Uplink Scheduling for Delay Sensitive Traffic in Broadband Wireless Networks

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCES

OF BİLKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Cemil Can Coşkun

July 2012

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Ezhan Karaşan(Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Nail Akar

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Kağan Gökbayrak

Approved for the Graduate School of Engineering and Sciences:

Prof. Dr. Levent Onural
Director of Graduate School of Engineering and Sciences

ABSTRACT

Uplink Scheduling for Delay Sensitive Traffic in Broadband Wireless Networks

Cemil Can Coşkun
M.S. in Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Ezhan Karaşan
July 2012

In wireless networks, there are two main scheduling problems: uplink (mobile station to base station) and downlink (base station to mobile station). During the downlink scheduling, scheduler at the base station (BS) has access to queue information of mobile stations (MS). On the other hand, for uplink scheduling, BS only has the partial information of the MS since distributing the detailed queue information from all MSs to BS creates significant overhead.

In this thesis, we propose a novel uplink scheduling algorithm for delay sensitive traffic in broadband wireless networks. In this proposed algorithm, we extend the bandwidth request/grant mechanism defined in IEEE 802.16 standard and send two bandwidth requests instead of one: one greedy and the other conservative requests. MSs dynamically update these bandwidth requests based on their queue length and bandwidth assignment in previous frames. The scheduler at the BS tries to allocate these bandwidth requests such that the system achieves a high goodput (defined as the rate of error-free packets delivered within a maximum allowed delay threshold) and bandwidth is allocated in a fair manner, both in short term and in steady state. The proposed scheduling algorithm

can utilize the network resources higher than 95% of the downlink scheduling algorithms that use the complete queue state information at the MS. Using just partial queue state information, the proposed scheduling algorithm can achieve more than 95% of the total goodput achieved by downlink scheduling algorithms utilizing whole state information. The proposed algorithm also outperforms several downlink scheduling algorithm in terms of short-term fairness.

Keywords: Wireless Networks, Delay sensitive traffics, Uplink Scheduling, Goodput

ÖZET

GENIŞ BANTLI KABLOSUZ AĞLARDA GECIKMEYE HASSAS UYGULAMALAR IÇIN YER-UYDU BAĞI ZAMANLAMALARI

Cemil Can Coşkun
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Doç. Dr. Ezhan Karaşan
Temmuz 2012

Kablosuz ağlarda iki çeşit zamanlama problemi vardır: Yer-uydu bağı (hareketli kullanıcıdan baz istasyonuna) ve uydu-yer bağı (baz istasyonundan hareketli kullanıcıya). Uydu-yer bağı zamanlamalarında, baz istasyonundaki zamanlayıcı hareketli kullanıcıların bütün küyruk bilgilerine erişebilir. Fakat, uydu-yer bağı zamanlamalarında, bütün hareketli kullanıcıların detaylı kuyruk bilgisini baz istasyonuna taşımak büyük bir ek yük yarattığından baz istasyonunda hareketli kullanıcılarla ilgili sadece kısmi bir bilgi vardır.

Bu tezde, geniş bantlı kablosuz ağlarda gecikmeye hassas uygulamalar için yenilikçi bir yer-uydu bağı algoritması önerdik. Önerilen algoritmada, IEEE 802.16'nın belirlediği standart talep/alan ayırma mekanizmasını genişleterek bir yerine iki tane bant genişliği talebi gönderdik: Bir tanesi istekli bir tanesi de tamahkar olmak üzere. Hareketli kullanıcılar kuyruk uzunlukları ve daha önce kendilerine ayrılan bant genişliğine göre dinamik bir şekilde bant genişliği taleplerini güncellediler. Baz istasyonundaki zamanlayıcı, bu bant genişliği taleplerine göre sistem başarılı çıktısını (hatasız olarak ve izin verilen en yüksek

gecikme sınırını aşmadan iletilmiş paketler olarak tanımlanmıştır) yüksek tutmaya çalışarak ve kısa ve uzun vadedeki eşit paylaşımı bozmadan kullanıcılara yerleri ayırır. Öneriler algoritma ağ kaynaklarını bütün kullanıcıların kuyruk bilgilerine hakim olan uydu-yer bağlantısı zamanlayıcı algoritmalarının elde ettiği faydanın %95i kadar daha fazla fayda getirecek şekilde kullanmıştır. Önerilen algoritma, kullanıcıların kuyruk bilgilerinin sadece bir kısmını kullanarak, bütün durum bilgisine sahip olan uydu-yer bağlantısı zamanlayıcı algoritmalarının elde ettiği başarılı sistem çıktısının %95inden daha fazla bir faydaya ulaşmıştır. Önerilen algoritma bir çok uydu-yer yönlü zamanlama algoritmalarının kısa vadede elde ettiği eşit paylaşımdan daha iyi bir kısa vade eşit paylaşımı elde etmiştir.

AnahtarKelimeler: Kablosuz ağlar, Gecikmeye hassas uygulamalar, Yer-uydu bağı zamanlayıcıları, Başarılı çıktı

ACKNOWLEDGMENTS

I would like to express my special thanks to my supervisor Assoc. Prof. Ezhan Karaşan whose guidance helped me in every step of the preparation of this thesis.

I also thank to Assoc. Prof. Nail Akar and Asst. Prof. Kağan Gökbayrak their valuable contributions to my thesis defense committee.

I want to thank to TUBITAK as well for supporting me financially through my MS degree program.

Finally, for their valuable supports in every step of my life, I am grateful to my mother, my father and my friends.

Contents

List of Figures

List of Tables

To Infinity and Beyond ...

Chapter 1

Introduction

Telecommunication investments have been rapidly increasing in recent years. It is estimated that the world-wide telecommunication industry revenues will grow from $2.1 trillion in 2012 to $2.7 trillion in 2017 [1]. While telecommunication industry continues to grow, wireless technologies are the main reason of this growth. A major part of this investment is on wireless technologies because wireless technologies allow user mobility. Owing to mobility provided by wireless technologies, people can access to information or communicate with other people anywhere without requiring wires. Given the importance placed on these services in today's culture, providing high-speed wireless services with a large coverage area has become very busy research area.

There are several research challenges in wireless networks. One such problem is the efficient scheduling of users' demand for bandwidth so that network resources are shared efficiently and user demand requirements are satisfied within an acceptable delay. There are two main scheduling problems in wireless networks, namely uplink (mobile station to base station) and downlink (base station to mobile station). For downlink transmissions, since all data is gathered at a

single point (base station (BS)) developing a centralized scheduling algorithm is relatively simple. On the other hand, in centralized uplink scheduling performed at the BS only part of the required information is available at the scheduler since traffic queues are held at mobile stations (MSs). Therefore additional information is needed from users. Since there is a trade-off between amount of information provided to BS (extra overhead) and resource usage efficiency, uplink scheduling is a more open area for research. Most of the existing researches on wireless scheduling have been concentrated on downlink scheduling whereas uplink scheduling has been explored less.

There are numerous different applications on the Internet tailored for different purposes. Each application has its own requirements in the network. Some applications require error-free data communication. Some of them need guaranteed bit rate. For some others, latency is the most important requirement. For applications which have maximum latency requirement, the delay experienced by a packet is what determines utility of the packet. Late received packets with a delay exceeding a certain threshold (generally taken as 200 ms for interactive applications) are useless for this type of applications. In order to measure the rate of data transfer from MSs to BS for evaluating the performances of uplink scheduling algorithms subject to maximum delay requirements, we use goodput which is the rate of bits that are delivered to the destination within a certain delay. Scheduling for delay sensitive traffics which requires not only timely delivery of packets, but also efficient use of network resources presents itself as a challenging problem.

In this thesis, a novel uplink scheduling algorithm for delay sensitive traffic for broadband wireless networks has been proposed. The main objectives of this algorithm are to allocate bandwidth with an efficiency close to downlink

scheduling algorithms, to improve total goodput of the system for delay sensitive applications (i.e. rate of packets delivered within a pre-determined maximum delay threshold) and at the same time have short and long term fairness among users.

Our scheduling algorithm is designed for delay sensitive traffics. For these type of applications, the delay experienced by a packet is not critical, as long as it is less than a certain threshold. Therefore, in our proposed scheduling algorithm, we assign higher priority to MSs that currently have an average delay exceeding a certain threshold. When the buffer length of the MS drops below the threshold, its high priority status ends.

In wireless networks, mobile users have different transmission rates because of varying channel conditions and they cannot send equal amount of data for equal air resources. When air resources are equally shared among users, it is called airtime fairness. When users send equal amount of data within a time period, it is called data-rate fairness. In our proposed scheduling algorithm, we try to share air resources equally among users. For that purpose, at the beginning, equal number of tokens is placed in each user's bucket. At the end of the each frame, tokens corresponding to user's bandwidth allocation are reduced from the user's bucket. At the beginning of each frame, constant number of tokens is added each user's bucket. While granting, users are sorted in descending order according to number of tokens if users do not have special conditions. Therefore, users who use more resources at previous frames are less likely to be granted more resources, e.g., conservative request is granted instead of greedy request.

In the literature, there are several centralized uplink scheduling algorithms that have been proposed [2–11]. [2–4] are concentrated on only real-time traffics.

On the other hand, [5–11] are proposed for multiclass traffics so other types of traffics (i.e. Unsolicited Grant Services (UGS), Best Effort (BE)) are also taken into account. In [2] and [4], it is assumed that queue information is available at BS for granting bandwidth requests. On the other hand, the proposed scheduling algorithm in the thesis uses partial information thus significantly reducing the extra overhead required for uplink scheduling. [3] allocates constant grant size to users when they are on state. The packet size is assumed to be constant in [2], therefore BS can obtain all necessary queue information by using the total queue size of each MS. On the contrary, packet size are varying in our scheduling algorithm, therefore uplink scheduling problem becomes more challenging.

In our simulations, we compare our proposed scheduling algorithm with downlink scheduling algoritmhs: Fully opportunistic downlink scheduling algorithm, max-min fairness downlink scheduling algorithm, round robin downlink scheduling algorithm, proportional fair scheduling algorithm and some different versions of our proposed scheduling algorithm. Although, we have significant disadvantages against these algorithms, our proposed scheduling algorithm can reach 97% of the total goodput of the algorithm which maximizes the total goodput of the system (i.e. fully opportunistic scheduling algorithm and Max-min fairness scheduling algorithm) On the other hand, we can reach 101% of total goodput of the round robin scheduling algorithm where users are active 40% of the time on the average. We can reach 103.5 percent of total goodput of round robin scheduling algorithms where users are active 80% of the time on average. Our proposed scheduling algorithm can utilize the resources for high load traffics with 95% efficiency of the round robin scheduling algorithm for a network where users are active 40% of time on the average and with 98 % efficiency of the round robin scheduling algorithm in networks where users are active 80 % of time. In low traffic loads, our bandwidth utilization is over the 99% for any downlink scheduling algorithm. The proposed method also performs comparably well with

the downlink algorithms in terms of short term fairness.

In our simulations, we compare our proposed scheduling algorithm with algorithms which send only one bandwidth request: greedy and conservative. Our proposed scheduling algorithm can reach 107% of the total goodput of the greedy algorithm and 200% of the total goodput of the conservative algorithm. Our proposed scheduling algorithm can utilize the resources for high traffic loads with 152% of conservative algorithm and 113% of greedy algorithm. The proposed method performs remarkably better than these two algorithms in terms of short term fairness.

Another important point to make: In IEEE 802.16 standard MSs make a single bandwidth request from the BS for the uplink scheduling channel. The scheduler either grants this bandwidth request or denies it. The standard also does not specify how the bandwidth request is determined by each MS. We extend the methodology defined in the standard in two different directions: First, instead of making a single bandwidth request, the mobile station makes two bandwidth requests, first one more conservative and the other more greedy. In the scheduling of user packets, it is possible to split some packets into multiple frames, called fragmentation, so that resources can be used more efficiently. Fragmentation incurs extra overhead due to required fragmentation headers are necessary for assembling the original packet. Fragmentation is not allowed in our scheduling algorithm to reduce overhead. Therefore, bandwidth requests must be delimited to contain unsplit packets. The scheduler either grants the greedy bandwidth request, or grants the conservative bandwidth request, or denies the requests. Second, we introduce an algorithm for MS in order to determine these bandwidth requests based on information on bandwidth assignments to all MSs

in previous frames and their own queue lengths.

The rest of the thesis is organized as follows. In Chapter 2, some useful background information about wireless communication is provided. Moreover, the basics of scheduling in wireless networks are explained. Furthermore, several required metrics to quantify performance of uplink scheduling algorithms are discussed. Existing wireless scheduling algorithms in the literature are also discussed.

In Chapter 3, the proposed uplink scheduling algorithm is presented. First, the bandwidth request part is explained. Second, granting part is explained. We also provide 3 different scenarios to explain the details of the proposed uplink scheduling algorithm: In the first scenario, network is uncongested and users can get their greedy requests. In the second scenario, there is a more congested network; some users get their conservative requests. In the third one, network is highly congested. Some users encounter packet losses and increased latency.

Chapter 4 describes the simulation environment and the parameters that are used for simulations such as burst profiles, path losses, etc. The results obtained from these simulation settings are presented and discussed.

Chapter 5 concludes the thesis.

Chapter 2

Background Information

Wireless telecommunication is basically data transmission between two or more devices without wires [12]. Wireless systems use radio signal frequencies for communication. Several devices use wireless technology to communicate such as cellular telephones, radio telegraphs, laptop computers, two-way radios etc. Wireless systems are preferred by users because of their flexibility. They do not enforce people to stay at a fixed position. Thanks to wireless communication, people can be mobile during the communication. Wireless communications are getting more popular as the number of companies which provide wireless communication services increases rapidly [13].

Wired communication is used to describe a type of communication which uses wires and cables to transmit data. Traditional home telephones and LAN communication are the most common examples of wired communications. Most of the wired networks use fiber-optic cables which provide clear signaling for transmission. Network with fiber optic cables ensure more signals than copper wiring systems. In addition, signals still can go over long distances reliably [14]. Wired communications are the most stable communication type since they

are affected less from weather conditions and environment compared to wireless networks.

Wireless networks are a satisfying alternative to wired networks. The main difference between wired and wireless networks is the physical cable. Wired networks communicate through wires which is a more stable technology than radio signals. Signal strength of a wireless connection may fluctuate because of changes in the wireless propagation medium. In addition, wired networks typically have higher transmission speeds than wireless ones. Furthermore, sending data on air causes security problems [15]. If transmitted data is not encrypted, anyone who receives the signal can easily access the data. Moreover, some problems of wireless networks do not exist in wired ones such as time varying channel capacity and location dependent errors. Additionally, interference from other users can decrease channel capacity for wireless networks as users share a common wireless transmission medium.

There are two main types of wireless communications: infrastructure based and ad hoc networks which are depicted in Figure 2.1 [13]. In infrastructure based wireless systems, clients can communicate with each other via fixed base station or access points. Multiple wireless access points are needed in order to cover an area. Access points assist data transfer among users. On the other hand, ad hoc networks are decentralized [16]. There is no infrastructure, each node participates in communication actively. Although ad hoc networks do not require any fixed infrastructure, their performances are typically worse than infrastructured based networks since mobility of users may result in deterioration in wireless transmission such as unavailability of routes.

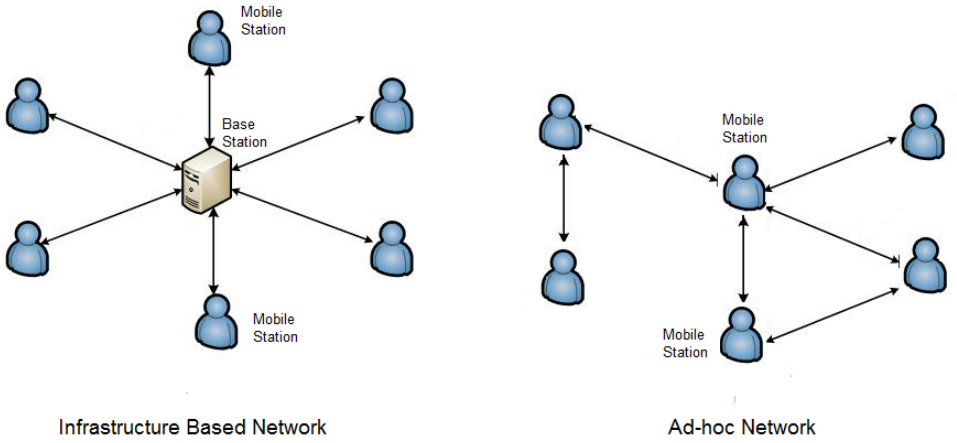

Figure 2.1: Infrastructure Based Networks vs. Ad-hoc Networks

In this thesis, infrastructure based networks are examined. In infrastructure based networks, there must be a base station (BS) which organizes all communications among users. All packets are gathered at BS before they are sent to their destination. One of the functions of the BS is to choose the packets to be transmitted (received) to (from) the mobile stations. This problem is called the packet scheduling or simply the scheduling problem. There are two type of scheduling in infrastructure based networks which are uplink and downlink. During the downlink scheduling, BS determines the data packets to be transmitted to mobile stations (MS). On the other hand, during the uplink scheduling BS chooses the data packets that it will receive from MSs and then inform MSs about the selected schedule. In this thesis, we study the uplink scheduling problem.

In this chapter, some useful background information is discussed. Firstly, brief information is provided about adaptive modulation and coding in Section 2.1. In Section 2.2, throughput and goodput concepts are introduced. In Section 2.3, different fairness metrics are discussed. Scheduling in wireless network will

be reviewed in section 2.4. Finally, in Section 2.5, fragmentation is investigated.

2.1 Adaptive Modulation and Coding

In mobile communication systems, the quality of signal received by BS varies because of the distance between MS and BS, log-normal shadowing, fading, noise and interference from other MSs etc. If the quality of signal is low, error probability increases. In order to improve the system capacity, decrease packet losses and enlarge coverage area, the transmitted signal can be modified which is called link adaptation [17]. Adaptive Modulation and Coding(AMC) is a link adaptation method and it aims to raise the system capacity [17]. By using AMC, each user can adapt its modulation and coding scheme individually. AMC does not change power of the signal, instead it adjusts the modulation and coding format with respect to the signal quality and channel conditions [18]. Users close to the BS usually use larger modulation constellations and higher code rates. On the other hand, users close to the cell boundary use smaller modulation constellations and lower code rates. In addition, the scheduler may consider application's delay and error sensitivity in choosing appropriate coding and modulation scheme. If channel changes very fast, AMC works poorly because choosing appropriate modulation technique and coding scheme based on previous observations of the channel is difficult [13].

2.2 Goodput and Throughput

In communication networks, throughput is basically average rate of successfully delivered packets and it is usually measured in bits per second. Total throughput

of a system is the sum of all data communications in the system. While calculating total throughput of a system, all packets are take into account. Although some received packets are useless, they are counted in total throughput of the system.

Each application may have different requirements such as minimum guaranteed bit rate, bit error rate, maximum latency etc. If an application has maximum latency requirement comparing throughput of users may be misleading. Therefore, another parameter is needed for such systems.

Since error-free received packets may be useless from the application point of view if it arrives after the allowed delay. For delay sensitive applications such as Voice-over-IP (VOIP), arrival time of a packet is extremely important. If a packet arrives to the destination after deadline, that packet becomes useless. Therefore, throughput is not sufficient index to measure total success of system. Thus, goodput is introduced which is the throughput of the useful bits, that are delivered to the destination within a certain delay. While calculating goodput, retransmitted packets or excessively delayed packets are ignored.

Table 2.1 gives information about which packets are counted and which are not while calculating throughput and goodput. Difference between throughput and goodput is that late received packets and packets with errors that require retransmission are not taken into account while calculating goodput because such packets are useless for real time applications.

Table 2.1: Type of packets taken into account for throughput and goodput calculations

	Throughput	Goodput
Packets received on time	√	√
Late received packets	√	X
Packets with uncorrectable errors	√	X
Lost packets	X	X

2.3 Fairness

In wireless networks, users have time varying channel characteristics due to shadowing and path losses. Because of these variations, users have different data transmission rates and they cannot send equal amount of data within a period of time even when they are assigned the same amount of airtime. This leads to the throughput-fairness dilemma. If bandwidth is shared equally among users, it may decrease total throughput of the system while distributing the resources equally to all users. Because of transmission rate differences, users can send altered amount of data for equal bandwidth allocation. So there are two types of fairness ideas: User can send equal amount of data or users are allocated equal amount of time in air (called "airtime fairness"). First one punishes users who have high data rate and assigns them small amount of bandwidth area because of low data rated users. This punishment also decreases total throughput of the system. In the second one, total transmitted data of users varies according to their distances from the BS. Users who have high data rates, can send more bytes within the same duration. If user's mobility is low and their data rates rarely change, users who have low data rates can send less amount of data even in the long term.

In order to quantify fairness of networks, different fairness metrics are defined :

$$F_R(T) = \Big(\sum_{m=1}^{N} R_m(T) \Big)^2 / \Big(N \cdot \sum_{m=1}^{N} R_m(T)^2 \Big) \tag{2.1}$$

In Equation (2.1), $R_m(T)$ refers to total amount of data sent in time interval T by user m and N denotes the total number of users in the system. $F_R(T)$ measures the fairness of the data rate distribution among the users within a duration T [19]. When $F_R(T) = 1$, all users received the same average data rate within the period. As $F_R(T)$ gets closer to 0, it means that the distribution of data rates obtained by users is more unbalanced.

$$F_A(T) = \Big(\sum_{m=1}^{N} A_m(T) \Big)^2 / \Big(N \cdot \sum_{m=1}^{N} A_m(T)^2 \Big) \tag{2.2}$$

In Equation (2.2) $A_m(T)$ refers to total amount of assigned area in air in time interval T by user m. $F_A(T)$ measures the fairness of the amount of allocated resources within a duration T [19]. When $F_A(T) = 1$, all users received the same amount of time in air within the period. As $F_A(T)$ gets closer to 0, it means that the distribution of allocated resources to users are more unbalanced.

Fairness can be measured over long time periods (called "long-term fairness") and over short time durations (called "short-term fairness"). A system can be considered as long-term fair, if total assigned bandwidth is proportionally similar to total generated traffic. For long-term fairness, relatively large T's are used to calculate the fairness index. On the other hand, relatively small T's are used

to calculate the short-term fairness index. For example, if simulation period is divided into K pieces and each user gets whole bandwidth during one interval, system becomes long term fair but it is extremely unfair for short intervals. If a system is short-term fair, it must also be a long term fair [20].

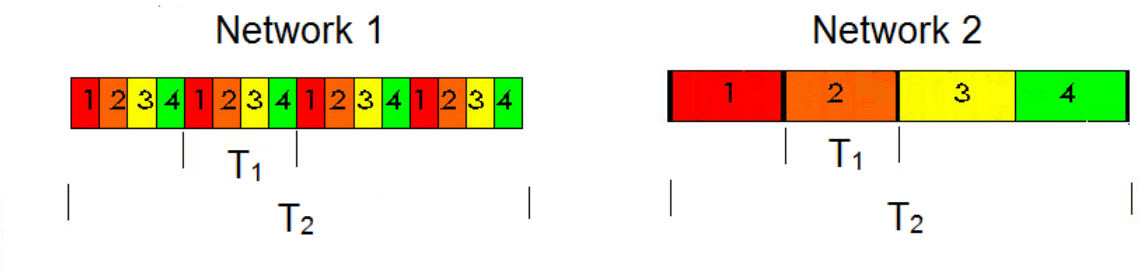

Figure 2.2: Short Term Fairness vs. Long Term Fairness

In Figure 2.2, there are two different networks with 4 users. First network divides time into smaller pieces and assigns each piece to one user only. On the other hand, second network divides time to larger pieces and assigns each piece to one user. Network 1 is both short term and long term fair, on the contrary, network 2 is only long term fair. If any new packet comes to user 1 during an interval assigned for user 2, this packet must wait for a long period until users 2-4 complete their transmissions.

Short-term fairness is an important parameter for applications which need low latency, e.g. VOIP, online gaming etc [21]. If a system is short term fair, users do not have any advantages on each other and they can access to channel with the same probability over a short time horizon, which leads to short access delays for packets. For delay sensitive applications, equal division of goodput is more important than the division of throughput. Determining fairness by using throughput may mislead because there are some useless packets in the calculation of throughput. Therefore, while calculating fairness, fair sharing of the goodput

must be the prior aim.

2.4 Scheduling in Wireless Networks

In wireless networks, each user has different channel qualities and traffic loads. Allocating bandwidth to users in the network in each frame is basically known as the scheduling problem. Scheduling is a fundamental part of wireless environments because (i) all resources are shared between users, (ii) users may interfere with each other if they transmit concurrently [22].

In infrastructured based wireless networks, users communicate with the BS. There are two main scheduling problems which are uplink (mobile station to base station) and downlink (base station to mobile station). During downlink, data is sent by a single source. Therefore there is no complicated process for power allocation and adjusting transmission delays. However, in uplink, each user sends its own data to a single source, therefore transmission timetable must be set carefully, as otherwise there can be interference and data losses.

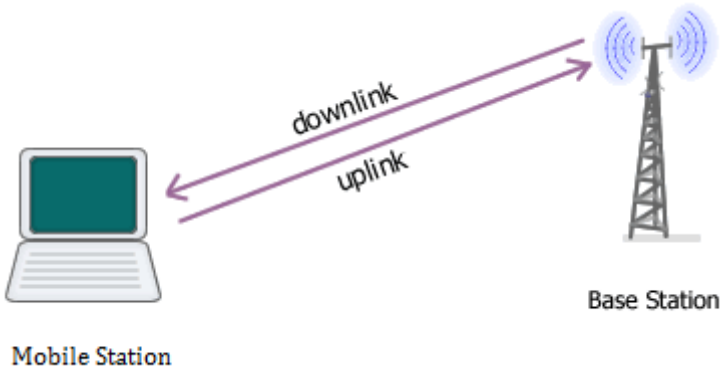

Figure 2.3: Downlink and Uplink Scheduling

The scheduling mechanism can be centralized or distributed. In centralized scheduling algorithms, scheduler at BS makes all decisions about bandwidth allocation. For downlink scheduling, while scheduler knows whole information about the network, using a centralized scheduling algorithm is a better choice. On the other hand, for uplink scheduling, BS may not have the complete state information of MSs. In order to implement centralized uplink scheduling some information about the states of MSs must be transferred to the BS.

