MS corresponds to transmission parameters that would result in the maximum data rate possible while providing an acceptable bit error rate (BER) for the current link conditions.

MS Categories: The MSs are divided into many (12 for HSDPA) different categories based on their capabilities. For example in HSDPA, MS with category $c = 1$ or 2 can only support data rates upto 1.2 Mbps using 5 simultaneous physical channels (codes) and has minimum inter time-slot interval $min\_iti(c)$ of 3. If user ms_i from category c_i is scheduled to transmit at t, the earliest time ms_i can be scheduled next is $t + min\_iti(c_i)$. Category 7 can theoretically (under perfect link conditions) support upto 7.21 Mbps using 10 codes and has min_iti of 1.

Hybrid Automatic Repeat Request: Hybrid automatic repeat request (HARQ) is a technique where MS stores previous transmissions that are in error in soft memory to be combined with future re-transmissions for decoding. For each packet the MS sends HARQ feedback (ACK or NACK) to inform the BS whether a retransmission is required or not. HARQ uses one of two different schemes, Chase combining where retransmissions are identical or incremental redundancy where retransmissions are not identical.

3 Existing Scheduling Algorithms

There are several classical algorithms from the literature such as the round robin (RR) algorithm, the maximum carrier-to-interference (C/I) algorithm and the proportional fair (PF) algorithm [12]. RR is a popular reference algorithm that provides a high degree of fairness between the users, but at the expense of the overall cell throughput. On the other hand, the maximum C/I algorithm maximizes cell throughput but makes no effort to provide fairness among users. It chooses the the receiver with the best channel condition, i.e. the highest SINR or carrier to interference ratio. The PF scheme offers a good trade-off between user fairness and achievable cell throughput. PF scheduler orders the receivers using the metric $\frac{r_i(t)}{x_i(t)}$ where $r_i(t)$ is the instantaneous data rate and $x_i(t)$ is the current throughput. PF does not make any QoS guarantees. It considers all users to be equally important and adds a fairness property with respect to user throughput.

4 Analysis of a Marginal Utility based Scheduling and Rate Assignment Algorithm

In this paper, we will focus on policy driven scheduling to provide differentiated quality of service among prioritized classes of users that guarantee user satisfaction rather than merely addressing throughput maximization or fairness. Such policies are valuable to network operators as they are conducive to achieving their operational goals of increasing user satisfaction while efficiently managing system resources, i.e. maximize throughput subject to both policy and resource constraints. Before getting into complex policy constrained SRA, we describe a framework that uses marginal utility functions for developing policy constrained SRA algorithms for code-multiplexed downlink transmission.

Utility measures are common in economics for comparing relative values of various goods and services. Similarly, in decision analysis, higher utility values are attributed to actions associated with more desirable outcomes. When encountering several alternatives, the choice with the highest utility is always selected. In other words, a utility function is a convenient way to represent preferences and indirectly reason about preferences. For our purposes, we will develop marginal utility functions, which represent rate of change in the utility function, to fit the policy constraints. Each user will be assigned a marginal utility value that determines the schedulability of the user according to the policy constraints. At any given time, the objective of the scheduler will be to maximize the total utility over all users.

Below, we will first present a standard discrete model for throughput, then transform it into a continuum model using an ordinary differential equation (ODE) and provide a solution in the sense of Filippov's. The throughput model is important as our final policies are based on throughput and consequently our utility functions are functions of throughput or data rate. We follow this with a discussion of our total utility maximization problem for code-multiplexed downlink SRA. We will show that under certain conditions, for any non-increasing and continuous marginal utility function of throughput, a greedy algorithm exists that maximizes the total utility value.

4.1 Throughput Model

Suppose that there are n users (MS), $ms_1, \cdots, ms_n$, served by a single BS. The BS is allocated M orthogonal channel codes. The maximum instantaneous data rate for ms_i that can be used at t is given by $r_i(t)$, where $r_i(t) \leq r_i^{max}$. r_i^{max} is a predefined upper-bound for $r_i(t)$, which depends on the MS category of ms_i. Note that we assume code multiplexing whereby the base station can transmit packets to multiple users during the same time-slot. The BS may use any MCS less than or equal to the one corresponding to the reported cqi. This translates to an effective data rate, which is a fraction of $r_i(t)$ and can be represented as $z_i(t)r_i(t)$, where z_i $(i = 1, \ldots, n)$ is a fractional value between zero and one chosen by the scheduler.

4.1.1 Standard discrete model for throughput

Denote by t_k $(k = 0, 1, \ldots)$ the k-th time slot and by τ the size of the time window for throughput measurement. Let $x_i(t_k)$ be the throughput for user i at time slot t_k. Since a typical scheduler S does not have a long memory, the throughput x_i is estimated recursively as follows:

$$x_i(t_{k+1}) = x_i(t_k) - \alpha x_i(t_k) + \alpha r_i(t_k) z_i(t_k), \qquad (1)$$

where $\alpha = \frac{1}{\tau}$ is the length of time slot and $z_i \in [0, 1]$ is a decision variable for ms_i, the value of which is determined by scheduler S. Equation (1) indicates that due to the window shifting by one time slot, a portion of the throughput, $\alpha\, x_i(t_k)$, is lost, while the amount $\alpha r_i(t_k) z_i(t_k)$ is gained for receiving at a non-zero rate of $r_i(t_k)$ at t_k .

4.1.2 Continuum model

Equation (1) can be rewritten as:

$$\frac{x_i(t_{k+1}) - x_i(t_k)}{\alpha} = r_i(t_k) z_i(t_k) - x_i(t_k).$$

This leads to the following continuum version

$$\frac{d}{dt} x_i(t) = \lambda_i(t) - x_i(t) \quad where \quad i \in \{1, \ldots, n\}, \qquad (2)$$

where x_i and $\lambda_i(t) = r_i(t) z_i(t)$ are considered to be continuous.

The job of an SRA algorithm is to determine $\boldsymbol{\lambda}(t) := (\lambda_1(t), \ldots, \lambda_n(t))$ at each time t, where $\boldsymbol{\lambda}(t) \in D_\lambda(t) \subset [0, r_1(t)] \times [0, r_2(t)] \times \cdots \times [0, r_n(t)]$. We will call the domain $D_\lambda(t)$ a *resource domain* for $\boldsymbol{\lambda}(t)$ at time t. A user ms_i is not scheduled at time t if $\lambda_i(t) = 0$. However, when $\boldsymbol{\lambda}$ is not continuous, how can one guarantee the existence of a solution to equation (2). It turns out that under some mild conditions on the resource domain $D_\lambda(t)$ and $\boldsymbol{\lambda}$, one can always be sure of the existence of a solution to equation (2). This is further elaborated in Appendix A.

4.2 Marginal Utility based Scheduling and Rate Assignment Algorithm

Here, we propose a utility based SRA algorithm $\boldsymbol{\lambda}$. Let $u_i(x) : [0, \infty) \longrightarrow \mathbb{R}_+$ be a marginal utility function of throughput x for each user ms_i, which is assumed to be non-increasing and continuous in keeping with *the law of diminishing returns*. In other words, as the throughput increases incremental utility gained from the increased throughput diminishes. Design and behavior of individual marginal utility functions depends on the policies, goals and constraints of a particular application. The integrated value $U_i(x_i) := \int_0^{x_i} u_i(x)dx$ is the total utility experienced by ms_i when throughput x_i is experienced. The utilities are comparable across users, although the difference in magnitude of utility for a single user has little significance. U_i is concave because the integrand u_i is non-increasing. Let $D_x \subset \mathbb{R}^n_+$ denote a compact domain of U, which contains all theoretically possible throughput configurations. We will call the domain D_x a *throughput domain* to make it distinguishable from resource domain D_λ.

