APPLICATION-BASED SCHEDULING FOR INDOOR FEMTOCELL NETWORKS

A Thesis Proposal

by

Devin Mungan

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Electrical and Electronics Engineering

Özyeğin University
January 2013

APPLICATION-BASED SCHEDULING FOR INDOOR FEMTOCELL NETWORKS

Approved by:

Professor M. Oğuz Sunay, Advisor
Department of Electrical and Electronics Engineering
Özyeğin University

Professor Ali Özer Ercan
Department of Electrical and Electronics Engineering
Özyeğin University

Professor İsmail Arı
Department of Computer Science
Özyeğin University

Date Approved: 7 January 2013

To My Family...

Sema, Necmi and Direnç

ABSTRACT

This thesis is concerned with designing a QoS-enabled, application-based scheduler for indoor femtocell networks. For this purpose, a novel scheduler is proposed which takes into account the quality of service (QoS) requirements of each active application as well as whether their sustainability is at risk at a given time. The scheduler aims to maintain an efficient use of the wireless radio resources and strives to achieve a user and application specific notion of fairness of service reception. Unlike current solutions, the scheduler differentiates between not only different users but also between applications of each user, and schedules data transmission amongst these applications so that QoS requirements for all applications are satisfied.

ÖZETÇE

Bu tez raporu, bina içi femto hücre ağları için uygulama bazlı ve servis kalitesini gözeten bir paket çizelgeleyici geliştirmek ile ilgilidir. Bu amaçla, bütün uygulamaların o anki zaman diliminde ulaştıkları servis kalitesini hesaba katan bir paket çizelgeleyici önerilmiştir. Önerilen çizelgeleyici, kablosuz sistem özkaynaklarını verimli bir şekilde kullanmak ile birlikte, bu özkaynakları hem uygulama hem de kullanıcı bazında adil bir şekilde paylaştırmayı hedeflemiştir. Hali hazırda bulunan çizelgeleyicilerin aksine, önerilen yöntem sadece kullanıcılar arasında değil, ayrıca kullanıcıların aktif olan bütün uygulamaları arasında ayrıştırma yapıyor ve de uygulamaların tümünün servis kalitesini eniyileyecek şekilde veri gönderimini gerçeklemeyi amaçlıyor.

ACKNOWLEDGEMENTS

First I would like to offer my sincere gratitude to my thesis advisor Assoc. Prof. Oğuz Sunay for his guidance in this study, his patience, support and belief in me since my undergraduate years. I also would like to thank him for his valuable advices that I highly benefit not only for my academic studies but also for my personal life.

I also would like to thank my friends and colleagues Gülden Ferazoğlu, Koray Kökten and Serkan Kirkgül in Wireless Information Systems Engineering Research Laboratory for their company and help.

Finally, would like to express my sincere thanks to my family for their love and support. Without them, I would not be able to become the person I am today.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The importance of wireless communications are increasing exponentially. Among these wireless communication techniques, cellular communication undoubtedly became one of the most popular and widely accepted implementation around the globe. In the early days of cellular communications, voice applications were the dominant traffic on the cellular networks and it was not until the early 2000s that data applications became available for usage on cellular devices. The standardization of UMTS by Third Generation Partnership Project (3GPP) and CDMA2000 by Third Generation Partnership Project 2 (3GPP2) and the following deployments of these standards practically started a new age for cellular communications. Today, thanks to the existing, continuously evolving standards such as HSPA, EV-DO and LTE, mobile operators are now capable of offering very high data rates to the subscribers and, it is possible to demand various types of data applications from the Internet without depending on a wired or WiFi solution which greatly damage mobility of subscribers.

In order to deliver high data rates, users have to experience high quality channels from the corresponding base stations of the mobile operators. In cellular networks, it is known that majority of these data requests are requested when the user is indoors [1]. Then, it is very important for mobile operators to provide good indoor coverage. Traditional approach to provide indoor coverage is to use macrocells. Macrocell base stations (MNB) are configured so that both indoor and outdoor users receive acceptable data rates, but signal strength decreases significantly for the case of indoor usage due to the wall penetration losses and relatively long distances between the transmitter and the receiver. Macrocells can still be used to increase indoor channel quality but in this approach, indoor users need to be allocated more power to balance the signal strength loss caused by above mentioned impairments, which in turn leads to low cell throughput. Deploying more base stations is another option but considering the price of a MNB, this solution surely causes sizable decrease in operator's revenues. To overcome these problems, femtocell networks are proposed to increase indoor capacity and channel qualities [2].

1.1 Femtocell Networks

We stated in previous section that how much it is important to provide high indoor coverage in order to fulfill customer needs. The state of the art approach to achieve this goal is to shrink the network cells so that the channel distance between user equipment and the serving station would be reduced. By decreasing the distance, channel would be more resilient to the impairments imposed by wireless nature of telecommunications.

Femtocells are operational by a home node B (HNB) that is connected to the mobile operators network via a third party backhaul. HNBs can provide a coverage range up to 50 meters with a low output power level typically less than 50 mW [3].

Figure 1.1 illustrates a basic femtocell deployment model in a real world environment. Here, user only needs to connect the plug and play HNB to a broadband access point and through a operator-deployed femtocell gateway, the user equipment will have access to Internet and the mobile operator's core network. Subscribers connected to HNBs therefore benefit from high channel qualities [4] and connect to cellular Internet without the usage of a MNB.

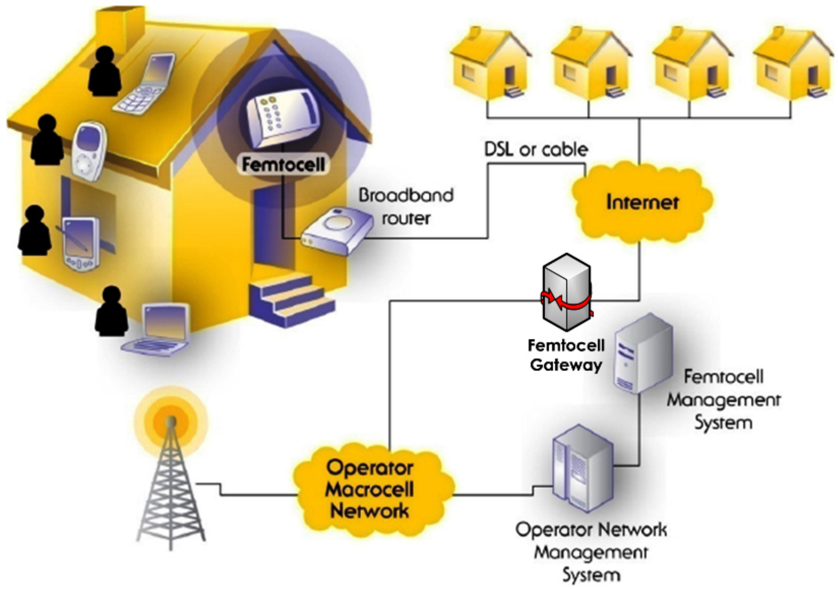

Figure 1: A Femtocell Network

1.2 Resource Allocation Phase

In the early days of telecommunications, circuit-switched networks (CSN) were widely in use where the two communicating nodes establish a dedicated channel (circuit) prior to communication. With this technique, full bandwidth of the channel is allocated to the two communicating nodes where the channel acts as a physical link between them. Obviously no scheduling mechanism is needed for this type of network because a dedicated channel with full capacity is assigned to nodes and the channel is dedicated as long as one of the communicators is down. In packet-switched networks (PSN) however, the scenario is very different. Most important characteristics of a PSN is that the transmitted data is grouped into *packets* that independently travel across a *non-dedicated* network which may be shared between several nodes. This method practically forms today's various communication networks including the Internet and packet-switched networks of cellular communications.

For PSN, the transmission medium is shared between active nodes whether the communication channel is wired or wireless. Therefore, a scheduling rule is needed for which users to receive service in a given time. Consider a centralized, Time division multiplexed (TDM) system where only 1 node is allowed to talk in a given time fragment. Assume further that the link is not capable of transmitting all the data created by the nodes in the network. In this case, a scheduling rule, operating in the switch, router or a base station, needs to be established to rank and selects users for every time fragment.

Before wireless communication technology, packet schedulers were also in use on wired networks, but wired networks have almost constant link capacities therefore channel conditions are not related in the resource allocation process. On the contrary, wireless networks have variable link rates due to the wireless channel impairments which dictates that these impairments must be considered in the scheduling phase. Due to these challenges, a new research area has emerged on wireless packet schedulers and various different proposals have been made on this topic. A brief literature survey will be given at the next chapter.

It should be clearly stressed here that, the scheduler holds great importance for a packet-switched communications network to efficiently operate. Current standards and techniques dictate that users have to aggressively compete for the wireless system resources and since the channel is shared, a well-designed scheduler is needed to rank the users with a specific decision mechanism and allocate the system resources accordingly.

1.3 Concepts

Since the transmission medium is shared between active subscribers in current cellular network standards and beyond, a scheduler operating on MAC layer has to decide for which users to receive transmission at a given transmission time interval (TTI).

Traditional schedulers for cellular communications aim to maximize/equalize each user's throughput by gaining advantage of multiuser diversity. This approach makes sense since the aim is to share the limited wireless resources as fairly as possible between a large number of users while attempting to maximize system throughput. By using multiuser diversity, subscribers are allowed to transmit bursty data, which results in high system throughput. Macrocell users are typically mobile, and such a user has rather low probability of requesting multiple, parallel data applications simultaneously. This has resulted in almost all of the scheduling algorithms that have appeared in the literature to be user-based [5–16]. However, the scenario is dramatically different for indoors. HNBs cover a much smaller area and number of users connected to a typical HNB is much less than the MNB scenario. Femtocell users are nomadic at worst if not stationary and the probability of requesting multiple data applications, each potentially has different QoS requirements, for a femtocell subscriber is much more likely than a mobile subscriber connected to a MNB. Traditional scheduling approaches and QoS-enabled schedulers designed for macrocells will not potentially perform well when the users request multiple data applications. In other words, existing schedulers do not attempt to differentiate between applications. Traffic destined to users is stored in user-based buffers at the transmitter even if they belong to different applications. Even if a QoS-enabled scheduler is set with user-based buffering, one QoS level will be set for each user. A user requesting a best effort service and a streaming service at the same time will be set a QoS level suitable for either streaming or best effort traffic. In both cases, the level set will damage the overall system and the user. If QoS level is set according to the streaming service, best effort application of the user will also be considered as a delay and jitter sensitive service and the corresponding application will potentially dominate the entire system resources resulting unfair resource allocation. If, on the other hand, the QoS level is set according to the best effort service, streaming application will most likely fail at delivering a pause-free, high quality video transmission to the subscriber. Therefore a new approach is essential to satisfy this type of aggressive usage where not only users but also each application of every user has to be considered for the resource allocation phase.

1.3.1 Primary Concept

Our primary concept in this thesis is to show that traditional schedulers used in macrocell cellular networks do not perform well in home based femtocell networks because of the aggressive usage behavior of the subscribers in terms of requesting data applications. An application-based scheduler, rather than a user-based scheduler must be operational in order to satisfy customer needs. By extensive simulations, we show that indeed this is the case where traditional schedulers actually fail providing a certain amount of QoS to the corresponding applications in terms of providing high quality video transmission, pause and delay free video teleconferencing applications and low web page loading times.

1.3.2 Secondary Concept

As a second step for our studies, after showing that traditional schedulers perform very poorly in terms of delivering a certain level of QoS, we intend to come up with a new scheduling rule designed specifically for femtocell networks. we propose a novel, application-based scheduler which differentiates not only between users but every application of each user so that the MAC layer scheduler can deliver sufficient QoS for different types of applications since each application can have its own distinct QoS needs for delivering high quality end usage experience. We also provide a multiple objective optimization (MOO) framework in order to optimize the parameters of our proposed scheduler.