During the centralized downlink scheduling, scheduler at BS has access to information such as packet arrival rate of MS, number of packets in each MS queue, success statistics of packets transmission of each MS, size of each packets in each MS queue etc. The basic operation of centralized downlink scheduling is depicted in Figure 2.4. Therefore, setting up a scheduling algorithm for a downlink scheduling becomes easier. Research on downlink scheduling is relatively straight forward than in uplink scheduling [23] because there are less uncertainties compared to uplink scheduling.

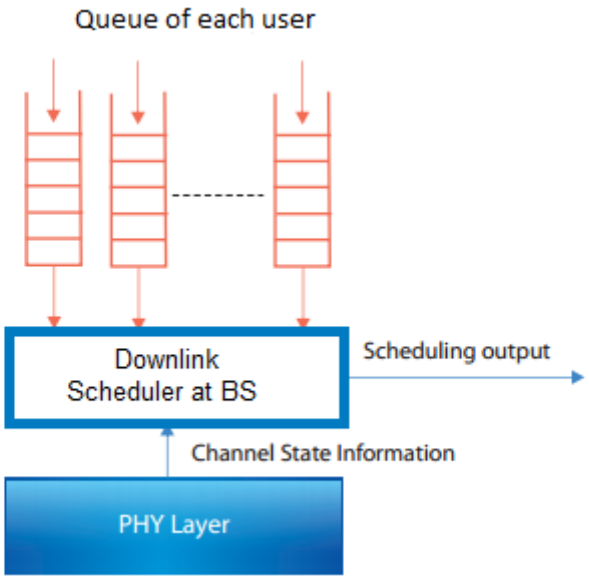

Figure 2.4: Centralized Downlink Scheduling

Centralized uplink scheduling can be designed similar to centralized downlink scheduling. However in order to do this, each MS must send its complete information to BS. Since distributing detailed information to the BS creates significant messaging overhead for uplink scheduling, in practice BS may only have partial information about MSs. While user's sending all the information is not a realistic approach, BS should make smart choices based on the limited information provided by MSs and collected by itself from earlier transmissions. In addition, MSs have to make careful choices about how much information will be sent because there is a sharp threshold: sending less information may mislead BS, conversely sending too much information causes waste of bandwidth. Therefore, MSs and BS must work collaboratively to have a good performing centralized uplink scheduling algorithm. The basic operation of centralized uplink scheduling is shown in Figure 2.5.

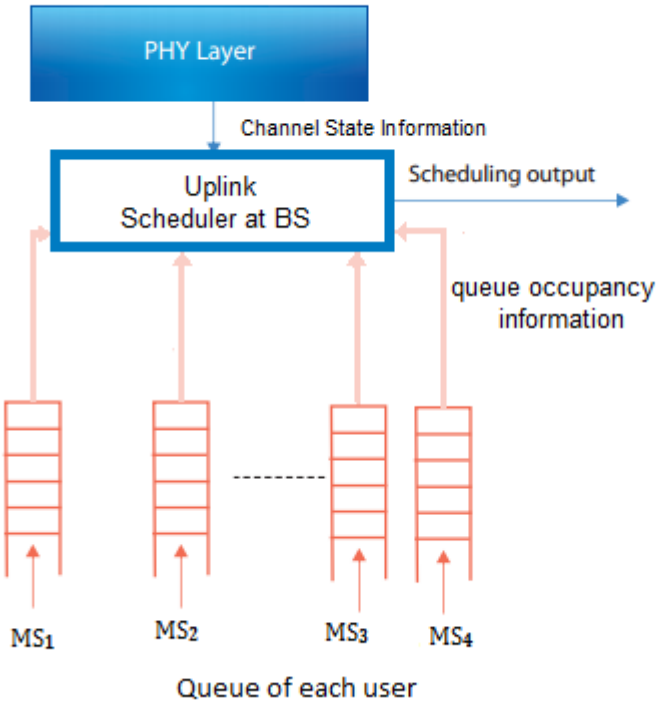

Figure 2.5: Centralized Uplink Scheduling

Because of these uncertainties, uplink scheduling is a more open area for research. In this thesis, we study centralized uplink scheduling. All users send their bandwidth requests to the BS and BS makes the decision about bandwidth allocation. MSs make some assumptions by using previous bandwidth allocations for bandwidth requests. MSs and BS work collaboratively to determine on bandwidth allocation.

Centralized uplink and downlink scheduling algorithms have been widely investigated in the literature [2–11, 24]. There are many different parameters that are optimized such as utility maximization, sum-rate maximization, achieving a desired quality of services (QoS), power efficiency and ergodic sum-rate maximization [24]. In sum-rate maximization, optimal solution is that each subcarrier is allocated by the user who has better channel conditions [24]. In order to improve fairness, utilization maximization idea is suggested, it ensures more fairness compared to throughput maximization algorithms such as proportional fairness. Moreover, channel-aware resource allocation systems assume that state of channel is always available to scheduler and there are always packets in the queue of each user [24]. The proposed scheduling algorithms in [5, 6, 8, 9] use different scheduling algorithm for different service classes. Paper [7] proposes to use opportunistic extension of deficit round robin (O-DRR) to satisfy delay requirement of different traffics. In paper [10], an adaptive proportional fairness(APF) scheduling is proposed. APF tries to guarantee all different classes quality of services (QoS). In paper [11], first allocate all UGS class packets and continue to allocate all packets of ertPS, rtPS, nrtPS and BE, respectively. Scheduling algorithms for real-time traffics are investigated in [2–4]. [2] and [4] use all queue information for granting. On the other hand, our proposed scheduling algorithm uses partial information. Paper [3] allocates constant grant size to users when they are on state. Packet size is assumed constant for [2], therefore it can obtain all necessary information by using total queue size of user. On the contrary,

packet size is varying in our scheduling algorithm, therefore scheduling problem becomes more challenging.

Wireless networks have limited resources, therefore choosing the most suitable scheduling algorithm is an important process. Scheduling algorithms can be viewed under two main categories which are throughput-optimal scheduling and fair scheduling [25]. In throughput-optimal scheduling, the scheduler at BS aims to increase total throughput of the system by allocating a larger amount of bandwidth to the users with better channel conditions. The resulting scheduling algorithm, however, is not fair. Fully opportunistic scheduling algorithm [26] is one of the most common throughput-optimal scheduling algorithms. On the other hand, airtime fair scheduling algorithms do not take into account channel condition of a user. Scheduler at BS tries to assign equal amount of bandwidth to each user such as the max-min fairness algorithm [27]. Alternatively, data-rate fairness algorithms such as round robin scheduling algorithm [28] try to equalize the amount of data sent. Some scheduling algorithms such as proportional fair scheduling algorithm [29] try to both maximize throughput and not to starve the other users.

2.4.1 Fully Opportunistic Scheduling Algorithm

Good scheduling algorithms always try to improve the spectrum efficiency. One way of doing that is assigning high percentage of bandwidth to users who has higher data rates. However in order to do that system must sacrifice from fairness. If users are wandering around the coverage area, long-term fairness may be satisfied however it is impossible to have short-term fairness because data rate of users does not change so frequently. Therefore during short periods, some users

get low or no bandwidth.

Scheduling Algorithm

1. Online users are sorted with respect to their transmission rates.
2. If there are equality between transmission rates, these users are sorted randomly.
3. Starting from the user who has the highest transmission rate, BS tries to send whole queue of each user.
4. If any packet does not fit in or there are no more packets in the queue, BS passes to next user.
5. This process continues until all packets are sent or whole bandwidth allocated.

Fully opportunistic systems always maximize the total throughput, however it sacrifices fairness among users. Therefore, some users starve when system is congested.

2.4.2 Round Robin Scheduling Algorithm

Round robin scheduling algorithm is used for time-sharing systems. Each user sends one of its packets during its term. Round robin scheduling algorithm forces each user to send equal number of packets.

Scheduling Algorithm

1. Online users are sorted starting from the marked user from the previous frame.
2. BS searches users one by one and add one packet from each user into the list of scheduled packets.
3. This process continues until, all packets are sent or total bandwidth utilized
4. The user who is next to the last assigned user is marked for the next frame.

Round robin scheduling algorithm is a relatively fair algorithm. Each user sends approximately equal number of packets at each frame. If users' average packet size is not varying, each user can send nearly equal amount of bytes. Users who have low transmission rates, are assigned larger bandwidth allocation because they require more air resources to send equal amount of bytes. Therefore, airtime fairness is not achieved in round robin scheduling. In addition, because of this property, network's total throughput is low.

2.4.3 Max-min Fairness Scheduling Algorithm

Aim of max-min fairness algorithm is dividing resources equally among users. However, some users may have fewer demands than its share. In this case, max-min fairness algorithm increases the resource of other users by sharing remained bandwidth equally among others.

Scheduling Algorithm

1. Users are sorted in ascending order by their demands.

2. Total bandwidth is divided by number of online users.

3. If demand of a user is lower than its share, rest of the bandwidth is split up again. And shares of the remained users increase.

4. If not, all users get their assigned shares.

By using max-min fairness algorithm, total bandwidth area is equally shared at each frame. Because of that, it is both short term and long term airtime fair. If users are highly mobile in the environment, system will be long term data-rate fair also. However total throughput may be low since users with low transmission rates utilize more air resources.

2.4.4 Proportional Fair Scheduling Algorithm

Proportional fair scheduling algorithm tries to both maximize throughput and fulfill user's minimal demands [19]. Proportional fair scheduling algorithm try to maximize instantaneous data rate over average data rate for each user. To obtain this, it uses (2.3):

$$P_m(s) = \frac{[DRC_m(s)]^\alpha}{[R_m(s)]^\beta} \tag{2.3}$$

where $\mathrm{DRC}_m(s)$ refers to instantaneous data rate for user m at time s. $\mathrm{R}_m(s)$ refers to average data rate received by user m which is calculated with Equation (2.4):

$$R_m(s) = (1 - \frac{1}{L_T})R_m(s-1) + \frac{1}{L_T}DRC'_m(s-1) \tag{2.4}$$

$\mathrm{DRC}'_m(s-1)$ refers to data rate of user m at time $s-1$, L_T is the averaging coefficient for the exponential weighted moving average. The parameters α and β

are set to 1 for conventional proportional fairness scheduling, When α increases, effect of instantaneous data-rate increases, therefore system will be more close to fully opportunistic. If β increases, affect of average data-rate increases, so system becomes more fair.

Scheduling Algorithm

1. $P_m(s)$ are updated and online users are sorted according to their $P_m(s)$.
2. If there are equality between users' $P_m(s)$s, they will be sorted randomly.
3. Starting from the user who has the highest $P_m(s)$, BS tries to send whole queue of each user.
4. If any packet does not fit in bandwidth allocation, BS will pass to next user.
5. This process continues until all packets are sent or whole bandwidth is allocated.

Proportional fair scheduling algorithm is more fair than fully opportunistic systems because of the denominator of equation (2.3). However, while α in (2.3) increases, importance of instantaneous data rate increases; therefore users who have higher data-rate will get larger space at bandwidth allocation and system become less fair.

Fully opportunistic scheduling algorithm always aims to improve total throughput of the system. Therefore, users with low data rates will be assigned small or no portion of total bandwidth for long term, if network is congested. Therefore, the system cannot be short-term fair and total goodput of system will

suffer from it. On the other hand, it can send more data than other scheduling algorithms by maximizing the throughput. If the system is highly congested, using fully opportunistic scheduling algorithm would be best choice because total throughput of system will increase and it can decrease total number of lost packets. In addition, congested traffic will be overcome in a shorter time period.

Each user can send equal amount of packets by using round robin scheduling algorithm. If packet sizes vary in large range, short-term data-rate fairness will be unbalanced. However, if packet sizes are nearly equal to each other, system will be short-term data-rate fair as well. While users send equal number of packets within the same time interval, the system will sacrifice from total throughput.

Max-min fairness scheduling algorithm always assures airtime fairness. System resources are equally shared among users. If users' data-rates do not change frequently in system, data-rate fairness is not provided because users with low data rates send less data than others. However, total throughput of the system will be greater than round-robin scheduling algorithm.

Proportional fairness algorithm is a hybrid scheduling algorithm. It both tries to improve the total throughput of the system and also try to satisfy minimum demand of users. Therefore, total throughput of the system will be less than fully opportunistic scheduling algorithm however it will be more fair.

In Table 2.2, algorithms are sorted according to importance of the given parameters. For example, fully opportunistic scheduling algorithm is the best algorithm to maximize total throughput of the system, however, it is the worst for the fairness of system.

Table 2.2: Comparison of Different Scheduling Algorithms

	Max. Thr.	S. term Fair	L. Term Fair
Fully Opportunistic	1	4	4
Round Robin	4	2	1
Max Min Fairness	3	2	2
Proportional Fairness	2	3	3

2.5 Fragmentation

If a network layer packet does not perfectly fit in the allocated bandwidth at the link layer, there are two choices: User can divide the packet into two or more pieces (fragmentation) or it can leave a portion of the assigned space empty. Unfilled allocated bandwidth is not a good option as it reduces the efficiency and thus goodput. However, fragmentation causes messaging overhead on the system which means extra bits transmitted over the link. Therefore, user must make wise choices about fragmentation.

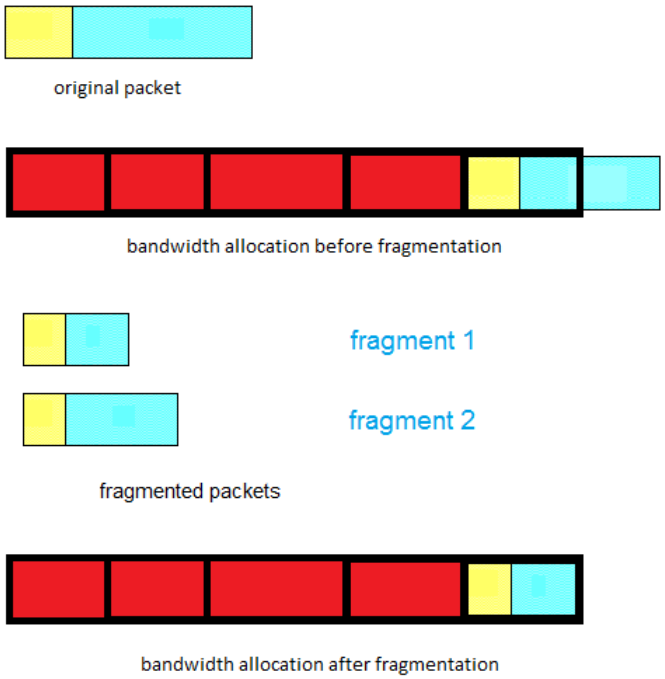

Figure 2.6: Fragmentation

In Figure 2.6, the last packet does not fit in the bandwidth allocation of user. Therefore, the packet is divided into two pieces and the packet transmitted in multiple frames. There will be two small packets, however the total number of transmitted bytes of these two fragments is more than the original packet due to extra overhead emanating from fragmentation. In this thesis, we assume that fragmentation is not allowed in order to reduce the overhead.

In the next chapter, firstly details of proposed uplink scheduling for delay sensitive traffic are provided. After that, brief information is provided about priority mechanism and its properties are discussed. Finally, 3 examples are provided to explain the details of the scheduling algorithm.

Chapter 3

Proposed Uplink Scheduling Algorithm for Delay Sensitive Traffics (USFDST)

In this thesis, a centralized uplink scheduling algorithm is proposed. During the centralized uplink scheduling, scheduler at BS decides on bandwidth allocation of each MS for the next uplink frame. In uplink scheduling, there are some difficulties. The scheduler at BS cannot obtain whole queue information about MSs because gathering all information at BS causes significant messaging overhead and it will decrease spectral efficiency of the network. Therefore, scheduler at BS must make its decision by using partial information which is provided by MSs. Meanwhile, MSs have to make smart choices about which information will be sent to BS. Under or over messaging may mislead the BS and make the scheduling more difficult.

In this thesis, MSs have an active role on scheduling. They decide on their bandwidth requirements. For this purpose, they use bandwidth allocation of

other users at previous frame, size of their queues, its own bandwidth allocation at previous frames. By using these parameters, each MS determines on two different bandwidth requests: One of them is greedy and the other is conservative. After that, user will send to BS only these two bandwidth requests and priority information. Priority basically means that user's queue is expanding and if it does not get more bandwidth allocation, it will start to lose packets or send excessively delayed packets which are useless for real-time applications. The aim of this scheduling algorithm is to improve total goodput of the system for real-time applications (e.g. VOIP). Therefore, decreasing number of excessively delayed packets is one of the significant objective of algorithm.

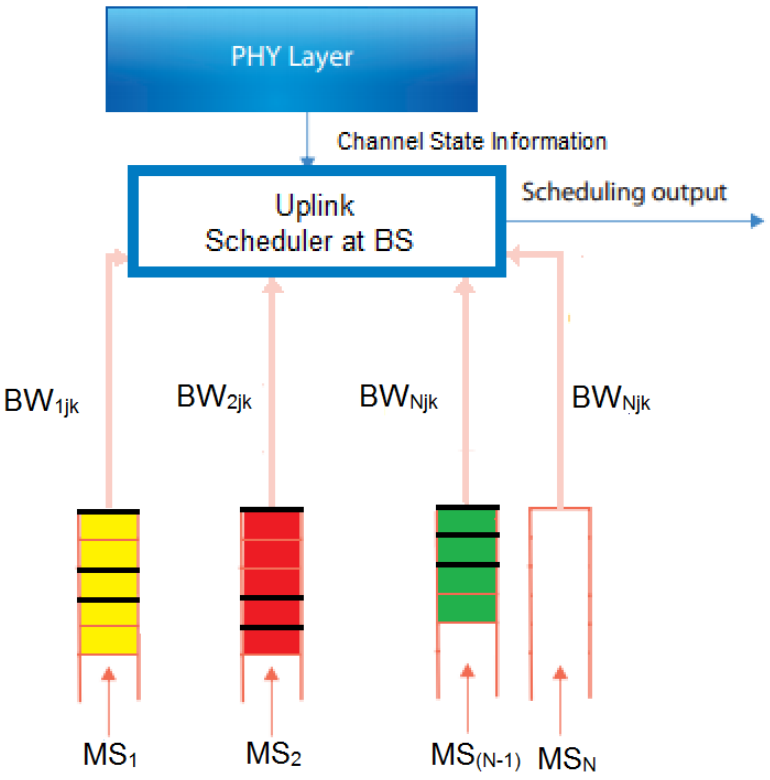

Figure 3.1: Request part of Algorithm

In Figure 3.1, each user sends its bandwidth requests to the scheduler at BS.

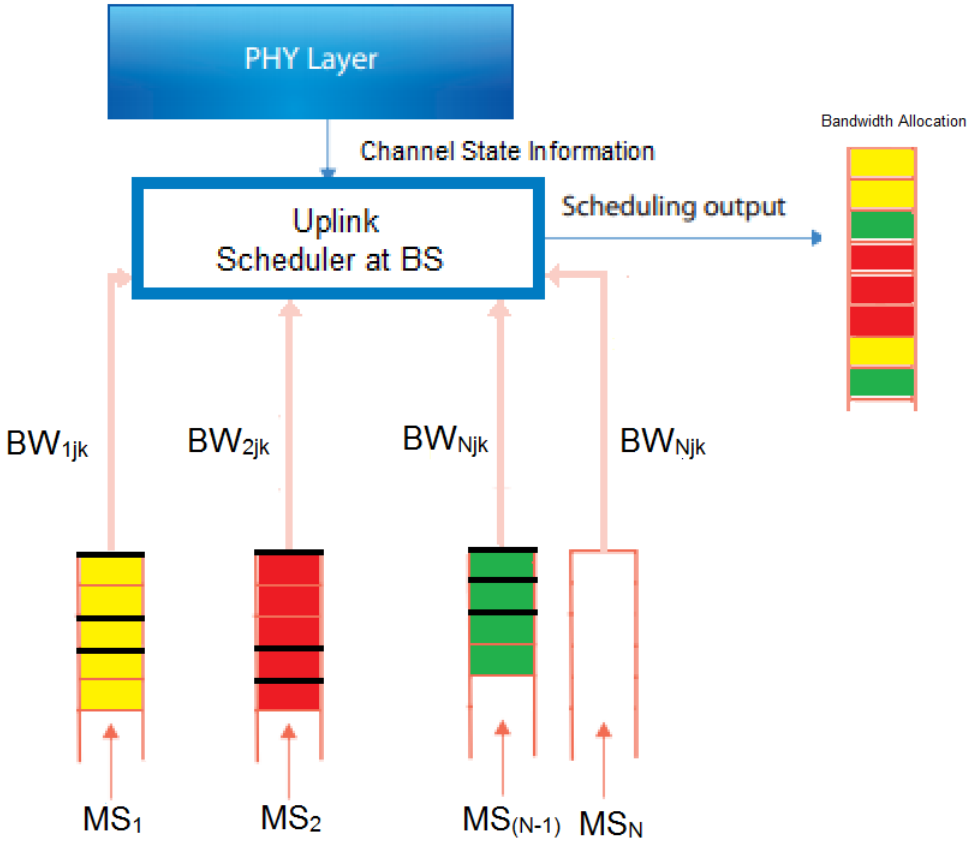

Figure 3.2: Granting part of Algorithm

In Figure 3.2, the scheduler at BS separates users into two group according to priority information: prior group, non-prior group. After that, the scheduler at BS sorts users in each group according to number of token in descending order and allocates conservative bandwidth requests of prior users using this order. If conservative requests of each user in prior group are satisfied, scheduler at BS starts to assign greedy request of these users according to same order. After that, scheduler at BS do same process for non-prior group.

In this chapter, details of proposed algorithm are explained in Section 3.1. In Section 3.2, priority is discussed. Finally, in Section 3.3 three different scenario is provided.

3.1 Uplink Scheduling Algorithm for Delay Sensitive Traffic

In this part, firstly request part of uplink scheduling algorithm for delay sensitive traffic is explained in Section 3.1.1. In Section 3.1.2, granting part of uplink scheduling algorithm for delay sensitive traffic is explained.

The following parametes are used by the proposed uplink scheduling algorithm:

- N : Number of users in the system
- N_{Aj} : Number of active users in the system at frame j
- BW_{ijk} : BW requests of user i at frame j (k=1: conservative, k=2: greedy)
- BWA_{ij} : Assigned bandwidth of user i at frame j
- BW_{ijkb} : highest limit for BW requests for user i at frame j (k=1: conservative, k=2: greedy)
- $Cond_{ij}$: Condition of user i at frame j. Condition of user i gives the relation between $BW_{ij:}$ and BWA_{ij}
- B : Total bandwidth of system
- R_{ij}: Tranmission rate of user i at frame j
- Pr_{ij} : if user i is alarmed at frame j, Pr_{ij}=1, else 0
- Tk_i: # of token of user i
- Tk : Constant token which is added to each user's bucket at the beginning of each frame

- BD_L : Use to determine on user's alarm set off or not. if TDR_i is less than BD_L and $Pr_{i(j-1)} = 1$, Pr_{ij} will set to 0.
- BD_H : Use to determine on user's alarm set on or not. if TDR_i is higher than BD_H and $Pr_{i(j-1)} = 0$, Pr_{ij} will set to 1.
- PoB: Percentage of occupied bandwidth in the previous frame
- QT_{ij}:Total number of bytes in the queue of user i at frame j
- TDR_i : Total number of needed frame to send the whole queue of user i, if its assigned bandwidth will be average of last 10 granted bandwidths
- $ML10Req_i$: Average bandwidth for user i assigned over the last 10 frames

3.1.1 Request Part of Uplink Scheduling Algorithm for Delay Sensitive Traffic

1. Add constant number of tokens to each user's bucket according to Formula 3.1

$$Tk_i = Tk_i + Tk \tag{3.1}$$

2. Decide on BW_{ij1b} and BW_{ij2b}
   - If $Cond_{ij}$=1

     When $Cond_{ij}$ =1, it means that $BW_{i(j-1)1}$ or $BW_{i(j-1)2} \neq 0$, however $BWA_{i(j-1)}$ =0

     - If $Pr_{ij} = 1$
       - If there is only one user whose $Cond_{ij}$=1

$$BW_{ij1b} = BW_{i(j-1)1} \tag{3.2}$$

$$BW_{ij2b} = min(B, 0.01 * B + BW_{ij1b}) \tag{3.3}$$

* If there are more users whose Cond_{ij}=1

$$BW_{ij1b} = BW_{i(j-1)1} \tag{3.4}$$

$$BW_{ij2b} = min(B, 0.005 * B + BW_{ij1b}) \tag{3.5}$$

– If Pr_{ij} =0

* If there is only one user whose Cond_{ij} =1

$$BW_{ij1b} = 0.75 * BW_{i(j-1)1} \tag{3.6}$$

$$BW_{ij2b} = min(B, 0.01 * B + BW_{ij1b}) \tag{3.7}$$

* If there are more users whose Cond_{ij}=1

$$BW_{ij1b} = 0.75 * BW_{i(j-1)1} \tag{3.8}$$

$$BW_{ij2b} = min(B, 0.005 * B + BW_{ij1b}) \tag{3.9}$$

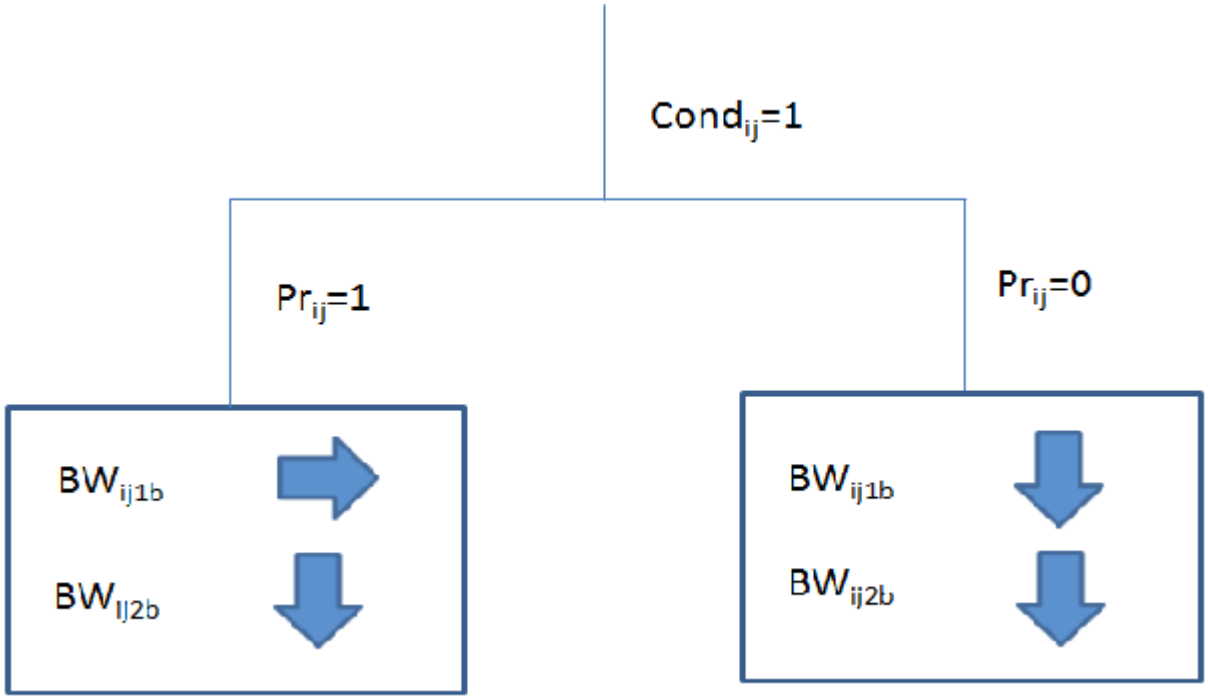

Figure 3.3: Cond$_{ij}$=1 Chart

Figure 3.3 explains the operation chart of users with Cond$_{ij}$=1. If a user does not get its conservative request, it means that whether network is congested or user's conservative request is too large to fit in. If there are more users in the network whose condition is 1, congestion is more likely reason. Therefore users must reduce their requests. However, if Pr$_{ij}$=1, reducing bandwidth requests may be harmful for the user because user needs large amount of bandwidth allocation immediately. Therefore, repeating the same request will be more rational solution. Otherwise, although user gets some bandwidth allocation, it may be useless for it.