Definition 1 Given a throughput configuration $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ of n users, the total utility $U : D_x \longrightarrow \mathbb{R}$ is defined by summing all the individual utilities, i.e.,

$$U(x_1, \ldots, x_n) = \sum_{i=1}^{n} \int_0^{x_i} u_i(x)dx. \tag{3}$$

It is easy to see that for a given $\mathbf{x}$, the total utility function U is concave and non-decreasing. Since D_x is compact and U is concave, U has a unique maximum over D_x. Now the question is how can we design an SRA algorithm that determines $\boldsymbol{\lambda}_t$ at time t so that the total utility U is maximized. Below, we present our proposed SRA algorithm $\boldsymbol{\lambda}_\infty$ that solves this utility maximization problem.

Algorithm 1 SRA algorithm $\boldsymbol{\lambda}_\infty$

```
Let $r_i(t)$ be the maximum achievable transmission rate for user $ms_i$ at time $t$, $M_sys$ be the total bandwidth capacity, and $C_{rem}$ be the remaining available capacity.
while ($C_{rem} > 0$) do
    Let $ms_i$ be the user such that $i = \arg\max_{j}\{U_j(r_j(t))\}$
    if ($r_i(t) < C_{rem}$) then
        Assign $r_i(t)$ to $ms_i$
    else {cannot support this user's max rate}
        Assign rate equivalent to $C_{rem}$ to $ms_i$
    end if
    $C_{rem}$ -= $r_i$
end while
```

The next theorem states that the resulting total utility $U(\mathbf{x}(t))$ under the SRA algorithm $\boldsymbol{\lambda}_\infty$ actually converges to the maximum of U over D.

Theorem 1 *Let D be a compact subset of $\mathbb{R}^n$. Let U_i be a set of differentiable and concave functions. Let M be the maximum value of total utility U over D. Then total utility $U(\mathbf{x}(t))$ under the solution $\boldsymbol{\lambda}_\infty$ converges to M. The time complexity of the algorithm is $O(n \log n)$.*

The proof of this theorem is provided in Appendix B.

5 Strict Channel Allocation Policy for Rate-sensitive services

In this section, we present a set of strict policies that are in sync with the interests of a network operator, such as maximizing revenue and improving user satisfaction, while efficiently managing system resources. The problem is to maximize throughput subject to both policy and resource constraints.

Suppose a service provider differentiates its users using Scheduling Priority Indicator (SPI) values, where $SPI \in \Pi = \{1, 2.., 15\}$ [25]. In general we assume higher classes (with higher SPI) have higher requirements of guaranteed bit rates or *gbr*. Although the problem can be generalized for any number of SPI values, for simplicity we will consider only SPI values, 1 and 2. We will call these two classes *Gold* and *Silver* classes, respectively with corresponding *gbr* values gbr_g and gbr_s.

We now list the notations that we will be using:

Notation	Meaning
p_i	ms_i's priority class
c_i	ms_i's MS category
gbr_{p_i}	ms_i's guaranteed bit rate
M	Number of of total orthogonal codes available to the BS
Λ	The set of channel quality indicators (cqi). $\Lambda = \{0, 1, \cdots, 30\}$
$codes(x)$	The number of required parallel codes for cqi x
$dr(x)$	The data rate associated to cqi x
$rep\_cqi_i(t)$	cqi reported by ms_i at time t
$r_i(t)$	$dr\ (rep\_cqi_i(t))$
$alloc\_cqi_i(t)$	cqi allocated to ms_i at time t
$\lambda_i(t)$	$dr(alloc\_cqi_i(t))$
$min\_iti(c)$	minimum inter tti interval for MS category c

Table 1 Summary of Variables

We assume that a user is on average under good enough channel conditions to be able to receive the *gbr* associated to its class since nothing can be done to satisfy users under bad channel conditions. Under this assumption, the goal of our policies is to define fair rules for governing resource allocation under all circumstances. First, when there are sufficient resources we guarantee each user its gbr and fairly distribute the surplus capacity . A user is considered unsatisfied when it is not receiving is requested gbr. When resource is scarce the scheduler tries to satisfy unsatisfied users from the higher classes before those from the lower classes. Although it is the admission control process' job to make sure that all admitted users can be satisfied, user mobility may cause deteriorated channel conditions and instantaneous overloads which in turn will result in unsatisfactory services. At this point, it behooves upon the scheduler to manage the resources in an efficient and prudent manner.

First, we state the resource constraints.

(RC1) $alloc\_cqi_i(t) \leq rep\_cqi_i(t)$, for all i, i.e. assigned cqi cannot exceed reported cqi.

(RC2) $\sum_{i=1}^{n} codes(alloc\_cqi_i(t)) \leq M$, i.e. total number of codes used does not exceed the maximum number allowed.

(RC3) If $alloc\_cqi_i(t) > 0$, then $\sum_{k=t-min\_iti(c_i)+1}^{t-1} \gamma_i(k) = 0$. This ensures that a user is not scheduled until its min_iti timeslots have passed.

The policy constraints are as follows:

(P1) A silver user can be scheduled only if all gold users have been satisfied.

(P2) If there are multiple unsatisfied gold users with throughput less than gbr_g, the gold user with the highest data rate $r_i(t)$ is scheduled.
Here the rationale is to bring as many users above their gbr threshold and thereby satisfy them as quickly as possible so that the other unsatisfied users can also be rescued.

(P3) A silver user i with throughput $x_i(t)$ less than gbr_s has higher priority than any satisfied gold or silver user.

(P4) If there are multiple unsatisfied silver users with throughput less than gbr_s while all gold users have been satisfied, a silver user with the highest value of $r_i(t)$ is scheduled.

(P5) When all users have met their gbr, surplus capacity must be proportionally distributed among the gold and silver users according to their gbrs when data rates are equal.

5.1 Marginal Utility Function

Let N be the set of users in the system who can receive at time-slot t. We will define marginal utility functions $M_{p_i}(x_i(t))$ for each class $p_i \in \{Gold, Silver\}$, whose purpose is to assign a utility value, following the policy rules described earlier, to each user $i \in N$ at time-slot t according to its class p_i, current data rate $r_i(t)$ and current throughput $x_i(t)$. The utility values will be used to determine which user(s) will be scheduled at t. Let the marginal utility functions be defined as,

$$M_{p_i}(x_i(t)) = P_{p_i}(x_i(t)) \cdot r_i(t) \tag{4}$$

where P_{p_i} denotes the preliminary utility function for class p_i, which we will define shortly. Then for any gold user $\hat{g}$ and silver user $\hat{s}$, P_{Gold} and P_{Silver} has to follow the following conditions:

(C1) For $0 \leq x_{\hat{g}}(t) < gbr_g$ and $0 \leq x_{\hat{s}}(t) \leq r_{max}$,

$$P_{Gold}(x_{\hat{g}}(t))r_{min} > P_{Silver}(x_{\hat{s}}(t))r_{max}.$$

In other words, the marginal utility value of an unsatisfied gold user must be greater than any silver user regardless of their data rates.