CHAPTER II

3G HSDPA NETWORKS

In this work, we focused on HSDPA enhancement of 3GPP [17], therefore a detailed understanding on the HSDPA standard and scheduling techniques is necessary to design a well-performed scheduling algorithm for HSDPA-based indoor femtocell networks.

HSDPA standard dictates that each user has to measure its time-varying channel quality every 2 ms, which is the duration of a single TTI. Base station continuously transmits a pilot signal to guide the mobiles channel quality measurements. The measured SINR value is mapped to a 5-bit channel quality indicator (CQI) by the user equipment (UE) and is reported back to the serving base station. Upon receiving the measured CQIs, base station determines the amount of data each user is capable of receiving, which is called transmission block size (TBS), without exceeding 10% block error probability (BLER) on the average. CQI-TBS mapping is UE specific. In our case we assume all UEs are Category 14 devices [18], which are compatible with our implementation of HSDPA release 10. We also assume that CQI feedbacks are received error-free at the transmitter side. For Category 14 devices, mapping is given in Table 1. Authors in [19] suggest that it is possible to model SINR-to-CQI mapping as;

$$CQI = \lfloor SINR + 4.5 \rfloor \tag{1}$$

The corresponding BLER at this channel quality can also be approximated as [20];

$$BLER = \{10^{2^{\frac{SINR-1.03CQI+5.26}{\sqrt{3}-\log(CQI)}}}\}^{\frac{-1}{0.7}} \tag{2}$$

after the BLER calculation for a given SINR, achievable data rate in terms of the previously calculated BLER can then be found as;

$$DRC = [TBS * (1 - BLER)] \tag{3}$$

In HSDPA, a total of 16 orthogonal channelization codes are available at the transmitter. Transmission power is equally divided between these codes. 1 code is allocated for the pilot signal transmission and the remaining 15 is used for data transmission. Up to 4 users can receive data in a single TTI since only 4 high speed-shared control channel (HS-SCCH) is available at the transmitter [21]. After the necessary mappings and calculations are completed at the base station side, a decision mechanism divides the available system resources amongst the eligible subscribers.

2.1 Scheduling in HSDPA Networks

Scheduling has a major impact on the performance of cellular data communication networks. Numerous studies have been published on different scheduling algorithms for cellular networks. The main purpose of all these algorithms has been to increase overall system throughput by efficiently using multiuser diversity and at the same time, divide the resources among subscribers fairly and equally. In this part, a brief literature survey about existing scheduling algorithms will be discussed in detail.

2.1.1 Max C/I Rule

Maximum carrier-to-interference (Max C/I) scheduler [9] is one of the well-known and widely used schedulers in the literature. In 2, we explained that each UE needs to report its CQI to the serving base station. Max C/I scheduler operating in the MAC layer of the base station makes use of CQI values in the decision process. These values are ranked in a descending order and system resources are allocated to the user with highest CQI value. If there are any resources left after serving the user, second ranked user is also served and so on. Surely this approach maximizes the utilization of multiuser diversity and overall network throughput, as the selected

Table 1: CQI to TBS Mapping for Category 14

CQI	TBS	Required # of Codes	Modulation
0	N/A	Out of range	Out of range
1	136	1	QPSK
2	176	1	QPSK
3	232	1	QPSK
4	320	1	QPSK
5	376	1	QPSK
6	464	1	QPSK
7	648	2	QPSK
8	792	2	QPSK
9	928	2	QPSK
10	1264	3	QPSK
11	1488	3	QPSK
12	1744	3	QPSK
13	2288	4	QPSK
14	2592	4	QPSK
15	3328	5	QPSK
16	3576	5	16-QAM
17	4200	5	16-QAM
18	4672	5	16-QAM
19	5296	5	16-QAM
20	5896	5	16-QAM
21	6568	5	16-QAM
22	7184	5	16-QAM
23	9736	7	16-QAM
24	11432	8	16-QAM
25	14424	10	16-QAM
26	15776	10	64-QAM
27	21768	12	64-QAM
28	26504	13	64-QAM
29	32264	14	64-QAM
30	38576	15	64-QAM

users will have the highest CQI values, resulting in highest possible transmission rates for the corresponding TTI. This scheduler is unfair since the users with good channel conditions will likely to dominate the limited wireless resources. In fact, the users at cell edges may starve for service considering that channel quality is highly related to the distance between the transmitter and the receiver.

2.1.2 RR rule

Round-Robin (RR) scheduling is another well-known scheduling rule in the literature [4]. As opposed to MaxC/I rule, RR scheduler is known by its fairness. RR algorithm schedules active users by selecting each one of them in sequence with equal service time. One issue with RR scheduling is that it does not consider subscribers' instantaneous channel qualities, therefore RR fails in exploiting multiuser diversity which causes low throughput across the network in overall. Another issue is that the concept of "fairness" should not be considered as allocating each user the same amount of system resources. In this case, users with streaming traffic and best-effort traffic are given equal service which in turn means equal importance.

2.1.3 PF Rule

Several fair schedulers are proposed for wireless packet data networks. Among these, the most popular scheduler in the literature is the proportional fair (PF) scheduler [8], [22], [23]. PF scheduler tries to maintain a balance between conflicting goals of maintaining a near-optimal network sum-rate capacity and allowing all users a fair access to system resources. By storing the average received data histories for each user over a certain size of sliding window, scheduler selects the users with highest achievable data rates relative to their data histories. PF scheduling algorithm calculates the rank of the corresponding user i from a total number of N users at a given TTI n as ;

$$P_i(n) = \frac{DRC_i(n)}{\overline{R_i(n)}}, \quad i = 1, 2, \ldots, N \tag{4}$$

Here $P_i(n)$ is the ranking parameter, $DRC_i(n)$ is the *achievable* data rate for user i at $TTI\ n$ and $\overline{R_i(n)}$ is the received data history of user i. Serving base station updates user data histories as follows:

$$\overline{R_i(n)} = \begin{cases} (1 - \frac{1}{w})\overline{R_i(n-1)} \ + \ \frac{1}{w}R_i(n), & \text{if scheduled,} \\ (1 - \frac{1}{w})\overline{R_i(n-1)}, & \text{o.w} \end{cases} \tag{5}$$

where $R_i(n)$ is the *assigned* rate to user i. note that $R_i(n)$ may differ from the *achievable* rate, $DRC_i(n)$, since the corresponding user may not have enough data in its buffer to support a transmission rate of $DRC_i(n)$, which imposes a necessity to lower $DRC_i(n)$ to a transmission rate that buffer can actually support, namely, $R_i(n)$. w is the sliding window size used for $\overline{R_i(n)}$ calculations and a typical w value is 1000 TTIs [22].

Note that PF scheduling algorithm still draws a large amount of attention both from academia and industry. Majority of cellular operators are deploying PF schedulers to their macrocell base stations and almost all research on wireless resource allocation include a comparison between the proposed framework and the PF rule. This is because PF algorithm leads almost equal service time distribution among users in the long run and produces higher overall network throughput than RR rule [24].

2.1.4 CQ-BMTA Rule

Authors in [25] propose the Channel Quality-Based Minimum Throughput Assurance (CQ-BMTA) scheduling algorithm. The purpose of this algorithm is to enhance the average throughput by increasing the scheduling rank of users with relatively low average throughput. In the proposed algorithm, the user with the highest priority is selected for transmission where the priority for user i at time t is calculated as follows:

$$\overline{P_i(n)} = \begin{cases} CQI, & \overline{R_i(n)} \geq C, \\ CQI \cdot W, & \text{O.W.} \end{cases} \tag{6}$$

Here, C is a predefined minimum throughput and $W = C/\overline{R_i(n)}$. According to above formula, users are ranked according to their instantaneous channel qualities if they achieve high throughput. However, if users average throughputs start dropping below C, their CQIs are multiplied by a weight (W) that is inversely proportional to their throughputs (i.e., W increases as a users average throughput decreases). Therefore, by introducing this weight, higher priorities are given to users with low average throughput, which increases the degree of fairness in the system.

2.1.5 Exponential Rule

Different variants and generalizations of the PF rule have been proposed in the literature. For example, the exponential rule (EXP-Rule) scheduler [12], [26], is a popular variant to PF rule which also considers the packet delays of the user-based queues. For the n-th TTI, the rank of user i is calculated by EXP-Rule scheduler as;

$$P_i(n) = a_i \frac{DRC_i(n)}{\overline{R_i(n)}} \exp\left(\frac{a_i d_i(n) - \overline{ad(n)}}{1 + \sqrt{\overline{ad(n)}}}\right) \tag{7}$$

where

$$\overline{ad(n)} = \frac{1}{N} \sum_{k=1}^{N} a_k d_k(n) \tag{8}$$

Here, $d_k(n)$ is the head-of-line (HOL) packet delay of user k at TTI n and $a_k > 0$,$k = 1, 2, \ldots, N$ are the assigned weights to set each user's desired quality of service [26]. Exponential part of the rule differentiates this scheduler from PF rule by means of weighted packet delays for each user. The exponent term will dominate the formula and increase the ranking of user i if the weighted packet delay of user k at TTI n,

$ad_k(n)$, exceeds $\sqrt{\overline{ad(n)}}$. If on the other hand, delay is small compared to $\sqrt{\overline{ad(n)}}$, then the exponent term will approach to 1 and the rule will operate as a PF scheduler. Relative to the individual packet delays of each user, the EXP scheduler incorporates the effects of both channel variations and service delay for the resource allocation stage.

2.1.6 M-LWDF Rule

A limited number of studies on this topic actually focused on fulfilling the QoS requirements of different applications and attempt to integrate them in the scheduling phase. One such scheduler is the modified largest weighted delay first (M-LWDF) [13] scheduler. M-LWDF rule calculates the rank of a user by multiplying each user queues' HOL, CQI and the predefined constant which is determined according to the service type of application in the queue. Therefore, queues can be prioritized with these predefined constant parameters and HOL delays are also used not to starve users with long waiting times.

Several other QoS-enabled scheduling algorithms also exist in literature. In [14], authors propose to map packet arrival rate, loss and delay to a three dimensional QoS-deviation metric and aim to tune between QoS provisioning and throughput optimizing in an adaptive manner depending on the current cell QoS-deviation level. Authors in [15] proposed a cross-layer design where the scheduler as well as the video transmission rate adaptation are optimized jointly with the goal of minimizing the number of pauses during playback, maximizing the video quality. It should be stressed here that in all of these studies, QoS requirements are described on user level and the main thrust is to share the available bandwidth among a large number of users with high mobility probabilities. Since femtocell subscribers have very low, if any, mobility, and are likely to have multiple active applications running, some, but not all which are QoS sensitive, the prior art schedulers will not be very suitable for a HNB.

To our knowledge, there has been no prior art in a scheduler design for a femtocell base station.

2.2 System Model

In this thesis, we focused mainly on a 3-tier cellular environment with each MNB having 1 km coverage radius and maximum transmission power of 43 dBm. We assume the HNB coverage area is 10 meters in radius and is operational at a random location within the center cell with a maximum transmission power of 13 dBm. We further assume that HNB uses the same carrier frequency as the MNBs without performing any interference avoidance or management techniques. We consider HSDPA Release 10 [21] without MIMO or dual cell configured, which allows data rates up to 19.2 Mbps. All UEs are assumed to be Category 14 devices. In other words, all users are capable of encoding and decoding all modulation schemes described within the HSDPA Release 10.

2.2.1 Data Traffic in Cellular Networks

According to [27], all data services can be categorized into 4 different QoS classes, namely, conversational, streaming, interactive and background. Services belonging to conversational class are highly delay sensitive while background services are the most delay insensitive applications. Both conversational and streaming classes require minimum bit rate guarantee while background class is highly throughput-oriented. Primary characteristics of the interactive class on the other hand, is the request response pattern. Conversational and streaming classes are more error-resilient, meaning, bit errors are tolerable since human perception cannot detect audio/video artifacts up to a certain level. Interactive and background classes have low tolerance to errors and payload content must be preserved for these classes of services. Conversational services include IP telephony and video conferencing whereas streaming services include real time or non-real time video streaming. Web browsing and online games are typical interactive services. Finally, all types of best-effort, delay intolerant services like email services and file download can be considered as background services.