- If Cond$_{ij}$=2

  When Cond$_{ij}$ =2, it means that BW$_{i(j-1)2} \neq 0$, however BWA$_{i(j-1)}$ = BW$_{i(j-1)1}$

– If there is no user whose Cond_{ij}=1 and there are some users whose $\text{Cond}_{ij} = 3$

  * Pr_{ij} =1

$$BW_{ij1b} = min(B, 0.02 * B + BW_{i(j-1)1}) \tag{3.10}$$

$$BW_{ij2b} = min(B, 0.01 * B + *BW_{ij1b}) \tag{3.11}$$

  * Pr_{ij} =0

$$BW_{ij1b} = min(B, 0.01 * B + BW_{i(j-1)1}) \tag{3.12}$$

$$BW_{ij2b} = min(B, 0.01 * B + BW_{ij1b}) \tag{3.13}$$

– If there is no user whose Cond_{ij}=1 and there is no user whose Cond_{ij}= 3 either

$$BW_{ij1b} = min(B, 0.01 * B + BW_{i(j-1)1}) \tag{3.14}$$

$$BW_{ij2b} = min(B, 0.01 * B + BW_{ij1b}) \tag{3.15}$$

– If there are some users whose Cond_{ij}=1 and there are some users whose Cond_{i}= 3 either

  * Pr_{ij}=1

$$BW_{ij1b} = min(B, 0.01 * B + BW_{i(j-1)1}) \tag{3.16}$$

$$BW_{ij2b} = min(B, 0.01 * B + BW_{ij1b}) \tag{3.17}$$

* $\Pr_{ij}$=0

$$BW_{ij1b} = min(B, BW_{i(j-1)1}) \tag{3.18}$$

$$BW_{ij2b} = min(B, 0.01 * B + BW_{ij1b}) \tag{3.19}$$

– If there are some users whose Cond_{ij}=1 and there is no user whose Cond_{ij}= 3

* $\Pr_{ij}$=1

$$BW_{ij1b} = min(B, BW_{i(j-1)1}) \tag{3.20}$$

$$BW_{ij2b} = min(B, 0.01 * B + BW_{ij1b}) \tag{3.21}$$

* $\Pr_{ij}$=0

$$BW_{ij1b} = min(B, 0.9 * BW_{i(j-1)1}) \tag{3.22}$$

$$BW_{ij2b} = min(B, 0.01 * B + *BW_{ij1b}) \tag{3.23}$$

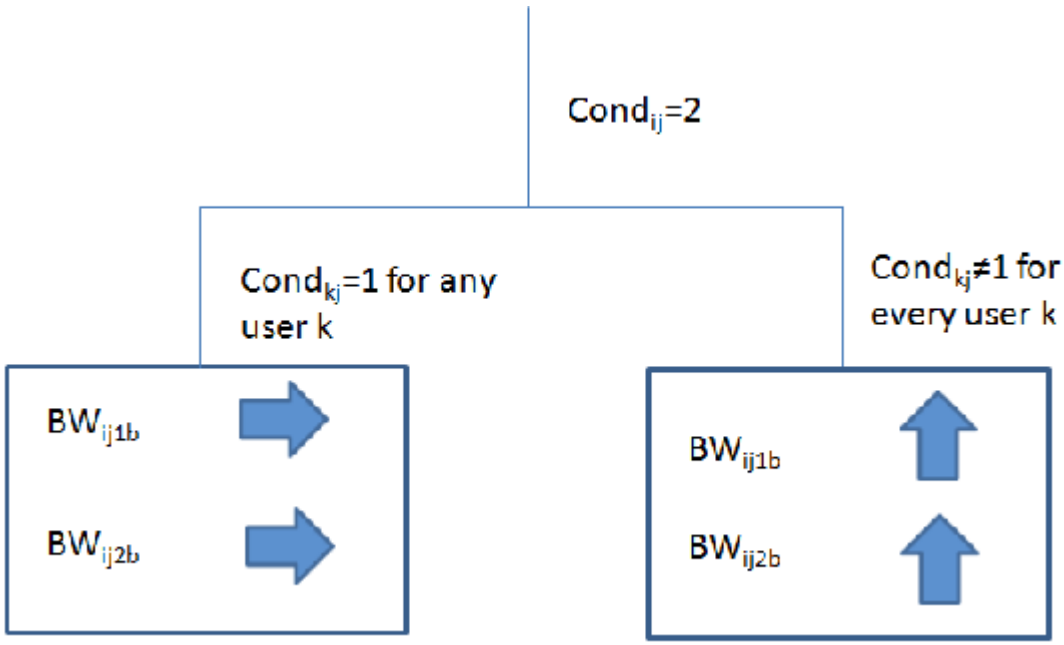

Figure 3.4: Cond_{ij}=2 Chart

Figure 3.4 explains the operation chart of users with Cond_{ij}=2. If a user gets its conservative request, its primary aim is protecting its previous bandwidth allocation and if it is possible, it will try to get its greedy request. If congestion is not severe (i.e. there is no user with Cond_{kj}=1), it increases its requests a little bit. Otherwise, it won't change its requests.

- If Cond_{ij}=3

  When Cond_{ij}=3, it means that $\text{BWA}_{i(j-1)} = \max(\text{BW}_{i(j-1)1}, \text{BW}_{i(j-1)2})$

  - If all online user's Cond_{ij}=3

    * If it sends all its queue at previous frame

$$BW_{ij1b} = min(B, 1.5 * BWA_{i(j-1)}) \tag{3.24}$$

$$BW_{ij2b} = min(B, 0.02 * B + BW_{ij1b}) \tag{3.25}$$

    * If it cannot send all its queue in the previous frame

$$BW_{ij1b} = min(B, 0.4 * B/n_{Aj} + 0.75 * (1/PrF) * BWA_{i(j-1)}) \tag{3.26}$$

$$BW_{ij2b} = min(B, 0.02 * B + *BW_{ij1b}) \tag{3.27}$$

  - If all online users' Cond_{ij} is not 3

    * If it sends its all queue in the previous frame

$$BW_{ij1b} = min(B, BWA_{i(j-1)}) \tag{3.28}$$

$$BW_{ij2b} = min(B, 0.02 * B + BW_{ij1b}) \tag{3.29}$$

* If it does not send all its queue in the previous frame

$$BW_{ij1b} = min(B, 0.4 * B/n_{Aj} + 0.6 * (1/PoB) * BWA_{i(j-1)}) \tag{3.30}$$

$$BW_{ij2b} = min(B, 0.02 * B + *BW_{ij1b}) \tag{3.31}$$

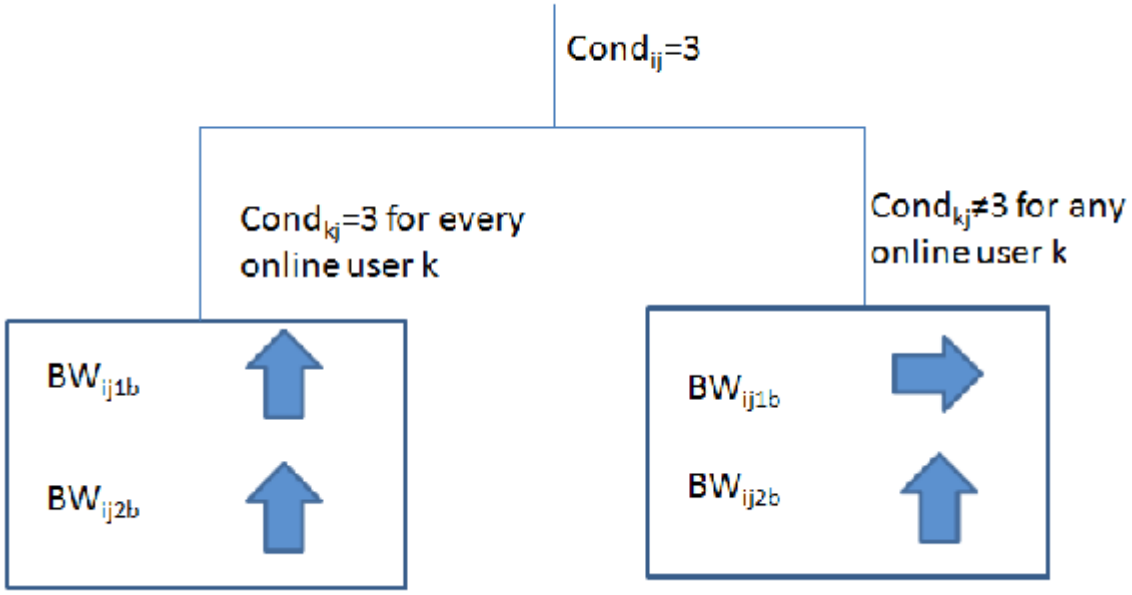

Figure 3.5: Cond$_{ij}$=3 Chart

Figure 3.5 explains the operation chart of users with Cond$_{ij}$=3. If user sends its whole queue in the previous frame, its bandwidth requirements are relatively low. It does not suffer highly from low bandwidth allocation at the next frame. Therefore, it will try to increase its bandwidth allocation if network is not congested (i.e. all online user's Cond$_{ij}$ = 3). On the other hand, if user does not send its all queue in the previous frame, bandwidth allocation is more necessary for the user in next frame. Therefore, it adjusts its requests according to the previous bandwidth allocation. If PoB is higher (i.e. bandwidth is full or nearly full), it will be

more conservative, otherwise it will be more aggressive.

- If Cond_{ij}=0

  It means$BW_{i(j-1)1}$ and $BW_{i(j-1)2}$ are equal to 0

$$BW_{ij1b} = BW_{min} \tag{3.32}$$

$$BW_{ij2b} = BW_{ij1b} + 1packet \tag{3.33}$$

$$BW_{min} = 0.8 * B/N \tag{3.34}$$

  For users who are not online in the previous frame, BW_{min} is a starting point. User does not have any information about network's condition therefore it starts slowly, and in the next frame, it will adjust its bandwidth requests according to the congestion of network.

3. Each user decides on whether it is alarmed or not according to the Equation (3.35):

$$TDR_i = QT_{ij}/R_{ij}/ML10Req_i \tag{3.35}$$

   if $\text{TDR}_i > \text{BD}_H$ and Pr_{ij}=0

   it sets Pr_{ij}=1

   else if $\text{TDR}_i < \text{BD}_L$ and Pr_{ij} =1

   it sets Pr_i=0

4. User will determines on bandwidth requests for next frame by using BW_{ij1b} and BW_{ij2b}. They do not use any fragmentation and request highest number of unsplit packets which is less than BW_{ij1b} and BW_{ij2b}. While deciding BW_{ij1}, if first packet in the queue is larger than BW_{ij1b}, user can extend it and will send first packet as a request. After decision on BW_{ij1}, if BW_{ij1} plus next packet in the queue exceeds BW_{ij2b}, BW_{ij2} will be assigned as next packet in the queue plus BW_{ij1}.

3.1.2 Granting Part of Uplink Scheduling Algorithm for Delay Sensitive Traffic

In this part, granting of uplink scheduling algorithm for delay sensitive traffic is explained. After that, an example scenario is provided.

1. Firstly, scheduler sort alarmed users with respect to their tokens.
2. After that if $B - \sum_{k=1}^{i-1} BWA_{kj} > BW_{ij1}$

$$BWA_{ij} = BW_{ij1} \tag{3.36}$$

3. If there is still empty space and $B - \sum_{k=1}^{i-1} BWA_{kj} > BW_{ij2} - BW_{ij1}$, the scheduler updates bandwidth assignment of user i as

$$BWA_{ij} = BW_{ij2} \tag{3.37}$$

4. After bandwidth allocation for alarmed users, scheduler assigns bandwidth to non-alarmed ones. Scheduler sorts them with respect to number of tokens. Next, if $B - \sum_{Pr_{ij}=1} BWA_{:j} - \sum_{k=1}^{i-1} BWA_{kj} > BW_{ij1}$

$$BWA_{ij} = BW_{ij1} \tag{3.38}$$

5. If there is still empty space and $B - \sum_{Pr_{ij}=1} BWA_{:j} - \sum_{k=1}^{i-1} BWA_{kj} > BW_{ij2} - BW_{ij1}$

   scheduler will update bandwidth assignment of user i as

$$BWA_{ij} = BW_{ij2} \tag{3.39}$$

6. $Tk_i = Tk_i - BWA_{ij}$

7. Assign condition of users

   - $BWA_{ij} = max(BW_{ij1}, BW_{ij2})$ then $\text{Cond}_{i(j+1)}$=3
   - $BWA_{ij} = BW_{ij1}$ then $\text{Cond}_{i(j+1)}$=2
   - BW_{ij1} or $\text{BW}_{ij2} \neq 0$ but $BWA_{ij} = 0$ then $\text{Cond}_{i(j+1)}$=1
   - BW_{ij1} and $\text{BW}_{ij2} = 0$ then $\text{Cond}_{i(j+1)}$=0

8. Determine ML10Req_i of users: User's oldest request is subtracted and newest request is added instead of that. However, if both requests of a user are less than BW_{min} and user is granted for its maximum request, then while calculating ML10Req_i, user's grant is assumed BW_{min} to avoid false alarm. On the other hand, if user has requested for a bandwidth but it is not granted any, then while calculating ML10Req_i, user's grant is taken as 0.

Granting Example

In this part, a granting example for uplink scheduling algorithm for delay sensitive traffic is provided.

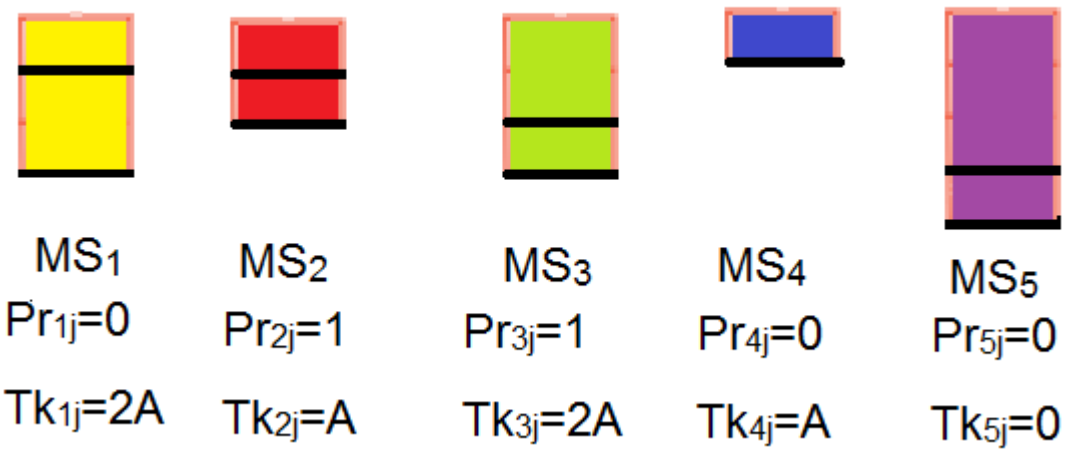

Figure 3.6: Bandwidth Requests of Users

In Figure 3.6, there are 5 users and their bandwidth request are shown. Two of users are alarmed and the others not.

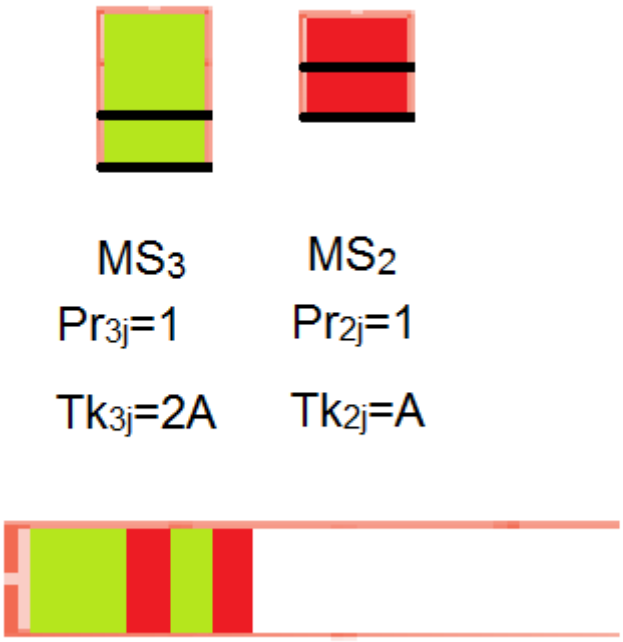

Figure 3.7: BW Allocation of alarmed users

In Figure 3.7, the scheduler at BS sorts alarmed users according to number of token. After that, allocate their conservative and greedy requests respectively.

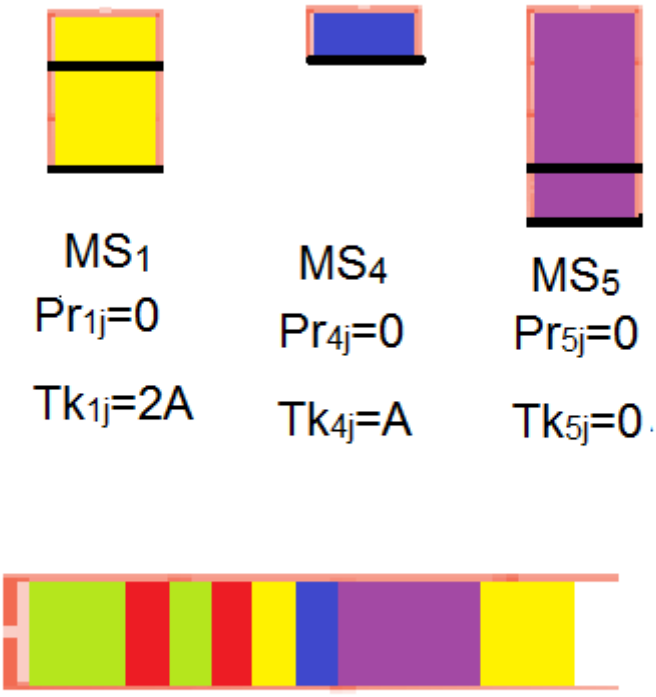

Figure 3.8: BW Allocation of Non-Alarmed users

In Figure 3.8, the scheduler at BS sorts non-alarmed users according to number of token and allocate their conservative and greedy requests respectively. Greedy request of MS_5 does not fit in bandwidth allocation. Therefore, it gets its conservative request.

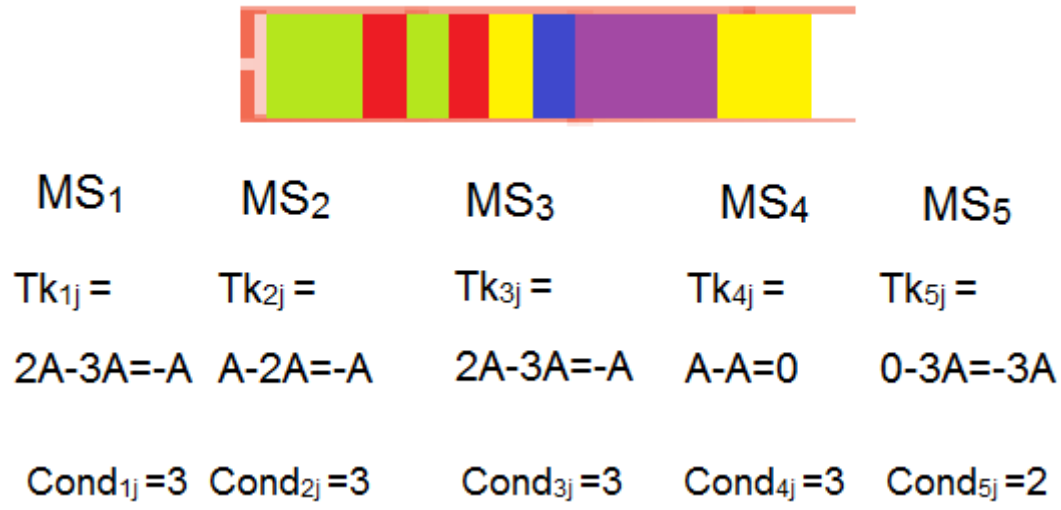

Figure 3.9: Updating Tk_i and to determine on Cond_{ij}

In Figure 3.9, the scheduler updates Tk_i and determines on Cond_{ij} of each user according to BWA_{ij}.

3.2 Priority

The aim of priority mechanism is (i) allocating more bandwidth to users whose queue cannot be emptied in reasonable amount of time, (ii) by early intervention, prevent users' queue to become full and start losing packets. However, this mechanism can be effective when number of alarmed users is low in the network. If all users are alarmed, this mechanism will be harmful for system because when users are alarmed, scheduling algorithm always try to expand their bandwidth allocation. Therefore, this scheduling algorithm may be useless for saturated networks.

3.3 Simulation Examples

In this section, brief explanation will be provided about the operation of the scheduling algorithm. There are three different scenarios. In the first one, network is uncongested; users can easily get their greedy request. In the second one, network is more congested. Therefore, some users will get their conservative request and in some cases, they won't get any bandwidth. Third one is the most congested network and priority mechanism will be explained in this part. In these scenarios, most common cases that can be faced are investigated.

3.3.1 Example 1- Uncongested network without priority

In this scenario, network is uncongested. We choose R_{ij} =1 for all users to make scenario simpler.

Users' previous Tk_i, $BW_{i(j-1)1}$, $BW_{i(j-1)2}$, $BWA_{i(j-1)}$ and $Cond_{ij}$ are in Table 3.1.

Table 3.1: Previous Parameters

User	$QT_{i(j-1)}$	Tk_i	$BW_{i(j-1)1}$	$BW_{i(j-1)2}$	$BWA_{i(j-1)}$	$Cond_{ij}$
1	8800	11000	7100	8800	8800	3
2	2700	23000	1600	1900	1900	3
3	2200	8000	2000	2200	2200	3
4	0	52000	0	0	0	0

Request

1. Firstly, number of tokens of users is updated as shown in Table 3.2.

Table 3.2: Token Updates

User	Tk_i
1	11000+4000=15000
2	23000+4000=27000
3	8000+4000=12000
4	52000+4000=56000

2. Highest limits are decided for BW Requests as shown in Table 3.3. If a user sends its whole queue in the previous frame, its necessity for bandwidth is relatively low. Therefore, for next frame, scheduler courages other users to make more greedy requests. To make these requests more realistic, users use PoB to determine next requests.

Table 3.3: Determination of BW_{ij1b} and BW_{ij2b}

User	BW_{ij1b}	BW_{ij2b}
1	1.5*8800=13200	0.02*20000+13200=13600
2	0.4*20000/3 + 0.75*(1/0.685)*1900=4747	0.02*20000+4747=5147
3	1.5*2200=3300	0.02*20000+3300=3700
4	0.8*20000/4=4000	4000+ 1packets

3. After that, users will decide on their requests as shown in Table 3.4 by using the limits given at Table 3.3.

Table 3.4: BW_{ij1} and BW_{ij2} values

User	QT_{ij}	BW_{ij1}	BW_{ij2}
1	5500	5500	0
2	4200	4200	0
3	4800	3000	3500
4	7800	3900	4800

Granting

1. Scheduler firstly, sorts user with respect to their tokens in descending order as given in Table 3.5.

Table 3.5: Sorting of users

User	Tk_i	BW_{ij1}	BW_{ij2}
4	56000	3900	4800
2	27000	4200	0
1	15000	5500	0
3	12000	3000	3500

2. After that, scheduler will decide on bandwidth allocation of users by using the order in Table 3.5. Scheduler assigns conservative requests of users at first. There is empty space in bandwidth, therefore scheduler assigns greedy requests of users also as given in Table 3.6.