(C2) For $0 \leq x_{\hat{s}}(t) < gbr_s$ and $gbr_g \leq x_{\hat{g}}(t) \leq r_{max}$,

$$P_{Silver}(x_{\hat{s}}(t))r_{min} > P_{Gold}(x_{\hat{g}}(t))r_{max}.$$

This condition states that the marginal utility value of an unsatisfied silver user must be greater than any satisfied gold user regardless of their data rates.

(C3) For $gbr_s \leq x_{\hat{s}}(t) \leq r_{max}$ and $gbr_g \leq x_{\hat{g}}(t) \leq r_{max}$,

$$P_{Silver}(x_{\hat{s}}(t)) = P_{Gold}\Big(\frac{gbr_g}{gbr_s} \cdot x_{\hat{s}}(t)\Big).$$

This condition ensures that marginal utility values among satisfied users increase proportionally according to their gbrs when data rates are equal.

(C4) For any two silver users, $\hat{s}$ and s',

$$P_{Silver}(x_{s'}(t))r_{min} > P_{Silver}(x_{\hat{s}}(t))r_{max},$$

for $0 \leq x_{s'}(t) < gbr_s$ and $gbr_s \leq x_{\hat{s}}(t) \leq r_{max}$. This ensures that an unsatisfied silver user receives a higher marginal utility value than a satisfied silver user regardless of their data rates.

(C5) For any two gold users, $\hat{g}$ and g',

$$P_{Gold}(x_{g'}(t))r_{min} > P_{Gold}(x_{\hat{g}}(t))r_{max},$$

for $0 \leq x_{g'}(t) < gbr_g$ and $gbr_g \leq x_{\hat{g}}(t) \leq r_{max}$. In other words, an unsatisfied gold user must have a higher marginal utility value than a satisfied gold user regardless of their data rates.

(C6) For any two unsatisfied users, $\widehat{uu}$ and uu' from the same class c, $P_c(x_{uu'}(t)) = P_c(x_{\widehat{uu}}(t))$ regardless of current throughput. In other words, among unsatisfied users from the class the ones with higher instantaneous data rates should have higher marginal utility.

Note that Condition (C1) and (C5) corresponds to policy (P1), (C2) and (C4) corresponds to (P3), and Condition (C3) corresponds to (P5). Policies (P2) and (P4) are addressed by (C6). We define functions P_{Gold} and P_{Silver} satisfying the conditions above as follows. Let $\beta = r_{max} - gbr_g$, $\alpha = \frac{r_{max}}{r_{min}}$ and x_i be the throughput of the user.

$$P_{Silver}(x_i) = \begin{cases} 0 & \text{if } p_i \neq Silver \\ \alpha\beta + 1 & \text{if } 0 \leq x_i < gbr_s \\ r_{max} - \frac{gbr_g}{gbr_s} x_i & \text{if } gbr_s \leq x_i \leq r_{max} \end{cases}$$

$$P_{Gold}(x_i) = \begin{cases} 0 & \text{if } p_i \neq Gold \\ \alpha^2\beta + \alpha + 1 & \text{if } 0 \leq x_i < gbr_g \\ r_{max} - x_i & \text{if } gbr_g \leq x_i \leq r_{max} \end{cases}$$

Functions P_{Gold} and P_{Silver} are shown in Fig. 1. Notice that maximum P_{Silver} value for a silver user $\hat{s}$ with throughput $x_{\hat{s}}(t) \geq gbr_s$ and maximum P_{Gold} value a gold user $\hat{g}$ with throughput $x_{\hat{g}}(t) \geq gbr_g$, is β. So the maximum marginal utility (M_c) for these users is βr_{max}. The minimum marginal utility for a silver user $\hat{s}$ with $x_{\hat{s}}(t) < gbr_s$ is $(\alpha\beta + 1)r_{min} > \beta r_{max}$. This satisfies conditions C2 and C4, and thus policy P3. Similarly, if $x_{\hat{s}}(t) < gbr_s$, $M_{Silver}(x_{\hat{s}}(t)) \leq (\alpha\beta + 1)r_{max}$. For a gold user with $x_{\hat{g}}(t) < gbr_g$, $M_{Gold}(x_{\hat{g}}(t)) \geq (\alpha^2\beta + \alpha + 1)r_{min} > (\alpha\beta + 1)r_{max}$. This satisfies conditions C1 and C5, and thus policy P1. It can also be noticed from the figure that when all users meet their gbr, condition C3 is also satisfied. Also since two gold users with throughtput less than gbr_g has the same P value, the one with the higher data rate will have a higher marginal utility, which conforms to P2. For similar reasons P4 is also satisfied.

Suppose there are more than two classes, i.e. $|\Pi| > 2$. Let gbr_{max} denote the gbr associated to the highest class or SPI. Then P_{p_i} can be generalized for any SPI value $p \in \Pi$ as

$$P_{p_i}(x_i) = \begin{cases} \alpha^{p_i}\beta + \alpha^{p_i - 1} + \cdots + 1, & \text{if } 0 \leq x_i < gbr_{p_i} \\ r_{max} - \frac{gbr_{max}}{gbr_{p_i}} x, & \text{if } gbr_{p_i} \leq x_i \leq r_{max} \end{cases}$$

5.2 Strict Policy Scheduling Algorithm

Let Θ be the list of users that are eligible to receive in time-slot t. A user is eligible if there are packets to be sent to that user and minimum inter time-slot interval (min_iti) for that user has elapsed since its last reception. Let ϕ be the number of codes left to be assigned. Initially it is set to M. During each time-slot t the following steps are used to produce the list of users that are scheduled to receive data.

Algorithm 2 SRA algorithm SPS

```
for all $i$ such that $i \in \Theta$ do
    Compute $M_{p_i}(x_i(t))$ using equation 4
end for
repeat
    Let $i$ be the user such that $i = \arg\max_{j \in \Theta}\{M_{p_j}(x_j(t))\}$
    if $(codes(rep\_cqi_i(t)) < \phi)$ then
        Send to user $i$ using data rate $r_i(t)$
    else
        Send to user $i$ at the maximum rate possible using $\phi$
        number of codes.
    end if
    Update $\phi$
    Remove $i$ from $\Theta$
until $((\phi == 0)$ or $(\Theta$ is empty$))$
```

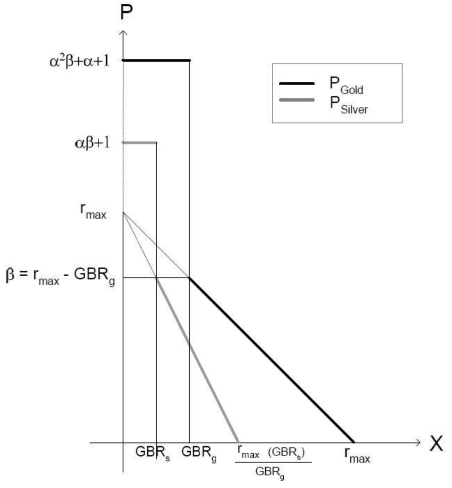


Fig. 1 P_{Gold} and P_{Silver}

5.3 Performance Results

We developed an HSDPA model by extending the UMTS model in OPNET simulator. Our system extensively simulates all the key entities of the network such as MS, BS, RNC, and SGSN with most of the functionalities at all the protocol layers.