An example of multiple data traffic requested by a subscriber is given in Fig. 2. Here, user requests a Skype session (conversational), a web page (interactive) and a file download (background) from the Internet simultaneously. Variance of packet inter-arrival times of Skype traffic is considerably low. Packet inter-arrival times of download traffic on the other hand, fluctuates dramatically. User demands an uncached web page around the beginning of the session and this results in a data rate burst as seen from the figure.

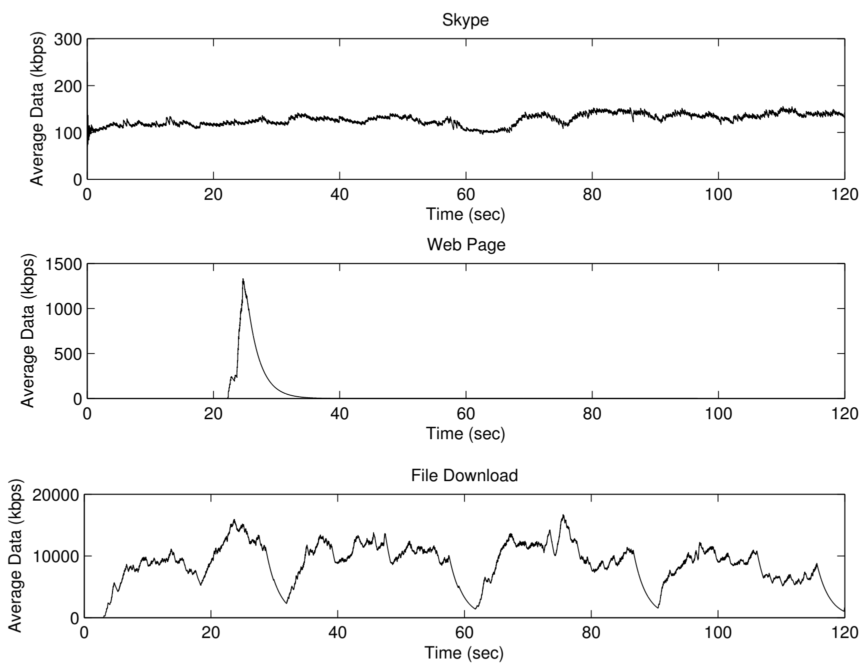

Figure 2: User 1 Data Traffic

2.2.2 Wireless Channel Models

The wireless channel imposes time-varying impairments in the form of thermal noise, interference, path loss, shadow fading as well as multipath fading to the communication link. In the system model, we consider all of these impairments. In order to model interference realistically, macrocell user equipments (MUE) are also assumed to be present in the center cell and mobile radio links between MNB and MUEs as well as links between HNB and femtocell user equipments (FUE) are modeled accordingly.

2.2.2.1 Path Loss Models

Walfisch-Ikegami Model is used for the path between MUEs and MNB and the model can be found in detail at [28]. The mixed indoor-outdoor model stated in [29] is used for computing the path loss from a MNB to the indoor FUEs. Finally, Multiwall Model [29] is considered in modeling the path loss between HNB and the associated FUEs.

2.2.2.2 Shadow Fading

Shadow fading is generally characterized by its one-dimensional distribution function that can be modeled as a log-normal distribution. We use an autocorrelation model for shadow fading proposed by Gudmundson [30]. Since the model consists of both indoor and outdoor environment, we use two different standard deviations of 8 dB and 4 dB for the outdoor and indoor connections, respectively.

2.2.2.3 Multipath Fading

We assume an urban environment for our simulations, therefore we use a Rayleigh fading channel considering that the existence of line-of-sight (LOS) component is less likely. In the same way for femtocell coverage area, Rayleigh fading is more likely to be modeled. In other words, Rayleigh fading channel is assumed for both indoors and outdoors.

CHAPTER III

APPLICATION-AWARE QOS SCHEDULER

For indoor femtocell networks, we propose a QoS-enabled scheduler that schedules among incoming flows rather than users. The incoming femtocell traffic is queued in application-specific queues rather than user based queues. For this purpose, the femtocell is assumed to perform some form of packet inspection. The incoming IP packets are sorted according to the applications of each user. To be able to do this, the source address, destination address, application specific traffic patterns may be used.

At a given TTI n, scheduler calculates the ranks of i^{th} user's j^{th} application as;

$$P_{ij}(n) = \left(\frac{DRC_i(n)}{\overline{R_i(n)}} \right) \cdot \exp\left\{ T[\alpha_{ij}(n)] + d_{ij}(n) \right\} \tag{9}$$

Here, $\alpha_{ij}(n)$ is the average *outgoing* HNB data rate of i^{th} user's j^{th} application calculated over a w of 500. T is a class specific, time-varying staircase function which determines the QoS requirement of the application at any given time. The window size for $\alpha(n)$ is selected such that the fluctuations of the incoming traffic rate for each service is captured more rapidly.

The PF part of the formula exploits the channel variations of each user and therefore provides fairness amongst users whereas the second part alters the priority of the corresponding application rapidly with respect to the average received data rate of the application. On the other hand, d parameter ensures the transmission of backlogged data in case the scheduler fails to transmit application traffic in a given window of w.

Sliding window size for averaging incoming data traffic holds great importance in terms of detecting throughput fluctuations of applications. There is a tradeoff between the window size and how fast we capture the fluctuations of a packet flow.

Small window size helps the system capture the variations in traffic better, but it should also be large enough to capture the bursty nature of the traffic. The streaming and conversational traffic are not throughput-oriented and it is unnecessary to send the incoming data to the subscriber immediately since these types of traffic usually requires an average constant bit rate. For this reason, it is not needed to capture the changes in these two traffic in an urgent manner. These bursty traffic can be spread in time and sent to the subscriber. However, interactive classes of traffic requires

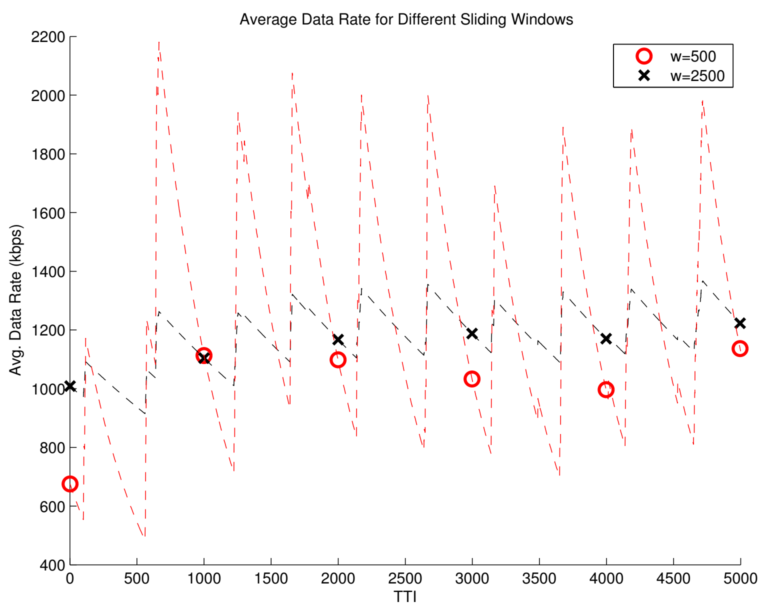

Figure 3: Avg. Data Rate with Two Different Sliding Windows

immediate transportation of the incoming data as well as background type of traffic. Incoming traffic averaged over two different window sizes is given at Fig. 3 Here, the averaged traffic is actually the same in both cases but the variations are much more obvious in the case with $w = 500$. This is because each arriving packet contributed to the average calculation by 1/w of its payload. With w being small, an incoming packet with a large payload can greatly increase the calculated average rate of the traffic. Therefore, window size for our algorithm also needs to be optimized for a better scheduling performance. A simplified representation of transmission of flows

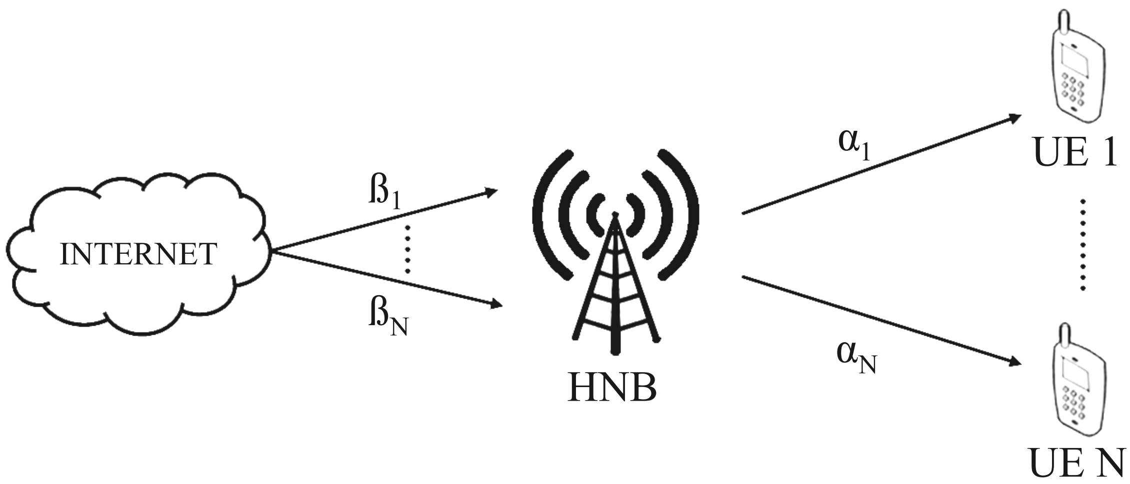

Figure 4: Overall Communications System

to the subscribers over a HNB is given at Figure 4. Here, β_i and α_i define the average HNB *incoming* and *outgoing* data rates for user i. In this work, we only consider the HNB part of this scheme and we focus on transmitting incoming flow of every application from HNB to the end users without introducing further packet delays that may cause the application unable to fulfill its QoS requirements, i.e., the aim is to equalize β_is with the corresponding α_is if the wireless system resources are adequate for this purpose. We also assume that β_is arrive to HNB as if the flow fully represents the data characteristics of the corresponding service, meaning, flow arrives to the HNB with no packet loss and the flow fulfills the QoS requirements for that service.

3.1 Comfort Zone Assignment

Estimation of the required bandwidth for each types of traffic classes and applications are possible with traffic monitoring. Therefore a comfort zone for each application can be determined for which the corresponding application satisfies a certain amount of QoS. Comfort zone can be defined for i^{th} user's j^{th} application as,

$$\sigma_{u,ij} > \alpha_{ij}(n) > \sigma_{l,ij} \tag{10}$$

where $\sigma_{u,ij}$ and $\sigma_{l,ij}$ are constant-valued upper and lower bounds determined by traffic monitoring for application j, respectively.