Table 3.6: Assigning BWA_{ij} and $Cond_{ij}$ and Updating Tk_i

User	QT_{ij}	BWA_{ij}	$Cond_{i(j+1)}$	Tk_i
1	5500	5500	3	15000-5500=9500
2	4200	4200	3	27000-4200=22800
3	4800	3500	3	12000-3500=8500
4	7800	4800	3	56000-4800=51200

In this example, network is not congested; therefore users can get their greedy bandwidth requests easily. If there is not any special arrangement for user 2(i.e. user who does not send its whole queue at previous frame), its BW_{ij1b} and BW_{ij2b} will be lower (2850 and 3250). The adjustment helps users to increase their BW_{ij1b} and BW_{ij2b} more quickly in uncongested networks.

3.3.2 Example 2-Congested network without priority

In this scenario, network is more congested than example 1. Therefore, users have to be more conservative about their bandwidth requests. We choose R_{ij} =1 for all users to make scenario simpler.

Users' previous Tk_i, $BW_{i(j-1)1}$, $BW_{i(j-1)2}$, $BWA_{i(j-1)}$ and $Cond_{ij}$ are provided in the Table 3.7.

Table 3.7: Previous Parameters

User	$QT_{i(j-1)}$	Tk_i	$BW_{i(j-1)1}$	$BW_{i(j-1)2}$	$BWA_{i(j-1)}$	$Cond_{ij}$
1	17000	16000	9200	9900	0	1
2	10000	22000	5500	6200	6200	3
3	4700	80000	4300	4700	4700	3
4	12300	-15000	5200	5600	5600	3

Request

1. Firstly, number of tokens of users is updated as shown in Table 3.8.

Table 3.8: Token Updates

User	Tk_i
1	16000+4000=20000
2	22000+4000=26000
3	80000+4000=84000
4	-15000+4000=-11000

2. Highest limits are decided for BW Requests as shown in Table 3.9. If a user did not assign any bandwidth for previous frame (e.g. user 1), there are two possibilities: network may be congested or previous requests are too big to fit in. Solution for both problem is to decrease requests. In addition, because of the arrangement for user with $Cond_{ij}$=3, users with $Cond_{ij}$=3 become more conservative because PoB is close to 1 (i.e. network is congested) and this mechanism forces them to make more conservative requests.

Table 3.9: BW_{ij1b} and BW_{ij2b} values

User	BW_{ij1b}	BW_{ij2b}
1	0.75*9200=6900	0.01*20000+6900=7100
2	0.4*20000/4 + 0.6*(1/0.82)*6200=6536	0.02*20000+6536=6936
3	4700	0.02*20000+4700=5100
4	0.4*20000/4+0.6*(1/0.82)*5600=6097	6097+ 400=6497

3. After that, users will decide on their requests as shown in Table 3.10 by using the limits at given at Table 3.9. When table is examined, BW_{1j2} is greater than BW_{1j2b} because after deciding on BW_{1j1}, size of the next packet in the queue of user 1 is 900 bytes. Fragmentation is not allowed in the algorithm; therefore user 1 can send a request greater than the predetermined limit.

Table 3.10: BW_{ij1} and BW_{ij2} values

User	QT_{ij}	BW_{ij1}	BW_{ij2}
1	23000	6400	7300
2	14000	6200	7100
3	600	600	0
4	11000	5400	6200

Granting

1. Scheduler, firstly, sorts user with respect to their tokens in descending order as shown in Table 3.11.

Table 3.11: Sorting of users

User	Tk_i	BW_{ij1}	BW_{ij2}
3	84000	600	0
2	26000	6200	7100
1	20000	6400	7300
4	-11000	5400	6200

2. After that, scheduler decides on bandwidth allocation of users as shown in Table 3.12 by using the order in Table 3.11. Scheduler assigns conservative requests of users at first. There is an empty space in bandwidth allocation, greedy requests of user 2 and 3 fit in this empty space, therefore scheduler updates their allocation.

Table 3.12: Assigning BWA_{ij} and $Cond_{ij}$ and Updating Tk_i

User	QT_{ij}	BWA_{ij}	$Cond_{i(j+1)}$	Tk_i
1	23000	6400	2	20000-5500=14500
2	14000	7100	3	26000-7100=18900
3	600	600	3	84000-600=83400
4	11000	5400	2	-11000-5400=-16400

Next frame:

Request

1. Firstly, number of tokens of users is updated as shown in Table 3.13 .

Table 3.13: Token Updates

User	Tk_i
1	14500+4000=18500
2	18900+4000=22900
3	83400+4000=87400
4	-16400+4000=-12400

2. Highest limits are decided for BW Requests as shown in Table 3.14. User 1 and user 4 do not get their greedy requests at previous frame, however the others can. Therefore, their primary aim is to protect their previous bandwidth allocation. User 2's bandwidth allocation is greater than average of network (20000/4=5000) and network is congested.

Table 3.14: BW_{ij1b} and BW_{ij2b} values

User	BW_{ij1b}	BW_{ij2b}
1	6400+200=6600	0.01*20000+6600=6800
2	0.4*20000/4 + 0.6*(1/0.975)*7100=6369	0.02*20000+6369=6769
3	600	0.02*20000+600=1000
4	5400+0.01*20000=5600	0.01*20000+ 5600=5800

3. After that, users will decide on their requests as shown in Table 3.15 by using limits at given in Table 3.14.

Table 3.15: BW_{ij1} and BW_{ij2} values

User	QT_{ij}	BW_{ij1}	BW_{ij2}
1	19000	6500	6700
2	7600	6300	6600
3	2400	1200	1700
4	7000	5500	6200

Granting

1. Scheduler firstly, sorts user with respect to their tokens descending order as shown in Table 3.16.

Table 3.16: Sorting of users

User	Tk_i	BW_{ij1}	BW_{ij2}
3	87400	1200	1700
2	22900	6300	6600
1	18500	6500	6700
4	12400	5500	6200

2. After that, scheduler decides on bandwidth allocation of users as shown in Table 3.17 by using the order in Table 3.16. Scheduler assigns conservative requests of users at first. There is an empty space in bandwidth allocation, greedy request of user 3 fits in this empty space, therefore scheduler updates its allocation.

Table 3.17: Assigning BWA_{ij} and $Cond_{ij}$ and Updating Tk_i

User	QT_{ij}	BWA_{ij}	$Cond_{i(j+1)}$	Tk_i
1	19000	6500	2	18500-6500=12000
2	7600	6300	2	22900-6300=16600
3	2400	1700	3	87400-1700=85700
4	7000	5500	2	-12400-5500=-17900

In the example 2, network is more congested than example 1. If users were aggressive like example 1, they might get no bandwidth allocation.

3.3.3 Example 3- Congested network with priority

This is the most congested scenario among the three examples that we consider. Pr_{ij}=1 for some users, it means that their queue length is too large and they can send them all in more than BD_H frame, if they assign $ML10Req_i$ for bandwidth. We choose R_{ij} =1 for all users to make scenario simpler. In addition, BD_L=5 and BD_H=10 for this example.

Users previous Tk_i, $BW_{i(j-1)1}$, $BW_{i(j-1)2}$, $BWA_{i(j-1)}$, $Cond_{ij}$, $Pr_{i(j-1)}$ and $ML10Req_i$ are provided in the Table 3.18 and 3.19.

Table 3.18: Previous Parameters I

User	$QT_{i(j-1)}$	Tk_i	$BW_{i(j-1)1}$	$BW_{i(j-1)2}$	$BWA_{i(j-1)}$
1	75000	-30000	7300	8000	8000
2	35000	10000	6700	7000	7000
3	40000	22000	4800	6000	4800
4	0	94000	0	0	0

Table 3.19: Previous Parameters II

User	$Cond_{ij}$	$Pr_{i(j-1)}$	$ML10Req_i$
1	3	1	4000
2	3	1	5500
3	2	0	4500
4	0	0	3000

Request

1. Firstly, number of tokens of users is updated as shown in Table 3.20.

Table 3.20: Token Updates

User	Tk_i
1	-30000+4000=-26000
2	10000+4000=14000
3	22000+4000=26000
4	94000+4000=98000

2. Highest limits are decided for BW Requests as given in Table 3.21.

Table 3.21: BW_{ij1b} and BW_{ij2b} values

User	BW_{ij1b}	BW_{ij2b}
1	0.4*20000/3+0.6*(1/0.99)*8000=7515	0.02*20000+7515=7915
2	0.4*20000/3 + 0.6*(1/0.99)*7000=6909	0.02*20000+6909=7309
3	0.01*20000+ 4800=5000	0.01*20000+5000=5200
4	0.8*20000/4=4000	4000+ 1packets

3. Priorities are updated as shown in Table 3.22. Although, TDR_3 is greater than TDR_2, user 2 is alarmed, while user 3 is not, because $Pr_{2(j-1)}$=1 and TDR_2 is over the BD_L. On the other hand, $Pr_{3(j-1)}$=0 and TDR_2 is under the BD_H.

   While calculating $ML10Req_i$, oldest bandwidth allocation is subtracted and last bandwidth allocation added instead.

Table 3.22: Pr_{ij} are updated

User	QT_{ij}	$ML10Req_i$	TDR_i	Pr_{ij}
1	75000	(4000*10-7000+8000)/10=4100	18.29	1
2	35000	(5500*10-4500+7000)/10=5750	6.08	1
3	40000	(4500*10-2500+4800)/10=4730	8.45	0
4	0	3000	0	0

4. After that, users will decide on their requests as shown in Table 3.23 by using the limits at given in Table 3.21.

Table 3.23: BW_{ij1} and BW_{ij2} values

User	QT_{ij}	BW_{ij1}	BW_{ij2}
1	71000	7500	8400
2	30000	6200	6600
3	42000	4500	4750
4	2000	2000	0

Granting

1. Scheduler, firstly, sorts user whose Pr_{ij}=1 with respect to their tokens in descending order. After that, it will sort other user with respect to their tokens in descending order either as shown in Table 3.24.

Table 3.24: Sorting of users

User	Tk_i	BW_{ij1}	BW_{ij2}	Pr_{ij}
2	14000	6200	6600	1
1	-26000	7500	8400	1
4	98000	2000	0	0
3	26000	4500	4750	0

2. After that, scheduler decides on bandwidth allocation of users as shown in Table 3.25 by using the order in Table 3.24. Scheduler assigns conservative requests of users with Pr_{ij}=1 at first. After that, there is an empty space, greedy request of user 1 and 2 fit in this empty space. therefore, scheduler updates their allocation. Finally, it will pass on the others and do same process in the same order. The aim of the idea is lessening queues of alarmed users to protect those users from packet losses and sending packets

with excessive delays.

Table 3.25: Assigning BWA_{ij} and $Cond_{ij}$ and Updating Tk_i

User	QT_{ij}	BWA_{ij}	$Cond_{i(j+1)}$	Tk_i
1	71000	8400	3	-26000-8400=-34400
2	30000	6600	3	14000-6600=7400
3	42000	0	1	26000-0=26000
4	2000	2000	3	98000-2000=96000

Next frame:

Request

1. Firstly, number of tokens of users is updated as given in Table 3.26.

Table 3.26: Token Updates

User	Tk_i
1	-34400+4000=-30400
2	7400+4000=11400
3	26000+4000=30000
4	96000+4000=100000

2. Highest limits are decided for BW Requests as given in Table 3.27.

Table 3.27: BW_{ij1b} and BW_{ij2b} values

User	BW_{ij1b}	BW_{ij2b}
1	0.4*20000/4+0.6*(1/0.85)*8400=7929	0.02*20000+7929=8329
2	0.4*20000/4 + 0.6*(1/0.85)*6200=6376	0.02*20000+6376=6776
3	0.75*4500=3375	0.005*20000+3375=3475
4	2000	0.02*20000+2000=2200

3. Priorities are updated as shown in Table 3.28. While calculating $ML10Req_3$, 0 is added because user made request at previous frame but it didn't get any. In addition, although $BWA_{4(j-1)}$=2000, 4000(BW_{min}) is added to calculate $ML10Req_4$ because it gets its maximum request and it is less than BW_{min}, this precaution is used to prevent false alarms. This user gets low bandwidth allocation because of its requests.

   At this frame, Pr_{2j} becomes 0, because TDR_2 is now lower than BD_L. At the same time, Pr_{3j} becomes 1 because TDR_3 is higher than BD_H.

Table 3.28: Pr_{ij} are updated

User	QT_{ij}	$ML10Req_i$	TDR_i	Pr_{ij}
1	63000	(4100*10-2500+8400)/10=4690	13.43	1
2	26000	(5750*10-0+6600)/10=6410	4.05	0
3	50000	(4730*10-7500+0)/10=3980	12.56	1
4	3500	(3000*10-2500+4000)/10=3150	1.11	0

4. After that, users will decide on their requests as shown in Table 3.29 by using limits given at Table 3.27.

Table 3.29: BW_{ij1} and BW_{ij2} values

User	QT_{ij}	BW_{ij1}	BW_{ij2}
1	63000	7700	8200
2	26000	6100	6600
3	50000	3200	4100
4	3500	1500	2200

Granting

1. Scheduler, firstly, sorts user whose Pr_{ij}=1 with respect to their tokens in descending order. After that, it will sort other user with respect to their tokens in descending order either as shown in Table 3.30.

Table 3.30: Sorting of users

User	Tk_i	BW_{ij1}	BW_{ij2}	Pr_{ij}
3	30000	3200	4100	1
1	-30400	7700	8200	1
4	100000	1500	2200	0
2	11400	6100	6600	0

2. After that, scheduler decides on bandwidth allocation of users as shown in Table 3.31 by using the order in Table 3.30. Scheduler assigns conservative requests of users with Pr_{ij}=1 at first. After that, there is an empty space in bandwidth, greedy requests of user 1 and 3 fit in this empty space. Therefore, scheduler updates their allocation. Finally, it will pass on the others and execute the same process in same order.

Table 3.31: Assigning BWA_{ij} and $Cond_{ij}$ and Updating Tk_i

User	QT_{ij}	BWA_{ij}	$Cond_{i(j+1)}$	Tk_i
1	63000	8200	3	-30400-8200=-38600
2	26000	6100	2	11400-6100=5300
3	50000	4100	3	30000-4100=25900
4	3500	1500	2	100000-1500=98500

Next frame:

Request

1. Firstly, number of tokens of users is updated as given in Table 3.32.

Table 3.32: Token Updates

User	Tk_i
1	-38600+4000=-34600
2	5300+4000=9300
3	25900+4000=29900
4	98500+4000=102500

2. Highest limits are decided for BW Requests as given in Table 3.33.

Table 3.33: BW_{ij1b} and BW_{ij2b} values

User	BW_{ij1b}	BW_{ij2b}
1	0.4*20000/4+0.6*(1/0.995)*8200=6944	0.02*20000+6944=7344
2	0.01*20000 + 6100=6300	0.02*20000+6300=6500
3	0.4*20000/4+0.6*(1/0.995)*4100=4472	0.02*20000+4472=4872
4	0.01*2000+1500=1700	0.01*20000+1700=1900

3. Priorities are updated as given in Table 3.34.

Table 3.34: Pr_{ij} are updated

User	QT_{ij}	$ML10Req_i$	TDR_i	Pr_{ij}
1	57000	(4690*10-6400+8200)/10=4870	11.70	1
2	22000	(6410*10-9500+6100)/10=6070	3.62	0
3	51000	(3980*10-1500+4100)/10=4240	12.03	1
4	2500	(3150*10-2500+1500)/10=3050	0.82	0

4. After that, users decides on their requests as shown in Table 3.35 by using limits at Table 3.33.

Table 3.35: BW_{ij1} and BW_{ij2} values

User	QT_{ij}	BW_{ij1}	BW_{ij2}
1	57000	6300	7100
2	22000	6200	6800
3	51000	4000	4600
4	2500	1700	2200

Granting

1. Scheduler, firstly, sorts user whose Pr_{ij}=1 with respect to their tokens in descending order. After that, it will sort other user with respect to their tokens in descending order either as shown in Table 3.36.

Table 3.36: Sorting of users

User	Tk_i	BW_{ij1}	BW_{ij2}	Pr_{ij}
3	29900	4000	4600	1
1	-34600	6300	7100	1
4	102500	1700	2200	0
2	9300	6200	6800	0

2. After that, scheduler decides on bandwidth allocation of users as shown in Table 3.37 by using the order in Table 3.36. Scheduler assigns conservative requests of users with Pr_{ij}=1 at first. After that, there is an empty space in bandwidth, greedy requests of user 1 and 3 fit in this empty space. Therefore, scheduler updates their allocation. Finally, it will pass on the others and execute the same process in same order.

Table 3.37: Assigning BWA_{ij} and $Cond_{ij}$ and Updating Tk_i

User	QT_{ij}	BWA_{ij}	$Cond_{i(j+1)}$	Tk_i
1	57000	7100	3	-34600-7100=-41700
2	22000	6200	2	9300-6200=3100
3	51000	4600	3	29900-4600=25300
4	2500	1700	2	102500-1700=100800

In the example 3, network is severely congested. As a result, some users have Pr_{ij}=1. It means that these users need help to lessen their queue length

to prevent from packet losses and excessively delayed packets. Scheduler firstly, allocates these users' requests to improve their bandwidth allocation. This is a temporary intervention to improve total goodput of system and decrease packet losses. However, this precaution can be harmful, when all users in the network are congested, because scheduler always try to expand bandwidth allocation of alarmed users. Therefore, if number of alarmed users in the network increases, total priority idea collapses and it can be harmful for the system.

In the next chapter, firstly details of simulation environment are provided. After that, simulation results are discussed.

Chapter 4

Simulation Results

In this chapter, the simulation environment and used parameters will be introduced in Section 4.1. In Section 4.2, simulation results are provided and obtained results are discussed.

4.1 Simulation Environment

In this section, firstly the simulation environment is described in Section 4.1.1. In Section 4.1.2, user's movement in simulation environment is provided. After that, computation of SNR is discussed in Section 4.1.3. In Section 4.1.4, burst profiles used in the simulation are reviewed. In Section 4.1.5, traffic model is described. In Section 4.1.6, computation of traffic load on system is explained. In Section 4.1.7, frame structure is provided. Finally, in Section 4.1.8, the other algorithms used for comparisons with the proposed algorithm.

4.1.1 WINNER 2: B1 Urban Microcell Scenario

WINNER 2 project proposed link level channel models for executing system level simulations of short range wide area wireless communication systems [30]. Following scenarios are examined in WINNER 2: indoor small office, large indoor hall, indoor-to-outdoor, urban micro-cell, bad urban micro-cell, outdoor-to-indoor, stationary feeder, suburban macro-cell, urban macro-cell, rural macro-cell, and rural moving networks [30]. Both non-light-of-sight (NLOS) and line-of-sight (LOS) conditions are considered in WINNER 2 channel models.

In our simulations, B1 model of WINNER 2 is used. B1 is a urban microcell scenario. Antennas are assumed to be below the tops of surrounding buildings. Streets are laid out in a Manhattan-like grid. LOS condition is valid for users who directly see the BS, however LOS can temporarily blocked by traffic or buildings.

The path loss computation for B1 is as follows:

The free space path loss is given by

$$PL_{free} = 46.4 + 20log_{10}(d[m]) + 20log_{10}(f[GHz]/5) \tag{4.1}$$

The path loss expressions and standard deviation of the log-normal shadow fading for LOS and NLOS conditions are given in Table 4.1. The calculations of the distance parameters d_1 and d_2 in this model are depicted in Figure 4.1

Table 4.1: Path Loss Parameters for B1

Scenario	Path Loss [dB]	Shadow Fading std(dB)	applicablity range and antenna height values
LOS	PL_{LOS} = max(22.7 $\log_{10}$ (d_1[m]) + 41.0 + 20 $\log_{10}$(f[GHz]/5.0),PL_{Free})	σ=3	$30m < d_1 < d'_{BP}$ $h_{BS} = 10m$ $h_{MS} = 1.5m$
	PL_{LOS} =40.0 $\log_{10}$(d_1 [m]) + 9.45 -17.3 $\log_{10}$($h'_{BS}[m]$) - 17.3 $\log_{10}$($h'_{MS}[m]$) + 2.7 $\log_{10}$(f[GHz]/5.0)	$\sigma = 3$	$d'_{BP} < d_1 < 5km$
NLOS	$PL_{NLOS} = PL_{LOS}$(d1) + 20 -12.5· n_j $10n_j \cdot \log_{10}(d_2[m])$ where n_j = max((2.8 - 0.0024d_1[m]),1.84)	$\sigma = 4$	$10m < d_1 < 5km$ $w/2 < d_2 < 2km$ w=20m h_{BS} =10 m h_{MS} = 1.5 m

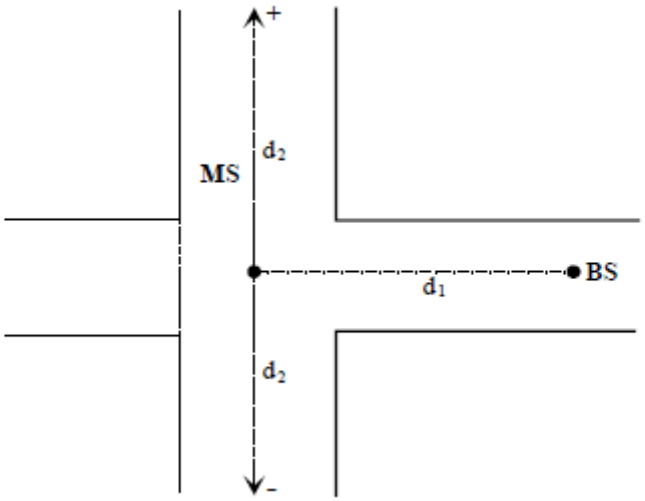

Figure 4.1: Geometry for d_1 and d_2 path-loss model

4.1.2 Displacement of users

We use a simulation area given by a 4x5 Manhattan-like grid given in Figure 4.2. Base station is at the center of the simulation environment. Only users on X1 are in LOS condition. Total simulation area is $1km^2$ however users can only move in 9 streets. Users wander in streets freely, when they come to crosses, they can go to any direction with equal probability. The probabilities of moving in different directions are depicted in Figure 4.3

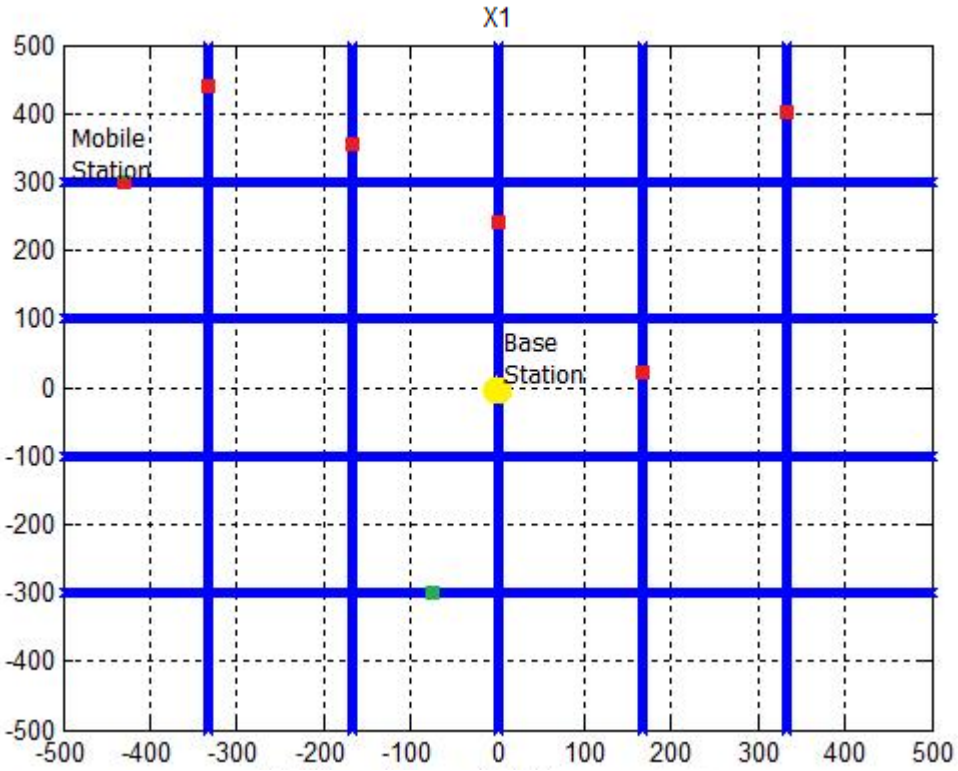

Figure 4.2: Simulation Environment

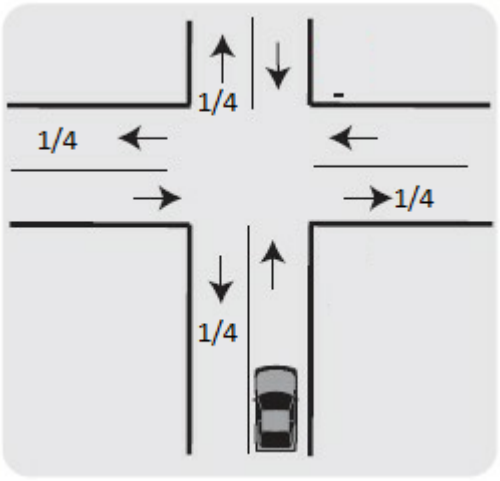

Figure 4.3: Cross Decision

Users cannot leave the simulation environment, therefore if they arrive at the edge of the area, they have to return back. Therefore total number of users in the network stays constant during the simulation.