The BS maintains a separate transmission queue for each mobile. An MS measures the SINR for each received packet and reports the corresponding cqi back to the BS. For this, actual value interface (AVI) tables [26] are used. For each MS category, we use a separate AVI table that maps a cqi (or data rate) to a corresponding threshold SINR value. An MS reports the maximum possible cqi according to the received SINR for every received packet. Error decision for a received packet is also made based on the threshold SINR for the cqi associated to the transmitted packet. At each time-slot the MAC-hs scheduler schedules a set of users based on the algorithm chosen. BS can fragment or concatenate packets based on the transport block sizes.

5.4 Simulation Setup

In order to evaluate our proposed schedulers we simulated several scenarios. We have a total of 20 HSDPA receivers (10 gold and 10 silver) in the network. The gbr for each class, the traffic load, and capacity of the network (parallel codes) are varied based on the scenarios. The main parameter settings of our simulations are shown in Table 2.

Parameter	Setting
Total Tx power	8 W
Number of parallel codes	5,10
cqi reporting interval	On every packet reception
HSDPA terminal category	1, 7
User receiver type	1-Rx Rake
Path loss model	Vehicular Outdoor
Shadow fading std.	10 DB
Site-to-site distance	2km
Window length for thpt measurement	1000 time-slot

Table 2 Summary of Simulation Parameters

5.5 Evaluating Strict Policy Scheduling Algorithm

5.5.1 Scenario 1

In this scenario, gbr_g and gbr_s are set to 250 kbps and 100 kbps respectively. All the users are equidistant from the BS and experiencing similar channel conditions. The offered traffic load destined to the Gold and Silver users are 300 and 150 kbps. 5 codes were allocated for the parallel data channels. A new gold user is added to the network after 50 seconds. For MS category 1 the downlink capacity using 5 codes is around 3.6 Mbps under good channel conditions. From Fig. 2(a) we can see that before adding the new user, all users are receiving at their gbr and the total downlink throughtput in the cell reaches cell capacity. After the new gold user joined, its throughput quickly went upto 250 kbps; however each silver user's throughput went down by about 25 kbps to make room for the new gold user. This indicates the proper enforcement of policy P1. Also notice, despite having a higher offered load than their gbr, the gold users do not receive at a higher rate than their gbr. Any remaining resource after satisfying all the gold users is being dedicated to the silver users. This complies with policy P3.

However Fig. 2(b) illustrates the disability of PF Scheduler in maintaining any QoS requirement. In this case, both gold and silver users receive similar throughput (150 kbps) as they are experiencing similar radio conditions (SINR) and the offered traffic is high enough to maintain that throughput. SPS also performed better in terms of system throughput producing a combined throughput of 3.5 Mbps compared to PF's 3.15 Mbps. The higher throughput is a result of giving higher priority to the gold users.

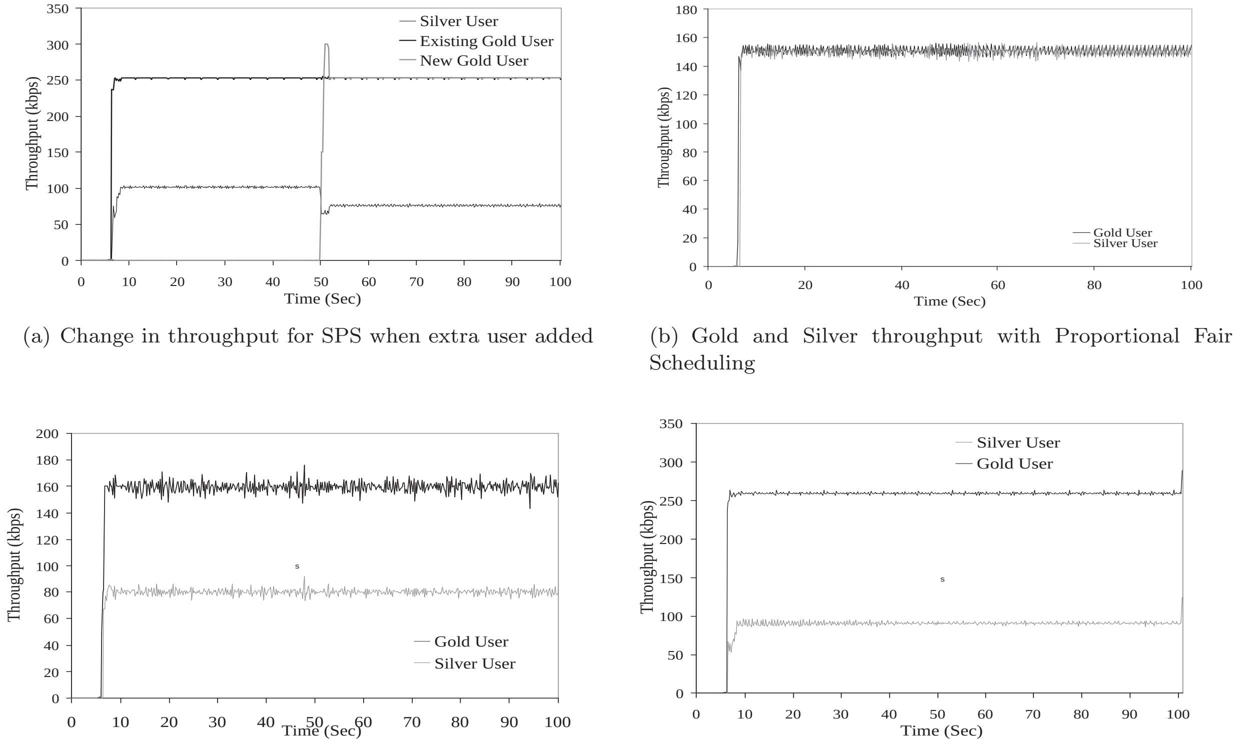
5.5.2 Scenario 2

For this case, gbr_g and gbr_s were set to 128 and 64 kbps respectively. The traffic load destined to the Gold and Silver users were 160 and 80 kbps. 10 codes were allocated for the parallel data channels. Notice that in Fig. 2(c) both type of users are receiving at a higher rate than their gbr at all times. The additional throughput is being distributed proportionally after satisfying all users' guaranteed bit rates. This complies with policy P5.

5.5.3 Scenario 3

In this scenario, we increase gbrs for gold and silver users to 256 kbps and 128 kbps respectively and set the number of codes allocated to 5 to simulate scarcity of resources. The offered traffic load destined to the Gold and Silver users were 300 and 150 kbps, respectively. In Figure 2(d) notice that the gold users are receiving at their required gbr at all times. However due to capacity constraints, the silver users are receiving at a lower rate (93 kbps) than their gbr since the total combined throughput reaches the cell capacity. This complies with policy P1. Also, due to the presence of unsatisfied silver users the satisfied gold users are not exceeding their required gbr. This complies with policy P3.

We varied the channel conditions of the silver users by changing their distances from the BS. The users closer to the BS experienced a higher throughput compared to the more distant ones since their instantaneous data rates were higher. This validates compliance to



(a) Change in throughput for SPS when extra user added

(b) Gold and Silver throughput with Proportional Fair Scheduling

(c) Throughput for SPS when offered traffic is more than requested gbr

(d) Throughput for SPS when total requested gbr exceeds capacity

Fig. 2 Throughput analysis

policy P4. On the other hand, for PF all the users, regardless of priority class, experienced a throughput of around 158 Mbps, and thus dissatisfied the gold users.