3.2 Class Specific T Parameter

In this work, T is modeled as a staircase function to ensure the corresponding application's average received rate to stay in the comfort zone. Every traffic class have a positive and a negative T value in our design. According to the whereabouts of $\alpha_{ij}(n)$ in terms of the application's comfort zone, T becomes positive, negative or zero. An example of a two-step T function is given at Figure 5. Here, the comfort zone is de-

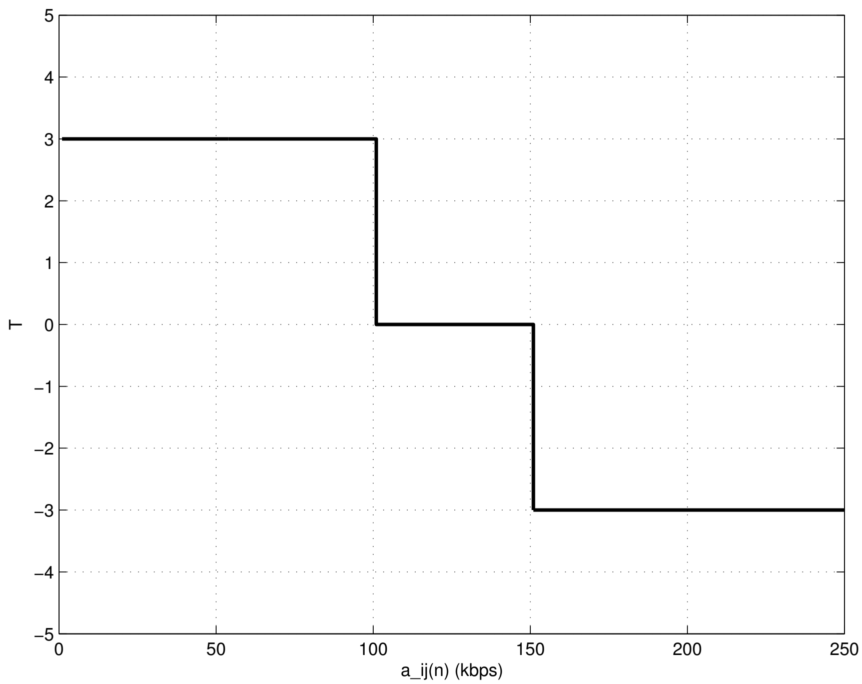

Figure 5: Sample T Function for Class X

fined to be between 100 kbps and 150 kbps. If the received rate of the corresponding application is lower than 100 kbps, then T will become 3 and the ranking of the application is increased by e^3. On the other hand if the rate is above the comfort zone, T will be updated as -3 and therefore the ranking will be decreased accordingly. If the incoming rate is in the comfort zone, then class specific T values should be chosen considering the QoS requirements of each traffic class. Since the most delay intolerant applications belong to conversational class, T high value of conversational class, $T_{c,h}$, is kept very high whereas the background low, $T_{b,l}$, and high, $T_{b,h}$, T values are kept very low which means all the available channelization codes, after transmitting to high priority flows, are assigned to background applications.

3.3 Adaptivity

In our prior work, application specific bounds are calculated as if the flow statistics are completely known at the scheduler [31]. Therefore bounds are kept constant during the entire flow transmission, but due to the bursty nature of IP traffic and wireless channel impairments, bound calculation has to be done dynamically at the transmitter side in order to form the comfort zones accurately. For adaptive bound calculation, we update the comfort zone definition as,

$$\sigma_{u,ij}(n) > \alpha_{ij}(n) > \sigma_{l,ij}(n) \tag{11}$$

where $\sigma_{u,ij}(n)$ and $\sigma_{l,ij}(n)$ are upper and lower bounds for application j at a given TTI n. $\sigma_{u,ij}(n)$ and $\sigma_{l,ij}(n)$ are dynamically calculated at each TTI n as,

$$\begin{aligned} \sigma_{l,ij}(n) = &\quad \beta_{ij}(n) \cdot \left(1 - \tfrac{\epsilon}{100}\right) \\ \sigma_{u,ij}(n) = &\quad \beta_{ij}(n) \cdot \left(1 + (\tfrac{100-\epsilon}{100})\right) \end{aligned} \tag{12}$$

Here, ϵ is a class specific offset percentage for application j's UMTS QoS class and $\beta_{ij}(n)$ is the average HNB *incoming* data rate for application j of user i. Since the scheduler already has the knowledge of $\beta(n)$s' for every application, $\sigma_{u,ij}(n)$ and $\sigma_{l,ij}(n)$ can be easily calculated with predefined ϵ values. According to the above bound calculation, each flow is surrounded by a window that is ϵ% lower and $(100 - \epsilon)$% higher than the average flow rate of the corresponding application, which in turn, defines the comfort zone for that application. A comfort zone assignment for an actual Skype teleconferencing session flow with $\epsilon = 20$ is given at Fig. 6 where $\sigma_{u,s}(n)$ and $\sigma_{l,s}(n)$ are upper and lower bounds for Skype flow's comfort zone, respectively.

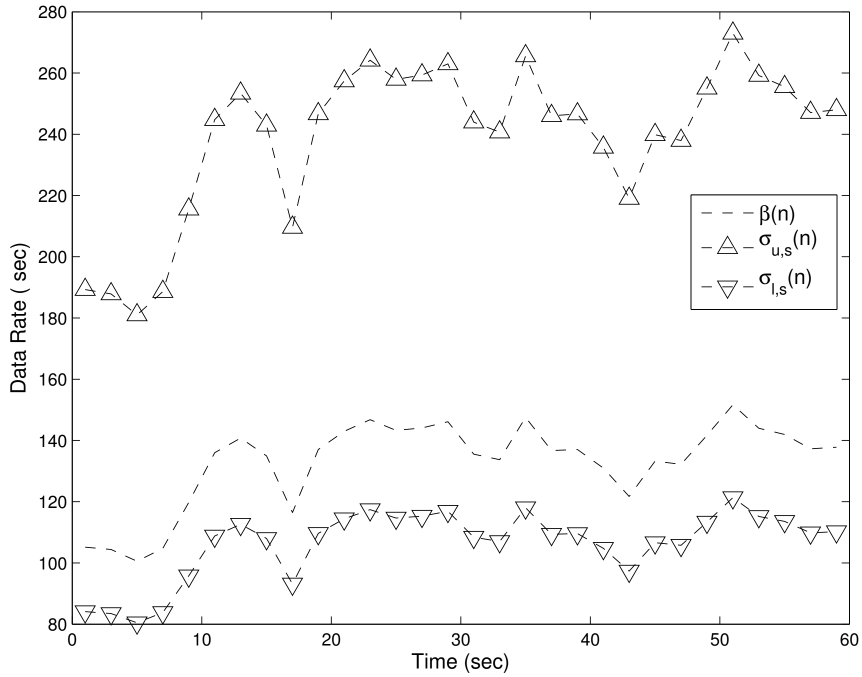

Figure 6: Comfort Zone Assignment for Skype Traffic

The relaxation from the actual traffic flows depends on the QoS sensitivity of the type of traffic under consideration. This is specified by ϵ for our scheme provided in this study. Low ϵ percentages result in a lower bound for comfort zone that is closer to HNB incoming rate for that application. In other words, tolerance to the rate drop is much more less than the case with higher ϵ percentages. Obviously, ϵ for conversational class should be minimum whereas for background services, ϵ can take relatively high percentage values.

Together with the T parameters, class-based ϵ values form the system's current operating mode, Ω, which is defined as;

$$\Omega = \{(T_{c,l}, T_{c,h}, \epsilon_c), (T_{s,l}, T_{s,h}, \epsilon_s), (T_{i,l}, T_{i,h}, \epsilon_i), (T_{b,l}, T_{b,h}, \epsilon_b)\} \tag{13}$$

where the first subscript indicates the belonging QoS traffic class and second subscript refers to *high* and *low* values of the corresponding parameter.

3.4 Resource Allocation Procedure

For every TTI, upon receiving all of the CQI values from the users, the scheduler calculates $P_{ij}(n)$ for every active application of each user. The application with the largest value is therefore scheduled. The scheduler then maps the CQI value of the scheduled user to a corresponding TBS value from Table 1. Associated with this value is the number of code channels necessary to service this application. If the necessary number of code channels is 15 then only this application is scheduled at that time slot. If it is below 15, the HNB may schedule additional applications simultaneously. Then the second ranked application is considered. If there are still remaining resources, the third ranked application is considered and so on. At a given time, if the data waiting in the application queue of the scheduled application is less than what may be transmitted with the associated CQI value, a lower CQI may be allocated for this application. This in return opens up resources for that time slot that may be used for other applications. Code allocation scheme at a given time slot n is presented in Algorithm 1. Here, P_{ij}s are calculated according to the proposed scheduling rule at (9) prior to code allocation step for scheduled applications. Along with the P_{ij} values, CQI feedbacks from all HNB subscribers are inputs to the scheme. P is the sorted index array of all P_{ij}s, i.e., P stores the index of applications to be scheduled in descending order and the algorithm allocates available resources starting from the application $P(s)$ $(s = 1)$ and s is the index of the scheduled application at n. DRC and CODE are the resulting *instantaneous* data rate and necessary number of codes to support this rate for a given CQI belonging to the application s, respectively. If the scheduled application s has sufficient backlogged data, the corresponding application will be served. Otherwise, the CQI feedback will be lowered until the backlogged data is higher than the assigned rate for that application. If the application queue has sufficient data but assigned number of codes exceeds $availableCodes$ at the HNB, CQI feedback for that user again is reduced until number of assigned codes matches *availableCodes*. After this process is completed, application s is allocated a rate of DRC and a subset of channelization codes CODE and after *availableCodes* is updated, resource allocation process continues if there are unused codes left at the HNB and HSDPA service limit (up to 4 users) will not be violated.

Algorithm 1 Code Allocation Algorithm

```
Require: P_ij , CQI_i
procedure SCHEDULING AT TTI n (P_ij, CQI_i)
    P ← Sort P_ij                                  ▷ Rank applications
    x ← 1
    while availableCodes > 0 do
        s ← P(x)                                   ▷ s is the scheduled application
        CQI ← CQI_i(s)
        DRC ← calculateDRC(CQI)
        CODE ← calculateCODE(CQI)
        while DRC > queue(s) do
            CQI ← CQI − 1
            DRC ← calculateDRC(CQI)
            CODE ← calculateCODE(CQI)
        end while
        while CODE > availableCodes do
            CQI ← CQI − 1
            DRC ← calculateDRC(CQI)
            CODE ← calculateCODE(CQI)
        end while
        x ← x + 1                                  ▷ schedule next application
        availableCodes ← availableCodes − CODE
    end while
end procedure
```

CHAPTER IV

MULTIPLE-OBJECTIVE OPTIMIZATION (MOO)

4.1 Introduction

Multiple-objective optimization, introduced by Pareto is concerned with finding solutions to optimization problems with multiple objectives $F = \{f_1, f_2, \ldots, f_K\}$. The objectives are possibly conflicting, therefore a unique solution is not possible for our resource allocation problem. A solution is called *Pareto-optimal* if any one of the objectives cannot be improved without sacrificing the other objectives for this solution [32, 33]. In this study, we determined pareto-optimal solutions for best operating point not only focusing on individual QoS classes but also considering the joint system performance.

For single objective optimization problems, it is possible to have multiple optimal solutions resulting in a unique optimal functional value. It is also possible to have multiple Pareto-optimal solutions in multi-objective optimization problems. However, unlike the single objective problems, the multiple Pareto-optimal solutions do not necessarily result in a unique functional value. In many instances, as different objective functions represent different system aspects on a specific scale, variance, and units of measurement, it is difficult to discriminate between these Pareto-optimal points and determine which one is better than the others. However, if relative importance weights for each of the objective functions is specified, a so-called best compromise solution may be determined.