Velocity of a user stays constant when the user moves over a straight line. The velocity may change when users arrive at corners or edges. After a decision made on a new direction, the new velocity of the user is chosen randomly. There are 3 different types of users in the system: slow, moderate, fast. Velocity of slow users are uniformly distributed between 0 km/h and 50km/h, velocity of moderate users are uniformly distributed between 0km/h and 100km/h, velocity of fast users are uniformly distributed between 50km/h and 100km/h.

4.1.3 Computation of Signal to Noise Ratio

Signal to Noise Ratio (SNR) is a measure of the ratio of the power of the desired signal to the power of background noise. In wireless communication networks, SNR and error probability are inversely proportional.

In order to calculate the SNR of a user, firstly, power of the received signal must be calculated. Equation (4.2) is used for that purpose:

$$P_r(dBm) = P_t(dBm) + G_t(dB) + G_r(dB) - PL(dB) \tag{4.2}$$

where P_r is the received power at the receiver, P_t is the transmission power, whereas G_t and G_r are transmitter and receiver antenna gains, respectively. After calculating the power at the receiver, SNR is calculated as:

$$SNR(dB) = P_r(dBm) + N_p(dB) \tag{4.3}$$

$$N_p = 10log_{10}(kTB) \tag{4.4}$$

where N_p is noise power, k is Boltzman constant which is 1.38x10^{-23} JK^{-1}, T is temperature in Kelvin (which is taken as 298 °K in this thesis) and B is channel bandwidth (which is taken as 40MHz in this thesis).

In our simulations, following parameters are used: P_t =34.771 dBm, G_t = 10 dB, G_r = 5 dB. PL is calculated from Winner 2 B1 model. From (4.4), the noise power N_p is obtained as -97.8 dBm at B=40MHz.

There are two types of fading in radio systems: fast and slow. The reason of fast fading is the multipath propagation. The other type of fading is slow fading

(i.e. shadow fading), which is caused by an obstacle on the way between mobile and base station. In the literature, shadowing fading is modeled using a one dimensional log-normal distribution. From the geometry of the channel between the MS and BS, the shadow fading values experienced by a user as time evolves are correlated. For shadow fading correlation, Gudmundson's time correlation model is used [31]. Time-correlation functions provide a statistical information about time dependent evolution of the signal. Proposed algorithm by Gudmundson achieved very good results in suburban and urban environments. The details of the Gudmundson's model are described below.

Signal strength is measured in mobile communication systems at regular intervals, therefore a discrete-time model is chosen. Logarithm of the received signal strength can be shown as $A(n)$. Assume that $A(n)$ has a Gaussian distribution and average distance between MS and BS is [31] :

$$\tau = P_t - K_2 \cdot log(d) \tag{4.5}$$

In Equation (4.5), P_t is transmitted power, K_2 is a propagation constant between 20 and 60 dB. This is taken from Okumura's model for large scale average of received signal strength in mobile radio systems [31].

As the correlation function the following decreasing function is used [31]:

$$R_A(k) = \sigma^2 a^{|k|} \tag{4.6}$$

$$a = \varepsilon_D^{v \cdot \psi / D} \tag{4.7}$$

In Equation (4.6), a is the correlation coefficient, it may be expressed as in Equation (4.7). $R_A(k)$ is the correlation between two points which are separated with distance D. The variance σ^2 is chosen between 3 dB and 10 dB. In Equation (4.7), v is the velocity of mobile. ψ is the sampling interval that is chosen 0.5s. ε_D is the correlation between two points.

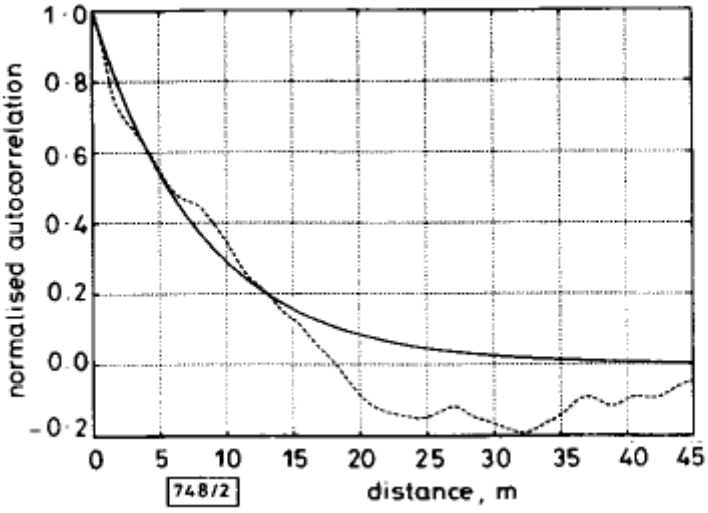

Figure 4.4: Normalized autocorrelation (measured and fitted) in urban environment

In Figure 4.4, measured autocorrelation is shown by solid line and fitted autocorrelation is shown by dotted line. After 20m, autocorrelation is close to zero and two variable acts like an independent variable. In our simulations, we recalculate each users shadow fading every 5m because they are still highly correlated with each other since the correlation coefficient is 0.5477 at 5m.

4.1.4 Burst Profiles

Following burst profiles are used in our simulations: 64-QAM ¾ , 64-QAM ⅔, 16-QAM ¾ 16-QAM ½ QPSK ¾ QPSK ½. In our simulations, network is assumed to be error free to simplify the simulations. Therefore, for each frame the highest profile is selected for each MS such that the measured SNR is sufficient to provide a BER of less than 10^{-3} [32]. The SNRs required in order to allow this BER for each burst profile are listed in Table 4.2.

Table 4.2: SNR required for consider burst profiles

Burst Profile	SNR Required(dB)
QPSK ½	3.5
QPSK ¾	6.5
16-QAM ½	9.0
16-QAM ¾	12.5
64-QAM ⅔	16.5
64-QAM ¾	18.5

If SNR of user is over 18.5dB, user uses the highest profile (64-QAM ¾). If SNR of user is less than 3.5dB, user cannot send data.

4.1.5 Traffic Model

During our simulations, each user is active with probability p. In our simulations, we used p=0.4 and p=0.8. The traffic arrival process at each MS is modeled as an on-off Markov modulated Poisson process [33].

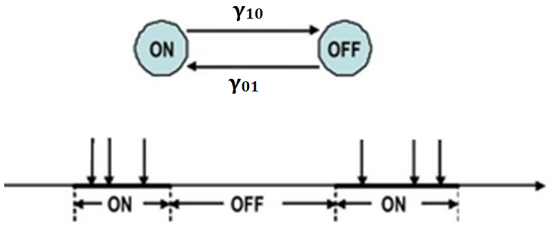

Figure 4.5: On-Off Markov Modulated Poisson Model

In the on state, packets are arriving according to a Poisson process with rate λ. In the off state, no packets are arriving. The rate of transitions between On to Off and Off to On are γ_{10} and γ_{01}, respectively. To obtain $p = 0.4$, γ_{10} is equal to 0.03 and γ_{01} is 0.02. To obtain $p = 0.8$, γ_{10} is selected as 0.01, and γ_{01} is equal to 0.04.

4.1.6 Traffic Load of Network

The average traffic load is given as

$$\rho = \frac{\lambda \cdot Np \cdot AP_{size}}{B \cdot AC} \tag{4.8}$$

where AC is the average spectral efficiency of users and AP_{size} is average packet size. In simulation, 40% of packets are assumed to be 64 bytes long (corresponding to acknowledgment packets). The rest of the packets are uniformly distributed between 65 bytes and 750 bytes. The resulting average packet size is given as:

$$AP_{size} = 0.4 * 64 + 0.6 * (65 + 750)/2 = 270.1 bytes \tag{4.9}$$

4.1.7 Frame Size

Size of each frame is 5ms.

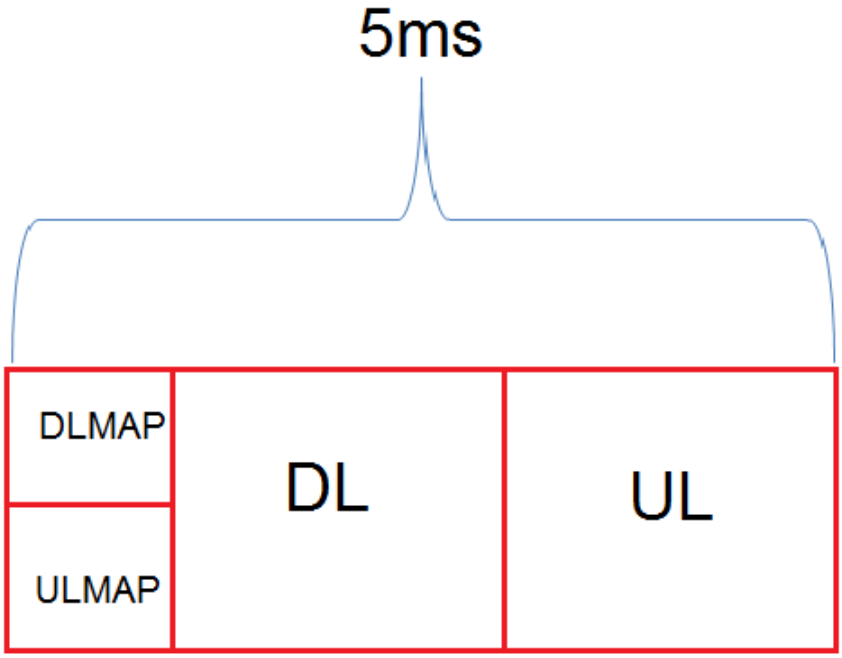

Figure 4.6: Frame Model

As shown in Figure 4.6, 20% of each frame is reserved for Uplink and Downlink scheduling MAP. Half of the remaining part of each frame is reserved for downlink scheduling and the other half is reserved for uplink scheduling communication.

4.1.8 Compared Algorithms

In our simulations, 8 different algorithms are for comparison:

1. Fully Opportunistic Downlink Scheduling Algorithm (OpporDL): as described in Section 2.4.1.

2. Opportunistic Uplink Scheduling Algorithm (OpporUL): its request part is exactly same as the proposed algorithm without priority. In granting, scheduler at BS sorts users with respect to their burst profiles in descending order.

3. Uplink Scheduling Algorithm for delay sensitive traffic (USFDST): as described in Chapter 3.

4. Uplink Scheduling Algorithm for delay sensitive traffic without priority (USFDST-NP): it is the same as the USFDST algorithm, however there is no priority mechanism.

5. Uplink Scheduling Algorithm for delay sensitive traffic (USFDST-DL): it is the same as the USFDST algorithm except that the scheduler has access to MS queue lengths so that if there is an empty space after USFDST scheduling, round robin scheduling algorithm is used to fill the empty space.

6. Round Robin Downlink Scheduling Algorithm (RR-DL) as explained in Section 2.4.2.

7. Max-min Fairness Downlink Scheduling Algorithm (Max-MinF) as explained in Section 2.4.3.

8. Proportional Fairness Downlink Scheduling Algorithm (PropFair) as explained in Section 2.4.4.

It is important to note that the downlink scheduling algorithms (1, 5, 6, 7, 8) have full access to information about MS queues whereas uplink scheduling algorithms (2, 3, 4) have only partial information about queue states.

4.2 Simulation Results

In the simulations, simulation time is set to 50 minutes corresponding to 600,000 frames. There are 10 users in the network: 3 slow, 4 moderate and 3 fast. 5 simulations have been carried out for each traffic load. 9 different average traffic loads are used for simulations between 0.5 and 0.8. Each user's queue is limited to 100 packets, any new packet is discarded and counted as lost if it arrives to a full queue. In addition to buffer overflows, any packet with an access delay exceeding 200ms is also assumed lost from the application point of view since there is a maximum delay requirement for delay sensitive traffic. We repeat our simulations for two different values of p: $p = 0.4$ and 0.8.

For delay sensitive networks, as long as the delay is not over the maximum latency, its distribution is not critical. Therefore, USFDST algorithm tries to decrease the total number of packets with no delay, meanwhile it tries to decrease the number of packets which is over the maximum allowed latency. In the following figures, distributions of delay of packets are provided for $p = 0.8$ and traffic load of $\rho = 0.8$.

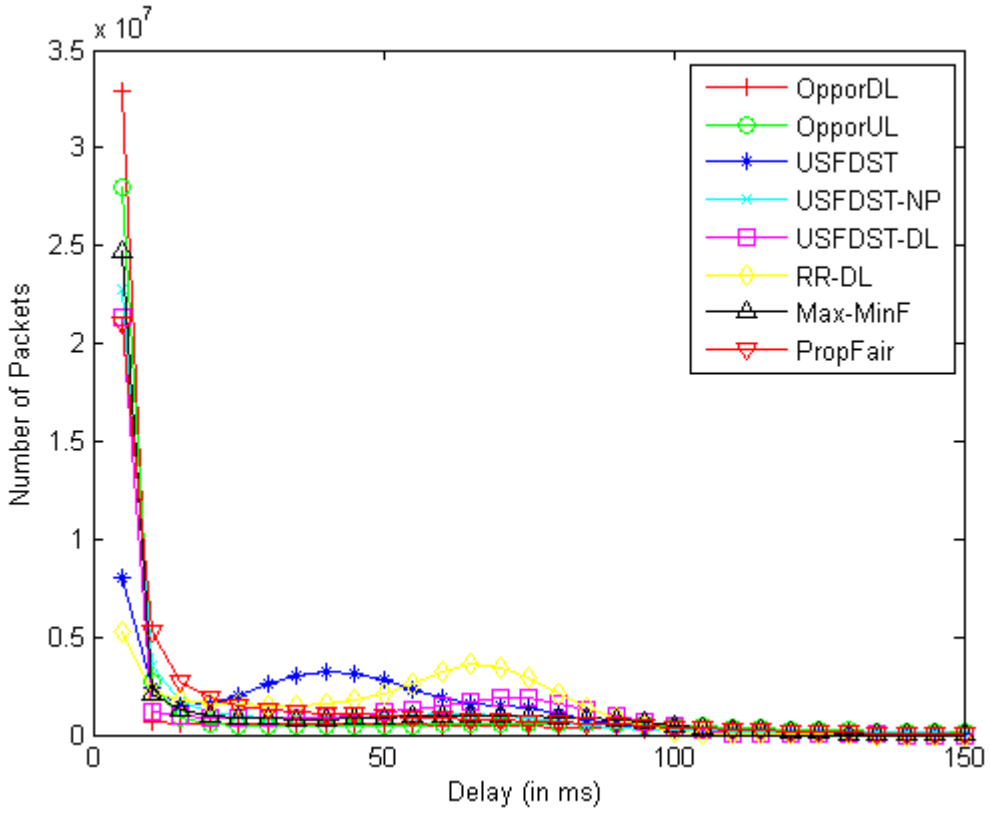

Figure 4.7: Histogram of Delay between 0 ms and 150ms

When we look at the Figure 4.7, USFDST algorithm has a very small number of packets with 0 delays. However between 25ms and 70ms, there is a significant increase for USFDST. The delay histogram can be seen in more detail in Figure 4.8.

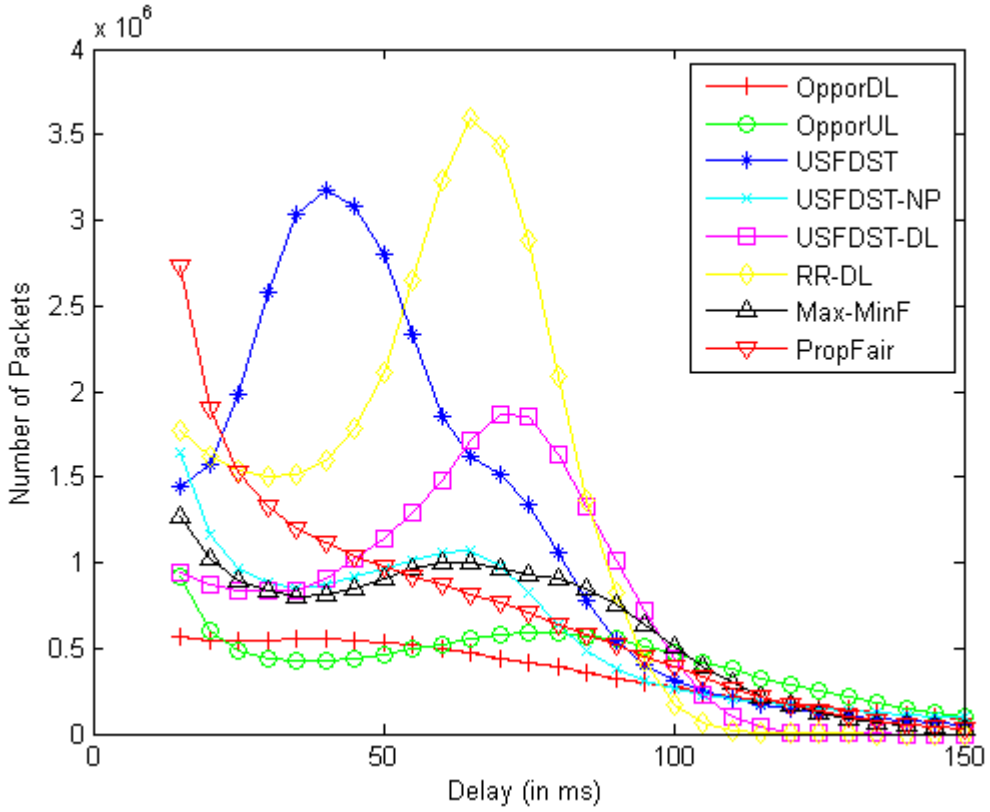

Figure 4.8: Histogram of Delay between 15 ms and 150ms

Figure 4.8 shows that limits of this increase is near the BD_L and BD_H. Therefore, these values can be adjusted according to the delay requirements of the application. The downlink version USFDST-DL has no packets exceeding a delay of about 120ms.

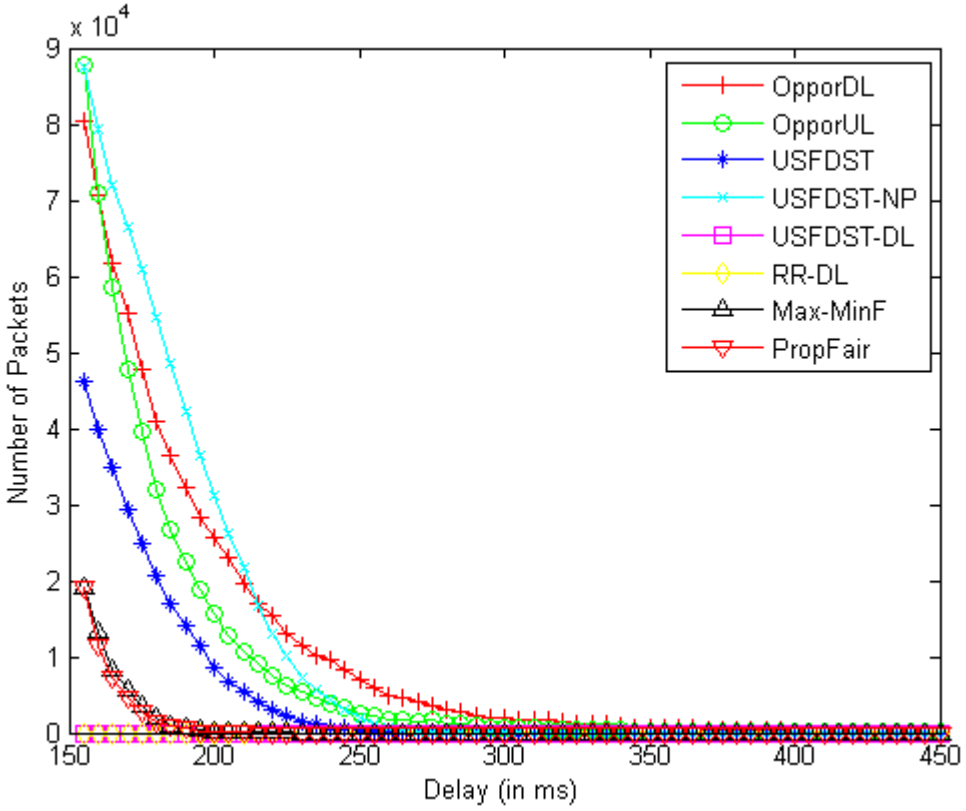

Figure 4.9: Histogram of Delay between 150 ms and 450ms

The histogram of the delay is depicted in Figure 4.9 for the range of delays between 150ms and 450ms. USFDST is the fifth best algorithm to decrease number of packets with excessive delays coming after RR-DL, USFDST-DL, Max-MinF and Proportional Fairness. For opportunistic algorithms, results are not very good as there is a significant number of packets which are sent with excessively delays.

4.2.1 Results for $p = 0.8$

Average spectral efficiency, in bits/sec/Hz, is used to measure spectral efficiency of the system. For the burst profiles used in this thesis, the spectral efficiency

ranges between 1 (for QPSK 1/2) and 4.5 (for 64-QAM 3/4). Average spectral efficiency in a frame, ASE, is calculated by Equation (4.10):

$$ASE(j) = TSB(j) / \sum_{i=1}^{N} BWA_{ij} \tag{4.10}$$

In Equation (4.10), ASE(j) is the average spectral efficiency of the network in frame j. TSB(j) is the total number of transmitted bits in frame j and BWA_{ij} is the total resources assigned to user i in the frequency domain (in units of sec*Hz) in frame j.

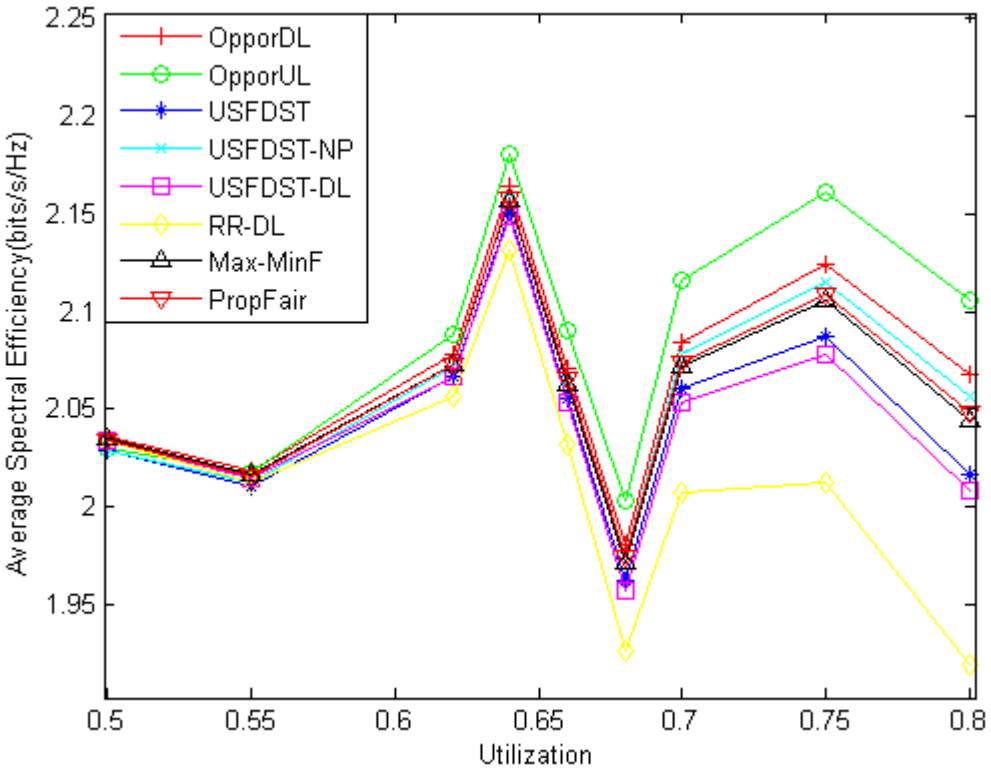

Figure 4.10: Average Spectral Efficiency (bits/s/Hz) vs Utilization

In Figure 4.10, the average spectral efficiency of each scheduling algorithm is close to each other for low traffic load cases. As expected, average spectral efficiencies of opportunistic algorithms are higher than others, especially at higher

traffic loads, whereas round-robin scheduling has the lowest ASE. USFDST and proportional fair scheduling algorithms lie in the middle.

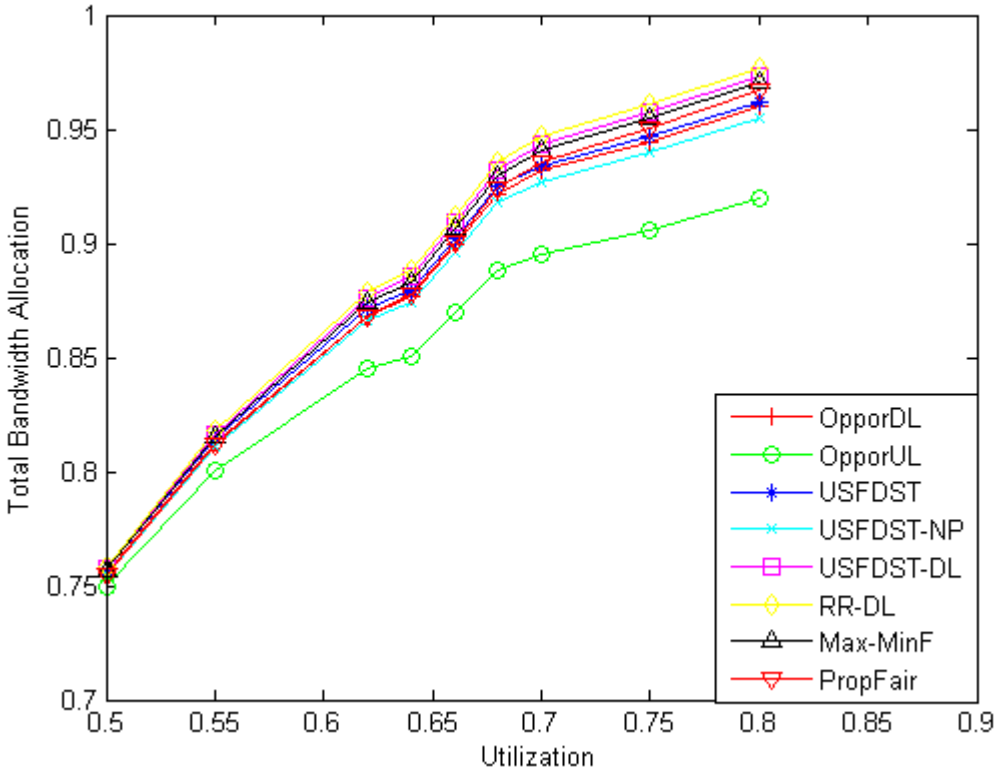

Figure 4.11: Average Resource Allocation

For uplink scheduling algorithms, filling total bandwidth area is more difficult than downlink scheduling algorithms because the scheduler does not have complete queue information of MSs. The average resource allocations achieved by different scheduling algorithms are plotted in Figure 4.11. We observe that USFDST can fill the total bandwidth close to the downlink scheduling algorithms although USFDST scheduler makes use of only partial information about queue states of MSs. In high traffic load cases, the difference between USFDST and the best filling algorithm (i.e. RR-DL) is less than 2%. The data for this figure is also tabulated in Table 4.3.