6 Loose Channel Allocation Policy

Next,we try to design a simpler and more relaxed scheduling policy that still finds a balance between the two goals of maximizing throughput (using opportunistic scheduling) and satisfying QoS constraints (gbr). Due to its relaxed nature of policy enforecement, we call it loose policy. Notice that this is different from the goal of PF algorithm, which disregards QoS constraint. We define the marginal utility function to be

$$M_{p_i}(x_i) = \frac{\lambda_i}{codes(alloc\_cqi_i(t))} \cdot \frac{gbr_{p_i}}{x_i(t)} \tag{5}$$

Notice that the first ratio contributes to the users with better instantaneous data rate per code and the second ratio emphasizes users who are furthest away from meeting their guaranteed bit rate. Next we discuss the algorithm in further detail.

In algorithm 6, the first for loop determines data rate required for each user to be able to maintain their

Algorithm 3 SRA algorithm LPS

```
Let $\Theta$ denote the set of eligible users in time-slot $t$
for all $i \in \Theta$ do
    Compute the required data rate to maintain gbr as
    $\Delta_i(t) = \tau(gbr_i - x_i(t-1)) + x_i(t-1)$, where $\tau$ is the size
    of the time window for throughput measurement.
    if $(dr(rep\_cqi_i(t)) \leq \Delta_i(t))$ then
        $alloc\_cqi_i(t) = rep\_cqi_i(t)$
    else
        Set $alloc\_cqi_i(t)$ to the smallest cqi such that
            (i) $alloc\_cqi_i(t) \leq rep\_cqi_i(t)$
            (ii) $dr(alloc\_cqi_i(t)) \geq \Delta_i(t)$.
    end if
    Compute $M_{p_i}(x_i(t))$ using equation 5
end for
repeat
    Let $k$ be the user such that $k = \arg\max_{j \in \Theta}\{M_{p_j}(x_j(t))\}$
    if $(codes(alloc\_cqi_k(t)) < \phi)$ then
        Schedule user $k$ to receive using data rate $\lambda_i(t)$
    else
        Schedule user $k$ to receive at the maximum rate possible
        using $\phi$ number of codes.
    end if
    Update $\phi$
    Remove $k$ from $\Theta$
until $\phi = 0$ or $\Theta$ is empty
```

guaranteed bit rate, if at all possible. The main idea is to distribute the bandwidth among as many users (fairness) but at the same time maintain QoS for as many users as possible. Once the rates are assigned, the repeat loop just goes through the users in descending order of their marginal utility and schedules them in a greedy knapsack-like manner.

6.1 Evaluating Loose Policy Scheduling Algorithm

We compared our LPS algorithm with the Max C/I and PF scheduling algorithms under the following scenario: There are ten gold and ten silver users all with MS category 7. On average the gold users are experiencing a worse (by around 5 db) radio condition than the silver users. Ten codes are allocated at BS for downlink transmission. Each gold user has a guaranteed bit rate of 384 kbps, whereas for the silver users it is 128 kbps. Traffic is offered to each gold and silver user at the same rate as their gbr. Figure 3(a) show the average throughputs of a typical gold and silver user under LPS, PF, and MaxC/I. Notice in Figure 3(a) the PF scheduler provides the silver users with a higher throughput (around 128 kbps) than the gold users (75 kbps), since they have better channel condition. At every time-slot, Max C/I always chooses the MS with the best channel condition. Consequently, the silver users receive at 128 kbps, whereas the gold users are totally starved. On the other hand, LPS meets both gold and silver users' throughput requirements. Moreover, since PF and Max C/I choose one user for each time-slot, the capacity cannot be fully utilized if the scheduled user's queue does not have enough data to match the chosen cqi or if the chosen cqi is low. Since the silver users that have lower offered traffic are being chosen more frequently (always by Max C/I), the cell capacity is being under utilized. On the other hand, as LPS chooses the minimum cqi that supports the gbr, it can send to multiple users at the same time, when M is large enough. This, together with the high gbr of the gold users, increases capacity utilization. Notice in Fig. 3(b) that LPS has an average downlink throughput of 5 Mbps compared to Max C/I's 1.28 Mbps and PF scheduler's 2 Mbps.

In a similar scenario, we moved one of the gold users further away from the BS than the rest of the users to worsen its radio conditions (20 db). Consequently, LCS reduced its average throughput to around 200 kbps, while still maintaining the gbr for all the other users. This indicates that LCS can also relax the QoS constraints in order to increase throughput.

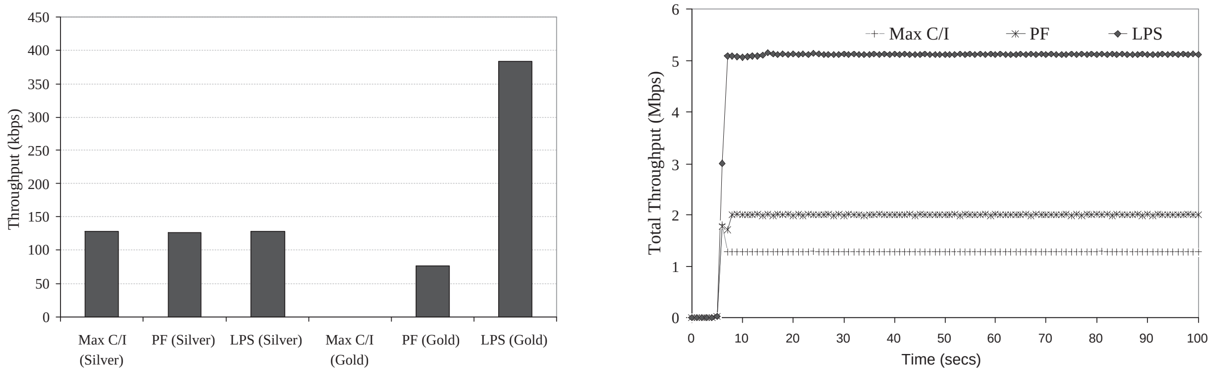
7 Conclusion

We have studied the multi-user scheduling and rate assignment problem for high speed CDMA based cellular networks. We start by addressing down-link SRA for throughput maximization under a multi-user multi-rate environment. We showed that a solution exists in the sense of Filippov's. We followed this with the continuous utility maximization problem for code-multiplexed downlink SRA. We proposed a greedy knapsack-like algorithm and proved convergence of utility under this algorithm to the maximum total utility over a fixed resource domain.

We also introduced the notion of resource and QoS aware SRA policies. Policies are designed to prioritize QoS classes during times of overload and to fairly distribute surplus capacity when load is manageable. We developed a marginal utility based algorithm, SPS, in accordance with our strict policies. This approach attempts to guarantee each user their requested quality of service keeping in view the current state of each channel as well as the priority classifications. We also introduced a more relaxed policy based algorithm, LPS, that tries to find a balance between throughput maximization and fairness. We experimentally show that our SPS and LPS algorithms very effectively satisfy high priority users by guaranteeing their QoS constraints if it is at all possible, and perform better than other well-known algorithms such as Max C/I and PF, especially under stressful conditions.