Let us assume that our joint scheduling problem consists of K objective functions and that each of them needs to be minimized. Then, we say that a Pareto-optimal solution, $\eta*$ exists if there is no other feasible solution, η, that achieves,

$$f_k(\eta) \leq f_k(\eta*), \qquad \forall k \in \{1, 2, \ldots, K\} \tag{14}$$

with at least one strict inequality, i.e., the Pareto-optimal solution, $\eta*$ is the best solution for the problem at hand in terms of the joint benefit of all objective functions at hand. In multiple-objective optimization, an infeasible solution that minimizes all the objective functions simultaneously is called the *utopia point* [15], and the Pareto-optimal solution has to have the shortest distance to the predefined utopia point, so the MOO framework stages can be listed as;

1. Define the objective functions(K in total)

2. Normalize the objective function $k, \forall k \in \{1, 2, \ldots, K\}$ as;

$$f_{k,scaled} = \frac{f_k - f_{min}}{f_{max} - f_{min}} \tag{15}$$

3. Calculate the K-dimensional euclidean distance to the utopia point for every system operating mode, Ω.

4.1.1 Problem Formulation

Application flows belonging to different types of QoS classes impose different QoS requirements. A streaming application is highly prone to jitter whereas a web application is more throughput oriented. Besides the various QoS requirements of data services, *quality of experience* (QoE), which is defined in [34] as: *"the overall acceptability of an application or service, as perceived subjectively by the end-user."*, also needs to be considered for the MOO framework. From a user point of view, conversational and streaming applications are required to experience minimum number of pauses with minimum durations. Web applications on the other hand, are expected to download the web page to the browser in a best-effort sense as quickly as possible. For our MOO framework, objective functions are formulated based on QoE of different types of QoS traffic classes and algorithm parameters are optimized accordingly.

4.1.2 Objective Functions

For conversational and streaming applications, the objective functions are determined as *average number of pauses* and *maximum duration of a pause* which are intended to be minimized. For interactive QoS class, we considered web applications where user demands an uncached web page and the objective function is set to be the *average page loading time* which also needs to be minimized. Finally for the background applications, the objective function is determined as the *total downloaded data on average* and this function obviously needs to be maximized as opposed to the other objective functions. So in our case there are 6 objective functions present in the system ($K = 6$) and some of them are contradictory which dictates that a *Pareto-optimal* solution needs to be found according to (14).

4.2 Optimization Stage

MOO framework designed in this study is a two-stage optimization. At first stage, all ϵ percentages are kept constant and T parameters for every QoS class are optimized accordingly. After T optimization is completed, the optimized T values are used in the next stage of ϵ optimization. In this section, first we provide optimum operating points considering the maximum benefit of each QoS class one at a time. Then, we also yield a $globally - optimal$ operating point where all QoS classes are jointly considered in the MOO framework.

4.2.1 T Optimization

For T optimization stage, distances from utopia point are calculated for all the operating points. Here, all ϵ percentages are kept constant and for practical purposes, $T_{b,h}$ is fixed at minimum so that background services are scheduled having the lowest rank. Figures 7-10 represents class-based MOO framework with distance to utopia point versus burden to the other objective functions for each operating point. The operating points marked as 'x' indicates the optimum operating point, OOP, for the corresponding QoS class. Resulting T values for every class is also given at Table 2.

Aside from calculating OOPs for each QoS class, we also derive the OOP with the case where all the objective functions are jointly minimized with respect to the utopia point and the distance to utopia point for every operating point is given at Fig. 11. Resultant T values, together with the optimal ϵ percentages are presented in Table 4.

For conversational traffic, we observe that operating points reside roughly in four distinct areas. Above two areas, where the cost for increasing conversational traffic performance is high, are because streaming and interactive T_h values are equal to 1. This means that these two traffic classes are performing poorly when the system focuses only on conversational traffic combining the low T_h values for other types of traffic. When $T_{s,h}$ and $T_{i,h}$ are higher than 1, cost decreases significantly since the system assigns higher T_h values when streaming and interactive applications perform poorly in terms of incoming traffic rate.

The same phenomenon appears also for optimization of streaming traffic class. Four areas are observed in terms of cost-distance tradeoff. the cases with high system cost is because of the performance of conversational applications. At these cases, $T_{s,h} = T_{c,h}$, therefore streaming can override conversational and cause high system cost on the remaining of the system. low cost - low performance case is because of low $T_{s,h}$ and high $T_{c,h}$ therefore performance of streaming applications decreases but since $T_{c,h}$ is high, remaining system can perform well. In low cost high performance case (lower-left area of Fig. 8), $T_{s,h}$ is higher than $T_{i,h}$ but lower than $T_{c,h}$ therefore both streaming applications and the remaining system can perform well.

We do not allow T value of interactive application to exceed neither $T_{s,h}$ nor $T_{c,h}$.

Therefore interactive applications depend heavily on the parameters of streaming and conversational applications which makes the performance of interactive services to spread rather uniformly as depicted in Fig. 9.

Fig.10 shows all the operating points for background traffic class. Here, operating points behave differently than other traffic types. This is because, we assume that background traffic is the least important traffic since it operates on best-effort logic and dominates all other types of traffic if not controlled carefully. We observe that whenever $T_{s,h}$ increases, performance of background applications suffer greatly since streaming traffic requires relatively higher bandwidth and consumes most of the wireless system resources. However, performance of background applications resides very close to the utopia point when $T_{s,h}$ is set to 1. Therefore, the "line-shaped" operating point area is observed when $T_{s,h} = 1$.

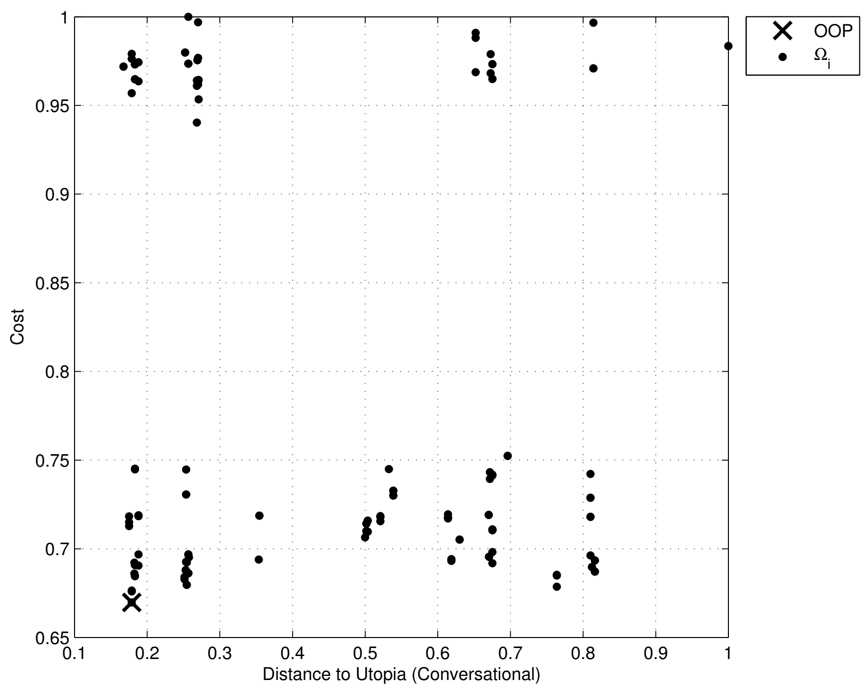

Figure 7: Distance from Utopia Point (Conversational) vs. Cost to Overall System - T Parameter

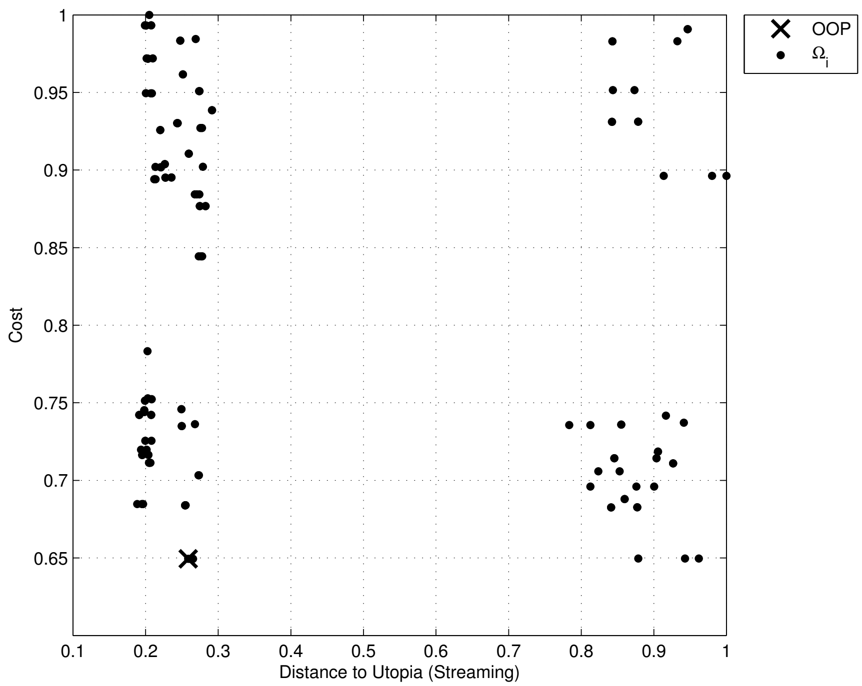

Figure 8: Distance from Utopia Point (Streaming) vs. Cost to Overall System - T Parameter

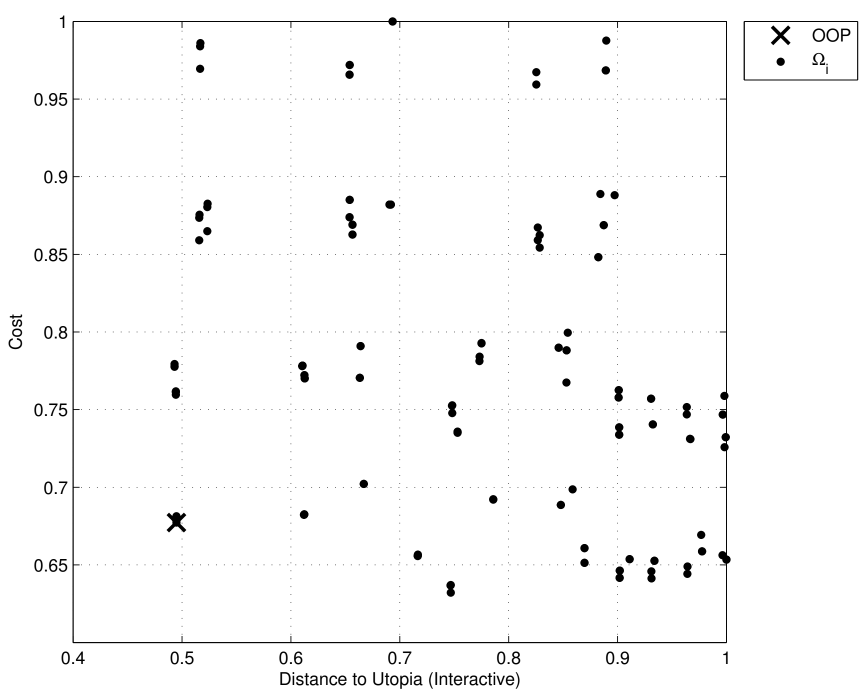

Figure 9: Distance from Utopia Point (Interactive) vs. Cost to Overall System - T Parameter