Table 4.3: Average Resource Allocation

	0.5	0.55	0.6	0.62	0.64	0.68	0.7	0.75	0.8
OpporDL	0.755	0.812	0.868	0.876	0.899	0.921	0.931	0.944	0.959
OpporUL	0.749	0.800	0.845	0.850	0.869	0.888	0.894	0.905	0.919
USFDST	0.756	0.814	0.871	0.879	0.902	0.924	0.934	0.946	0.961
USFDST-NP	0.754	0.811	0.865	0.874	0.896	0.917	0.926	0.939	0.954
USFDST-DL	0.757	0.816	0.876	0.885	0.908	0.932	0.943	0.957	0.973
RR	0.758	0.817	0.878	0.887	0.911	0.935	0.947	0.961	0.976
Max-MinF	0.757	0.815	0.873	0.883	0.906	0.929	0.940	0.954	0.970
PropFair	0.757	0.814	0.870	0.879	0.904	0.929	0.939	0.954	0.970

As USFDST algorithm is developed for delay sensitive applications, improving total goodput of the system is the main objective of the algorithm without disturbing the short and long term fairness. In Figure 4.12, total goodput of the system is plotted for all algorithms considered. Goodputs are close to each other for low traffic loads (i.e. between 0.5 and 0.65). Since user's demands are low at these loads, missed scheduling opportunities can be easily compensated in the following frames. At higher loads, there are more differences between goodputs, but the differences are still within 5%.

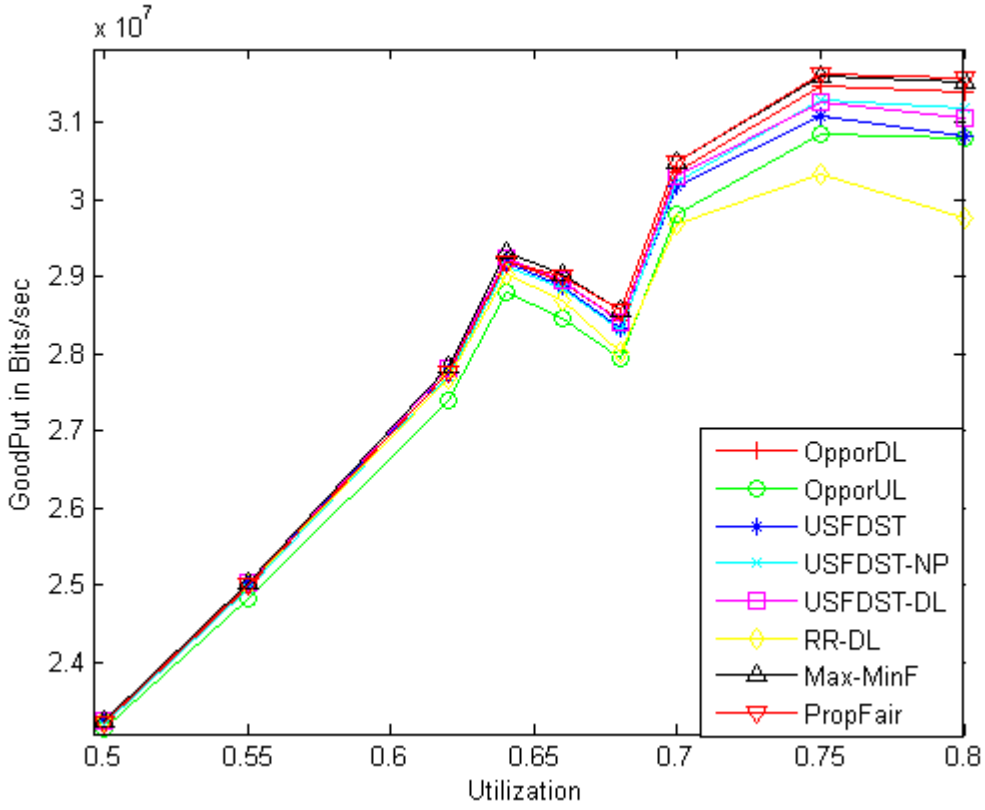

Figure 4.12: Goodput vs. Traffic Load of Network

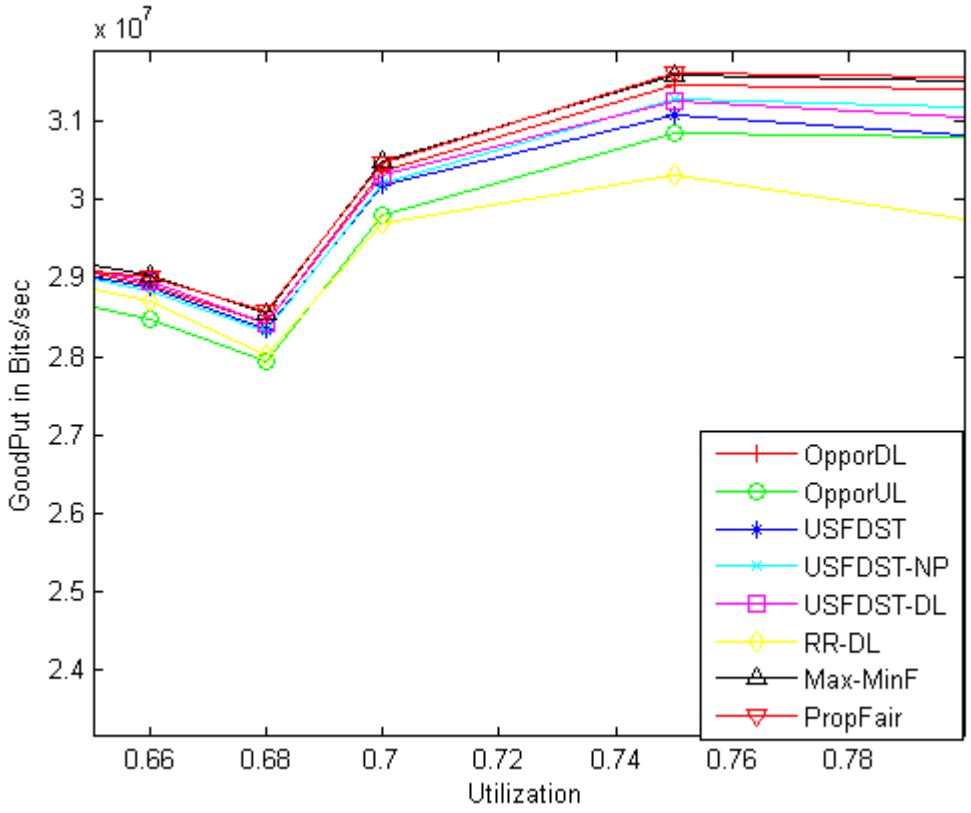

Figure 4.13: Details of the high traffic load cases

In Figure 4.13, high traffic load cases are shown in more detail. When high traffic load cases are examined, Max-MinF and proportional fair scheduling are the best scheduling algorithm to improve total goodput of the network. It is followed by OpporDL and USFDST-NP. However USFDST can reach 97% of the total goodput of these downlink algorithms. USFDST works worse than USFDST-NP, because of priority mechanism starts to affect system negatively for high traffic load cases as more users are in the alarmed state more often.

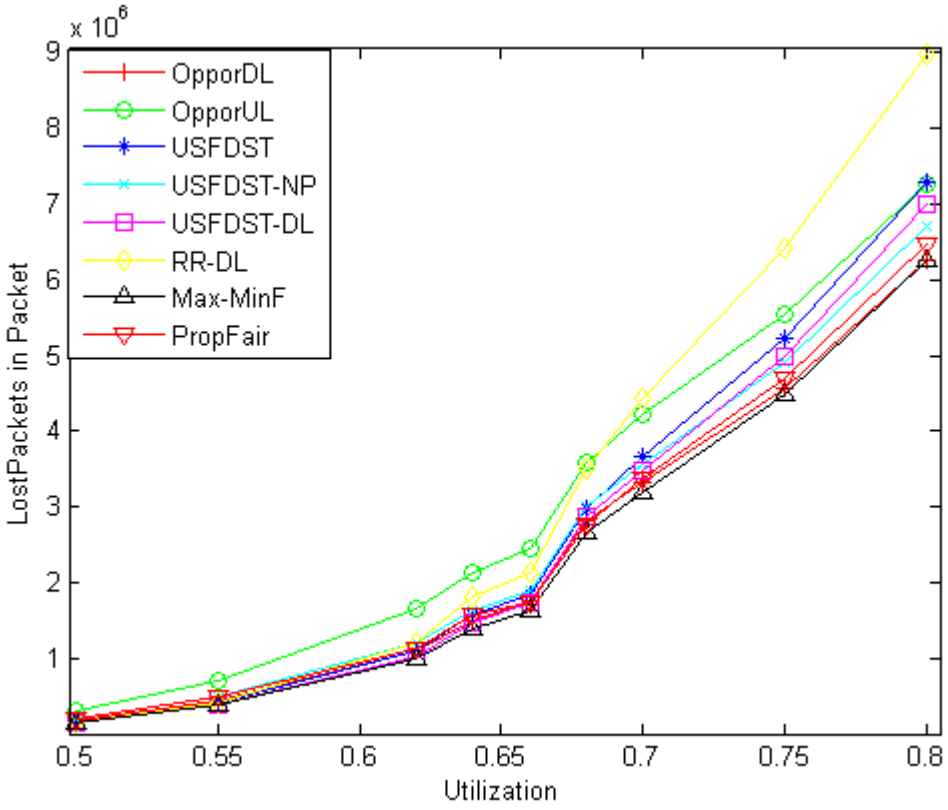

Figure 4.14: Total Number of Lost Packets

Figure 4.14 shows utilization vs. total number of lost packets in the network. Goodput and total number of lost packets are inversely proportional as expected. When traffic load is increasing, total number of lost packet is also increasing. There is an exponential increase at packet losses after traffic load is above 0.65 because the congestion of the network becomes more severe and it causes more packet losses.

Since users can be offline for some durations, we propose a new short term fairness metric. In this fairness metric, system discards offline users by only counting the packets which are queued before the duration and arrive during the duration. This fairness metric is provided in the Equation (4.11):

$$FI(\Delta) = \frac{(\sum_{i=1}^{U} FX_i)^2}{U * \sum_{i=1}^{U} FX_i^2} \tag{4.11}$$

$$FX_i(\Delta) = \frac{TSB_i(t, t+\Delta)}{NIB_i(t, t+\Delta) + QP_i(t)} \tag{4.12}$$

In Equation (4.12), U is the number of online users, Δ is the short-term fairness horizon (taken as 50 frames in this thesis). $\mathrm{TSB}_i(t,t+\Delta)$ is total transmitted bits in the interval $(t,t+\Delta)$ by user i, $\mathrm{NIB}_i(t,t+\Delta)$ is the number of arrived bits in interval $(t,t+\Delta)$ to user i, and $\mathrm{QP}_i(t)$ is the number of total bits in the queue of user i at time t. Therefore, only users with demands within the time period are taken into account in calculating the short term fairness.

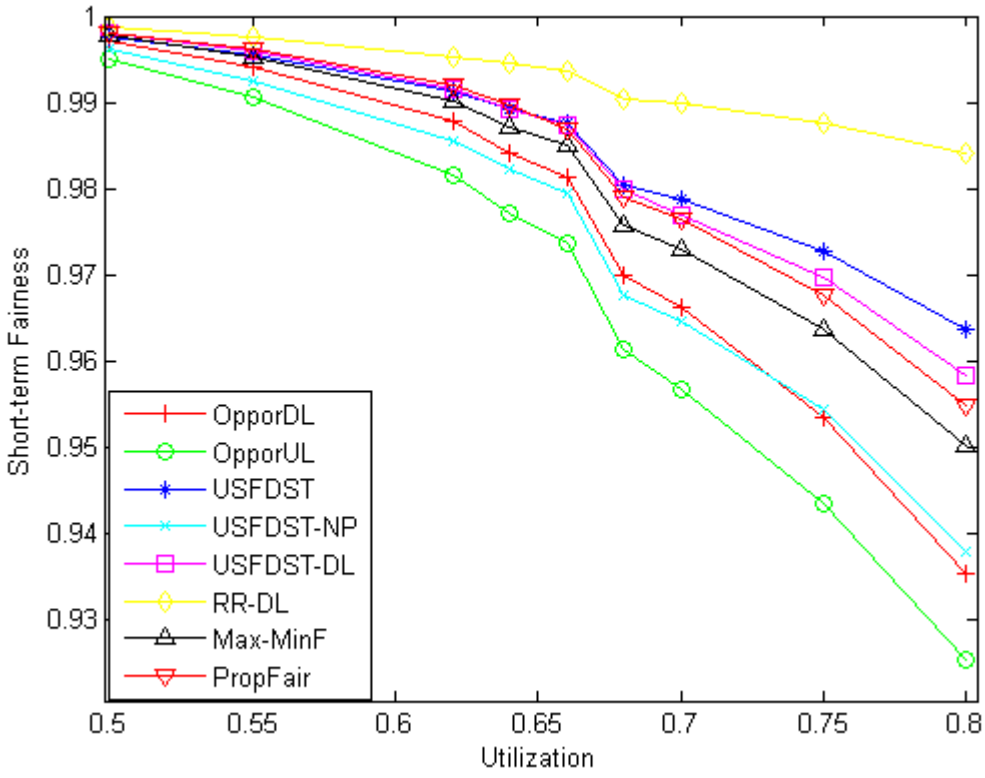

Figure 4.15: Short Term fairness of Algorithms

Figure 4.15 shows short-term fairness of the algorithms. USFDST algorithm is the best algorithm in sharing bandwidth fairly among users except RR-DL. USFDST-NP starts to sacrifice from fairness among users at high traffic load. If bandwidth is shared among users equally, users with low transmission rates can send fewer amount of data in the same interval. Therefore, in congested networks, queue of these users starts to expand. Priority mechanism increases bandwidth allocation of these users, and it improves the short-term fairness, while slightly sacrificing from the total goodput.

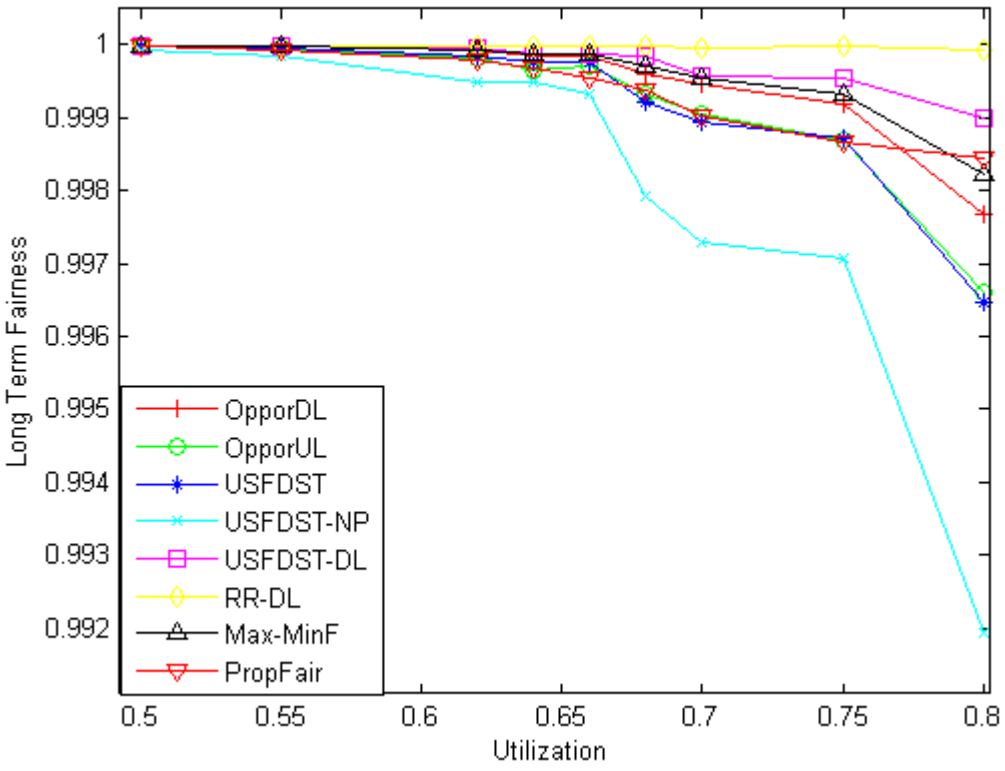

Figure 4.16: Long Term fairness of Algorithms

Figure 4.16 shows long term fairness vs. utilization of the network. Simulation time is long enough, therefore, there is not significant differences between

long term fairness of the algorithms. Each algorithm achieves a very high long-term fairness.

4.2.2 Results for $p = 0.4$

The simulation based comparative study is repeated in the same environment now with $p = 0.4$ where users are less active.

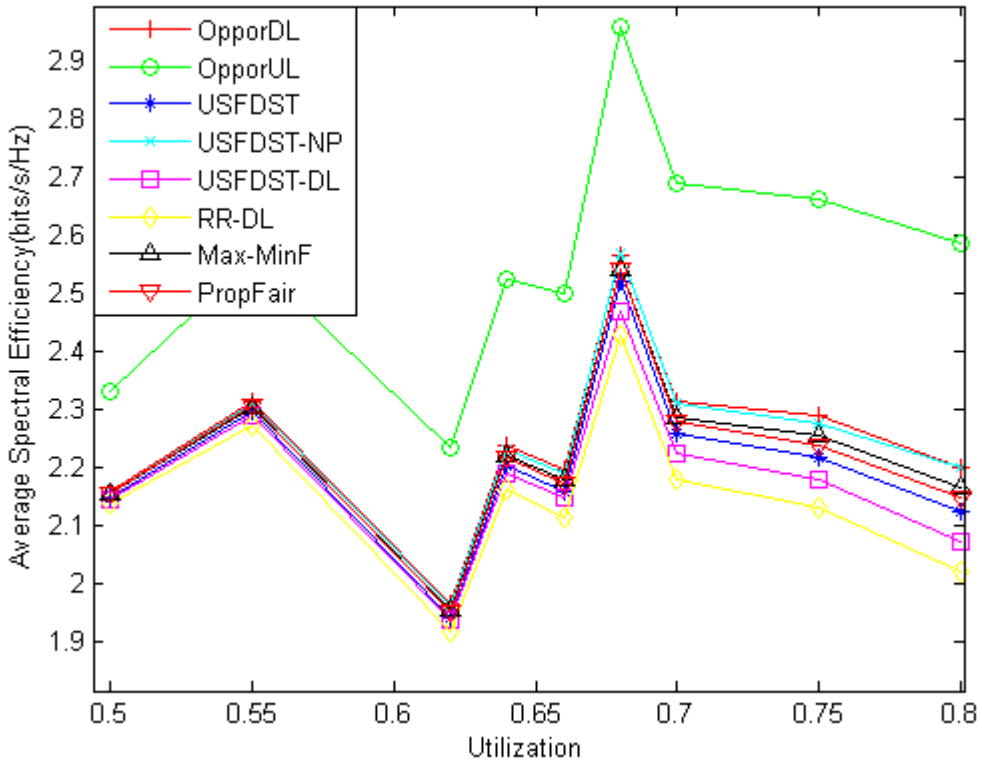

Figure 4.17: Average Spectral Efficiency(bits/s/Hz) vs Utilization

In Figure 4.17, average spectral efficiency of each scheduling algorithm is close to each other for low traffic load cases except OpporUL. Average spectral efficiency of OpporUL is remarkably higher than other algorithms. It means that OpporUL favors user with higher transmission rates in scheduling. In general,

opportunistic scheduling algorithms achieve highest average spectral efficiency. Similar to $p = 0.8$, round-robin scheduling has the lowest spectral efficiency whereas USFDST lies in the middle.

When Figures 4.10 and 4.17 are compared, average spectral efficiency is greater for $p = 0.4$. The reason is that λ for $p = 0.4$ is greater than λ for $p = 0.8$. When user is in the on state, queues of users are expanding more quickly for $p = 0.4$. Therefore, users lose more packets, when their transmission rates are low (i.e. when they are using lower burst profiles) when $p = 0.4$. Consequently, the average spectral efficiency for $p = 0.4$ is greater than $p = 0.8$ as users with lower burst profiles have reduced traffic due to buffer overflows.

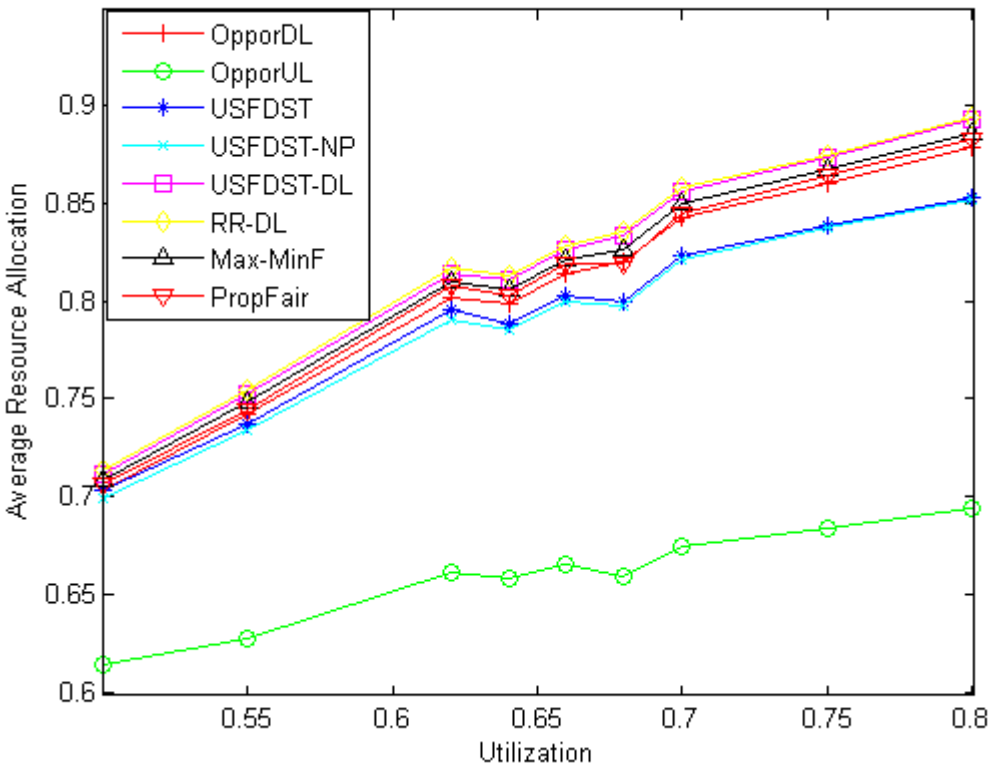

Figure 4.18: Average Resource Allocation

In Figure 4.18, the average amount of resources allocated to MSs is plotted. We observe that USFDST can fill the total area close to the downlink scheduling algorithms in low traffic load cases. In high traffic load cases, the difference increases, however it is not more than 5%. The data for this figure is also tabulated in the Table 4.4.

Table 4.4: Average Resource Allocation

	0.5	0.55	0.6	0.62	0.64	0.68	0.7	0.75	0.8
OpporDL	0.703	0.743	0.801	0.798	0.814	0.820	0.842	0.860	0.879
OpporUL	0.614	0.628	0.662	0.659	0.665	0.660	0.675	0.684	0.694
USFDST	0.703	0.737	0.795	0.789	0.803	0.799	0.823	0.838	0.853
USFDST-NP	0.699	0.734	0.791	0.786	0.800	0.798	0.821	0.837	0.851
USFDST-DL	0.712	0.752	0.813	0.811	0.826	0.833	0.856	0.873	0.892
RR	0.713	0.755	0.817	0.813	0.829	0.835	0.858	0.875	0.894
Max-MinF	0.708	0.749	0.809	0.806	0.821	0.827	0.850	0.867	0.886
PropFair	0.706	0.744	0.808	0.802	0.819	0.819	0.845	0.864	0.882

When Figures 4.11 and 4.18 are compared, we observe that USFDST performs more effectively, when users are more active in the system. When $p = 0.4$, users arrival rate (λ) increases, and the correlation between current queues of users and previous bandwidth requests are weaker. Therefore, USFDST performs worse for $p = 0.4$.