References

1. D. Zhao, X. Shen, and J. Mark, "Radio resource management for cellular cdma systems supporting heterogeneous services," *Mobile Computing, IEEE Transactions on*, vol. 2, no. 2, pp. 147–160, April-June 2003.
2. M. Shariat, A. Quddus, S. Ghorashi, and R. Tafazolli, "Scheduling as an important cross-layer operation for emerging broadband wireless systems," *Communications Surveys Tutorials, IEEE*, vol. 11, no. 2, pp. 74 –86, quarter 2009.
3. H. Jiang, W. Zhuang, and X. Shen, "Cross-layer design for resource allocation in 3g wireless networks and beyond," *Communications Magazine, IEEE*, vol. 43, no. 12, pp. 120–126, Dec. 2005.
4. L. Xu, X. Shen, and J. Mark, "Dynamic fair scheduling with qos constraints in multimedia wideband cdma cellular networks," *Wireless Communications, IEEE Transactions on*, vol. 3, no. 1, pp. 60–73, Jan. 2004.
5. M. Conti and E. Gregori, "Traffic and interference adaptive scheduling for internet traffic in umts," *Mob. Netw. Appl.*, vol. 9, no. 4, pp. 265–277, 2004.
6. O. Yu, E. Saric, and A. Li, "Adaptive rate-scheduling with reactive delay control for next generation cdma wireless mobile systems," *EURASIP J. Wirel. Commun. Netw.*, vol. 2006, no. 2, pp. 54–54, 2006.



(a) Comparing single user throughput

(b) Comparing network throughput

Fig. 3 Comparing throughput among LPS, PF and Max C/I

7. X. Liu, E. Chong, and N. Shroff, "A framework for opportunistic scheduling in wireless networks," *Computer Networks Journal*, vol. 41, no. 4, pp. 451–474, 2003.
8. A. Farrokh and V. Krishnamurthy, "Opportunistic scheduling for streaming multimedia users in high-speed downlink packet access (hsdpa)," *Multimedia, IEEE Transactions on*, vol. 8, no. 4, pp. 844 –855, aug. 2006.
9. O. S. Shin and K. B. Lee, "Antenna-assisted round robin scheduling for mimo cellular systems," *IEEE Communication Letters*, pp. 109–111, March 2003.
10. R. W. Heath and A. J. Paulraj, "Multiuser diversity for mimo wireless systems with linear receivers," in *Asilomal Conference on Signals, Systems & Computers 2003*, Nov. 2003, pp. 982–986.
11. D. Niyato, E. Hossain, and D. Kim, "Joint admission control and antenna assignment for multiclass qos in spatial multiplexing mimo wireless networks," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 9, pp. 4855 –4865, september 2009.
12. T. Kolding, F. Frederiksen, and P. Mogensen, "Performance aspects of wcdma systems with high speed downlink packet access (hsdpa)," in *Vehicular Technology Conference*, 2002, pp. 477– 481.
13. M. Dianati, X. Shen, and K. Naik, "Cooperative fair scheduling for the downlink of cdma cellular networks," *Vehicular Technology, IEEE Transactions on*, vol. 56, no. 4, pp. 1749 –1760, july 2007.
14. A. Jalali, R. Padovani, and R. Pankaj, "Data throughput of cdma-hdr a high efficiency-high data rate personal communication wireless system," in *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, vol. 3, pp. 1854–1858.
15. T. Kolding, "Link and system performance aspects of proportional fair scheduling in wcdma/hsdpa," in *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, vol. 3, pp. 1717– 1722.
16. G. Caire, R. Muller, and R. Knopp, "Hard fairness versus proportional fairness in wireless communications: The single-cell case," *Information Theory, IEEE Transactions on*, vol. 53, no. 4, pp. 1366 –1385, april 2007.
17. O. Sallent, J. Perez-Romero, R. Agusti, and F. Casadevall, "Provisioning multimedia wireless networks for better qos: Rrm strategies for 3g w-cdma," *Communications Magazine, IEEE*, vol. 41, no. 2, pp. 100–106, Feb 2003.
18. H. Jiang, W. Zhuang, X. Shen, and Q. Bi, "Quality-of-service provisioning and efficient resource utilization in cdma cellular communications," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 1, pp. 4 – 15, jan. 2006.
19. V. Huang and W. Zhuang, "Qos based fair resource allocation in multi-cell td/cdma communication systems," *Wireless Communications, IEEE Transactions on*, vol. 5, no. 2, pp. 339 – 346, feb. 2006.
20. A. Stamoulis, N. Sidiropoulos, and G. Giannakis, "Time-varying fair queueing scheduling for multicode cdma based on dynamic programming," *Wireless Communications, IEEE Transactions on*, vol. 3, no. 2, pp. 512–523, March 2004.
21. V. Huang and W. Zhuang, "Qos-oriented packet scheduling for wireless multimedia cdma communications," *Mobile Computing, IEEE Transactions on*, vol. 3, no. 1, pp. 73–85, Jan-Feb 2004.
22. M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Communications Magazine*, vol. 39, no. 2, pp. 150–154, Feb. 2001.
23. P. Hosein, "Qos control for wcdma high speed packet data," in *4th International Workshop on Mobile and Wireless Communications Network*, 2002, pp. 169 – 173.
24. ——, "Scheduling of voip traffic over a time-shared wireless packet data channel," in *IEEE International Conference on Personal Wireless Communications, 2005. ICPWC 2005.*, Jan. 2005, pp. 38 – 41.
25. H. Holma and A. Toskala, Eds., *HSDPA/HSUPA for UMTS.* Wiley, 2006.
26. S. Hmlinen, P. Slanina, M. Hartman, A. Lappetelinen, H. Holma, and O. Salonaho, "A novel interface between link and system level simulations," in *ACTS Mobile Telecommun*, Oct. 1997, pp. 599–604.
27. A. F. Filippov, "Differential equations with discontinuous right-hand side," *Mat. Sb.*, vol. 51 (93), 1960.
28. ——, "Differential equations with discontinuous right-hand sides," *Trans. Amer. Math. Soc.*, vol. 42, 1964.
29. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms.* McGraw Hill, 2001.

Appendix A - Filippov solution

Consider a system of ordinary differential equations

$$\frac{d\mathbf{x}}{dt}(t) = f(t, \mathbf{x}(t)). \tag{6}$$

To deal with the case where $f(t, \mathbf{x})$ is discontinuous in $\mathbf{x} = (x_1, \ldots, x_n)$, we consider the differential inclusions

$$\frac{d\mathbf{x}}{dt}(t) \in F(t, \mathbf{x}(t)),$$

where F is a set-valued map which associates with a point $(t, \mathbf{x}) \in \mathbb{R} \times \mathbb{R}^n$ a set $F(t, \mathbf{x}) \in \mathbb{R}^n$. One way to transform a given ordinary differential equation into a differential inclusion is proposed by Filippov [27, 28].

Definition 2 ([Filippov]) A vector function $\mathbf{x}(t)$ is called a solution of Equation (6) on $[t_0, t_1]$ if $\mathbf{x}(t)$ is absolutely continuous on $[t_0, t_1]$ and for almost all $t \in [t_0, t_1]$

$$\frac{d\mathbf{x}}{dt}(t) \in K[f](t, \mathbf{x}(t))$$

where

$$K[f](t, \mathbf{x}(t)) = \bigcap_{\delta > 0} \bigcap_{\mu(N)=0} \overline{\mathrm{conv}} f(t, B(\mathbf{x}(t), \delta) - N),$$

that is, the smallest closed convex set containing all limit values of f at $\mathbf{x} = \mathbf{x}(t)$.