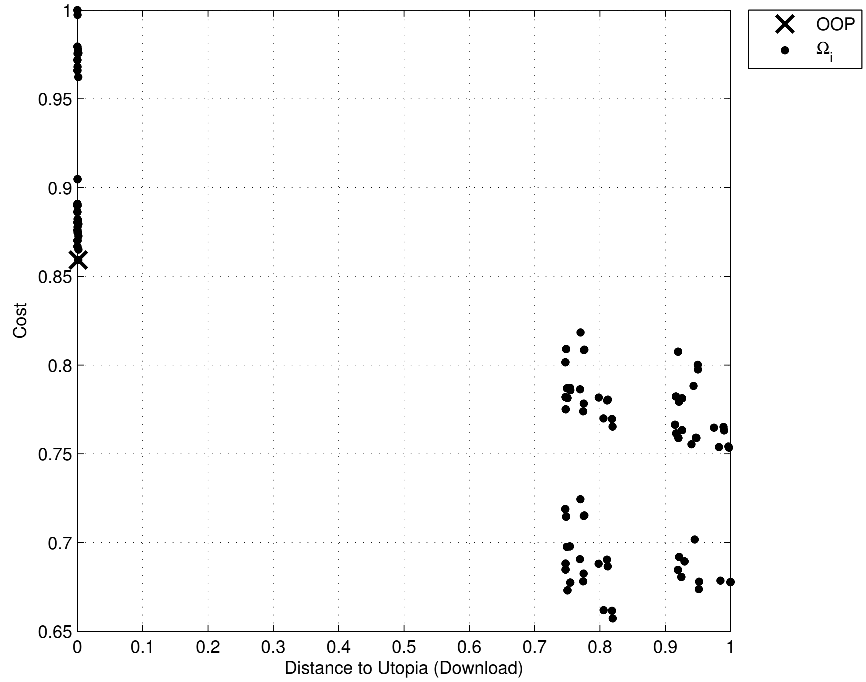

Figure 10: Distance from Utopia Point (Background) vs. Cost to Overall System - T Parameter

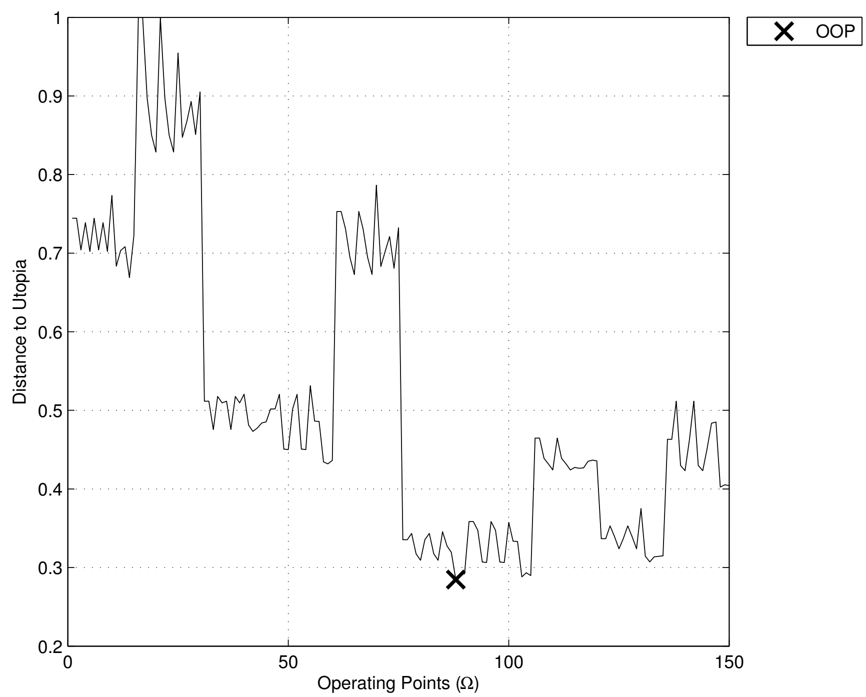

Figure 11: Distance from Utopia Point for each Ω - T Parameter

Table 2: Class-Based Optimal T Values

	$T_{c,h}$	$T_{c,l}$	$T_{s,h}$	$T_{s,l}$	$T_{i,h}$	$T_{i,l}$	$T_{b,h}$	$T_{b,l}$
Conversational	5	-1	3	-1	1	-1	0	-5
Streaming	5	-1	3	-1	3	-1	0	-5
Interactive	5	-1	3	-1	3	-1	0	-5
Background	5	-1	1	-1	1	-1	0	-3

4.2.2 ϵ Optimization

We followed the same procedure for optimizing class-based ϵ percentages. Again for practical purposes, ϵ_b is set to 100 which means the ranking of background applications are kept minimal. Figures 12-15 represent class-based ϵ optimization results and OOPs are found again based on the shortest euclidean distance to utopia point. ϵ values for QoS class-based optimization is given at Table 3. Normally, one would expect that optimum ϵ percentage for a QoS class x should be minimum whereas the remaining ϵ values should be maximum in order to maximize the performance of the given QoS class x. That would be the case if only concern was to minimize the objective function of x, but we followed the method of minimizing x's objective functions while carefully observing the class x's overall burden on the system. This enables us to retrieve $near - optimal$ operating points for that given class.

The case with joint optimization is slightly different. In that case, all objective functions are jointly minimized in terms of euclidean distance to utopia point which is presented in Fig. 16. We used these jointly optimized T and ϵ values in our performance comparison with other schedulers and these values are given at Table 4.

For conversational applications, we observe from Fig. 12 that almost all operating points beside few outliers reside very close to the utopia point. This is because the bandwidth requirement of conversational services are relatively low (100-150 kbps) and therefore variations of ϵ parameter of other applications do not change the performance of conversational applications significantly.

Again, we observe that the performance of streaming applications are very close to utopia point. The reason is the same with conversational case and conversational and interactive applications do not require enough bandwidth to be able to hurt the performance of streaming applications. Background applications could hurt the performance of streaming applications indeed, but as explained before, we do not allow $T_{b,h}$ higher than $T_{s,h}$ therefore the algorithm automatically prevents background applications overriding streaming applications.

Interactive applications on the other hand, are more susceptible to ϵ variations of other applications. When the ϵ percentage for streaming applications are lower, i.e., when the comfort zone for interactive applications is less favorable, streaming applications start hurting the objective function of interactive applications. Keep in mind that it would not be the case if we did allow $T_{i,h}$ to exceed $T_{s,h}$ which in turn would be more beneficial to interactive applications and more destructive for streaming applications.

Since background applications are assumed to be the least important, both T and ϵ parameters of background class are set at minimum level. Therefore, performance of background applications are highly correlated to other classes ϵ percentages. There are 5 areas which the operating points are grouped into as depicted in Fig. 15. The group that resides almost on the utopia point in terms of background performance is because of the large ϵ percentage of streaming class. At this area, streaming class percentage is %50 which in turn enables the system to allocate resources to background applications more frequently. whereas the group that performs the worst (the first group from right) is due to the %5 percentage assignment for streaming classes. Consequently, it is safe to observe that the performance of background applications are basically determined by ϵ percentages of streaming applications.

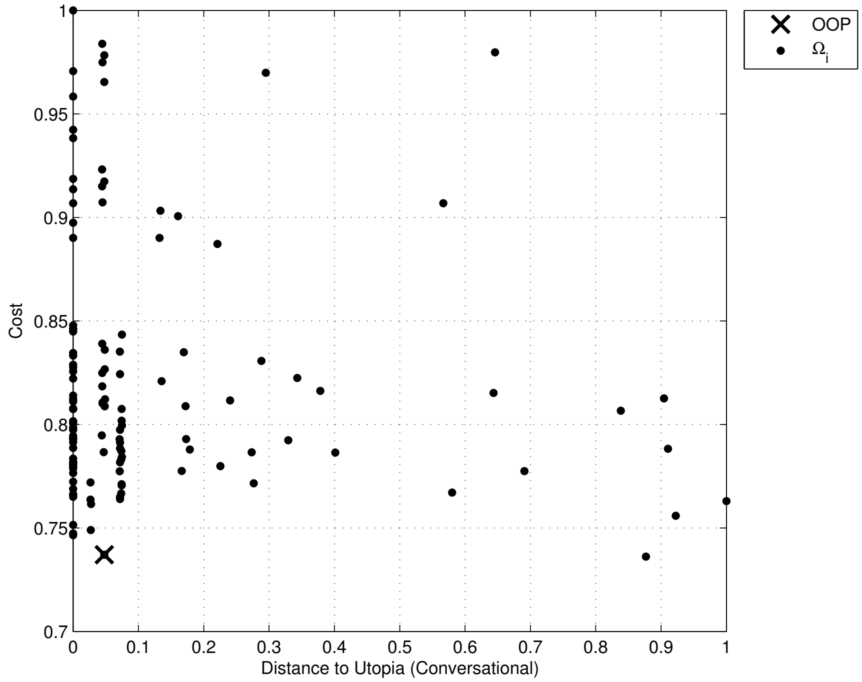

Figure 12: Distance from Utopia Point (Conversational) vs. Cost to Overall System - ϵ Parameter

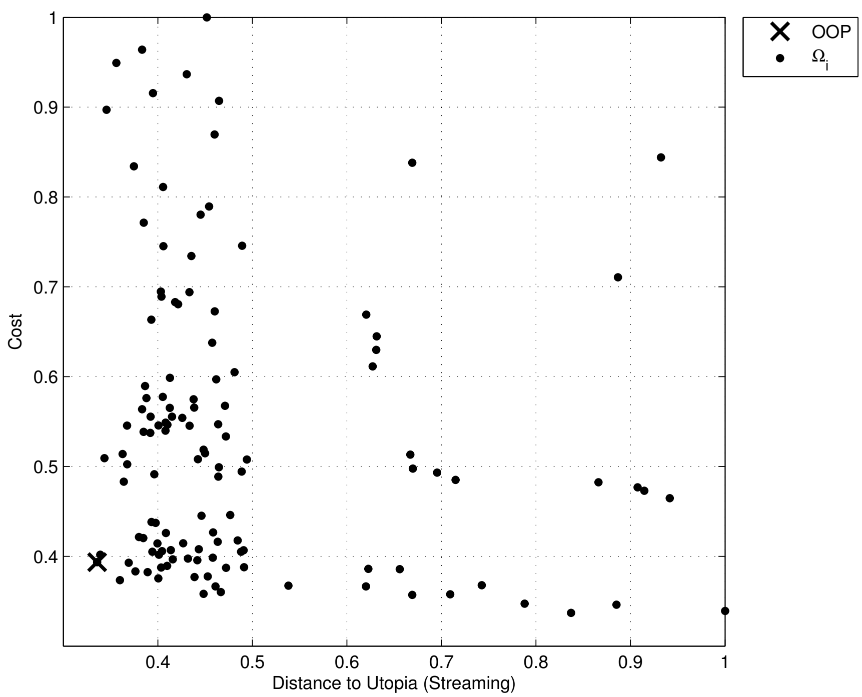

Figure 13: Distance from Utopia Point (Streaming) vs. Cost to Overall System - ϵ Parameter

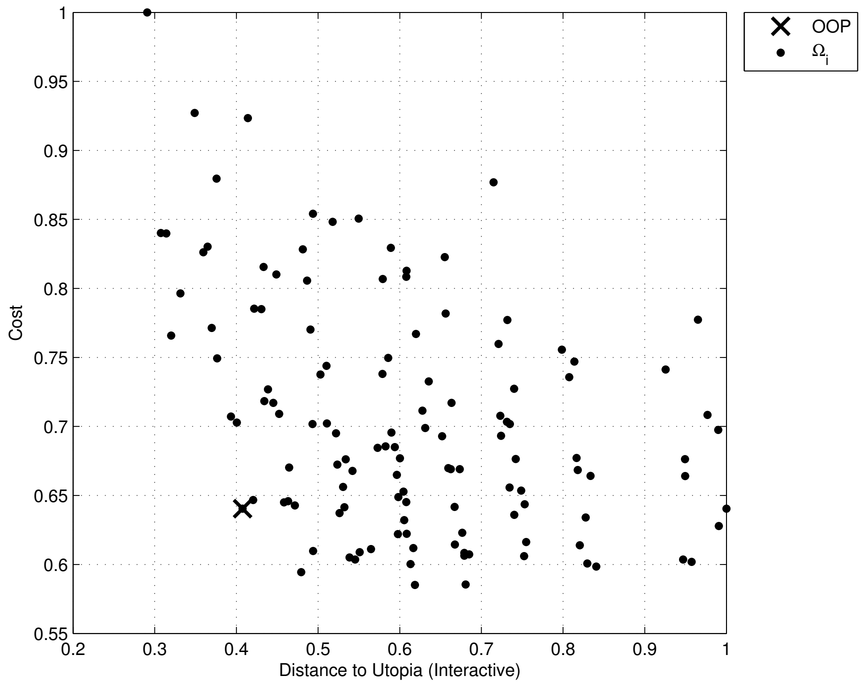

Figure 14: Distance from Utopia Point (Interactive) vs. Cost to Overall System - ϵ Parameter