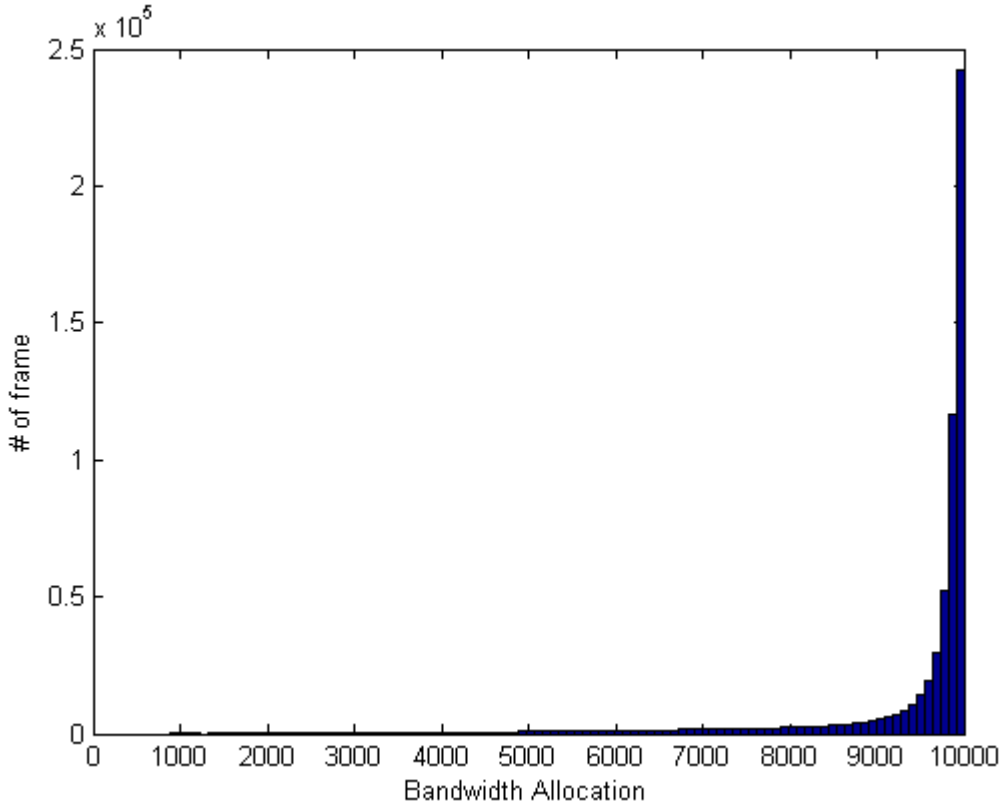

Figure 4.19: Bandwidth Allocation of USFDST for Traffic Load is 0.75 and $p = 0.8$

In Figure 4.19, bandwidth allocation of USFDST is plotted of $\rho = 0.75$ traffic load and $p = 0.8$. We observe that for 78% of the frames, USFDST algorithm can fill more than 95% of the bandwidth.

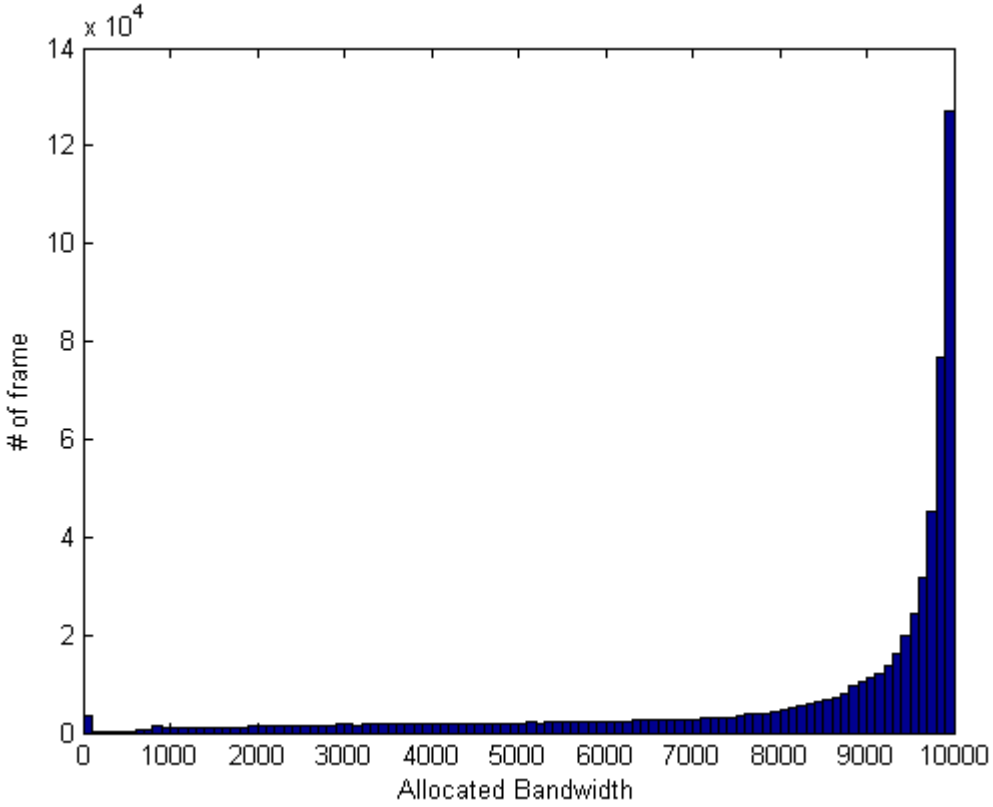

Figure 4.20: Bandwidth Allocation of USFDST for Traffic Load is 0.75 and $p = 0.4$

Figure 4.20 shows bandwidth allocation of USFDST when traffic load $\rho = 0.75$ and $p = 0.4$. We observe that for 51% of the frames, USFDST algorithm can fill more than 95% of the bandwidth. Therefore, USFDST performs more effectively for larger p's since correlation between current queues of users and previous bandwidth request are more stronger which is crucial for USFDST algorithm.

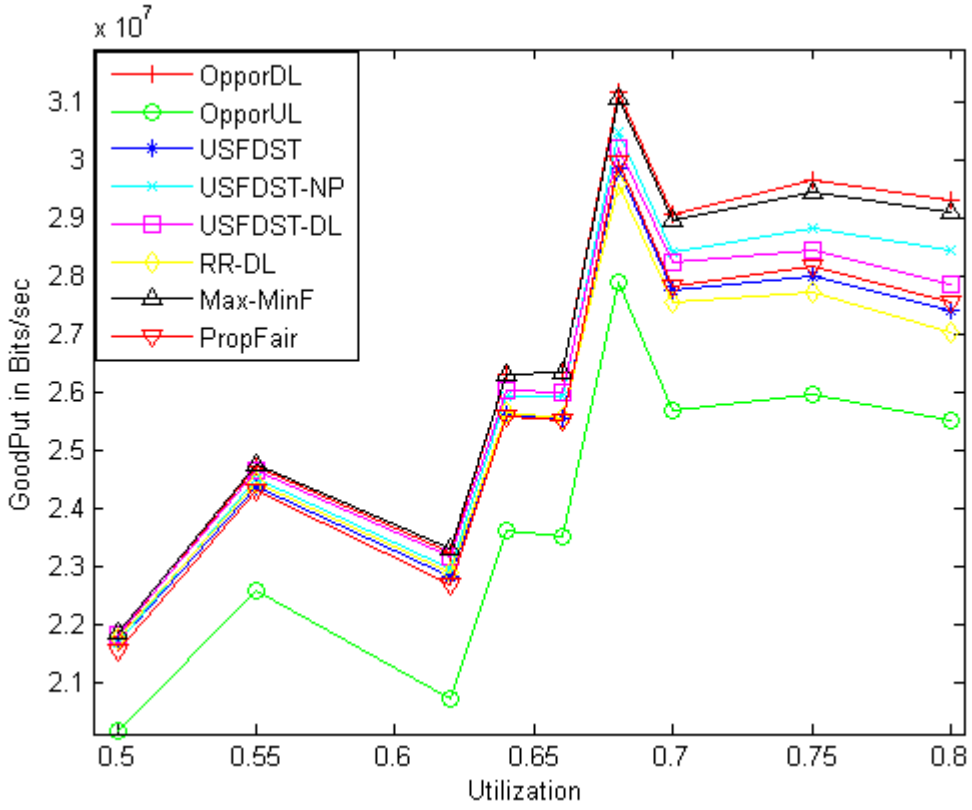

Figure 4.21: Goodput vs. Traffic Load of Network

In Figure 4.21, total goodput of the system is close to each other for low traffic loads (i.e. between 0.5 and 0.65) except OpporUL. In low traffic load cases, missed scheduling opportunities can be easily compensated at the following frames. However as the traffic load increases, differences are more distinctive.

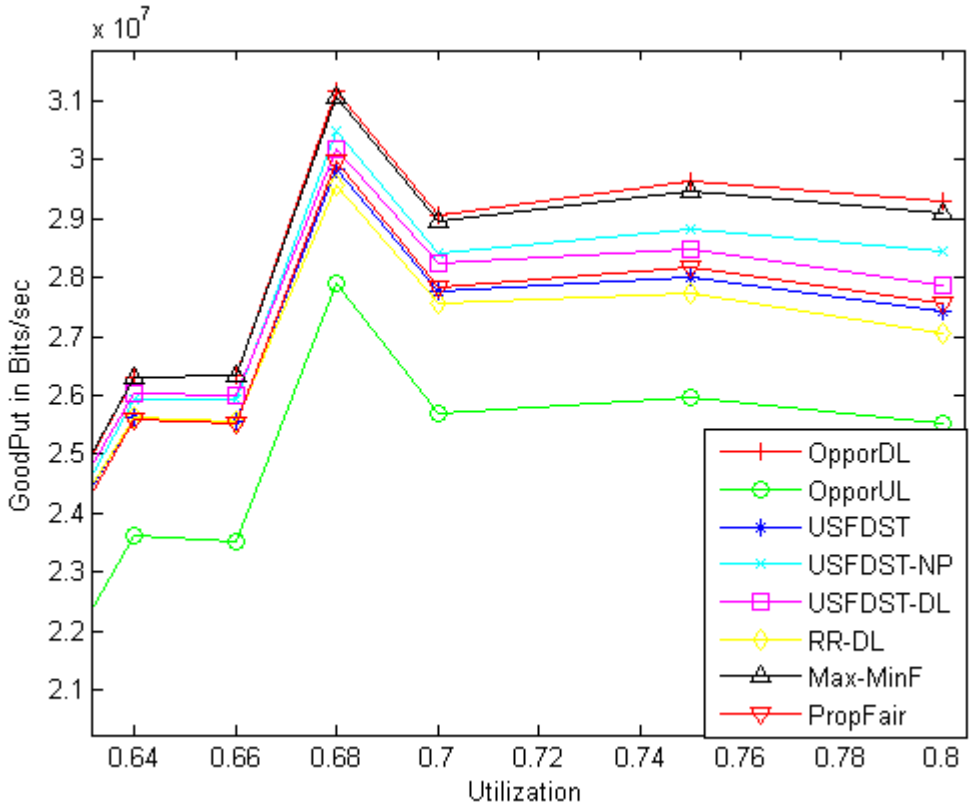

Figure 4.22: Details of the high traffic load cases

In Figure 4.22, goodput is plotted for traffic loads between 0.65 and 0.8. OpporDL and Max-MinF perform as the best scheduling algorithms in terms of total goodput of the network. However, USFDST can reach 97% of the total goodput of these algorithms although it does not have full access to information about MS queues which is available to downlink scheduling algorithms. USFDST works worse than USFDST-NP, which is the same algorithm without priority, because priority mechanism starts to affect system negatively for high traffic load cases. Users are in the alarmed state more often in high traffic load cases. As explained before, if there are more alarmed users simultaneously, this can degrade the efficiency.

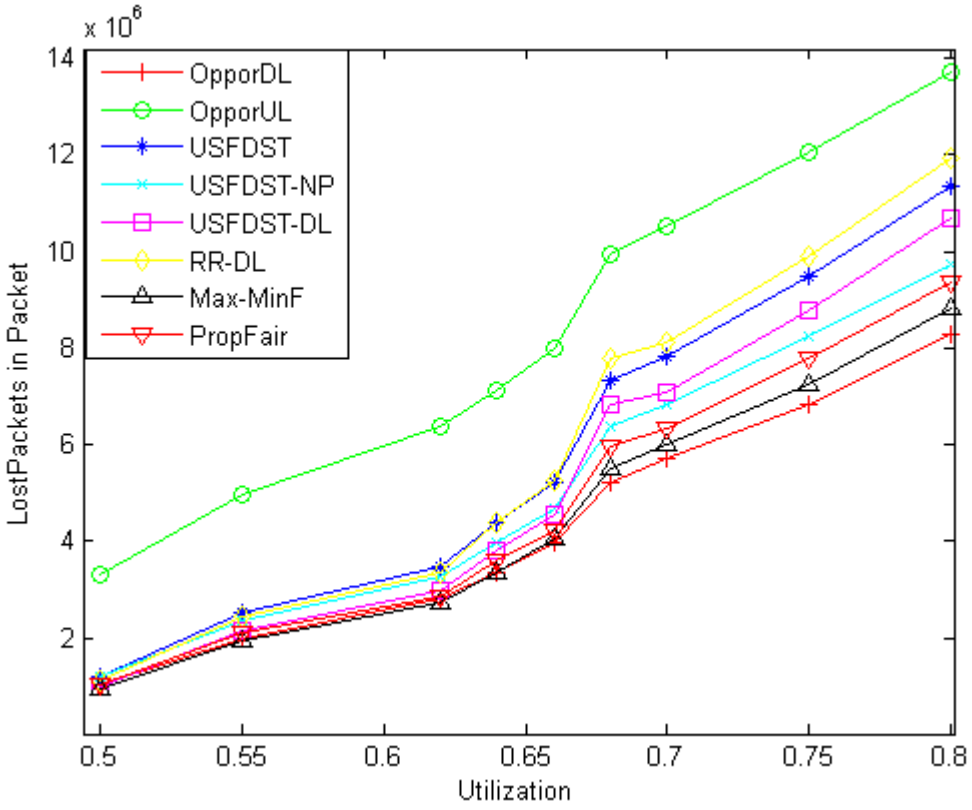

Figure 4.23: Total Number of Lost Packets

Figure 4.23 shows utilization vs. total number of lost packets in the network. Goodput and total number of lost packets are inversely proportional as expected. When traffic load is increasing, total number of lost packets is also increasing. There is an exponential increase in packet losses after traffic load exceeds 0.65, because congestion of the network becomes more severe and it causes more packet losses.

When Figure 4.14 and Figure 4.23 are compared, total number of lost packets for $p = 0.4$ is 77% more than total number of lost packets for $p = 0.8$ because packet arrivals are more bursty. Therefore, network become congested more quickly.

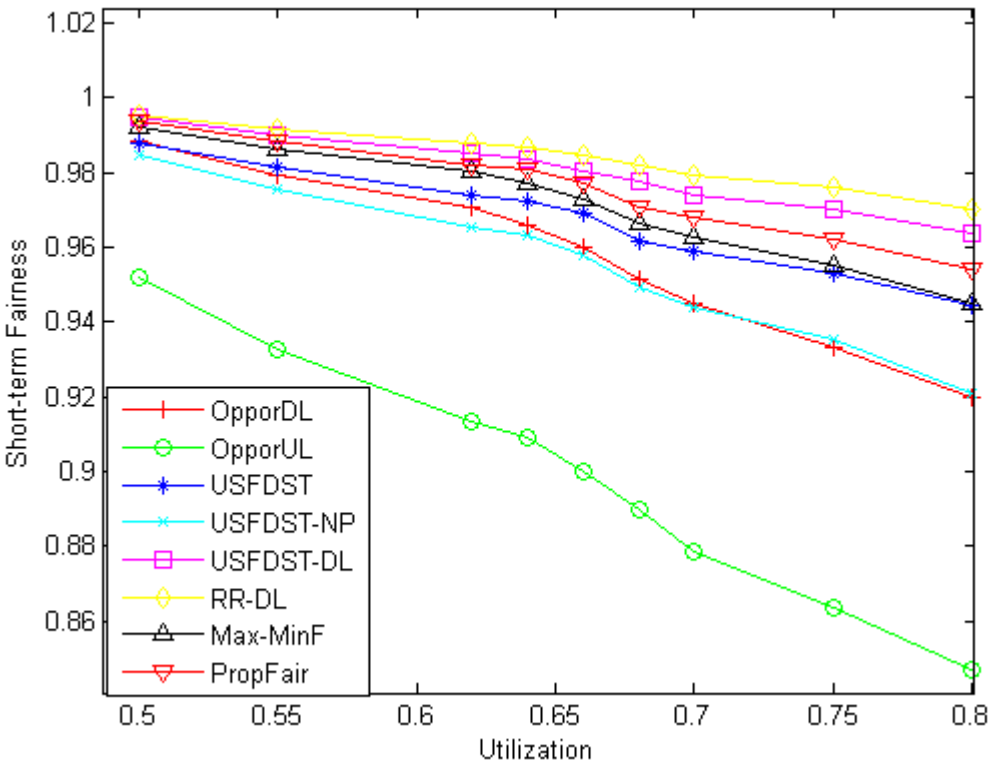

Figure 4.24: Short Term fairness of Algorithms

Figure 4.24 shows short-term fairness of the algorithms according to the metric given in Equation (4.11). USFDST algorithm achieves higher short-term fairness than UFDST-NP because of the lack of priority mechanism which favors some users for some time intervals. When the network is congested, queues of users with low transmission rates start to expand, and the priority mechanism increases the bandwidth allocation of these users resulting in a degradation in short term fairness. As expected the round-robin scheduling algorithm achieves the highest short-term fairness.

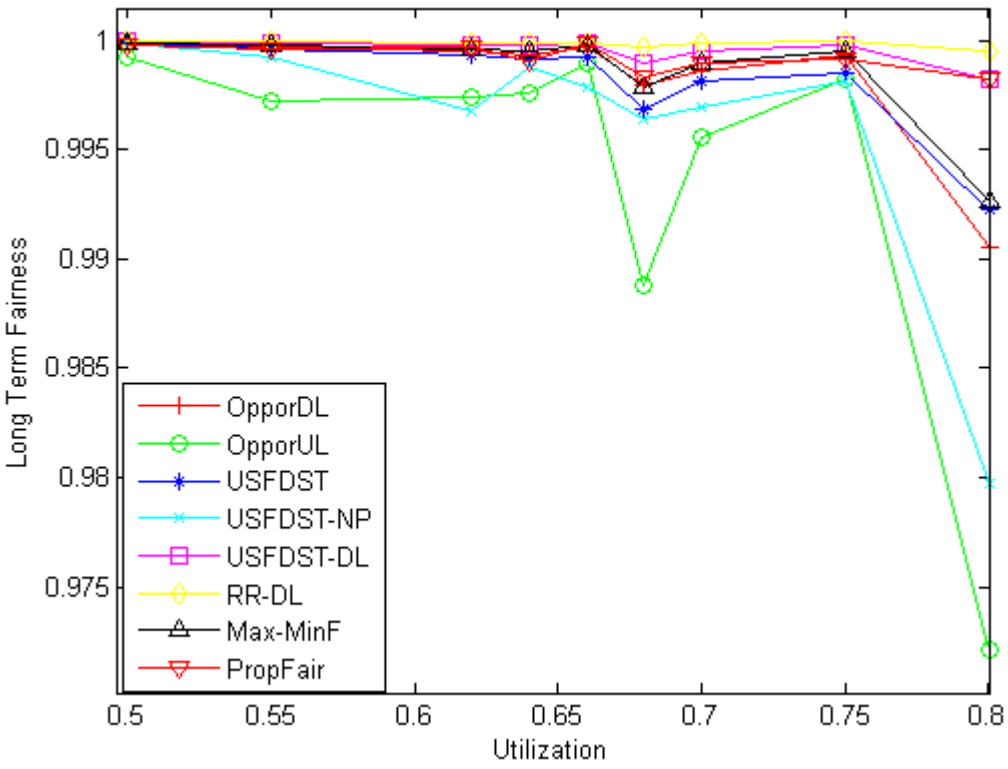

Figure 4.25: Long Term fairness of Algorithms

Figure 4.25 shows long term fairness vs. utilization of the network. Simulation time is long enough, therefore, there is not significant differences between long term fairness of the algorithms. Each algorithm achieves a long-term fairness metric close to 1.

4.2.3 Results for $p = 0.4$ one bandwidth request

In this section, we compare our algorithm against the algorithms which send only one bandwidth request. For greedy algorithm, users send only greedy request of USFDST to the scheduler and for conservative algorithm, users send conservative request of USFDST to the scheduler.

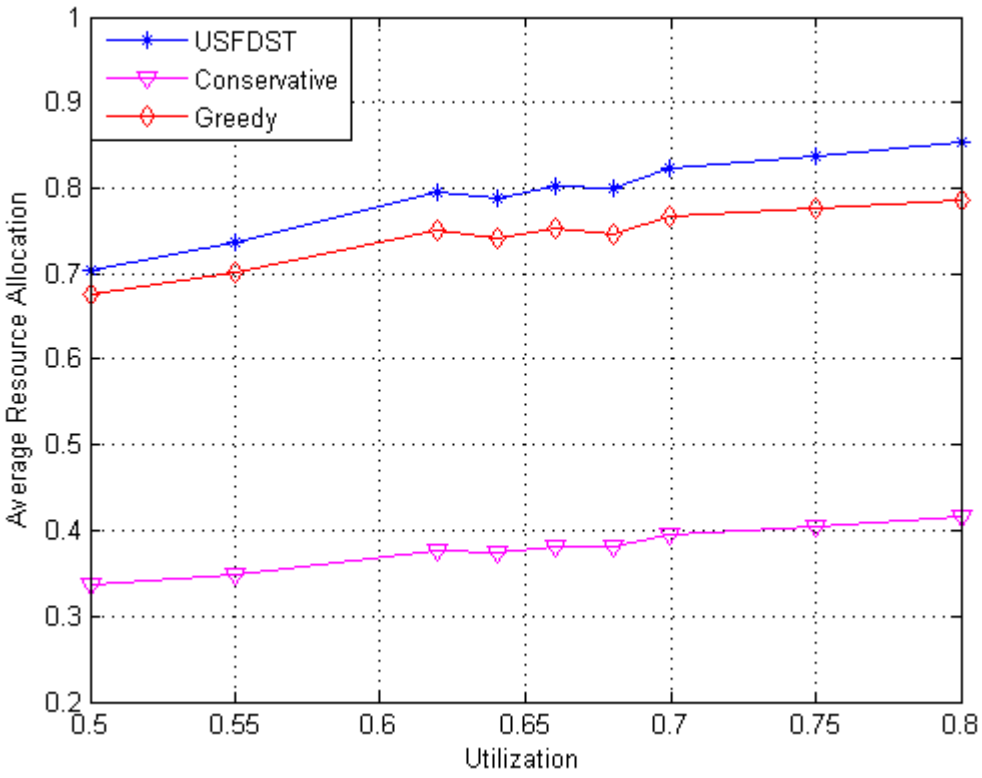

Figure 4.26: Average Resource Allocation

The average resource allocations achieved by algorithms are plotted in Figure 4.26. We observe that USFDST can fill the total bandwidth more accurately than the other two algorithms. In high traffic load cases, the difference between USFDST and the greedy algorithm is 7%. USFDST can fill the total bandwidth at least twice as better as conservative algorithm.

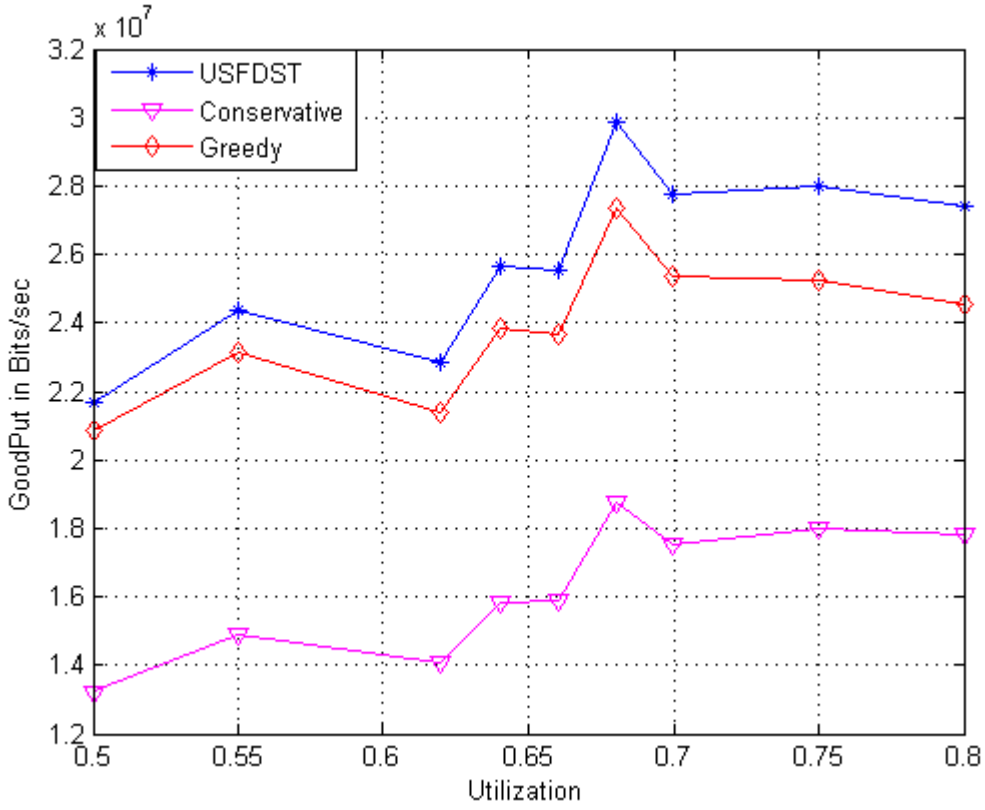

Figure 4.27: Goodput vs. Traffic Load of Network

As expected, USFDST algorithm works better than both greedy and conservative algorithms because it sends two bandwidth requests instead of one, it will increase the flexibility of scheduler. In Figure 4.27, total goodput of the system is plotted for these three algorithms. USFDST can reach 152% of the total goodput of the conservative algorithm and 113% of greedy algorithm for highly loaded cases. For low traffic load cases, conservative algorithm works worse than greedy algorithm because assigning greedy request of a user is more likely for low traffic load.