We can obtain a rough information on the trajectories of solutions to (2) as the next proposition shows, understanding the solutions in the sense of Filippov's. Let $C(\mathbf{x}, D_\lambda) := \{(1-t)\mathbf{x} + t\mathbf{y} \,|\, 0 \leq t \leq 1,\ \mathbf{y} \in D_\lambda\}$. For a convex set D_λ, it is easy to verify that the set $C(\mathbf{x}, D_\lambda)$ is also convex. Clearly, $D_\lambda \subset C(\mathbf{x}, D_\lambda)$ for any $\mathbf{x}$, and $C(\mathbf{x}, D_\lambda) = D_\lambda$ if $\mathbf{x} \in D_\lambda$.

Proposition 1 *Let D_λ be a compact convex set. Consider the following initial value problem:*

$$\frac{d\mathbf{x}}{dt}(t) = \boldsymbol{\lambda}(t, \mathbf{x}(t)) - \mathbf{x}(t), \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{7}$$

where $\boldsymbol{\lambda}(t, \mathbf{x}) \in D_\lambda$ for all $t \in [0, \infty)$. Then we have the following:

1. *there exists a solution $\mathbf{x}(t)$ to the equation (7) in the sense of Filippov;*
2. *$\mathbf{x}(t) \in C(\mathbf{x}_0, \Omega)$ for all $t \in [0, \infty)$;*
3. *$C(\mathbf{x}(t), \Omega) \to \Omega$ as $t \to \infty$.*

In view of the second and the third results of Proposition 1, we find that once the trajectory of a solution $\mathbf{x}(t)$ to (7) enters the resource domain D_λ, the remaining trajectory will be confined to D_λ.

Appendix B - Total Utility Maximization Problem (TUMP)

The goal of TUMP is to find a throughput configuration $\mathbf{x} = (x_1, \ldots, x_n)$ such that the total utility $U(x_1, \ldots, x_n)$ as given by equation 3 is maximized subject to the constraint:

$$(x_1, \ldots, x_n) \in D_x. \tag{8}$$

In a realistic scenario, even if we know in advance where U has a maximum over a throughput domain D_x, we cannot make the throughput configuration $\mathbf{x}$ jump from an initial position $\mathbf{x}_0$ right into some optimal point $\mathbf{x}^* \in D_x$ in a few steps (time slots), due to the limited amount of packets available to each user. Therefore, we need to devise an overall control strategy to guide the trajectory of $\mathbf{x}$ step by step to some optimal point $\mathbf{x}^*$ of U over the throughput domain D_x. To this end, the current throughput and marginal utility information are required when decisions are made dynamically. To address this situation, we formulate an open loop feedback control problem corresponding to the TUMP as follows:

Appendix B.1 Total Utility Maximization via Control(TUMC)

For a resource domain $D_\lambda(t)$, find an SRA algorithm $\boldsymbol{\lambda}(\mathbf{x}; \mathbf{u})$ under the constraints

$$\begin{aligned} &(1)\ \boldsymbol{\lambda}(\mathbf{x}(t); \mathbf{u}(\mathbf{x}(t))) \in D_\lambda(t), \\ &(2)\ \frac{d\mathbf{x}}{dt}(t) = \boldsymbol{\lambda}(\mathbf{x}(t); \mathbf{u}(\mathbf{x}(t))) - \mathbf{x}(t), \\ &(3)\ \mathbf{x}(0) = \mathbf{x}_0 \end{aligned} \tag{9}$$

such that the total utility $U(\mathbf{x}(t))$ converges to the maximum of U over a domain D_x.

At this point, we remark on the relationship between the throughput domain D_x and the resource domain D_λ. If the throughput domain D_x is large in comparison with the resource domain D_λ, the total utility $U(\mathbf{x}(t))$ may never be able to reach near the unique maximum of U over D_x under any control $\boldsymbol{\lambda}$ due to possible degradation of wireless channel conditions. In view of the results of Proposition 1, it is not so far-fetched to treat the resource domain D_λ and the throughput domain D_x as the same domain, simply denoting D.

Now consider a time-varying resource domain

$$\begin{aligned} D_\lambda(t) = \{(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^n \,|\, \lambda_1 + \cdots + \lambda_n \leq M_{\mathrm{sys}}, \\ 0 \leq \lambda_i \leq r_i(t) \leq r_i^{\max},\ 1 \leq i \leq n\}, \end{aligned}$$

where for each i, $r_i^{\max}$ is a pre-specified upper bound for $r_i(t)$. The constant $M_{\mathrm{sys}} = r_{max}$ is the total capacity in terms of data rate for the scheduler. Typically $r_i(t)$ is treated as a random process, since it should reflect the time-varying and location-dependent characteristics of the wireless channel.

For simplicity of the proof, we assume from now on that each r_i $(i = 1, \ldots, n)$ is *fixed*, i.e. $r_i(t) = r_i$ for all t. Thus D_λ is fixed with respect to time variable t. However, to avoid an accusation of oversimplification for the model, we instead consider two scenarios: (i) a scenario in which the current channel conditions are so poor that there is no way to avoid a decrease in total utility as a result of throughput degradations, and (ii) a scenario in which the current channel conditions are fairly good enough so that there is room for improving the current total utility for users. In terms of ordinary differential equation, the first scenario translates into an initial value problem with its initial throughput $\mathbf{x}_0$ *outside* the resource domain D_λ, while the second scenario translates into the one with its initial throughput $\mathbf{x}_0$ *inside* the resource domain D_λ. We will prove the convergence of our guided trajectories to optimal points of U over D without imposing any condition on the position of initial throughput $\mathbf{x}_0$ in (9).

The next proposition, obtained by applying the Karash-Kuhn-Tucker(KKT) conditions to TUMC, provides a useful information on the optimal points of U over a domain $D = \{(x_1, \ldots, x_n) \in \mathbb{R}^n \,|\, x_1 + \cdots + x_n \leq M_{\mathrm{sys}},\ 0 \leq x_i \leq r_i,\ i = 1, \ldots, n\}$.

Proposition 2 *The total utility U has a maximum on D. Furthermore, if U has a maximum at $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_n^*) \in D$, there exist $\mu_1^*, \ldots, \mu_n^* \geq 0$ and $\eta_1^*, \ldots, \eta_n^* \geq 0$ such that*

$$\begin{aligned} (i)\quad & u_1(x_1^*) + \mu_1^* - \eta_1^* = u_2(x_2^*) + \mu_2^* - \eta_2^* = \cdots \\ & = u_n(x_n^*) + \mu_n^* - \eta_n^* \neq 0, \\ (ii)\quad & \mu_i^* x_i^* = 0, \\ (ii)\quad & \eta_i^*(x_i^* - r_i) = 0, for \quad i = 1, \ldots, n. \end{aligned}$$

Continuous Scheduling and Rate Assignment Algorithm:

There may possibly be numerous SRA algorithms which solve the TUMC. As our SRA algorithm we will choose the one which maximizes the instantaneous increase $dU(\mathbf{x}(t))/dt$ of the total utility U at each time unit t. As we shall see, this"greedy and near-sighted" choice may not be optimal but solves the TUMC successfully when it is assumed that $D_x = D_\lambda$. Considering uncertain or dynamically changing wireless channel conditions, we find it reasonable to expect that this greedy choice would be more appropriate than any other choice.