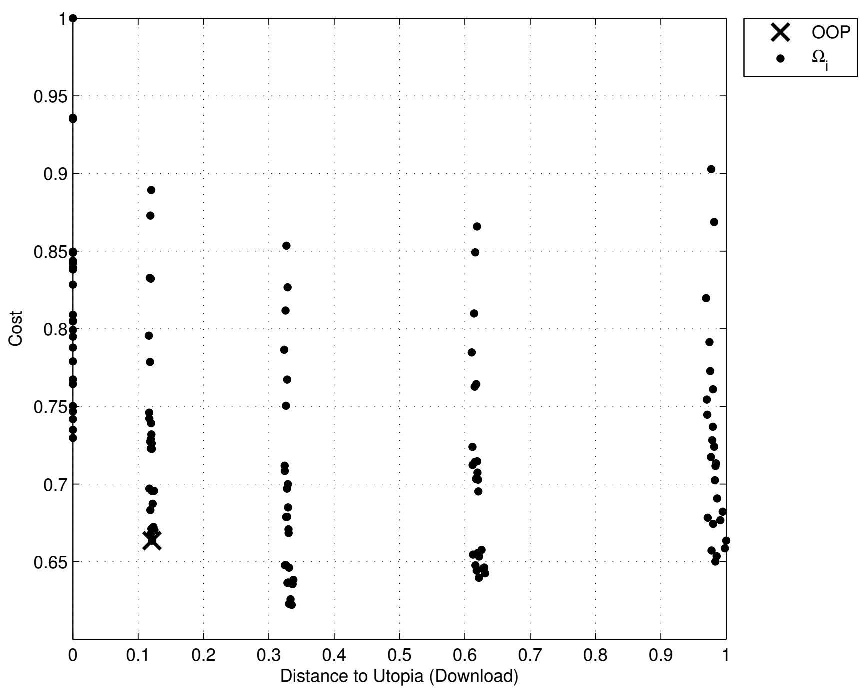

Figure 15: Distance from Utopia Point (Background) vs. Cost to Overall System - ϵ Parameter

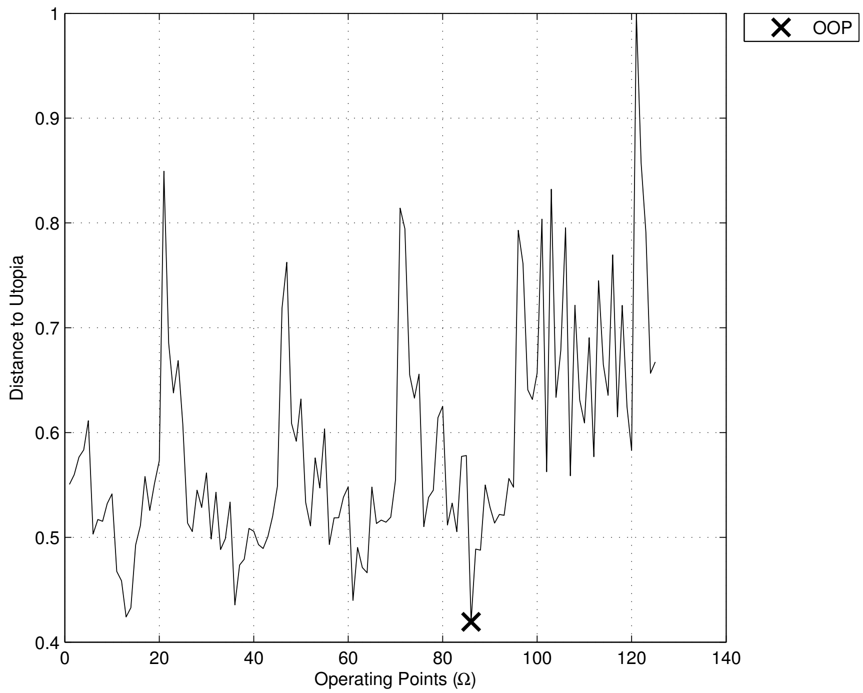

Figure 16: Distance from Utopia Point for each Ω - ϵ Parameter

Table 3: Class-Based Optimal ϵ Values

	ϵ_c	ϵ_s	ϵ_i	ϵ_b
Conversational	35	25	5	100
Streaming	5	25	25	100
Interactive	15	35	5	100
Background	15	50	5	100

Table 4: Jointly Optimal ϵ and T Values

	T_h	T_l	ϵ
Conversational	5	-1	35
Streaming	3	-1	25
Interactive	1	-1	5
Background	0	-5	100

CHAPTER V

PERFORMANCE ANALYSIS

Extensive simulations have been conducted to evaluate the performance of the proposed scheduler for the indoor femtocell as well as for the PF, Max C/I and Exp-rule schedulers. As a benchmark, performance of the system has been evaluated for cases where a single user with a single application is present in the system.

We modeled the packet traffic carefully considering the UMTS traffic classes and different types of data applications are chosen while performing the simulations of performance analysis. Simulation environment consists of 3 active users connected to the HNB within the center cell and that all users demand multiple data services simultaneously. User 1 is assumed to have an active Skype (conversational) video conferencing session, a web browsing (interactive) session as well as a large FTP file download (background) session. User 2 is assumed to have a web browsing session and a live video broadcast service (Tivibu) with a resolution of 640x480 (streaming). User 3 is assumed to have a live video broadcast session (Digiturk) with a resolution of 1280x720 and a large FTP file download session. In other words, all 4 UMTS traffic classes are considered in the phase of traffic modeling. In order to create a realistic traffic modeling, we used real packet traces collected from early mentioned applications. 3 computers were configured in the same network having a broadband connection with 50 Mbps to act like the 3 users in the setup. All associated applications for all users have been started simultaneously on the computers. Application traces were collected using WireShark Network Analyzer tool where the actual IP packet sizes and packet inter-arrival times have been recorded. Gathered traces are modeled as the incoming traffic to the corresponding HNB.

We assume that for streaming and conversational traffic, the user equipment buffer stores a certain amount of data before the playback starts in order to prevent excessive number of pauses during the course of the service. The time it takes to store the initial data is called the pre-roll delay. In our simulations, the pre-roll delay is calculated as the overall average received bits/(TTI·2000) which produces reasonable application-based pre-roll durations. Simulations have been repeated over a session duration of 2 minutes in 1000 different user locations within the femtocell. The following performance metrics have been used to asses the system performance:

- ◇ ***Conversational and Streaming Applications:*** Average and total number of pauses, average and maximum duration of pauses, average and maximum pre-roll delays.
- ◇ ***Interactive Applications:*** Average page loading time of an uncached web page.
- ◇ ***Background Applications:*** Average total file size downloaded over a 2 minute session

The results are tabulated in Tables 5-9.

5.0.3 Simulation Results

From the results, we first observe that the traditional user-based scheduling performance is unacceptable for an indoor femtocell scenario when only 3 users, each with multiple applications running, are present. Especially if a best-effort service is active for a user, remaining delay-sensitive applications present for that user greatly suffer since best effort traffic will eventually dominate the user-based queue. When we focus on user 1, we observe that with traditional schedulers operating on user-based queues, Skype and web applications fail to provide an acceptable QoE to the subscriber. This is because the file download application overpowers the other two applications and floods the user-based queue of user 1 at HNB side. In fact, only 34.2% of the locations within the cell can provide a no pause service to the Skype application with employing user-based queues and using MAX C/I scheduler. A similar conclusion applies also to user 3 who has both a file download and a live video broadcast service running. Pause durations as high as 19 seconds are observed for Digiturk application of user 3 when traditional resource allocation methods are employed. The web browsing applications belonging to interactive QoS class are also negatively impacted by the presence of background class traffic. Web page loading times of 21 seconds on the average are clearly unacceptable concerning the QoE for the subscribers.

We also observe that application-based queues and associated QoS-aware scheduler that we proposed result in significant gains to the overall system performance. All applications except the background class perform exceptionally well. In fact, the performances of each of these applications are almost identical to where that application has access to all system resources at all times, i..e. no other user and application is present in the system. For example, the Digiturk application has pause free performance in 99.4% of the femtocell coverage area, compared to 100% for a single application scenario. Note here that for Skype and Tivibu applications, results for proposed algorithm are identical to the single application case. The tremendous performance gain due to the user-based queues and proposed scheduler comes at the expense of a modest drop of at most 8% in the overall file download size compared to the traditional user-based queues with PF, Max C/I or EXP-rule schedulers.

Besides the throughputs of individual applications, another immensely important performance metric is the overall system throughput. We are perfectly aware that this metric is widely used among the studies involving scheduler design. This is why we also consider this metric as another performance indicator. As mentioned before, each scheduler is run 1000 times with different channel conditions and each run is 2 minutes. Therefore total network throughputs of these 1000 different runs

	Max C/I	PF	EXP-Rule	Proposed	Single App.
Average Number of Pauses	92.85	108.29	111.65	0	0
Average Duration of Pauses	0.193 sec	0.223 sec	0.197 sec	0 sec	0 sec
Maximum Observed Pause	5.748 sec	8.724 sec	4.304 sec	0 sec	0 sec
% Geography with No Pause	34.2%	11%	11.8%	100%	100%
Average Pre-Roll Delay	4.903 sec	4.903 sec	4.905 sec	4.363 sec	4.356 sec
Maximum Pre-Roll Delay	10.206 sec	10.206 sec	10.168 sec	4.396 sec	4.356 sec

Table 5: Performance of the Skype Application

	Max C/I	PF	EXP-Rule	Proposed	Single App.
Average Number of Pauses	9.01	5.85	0.52	0	0
Average Duration of Pauses	0.332 sec	0.362 sec	0.611 sec	0 sec	0 sec
Maximum Observed Pause	17.704 sec	17.640 sec	8.468 sec	0 sec	0 sec
% Geography with No Pause	57.5%	63.5%	86.9%	100%	100%
Average Pre-Roll Delay	3.568 sec	3.568 sec	3.557 sec	3.482 sec	3.417 sec
Maximum Pre-Roll Delay	5.454 sec	5.454 sec	4.746 sec	3.994 sec	3.800 sec

Table 6: Performance of the Tivibu Application

are averaged and labeled as ”average total system throughput” and these values are given in Table 10.

It is clearly seen from Table 10 that using our proposed scheduler instead of traditional schedulers has almost no negative impact on the total system throughput. Highest throughput has been reached with Max C/I scheduler but even with this scheduling scheme, the throughput difference is around 20 Mbits which is highly tolerable considering the gains of delay-intolerant conversational and streaming applications.