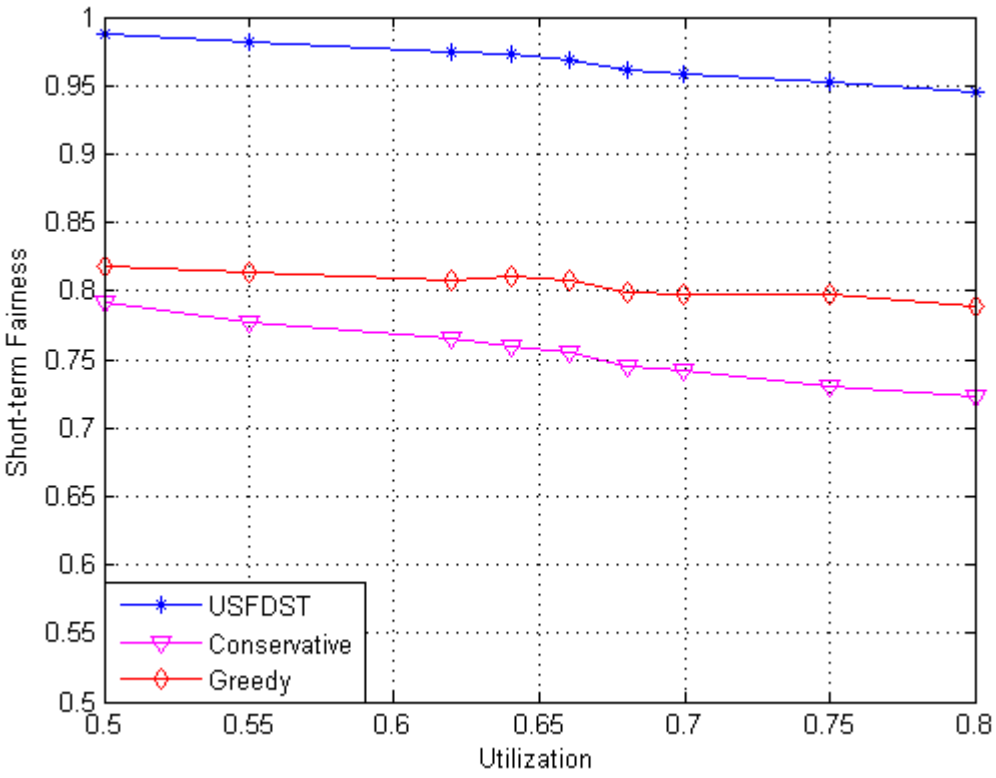

Figure 4.28: Short Term fairness of Algorithms

Figure 4.28 shows short-term fairness of the algorithms according to the metric given in Equation (4.11). USFDST algorithm achieves higher short-term fairness than both conservative and greedy algorithms. Both conservative and greedy algorithms have really bad results for short-term fairness, because they send only one bandwidth request. Scheduler either grant or reject this request, so for some of the frames, user get zero allocation and this affects short-term fairness negatively.

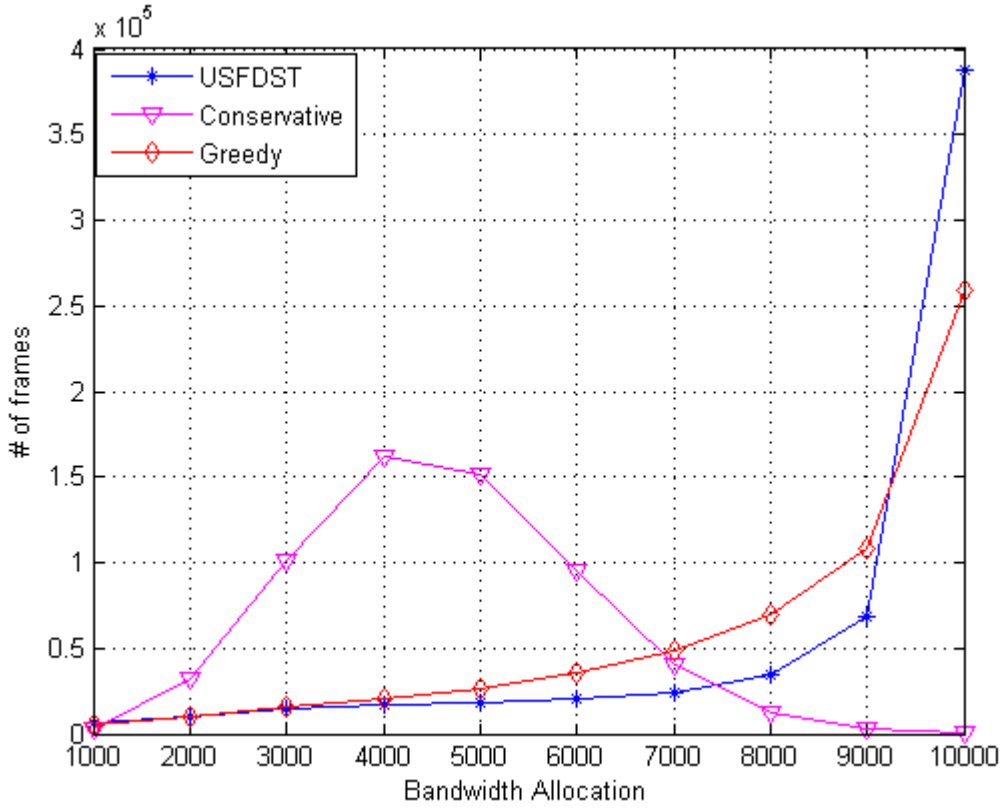

Figure 4.29: Bandwidth Allocation of algorithms for Traffic Load is 0.75 and $p = 0.4$

In Figure 4.29, bandwidth allocation of algorithms are plotted of $\rho = 0.75$ traffic load and $p = 0.4$. We observe that USFDST can fill more than 90% of frames for half of the simulation. On the other hand, for conservative algorithm, it cannot allocate bandwidth accurately, 90% bandwidth allocation can be achieved for only small portion of frames.

4.2.4 Results for $p = 0.8$ one bandwidth request

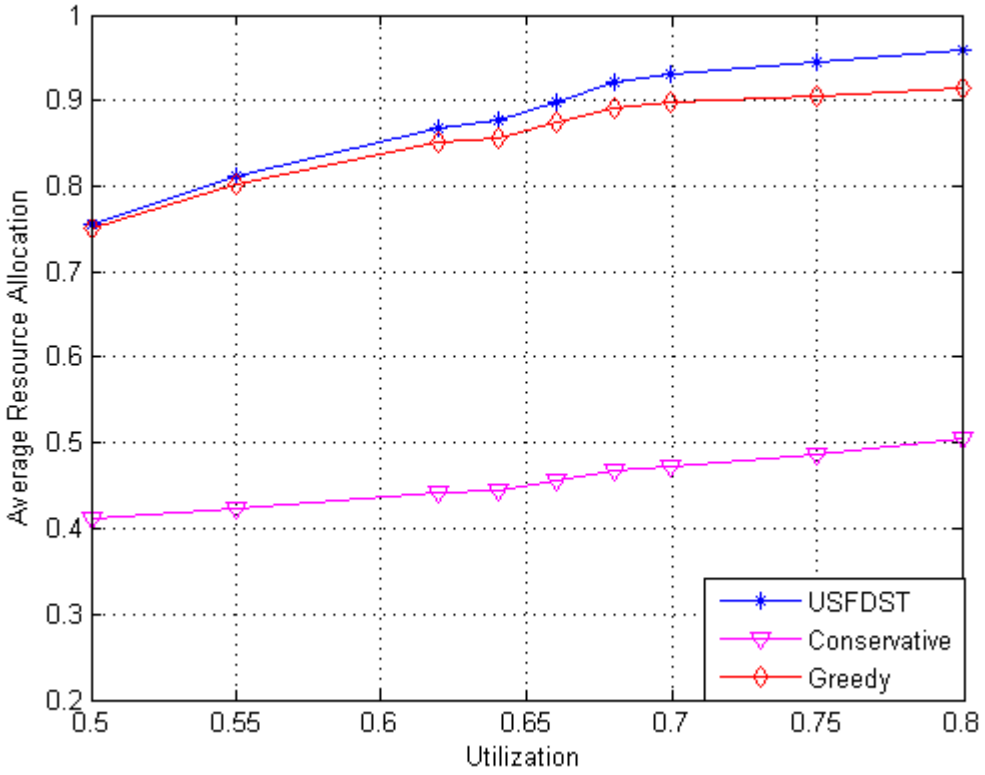

Figure 4.30: Average Resource Allocation

The average resource allocations achieved by algorithms are plotted in Figure 4.30. We observe that USFDST can fill the total bandwidth more efficiently than the other two algorithms. It can reach higher average resource allocation. In high traffic load cases, the difference between USFDST and the greedy algorithm is 4%. USFDST can reach 190% of resource allocation of conservative algorithm. For $p = 0.8$, the difference between USFDST and greedy algorithm decreases because correlation between requests increases.

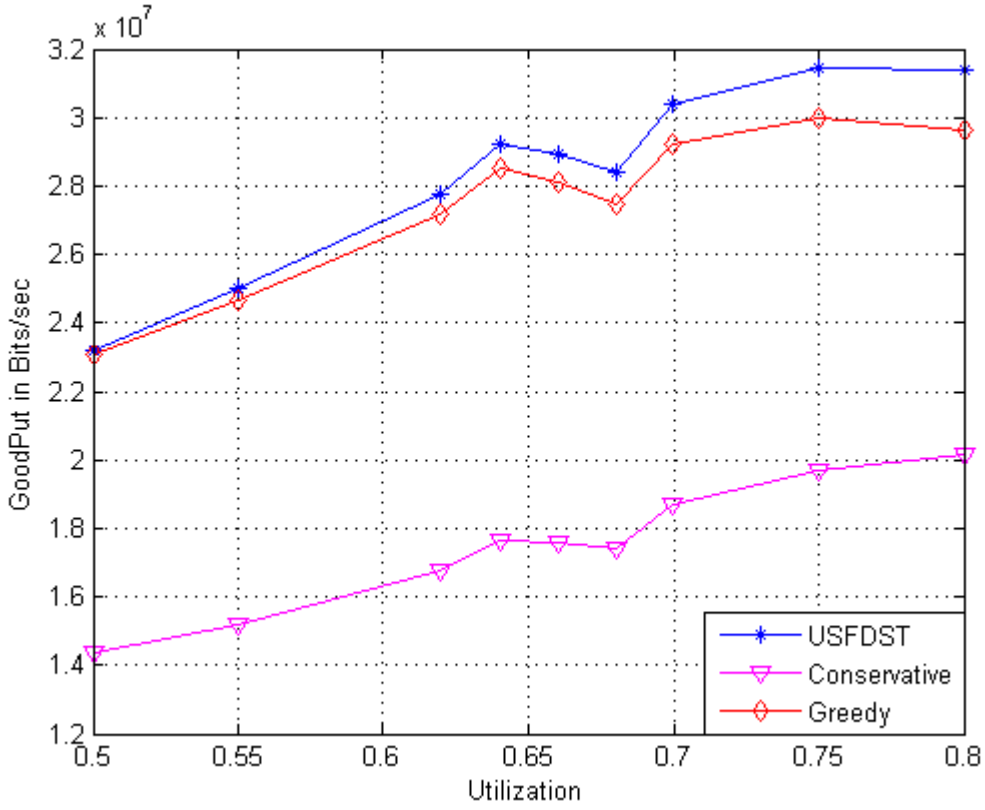

Figure 4.31: Goodput vs. Traffic Load of Network

USFDST algorithm can reach higher goodput results than both greedy and conservative algorithms. In Figure 4.31, total goodput of the system is plotted for these three algorithms. USFDST can reach 155% of the total goodput of the conservative algorithm and 106% of greedy algorithm for highly loaded cases.

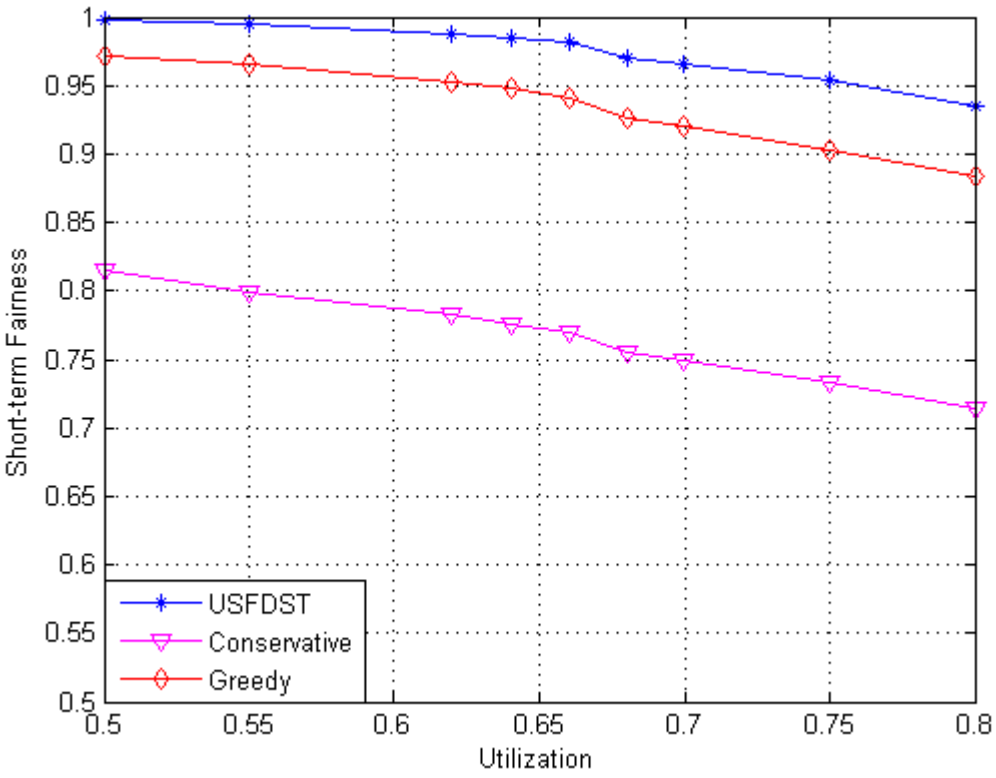

Figure 4.32: Short Term fairness of Algorithms

Figure 4.32 shows short-term fairness of the algorithms according to the metric given in Equation (4.11). USFDST algorithm achieves higher short-term fairness than both conservative and greedy algorithms.

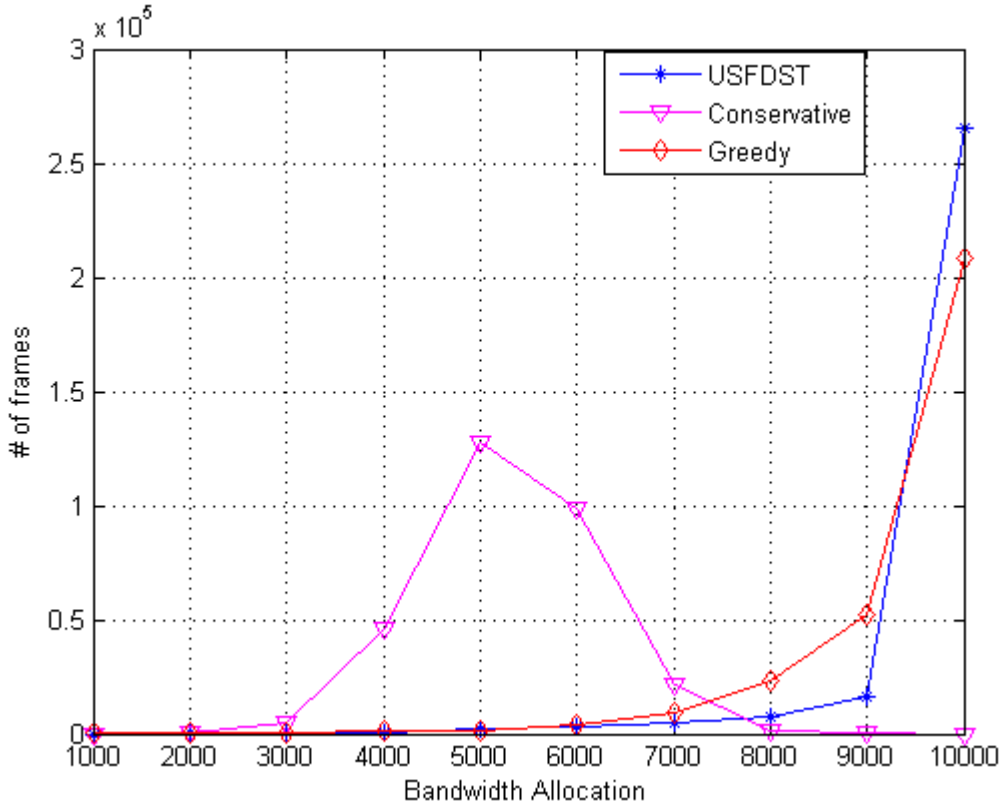

Figure 4.33: Bandwidth Allocation of algorithms for Traffic Load is 0.75 and $p = 0.8$

In Figure 4.33, bandwidth allocation of algorithms are plotted of $\rho = 0.75$ traffic load and $p = 0.8$. We observe that USFDST can fill more than 90% of frames for half of the simulation. On the other hand, for conservative algorithm, it cannot allocate bandwidth accurately and there aren't any frames which is allocated more than 90%.

Chapter 5

CONCLUSIONS

In this thesis, centralized uplink scheduling problem for delay sensitive traffic is studied in broadband wireless networks. For delay sensitive applications, total goodput of the system is more important than total throughput of the system since packets with excessively delays do not provide any service for the application. Therefore, our primary aim is to improve the total goodput of the network and to allocate bandwidth as efficiently as downlink algorithms.

Our scheduling algorithm is one of the best in terms of short term fairness among the scheduling algorithms considered. This is provided by the priority mechanism. Users with low transmission rates does not starve for bandwidth once they switched to high priority and the scheduler assigns more resources to these users until their queue lengths drop below a threshold.

For delay sensitive applications, sending packet with no delay or sending it with a delay less than maximum latency has no difference from the application point of view. We benefit from this property in our scheduling algorithm. We decrease the number of zero delayed packets at the cost of increased number of

packets with nonzero, but less than the maximum allowed delays.

Our scheduling algorithm is designed for an uplink scheduling traffic, so it has partial information about MSs queue. However despite of this disadvantage, the proposed USFDST algorithm can utilize the resources with an efficiency of 95% ($p = 0.4$) and 98% ($p = 0.8$) for more bursty and less bursty traffics respectively. Other uplink scheduling algorithm (i.e. OpporUL) works distinguishable worse than USFDST. For $p = 0.4$ and traffic load $\rho = 0.8$, while OpporUL can fill the 69% of resources on average, USFDST reaches 85.3% resource allocation on average which is 28% higher than the OpporDL.

USFDST works better for less bursty traffics. It utilizes more than 95% of resources for 78% of the frames when traffic load $\rho = 0.75$ and $p = 0.8$. On the other hand, it can reach only 58% of %95 of resource allocation when traffic load $\rho = 0.75$ and $p = 0.4$.

Our proposed scheduling algorithm is designed for delay sensitive applications, therefore it is trying to improve total goodput of the system. In low traffic loads, the total goodput difference between USFDST and downlink algorithms are less than 1%. When the traffic load increases, the difference between USFDST and the most maximizing goodput algorithm (i.e. Max-MinF) is less than 3% when $p = 0.8$. When $p = 0.4$, the difference grows, but it is not more than %5. Although our scheduling algorithm is designed for uplink scheduling, it achieves better results for total goodput than a downlink scheduling algorithm round robin.

Our proposed scheduling algorithm can reach 107% of the total goodput of the greedy algorithm and 200% of the total goodput of the conservative algorithm. Our proposed scheduling algorithm can utilize the resources with efficiency 152% and 113% of conservative and greedy algorithm respectively. The proposed method performs remarkably better than these two algorithms in terms of short term fairness.

On possible future research direction is a hybrid scheduling algorithm where when the network is heavily congested, scheduling algorithm can cancel priority mechanism and assign bandwidth without it or assign bandwidth opportunistically. Another problem that may require future work is investigation the effects of fragmentation on the performances of the uplink scheduling algorithms.

Bibliography

[1] http://www.prnewswire.com/news-releases/worldwide-telecommunications-industry-revenue-to-reach-27-trillion-by-2017-says-insight-research-corp-136622263.html.

[2] W. Nie, H. Wang, and N. Xiong, "Low-overhead uplink scheduling through load prediction for wimax real-time services," *Communications, IET*, vol. 5, pp. 1060 –1067, may 2011.

[3] H. Lee, T. Kwon, and D.-H. Cho, "An efficient uplink scheduling algorithm for voip services in ieee 802.16 bwa systems," in *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, vol. 5, pp. 3070 – 3074 Vol. 5, sept. 2004.

[4] H. Rath, A. Bhorkar, and V. Sharma, "An opportunistic drr (o-drr) uplink scheduling scheme for ieee 802.16-based broadband wireless networks," in *International Conference on Next Generation Networks (ICNGN), Mumbai, India*, pp. 1 –5, 9-11 Feb 2006.

[5] K. Wongthavarawat and A. Ganz, "Packet scheduling for qos support in ieee 802.16 broadband wireless access systems," *International Journal of Communication Systems*, vol. 16, no. 1, pp. 81–96, 2003.

[6] P. Rengaraju, C.-H. Lung, and A. Srinivasan, "Qos assured uplink scheduler for wimax networks," in *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, pp. 1 –5, sept. 2010.

[7] H. Rath, A. Bhorkar, and V. Sharma, "Nxg02-4: An opportunistic uplink scheduling scheme to achieve bandwidth fairness and delay for multiclass traffic in wi-max (ieee 802.16) broadband wireless networks," in *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, pp. 1 –5, 27 2006-dec. 1 2006.

[8] N. A. Ali, P. Dhrona, and H. Hassanein, "A performance study of uplink scheduling algorithms in point-to-multipoint wimax networks," *Computer Communications*, vol. 32, no. 3, pp. 511 – 521, 2009. Adaptive Multicarrier Communications and Networks.

[9] J. Lin and H. Sirisena, "Quality of service scheduling in ieee 802.16 broadband wireless networks," in *Industrial and Information Systems, First International Conference on*, pp. 396 –401, aug. 2006.

[10] X. Meng, "An efficient scheduling for diverse qos requirements in wimax," Master's thesis, University of Waterloo.

[11] V. Singh and V. Sharma, "Efficient and fair scheduling of uplink and downlink in ieee 802.16 ofdma networks," in *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, vol. 2, pp. 984 –990, april 2006.

[12] W. Tranter, K. Shanmugan, T. Rappaport, and K. Kosbar, *Principles of communication systems simulation with wireless applications*. Upper Saddle River, NJ, USA: Prentice Hall Press, first ed., 2003.

[13] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[14] D. N. C. Tse and P. Viswanath, *Fundamentals of Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[15] J. S. A. Perrig and D. Wagner, "Security in wireless sensor networks," vol. 47, pp. 53 – 57, June 2004.

[16] C.-K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. Prentice Hall PTR, 2001.

[17] `http://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_17/docs/PDFs/R1-00-1395.pdf`.

[18] M.-S. Alouini and A. J. Goldsmith, "Adaptive modulation over nakagami fading channels," *Wirel. Pers. Commun.*, vol. 13, pp. 119–143, May 2000.

[19] C. Wengerter, J. Ohlhorst, and A. von Elbwart, "Fairness and throughput analysis for generalized proportional fair frequency scheduling in ofdma," in *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, vol. 3, pp. 1903 – 1907 Vol. 3, may-1 june 2005.

[20] C. Barrett, M. Marathe, D. Engelhart, and A. Sivasubramaniam, "Analyzing the short-term fairness of ieee 802.11 in wireless multi-hop radio networks," in *Modeling, Analysis and Simulation of Computer and Telecommunications Systems, 2002. MASCOTS 2002. Proceedings. 10th IEEE International Symposium on*, pp. 137 – 144, 2002.

[21] G. Berger-Sabbatel, A. Duda, M. Heusse, and F. Rousseau, "Short-term fairness of 802.11 networks with several hosts," in *Proceedings of the Sixth IFIP TC6/WG6.8 Conference on Mobile and Wireless Communication Networks (MWCN 2004)*, vol. 162 of *IFIP International Federation for Information Processing*, (Paris, France), pp. 263–274, Springer Boston, Oct. 25–27, 2004.

[22] A. Pantelidou and A. Ephremides, "Scheduling in wireless networks," *Found. Trends Netw.*, vol. 4, pp. 421–511, Apr. 2011.

[23] N. Joshi, S. R. Kadaba, S. Patel, and G. S. Sundaram, "Downlink scheduling in cdma data networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, (New York, NY, USA), pp. 179–190, ACM, 2000.

[24] E. Yaacoub and Z. Dawy, "A survey on uplink resource allocation in ofdma wireless networks," *Communications Surveys Tutorials, IEEE*, vol. 14, pp. 322 –337, quarter 2012.

[25] J. So, H.-C. Jeon, and D. Ann, "Joint proportional fair scheduling for uplink and downlink in wireless networks," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pp. 1 –4, may 2011.

[26] D. N. C. Tse, "Multiuser diversity in wireless networks," Apr. 2001.

[27] D. Bertsekas and R. Gallager, *Data Networks*. Prentice-Hall, 2001.

[28] J. Nagle, "On packet switches with infinite storage," *Communications, IEEE Transactions on*, vol. 35, pp. 435 – 438, apr 1987.

[29] 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 vol.3, 2000.

[30] P. K. et al., "Winner ii interim channel models," in *IST-4-027756 WINNER II D1.1.1 V1.1*, nov 2006.

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

[32] A. J. R. Joao Pedro Eira, "Analysis of wimax data rate performance," nov. 2007.

[33] W. Fischer and K. Meier-Hellstern, "The markov-modulated poisson process (mmpp) cookbook," *Performance Evaluation*, vol. 18, no. 2, pp. 149 – 171, 1993.