Suppose that U is differentiable with respect to $\mathbf{x}$. Applying the chain rule to $U \circ \mathbf{x}$, we obtain the instantaneous increase of the total utility as follows:

$$\frac{dU(\mathbf{x}(t))}{dt} = \mathbf{u}(\mathbf{x}(t)) \cdot (\boldsymbol{\lambda}(t) - \mathbf{x}(t))$$

As we mentioned before, our control strategy is to choose $\boldsymbol{\lambda}$ as the one which maximizes the instantaneous increase of the total utility at each time.

Sub-Problem (SP):

Given $\mathbf{x}$ and $\mathbf{u}$, maximize
$\mathbf{u} \cdot (\boldsymbol{\lambda}(\mathbf{x}; \mathbf{u}) - \mathbf{x})$
subject to the constraint

$$\boldsymbol{\lambda}(\mathbf{x}; \mathbf{u}) \in D.$$

It will turn out later that this problem is reduced to a fractional knapsack problem and is solvable in $O(n \log n)$. Before that we give a more intuitive and geometric approach to SP.

To each $\mathbf{x} \in D$ is assigned a unique marginal utility vector $(u_1, u_2, \ldots, u_n)$, which is the gradient $\nabla U(\mathbf{x})$ of the total system utility $U(\mathbf{x})$. We can naturally define a vector field $\mathbf{u}$ on D by $\mathbf{u}(\mathbf{x}) = \nabla U(\mathbf{x})$. It is well-known that the gradient of a function f is locally the direction of the fastest increase in the value of f. Therefore, to get the fastest increase of the total utility U at $\mathbf{x}$, we should go from $\mathbf{x}$ in the direction of $\mathbf{u}(\mathbf{x})$. This discussion motivates us to define a SRA algorithm $\boldsymbol{\lambda}_\infty$ which is given as follows:

$$\boldsymbol{\lambda}_\infty(\mathbf{x}; \mathbf{u}) := \lim_{s \to \infty} \pi_D(\mathbf{x} + s\mathbf{u}),$$

where π_D is the projection map onto the closed convex set D. The next proposition shows that the SRA algorithm $\boldsymbol{\lambda}_\infty$ is a solution of SP.

Proposition 3 *Let D be a compact convex subset of $\mathbb{R}^n$. Then the limit $\boldsymbol{\lambda}_\infty$ exists in D. Furthermore, we have*

$$\sup_{\boldsymbol{\lambda} \in D} (\boldsymbol{\lambda} - \mathbf{x}) \cdot \mathbf{u} = (\boldsymbol{\lambda}_\infty - \mathbf{x}) \cdot \mathbf{u}. \tag{10}$$

As is seen in Proposition 3, the SRA algorithm $\boldsymbol{\lambda}_\infty$ is similar to a numerical optimization method such as the Method of Feasible Directions. A typical Method of Feasible Directions is comprised of two steps: search direction step and line search step. Crucial differences between a typical method of feasible directions and the SRA algorithm $\boldsymbol{\lambda}_\infty$ are

- each step size in $\boldsymbol{\lambda}_\infty$ is already fixed, say $\alpha = \frac{1}{\tau}$, in our case (thus our SRA algorithm consists only of search direction step);
- the initial position $\mathbf{x}_0$ may not be in the feasible space D.

The next theorem states that the resulting total utility $U(\mathbf{x}(t))$ under the SRA algorithm $\boldsymbol{\lambda}_\infty$ actually converges to the maximum of U over D. In other words, the SRA algorithm $\boldsymbol{\lambda}_\infty$ solves the TUMC.

Theorem 2 (Convergence) *Let D be a compact subset of $\mathbb{R}^n$. Let U be a differentiable and concave function. Let M be the maximum value of U over D. Then for any solution $\mathbf{x}(t)$ of the following initial value problem*

$$\frac{d\mathbf{x}}{dt}(t) = \boldsymbol{\lambda}_\infty(\mathbf{x}(t); \mathbf{u}(\mathbf{x}(t)) - \mathbf{x}(t), \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{11}$$

the value $U(\mathbf{x}(t))$ converges to M as $t \to \infty$. Furthermore, the convergence speed is exponential.

As we have mentioned before, we understand the solutions of Equation (11) in the Filippov's sense. Note that Theorem 1 does not assume that the initial position $\mathbf{x}_0$ have to be in the domain D and the convergence speed is exponential, which tells us that the speed is fast enough so that the throughput configuration $\mathbf{x}$ can catch up with the time-varying domain D almost immediately as long as the domain D changes slowly enough with time.

Reduction to Fractional Knapsack Problem:

If one wants to implement $\boldsymbol{\lambda}_\infty$ in a real application, one still needs to find an efficient way to evaluate $\boldsymbol{\lambda}_\infty(\mathbf{x}; \mathbf{u})$ given $\mathbf{x}$ and $\mathbf{u}$. Finding $\boldsymbol{\lambda}_\infty(\mathbf{x}; \mathbf{u})$ on the domain D can be easily reduced to finding the optimal solution of a fractional knapsack problem. A typical form of *Fractional Knapsack Problem (FKP)* is given as follows:

Given weights $(w_1, \ldots, w_n)$ and values $(v_1, \ldots, v_n)$,
find $(z_1, \ldots, z_n)$ that maximizes $v_1 z_1 + \cdots + v_n z_n$
subject to

1. $w_1 z_1 + \cdots + w_n z_n \leq C$, {C is called the capacity of the knapsack }
2. $0 \leq z_i \leq 1, \quad i = 1, \ldots, n$.

It is well-known that a fractional knapsack problem is solvable by a greedy strategy with $O(n \log n)$ [29] as stated in the theorem below.

Theorem 3 *The greedy algorithm that always selects the object with better value to weight ratio always finds an optimal solution to the Fractional Knapsack problem.*

Now let us rewrite our sub-problem of maximizing the instantaneous increase of total utility U:

Sub-Problem (SP):

Given $\mathbf{x}$ and $\mathbf{u}$, at each time t, find $(z_1, \ldots, z_n)$ that maximizes

$$u_1(r_1 z_1 - x_1) + \cdots + u_n(r_n z_n - x_n)$$

subject to

$$r_1 z_1 + \cdots + r_n z_n \leq M_{sys},$$
$$0 \leq z_i \leq 1, \quad i = 1, \ldots, n,$$

It is easy to see that the new SP can be reduced to a fractional knapsack problem; indeed, this can be done by dropping the term $-(u_1 x_1 + \cdots + u_n x_n)$, which is constant from the objective function of the above problem, and setting

$$w_i := r_i, \quad \text{and } v_i := u_i r_i.$$

To summarize, we obtain the following result.

Corollary 1 *The optimal solution of SP can be found by assigning to the users with higher marginal utility the maximum transmission rate they can receive as long as there is leftover bandwidth capacity. The time complexity of the algorithm is $O(n \log n)$.*

Corollary 1 together with Theorem 2 completes the proof of Theorem 1.