	Max C/I	PF	EXP-Rule	Proposed	Single App.
Average Number of Pauses	14.42	10.78	7.92	0.01	0
Average Duration of Pauses	0.826 sec	0.812 sec	0.916 sec	0.137 sec	0 sec
Maximum Observed Pause	19.050 sec	16.704 sec	11.604 sec	0.256 sec	0 sec
% Geography with No Pause	43.1%	43.5%	49.1%	99.4%	100%
Average Pre-Roll Delay	8.599 sec	8.599 sec	8.449 sec	7.224 sec	7.213 sec
Maximum Pre-Roll Delay	29.536 sec	29.536 sec	29.492 sec	7.834 sec	7.482 sec

Table 7: Performance of the Digiturk Application

Scheduler	Average Total Downloaded Data	
	User 1	User 2
Max C/I	794.72 Mbits	212.13 Mbits
PF	755.51 Mbits	222.64 Mbits
EXP_Rule	755.08 Mbits	223.90 Mbits
Proposed	733.24 Mbits	219.77 Mbits
Single App.	880.86 Mbits	238.13 Mbits

Table 8: Performance of the File Download Application

Scheduler	Average Webpage Download Time	
	User 1	User 2
Max C/I	21.115 sec	6.815 sec
PF	20.979 sec	6.767 sec
EXP_Rule	21.712 sec	5.286 sec
Proposed	4.609 sec	4.038 sec
Single App.	3.673 sec	3.672 sec

Table 9: Performance of the Web Browsing Application

Scheduler	Average Total System Throughput
MAX C/I	1.3469 Gbits
PF	1.3312 Gbits
EXP-Rule	1.3316 Gbits
Proposed	1.3226 Gbits
Single App.	1.4949 Gbits

Table 10: Overall Network Throughputs

5.0.4 Application-Based Scheduling Demo

Simulation results and heuristics prove that a new scheduling approach with application-based queueing is essential for indoor femtocell networks. In this work, we also implemented the concept of application-based scheduling in a HNB and build our testbed by setting up the HNB into a home network in order to observe the performance gain for our proposed scheduler. As expected, results are very similar to our simulation results and the benefits of application-based scheduling are tremendous compared to the standard traditional based PF scheduler.

Femtocell networks are not yet operational in Turkey, therefore we used an emulator for mobile operator's core network and HNB is connected to the internet via the simulated 3G-HSDPA core network of the mobile operator. A computer is connected to the HNB via a 3G USB dongle and the testbed is presented at Fig. 17. One file download session and one live video streaming session (Tivibu) is demanded simultaneously in order to observe the effects of file download session on a video streaming application. For both cases (PF scheduling and application-based scheduling), these two applications are demanded for 600 seconds and total downloaded file sizes and pause number and durations for video streaming application are recorded. For traditional PF scheduling case, we observed 11 pauses for which the minimum and maximum pause durations were 14 seconds and 40.7 seconds, respectively. 8 out of 11 pauses lasted more than 30 seconds which in total, corresponds to 328 seconds of pause in a 600 seconds of video streaming session. Total downloaded file size on the other hand, was 180,93 MB for the traditional case. For the application-based scheduling case, we did not encounter any pause for the entire duration of the live video stream whereas the total downloaded file size was 164,23 MB which corresponds a performance loss of around 9% for the file download application.

Figure 17: Application-Based Demo Testbed

CHAPTER VI

CONCLUSION

Compared to its outdoor counterpart, an indoor cellular data user is much more likely to have multiple applications open, each potentially requiring different QoS constraints. The current outside-in approach in todays cellular systems offers the same solution to indoor users as it does to outdoor users, which is a user-based scheduler. Here, understandably, the goal is to equitably and fairly divide the system resources amongst users in such a way that system throughput is as high as possible. Spectrum being such a scarce and expensive resource, the outdoor cellular system simply tries to maximize its efficient use while providing service over a large geography where users with vastly different channel qualities are fighting for service.

However, the scenario is dramatically different for an indoor user. When a HNB is deployed indoors, its coverage area is much smaller in radius (usually measured in meters) than a outdoor cell. Therefore, the difference in the relative channel qualities of the users is not as high, lowering the significance of the notion fairness. Additionally, the typical number of users connected to a femtocell base station at a given time is likely to be much lower than those connected to a MNB. In fact, most indoor femtocells will likely be closed access, limiting use to only members of a given household. Aggressive multiuser diversity for spectrum efficiency is not really as fundamental in this scenario. Last, but not least, these fewer users are much more likely to demand multiple, parallel data services at a given time. In this scenario, this paper shows that a different resource allocation scheduling needs to be developed for a femtocell base station.

We show that the indoor femtocell benefits tremendously from application-based scheduling. This is mainly due to the likelihood of multiple active applications per user in the indoor scenario. If one were to insist on user-based scheduling, we show that the background class data traffic simply floods the user queues leaving one with an impossible task of satisfying the QoS requirements of other classes of traffic whose data is stored in the same queue.

We also propose an application-based QoS-aware scheduler in this thesis and demonstrate the performance of application-based scheduling concept on a femtocell network. The scheduler treats each application differently and application priorities are adapted to the arrival rate of their corresponding traffic to the femtocell queues. The QoS requirements of the interactive, conversational and streaming class of services are handled via an exponential term in the scheduler which increases or decreases the application priority in getting scheduled based on whether acceptable application behavior can be sustained. This scheduler performs very well and results in QoS performances for each application almost identical to cases where the entire femtocell resource pool is reserved for that application for the entire session. This comes at the expense of a modest performance loss of around 8% for the background class traffic.

Bibliography

[1] V. Chandrasekhar, J. Andrews, and A. Gatherer, "Femtocell networks: a survey," *Communications Magazine, IEEE*, vol. 46, pp. 59 –67, September 2008.

[2] picoChip Designs Ltd., "The case for home base stations," technical white paper, Sep 2008.

[3] S. R. S. et al, *FEMTOCELLS Opportunities And Challenges for Business And Technology.* John Wiley & Sons, 2009.

[4] J. Weitzen and T. Grosch, "Comparing coverage quality for femtocell and macrocell broadband data services," *Communications Magazine, IEEE*, vol. 48, pp. 40 –44, january 2010.

[5] S. Murugesan, P. Schniter, and N. Shroff, "Multiuser Scheduling in a Markov-Modeled Downlink Using Randomly Delayed ARQ Feedback," *Information Theory, IEEE Transactions on*, vol. 58, pp. 1025 –1042, Feb. 2012.

[6] D. Kitazawa, L. Chen, H. Kayama, and N. Umeda, "Downlink packet-scheduling considering transmission power and qos in cdma packet cellular systems," in *Mobile and Wireless Communications Network, 2002. 4th International Workshop on*, pp. 183 – 187, 2002.

[7] J.-L. Chen and H.-C. Chao, "Fair resource sharing scheduling for cellular data services with qos provisioning," *Communications, IEE Proceedings-*, vol. 148, pp. 144 –148, jun 2001.

[8] E. Chaponniere, P. Black, J. Holtzman, and D. Tse, "Transmitter directed code division multiple access system using path diversity to equitably maximize throughput," September 2002.

[9] 3GPP, "Physical Layer Aspects of UTRA High Speed Downlink Packet Access," TR 25.848, 3rd Generation Partnership Project (3GPP), Apr. 2001.

[10] Kleinrock, L., *Queueing Systems. Volume 1: Theory.* John Wiley & Sons, 1975.

[11] E. Kartsakli, J. Alonso-Zárate, L. Alonso, and C. Verikoukis, "Cross-layer scheduling with qos support over a distributed queuing mac for wireless lans," *Mob. Netw. Appl.*, vol. 14, pp. 709–724, Dec. 2009.

[12] S. Shakkottai and A. L. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: The exponential rule," *Analytic Methods in Applied Probability, American Mathematical Society Translations*, vol. 207, pp. 185–202, 2002.

[13] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *Communications Magazine, IEEE*, vol. 39, pp. 150 –154, February 2001.

[14] R. K. Beidokhti, M. H. Yaghmaee Moghaddam, and J. Chitizadeh, "Adaptive QoS scheduling in wireless cellular networks," *Wirel. Netw.*, vol. 17, pp. 701–716, Apr. 2011.

[15] T. Ozcelebi, M. Oguz Sunay, A. Murat Tekalp, and M. Reha Civanlar, "Cross-layer optimized rate adaptation and scheduling for multiple-user wireless video streaming," *Selected Areas in Communications, IEEE Journal on*, vol. 25, pp. 760 –769, May 2007.

[16] X. Liu, E. K. P. Chong, and N. B. Shroff, "A Framework for Opportunistic Scheduling in Wireless Networks," *Computer Networks*, vol. 41, pp. 451–474, 2003.

[17] 3GPP, "High Speed Downlink Packet Access (HSDPA); Overall description; Stage 2," TS 25.308, 3rd Generation Partnership Project (3GPP), Sept. 2008.

[18] 3GPP, "UE Radio Access Capabilities," TR 25.306 v10.7.0, 3rd Generation Partnership Project (3GPP), July 2012.

[19] K. Freudenthaler, A. Springer, and J. Wehinger, "Novel SINR-to-CQI Mapping Maximizing the Throughput in HSDPA," in *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pp. 2231 –2235, march 2007.

[20] M. Wrulich, W. Weiler, and M. Rupp, "HSDPA Performance in a Mixed Traffic Network," in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pp. 2056 –2060, may 2008.

[21] 3GPP, "Physical layer procedures (FDD)," TS 25.214, 3rd Generation Partnership Project (3GPP), June 2011.

[22] R. Racic, D. Ma, H. Chen, and X. Liu, "Exploiting and Defending Opportunistic Scheduling in Cellular Data Networks," *Mobile Computing, IEEE Transactions on*, vol. 9, pp. 609 –620, may 2010.

[23] 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 Vol.3, oct. 2003.

[24] J. H. Rhee, T. H. Kim, and D. K. Kim, "A wireless fair scheduling algorithm for 1 times;ev-do system," in *Vehicular Technology Conference, 2001. VTC 2001 Fall. IEEE VTS 54th*, vol. 2, pp. 743 –746 vol.2, 2001.

[25] B. Al-Manthari, H. Hassanein, and N. Nasser, "Packet scheduling in 3.5g high-speed downlink packet access networks: breadth and depth," *Network, IEEE*, vol. 21, pp. 41 –46, jan.-feb. 2007.

[26] L.-C. Wang and Ming-Chi, "Comparisons of Link-Adaptation-based scheduling algorithms for the WCDMA system with high-speed downlink packet access," *Electrical and Computer Engineering, Canadian Journal of*, vol. 29, pp. 109 – 116, January/April 2004.

[27] 3GPP, "Quality of Service (QoS) concept and architecture," TS 23.107, 3rd Generation Partnership Project (3GPP), Sept. 2007.

[28] Stüber, G. L., *Principles of Mobile Communication.* Kluwer Acedemic Publishers, 2001.

[29] 3GPP, "FDD Base Station (BS) classification," TR 25.951, 3rd Generation Partnership Project (3GPP), June 2007.

[30] M. Gudmundson, "Correlation model for shadow fading in mobile radio systems," *Electronics Letters*, vol. 27, pp. 2145 –2146, nov. 1991.

[31] D. Mungan, G. Ferazoğlu, O. Sunay, R. Civanlar, and B. Kaytaz, "Advantages of application-based scheduling for indoor femtocells," in *Proceedings of the 6th ACM workshop on Wireless multimedia networking and computing*, WMuNeP '11, (New York, NY, USA), pp. 55–62, ACM, 2011.

[32] C. Atici and M. O. Sunay, "High data-rate video broadcasting over 3g wireless systems," *Broadcasting, IEEE Transactions on*, vol. 53, pp. 212 –223, march 2007.

[33] H. Ren, W. Zhou, N. Ken'ichi, W. Gao, and Q. Wu, "Multi-objective optimization for the operation of distributed energy systems considering economic and environmental aspects," *Applied Energy*, vol. 87, no. 12, pp. 3642–3651, 2010.

[34] B. Han, X. Zhang, Y. Qi, Y. Gao, Y. Liu, and D. Yang, "A surrounding-adaptive qoe adjustment approach for optimization of capacity and quality of air interface," in *Telecommunications (ICT), 2012 19th International Conference on*, pp. 1 –5, april 2012.

[35] C. S. Inc., "The zettabyte era," technical white paper, May 2012.

VITA

Devin Mungan was born in Eskişehir, Turkey on September 11 1987. He graduated from Bursa Science High School in 2004. In 2009, he received his B.Sc degree of Electrical & Electronics Engineering from Koç University. He joined Özyeğin University in September 2010 where he is currently a M.Sc student. He was a part of the Femtocell Project team funded by Argela and his advisor is Assoc. Prof. M. Oğuz Sunay. At July 2012, he joined AirTies Wireless Team as a software engineer.